KU LEUVEN

ARENBERG DOCTORAL SCHOOL

FACULTY OF ENGINEERING SCIENCE

Model reduction of contact problems in flexible multibody dynamics

With emphasis on dynamic gear contact problems

Bart Blockmans

Supervisor:
Prof. dr. ir. W. Desmet

Co-supervisor:
Dr. ir. T. Tamarozzi

Dissertation presented in partial fulfilment of the requirements for the degree of Doctor of Engineering Science (PhD):
Mechanical Engineering
September 2018

In beloved memory of
Jan Roosen

Model reduction of contact problems in flexible multibody dynamics

With emphasis on dynamic gear contact problems

Bart BLOCKMANS

Examination committee:
Prof. dr. ir. O. Van der Biest, chair
Prof. dr. ir. W. Desmet, supervisor
Dr. ir. T. Tamarozzi, co-supervisor
Prof. dr. ir. K. Meerbergen
Dr. ir. F. Naets
(University of Leuven)
Prof. dr. ir. O. Brüls
(University of Liège)
Prof. dr. ir. D. Rixen
(Technical University of Munich)

Dissertation presented in partial fulfilment of the requirements for the degree of Doctor of Engineering Science (PhD): Mechanical Engineering

September 2018

Uitgegeven in eigen beheer, Bart Blockmans, Celestijnenlaan 300, B-3001 Leuven (Belgium)

Acknowledgements

This PhD thesis contains the most important results obtained during the five years of research that I conducted at the University of Leuven. It goes without saying that this undertaking wouldn't have been possible without the support and encouragements of the people in my environment, both at and outside of work *.

Eerst en vooral wil ik mijn promotor, Prof. Wim Desmet, bedanken om mij de kans te geven uitdagend doctoraatsonderzoek te verrichten en deel uit te maken van de geweldige onderzoeksgroep die de *Noise and Vibration Research Group* is. Vanaf het begin van mijn doctoraatstraject heb ik me bevoorrecht gevoeld door de vrijheid die je me gaf en het vertrouwen dat je in me leek te hebben zonder dat ik er – zo voelde het alleszins – veel voor in de plaats moest doen. Ik waardeer in het bijzonder de gesprekken waarin je jouw visie op mijn werk en op de onderzoeksgroep in het algemeen met me deelde: bruisend van enthousiasme verliet ik telkens jouw bureau en hongerig zette ik me aan het werk om mijn onderzoek nog te verbeteren.

Tommaso, I never asked myself what this PhD thesis would have looked like if I wouldn't have had your guidance and support – simply because I don't *want* to think about that. You drew my attention to model reduction, sparked my interest for dynamic contact problems, and continuously inspired me with your ideas. You promoted my work both in and outside of the university long before you officially became my co-promoter. Thank you for the many valuable lessons learned while working with you, and for the good times we had.

I would like to express my sincere gratitude to the remaining members of my supervisory and examination committees. Prof. Johan Meyers, I highly value the interactions we had during the first years of my PhD: you broadened my horizon, inspired me technically, and helped me shape my research trajectory. Prof. Olivier Brüls, Prof. Karl Meerbergen, Prof. Daniel Rixen and Frank, thank you for critically reviewing my manuscript and for making valuable suggestions that improved the quality of this thesis and allowed me to identify future research directions. It is an honor to have such a highly-regarded committee of scientists reading and thinking about my work. Frank, I want to additionally thank you for your constructive feedback throughout the years, and for being such an inspirational member of our research group: I am looking forward to intensify our collaboration in the upcoming months and years.

Jakob, wat als ik het hogervermelde gedachtenexperiment alsnog uitvoer, zij het met jou in de verwijderde rol? Het is niet ondenkbaar dat mijn manuscript er dan wat de grote lijnen betreft gelijkaardig had uitgezien. Alleen: veel en veel minder goed. Jij bracht het beste in mij naar boven, en ik ben dankbaar voor de fijne jaren die we als *partners in crime* op het bureau doorbrachten. Je

bent een van de beste onderzoeksingenieurs waar ik ooit mee mocht samenwerken, en ik hoop van harte dat onze professionele paden in de toekomst opnieuw zullen kruisen.

Niccolò, it would be stating the obvious if I wrote that rather than drastically improving the quality of my PhD, you made doing a PhD a lot more fun. Because indeed, you lighten up the atmosphere wherever you go and it was a pleasure having you around as a good colleague and friend. But truth be told, I never heard you talk as passionate about women and Italian beaches as you did about the singular value decomposition and model reduction. Thank you for making me think things through when I first took them for granted, and for instigating me with your entrepreneurial spirit and energy.

Italian craftsmanship has produced even more excellent young men: Rocco and Francesco, thank you for your kindness and generosity, it was and is a pleasure working with you.

Claus, bedankt dat ik met al mijn praktische, persoonlijke en filosofische vragen vanaf dag één zonder aarzeling bij jou terecht kon. Jan Croes, bedankt voor de vele nuttige adviezen over het beheer van projecten, tijd en het leven zelf, en voor de aanstekelijke lachsalvo's. Bert Pluymers, bedankt om mij steeds met raad en daad bij te staan in de voorbereiding van het leven in onze groep na mijn doctoraat. Gert Heirman, bedankt om me tijdens de begeleiding van mijn masterproef warm te maken voor methodologisch onderzoek in de ingenieurswetenschappen, en voor de inspirerende interacties die we hadden tijdens het eerste jaar van mijn doctoraat. Bedankt Joram, Pepijn, Brecht en Bart Forrier voor de steun en luisterende oren tijdens de iets moeizamere momenten van mijn doctoraatstraject, en voor de aangename momenten daarbuiten. Martijn, Ward, Daniël, Jelle, Simon & Simone, en de vele andere collega's van "de nieuwe generatie": telkens een van jullie ons labo vervoegde realiseerde ik me opnieuw hoe gelukkig ik me mag prijzen deel uit te maken van een groep van zo'n knappe koppen; bedankt om er te zijn, en voor jullie ongezien grote collegialiteit.

Doctoraatsonderzoek is een serieuze zaak, maar de boog kan niet altijd gespannen staan. Ik weet niet of dit werk me gelukt zou zijn indien ik de differentiaalvergelijkingen en modelreductieschema's niet op regelmatige basis had kunnen inwisselen voor de eenvoud van de fiets en de mensen in wiens gezelschap ik dit effectiefste aller voertuigen berijd: bedankt aan de vrienden van wielerclub Blosom Meerbeek en Racing Team Inseguitori Leuven voor de meer dan welgekomen afleiding tijdens onze gezamenlijke fietstochten, bergvakanties en wedstrijden.

Mama en papa, als een doctoraatsonderzoek tot een goed einde brengen al een grote onderneming mag heten, dan is als zoon je ouders willen bedanken voor de gegeven kansen helemáál onbegonnen werk. Ik kan alleen maar hopen dat jullie aan een half woord genoeg hebben, en me

voorts beperken tot deze meer bevattelijke dankbetuiging: bedankt voor jullie onophoudelijke steun en aanmoedigingen, jullie goede zorgen en jullie oprechte interesse in mijn onderzoek.

Bobonne, Grote Peter, het is me gelukt, ik ben doctor geworden! Neen, niet in Gasthuisberg: ik genees ernstig zieke computersimulaties, weten jullie nog? Het doet me pijn dat de muren aan jullie kant van m'n huis koud zijn geworden voordat ik jullie deze tekst kon laten zien. Maar het troost me te weten dat hij niet de minste invloed gehad zou hebben op hoe jullie over mij dachten. Bedankt voor jullie onvoorwaardelijke liefde en steun, ook tijdens de drukste en meest afwezige maanden van mijn doctoraat.

Nonkel Jan, hoewel het tragische omstandigheden waren die ertoe noopten, houdt het volkomen steek dat jij de eerste was die deze tekst las. Jij was de reden dat ik voor de ingenieurswetenschappen koos, en deze tekst is het resultaat van de diepgaande invloed die je op me had tijdens mijn studies en onderzoek. Alleen al daarom blijf ik je voor altijd dankbaar.

Bedankt aan de overige leden van mijn familie voor hun aanmoedigingen en medeleven, in het bijzonder tijdens de laatste fase van mijn doctoraat: Katrien, Peter, Ilse, bomma en bompa, Reinhilde en William, mijn schoonzussen en schoonbroers, mijn tantes en nonkels, mijn neef Tom (hierna is 't aan jou, hé!) en mijn vele lieve nichten.

Tot slot wil ik de vrouw bedanken op wiens steun ik al ruim een half leven lang iedere dag mag rekenen. Hanne, ik heb de laatste maanden bijzonder veel van je gevraagd. Zelfs zonder het te moeten vragen: zoals steeds wist je wat ik nodig had voordat ik het zelf besefte, en je versaagde niet wanneer dagen weken en weken maanden werden. Bedankt voor alles, zonder jou was het me niet gelukt.

Bart Blockmans
September 2018

* Needless to say, financial support is just as vital to a PhD as the mental and technical support. I would like to thank the KU Leuven Research Fund for supporting the IDO project that allowed me to investigate drivetrain modelling techniques while interacting with researchers of various scientific backgrounds. This work also benefited from the Dynamical Systems, Control and Optimization (DYSCO) network that was launched by the Belgian Science Policy Office (BELSPO) in the context of the Interuniversity Attraction Poles programme. Finally, I would like to thank the Agency of Innovation and Entrepreneurship (VLAIO) and Flanders Make for their support in numerous R&D projects.

A special thanks goes out to the industry representatives who stuck out their necks and supported my research work through bilateral projects, well-aware of the risks involved in ongoing research and compassionate when things didn't go as smooth as planned: thank you Frederik Vanhollebeke, Ben Marrant, Joris Peeters, Kris Smolders, Stijn Pittois, Dieter Laes and Yann Vonderscher.

Summary

The research presented in this dissertation is concerned with the efficient solution of dynamic contact problems by means of distributed-parameter methods. Such problems are encountered in the computer-aided design of mechanical systems, where contact interactions are ubiquitous and often vital to the proper functioning of the system. With weight reduction and increased operating frequencies among the primary objectives of present-day engineering, a dynamic description of these problems becomes increasingly important.

Distributed-parameter methods such as the Finite Element (FE) method are often combined with the Flexible Multibody Dynamics (FMBD) approach to construct mathematical models of the interacting components when their geometries are complex and/or deformable. These models however come at a high computational cost due to three main reasons: the high number of Degrees Of Freedom (DOFs) involved, the large number of time increments needed to numerically solve the time-dependent equilibrium equations, and finally the high number of Floating Point Operations (FLOPs) required to evaluate the nonlinear contact forces.

In this dissertation an attempt is made to mitigate the computational burden of dynamic contact problems by tackling all three of these issues. While the methods and procedures discussed in this dissertation apply to a broad class of contact problems, the dynamic gear contact problem is selected as a common theme throughout the text.

The first and central contribution of this dissertation is the development of a novel Model-Order Reduction (MOR) method that is specifically tailored to the solution of FE-based contact problems in FMBD. The method belongs to the class of projection-based MOR methods and can be seen as a parametric variant of the Component Mode Synthesis (CMS) approach. By judiciously choosing the ingredients of the reduction basis, the method is shown to reduce the number of DOFs in the full-order model with several orders of magnitude, far beyond what is attainable using classical CMS. In comparison to the former, the computational complexity of the method is shown to be two orders of magnitude smaller while maintaining similar levels of accuracy. The method is first developed and validated for the perfectly-aligned gear contact problem, and then extended to the case of spatially misaligned gear pairs and multi-mesh gear trains.

The second contribution of this dissertation is the systematic investigation of numerical integration schemes for dynamic contact problems in light of the novel MOR method. Owing to the significant dimensional reduction that is enabled by this method, both explicit and implicit integrators are brought to the fore. The optimal integration scheme is selected by comparing the performances of a number of well-established adaptive integrators in solving the reduced-order gear contact problem. A number of the method's distinguishing features are used to further optimize the selected integration scheme, including the availability of an analytical generalized contact force Jacobian and the clustering of the quasi-static contact deformations into a comparatively small subset of shape vectors. Compared to the fixed-increment central difference

integrator that is used as a reference integrator in this dissertation, the selected integration schemes enable a reduction of overall computing time with a factor of 20 and more.

Having reduced the number of DOFs and optimized the numerical integration scheme, up to 98% of the computing time is spent on evaluating the FE-based contact forces. The cost of this operations scales quadratically with the number of finite elements used to discretize the contact surfaces, thereby partially jeopardizing the order reduction that is brought about by the novel MOR method. In order to remove the dependence of the ROM's complexity on the dimensions of the underlying FE model, the application of two well-established hyperreduction schemes to the analysis of dynamic contact problems is investigated. Particular challenges that are dealt with are the development of efficient hyperreduction schemes in the context of sliding contact interactions, and the preservation of Newton's law of action and reaction, which is vital to the robust solution of dynamic contact problems. Using the extended hyperreduction schemes, overall computational gains of roughly one order of magnitude as compared to the non-hyperreduced ROM are observed.

By lowering the number of DOFs, optimizing the numerical integration scheme and reducing the number of FLOPs in the contact force evaluation, this dissertation reduces the overall simulation cost of dynamic contact simulations with three to four orders of magnitude as compared to standard reduction techniques, without jeopardizing the first-principles approach of FE modelling. While the real-time simulation of complex contact interactions remains a challenge yet to be met, the current work moves these kinds of simulations from institutional supercomputing clusters to the mechanical engineer's personal computer.

Samenvatting

Deze verhandeling behandelt de efficiënte oplossing van dynamische contactproblemen aan de hand van eindige-elementenmethoden. Dergelijke problemen komen veelvuldig voor in het computerondersteund ontwerp van mechanische systemen waarin contact-interacties een centrale rol spelen. Voorbeelden van dergelijke systemen zijn de tandwielkasten en lagers in moderne windturbines, de kettingen en tandwielen in onze fietsen, de banden van onze wagens tijdens hun interactie met het wegdek, en de nok-volgersystemen in onze huishoudtoestellen. Vanwege de steeds hogere werkings-frequenties en de toenemende gewichtsreductie van deze systemen in de hedendaagse ingenieurspraktijk, wint de dynamische beschrijving van deze systemen alsmaar aan belang.

Eindige-elementenmethoden laten toe wiskundig handelbare computermodellen op te stellen van flexibele componenten door de geometrie van deze componenten onder te verdelen in een veelheid van eenvoudige elementen met ieder een beperkt aantal vrijheidsgraden. De interactie van de aldus gediscretizeerde componenten met andere componenten in het systeem wordt op elegante wijze beschreven door de flexibele meerlichamendynamica, die de tijdsafhankelijke evenwichtsvergelijkingen van wille-keurige meerlichamensystemen bestudeert. De computergestuurde oplossing van deze vergelijkingen is niet zelden een buitensporig rekenintensieve taak omwille van drie redenen: het hoge aantal vrijheidsgraden in het gediscretizeerde systeemmodel, het hoge aantal tijdstappen benodigd voor de stapsgewijze oplossing van de evenwichts-vergelijkingen, en het hoge aantal computeroperaties dat nodig is voor het berekenen van de niet-lineaire contactkrachten. Het gevolg hiervan is dat dynamische contact-simulaties van industriële mechanismen veelal voorbehouden zijn aan instellingen die over performante rekenclusters, tijd en geld beschikken.

Deze verhandeling heeft als doel de hoge rekenlast van dynamische contactproblemen te reduceren door elk van de voorgenoemde drie punten aan te pakken. Hoewel de methoden en procedures die in deze verhandeling besproken worden toepasbaar zijn op een veelheid van industriële contactproblemen, werd het dynamische tandwielprobleem geselecteerd als gemeenschappelijk thema doorheen de tekst.

De eerste en centrale bijdrage van deze verhandeling is de ontwikkeling van een nieuw modelreductieschema voor de oplossing van contactproblemen in de flexibele meerlicha-mendynamica. De methode behoort tot de projectiegebaseerde reductiemethoden en kan beschouwd worden als een parametrische variant van de *Component Mode Synthesis* (CMS) benadering. Dankzij een nauwkeurige selectie van de ingrediënten van de reductiebasis wordt aangetoond dat de voorgestelde methode in staat is het aantal vrijheidsgraden met meerdere grootteordes te verlagen, ver voorbij wat mogelijk is met de klassieke CMS-benadering. De numerieke complexiteit van de nieuwe methode – uitgedrukt in het aantal benodigde

computeroperaties per tijdstap – is ongeveer twee grootteordes kleiner dan de complexiteit van de CMS-benadering, terwijl de nauw-keurigheid nagenoeg behouden blijft.

De tweede bijdrage van deze verhandeling is een systematisch onderzoek van de numerieke methoden die beschikbaar zijn voor de oplossing van de evenwichts-vergelijkingen van de contactdynamica, rekening houdend met de specifieke karakter-istieken van de nieuwe modelreductiemethode. Na identificatie van het optimale oplossingsschema worden de voorgenoemde karakteristieken aangewend om het geselecteerde schema verder te verbeteren. De benodigde rekentijd voor het oplossen van de dynamische evenwichtsvergelijkingen met behulp van het aldus bekomen schema is 20 maal korter dan de benodigde rekentijd van de veelgebruikte centrale-differentie-methode.

Eenmaal het aantal vrijheidsgraden gereduceerd en het numerieke oplossingsschema geoptimaliseerd zijn, wordt tot 98% van de totale oplossingstijd besteed aan het evalueren van de eindige-elementengebaseerde contactkrachten. De numerieke kost van deze operatie schaalt kwadratisch met het aantal elementen in de ruimtelijke discretisatie van de contactoppervlakken. Om de afhankelijkheid van het gereduceerde model van de dimensies van het onderliggende eindige-elementenmodel te elimineren, werden twee hyperreductiemethoden toegepast op het dynamische contactprobleem. Specifieke uitdagingen die daarbij behandeld dienden te worden zijn de ontwikkeling van een efficiënt hyperreductieschema in de context van glijdende contactinteracties, en het behoud van Newton's wet van actie en reactie, die cruciaal is voor de robuuste oplossing van dynamische contactproblemen. Gebruik makend van het verbeterde hyperreductie-schema werd de totale rekenlast verlaagd met één bijkomende grootteorde in vergelijking met het niet-hypergereduceerde contactmodel.

Door het aantal vrijheidsgraden te verlagen, het numerieke oplossingsschema te optimaliseren en het aantal computeroperaties in de berekening van de contactkrachten te reduceren is deze verhandeling erin geslaagd om de totale rekenkost van dynamische contactsimulaties te verlagen met drie tot vier grootteordes ten opzichte van klassieke reductiemethoden, zonder te raken aan de fundamentele principes van de eindige-elementenmodellering. Hoewel de simulatie van complexe contactinteracties in reële tijd een grote uitdaging blijft, zorgt deze verhandeling ervoor dat dit soort simulaties binnen het handbereik van de werktuigkundig ingenieur en diens persoonlijke computer zijn gekomen.

Contents

List of symbols and abbreviations

General symbols – Latin

A	Action functional
$\boldsymbol{A}$	Rotation matrix
$\boldsymbol{B}$	Matrix expressing relationship between $\dot{\theta}$ and $\boldsymbol{A}$
$\boldsymbol{b}$	Specific body force acting on a material point
$\boldsymbol{b}$	Summed snapshot vector in the ECSW scheme
b	Face width of a gear
$\boldsymbol{C}$	Covariance matrix
$\boldsymbol{c}$	Coefficient vector of the DEIM-based hyperreduced model
$\boldsymbol{D}$	Damping matrix
d	Pitch diameter of a gear
E	Modulus of elasticity
E_r	Elemental reduced mesh indices
$\boldsymbol{e}$	Vector of local truncation errors
$\boldsymbol{e}_{\rho_i}$	ρ_i-th column of the identity matrix
ϵ	Error functional
ϵ_N	Penalty factor
$\boldsymbol{f}$	Vector of generalized forces
$\mathbf{f}_{\mathrm{c}}$	Vector of generalized contact forces
$\mathbf{f}_{\mathrm{ext}}$	Vector of generalized external forces
$\mathbf{f}_{\mathrm{v}}$	Vector of generalized quadratic velocity forces
$\boldsymbol{G}$	Constraint Jacobian
$\boldsymbol{G}$	Matrix expressing relationship between $\dot{\boldsymbol{\theta}}$ and $\boldsymbol{\omega}$
$\boldsymbol{G}$	Elemental snapshot matrix in the ECSW scheme
$\boldsymbol{g}$	Constraint residual
g_N	Normal gap function
h	Dimensions of the hyperreduced model
h	Time increment in a numerical integration scheme
$\boldsymbol{I}$	Identity matrix
$\boldsymbol{J}$	Jacobian matrix
$\boldsymbol{K}$	Stiffness matrix
L	Length of a shaft

L	Lagrangian of a system
$\boldsymbol{M}$	Mass matrix
$\mathcal{M}$	Matrix manifold
m	Mass
m_n	Normal modulus of a gear
N	Number of generalized coordinates in the FOM
N	Finite element shape function
N	Dimensions of the full-order model
N^i	Finite element shape function corresponding to the *i*-th node
N_b	Number of interface or boundary displacements
N_c	Number of constrained displacements
N_f	Number of unconstrained displacements
N_{fef}	Number of finite elements along the involute curve
N_{few}	Number of finite elements along the width
N_i	Number of internal displacements
$\boldsymbol{n}$	Unit surface normal
n_c	Number of contact shapes in reduction basis
n_f	Number of generalized elastic coordinates
n_m	Number of misaligned contact shapes
n_{ne}	Number of finite element nodes per element
n_r	Number of rigid-body coordinates
n_s	Number of snapshots
n_y	Number of states
n_θ	Number of angular samples
$n_{\boldsymbol{\lambda}}$	Number of constraint equations
$\emptyset$	Empty matrix
$\boldsymbol{P}$	Boolean matrix
p	Interpolation parameter
$\boldsymbol{Q}_c$	Generalized contact force
$\boldsymbol{q}$	Vector of generalized coordinates
q	Load per unit width
$\boldsymbol{q}_f$	Vector of generalized elastic coordinates
$\boldsymbol{R}$	Vector of Cartesian coordinates
$\boldsymbol{r}$	Deformed position vector
$\boldsymbol{r}$	Numerical residual of a system of equilibrium equations
r	Dimensions of the reduced-order model
$\boldsymbol{r}_0$	Undeformed position vector

$\boldsymbol{T}$	Transformation matrix in congruence transformation
T	Kinetic energy of a system
T^1	Torque applied to the driving gear
$\boldsymbol{t}$	Surface traction
t	Time
$\boldsymbol{t}_T$	Vector of tangential stresses
U	Potential energy of a system
$\boldsymbol{u}$	Elastic displacement vector
u	Elastic displacement in the direction of $\mathbf{X_1}$
u	Transmission ratio of a gear pair
$\boldsymbol{u}_b$	Vector of interface or boundary displacements
$\boldsymbol{u}_c$	Vector of constrained displacements
$\boldsymbol{u}_f$	Vector of unconstrained or free displacements
$\boldsymbol{u}_i$	Vector of internal displacements
$\boldsymbol{V}$	Left-basis matrix of the reduced-order model
$\mathcal{V}$	Left-subspace of the reduced-order model
$\mathbb{V}$	Projection matrix physical to generalized point forces
$\boldsymbol{v}$	Vector of generalized velocities
v	Elastic displacement in the direction of $\mathbf{X_2}$
$\boldsymbol{W}$	Right-basis matrix of the reduced-order model
W	Work
$\mathcal{W}$	Right-subspace of the reduced-order model
$\boldsymbol{w}$	Weighting function
w	Elastic displacement in the direction of $\mathbf{X_3}$
$\boldsymbol{X}$	Snapshot matrix
$\boldsymbol{x}$	Position vector of a material point in continuum mechanics
$\boldsymbol{Y}$	Basis matrix of the projection-based hyperreduced model
$\mathcal{Y}$	Subspace of the projection-based hyperreduced model
$\boldsymbol{y}$	State vector
z	Number of gear teeth
z_g	Axial position of a gear on a shaft

General symbols – Greek

α_n	Normal pressure angle of a gear
β	Helix angle of a gear
$\boldsymbol{\Gamma}$	Logarithmically mapped matrix
Γ	Boundary
δ	Variance of a quantity
$\boldsymbol{\varepsilon}$	Cauchy strain tensor
ε	Contact ratio of a gear pair
η	Isoparametric parameter
$\boldsymbol{\theta}$	Vector of rotational parameters
θ	Third Bryant angle
$\boldsymbol{\Lambda}$	Diagonal eigenvalue matrix
$\boldsymbol{\lambda}$	Vector of Lagrange multipliers
λ	Proper orthogonal value
$\boldsymbol{\mu}$	Parameter combination in parametric model reduction
μ	Isoparametric parameter
μ_D	Dynamic friction coefficient
μ_S	Static friction coefficient
ν	Poisson's ratio
ξ	Isoparametric parameter
ξ^e	Weighting factor associated with element *e*
ρ	Material density
ϱ	Radius of curvature at the contact point
$\boldsymbol{\sigma}$	Cauchy stress tensor
τ	Angular pitch of a gear
$\boldsymbol{\Phi}$	Set of retained eigenvectors or POD modes
$\boldsymbol{\varphi}$	Proper orthogonal vector
ϕ	First Bryant angle
$\boldsymbol{\Psi}$	Set of static modes
$\boldsymbol{\Psi}_a$	Set of attachment modes
$\boldsymbol{\Psi}_c$	Set of constraint modes
$\boldsymbol{\psi}$	Static mode
ψ	Second Bryant angle
Ω	Volume
$\boldsymbol{\omega}$	Angular velocity vector
ω	Angular eigenfrequency

Superscripts

e	Refers to the *e*-th element
i	Refers to node number (in chapter 2); refers to the *i*-th body (elsewhere)
j	Refers to the *j*-th node
m	Refers to quantities associated with the master body
s	Refers to quantities associated with the slave body
t	Refers to the *t*-th engaging tooth pair
u	Refers to unassembled vector or matrix quantities

Subscripts

A	Refers to the initial point of tooth engagement
a	Refers to attachment modes
b	Refers to bearing quantities
c	Refers to contact shapes
d	Refers to dependent coordinates
E	Refers to the final point of tooth engagement
f	Refers to quantities associated with the flexible DOFs of a body
FE	Refers to quantities belonging to the finite element
i	Refers to independent coordinates
j	Refers to the *j*-th angular configuration
k	Refers to the current iteration
n	Refers to the current time step
p	Refers to the order of convergence of a numerical integration scheme
s	Refers to static modes

Operators and accents

$\hat{\blacksquare}$	Prescribed quantity
$\bar{\blacksquare}$	Quantities expressed with respect to body-attached reference frame
$\blacksquare : \blacksquare$	Tensor product
$\lvert\blacksquare\rvert$	Absolute value
$\dot{\blacksquare}$	First derivative with respect to time

$\ddot{\blacksquare}$	Second derivative with respect to time
$\dddot{\blacksquare}$	Third derivative with respect to time
$\nabla\blacksquare$	Gradient operation
$\blacksquare^T$	Transpose
$\bigcup\blacksquare$	Assembled sum
$E\{\blacksquare\}$	Average value

Abbreviations

ANCF	Absolute Nodal Coordinate Formulation
ATOL	Absolute tolerance
BDF	Backward Differentiation Formula
CAA	Computer Aided Analysis
CAD	Computer Aided Design
CAE	Computer Aided Engineering
CAM	Computer Aided Manufacturing
CD	Central Difference
CMS	Component Mode Synthesis
CPU	Central Processing Unit
CS	Contact Shape
DAE	Differential-Algebraic Equation
DEIM	Discrete Empirical Interpolation Method
DOF	Degree Of Freedom
DTE	Dynamic Transmission Error
ECSW	Energy-Conserving Weighting and Sampling
EOM	Equation Of Motion
ETOL	Error tolerance
FEM	Finite Element Method
FFR	Floating Frame of Reference
FLOP	Floating Point Operation
FOM	Full-Order Model
GCFE	Gear Contact Force Element
GMP	Global Modal Parameterization
ICS	Interpolated Contact Shape
ITCS	Interpolated Tooth Contact Shape
LOA	Line Of Action
LSF	Load Sharing Factor
MBSD	Multibody System Dynamics
MICS	Misaligned Interpolated Contact Shapes

MIMO	Multiple Input Multiple Output
MOR	Model Order Reduction
NLFEM	Non-Linear Finite Element Method
NLMOR	Non-Linear Model Order Reduction
NM	Normal Modes
NNLS	Non-Negative Least Squares
NTS	Node-To-Surface
NURBS	Non-Uniform Rational B-Splines
ODE	Ordinary Differential Equation
PDE	Partial Differential Equation
PECE	Predict-Evaluate-Correct-Evaluate
PMOR	Parametric Model Order Reduction
POD	Proper Orthogonal Decomposition
POM	Proper Orthogonal Modes
POV	Proper Orthogonal Vectors
RK	Runge-Kutta
RMS	Root-Mean-Square
ROB	Reduced-Order Basis
ROM	Reduced-Order Model
RTOL	Relative tolerance
SMS	Static Modes Switching
STE	Static Transmission Error
SVD	Singular Value Decomposition
TCS	Tooth Contact Shape
TE	Transmission Error
TPWL	Trajectory Piecewise-Linear

Chapter 1

Introduction

1.1 Dynamic contact modelling in multibody system design and analysis

The primary objective of mechanical engineering is to design and manufacture marketable products of high quality. Today's industries are increasingly utilizing digital computers in every phase of the design and manufacturing of their products. The field of Computer Aided Engineering (CAE), which comprises Computer-Aided Design (CAD) and Computer-Aided Manufacturing (CAM), has grown tremendously in the last four decades and holds within it the prospect of the fully automated factory. Essential to the successful implementation of CAD in the product lifecycle is the development of reliable and efficient Computer-Aided Analysis (CAA) techniques. These techniques allow the mechanical engineer to simulate the behavior of a product prior to its actual production, thereby substituting traditional laboratory or field tests and reducing the cost and duration of the product lifecycle. In addition, CAA techniques are essential in the generation of a product's digital twin: a computer model of the product that runs in parallel with it for various purposes, including model-based condition monitoring, state estimation, and model-predictive control. Therefore, these techniques are at the heart of what is often referred to as the Fourth Industrial Revolution [1].

This dissertation is concerned with the development of CAA techniques for products of a particular kind, namely mechanical systems whose inner workings rely on contact interactions. Such products are ubiquitous in both industry and everyday life, with examples including the gear trains and rolling element bearings in wind turbine gearboxes, the chain and gears in our bikes, the tire-road and wheel-railway interactions in our automobiles and railway vehicles, and the cam-follower systems in a variety of industrial and household machinery. Among the effects that are associated with mechanical contact interactions are abrupt changes in system velocities, elastic and plastic deformations in and away from the contact zones, stress wave propagation throughout the system, frictional dissipation of energy, and wear. These dynamic phenomena are among the most important and complex to model in mechanical system engineering and their proper analysis remains a challenge to this day.

The branch of mechanics that deals with the dynamic analysis of mechanical systems is called Multibody System Dynamics (MBSD) [2]. A multibody system is defined as a collection of rigid and deformable components that undergo large translational and rotational displacements and that are interconnected by kinematic joints (e.g. a revolute joint, a cardan joint) and possibly some force elements (e.g. a contact force), see Fig. 1.1. The theory of MBSD has its roots in Rigid Body Mechanics [3], that studies the nonlinear motion of undeformable bodies, and Structural Dynamics [4], in which the deformation of (stationary) flexible bodies is the main concern. Similar to Continuum Mechanics [5], MBSD deals with the description of the general motion of flexible bodies, but unlike the former, the focus is on system analysis, efficient computer implementation, and the treatment of constraints (resulting a.o. from kinematic joints). While the rigidity assumption applies to a large variety of multibody systems, flexible MBSD theory [6] is becoming increasingly important due to the ongoing trend of designing mechanical systems that are more lightweight, faster and more energy-efficient. Examples of technical systems whose design and/or analysis nowadays relies heavily on flexible MBSD modelling include wind turbine gearboxes, lightweight robotic manipulators, biomechanical systems, and a variety of systems stemming from the automotive and aerospace industries, including combustion engines and landing gears.

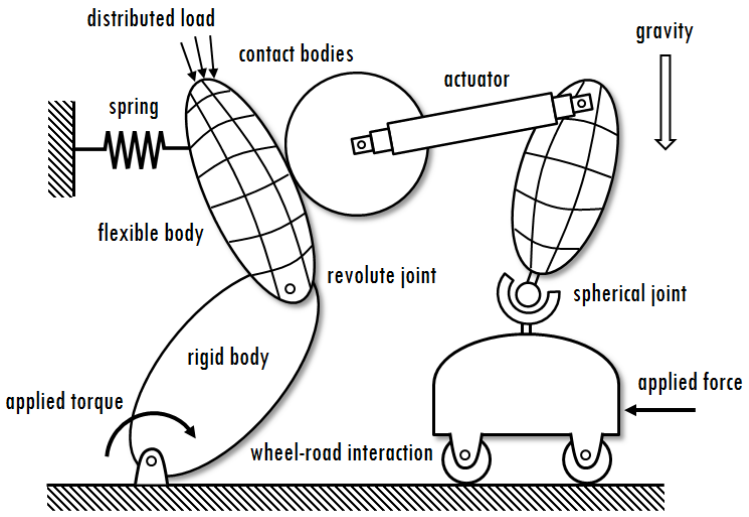


Fig. 1.1 Abstract representation of a multibody system with its most significant components: bodies, joints, and force elements

Contact interactions occur frequently in multibody systems and in many cases the proper functioning of the system relies on them. The classical contact problem is a long-standing and complex issue in engineering mechanics that took a great leap forward since the monumental work of Hertz [7] in 1882. In the past few decades, contact modelling in MBSD has received considerable attention and it remains an active area of research and development to this day. While recent developments such as Mortar methods [8] and NURBS-enhanced contact analysis [9] represented great strides in the quest for increased *accuracy* in computational contact modelling, considerably less progress has been made in increasing its *efficiency*. Nonetheless, it is evident that

if CAA techniques are to be instrumental in such fields as computer-aided design and digital twin development, efficiency is key. Contrary to popular belief, more powerful computers won't resolve the efficiency issue, as Moore's law can't keep up with the engineer's perpetual desire of modelling ever more complex systems. Instead, this dissertation is an attempt to tackle the efficiency issue in computational contact mechanics from the *methodological* point of view.

Since the advent of computational contact mechanics, a large variety of numerical methods for solving contact problems has emerged [10]. By far the most powerful among these methods – and arguably the only serious candidates for CAD – are those that are based on a geometrical discretization of the interacting bodies, namely the Finite Element Method (FEM) [11] and its isogeometric variants [12]. One of the key attributes of the FEM is that it is based on the partitioning of the problem domain into a number of small, non-overlapping subdomains (called the finite elements) over which the solution is approximated by simple local functions (generally polynomials). The finite elements are then interconnected at a finite number of points along the element boundaries (called the *nodes*) and the function values at these nodes are the Degrees Of Freedom (DOFs) of the FE model. In doing so, the FEM allows for an accurate yet mathematically tractable representation of complex geometries – a prerequisite for the accurate solution of every real-life contact problem. However, the high geometrical resolution required by FE-based contact analysis comes at the price of a large number of finite elements and therefore a large number of DOFs (not seldom expressed using 5-digit numbers). These large numbers are detrimental to the efficiency of the method: for each DOF in the model an equilibrium equation needs to be solved, while at the same time the number of Floating Point Operations (FLOPs) required for evaluating the contact forces increases dramatically with the number of finite elements.

In *dynamic* contact analyses, another important contributor to the computational complexity of the method relates to the nature of the contact phenomenon itself. Its main characteristics are very short duration, high force levels, large and abrupt changes in the system velocities, and impulse-like excitations of the system's vibration modes. When integrating the system's equations of motion (i.e. simulating the system's behavior in time by progressively solving a set of dynamic equilibrium equations at a sequence of time instants), these characteristics necessitate the use of time intervals that are comparatively smaller than what is dictated by the accuracy requirements of the analysis, reaching down to the nanosecond time scale. Analyzing a dynamic contact problem using the FEM therefore involves solving *large* systems of *costly* equilibrium equations a *huge* number of times.

This dissertation is aimed at the development of numerical methods for the efficient solution of FE-based contact problems in MBSD. The efficiency issue in computational contact mechanics is tackled from three angles, namely 1) the reduction of the number of DOFs in the FE-representation of the interacting bodies, 2) the reduction of the cost of evaluating the nonlinear contact forces, and 3) the reduction of the number of time steps required to integrate the equations

of motion (see Fig. 1.2). Generally speaking, the methods and procedures developed in this dissertation apply to a broad class of multibody contact problems. Nevertheless some of the peculiarities of the methods depend on the application at hand. For that reason, and for clarity of exposition, a single application was selected as a common theme throughout the dissertation. Given the distinguished place of gears in the history of mechanical engineering and owing to their continuing importance in present-day machine design, the choice was made for the *dynamic gear contact problem*.

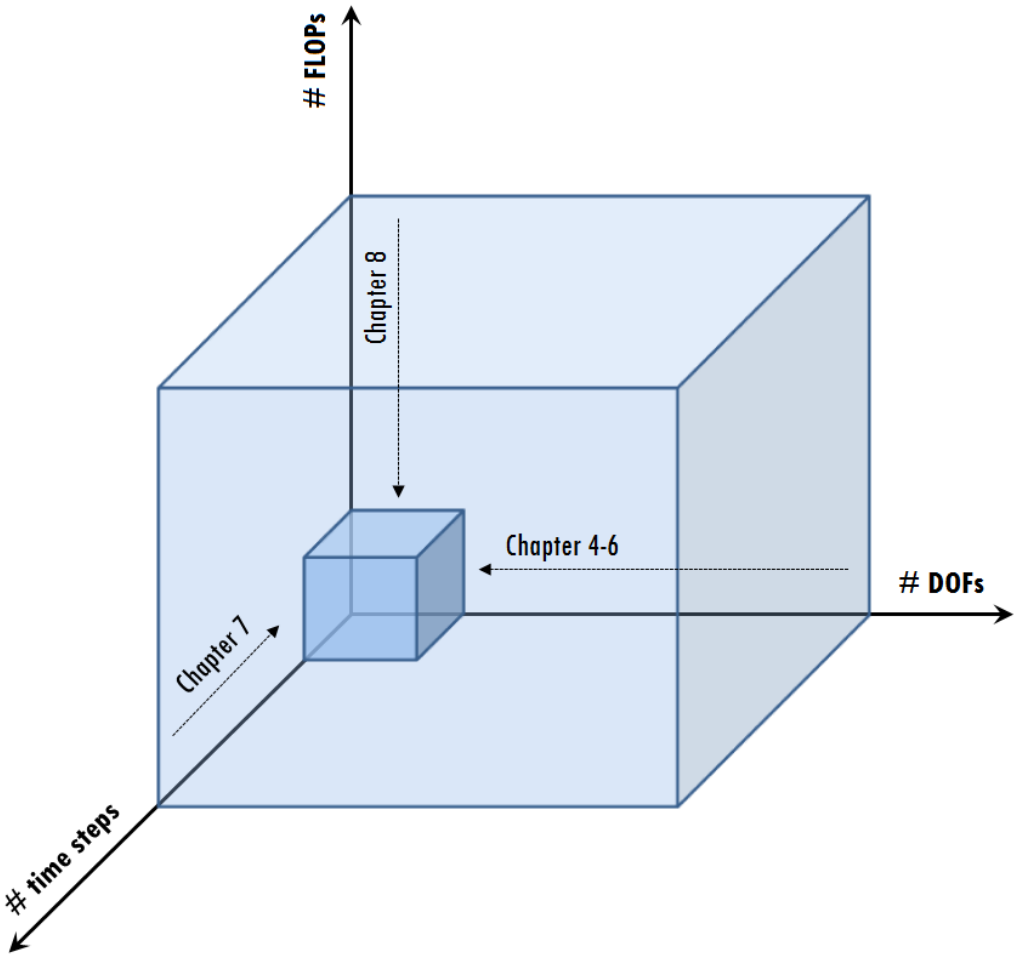


Fig. 1.2 Current computational cost of a FE-based dynamic contact simulation (big cube) and the aims of this dissertation (small cube)

1.2 State-of-the-art in dynamic contact modelling

1.2.1 Lumped parameter modelling

In the previous section it was argued that contact interactions in mechanical systems are accompanied by a series of important dynamic phenomena including abrupt changes in system velocities and vibrational wave propagation. In order to investigate the effect of these phenomena

on the behavior of the system, a dynamic (as opposed to static) contact analysis is required. To this end, a large variety of numerical modelling approaches have emerged in the last decades, differing in accuracy, complexity and efficiency. A primary distinction between the various modelling approaches in computational contact mechanics (and in numerical modelling in general) is that between *lumped* versus *distributed* parameter modelling. Given the importance of *geometry* in the design of mechanical components, distributed parameter approaches will be the main focus of this dissertation. Nevertheless, lumped parameter approaches continue to play an important role in the early design stage of mechanical systems, as well as in the system-level (as opposed to component-level) analysis of large multibody systems.

In the lumped parameter approach, the geometry of spatially distributed systems is simplified into a topology of discrete entities that approximate the behaviour of the original system under certain assumptions. Given a pair of engaging gears, the *inertias* of the respective gears depend on their material distribution whereas the *stiffness* of the gear mesh depends on such factors as the number of teeth in contact, the shape of the contact zones and the geometry of the gear blanks (see Fig. 1.3a). In the lumped parameter approach, a gear pair is typically simplified into a pair of rotating cylinders with constant inertias, while the stiffness of the gear mesh is *lumped* into a spring (with either constant or variable spring stiffness) that connects the two cylinders along the line of action (see Fig. 1.3b). This approach of lumping the inertia and stiffness properties of a contact problem has been successfully applied in such fields as gear, bearing and wheel-railway contact analysis.

In gear contact analysis, the various lumped parameter approaches differ mainly in the way the lumped gear mesh stiffness is defined. In [13], this stiffness is obtained by summing the local tooth contact stiffnesses (determined using the FEM prior to the actual simulation) at the various points of contact, whereas in [14] the average mesh stiffness of a gear pair given by the ISO standard [15] is scaled using an analytical formula that depends on the angular configuration and the contact ratio of the gear pair. Other approaches rely on classical contact theory (most notably the works of Hertz [16], Boussinesq [17] and Weber-Banashek [18]) for computing the local tooth contact stiffnesses [19], [20]. While these approaches are primarily designed for the *torsional* analysis of geared systems, the lumped parameter approach has also been successfully applied in three-dimensional gear contact modelling, see e.g. [21]. A noteworthy and particularly successful application of the lumped parameter approach in three-dimensional bearing modelling can be found in [22], where the contact interactions between the rolling elements and raceways are modelled using the Hertzian solution for sphere-to-sphere contact.

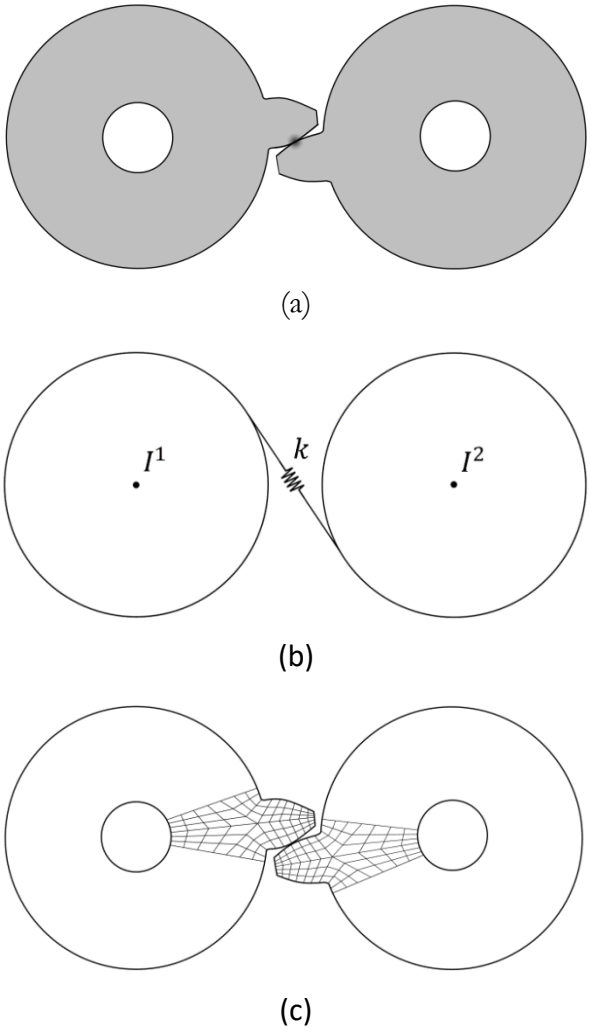


Fig. 1.3 Modelling of a pair of gears, with (a) the physical gear pair, (b) the lumped parameter model, and (c) the distributed parameter model

1.2.2 The finite element method and the need for model reduction

While lumped parameter methods enable the rapid construction and efficient evaluation of numerical models for a broad class of engineering problems, they lack the amount of modelling complexity that is required to evaluate a component's dynamic behavior with sufficient (geometrical) detail. Moreover, these methods are generally based on a large number of assumptions that significantly limits their range of applicability. Therefore, when a CAA-engineer wants to investigate the influence of certain design modifications on the stress fields in a mechanical component during a series of dynamic events, he or she will have to resort to a distributed parameter approach. In such an approach, the geometrical extent of the mechanical system under consideration is taken into account while a considerable part of the lumped parameter assumptions is replaced by first principles. Contrary to their lumped parameter counterparts, distributed parameter models are infinite-dimensional space-time models that are described by time-dependent Partial Differential Equations (PDEs). Their numerical solution

involves the discretization of spatial derivatives, yielding finite-dimensional models that are described by large systems of semi-discrete (discrete in space, but continuous in time) Ordinary Differential Equations (ODEs). A variety of numerical discretization methods is currently available to the CAA-engineer, including finite difference and finite volume methods [23], spectral methods [24], the method of lines [25] and the FEM [11]. In continuum mechanics, and in computational contact mechanics in particular, the FEM has proven to outperform the others in terms of versatility and ease-of-use [10], as is evidenced by its wide-spread use in commercial CAE-software.

The most general application of the FEM to the analysis of flexible multibody systems is offered by the Nonlinear FEM (NLFEM). The degrees of freedom of the NLFEM are the absolute nodal translations and rotations with respect to a unique inertial reference frame. As no distinction is made between a component's rigid body motion and elastic deformation, a natural representation of many nonlinear effects including large displacements and deformations is obtained [26]. However, the consistent description of large rotations requires a nonlinear geometric formulation of relatively high computational complexity, see e.g. the work of Géradin and Cardona [27], where a systematic treatment of flexible multibody systems using the NLFEM is described. In order to circumvent the mathematically involved description of large rotations, Shabana [28] introduced the Absolute Nodal Coordinate Formulation (ANCF), where nodal translations and slopes instead of rotations are used as DOFs. While this approach has led to the development of a considerable library of (most notably beam and shell) ANCF finite elements [29], no consistent procedure for the general treatment of solid flexible multibody systems is currently available.

Under the assumption of small strains and displacements, the linear FEM offers a mathematically tractable and computationally efficient procedure for simulating the dynamic behavior of flexible bodies. Despite these stringent assumptions, the method is often used in the analysis of flexible multibody systems by combination with the Floating Frame of Reference (FFR) formulation [6]. This formulation separates the general motion of a flexible body into the gross nonlinear motion of the body's reference frame and the linear-elastic deformation of the body with respect to this frame, describing the latter using linear finite elements. As in most mechanical systems involving contact interactions the proper functioning of the system relies on the elastic deformations being small, the FFR formulation yields a valid approach for their numerical analysis.

Having eliminated the mathematical complexity from the continuum description of flexible bodies, the CAA-engineer is left to cope with the high dimensionality of FE discretizations in computational contact mechanics. The accurate computation of contact forces in FE analysis relies heavily on the FE model's geometrical resolution, and hence on the number of nodal DOFs. While the large numbers of DOFs typical of FE-based contact analyses may be computationally feasible in *static* contact analyses, they cease to be so under *dynamic* conditions, where large systems of dynamic equilibrium equations need to be solved a huge number of times.

Some authors have proposed to combine the FEM with semi-analytical solutions for the displacement fields in the zone of contact, thereby eliminating the need for highly refined FE meshes. These approaches separate the bulk deformation of the interacting components from the nonlinear local displacement field at the contact surfaces, computing the former using linear FE on a coarse mesh and obtaining the latter from classical contact theory. Most notably, Vijayakar [30] combines the FEM with the Boussinesq solution for a point load acting on an infinite half-space, while Andersson and Vedmar [19] compute the local deformation due to Hertzian contact pressures using a formula derived by Weber and Banaschek [18]. Although these approaches require significantly less DOFs as compared to traditional FE-based contact simulations, the procedure of matching the analytical and FE solutions at a certain depth below the contact surface is computationally involved, resulting in relatively high costs per time increment [31]. Moreover, an energy-consistent adaptation of these approaches to the formalism of flexible MBSD has yet to be pursued. Finally, the question as to whether an infinite half-space serves as a good approximation of the actual contact zone is highly case-specific.

The use of conventional finite elements to represent flexible bodies in multibody systems can be rendered practical with the aid of Model Order Reduction (MOR) techniques. These techniques originate from the fields of control theory [32] and structural dynamics [33] and were primarily designed for reducing the sizes of, respectively, first and second-order *linear* systems. While second-order MOR techniques have been the methods of choice in both academic research and commercial software since the advent of flexible MBSD, first-order methods are becoming increasingly popular in recent years [34, 35]. However, neither of these methods can readily be applied to general contact problems to yield computationally efficient Reduced-Order Models (ROMs). In the next three sections, the main challenges in applying MOR techniques to FE-based contact analyses are discussed and an overview is given of some of the attempts undertaken to deal with these challenges.

1.2.3 Model reduction in dynamic contact analysis

Contact problems in multibody dynamics are characterized by significant variations in the location and size of the contact area. For a flexible body discretized using the FEM, this implies a multitude of possibly - although not necessarily simultaneously - loaded DOFs. Furthermore, as contact interactions generally involve steep stress gradients and relatively small volumes of stressed material, highly refined meshes are mandatory to correctly capture these stress fields. This Multiple Input – Multiple Output (MIMO) behavior of the multibody contact problem poses considerable difficulties to classical model reduction techniques.

The MOR techniques considered in this dissertation belong to the class of projection-based methods [32]. These methods approximate the unknown state vector in a basis of reduced dimension and project the governing equations onto a suitably defined low-dimensional subspace. In the modal transformation method, this subspace is spanned by the dominant eigenvectors of the flexible body and the ROM is obtained from a symmetric Galerkin projection. In [36], this

method was used to investigate the dynamic behavior of a pair of impacting gears. Notwithstanding the computationally favorable properties of the reduced-order matrices, the high number of eigenvectors required to resolve local deformations and stresses considerably restricts the capabilities of the method in contact simulations. The Component Mode Synthesis (CMS) approach [33] retains only the dynamically reacting eigenvectors and augments these with static shape vectors such as constraint or attachment modes to compensate for the omitted high-frequency eigenvectors. The guaranteed static completeness of CMS has brought numerous researchers to use the method for solving dynamic contact problems [37], including those involving gear contact [38]. However, the addition of a static shape vector for each boundary degree of freedom that can possibly be loaded during the simulation results in ROMs having several hundreds or even thousands of DOFs, particularly in the case of highly refined FE meshes. Some authors have proposed modified versions of the traditional CMS method to overcome this curse of dimensionality and render the method computationally feasible for multibody contact problems. In the dynamic bearing model of [39, 40], the multitude of static shape vectors in the component mode set of the housing is replaced by a comparatively small set of global constraint modes. These semi-analytically defined modes account for the static deformation of the outer raceway and housing caused by the contact forces in the bearing, with only minor losses in static completeness. However, the method is specifically designed for the simulation of rolling element bearings, whose mathematically simple geometries naturally allow for such semi-analytically defined shape vectors, and cannot be readily applied to more general contact problems.

A statically complete and computationally tractable procedure for the analysis of dynamic contact problems is provided by the Static Modes Switching (SMS) method [41], which is applied to the gear contact problem in [42]. Taking advantage of the observation that at each time step of the simulation only few out of many interface DOFs will be loaded simultaneously, SMS makes use of a discontinuously changing basis matrix of minimal dimensions, containing only those static shape vectors that actually contribute to the solution. For this approach to yield valuable results, special care has to be taken with regard to the construction of the basis matrix [43]. Furthermore, the approach should be combined with time integration schemes exhibiting high-frequency numerical damping in order to alleviate the influence of artificially introduced discontinuities that result from removing static shape vectors from the basis matrix. While the SMS approach yields reduced-order models having significantly less DOFs as compared to CMS-based ROMs, the efficiency of these ROMs remains sensitive to the resolution of the FE mesh, and decreases as the number of simultaneous contact interactions increases.

The application of first-order model reduction techniques to contact problems in multibody dynamics is considerably less widespread then the use of second-order methods. This is partly due to the lack of direct and clear physical interpretability of the basis vectors obtained using MOR techniques based on e.g. Gramian matrices [44] or Krylov-subspaces [45]. Moreover, as these techniques primarily seek to approximate the input-output behavior of the full-order system, a good approximation of field variables such as stresses and strains is not guaranteed unless a

prohibitively large number of input-output relations is defined. An interesting contribution in this regard is due to [46], who proposes a two-step approach for the reduction of systems having a large number of terminals. Despite these and similar efforts to efficiently reduce MIMO systems, the suitability and merits of first-order MOR techniques to solve industrially relevant multibody contact problems yet need to be demonstrated.

The MOR techniques discussed thus far are based on system-theoretic considerations of the underlying equations of motion. Another important class of MOR methods are the so-called snapshot based methods, including the Reduced Basis Method (RBM) [47, 48] and the Proper Orthogonal Decomposition (POD) [49]. These methods are characterized by an (often expensive) offline training phase during which so-called *snapshots* (or solution vectors) of the field variables are gathered by numerically querying the full-order model. The goal is then to construct an optimal approximation space for these snapshots by identifying their dominant directions, typically using a matrix factorization technique such as the Singular Value Decomposition (SVD) [50]. Finally, the ROM is obtained by a Galerkin projection of the Equations Of Motion (EOMs) onto the space spanned by these dominant directions. In [51], this approach is successfully applied to a number of academic examples involving static and dynamic contact interactions. As contact problems become more complex, however, both the design as the actual execution of the offline training phase become prohibitively elaborate. Nevertheless, the main idea behind snapshot based MOR methods has proven to be valuable in the fields of both parametric/nonlinear MOR as well as hyperreduction, as will be discussed in the next two sections.

1.2.4 Parametric and nonlinear model order reduction

The dynamic simulation of gears and bearings is characterized by contact loads moving across the boundaries of the interacting components. When modeling these systems, it thus seems natural to parameterize the location of the contact forces in the Full-Order Model (FOM). For the corresponding ROM to be computationally tractable, this parameter dependency should be preserved throughout the reduction process. The latter is the goal of Parametric Model Order Reduction (PMOR) [52].

Parametric model reduction deals with the generation of ROMs that approximate the original FOM over a range of parameters that may have entered the model in several ways, representing, for example, material parameters, component geometry, system configuration, or boundary conditions - the latter being of particular importance to contact problems. In constructing the parametric ROM, a distinction is made between the use of global and local basis matrices. Global basis matrices capture the parametric dependence by embedding information about the entire parameter space (i.e. by sampling many parameter values; this is the approach typically adopted in snapshot based PMOR) [53], whereas local basis matrices are computed for a specific parameter realization. Using the latter approach, the parametric ROM can be constructed either by interpolating the local basis matrices over the parameter space [54] or by first constructing

locally reduced models and then interpolating the reduced models themselves, leading to real-time capabilities [55]. The question of how to correctly interpolate local basis matrices or locally reduced models is an active area of research. In [56], local basis matrices are interpolated on a tangent space to the Stiefel manifold so as to preserve orthonormality of the basis vectors, while in [57] the polynomial interpolation of locally reduced models is preceded by a congruence transformation of the basis matrices in order to express the reduced system matrices in a common generalized coordinate system.

Systems with parametric boundary conditions represent a special case of parametric dependence, in the sense that the system matrices of the FOMs (i.e. the mass, damping and stiffness matrices in structural dynamics, or the state matrix in control and systems theory) are not parameter-dependent themselves, yet the reduced-order matrices become parameter-dependent through the process of model reduction. PMOR as such has been applied to the moving loads problem in structural dynamics in [58], where it was used to capture the dynamic stress fields resulting from point loads travelling across the flanks of a clamped elastic gear. For each moving load, the proposed reduction scheme interpolates between a set of local basis vectors according to the parameterized location of the load. Taking the implicit time-dependence of the basis matrix into account in the Galerkin projection of the equations of motion, a parametric ROM is obtained of which the system matrices not only depend on the parameters themselves, but also on their first and second derivatives. In [59, 60] this approach is extended to the spatial analysis of ball and roller bearings under dynamic operating conditions. These studies show that the mere interpolation of local basis matrices or locally reduced models might lead to energy-inconsistent ROMs when dynamic parameter variations are at play.

The interpolation of locally reduced models is also frequently encountered in the field of Nonlinear Model Order Reduction (NLMOR). Here, the local ROMs are typically obtained through linearization of the FOM at discrete points along a reference trajectory [61]. Of particular importance to the present work is the Global Modal Parameterization (GMP) method, which is conceived as a system-level NLMOR technique for multibody systems [62, 63, 64]. In this approach, the flexible deformation of a mechanism is considered as a small deviation with respect to the undeformed rigid-body configuration, which is parameterized by a minimal set of rigid DOFs. The elastic displacements are described by a superposition of interpolated system-level shape vectors that are derived from the linearized mechanism at a number of reference configurations.

Despite the challenges in correctly interpolating between locally reduced models and in dealing with dynamic parameter variability, parametric and nonlinear model reduction techniques have the potential of reducing the dimensions of large-scale dynamic contact problems considerably further than classical MOR techniques. Taking advantage of this potential in the efficient solution of contact problems in flexible MBSD is one of the main objectives of this dissertation.

1.2.5 Hyperreduction of nonlinear reduced models

In the previous two paragraphs it was shown how projection-based MOR restricts the solution space of a FE model to a smaller subspace, thereby reducing the number of DOFs. This approach yields computationally efficient ROMs if the equations of motion of the underlying FOM are linear. If instead the underlying equations of motion contain a nonlinear term, such as configuration-dependent elastic or contact forces, computing the nonlinear term in the ROM requires evaluating the high-dimensional nonlinear term in the full-order solution space prior to its reduction. A second stage of model reduction is thus required to reduce the complexity of the nonlinear ROM and to obtain a truly efficient ROM. This is where the field of *hyperreduction* comes into play.

The main idea behind most if not all successful hyperreduction techniques is to approximate a nonlinear vector function through the selective evaluation of a comparatively small subset of its components. In projection-based hyperreduction methods the nonlinear function is projected onto an empirically derived basis of reduced dimensions and the unknown amplitudes of the basis vectors are collocated to the subset of queried function components [65, 66]. Other hyperreduction methods approximate the nonlinear function through direct weighting of the queried function components using a precomputed set of weighting factors [67, 68]. While hyperreduction allows for a considerable reduction of the complexity of nonlinear ROMs, it generally relies on a computationally expensive offline training phase involving the FOM in order to determine an appropriate set of basis vectors or weighting factors.

Popular hyperreduction schemes in the field of finite element analysis include the Discrete Empirical Interpolation Method (DEIM) [66] and the Energy-Conserving Sampling and Weighting (ECSW) method [68]. The DEIM computes the reduced-order nonlinear function in two steps: first the full-order nonlinear function is approximated through cheap evaluation of a small number of function components, followed by the projection of the approximated function onto the subspace in which the ROM resides. The ECSW method instead directly approximates the reduced-order nonlinear function by means of an energy-based weighted summation of reduced-order function components. Contrary to the DEIM, the ECSW method is proven to preserve the Lagrangian structure of the underlying equations of motion when generating the hyperreduced-order model, yielding superior stability properties as compared to the potentially unstable DEIM [68].

Applications of hyperreduction in literature are mainly concerned with nonlinearities arising from either nonlinear material laws (material nonlinearities) [68, 69, 70], large motions and/or deformations (geometrical nonlinearities) [71] or both [68]. In these applications, it is common for the nonlinear function to be composed of elemental or nodal contributions involving *all* the elements or nodes in the full-order model. The high numerical complexity of the nonlinear function in these applications stems primarily from the multitude of elements or nodes that need to be queried, rather than from the cost of evaluating the nonlinear function at the elemental or

nodal level. Therefore, considerable computational savings can be achieved by expressing the nonlinear function in terms of a comparatively small number of elemental or nodal contributions. The same is not true for multibody contact problems. In these problems, the nonlinear contact forces are composed of expensive elemental and/or nodal contributions involving only a *small subset* of elements or nodes in the full-order model. Furthermore, due to the lack of predefined one-to-one relationships between function components and the DOFs that need to be queried, the computational savings of standard hyperreduction schemes are limited considerably. While the majority of CPU time in a typical dynamic contact analysis is spent on evaluating the nonlinear contact forces, the potential of hyperreduction in the field of computational contact mechanics is yet to be demonstrated.

1.3 Objectives, contributions and thesis outline

1.3.1 Objectives

The main objective of this dissertation is to improve the efficiency of FE-based contact simulations in flexible multibody dynamics. Three main reasons for the high computational cost of such simulations have been identified above, namely the high number of DOFs in the FE-representation of the contacting bodies, the high numerical complexity of the nonlinear contact forces, and the small characteristic time length of dynamic contact phenomena. This dissertation aims at alleviating the computational burden associated with each one of these aspects, respectively by:

- Developing a **novel MOR technique** that is tailored to FE-based contact simulations in flexible MBSD and that allows for a further reduction of the DOFs as compared to conventional MOR schemes. Ideally, the newly developed technique should be relatively simple from the algorithmic and bookkeeping point of view and should impose no additional constraints on the numerical integrator, thereby promoting the broader use of the technique, e.g. in commercial flexible MBSD software.

- Developing a **novel hyperreduction scheme** for the efficient computation of nonlinear contact forces in flexible multibody systems. The contact interactions in such systems are generally characterized by large variations in the size and location of the contact zones, which poses significant challenges to standard hyperreduction techniques. Again, algorithmic simplicity is pursued in all phases of the development.

- Developing an **efficient numerical integration scheme** that effectively deals with the small characteristics time lengths of dynamic contact problems while meeting the accuracy and stability requirements of the analysis. This objective includes a numerical investigation

into the performance of well-established numerical integrators in the solution of multibody contact problems and into the precise mechanisms that drive these integrators to use prohibitively small time increments.

The primary objective of this dissertation is concerned with reducing the cost of the *online* phase of the simulation (i.e. the actual integration of the EOMs, starting after completion of all pre-processing duties including the assembly of system matrices and the computation of reduced basis vectors). In addition to this objective, a secondary goal is to minimize the cost (and the required user-input) of the *offline* simulation phase. The latter has received little attention in the literature on MOR and hyperreduction, yet constitutes a considerable part of the overall simulation time and effort.

Finally, the techniques developed in this dissertation are meant to aid in the efficient analysis of industrial-sized multibody contact problems. Notwithstanding the merits of so-called 'academic examples', the performances of the novel techniques are ultimately to be assessed based on applications that have similar size and complexity as those encountered in industry.

1.3.2 Contributions

The main contributions of this dissertation can be summarized as follows:

- Development and validation of a **novel MOR technique** for the efficient solution of dynamic contact problems. The novel MOR technique belongs to the class of parametric MOR techniques and relies on the concept of online interpolation of Reduced-Order Bases (ROBs) based on the actual configuration of the multibody system. The ROBs consist of the dominant eigenvectors of the interacting flexible bodies augmented with a small but continuously variable set of *contact shape vectors*. As such, the method can be considered as an extension of the method presented in [58] to the analysis of flexible multibody systems with contact, while it shares with GMP [64] the configuration-dependent interpolation of basis vectors. The proposed method is shown to yield ROMs with considerably less DOFs than attainable by traditional MOR techniques, while preserving accuracy on displacement and stress level. Moreover, given the specific choice of basis vectors, the method is insensitive to the number of nodal DOFs in the full-order FE model. The method is applied to the dynamic analysis of perfectly aligned as well as misaligned gear pairs and gear trains, including planetary gear sets.

- Development and validation of **two novel hyperreduction schemes** for contact problems in flexible MBSD. The novel schemes are based on an adaptation of the DEIM [66] and ECSW method [68], respectively, and are formulated in terms of configuration-dependent basis vectors and weighting factors. The latter are recycled from the

computation of the above-mentioned contact shape vectors, thereby avoiding expensive training simulations during the construction of the reduced-order model. Two essential aspect of the proposed schemes are the conservation of Newton's third law of motion (action equals reaction) and the proper selection and cost-efficient evaluation of a representative subset of nonlinear function components. The novel hyperreduction schemes are shown to significantly reduce the cost of computing the nonlinear contact forces while being relatively insensitive to the mesh density of the full-order FE model.

- Development and validation of **two efficient numerical integration schemes** tailored to FE-based contact simulations, preceded by an empirical assessment of the performance of a number of well-established time integrators. The developed integration schemes are based on the explicit Central Difference (CD) method [72] and the family of Newmark integrators [73], respectively, and are equipped with error estimators and adaptive time-stepping. The main features of these schemes are the numerical elimination of high-frequency solution components, the availability of an analytical expression for the nonlinear contact force Jacobian, and the efficient inversion of the configuration-dependent mass matrix.

With regard to the secondary goal mentioned above – that of reducing the cost and required user input of the *offline* simulation phase – a number of smaller contributions have been made, including the use of an Intermediate ROM (iROM) and the development of a procedure to obtain analytical contact Jacobians, a procedure to exploit symmetry in constructing the reduced-order basis matrices, and a mixed-mesh formulation of the contact solution that allows to reduce not only CPU time but also memory usage.

In search for an illustrative application of industry size and complexity, the dynamic gear contact problem was selected as a common theme throughout this dissertation. The tools and techniques developed in this text have been implemented by the author in a MATLAB-based toolbox called MUTANT, which is an acronym for Multibody Transient Analysis of Transmissions. The toolbox allows for the highly-automated analysis of both spur and helical, internal and external gears, and has been successfully applied in a number of industrial case studies where the use of commercial software with similar accuracy targets would not have been computationally feasible.

1.3.3 Thesis outline

The presented research objectives are addressed in eight chapters according to the structure shown in Fig. 1.4. Chapter 4 presents the novel MOR technique and can be considered as the central chapter in this dissertation. Chapters 5 and 6 can be read independently starting from Chapter 4 and discuss improvements and extensions of the novel MOR method. Chapter 7 deals with the efficient solution of the reduced-order equations of motion, while in Chapter 8 the efficient computation of the nonlinear contact forces is investigated. Considerable attention is paid to

presenting the theoretical background (Chapter 2) and the selected case study (Chapter 3), however these chapters can be read selectively, based on the reader's experience and interests.

The content of the upcoming seven chapters can be summarized as follows:

- In **Chapter 2** the theoretical background of the methods developed in this dissertation is discussed. The chapter introduces the most important concepts of finite element and flexible multibody modelling, model reduction, the treatment of contact interactions, and the numerical integration of the equations of motion. The theoretical background is included for two reasons: first, to clearly separate the newly developed techniques from the state-of-the-art techniques they are built on, and secondly, to make the thesis self-contained and accessible to the less experienced CAA-engineer.

- In **Chapter 3** the full-order dynamic gear contact problem is presented. Starting from the analytical description of involute gear geometry, the procedure to discretize the flexible gears and formulate their equations of motion in the floating frame of reference framework is described. Furthermore, an overview of the study cases used throughout this dissertation is given, as well as an assessment of the accuracy of the full-order model based by comparison with commercial NLFE software and experimental measurements.

- In **Chapter 4** a novel MOR technique for dynamic contact problems is introduced and applied to the particular case of a perfectly aligned pair of parallel-axis gears. Due to the perfect alignment of the gears, out-of-plane motion of the gear pair can be neglected which considerably simplifies the main features of the proposed technique (construction of the basis matrix, interpolation of the basis vectors, derivation of the reduced equations of motion). Additional topics that are dealt with in this chapter are the mixed-mesh formulation of the ROM, the analytical derivation of the Jacobian matrix of the contact forces, and an assessment into the relative importance of inertial terms in the EOMs of flexible multibody systems.

- In **Chapter 5** two possible shortcomings of the novel MOR technique are identified and resolved by introducing two modifications to the original scheme. The improved model accuracy enabled by these modifications comes at the price of a small increase in the number of DOFs and computational complexity, yet the computational savings of the improved method remain of the same order of magnitude as the original MOR scheme.

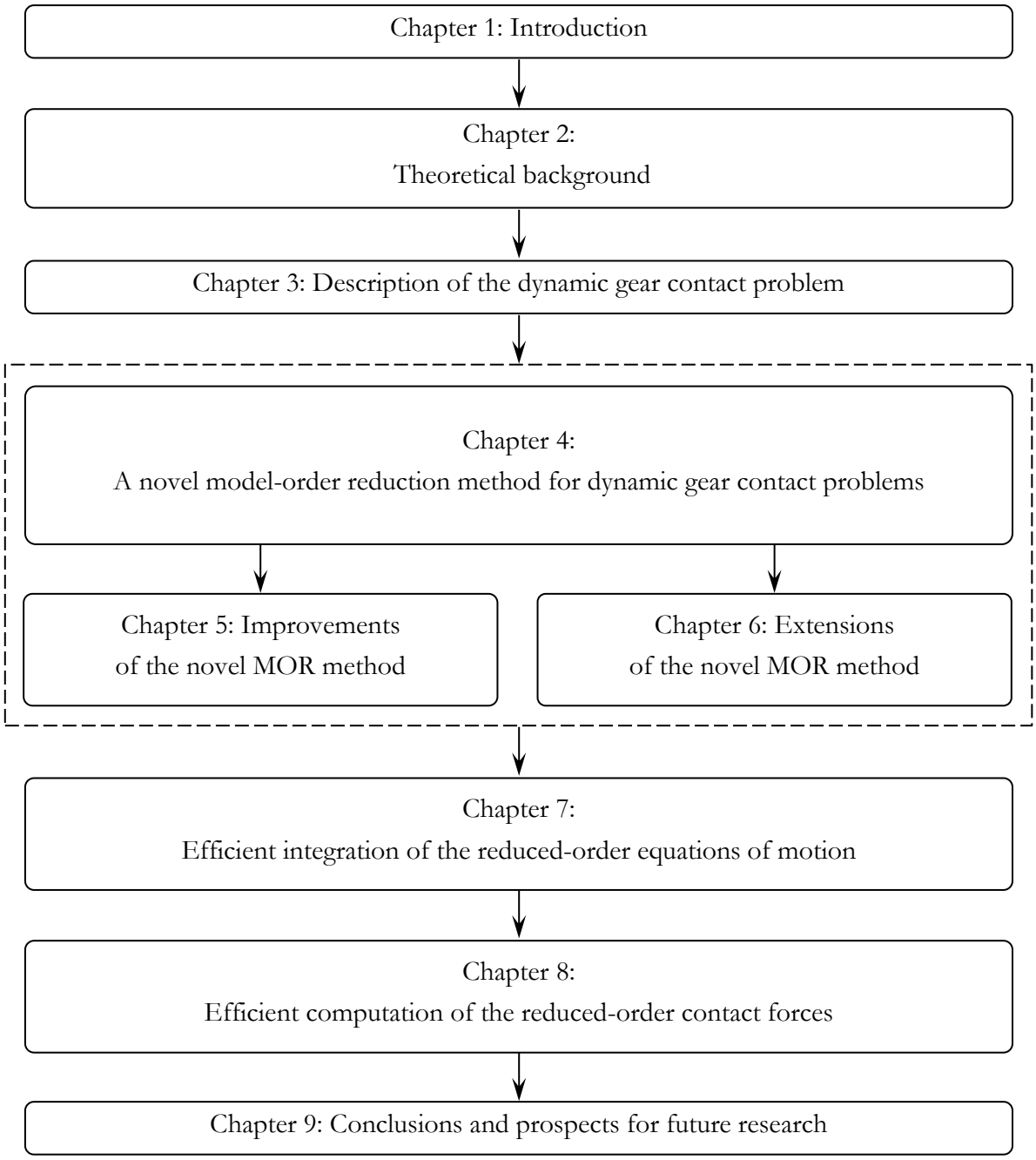


Fig. 1.4 Structure of this dissertation

- In **Chapter 6** the proposed MOR technique is extended to general multibody contact problems (i.e. including three-dimensional nonlinear motion) and applied to the analysis of a pair of misaligned gears. Particular attention is paid to the efficient and automized construction of the reduced-order basis matrix using a greedy sampling method. In addition, the proposed MOR scheme is applied to the analysis of planetary gear sets involving multiple simultaneous contacts and several flexible bodies undergoing large rotations and translations. Finally, the method is used to construct an efficient three-dimensional force element that can be integrated in commercial flexible multibody software.

- In **Chapter 7** an optimal explicit and implicit integration scheme for solving the reduced-order ODEs and Differential-Algebraic Equations (DAEs) is developed based on a comparative analysis of several well-established integrators. Special attention is paid to the elimination of high-frequency solution components in explicit schemes and efficient updates of the system's Jacobian in implicit schemes.

- In **Chapter 8** two novel hyperreduction schemes for the efficient computation of the reduced-order contact forces is described, with particular focus on the computation and selection of the basis vectors and weighting factors, and their parameterization with respect to the configuration of the multibody system. The efficiency of the hyperreduction schemes is demonstrated by application to the dynamic gear contact problem.

- In **Chapter 9** the main conclusions drawn during this investigation are summarized and promising directions for future research are identified and discussed.

Chapter 2

Theoretical background

In this chapter the theoretical background of the methods presented in subsequent chapters is described. The governing equations of flexible multibody dynamics will be introduced starting from first principles, with special focus on the theory of model reduction, the mathematical treatment of contact interactions, and the numerical integration of the equations of motion. The experienced reader may choose to skip this chapter on first reading, referring back to it when needed in later chapters.

2.1 Fundamental equations of continuum mechanics

2.1.1 Hamilton's principle and Lagrange's equations of motion

The theory of continuum mechanics offers the most general description of the motion and elastic behavior of material objects on the macroscopic level. It represents a vast and complicated subject on which there is enormous literature [5, 74]. The fundamental principle of continuum mechanics is Hamilton's principle, which states that the true evolution of a system described by N generalized coordinates $\boldsymbol{q} = (q_1, q_2, \cdots, q_N)$ between two states $\boldsymbol{q}_0 = \boldsymbol{q}(t_0)$ and $\boldsymbol{q}_1 = \boldsymbol{q}(t_1)$ at two specified times t_0 and t_1 is a stationary point (i.e. a point where the variation is zero) of the action functional A. The latter is defined as the integral of the system's Lagrangian over time, taken along the trajectory of the system between the initial and final states, i.e.

$$A \coloneqq \int_{t_0}^{t_1} L(\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), t)dt \tag{2.1.1}$$

where the Lagrangian of a non-relativistic mechanical system is defined as the quantity $L = T - U$, where T and U represent, respectively, the kinetic and potential energies of the system. Mathematically, Hamilton's principle can thus be expressed as $\delta A = 0$, or more elaborate:

$$\delta A = \sum_{i=1}^{N} \left[\frac{\partial A}{\partial q_i} \delta q_i + \frac{\partial A}{\partial \dot{q}_i} \delta \dot{q}_i \right] = 0 \tag{2.1.2}$$

When the system is subjected to non-potential external forces $\boldsymbol{f}_{ext}$, Hamilton's principle takes the more general form (also referred to as the principle of Lagrange-d'Alembert):

$$\delta A - \int_{t_0}^{t_1} \delta W \, dt = \int_{t_0}^{t_1} [\delta(T - U) - \delta W] \, dt = 0 \tag{2.1.3}$$

where δW represents the virtual work done by the external forces, i.e. $\delta W = \boldsymbol{f}_{ext} \delta \boldsymbol{q}$. Substitution of Eq. 2.1.1 and Eq. 2.1.2 in the above equation and applying integration-by-parts yields Lagrange's equations of motion:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{\boldsymbol{q}}} - \frac{\partial L}{\partial \boldsymbol{q}} = \frac{d}{dt}\frac{\partial T}{\partial \dot{\boldsymbol{q}}} - \frac{\partial T}{\partial \boldsymbol{q}} + \frac{\partial U}{\partial \boldsymbol{q}} = \boldsymbol{f}_{ext} \tag{2.1.4}$$

These equations describe the dynamic behavior of all non-relativistic mechanical systems, including flexible multibody systems.

2.1.2 Equations of motion of an elastic body

The Equations Of Motion (EOMs) of an elastic continuum are obtained from Hamilton's general principle (Eq. 2.1.3) by considering a continuous body with density ρ, volume Ω and boundary Γ, see Fig. 2.1. Let $\boldsymbol{u} = (u_1, u_2, u_3)^T$ denote the displacement vector of a material particle in position $\boldsymbol{x} = (x_1, x_2, x_3)^T$ in the body. The virtual work done on the body at time t by the body force $\rho \boldsymbol{b}$ and the specified surface traction $\hat{\boldsymbol{t}}$ in moving through a virtual displacement $\delta \boldsymbol{u}$ is given by:

$$\delta W = \int_{\Omega} \rho \boldsymbol{b} \cdot \delta \boldsymbol{u} \, dV + \int_{\Gamma_\sigma} \hat{\boldsymbol{t}} \cdot \delta \boldsymbol{u} \, dA \tag{2.1.5}$$

where Γ_σ denotes the portion of the boundary Γ on which surface tractions $\hat{\boldsymbol{t}}$ are specified. The kinetic energy of the body and the potential internal energy stored in the body due to elastic deformation are defined as

$$T = \frac{1}{2}\int_{\Omega} \rho \dot{\boldsymbol{u}} \cdot \dot{\boldsymbol{u}} \, dV \quad , \quad U = \frac{1}{2}\int_{\Omega} \boldsymbol{\sigma} : \boldsymbol{\varepsilon} \, dV \tag{2.1.6}$$

where $\boldsymbol{\sigma}$ and $\boldsymbol{\varepsilon}$ are the Cauchy stress and strain tensors of the body.

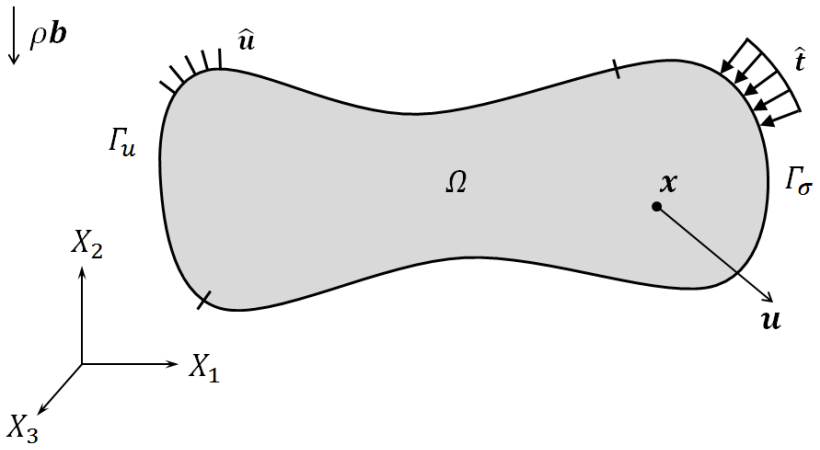


Fig. 2.1 Continuum description of an elastic body

In its most general form, the kinematic relation between strains and displacements in an elastic body is given by the nonlinear Green strain tensor [74]. However, when all displacement gradients are small (i.e. $|\boldsymbol{\nabla u}| \ll 1$), the nonlinear terms in this tensor can be neglected and the strain-displacement relation is given by the infinitesimal strain tensor:

$$\boldsymbol{\varepsilon} = \frac{1}{2}[\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T] \tag{2.1.7}$$

where the gradient $\nabla(\cdot)$ represents the vector of partial derivatives with respect to the various components of $(\cdot)$. The rectangular Cartesian components of the infinitesimal strain tensor are given by

$$\begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{12} \\ \varepsilon_{13} \\ \varepsilon_{23} \end{Bmatrix} = \begin{Bmatrix} \partial u_1/\partial x_1 \\ \partial u_2/\partial x_2 \\ \partial u_3/\partial x_3 \\ \frac{1}{2}(\partial u_1/\partial x_2 + \partial u_2/\partial x_1) \\ \frac{1}{2}(\partial u_1/\partial x_3 + \partial u_3/\partial x_1) \\ \frac{1}{2}(\partial u_2/\partial x_3 + \partial u_3/\partial x_2) \end{Bmatrix} = \begin{bmatrix} \partial/\partial x_1 & 0 & 0 \\ 0 & \partial/\partial x_2 & 0 \\ 0 & 0 & \partial/\partial x_3 \\ \frac{1}{2}\partial/\partial x_2 & \frac{1}{2}\partial/\partial x_1 & 0 \\ 0 & \frac{1}{2}\partial/\partial x_3 & \frac{1}{2}\partial/\partial x_1 \\ \frac{1}{2}\partial/\partial x_3 & 0 & \frac{1}{2}\partial/\partial x_2 \end{bmatrix} \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} = [\partial] \begin{Bmatrix} u_1 \\ u_2 \\ u_3 \end{Bmatrix} \tag{2.1.8}$$

where ε_{11}, ε_{22}, and ε_{33} are the infinitesimal normal strains and ε_{12}, ε_{13}, and ε_{23} are the infinitesimal shear strains. Substitution of Eq. 2.1.5-2.1.7 in Hamilton's general principle (Eq. 2.1.3) yields

$$0 = \int_{t_0}^{t_1} \left[\int_\Omega (\rho \ddot{\boldsymbol{u}} - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} - \rho \boldsymbol{b}) \cdot \delta \boldsymbol{u} \, dV + \int_{\Gamma_\sigma} (\boldsymbol{t} - \hat{\boldsymbol{t}}) \cdot \delta \boldsymbol{u} \, dA \right] dt \tag{2.1.9}$$

where integration-by-parts and gradient theorems were used to arrive at the above equation. Because the virtual displacements $\delta \boldsymbol{u}$ are arbitrary for $\boldsymbol{x}$ in Ω and on Γ_σ, the following equations hold:

$$\rho\ddot{\boldsymbol{u}} - \nabla \cdot \boldsymbol{\sigma} - \rho\boldsymbol{b} = \mathbf{0} \text{ in } \Omega \tag{2.1.10a}$$

$$\boldsymbol{t} - \hat{\boldsymbol{t}} = \mathbf{0} \quad \text{on } \Gamma_\sigma \tag{2.1.10b}$$

$$\boldsymbol{u} - \hat{\boldsymbol{u}} = \mathbf{0} \quad \text{on } \Gamma_u \tag{2.1.10c}$$

where Γ_u denotes the portion of the boundary Γ on which displacements $\hat{\boldsymbol{u}}$ are prescribed. Eq. 2.1.10a represent the EOMs of an elastic body undergoing small displacements and subjected to the boundary conditions prescribed in Eq. 2.1.10b and 2.1.10c.

2.1.3 Constitutive equations of elastic isotropic materials

In order to solve Eq. 2.1.10, additional information regarding the object's material properties is required. This information is provided by the material's constitutive equations: mathematical models of the behavior of materials that are not derived from any physical principles, yet are validated against experimental results. In this dissertation the body is assumed to be linear-elastic (i.e. it recovers its original form upon removal of the forces causing deformation) with homogeneous and isotropic material properties (i.e. independent of location and direction). For such materials, the stress-strain relations expressed by Hooke's law take the following form:

$$\sigma_{ij} = \frac{E}{1+\nu}\varepsilon_{ij} + \frac{\nu E}{(1+\nu)(1-2\nu)}\varepsilon_{kk}\delta_{ij} \tag{2.1.11}$$

where E is the material's modulus of elasticity and ν is the material's Poisson ratio. Furthermore, summation on repeated indices is implied and $\delta_{ij} = 1$ if $i = j$ and $\delta_{ij} = 0$ otherwise. The above equation can be written in matrix form as follows:

$$\begin{Bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{33} \\ \sigma_{12} \\ \sigma_{13} \\ \sigma_{23} \end{Bmatrix} = E' \begin{bmatrix} 1-\nu & \nu & \nu & 0 & 0 & 0 \\ & 1-\nu & \nu & 0 & 0 & 0 \\ & & 1-\nu & 0 & 0 & 0 \\ & & & 1-2\nu & 0 & 0 \\ & symm. & & & 1-2\nu & 0 \\ & & & & & 1-2\nu \end{bmatrix} \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{12} \\ \varepsilon_{13} \\ \varepsilon_{23} \end{Bmatrix} = [C] \begin{Bmatrix} \varepsilon_{11} \\ \varepsilon_{22} \\ \varepsilon_{33} \\ \varepsilon_{12} \\ \varepsilon_{13} \\ \varepsilon_{23} \end{Bmatrix} \tag{2.1.12}$$

where $E' = E/(1+\nu)(1-2\nu)$, σ_{11}, σ_{22}, and σ_{33} are the normal stresses and σ_{12}, σ_{13}, and σ_{23} are the shear stresses.

2.2 The finite element method in structural dynamics

2.2.1 Weak form of the equations of motion

In the previous section the EOMs (Eq. 2.1.10), the kinematic relations (Eq. 2.1.7) and the constitutive material equations (Eq. 2.1.11) of linear-elastic objects undergoing small displacements are described. The analytical solution of these equations in strong form is generally not possible; therefore numerical methods are used to find approximate solutions. The Finite Element Method (FEM) [11, 75] is one of the most powerful numerical approximation methods in structural dynamics, and definitely the most popular. Two fundamental aspects of the method are at the heart of its great versatility and success: firstly, the partitioning of the problem domain into a number of small, non-overlapping subdomains (called finite elements) over which the solution is approximated by simple local functions; and secondly, the formulation of the underlying equations in *weak* or integral form, thereby enabling the summation of element contributions to integrals over the whole solution domain.

Different approaches exist for transforming the governing equations of continuum mechanics to the weak form, including variational approaches and the weighted residual approach [11]. As the latter offers the most general and direct procedure for deriving the finite element equations, it is briefly described here. The weighted residual approach proceeds in three steps: first an approximate solution $\tilde{\boldsymbol{u}}(\boldsymbol{x}) = \sum_i \boldsymbol{N}^i(\boldsymbol{x})\boldsymbol{c}^i$ to the field variable $\boldsymbol{u}(\boldsymbol{x})$ is assumed in terms of some selected functions $\boldsymbol{N}^i(\boldsymbol{x})$ and unknown coefficients $\boldsymbol{c}^i$. Then an expression for the residual $\boldsymbol{r}$ in terms of these unknown coefficients is computed by inserting $\tilde{\boldsymbol{u}}$ into the governing differential equations. Finally, the unknown coefficients $\boldsymbol{c}^i$ are determined by selecting a weighting function $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{x})$ and enforcing the integral of the weighted residual $\boldsymbol{w} \cdot \boldsymbol{r}$ over the problem domain to be zero. Among the different methods that are available for selecting the weighting function, the Galerkin method is most often used in FE analysis. In this method, the weighting functions are chosen to be the known functions $\boldsymbol{N}^i(\boldsymbol{x})$ of the assumed solution[1].

Following the procedure outlined above, the weak form of the equations of motion of linear-elastic solids (Eq. 2.1.10a) are obtained in the following manner. Insertion of the approximate solution $\tilde{\boldsymbol{u}}$ into Eq. 2.1.10a leads to:

$$\rho\ddot{\tilde{\boldsymbol{u}}} - \boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\tilde{\boldsymbol{u}}) - \rho\boldsymbol{b} = \boldsymbol{r} \tag{2.2.1}$$

where the residual $\boldsymbol{r}$ is a function of both space and time. The residual is now reduced to zero in a weak sense by multiplying $\boldsymbol{r}$ with the selected weighting function $\boldsymbol{w}$ and integrating the residual over the problem domain:

[1] Galerkin's method is equivalent to the principle of virtual work, which states that a deformable body is in dynamic equilibrium when the virtual work due to external forces balances the virtual work due to internal forces when the body undergoes virtual consistent displacements.

$$\int_{\Omega} \boldsymbol{w} \cdot \boldsymbol{r} dV = 0 \quad \Rightarrow \quad \int_{\Omega} \rho \boldsymbol{w} \cdot \left(\ddot{\tilde{\boldsymbol{u}}} - \boldsymbol{b}\right) dV - \int_{\Omega} \boldsymbol{w} \cdot [\boldsymbol{\nabla} \cdot \boldsymbol{\sigma}(\tilde{\boldsymbol{u}})] dV = 0 \tag{2.2.2}$$

By partial integration of the second term in the above equation, application of the divergence theorem and introduction of the traction boundary condition (Eq. 2.1.10b), the weak form of the equations of motion is obtained:

$$\int_{\Omega} \rho \boldsymbol{w} \cdot \left(\ddot{\tilde{\boldsymbol{u}}} - \boldsymbol{b}\right) dV + \int_{\Omega} \boldsymbol{\nabla} \boldsymbol{w} \cdot \boldsymbol{\sigma}(\tilde{\boldsymbol{u}}) dV - \int_{\Gamma_\sigma} \boldsymbol{w} \cdot \hat{\boldsymbol{t}} dA = 0 \tag{2.2.3}$$

Note that Eq. 2.2.3 requires a weaker continuity on the field variables then the corresponding strong form of the equations (once differentiable instead of twice). Moreover, this integral form of the EOMs allows the problem domain to be partitioned into a number of small subdomains whose integral contributions can easily be computed and then summed to determine the integral over the whole domain. This is the essence of the FEM.

2.2.2 The isoparametric eight-noded solid element

In the FEM, the problem domain is discretized into a patchwork of n_e finite-dimensional elements that are interconnected at a number of points (called the *nodes*) along the element boundaries, see Fig. 2.2. The field variable $\boldsymbol{u}(\boldsymbol{x})$ within one finite element is interpolated based on the values of $\boldsymbol{u}(\boldsymbol{x})$ at the nodes of the element, i.e.

$$\boldsymbol{u}(\boldsymbol{x}) \approx \tilde{\boldsymbol{u}}(\boldsymbol{x}) = \sum_{i=1}^{n_{ne}} \boldsymbol{N}^i(\boldsymbol{x})\, \boldsymbol{u}^i \tag{2.2.4}$$

where n_{ne} is the number of nodes of the element, $\boldsymbol{N}^i(\boldsymbol{x})$ are the interpolation or shape functions which are defined within the element volume Ω_e, and $\boldsymbol{u}^i$ is the value of the field variable at the *i*-th node. The element's shape functions can be chosen arbitrarily, yet they have to fulfil certain convergence requirements which are summarized by the concepts of completeness, compatibility and stability [11]. Most finite elements used in structural dynamics have polynomial shape functions that easily meet these requirements.

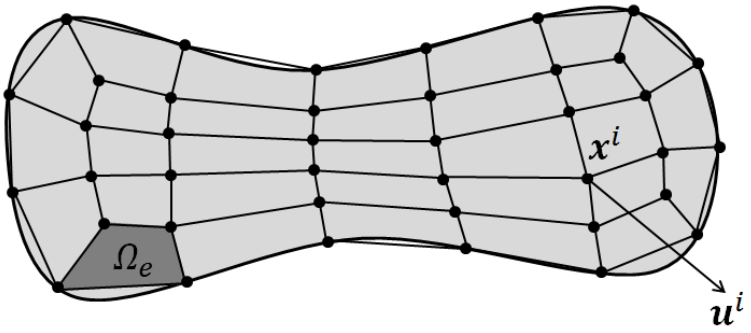


Fig. 2.2 Finite element discretization of the elastic body

A large variety of finite elements is currently available in FE software, including beam, plate shell and solid elements. In FE-based contact analysis, three-dimensional solid elements having low polynomial order of shape functions are generally used [76]. In this dissertation, the linear eight-noded solid element is used to discretize the geometry of the interacting flexible bodies. This element has three translational DOFs per node (u^i, v^i, and w^i) and belongs to the family of so-called isoparametric elements. In these elements, both the element geometry and the field variables are interpolated by the same shape functions, which in the case of the eight-noded solid element take the following general form:

$$N^i = \frac{1}{8}\left(1 + \xi\xi^i\right)\left(1 + \eta\eta^i\right)\left(1 + \mu\mu^i\right) \tag{2.2.5}$$

Note that these shape functions are defined in terms of the isoparametric parameters $\xi, \eta,$ and μ, whose values vary between -1 and 1 throughout the volume of the element. The parameters ξ^i, η^i and μ^i denote the parameter values corresponding to the *i*-th node of the element, according to the node numbering convention shown in Table 2.1.

Node	ξ	η	μ
1	-1	-1	-1
2	+1	-1	-1
3	+1	+1	-1
4	-1	+1	-1
5	-1	-1	+1
6	+1	-1	+1
7	+1	+1	+1
8	-1	+1	+1

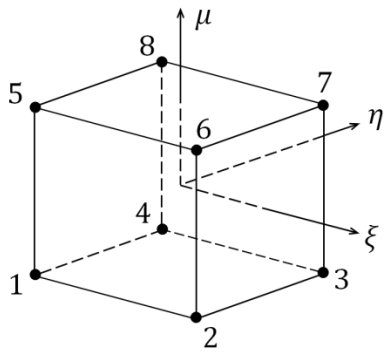


Table 2.1 Node numbering convention for the isoparametric eight-noded solid element

The isoparametric interpolation of the element's geometry and the approximated field variables within one finite element Ω_e is mathematically expressed as

$$\boldsymbol{x}(\xi,\eta,\mu)=\sum_{i=1}^{n_{ne}} N^i(\xi,\eta,\mu)\,\boldsymbol{x}^i \qquad , \quad \tilde{\boldsymbol{u}}(\boldsymbol{x}(\xi,\eta,\mu))=\sum_{i=1}^{n_{ne}} N^i(\xi,\eta,\mu)\,\boldsymbol{u}^i \tag{2.2.6}$$

where $\boldsymbol{x}^i=\left(x_1^i,x_2^i,x_3^i\right)^T$ is the vector of physical coordinates of the *i*-th node. Essential to the isoparametric concept is the mapping that relates physical coordinates $\boldsymbol{x}$ and isoparametric parameters $\boldsymbol{\xi}=(\xi,\eta,\mu)$. This mapping is expressed in terms of the Jacobian matrix of partial derivatives of $\boldsymbol{x}$ w.r.t. $\boldsymbol{\xi}$ and can easily be computed using Eq. 2.2.6 (see e.g. [11]). Finally, for solid elements having three DOFs per node, Eq. 2.2.6 (as well as Eq. 2.2.4) can be written in matrix form as follows:

$$\tilde{\boldsymbol{u}}=\sum_{i=1}^{n_{ne}} \boldsymbol{N}^i\,\boldsymbol{u}^i=\begin{bmatrix} N^1 & 0 & 0 & N^2 & \cdots & 0\\ 0 & N^1 & 0 & 0 & & 0\\ 0 & 0 & N^1 & 0 & & N^{n_{ne}}\end{bmatrix}\begin{Bmatrix} u_1^1\\ \vdots\\ u_3^{n_{ne}}\end{Bmatrix}=[\boldsymbol{N}]\boldsymbol{u}_e \tag{2.2.7}$$

where $[\boldsymbol{N}]\in\mathbb{R}^{3\times 3n_{ne}}$ is the matrix of element shape functions and $\boldsymbol{u}_e\in\mathbb{R}^{3n_{ne}\times 1}$ is the vector of nodal DOFs of the element.

2.2.3 The discretized equations of motion

Having defined the assumed approximate solution (Eq. 2.2.4) over the element domain Ω_e, the element's contribution to the weak form of the equations of motion (Eq. 2.2.3) can be evaluated. Using Galerkin's method for selecting the weighting function $\boldsymbol{w}(\boldsymbol{x})$ (i.e. the weighting functions are the element's shape functions), Eq. 2.2.3 can be written as

$$\int_{\Omega_e}\rho[\boldsymbol{N}]^T([\boldsymbol{N}]\ddot{\boldsymbol{u}}_e-\boldsymbol{b})dV+\int_{\Omega_e}([\partial][\boldsymbol{N}])^T[C][\partial][\boldsymbol{N}]\boldsymbol{u}_e dV-\int_{\Gamma_{\sigma,e}}[\boldsymbol{N}]\cdot\hat{\boldsymbol{t}}dA=0 \tag{2.2.8}$$

where use was made of the kinematic and constitutive relations defined in Eq. 2.1.8 and 2.1.12 and the matrix form introduced in Eq. 2.2.7. Note that the integrals are now taken over the volume of the element Ω_e, instead of over the full problem domain Ω. Introducing the element's mass and stiffness matrices (where N_e is the number of DOFs in the element; $N_e=3n_e$ for elements having three DOFs per node),

$$\boldsymbol{M}_e\overset{\text{def}}{=}\int_{\Omega_e}\rho[\boldsymbol{N}]^T[\boldsymbol{N}]dV\in\mathbb{R}^{N_e\times N_e} \quad , \quad \boldsymbol{K}_e\overset{\text{def}}{=}\int_{\Omega_e}([\partial][\boldsymbol{N}])^T[C][\partial][\boldsymbol{N}]dV\in\mathbb{R}^{N_e\times N_e} \tag{2.2.9}$$

Eq. 2.2.8 can be written in compact form as

$$\boldsymbol{M}_e\ddot{\boldsymbol{u}}_e + \boldsymbol{K}_e\boldsymbol{u}_e = \boldsymbol{f}_e \tag{2.2.10}$$

where the element's external force vector $\boldsymbol{f}_e$ is defined as

$$\boldsymbol{f}_e \overset{\text{def}}{=} \int_{\Omega_e} \rho[\boldsymbol{N}]^T \boldsymbol{b} dV + \int_{\Gamma_{\sigma,e}} [\boldsymbol{N}] \cdot \hat{\boldsymbol{t}} dA \in \mathbb{R}^{N_e} \tag{2.2.11}$$

Finally, evaluating the weak form of the equations of motion (Eq. 2.2.3) over the full problem domain Ω involves summing all the element contributions while taking into account the kinematic compatibility between the different elements. In doing so, the general form of the equations of motion of a flexible linear-elastic body are obtained as

$$\boldsymbol{M}_{FE}\ddot{\boldsymbol{u}}(t) + \boldsymbol{K}_{FE}\boldsymbol{u}(t) = \boldsymbol{f}_{FE}(t) \tag{2.2.12}$$

where $\boldsymbol{u} \in \mathbb{R}^N$ is the vector of all N nodal DOFs in the FE model. The global mass matrix $\boldsymbol{M}_{FE} \in \mathbb{R}^{N\times N}$, stiffness matrix $\boldsymbol{K}_{FE} \in \mathbb{R}^{N\times N}$, and external force vector $\boldsymbol{f}_{FE} \in \mathbb{R}^N$ are determined from

$$\boldsymbol{M}_{FE} = \bigcup_{e=1}^{n_e} \boldsymbol{M}_e \quad , \quad \boldsymbol{K}_{FE} = \bigcup_{e=1}^{n_e} \boldsymbol{K}_e \quad , \quad \boldsymbol{f}_{FE} = \bigcup_{e=1}^{n_e} \boldsymbol{f}_e \tag{2.2.13}$$

where the $\cup$ operator is used to denote that the summation of element quantities involves an assembly process. Eq. 2.2.12 represents the fundamental equation of linear structural FE analysis and has the form of a semi-discrete system of N coupled linear second-order Ordinary Differential Equations (ODEs).

2.3 Model reduction in structural dynamics

2.3.1 Model reduction through projection

In the previous section, the EOMs of linear-elastic solids (Eq. 2.1.10) were transformed to weak form (Eq. 2.2.3) and discretized using the FEM to obtain the fundamental equations of linear structural FE analysis (Eq. 2.2.12). The latter are formulated in terms of the nodal DOFs of the FE model, the number of which may reach up to the tens or hundreds of thousands, especially in contact analyses. Often the dynamic simulation of such high-dimensional FE models requires computational resources that are not available. In such cases, Model Order Reduction (MOR) techniques are essential to reduce the dimensions of the Full-Order Model (FOM) while preserving the main features of the original model, thereby enabling the simulation of the problem in a time-and-memory-effective way.

The MOR techniques discussed in this section belong to the class of projection-based methods [32]. In these methods the vector of nodal DOFs $\boldsymbol{u}(t) \in \mathbb{R}^N$ is approximated by projection onto a r-dimensional linear subspace $\mathcal{V}$ spanned by the basis $\boldsymbol{V}$, where $\boldsymbol{V} = [\boldsymbol{v}_1 \quad \cdots \quad \boldsymbol{v}_r] \in \mathbb{R}^{N \times r}$ and $r \ll N$. This projection is mathematically expressed as

$$\boldsymbol{u}(t) \approx \boldsymbol{V}\boldsymbol{q}_f(t) \tag{2.3.1}$$

where $\boldsymbol{q}_f \in \mathbb{R}^r$ is the vector of reduced or generalized flexible coordinates. Substitution of this equation into Eq. 2.2.12 yields

$$\boldsymbol{M}_{FE}\boldsymbol{V}\ddot{\boldsymbol{q}}_f(t) + \boldsymbol{K}_{FE}\boldsymbol{V}\boldsymbol{q}_f(t) = \boldsymbol{f}_{FE}(t) + \boldsymbol{r}(t) \tag{2.3.2}$$

where $\boldsymbol{r}(t)$ is the residual measuring the error introduced by the approximation (Eq. 2.3.1). The resulting system of ODEs now has N equations for only r unknowns. While the projection on $\mathcal{V}$ forces the *solution* of Eq. 2.2.12 to lie in $\mathcal{V}$, another r-dimensional subspace $\mathcal{W}$ with basis $\boldsymbol{W} = [\boldsymbol{w}_1 \quad \cdots \quad \boldsymbol{w}_r] \in \mathbb{R}^{N \times r}$ is selected that restricts the *system dynamics* to $\mathcal{W}$ and thereby forces the residual to be orthogonal to $\mathcal{W}$. This is done by multiplying Eq. 2.3.2 from the left with $\boldsymbol{W}^T$, which results in

$$\boldsymbol{W}^T\boldsymbol{M}_{FE}\boldsymbol{V}\ddot{\boldsymbol{q}}_f(t) + \boldsymbol{W}^T\boldsymbol{K}_{FE}\boldsymbol{V}\boldsymbol{q}_f(t) = \boldsymbol{W}^T\boldsymbol{f}_{FE}(t) + \boldsymbol{W}^T\boldsymbol{r}(t) \tag{2.3.3}$$

Taking into account the orthogonality of the residual w.r.t. the subspace $\mathcal{W}$, i.e. $\boldsymbol{W}^T\boldsymbol{r} = \boldsymbol{0}$, the above equation can be simplified to the following form:

$$\boldsymbol{M}_{WV}\ddot{\boldsymbol{q}}_f(t) + \boldsymbol{K}_{WV}\boldsymbol{q}_f(t) = \boldsymbol{f}_W(t) \tag{2.3.4}$$

where the reduced mass and stiffness matrices and the generalized force vector are defined as

$$\boldsymbol{M}_{WV} \stackrel{\text{def}}{=} \boldsymbol{W}^T\boldsymbol{M}_{FE}\boldsymbol{V} \in \mathbb{R}^{r \times r} \quad , \quad \boldsymbol{K}_{WV} \stackrel{\text{def}}{=} \boldsymbol{W}^T\boldsymbol{K}_{FE}\boldsymbol{V} \in \mathbb{R}^{r \times r} \quad , \quad \boldsymbol{f}_W \stackrel{\text{def}}{=} \boldsymbol{W}^T\boldsymbol{f}_{FE} \in \mathbb{R}^r \tag{2.3.5}$$

Eq. 2.3.4 now represents a system of r ODEs in r unknowns that describes the dynamic behaviour of the original FE model in terms of the selected basis vectors. The process of arriving from Eq. 2.2.12 to Eq. 2.3.4 is graphically depicted in Fig. 2.3. In the case where $\boldsymbol{W} = \boldsymbol{V}$, the projection is referred to as a *Galerkin* projection, whereas $\boldsymbol{W} \neq \boldsymbol{V}$ corresponds to a *Petrov-Galerkin* projection. Appropriate choices for the basis matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ are discussed in Section 2.3.3. Generally, they are obtained from system-theoretic considerations and depend on the system's mass and stiffness matrix, and on the vector of external forces (or, more specifically, the DOFs on which these forces act).

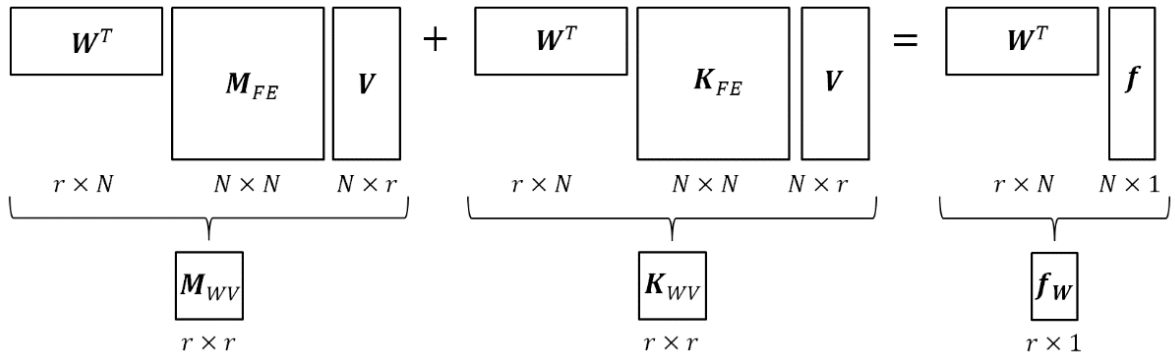


Fig. 2.3 Schematic representation of projection-based model reduction

Eq. 2.2.12 and the reduced form of Eq. 2.3.4 both represent systems of *linear* ODEs. One source of non-linearity that might enter into these equations without jeopardizing the validity of the underlying EOMs (Eq. 2.1.10) is the dependency of the external force vector on the nodal DOFs (and their temporal derivatives). This dependency occurs e.g. in contact problems where the contact forces depend on the nodal displacements of the interacting bodies, i.e. $\boldsymbol{f}_{FE} = \boldsymbol{f}_{FE}(\boldsymbol{u}(t))$. In that case, Eq. 2.3.4 should be written as

$$\boldsymbol{M}_{WV}\ddot{\boldsymbol{q}}_f(t) + \boldsymbol{K}_{WV}\boldsymbol{q}_f(t) = \boldsymbol{W}^T\boldsymbol{f}_{FE}(\boldsymbol{V}\boldsymbol{q}_f(t)) \tag{2.3.6}$$

While this equation still represents a system of r ODEs in r unknowns, the cost of computing the reduced external force vector scales with the size of the original model, i.e. with N. In order to reduce the cost of this computation, projection-based *hyperreduction* methods approximate the nonlinear function $\boldsymbol{f}_{FE}(\boldsymbol{u}(t))$ by projection onto a subspace $\mathcal{Y}$ spanned by $\boldsymbol{Y} = [\boldsymbol{y}_1 \quad \cdots \quad \boldsymbol{y}_h] \in \mathbb{R}^{N\times h}$, where $h \ll N$. This projection is mathematically expressed as:

$$\boldsymbol{f}_{FE}(\boldsymbol{u}(t)) \approx \boldsymbol{Y}\boldsymbol{c}(t) \tag{2.3.7}$$

where $\boldsymbol{c}(t) \in \mathbb{R}^h$ is the coefficient vector. The latter is typically computed by querying the nonlinear function $\boldsymbol{f}_{FE}(\boldsymbol{u}(t))$ at only a few entries, which can be expressed as $\boldsymbol{c}(t) = \boldsymbol{P}^T\boldsymbol{f}_{FE}$, where the boolean matrix $\boldsymbol{P} = [\boldsymbol{e}_{\rho_1} \quad \cdots \quad \boldsymbol{e}_{\rho_h}] \in \mathbb{R}^{N\times h}$ selects these entries and $\boldsymbol{e}_{\rho_i}$ is the ρ_i-th column of the identity matrix $\boldsymbol{I}_N$. Substitution of Eq. 2.3.7 into Eq. 2.3.6 yields the hyperreduced semi-discretized EOMs:

$$\boldsymbol{M}_{WV}\ddot{\boldsymbol{q}}_f(t) + \boldsymbol{K}_{WV}\boldsymbol{q}_f(t) = \boldsymbol{W}^T\boldsymbol{Y}\boldsymbol{P}^T\boldsymbol{f}_{FE}(\boldsymbol{u}(t)) \tag{2.3.8}$$

where the cost of computing the reduced external force vector now scales with the number of vectors in the basis matrix $\boldsymbol{Y}$, i.e. with h.

2.3.2 Model reduction of parametric systems

One common feature of projection-based model reduction methods is the *offline* (i.e. prior to the actual simulation) computation of a set of basis vectors and the projection of the governing EOMs on the subspaces spanned by these vectors. The possibly high cost of this procedure is generally justified by the computationally beneficial properties of the obtained ROM during the *online* phase of the simulation, which can span multiple simulation runs. However, often the equations of motion of a model depend on a set of parameters that may have entered the model in various ways, representing e.g. material properties, geometry and configuration parameters, and boundary conditions. As one cannot afford to create a new ROM for every change in the parameter values, a parametric ROM that is robust with respect to these parameter changes is called for. The construction of such a model is the aim of Parametric Model-Order Reduction (PMOR) [52].

The parametric form of Eq. 2.2.12 can be written as

$$\boldsymbol{M}_{FE}(\boldsymbol{\mu})\ddot{\boldsymbol{u}}(t) + \boldsymbol{K}_{FE}(\boldsymbol{\mu})\boldsymbol{u}(t) = \boldsymbol{f}_{FE}(\boldsymbol{\mu}, t) \tag{2.3.9}$$

where $\boldsymbol{\mu} \in \mathbb{R}^p$ denotes the set of p parameters. The objective is to obtain a ROM that preserves the parametric dependence of the FOM without requiring a re-evaluation of the basis matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ for every new set of parameter values. The latter is realized by introducing parametric dependence in the basis matrices themselves, for which there exist two kinds of approaches: *global* approaches and *local* approaches. Local approaches lead to ROMs having fewer DOFs then their global counterparts, yet in general their algorithmic complexity is higher.

2.3.2.1 Parametric model reduction using a global approach

In a global approach, a single pair of basis matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ is computed by sampling over a sufficiently broad range of parameters. Let $\boldsymbol{V}_1, \dots, \boldsymbol{V}_k$ and $\boldsymbol{W}_1, \dots, \boldsymbol{W}_k$ denote the (local) basis matrices that yield an optimal reduction of the FOM at the parameter realizations $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$, respectively. Then the global basis matrices are constructed by concatenating these local basis matrices, i.e.:

$$\boldsymbol{V} = [\boldsymbol{V}_1 \quad \dots \quad \boldsymbol{V}_k] \in \mathbb{R}^{N \times kr} \quad \text{and} \quad \boldsymbol{W} = [\boldsymbol{W}_1 \quad \dots \quad \boldsymbol{W}_k] \in \mathbb{R}^{N \times kr} \tag{2.3.10}$$

As this step might lead to rank-deficient global matrices, the Singular Value Decomposition (SVD) is usually applied to remove the rank-deficient components from $\boldsymbol{V}$ and $\boldsymbol{W}$, yielding global basis matrices with orthogonal columns. The resulting matrices can then be used to compute a parametric ROM via the Petrov-Galerkin projection (see Eq. 2.3.4):

$$\boldsymbol{W}^T\boldsymbol{M}_{FE}(\boldsymbol{\mu})\boldsymbol{V}\ddot{\boldsymbol{q}}_f(t) + \boldsymbol{W}^T\boldsymbol{K}_{FE}(\boldsymbol{\mu})\boldsymbol{V}\boldsymbol{q}_f(t) = \boldsymbol{W}^T\boldsymbol{f}_{FE}(\boldsymbol{\mu}, t) \tag{2.3.11}$$

with $\boldsymbol{q}_f$ as defined in Eq. 2.3.1. Note that for every change in parameter values, the reduced-order mass and stiffness matrices need to be recomputed by forming and re-projecting the corresponding full-order matrices for the new parameter values. Given the high cost of this procedure, global parametric ROMs are only effective when the parametric dependence of the FOM can be isolated from the expensive matrix operations, such that the latter can be computed offline. This is e.g. the case for systems with affine parametric dependence and for systems with sparse system matrices that can be efficiently reconstructed using hyperreduction methods (see [77]). An effective parametric ROM is also obtained when only the vector of external forces depends on $\boldsymbol{\mu}$. In that case Eq. 2.3.11 takes the following simplified form:

$$\boldsymbol{M}_{WV}\ddot{\boldsymbol{q}}_f(t) + \boldsymbol{K}_{WV}\boldsymbol{q}_f(t) = \boldsymbol{W}^T\boldsymbol{f}_{FE}(\boldsymbol{\mu}, t) \tag{2.3.12}$$

where $\boldsymbol{M}_{WV}$ and $\boldsymbol{K}_{WV}$ (Eq. 2.3.5) need to be computed only once.

2.3.2.2 Parametric model reduction using a local approach

Whereas the global approach is based on the concatenation of a sequence of local basis matrices to construct global matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ that span a broad range of parameters (Eq. 2.3.10), the local approach uses these local basis matrices themselves to construct local ROMs. Let $\boldsymbol{V}_1, \dots, \boldsymbol{V}_k \in \mathbb{R}^{N\times r}$ and $\boldsymbol{W}_1, \dots, \boldsymbol{W}_k \in \mathbb{R}^{N\times r}$ again denote the basis matrices that reduce the full-order model at the respective parameter sets $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$, and let $\boldsymbol{\mu}_{k+1}$ be a new combination of parameter values. The local approach will then construct a new local ROM for $\boldsymbol{\mu}_{k+1}$ in one of two ways: either by interpolating the reduced basis matrices to obtain new reduced bases $\boldsymbol{V}_{k+1}$ and $\boldsymbol{W}_{k+1}$, or by interpolating the reduced model matrices (e.g. the reduced mass and stiffness matrix) of the ROMs corresponding to the k precomputed reduced-order bases. Mathematically, these two local approaches can respectively be expressed as

$$\boldsymbol{W}(\boldsymbol{\mu})^T\boldsymbol{M}_{FE}(\boldsymbol{\mu})\boldsymbol{V}(\boldsymbol{\mu})\ddot{\boldsymbol{q}}_f(t) + \boldsymbol{W}(\boldsymbol{\mu})^T\boldsymbol{K}_{FE}(\boldsymbol{\mu})\boldsymbol{V}(\boldsymbol{\mu})\boldsymbol{q}_f(t) = \boldsymbol{W}(\boldsymbol{\mu})^T\boldsymbol{f}_{FE}(\boldsymbol{\mu}, t) \tag{2.3.13a}$$

$$\boldsymbol{M}_{WV}(\boldsymbol{\mu})\ddot{\boldsymbol{q}}_f(t) + \boldsymbol{K}_{WV}(\boldsymbol{\mu})\boldsymbol{q}_f(t) = \boldsymbol{W}(\boldsymbol{\mu})^T\boldsymbol{f}_{FE}(\boldsymbol{\mu}, t) \tag{2.3.13b}$$

where the interpolated matrices are defined as

$$\boldsymbol{V}(\boldsymbol{\mu}) = intp(\boldsymbol{\mu}, \boldsymbol{\mu}_1 \to \boldsymbol{V}_1, \dots, \boldsymbol{\mu}_k \to \boldsymbol{V}_k)\ ,\ \boldsymbol{W}(\boldsymbol{\mu}) = intp(\boldsymbol{\mu}, \boldsymbol{\mu}_1 \to \boldsymbol{W}_1, \dots, \boldsymbol{\mu}_k \to \boldsymbol{W}_k) \tag{2.3.14a}$$

$$\boldsymbol{M}_{WV}(\boldsymbol{\mu}) = intp(\boldsymbol{\mu}, \boldsymbol{\mu}_1 \to \boldsymbol{W}_1^T\boldsymbol{M}_{FE}(\boldsymbol{\mu}_1)\boldsymbol{V}_1, \dots, \boldsymbol{\mu}_k \to \boldsymbol{W}_k^T\boldsymbol{M}_{FE}(\boldsymbol{\mu}_k)\boldsymbol{V}_k) \tag{2.3.14b}$$

and similar for $\boldsymbol{K}_{WV}(\boldsymbol{\mu})$. In the above equations, the operator intp( ) is used to indicate generalized interpolation based on its arguments.

From Eq. 2.3.13a, it is immediately apparent that for general parametric systems the interpolation of local reduced bases comes with the same computational burden as Eq. 2.3.11, and

thus only leads to efficient ROMs under the same conditions as discussed in the previous section (i.e. when the system has affine or approximately affine parameter dependence, or when only the vector of external forces is parametric).

Congruence transformation In the case of general parametric dependence, a new ROM is usually obtained by interpolating the reduced-order system matrices of the precomputed local ROMs. As these local ROMs can be expressed in different subspaces, a so-called *congruence transformation* [78] is usually applied prior to the interpolation in order to enforce maximal consistency among the various local ROMs. The first step of this transformation involves the selection of a *reference* reduced system. One can either select one of the precomputed local reduced bases $\boldsymbol{V}_i$ and $\boldsymbol{W}_i$ [1], or construct the global matrices $\boldsymbol{V}$ and $\boldsymbol{W}$ (Eq. 2.3.10) and use their respective SVDs to identify the r dominant directions (i.e. the r singular vectors corresponding to the r largest singular values of $\boldsymbol{V}$ and $\boldsymbol{W}$, respectively) [57]. Then the transformation matrices $\boldsymbol{T}_{V,i} \in \mathbb{R}^{r\times r}$ and $\boldsymbol{T}_{W,i} \in \mathbb{R}^{r\times r}$ are computed by solving the following minimization problems (for $i = 0 \dots k$):

$$\boldsymbol{T}_{V,i} = \underset{\boldsymbol{T}}{min}\|\boldsymbol{V}_i\boldsymbol{T} - \boldsymbol{V}_R\|_F \in \mathbb{R}^{r\times r} \quad , \quad \boldsymbol{T}_{W,i} = \underset{\boldsymbol{T}}{min}\|\boldsymbol{W}_i\boldsymbol{T} - \boldsymbol{W}_R\|_F \in \mathbb{R}^{r\times r} \tag{2.3.15}$$

subject to $\boldsymbol{T}_{V,i}^T\boldsymbol{T}_{V,i} = \boldsymbol{I}_r$ and $\boldsymbol{T}_{W,i}^T\boldsymbol{T}_{W,i} = \boldsymbol{I}_r$, where $\boldsymbol{V}_R$ and $\boldsymbol{W}_R$ denote the selected reference bases and $\|\cdot\|_F$ is the Frobenius norm. This optimization problem is known as the orthogonal Procrustes[2] problem [79] and can be analytically solved using the SVD. Given the SVDs $\boldsymbol{V}_i^T\boldsymbol{V}_R = \boldsymbol{U}_{V,i}\boldsymbol{\Sigma}_{V,i}\boldsymbol{D}_{V,i}^T$ and $\boldsymbol{W}_i^T\boldsymbol{W}_R = \boldsymbol{U}_{W,i}\boldsymbol{\Sigma}_{W,i}\boldsymbol{D}_{W,i}^T$, the solutions to Eq. 2.3.15 are $\boldsymbol{T}_{V,i} = \boldsymbol{U}_{V,i}\boldsymbol{D}_{V,i}^T$ and $\boldsymbol{T}_{W,i} = \boldsymbol{U}_{W,i}\boldsymbol{D}_{W,i}^T$, respectively, and the transformed local basis matrices are defined by $\widetilde{\boldsymbol{V}}_i = \boldsymbol{V}_i\boldsymbol{T}_{V,i} \in \mathbb{R}^{N\times r}$ and $\widetilde{\boldsymbol{W}}_i = \boldsymbol{W}_i\boldsymbol{T}_{W,i} \in \mathbb{R}^{N\times r}$. Note that this transformation corresponds merely to a change of basis and does not influence the dynamics of the local ROMs. The transformed reduced-order equations are now given by

$$\widetilde{\boldsymbol{M}}_{WV}(\boldsymbol{\mu})\ddot{\widetilde{\boldsymbol{q}}}_f(t) + \widetilde{\boldsymbol{K}}_{WV}(\boldsymbol{\mu})\widetilde{\boldsymbol{q}}_f(t) = \widetilde{\boldsymbol{W}}_i^T\boldsymbol{f}_{FE}(\boldsymbol{\mu}, t) \tag{2.3.16}$$

where $\widetilde{\boldsymbol{q}}_f = \boldsymbol{T}_{V,i}^{-1}\boldsymbol{q}_f$ and where the congruence-transformed mass and stiffness matrices are defined by $\widetilde{\boldsymbol{M}}_{WV}(\boldsymbol{\mu}) = \boldsymbol{T}_{W,i}^T\boldsymbol{M}_{WV}(\boldsymbol{\mu})\boldsymbol{T}_{V,i}$ and $\widetilde{\boldsymbol{K}}_{WV}(\boldsymbol{\mu}) = \boldsymbol{T}_{W,i}^T\boldsymbol{K}_{WV}(\boldsymbol{\mu})\boldsymbol{T}_{V,i}$. In some cases, the consistency between the various local ROMs can be improved by truncating the vectors in $\boldsymbol{V}_i$ and $\boldsymbol{V}_R$ (and likewise, in $\boldsymbol{W}_i$ and $\boldsymbol{W}_R$) for which the relative subspace angles exceed a certain treshold [80].

Matrix interpolation Having enforced consistency among the local ROMs, a new ROM can be constructed through interpolation of the reduced system matrices. These matrices belong to certain matrix manifolds that are characterized by intrinsic properties such as symmetry, orthogonality, or positive definiteness. The goal of matrix interpolation in the framework of

[2] In Greek mythology, Procrustes was a bandit from Attico who made his victims fit his iron bed by stretching them or cutting off their legs.

parametric MOR is to preserve these properties in the interpolated ROM. This can be accomplished using a procedure that was first presented in [81] and that involves interpolating the reduced matrices on the tangent space to the selected manifold. Let $\boldsymbol{M}_1, \dots, \boldsymbol{M}_k$ denote a set of precomputed reduced matrices, corresponding to the parameter combinations $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$, respectively, and let $\boldsymbol{\mu}_{k+1}$ denote a new set of parameters. The corresponding interpolated reduced matrix $\boldsymbol{M}_{k+1}$ can then be obtained using the following procedure (see Fig. 2.4):

1. Select an appropriate manifold $\mathcal{M}$ for interpolating the reduced matrices $\boldsymbol{M}_1, \dots, \boldsymbol{M}_k$;
2. Choose a reference element $\boldsymbol{M}_R$ in the manifold, e.g. $\boldsymbol{M}_R = \boldsymbol{M}_1$;
3. Select a subset of reduced matrices $\boldsymbol{M}_1, \dots, \boldsymbol{M}_l$ that lie sufficiently close to $\boldsymbol{M}_R$ and map these onto the tangent space to $\mathcal{M}$ at $\boldsymbol{M}_R$ using the logarithmic mapping $\boldsymbol{\Gamma}_i = Log_{\boldsymbol{M}_R}(\boldsymbol{M}_i)$, for , $i = 1 \dots l$;
4. Interpolate the mapped matrices $\boldsymbol{\Gamma}_i$ to obtain $\boldsymbol{\Gamma}_{k+1}$, e.g. using the following generic interpolation formula: $\boldsymbol{\Gamma}_{k+1} = \sum_{i=1}^{l} \omega_i(\boldsymbol{\mu}_{k+1})\boldsymbol{\Gamma}_i$;
5. Map the interpolated matrix $\boldsymbol{\Gamma}_{k+1}$ back to the manifold $\mathcal{M}$ using the exponential mapping $\boldsymbol{M}_{k+1} = Exp_{\boldsymbol{M}_R}(\boldsymbol{\Gamma}_{k+1})$.

This procedure can be summarized in the following formula:

$$\boldsymbol{M}_{k+1} = Exp_{\boldsymbol{M}_R}\left(\sum_{i=1}^{l} \omega_i(\boldsymbol{\mu}_{k+1}) Log_{\boldsymbol{M}_R}(\boldsymbol{M}_i)\right) \tag{2.3.17}$$

Analytical expressions for the logarithmic and exponential mappings associated with some important matrix manifolds can be found in [81]. The selection of the appropriate manifold is problem dependent and is ultimately determined by the intrinsic matrix property that needs to be preserved. In structural dynamics, the reduced mass and stiffness matrices $\boldsymbol{M}_{WV}$ and $\boldsymbol{K}_{WV}$ are usually interpolated on the tangent space to the manifold of the symmetric positive definite matrices (also referred to as SPD(*n*), where *n* denotes the size of the matrices), while the basis matrix $\boldsymbol{W}$ is often interpolated on the tangent space to the Grassman manifold [54]. For SPD(*n*), the logarithmic and exponential mappings have the following form [81]:

$$\boldsymbol{\Gamma} = Log_{\boldsymbol{M}_R}(\boldsymbol{M}) = logm(\boldsymbol{M}_R^{-0.5}\boldsymbol{M}\boldsymbol{M}_R^{-0.5}) \tag{2.3.18a}$$

$$\boldsymbol{M} = Exp_{\boldsymbol{M}_R}(\boldsymbol{\Gamma}) = \boldsymbol{M}_R^{0.5} expm(\boldsymbol{\Gamma})\boldsymbol{M}_R^{0.5} \tag{2.3.18b}$$

where $\mathrm{logm}(\cdot)$ and $\mathrm{expm}(\cdot)$ denote the matrix logarithm and exponential, respectively. The manifold of real matrices has the superfluous mappings $\boldsymbol{\Gamma} = Log_{\boldsymbol{M}_R}(\boldsymbol{M}) = \boldsymbol{M}$ and $\boldsymbol{M} =$

$Exp_{\boldsymbol{M}_R}(\boldsymbol{\Gamma}) = \boldsymbol{\Gamma}$, which upon substitution in Eq. 2.3.18b yields the following straightforward interpolation formula:

$$\boldsymbol{M}_{k+1} = \sum_{i=1}^{l} \omega_i(\boldsymbol{\mu}_{k+1})\boldsymbol{M}_i \tag{2.3.19}$$

Of course, this formula guarantees no other intrinsic properties than the real-valuedness of the interpolated matrices.

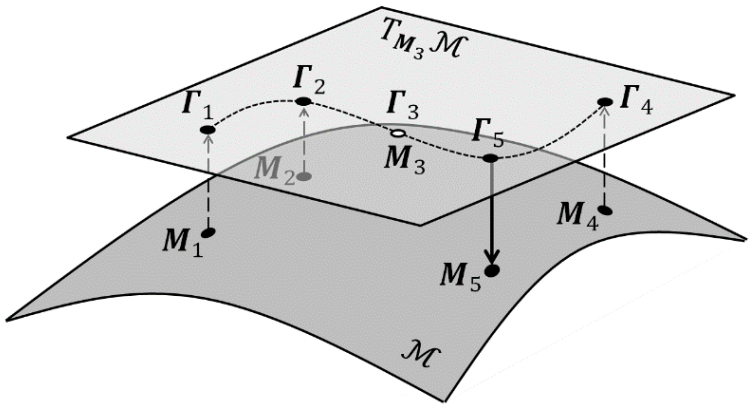


Fig. 2.4 Interpolation of $\boldsymbol{M_i}, \boldsymbol{i = 1..4}$ on a tangent space to the matrix manifold $\boldsymbol{\mathcal{M}}$ to obtain $\boldsymbol{M_5}$

2.3.2.3 Adaptive parameter sampling using a Greedy approach

This section on parametric MOR is concluded with a brief discussion on the important topic of parameter sampling. In a model reduction setting, the goal of parameter sampling is to select a minimal set of parameters $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_k$ that yields ROMs that are robust to arbitrary parameter changes. While physical insight into the system's behavior can sometimes be useful in determining the sampling locations, a more systematic approach is preferable, especially when the number of parameters is large. Standard sampling methods such as the uniform [82] or the Latin hypercube sampling method [83] quickly become too expensive as the number of samples needed to cover the parameter space grows exponentially with the number of parameters. Random sampling methods, on the other hand, can be effective but have no means of identifying important regions in the parameter space, which is essential in the construction of parameter-robust ROMs.

What is needed instead are sampling methods that are *problem-aware* and *adaptive*. The selection of parameter samples in such methods is guided by the underlying system behavior and proceeds sequentially in an adaptive manner. Additionally, these methods provide a quantifiable measure of the comprehensiveness of the selected set of parameter samples. One such method is the Greedy sampling approach [84], which proceeds as follows. Let $\boldsymbol{V}^l$ and $\boldsymbol{W}^l$ denote the concatenation of the local reduced bases $\boldsymbol{V}_1, \dots, \boldsymbol{V}_l$ and $\boldsymbol{W}_1, \dots, \boldsymbol{W}_l$ obtained by querying the full-order model (Eq. 2.3.9) at the parameters $\boldsymbol{\mu}_1, \dots, \boldsymbol{\mu}_l$. The corresponding ROM is

$$\boldsymbol{M}_{WV}^{l}(\boldsymbol{\mu})\ddot{\boldsymbol{q}}_{f}^{l}(t)+\boldsymbol{K}_{WV}^{l}(\boldsymbol{\mu})\boldsymbol{q}_{f}^{l}(t)=\boldsymbol{W}^{l^{T}}\boldsymbol{f}_{FE}(\boldsymbol{\mu},t) \tag{2.3.20}$$

where $\boldsymbol{M}_{WV}^{l}(\boldsymbol{\mu})=\boldsymbol{W}^{l^{T}}\boldsymbol{M}_{FE}(\boldsymbol{\mu})\boldsymbol{V}^{l}$ and likewise for $\boldsymbol{K}_{WV}^{l}(\boldsymbol{\mu})$. A new parameter sample $\boldsymbol{\mu}_{l+1}$ is then obtained by finding the parameters at which the error between the current ROM and the FOM is largest, i.e.

$$\boldsymbol{\mu}_{l+1}=\underset{\boldsymbol{\mu}}{arg\,max}\ \epsilon(\boldsymbol{u}(\boldsymbol{\mu},t),\boldsymbol{q}_{f}^{l}(\boldsymbol{\mu},t))=\underset{\boldsymbol{\mu}}{arg\,max}\int_{t_i}^{t_f}\left\|\boldsymbol{u}(\boldsymbol{\mu},t)-\boldsymbol{V}^{l}\boldsymbol{q}_{f}^{l}(\boldsymbol{\mu},t)\right\|_{2}^{2}dt \tag{2.3.21}$$

where t_i and t_f are the initial and final times of the time horizon of interest, and $\|\cdot\|_2$ denotes the Eucledian norm. Note that the error functional ϵ in Eq. 2.3.21 is defined in terms of the vector of nodal displacements $\boldsymbol{u}(\boldsymbol{\mu},t)$, but could as well be expressed in terms of the system's full state vector $\boldsymbol{y}=\{\boldsymbol{u}^{T}\quad\dot{\boldsymbol{u}}^{T}\}^{T}$ or a subset thereof. Once Eq. 2.3.21 is solved, the full-order model is queried at the new parameters $\boldsymbol{\mu}_{l+1}$ to obtain the new local bases $\boldsymbol{V}_{l+1}$ and $\boldsymbol{W}_{l+1}$. Then with the updated ROM (Eq. 2.3.20) the process is repeated until the maximal error determined by Eq. 2.3.21 is below a certain threshold.

Two main approaches have been developed to solve the maximization problem in Eq. 2.3.21. The first one is based on an exhaustive search over a discrete set of preselected candidate parameters, while the second approach solves the continuous form of Eq. 2.3.21 directly using a gradient-based approach. In either of these approaches, using the actual reduced model error as in Eq. 2.3.21 leads to a computationally intractable procedure, since it involves evaluating the full-order solution $\boldsymbol{u}(\boldsymbol{\mu},t)$ for a large number of parameter combinations. Instead, an inexpensive a-posteriori error estimator that correlates well with the actual error is typically used. A popular error estimator is the norm of the residual vector that is obtained by insertion of the approximation $\boldsymbol{V}^{l}\boldsymbol{q}_{f}^{l}$ in the full-order equations of motion (Eq. 2.2.12) [84], i.e.

$$\boldsymbol{r}(\boldsymbol{\mu},t,l)=\boldsymbol{M}_{FE}\boldsymbol{V}^{l}\ddot{\boldsymbol{q}}_{f}^{l}(t)+\boldsymbol{K}_{FE}\boldsymbol{V}^{l}\boldsymbol{q}_{f}^{l}(t)-\boldsymbol{f}_{FE}(t) \tag{2.3.22}$$

The new parameter combination $\boldsymbol{\mu}_{l+1}$ is then obtained by maximizing the Eucledian norm of Eq. 2.3.22 over the selected time horizon $[t_i,t_f]$.

2.3.3 Selected model-order reduction methods

The two previous sections discussed the main principles of projection-based (parametric) model reduction without reference to the various model reduction techniques themselves. These techniques are distinguished mainly by the choice of the basis matrices $\boldsymbol{V}$ and $\boldsymbol{W}$. In this section, two MOR techniques that relate to the methods developed in later chapters will be presented: the Component Mode Synthesis approach (Section 2.3.3.1) and the Proper Orthogonal

Decomposition method (Section 2.3.3.2). This excludes popular projection-based MOR methods such as moment matching [44, 85] and balanced truncation [86, 87], which are increasingly being used in structural and flexible multibody dynamics [88, 34], as well as non-linear MOR techniques such as the Trajectory Piecewise-Linear (TPWL) approach [89, 90], Global Modal Parameterization (GMP) [62, 64] and the modal derivatives approach [91]. Comprehensive overviews of MOR techniques for structural and flexible multibody dynamics can be found in [35, 92, 93].

2.3.3.1 Component Mode Synthesis

The Component Mode Synthesis (CMS) approach combines the concepts of component-level analysis and MOR. It was originally developed in the mid-1960s by Hurty [94] to enable the substructured analysis of complex structures for which the analysis as a whole was not computationally feasible at that time. Craig and Bampton [95] generalized Hurty's approach and introduced the so-called *fixed-interface* CMS methods, whereas MacNeal [96], Rubin [97], and Craig and Chang [98] developed *free-interface* methods. Nowadays, CMS is the standard approach for reducing the dimensions of structural FE models in most commercial flexible multibody software.

In the CMS approach, the dynamic behavior of a component is described in terms of a limited set of so-called *component modes*. These are the assumed mode shapes that serve as basis vectors in the projection of the nodal displacements (Eq. 2.3.1) within a component. The selection of these component modes is what differentiates one CMS method from another. In their most general application, component modes come in three types: rigid body modes $\boldsymbol{\Phi}_r$ to account for the displacement of the component as a whole without elastic deformation; vibration modes $\boldsymbol{\Phi}$ to account for the component's dynamics; and static modes $\boldsymbol{\Psi}$ to describe the interaction of the component with neighboring components. In flexible multibody dynamics, rigid body modes are usually excluded from the basis matrix, as rigid body motion is inherently included in most MBSD formulations (see Section 2.4.1). Overviews of the different component modes and the various combinations thereof to form consistent CMS reduction bases can be found in [4, 99, 100].

An essential step in the computation of component modes is the partitioning of the vector of nodal DOFs $\boldsymbol{u} \in \mathbb{R}^N$ into particular sets. A first distinction is made between the set of constrained DOFs $\boldsymbol{u}_c \in \mathbb{R}^{N_c}$ and the set of unconstrained or free DOFs $\boldsymbol{u}_f \in \mathbb{R}^{N_f}$, with $N = N_c + N_f$. The latter contains the DOFs that are free to vibrate and is further partitioned into the set of boundary (or interface) DOFs $\boldsymbol{u}_b \in \mathbb{R}^{N_b}$ (i.e. the set of DOFs through which the interaction with neighboring components takes place) and the set of internal DOFs $\boldsymbol{u}_i \in \mathbb{R}^{N_i}$, with $N_f = N_b + N_i$. Eliminating the constrained DOFs from Eq. 2.2.12 and partitioning the resulting equations, one obtains:

$$\boldsymbol{M}_{ff}\ddot{\boldsymbol{u}}_f + \boldsymbol{K}_{ff}\boldsymbol{u}_f = \boldsymbol{f}_f \quad \xrightarrow{part.} \quad \begin{bmatrix} \boldsymbol{M}_{bb} & \boldsymbol{M}_{bi} \\ \boldsymbol{M}_{ib} & \boldsymbol{M}_{ii} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{u}}_i \end{Bmatrix} + \begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{bmatrix} \begin{Bmatrix} \boldsymbol{u}_b \\ \boldsymbol{u}_i \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_b \\ \boldsymbol{0} \end{Bmatrix} \quad (2.3.23)$$

where it was assumed no external forces act on the internal DOFs ($\boldsymbol{f}_i = \mathbf{0}$).

In free-interface CMS methods, the set of vibration modes $\boldsymbol{\Phi}$ is composed of free-interface vibration modes which are obtained by solving the eigenvalue problem associated with Eq. 2.3.23:

$$\left(\boldsymbol{K}_{ff} - \omega_{f,m}^2 \boldsymbol{M}_{ff}\right)\boldsymbol{\phi}_{f,m} = \mathbf{0} \tag{2.3.24}$$

where $\omega_{f,m}^2$ and $\boldsymbol{\phi}_{f,m} \in \mathbb{R}^{N_f}$ denote the *m*-th free-interface eigenfrequency and its associated eigenmode. Generally these eigenmodes are normalized with respect to the mass matrix such that the following identities hold:

$$\boldsymbol{\Phi}_f^T \boldsymbol{M}_{ff} \boldsymbol{\Phi}_f = \boldsymbol{I}_f \quad , \quad \boldsymbol{\Phi}_f^T \boldsymbol{K}_{ff} \boldsymbol{\Phi}_f = \boldsymbol{\Lambda}_f \tag{2.3.25}$$

where $\boldsymbol{I}_f$ is the unity matrix of dimensions $N_f \times N_f$ and $\boldsymbol{\Lambda}_f$ is a diagonal matrix with the eigenvalues as diagonal entries. Fixed-interface vibration modes, on the other hand, are computed by constraining the boundary DOFs in Eq. 2.3.23 ($\boldsymbol{u}_b = \ddot{\boldsymbol{u}}_b = \mathbf{0}$) and solving the eigenvalue problem that is obtained by replacing the subscripts '*f*' in Eq. 2.3.24 by '*i*'.

With regard to the set of static modes $\boldsymbol{\Psi}$, three main types can be identified. *Constraint modes* $\boldsymbol{\Psi}_c$ are used in fixed-interface methods and are obtained by subsequently constraining all boundary DOFs but one in Eq. 2.3.23 (neglecting inertial forces), imposing a unit displacement to the latter and computing the resulting static displacement field:

$$\begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{bmatrix} \begin{Bmatrix} \boldsymbol{I}_b \\ \boldsymbol{\Psi}_{c,i} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{R}_b \\ \mathbf{0} \end{Bmatrix} \quad \rightarrow \quad \boldsymbol{\Psi}_c = \begin{Bmatrix} \boldsymbol{I}_b \\ \boldsymbol{\Psi}_{c,i} \end{Bmatrix} = \begin{bmatrix} \boldsymbol{I}_b \\ -\boldsymbol{K}_{ii}^{-1} \boldsymbol{K}_{ib} \end{bmatrix} \tag{2.3.26}$$

where $\boldsymbol{\Psi}_{c,i}$ represents the components of the constraint modes corresponding to the internal DOFs and $\boldsymbol{R}_b$ is the vector of reaction forces associated with imposing unit displacements in the boundary DOFs. *Attachment modes* $\boldsymbol{\Psi}_a$ (and their variants) are used in free-interface methods and are obtained by successively applying a unit load to the respective boundary DOFs while leaving the other DOFs unloaded:

$$\begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{bmatrix} \boldsymbol{\Psi}_a = \begin{Bmatrix} \boldsymbol{I}_b \\ \mathbf{0} \end{Bmatrix} \quad \rightarrow \quad \boldsymbol{\Psi}_a = \begin{bmatrix} \boldsymbol{K}_{bb} & \boldsymbol{K}_{bi} \\ \boldsymbol{K}_{ib} & \boldsymbol{K}_{ii} \end{bmatrix}^{-1} \begin{Bmatrix} \boldsymbol{I}_b \\ \mathbf{0} \end{Bmatrix} \tag{2.3.27}$$

Attachment modes are usually calculated such that they are guaranteed to be linearly independent of the set of kept free-interface vibration modes. One way of obtaining these so-called *residual attachment modes* $\boldsymbol{\Psi}_{ra}$ involves solving a sequence of reduced eigenvalue problems, as depicted in

Algorithm 2.1. The resulting modes are mass and stiffness orthogonal with respect to the free-interface vibration modes, i.e.

$$\boldsymbol{\Phi}_f^T \boldsymbol{M}_{ff} \boldsymbol{\Psi}_{ra} = \boldsymbol{0} \quad , \quad \boldsymbol{\Phi}_f^T \boldsymbol{K}_{ff} \boldsymbol{\Psi}_{ra} = \boldsymbol{0} \tag{2.3.28}$$

Algorithm 2.1 Procedure for obtaining residual attachment modes

Input: set of eigenmodes $\boldsymbol{\Phi}_f \in \mathbb{R}^{N_f \times n_k}$, set of attachment modes $\boldsymbol{\Psi}_a \in \mathbb{R}^{N_f \times n_a}$

Output: set of residual attachment modes $\boldsymbol{\Psi}_{ra} \in \mathbb{R}^{N_f \times n_a}$

1. **for** $j = 1 \dots n_a$ **do**
2. Assemble the matrix $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{\Phi}_f & \boldsymbol{\psi}_a^j \end{bmatrix}$
3. Compute the matrices $\boldsymbol{M}_{VV} = \boldsymbol{V}^T \boldsymbol{M}_{ff} \boldsymbol{V}$ and $\boldsymbol{K}_{VV} = \boldsymbol{V}^T \boldsymbol{K}_{ff} \boldsymbol{V}$
4. Solve the reduced eigenvalue problem $(\boldsymbol{K}_{VV} - \omega^2 \boldsymbol{M}_{VV})\boldsymbol{\Phi} = \boldsymbol{0}$
5. Mass-normalize $\boldsymbol{\Phi}$ with respect to $\boldsymbol{M}$
6. Obtain the *j*-th residual attachment mode from $\boldsymbol{\Phi} = \begin{bmatrix} \boldsymbol{\Phi}_f & \boldsymbol{\psi}_{ra}^j \end{bmatrix}$
7. **end**

Finally, *inertia relief modes* $\boldsymbol{\Psi}_{ir}$ are used in both fixed-interface and free-interface methods to account for the static deformation of a component due to rigid body acceleration of the component. One out of many ways of computing these modes (due to Hintz [101]) is by constraining all boundary DOFs and applying unit d'Alembert forces due to rigid body mode acceleration to the internal DOFs (see also [99]). Not seldom, however, the inertia-relief displacements are well-captured by the body's lower vibration modes, which is why in most commercial multibody software they are not included by default. An overview of the most commonly used components modes is shown in Fig. 2.5.

A number of statically and dynamically complete modes sets can be assembled from the component modes described above. Such mode sets allow for the exact computation of the static response of the component subjected to external forces applied to the boundary DOFs and inertial forces due to rigid-body acceleration, in addition to capturing the dominant dynamics of the component (by including a sufficient number of vibration modes). An example of a complete free-interface CMS set is the following:

$$\boldsymbol{V} = [\boldsymbol{\Phi} \quad \boldsymbol{\Psi}] = [\boldsymbol{\Phi}_f \quad \boldsymbol{\Psi}_{(r)a} \quad \boldsymbol{\Psi}_{ir}] \tag{2.3.29}$$

where the component is assumed to have no rigid-body freedom. Often static completeness with respect to rigid-body accelerations is abandoned in favor of increased sparseness of the resulting reduced-order matrices. An example of such a mode set is the popular Craig-Bampton set [95], which is composed of fixed-interface vibration modes and constraint modes.

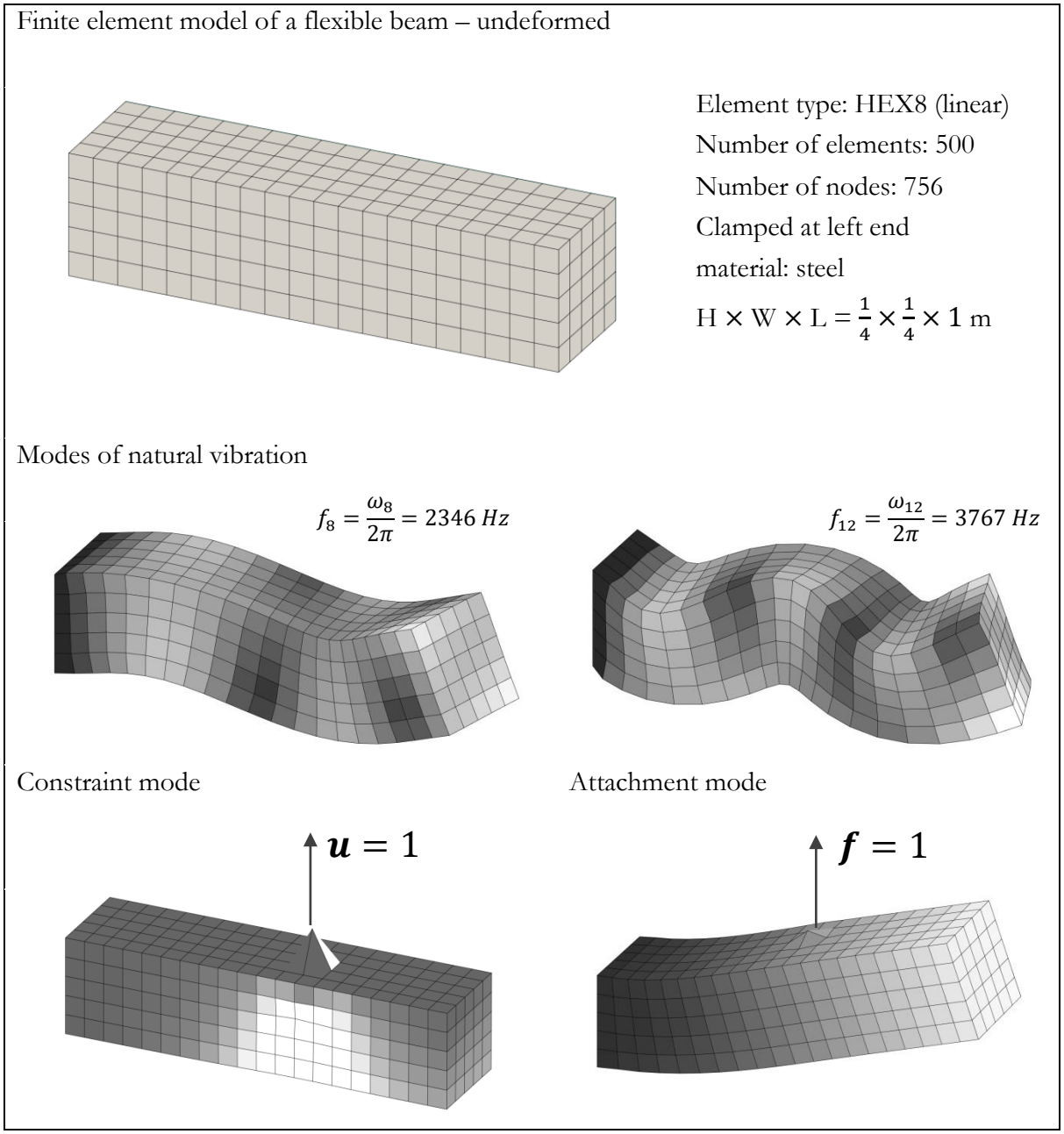


Fig. 2.5 Component modes of a flexible beam clamped at the left end

Model reduction using CMS always proceeds via a Galerkin projection of the full-order EOMs, i.e. $\boldsymbol{W} = \boldsymbol{V}$. Substitution of Eq. 2.3.29 in Eq. 2.3.3-2.3.4 yields the reduced-order equations

$$\begin{bmatrix} \boldsymbol{M}_{\Phi\Phi} & \boldsymbol{M}_{\Phi\Psi} \\ \boldsymbol{M}_{\Psi\Phi} & \boldsymbol{M}_{\Psi\Psi} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{q}}_{\Phi} \\ \ddot{\boldsymbol{q}}_{\Psi} \end{Bmatrix} + \begin{bmatrix} \boldsymbol{K}_{\Phi\Phi} & \boldsymbol{K}_{\Phi\Psi} \\ \boldsymbol{K}_{\Psi\Phi} & \boldsymbol{K}_{\Psi\Psi} \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_{\Phi} \\ \boldsymbol{q}_{\Psi} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\Phi}^T \boldsymbol{f} \\ \boldsymbol{\Psi}^T \boldsymbol{f} \end{Bmatrix} \tag{2.3.30}$$

where the reduced-order mass matrices are defined as

$$\boldsymbol{M}_{\Phi\Phi} = \boldsymbol{\Phi}^T \boldsymbol{M}_{FE} \boldsymbol{\Phi} \in \mathbb{R}^{n_k \times n_k} \ , \ \boldsymbol{M}_{\Psi\Psi} = \boldsymbol{\Psi}^T \boldsymbol{M}_{FE} \boldsymbol{\Psi} \in \mathbb{R}^{n_b \times n_b}$$

$$\boldsymbol{M}_{\Phi\Psi} = \boldsymbol{M}_{\Psi\Phi}^T = \boldsymbol{\Phi}^T \boldsymbol{M}_{FE} \boldsymbol{\Psi} \in \mathbb{R}^{n_k \times n_b} \tag{2.3.31}$$

and likewise for the reduced-order stiffness matrices. In the particular case where $\boldsymbol{V}$ is composed of mass-normalized free-interface vibration modes $\boldsymbol{\Phi}_f$ and residual attachment modes $\boldsymbol{\Psi}_{ra}$, Eq. 2.3.30 takes the following form (taking into account Eq. 2.3.25):

$$\begin{bmatrix} \boldsymbol{I}_{n_k} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{\Psi\Psi} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{q}}_\Phi \\ \ddot{\boldsymbol{q}}_\Psi \end{Bmatrix} + \begin{bmatrix} \boldsymbol{\Lambda}_f & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{\Psi\Psi} \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_\Phi \\ \boldsymbol{q}_\Psi \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{\Phi}^T \boldsymbol{f} \\ \boldsymbol{\Psi}^T \boldsymbol{f} \end{Bmatrix} \tag{2.3.32}$$

which represents a system of $n_k + n_b$ ODEs yielding a reduced-order solution to Eq. 2.2.12.

2.3.3.2 Proper Orthogonal Decomposition

Described as "a safe haven in the intimidating world of nonlinearity" [102], the Proper Orthogonal Decomposition[3] (POD) [49] is an effective and versatile method for the reduction of nonlinear and/or parametric systems. Despite its widespread use in nonlinear system analysis, the mathematical framework of the method is purely linear. More specifically, given a time-dependent, high-dimensional vector $\boldsymbol{x}(t) \in \mathbb{R}^N$ and a set of r orthonormal basis vectors $\boldsymbol{\Phi} = \{\boldsymbol{\varphi}_1 \quad \cdots \quad \boldsymbol{\varphi}_r\} \in \mathbb{R}^{N \times r}$, the POD finds an optimal approximation to $\boldsymbol{x}(t)$ based on the following linear combination:

$$\boldsymbol{x}(t) \approx \boldsymbol{x}(r,t) = \sum_{i=1}^{r} \boldsymbol{\varphi}_i \lambda_i(t) = \boldsymbol{\Phi}\boldsymbol{\lambda}(t) \text{ with} \qquad \lambda_i(t) = \boldsymbol{\varphi}_i^T \boldsymbol{x}(t) \tag{2.3.33}$$

where the set of basis vectors $\boldsymbol{\Phi}$ is found by solving the extreme-value problem:

$$\boldsymbol{\Phi} = \min_{\boldsymbol{\Phi}} E\{\|\boldsymbol{x}(t) - \boldsymbol{x}(r,t)\|^2\} \tag{2.3.34}$$

subject to $\boldsymbol{\varphi}_i^T \boldsymbol{\varphi}_j = \delta_{ij}$ (i.e. the orthonormality condition). The POD thus minimizes the averaged squared distance between the original vector $\boldsymbol{x}(t)$ and its reduced linear approximation $\boldsymbol{x}(r,t)$ (Eq. 2.3.33)[4]. In the theory of POD, the orthonormal basis vectors $\{\boldsymbol{\varphi}_1 \quad \cdots \quad \boldsymbol{\varphi}_r\}$ are called the Proper Orthogonal Modes (POMs) and the corresponding coefficients $\{\lambda_1(t) \quad \ldots \quad \lambda_r(t)\}$ are the Proper Orthogonal Values (POVs). While the linear nature of the method is appealing due to the availability of linear operator theory, it is also its main limitation when $\boldsymbol{x}(t)$ resides on a nonlinear manifold; in that case, the POD can only determine the optimal approximating *linear* manifold.

[3] Also known as the Karhunen-Loève decomposition and Principal Component Analysis (PCA) [210].

[4] As a matter of fact, among *all* linear MOR techniques, the POD obtains the minimum averaged squired distance between the original signal and its reduced representation, which is why the POD is often referred to as the optimal linear algorithm.

In practice, the data in $\boldsymbol{x}(t)$ are discretized in time and collected in a so-called snapshot matrix $\boldsymbol{X} = \{\boldsymbol{x}_1 \quad \cdots \quad \boldsymbol{x}_{n_s}\} \in \mathbb{R}^{N \times n_s}$, where n_s denotes the number of snapshots and $\boldsymbol{x}_i = \boldsymbol{x}(t_i)$. It is then a well-known fact from the theory of multivariate statistics [103] that the POMs that solve Eq. 2.3.34 are given by the r dominant eigenvectors of the covariance matrix:

$$\boldsymbol{C} = \boldsymbol{X}\boldsymbol{X}^T \in \mathbb{R}^{N \times N} \tag{2.3.35}$$

whereas the POVs are given by the associated eigenvalues. The eigenvectors of $\boldsymbol{C}$ are also the left singular vectors of the snapshot matrix $\boldsymbol{X}$. The latter can readily be confirmed by substitution of the SVD of $\boldsymbol{X}$ ($\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{D}$) into the above equation and noting that $\boldsymbol{C}$ is a symmetric positive definite matrix by definition (Eq. 2.3.35) and thus has the eigenvalue decomposition $\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T$. In doing so, one finds that $\boldsymbol{\Lambda} = \boldsymbol{\Sigma}^2$, i.e. the eigenvalues of $\boldsymbol{C}$ (and therefore also the POVs) are the squares of the singular values of the snapshot matrix $\boldsymbol{X}$. The latter provide important information regarding the energy ϵ contained in the vector sequence $\boldsymbol{X}$, which is defined via the Frobenius norm:

$$\epsilon(\boldsymbol{X}) = \|\boldsymbol{X}\|_F^2 = \sum_{i=1}^{N}\sum_{j=1}^{n_s}\left|\boldsymbol{X}_{ij}\right|^2 = trace(\boldsymbol{X}^*\boldsymbol{X}) = \sum_{i=1}^{min\,(N,n_s)} \sigma_i^2 = \sum_{i=1}^{min\,(N,n_s)} \lambda_i \tag{2.3.36}$$

where $\boldsymbol{X}^*$ denotes the conjugate transpose of $\boldsymbol{X}$ and σ_i is its *i*-th singular value. From these considerations it follows that the POVs provide a measure of the energy contained in the POMs (and therefore in the approximation $\boldsymbol{x}(r,t)$), found by truncating the summation in Eq. 2.3.36 after the r-th term.

When applied to a structural dynamics problem (Eq. 2.2.12), the POMs are determined by collecting a number of n_s snapshots of the displacement field $\boldsymbol{u}(t)$, forming the corresponding snapshot matrix $\boldsymbol{X}$ and computing its SVD $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{D}$ (see Fig. 2.6). Then the basis matrix $\boldsymbol{\Phi}$ is obtained by retaining the left singular vectors corresponding to the r largest singular values ($\boldsymbol{\Phi} = \boldsymbol{U}(:,1:r)$ in MATLAB notation) and projecting the full-order EOMs on the subspace spanned by $\boldsymbol{\Phi}$ using a Galerkin projection (i.e. $\boldsymbol{V} = \boldsymbol{W} = \boldsymbol{\Phi}$) (Eq. 2.3.3-2.3.5). In the case of parametric systems, the basis matrix can be derived from a snapshot matrix that is both time and parameter dependent, i.e. $\boldsymbol{X} = [\boldsymbol{u}(\boldsymbol{\mu}_1, t_1) \quad \cdots \quad \boldsymbol{u}(\boldsymbol{\mu}_{n_s}, t_{n_s})]$, thereby including parametric dependence in the ROM. Finally, in projection-based hyperreduction schemes (Eq. 2.3.7) the POD is often the method of choice for constructing the basis matrix $\boldsymbol{Y}$ (see Section 2.3.4.1).

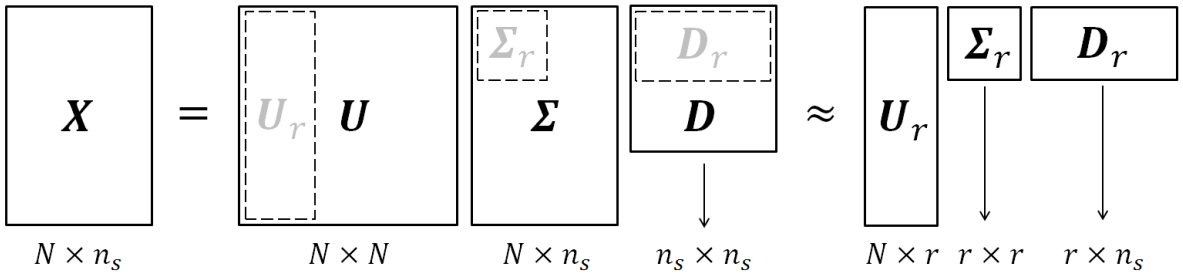


Fig. 2.6 Singular value decomposition and approximation of the snapshot matrix $\boldsymbol{X}$

As shown above, the basis matrix in a POD-based model reduction scheme is directly determined from the system response, without reference to the governing system equations or matrices (as opposed to e.g. component mode synthesis methods). The snapshot-dependent nature of POD is often seen as the weakest point of the method: firstly because the computation of a proper set of basis vectors relies entirely on the availability of adequate snapshots (and all the error measures of the method are expressed relative to these snapshots); secondly because it doesn't allow for a general physical interpretation of the POMs extracted from the decomposition[5]. Nevertheless, given its linear nature and the optimality of its approximation in terms of the squared distances (Eq. 2.3.34), the POD remains one of the most popular methods for reducing the dimensions of nonlinear and parametric systems.

2.3.4 Selected hyperreduction methods

The objective of hyperreduction is to reduce the numerical complexity of nonlinear functions through the selective evaluation of a subset of function components, as was briefly introduced at the end of Section 2.3.1. In this section, two hyperreduction techniques that relate to the techniques developed in later chapters will be discussed in more detail: the Discrete Emperical Interpolation Method (Section 2.3.4.1) and the Energy-Conserving Weighting and Sampling method (Section 2.3.4.2). Other popular hyperreduction methods like the Gauss-Newton with Approximated Tensors (GNAT) method [104, 105] and the Gappy POD method [106] will not be considered.

2.3.4.1 The Discrete Empirical Interpolation Method

In Section 2.3.1 the computational inefficiency of reducing a non-linear function $\boldsymbol{f}(\boldsymbol{u}(t)) \in \mathbb{R}^N$ using a Petrov-Galerkin projection was discussed. After projection, this function has the following form (see Eq. 2.3.6):

[5] However, in the particular case of free vibrations, a clear resemblance between the system's natural vibration modes and the POMs extracted from a sequence of free vibration snapshots was observed, while in nonlinear system analysis the POD was observed to yield the best linear representation of the system's nonlinear normal modes [213].

$$\boldsymbol{f}_W\left(\boldsymbol{q}_f(t)\right) = \boldsymbol{W}^T\boldsymbol{f}(\boldsymbol{V}\boldsymbol{q}_f(t)) \in \mathbb{R}^r \tag{2.3.37}$$

where the computational complexity of $\boldsymbol{f}_W$ depends on the cost of evaluating $\boldsymbol{f}$ (which scales with N) and of computing two matrix-vector products ($\boldsymbol{V}\boldsymbol{q}_f$ and $\boldsymbol{W}^T\boldsymbol{f}$). As a result, evaluating the reduced-order nonlinear function can be as or even more expensive then evaluating the full-order nonlinear function.

An effective way to overcome this computational burden is offered by the Discrete Empirical Interpolation Method (DEIM), which was introduced in [66] as a discrete version of the continuous EIM [107]. The main idea of DEIM is to approximate the nonlinear function $\boldsymbol{f}(t)$ by projecting it onto a subspace $\mathcal{Y}$ that is spanned by a basis of dimension $h \ll N$, i.e.

$$\boldsymbol{f}(t) \approx \boldsymbol{Y}\boldsymbol{c}(t) \tag{2.3.38}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1 \quad \cdots \quad \boldsymbol{y}_h] \in \mathbb{R}^{N\times h}$ and $\boldsymbol{c}(t) \in \mathbb{R}^h$ is the corresponding coefficient vector. Eq. 2.3.38 represents an overdetermined system of equations that can be solved by selecting h distinguished rows using the boolean matrix $\boldsymbol{P} = [\boldsymbol{e}_{\rho_1} \quad \cdots \quad \boldsymbol{e}_{\rho_h}] \in \mathbb{R}^{N\times h}$ ($\boldsymbol{e}_{\rho_i}$ is the ρ_i-th column of the identity matrix $\boldsymbol{I}_N$). Premultiplication of Eq. 2.3.38 by $\boldsymbol{P}^T$ yields

$$\boldsymbol{P}^T\boldsymbol{f}(t) = \boldsymbol{P}^T\boldsymbol{Y}\boldsymbol{c}(t) \quad \rightarrow \quad \boldsymbol{c}(t) = (\boldsymbol{P}^T\boldsymbol{Y})^{-1}\boldsymbol{P}^T\boldsymbol{f}(t) \tag{2.3.39}$$

Hence, the nonlinear function is approximated by (see Fig. 2.7)

$$\boldsymbol{f}(t) \approx \boldsymbol{Y}(\boldsymbol{P}^T\boldsymbol{Y})^{-1}\boldsymbol{P}^T\boldsymbol{f}(t) \tag{2.3.40}$$

where $\boldsymbol{P}^T\boldsymbol{f}(t)$ indicates that the nonlinear function $\boldsymbol{f}(t)$ needs to be evaluated only at the locations specified by the Boolean matrix $\boldsymbol{P}$, and $\boldsymbol{Y}(\boldsymbol{P}^T\boldsymbol{Y})^{-1} \in \mathbb{R}^{N\times h}$ is the function that interpolates these nonlinear function values at the remaining locations of $\boldsymbol{f}(t)$.

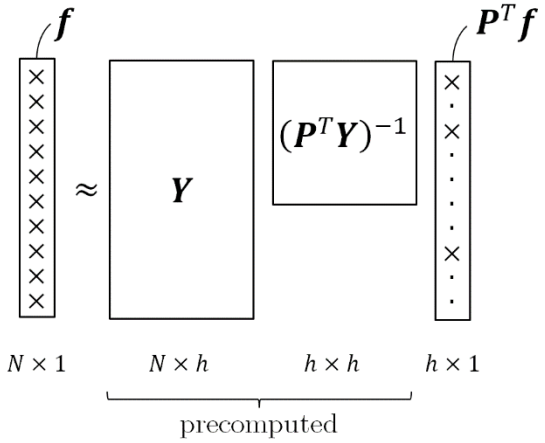


Fig. 2.7 Approximation of the nonlinear function $\boldsymbol{f}$ using the DEIM

To obtain the approximation defined by Eq. 2.3.40, two ingredients must be specified: the projection basis $\boldsymbol{Y} = [\boldsymbol{y}_1 \quad \cdots \quad \boldsymbol{y}_h] \in \mathbb{R}^{N \times h}$ and the so-called interpolation indices $\{\rho_1 \quad \cdots \quad \rho_h\}$ that select at which locations $\boldsymbol{f}(t)$ should be evaluated. The projection basis $\boldsymbol{Y}$ is generally determined using the POD method (see Section 2.3.3.2), i.e. by computing the SVD of the nonlinear snapshot matrix $\boldsymbol{X} = [\boldsymbol{f}(\boldsymbol{u}(t_1)) \quad \cdots \quad \boldsymbol{f}(\boldsymbol{u}(t_{n_s}))]$. The interpolation indices are computed inductively from the basis $\boldsymbol{Y}$ using Algorithm 2.2. The algorithm starts by setting the first interpolation index equal to the index of the largest element in $\boldsymbol{y}_1$ (lines 1-2 in Algorithm 2.2). Then the remaining indices are selected so that each of them corresponds to the largest element of the residual vector that measures the error between the next basis vector and its approximation obtained using the basis vectors that have already been considered (lines 4-7 in Algorithm 2.2). Note that it is important that the bases vectors in $\boldsymbol{Y}$ are linearly independent and ordered according to their importance (two conditions that are naturally fulfilled if the POD is used to construct $\boldsymbol{Y}$).

Algorithm 2.2 Procedure for computing the DEIM interpolation indices

Input: basis of linear independent vectors $\boldsymbol{Y} = [\boldsymbol{y}_1 \quad \cdots \quad \boldsymbol{y}_h] \in \mathbb{R}^{N \times h}$

Output: interpolation indices $\boldsymbol{\rho} = \{\rho_1 \quad \cdots \quad \rho_h\} \in \mathbb{R}^h$, boolean matrix $\boldsymbol{P} \in \mathbb{R}^{N \times h}$

1. Find index of maximum value in $|\boldsymbol{y}_1|$ and set equal to ρ_1
2. $\boldsymbol{Y}_1 = [\boldsymbol{y}_1], \boldsymbol{P} = [\boldsymbol{e}_{\rho_1}], \boldsymbol{\rho} = \{\rho_1\}$
3. **for** $i = 2 \ldots h$ **do**
4. Solve $(\boldsymbol{P}^T \boldsymbol{Y}_{i-1})\boldsymbol{c} = \boldsymbol{P}^T \boldsymbol{y}_i$
5. Compute the residual vector $\boldsymbol{r} = \boldsymbol{y}_i - \boldsymbol{Y}_{i-1}\boldsymbol{c}$
6. Find index of maximum value in $|\boldsymbol{r}|$ and set equal to ρ_i
7. $\boldsymbol{Y}_i = [\boldsymbol{Y}_{i-1} \quad \boldsymbol{y}_i], \boldsymbol{P} = [\boldsymbol{P} \quad \boldsymbol{e}_{\rho_1}], \boldsymbol{\rho} = \{\boldsymbol{\rho} \quad \rho_i\}$
8. **end**

The approximation of Eq. 2.3.40 is particularly efficient when the nonlinear function $\boldsymbol{f}(\boldsymbol{u}(t))$ is scalar-valued, i.e. when each component of $\boldsymbol{f}(\boldsymbol{u}(t))$ depends on the corresponding component of its argument $\boldsymbol{u}(t)$. In that case, the cost of evaluating $\boldsymbol{P}^T\boldsymbol{f}$ is simply the cost associated with h scalar-valued functions $f^i(\boldsymbol{V}^i\boldsymbol{q}_f(t))$, where the superscript i is used to denote the row index of the associated vector or matrix. However, in structural finite element analysis, the nonlinear function $\boldsymbol{f}(\boldsymbol{u}(t))$ is usually a vector-valued function whose components depend on the displacements of all the nodes belonging to the neighbouring elements. This can be detrimental to the efficiency of the DEIM, as computing a single component of $\boldsymbol{f}(\boldsymbol{u}(t))$ might require processing a large number of finite elements and nodal displacements (depending, respectively, on the element connectivity and the polynomial order of the finite elements).

In order to make the DEIM more amenable to structural FE analysis, the Unassembled DEIM (UDEIM) was introduced in [69] (see also [108]). The UDEIM operates on the unassembled nonlinear function $\boldsymbol{f}_u(\boldsymbol{u}_u(t))$ that is obtained by vertically stacking the various element contributions, i.e.

$$\boldsymbol{f}_u(\boldsymbol{u}_u(t)) = \left\{\boldsymbol{f}_e^1(\boldsymbol{u}_e^1(t))^T \quad \dots \quad \boldsymbol{f}_e^{n_e}\left(\boldsymbol{u}_e^{n_e}(t)\right)^T\right\}^T \tag{2.3.41}$$

where superscripts are used to denote element indices, and n_e is the number of elements. The unassembled projection matrix $\boldsymbol{Y}_u$ is obtained from the SVD of the unassembled snapshot matrix $\boldsymbol{X}_u = [\boldsymbol{f}_u(\boldsymbol{u}_u(t_1)) \quad \cdots \quad \boldsymbol{f}_u(\boldsymbol{u}_u(t_{n_s}))]$, whereas the interpolation indices are determined as before (see Algorithm 2.2, with $\boldsymbol{Y}_u$ as input). Although the unassembled nonlinear function $\boldsymbol{f}_u$ is of larger dimensions than its assembled counterpart $\boldsymbol{f}$ (resulting in a higher offline cost associated with the SVD of the unassembled snapshot matrix), each component of $\boldsymbol{f}_u$ depends on a single finite element only, which significantly decreases the online cost of the method as compared to the original DEIM. Once an approximation to the unassembled function $\boldsymbol{f}_u$ is obtained, it is transformed to its assembled form using the rectangular assembly matrix $\boldsymbol{P}_a$. The UDEIM approximation of the nonlinear function $\boldsymbol{f}(t)$ is thus obtained as:

$$\boldsymbol{f}(t) \approx \boldsymbol{P}_a\boldsymbol{Y}_u(\boldsymbol{P}_u^T\boldsymbol{Y}_u)^{-1}\boldsymbol{P}_u^T\boldsymbol{f}_u(t) \tag{2.3.42}$$

where $\boldsymbol{P}_u$ is the Boolean matrix that selects h components of the unassembled function $\boldsymbol{f}_u(t)$.

2.3.4.2 The Energy-Conserving Weighting and Sampling method

As opposed to the DEIM and other projection-based hyperreduction methods, the Energy-Conserving Weighting and Sampling (ECSW) [67, 68] method directly approximates the *reduced-order* nonlinear function $\boldsymbol{f}_W$ (Eq. 2.3.37). The starting point of the ECSW method is the element-wise construction of $\boldsymbol{f}_W$ and its approximation by a weighted summation of a subset of elemental contributions, i.e.

$$\boldsymbol{f}_W(\boldsymbol{q}_f) = \sum_{e=1}^{n_e} \boldsymbol{W}^{eT}\boldsymbol{f}^e(\boldsymbol{W}^e\boldsymbol{q}_f) \approx \sum_{E_r} \xi^e\,\boldsymbol{W}^{eT}\boldsymbol{f}^e(\boldsymbol{W}^e\boldsymbol{q}_f) \tag{2.3.43}$$

In this equation, $\boldsymbol{W}^e \in \mathbb{R}^{N_e\times r}$ is the restriction of the matrix $\boldsymbol{W} \in \mathbb{R}^{N\times r}$ to the rows associated with element e, $\boldsymbol{f}^e$ is the elemental contribution due to element e to the full-order nonlinear function $\boldsymbol{f}$, n_e is the total number of elements involved in evaluating this function, N_e is the number of DOFs associated with the element e, $E_r = \{e \in \{1,2,\dots,n_e\} : \xi^e \neq 0\}$ is the set of element indices of the so-called *reduced mesh* (i.e. these are the elements whose contribution to $\boldsymbol{f}_W$ is weighted to obtain an accurate approximation of $\boldsymbol{f}_W$), and ξ^e is the weighting factor associated with element e.

The reduced mesh and element weighting factors are obtained by considering the ROM at n_s different training configurations and forming the elemental snapshot matrix $\boldsymbol{G}$ and the summed snapshot vector $\boldsymbol{b}$:

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{G}_1 \\ \vdots \\ \boldsymbol{G}_{n_s} \end{bmatrix} = \begin{bmatrix} \boldsymbol{G}_{11} & \cdots & \boldsymbol{G}_{1n_e} \\ \vdots & \ddots & \vdots \\ \boldsymbol{G}_{n_s1} & \cdots & \boldsymbol{G}_{n_sn_e} \end{bmatrix} \in \mathbb{R}^{(r\cdot n_s)\times n_e}, \qquad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{b}_1 \\ \vdots \\ \boldsymbol{b}_{n_s} \end{bmatrix} \in \mathbb{R}^{r\cdot n_s} \tag{2.3.44}$$

where $\boldsymbol{G}_{ie}$ and $\boldsymbol{b}_i$ are defined by

$$\boldsymbol{G}_{ie} = \boldsymbol{W}^{e^T}\boldsymbol{f}^e\left(\boldsymbol{W}^e\boldsymbol{q}_f^{(i)}\right) \in \mathbb{R}^r, \qquad \boldsymbol{b}_i = \boldsymbol{f}_W\left(\boldsymbol{q}_f^{(i)}\right) = \sum_{e=1}^{n_e} \boldsymbol{G}_{ie} \in \mathbb{R}^r \tag{2.3.45}$$

and where the superscript (i) in refers to the *i*-th training configuration. Having assembled , the element weighting factors $\boldsymbol{\xi} = \{\xi^e\}, e = 1,2,\dots,n_e$ are obtained by solving the constrained minimization problem

$$\boldsymbol{\xi} = \arg\min_{\boldsymbol{\xi}} \|\boldsymbol{\xi}\|_0 \quad \text{s.t.} \quad \|\boldsymbol{G}\boldsymbol{\xi} - \boldsymbol{b}\|_2 \le \tau\|\boldsymbol{b}\|_2 \ , \boldsymbol{\xi} \ge \boldsymbol{0} \tag{2.3.46}$$

where $\|\cdot\|_0$ denotes the zero norm, which counts the number of non-zero components in the vector to which it is applied, and τ is a constant that controls the accuracy of the approximation. Setting $\tau = 0$ yields the trivial solution $\boldsymbol{\xi} = 1\forall e \in \{1,2,\dots,n_e\}$, while setting τ to the recommended value between 0.01 and 0.05 is expected to yield a sparse solution of Eq. 2.3.46, where the number of indices in E_r is considerably smaller than n_e.

An exact solution of Eq. 2.3.46 is hard to find due to the algorithmic complexity of introducing a zero norm in a constrained minimization problem. Therefore, Eq. 2.3.46 is generally reformulated as the following Non-Negative Least Squares (NNLS) problem:

$$\boldsymbol{\xi} = \arg\min_{\boldsymbol{\xi}} \|\boldsymbol{G}\boldsymbol{\xi} - \boldsymbol{b}\|_2^2 \quad \text{s.t.} \quad \boldsymbol{\xi} \ge \boldsymbol{0} \tag{2.3.47}$$

which is then solved by a sparse variant of the active-set algorithm of Lawson and Hanson [109]. In this sparse variant, a sparse solution to the above NNLS problem is found by prematurely terminating the algorithm of Lawson and Hanson when either the selected error measure drops below a user-specified threshold, or when the number of elements in the active set reaches a user-specified value. In Algorithm 2.3 the sparse NNLS solver of Lawson and Hanson with threshold-based termination criterion is depicted. Note that the above NNLS problem needs to be solved only once prior to the actual simulation.

The ECSW method distinguishes itself from other popular hyperreduction methods in that it preserves the Lagrangian structure and hence the numerical stability properties of the discretized FOM. Because of this, the method was shown to be yield time-stable ROMs in a number of applications where other hyperreduction schemes such as the DEIM was not, see e.g. [68].

Algorithm 2.3 Sparse solution to the NNLS problem

Input: snapshot matrices $\boldsymbol{G}$ and $\boldsymbol{b}$, approximation constant τ

Output: reduced mesh indices E_r and element weighting factors $\boldsymbol{\xi}$

Initialize $E_r \leftarrow \emptyset,\ Z \leftarrow \{1,2,\dots,n_e\},\ \boldsymbol{\xi} = \boldsymbol{0}$
while $\|\boldsymbol{G\xi} - \boldsymbol{b}\|_2 > \tau\|\boldsymbol{b}\|_2$ **do**
  Compute residue $\boldsymbol{r} = \boldsymbol{b} - \boldsymbol{G\xi}$
  Compute dual vector $\boldsymbol{w} = \boldsymbol{G}^T\boldsymbol{r}$
  Find index of maximum value in $\boldsymbol{w}$: $[\sim, \rho_{max}] = \max(\boldsymbol{w})$
  Add ρ_{max} to E_r, remove ρ_{max} from Z: $E_r \leftarrow E_r \cup \rho_{max},\ Z \leftarrow Z \setminus \rho_{max}$
  Solve the least squares problem: $\boldsymbol{z}_{E_r} = \boldsymbol{G}_{E_r} \backslash \boldsymbol{b}$
  while $\mathrm{sum}(\boldsymbol{z}_{E_r} < 0) > 0$
    Compute $\alpha = \min\left(\boldsymbol{\xi}/(\boldsymbol{\xi} - \boldsymbol{z}_{E_r})\right)$
    Update $\boldsymbol{\xi}$: $\boldsymbol{\xi} \leftarrow \boldsymbol{\xi} + \alpha(\boldsymbol{z}_{E_r} - \boldsymbol{\xi})$
    Set Z equal to all zero indices of $\boldsymbol{\xi}$
    $E_r \leftarrow \{1,2,\dots,n_e\} \setminus Z$
    Solve the least squares problem: $\boldsymbol{z}_{E_r} = \boldsymbol{G}_{E_r} \backslash \boldsymbol{b}$
  end
  Set $\boldsymbol{\xi}$ equal to $\boldsymbol{z}_{E_r}$: $\boldsymbol{\xi} \leftarrow \boldsymbol{z}_{E_r}$
end

2.4 Equations of motion of flexible multibody systems

2.4.1 The floating frame of reference approach

The previous section dealt with the efficient simulation of large-scale FE models in linear structural dynamics (Eq. 2.2.12). These models result from the semi-discretization of the PDEs that describe the dynamic behavior of linear-elastic objects undergoing small deformations (Eq. 2.1.10). In this section, large nonlinear rigid-body motion is added to the DOFs of these objects, still assuming deformations are small (and thus described by the continuum equations developed thus far). A mathematically elegant framework for this extension is offered by the Floating Frame of Reference (FFR) approach [6], which allows for a natural combination of the equations of structural and rigid body dynamics.

The main idea of the FFR approach can be summarized as splitting the overall motion of a flexible body into the mean rigid-body motion of a body-attached reference frame, and the superposed linear elastic deformation of the body with respect to this frame. The body's flexible deformation can therefore be described using the linear FEM (see Section 2.2), which in turn enables linear model reduction techniques to be used to reduce the number of DOFs in the resulting FE model (see Section 2.3). In accordance with the separation into rigid-body and flexible

motion, the configuration of a flexible body is identified using two sets of coordinates, namely the *reference* and *elastic* coordinates. The reference coordinates $\boldsymbol{q}_r$ define the location and orientation of the body-attached reference frame $\boldsymbol{x_1 x_2 x_3}$ relative to a global reference frame $\boldsymbol{X_1 X_2 X_3}$ that is fixed in space (see Fig. 2.8), and can be partitioned as follows:

$$\boldsymbol{q}_r = \{\boldsymbol{R}^T \quad \boldsymbol{\theta}^T\}^T \in \mathbb{R}^{n_r} \tag{2.4.1}$$

where $\boldsymbol{R} \in \mathbb{R}^3$ is a set of Cartesian coordinates that define the location of the origin of the body reference, and $\boldsymbol{\theta} \in \mathbb{R}^{n_\theta}$ is a set of rotational coordinates that describe its orientation. Among the various parameter sets that are available in MBSD theory, the three-parameter set of Bryant angles was selected in this dissertation (see Appendix C). The elastic coordinates $\boldsymbol{q}_f$ are defined by the model reduction technique that is used to reduce the number of nodal DOFs in the FE representation of the flexible body. Typically, projection-based model reduction is applied such that $\boldsymbol{q}_f$ is defined by Eq. 2.3.1, i.e. $\boldsymbol{u} \approx \boldsymbol{V q}_f$, where $\boldsymbol{u} \in \mathbb{R}^N$ is the vector of nodal DOFs, $\boldsymbol{V} \in \mathbb{R}^{N \times n_f}$ is the basis matrix of the selected MOR scheme, and $\boldsymbol{q}_f \in \mathbb{R}^{n_f}$. The total set of coordinates describing the configuration of a flexible body is therefore

$$\boldsymbol{q} = \{\boldsymbol{q}_r^T \quad \boldsymbol{q}_f^T\}^T = \{\boldsymbol{R}^T \quad \boldsymbol{\theta}^T \quad \boldsymbol{q}_f^T\}^T \in \mathbb{R}^{n_q} \tag{2.4.2}$$

In the following, it is assumed that the FE model of the flexible body is constructed using eight-noded hexahedral elements having three translational DOFs per node (see Section 2.2.2). Furthermore, the mass matrix is assumed to be lumped (i.e. all mass is assumed to be concentrated in the nodes), hence the kinetic energy of the FE model can be determined by summing the kinetic energies of the individual nodes.

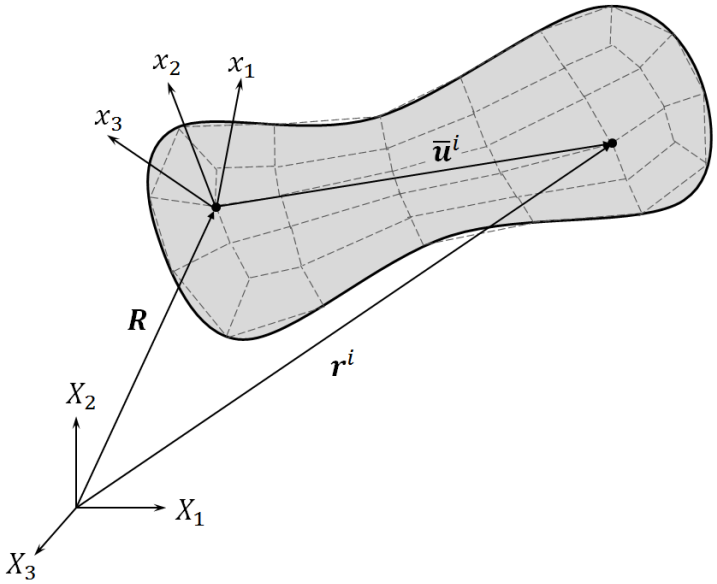


Fig. 2.8 The floating frame of reference representation of a flexible body

2.4.2 Kinetic and potential energy of a flexible body

Position The global position $\boldsymbol{r}^i$ of the *i*-th node of the flexible body (see Fig. 2.8) can be written as

$$\boldsymbol{r}^i = \boldsymbol{R} + \boldsymbol{A}\bar{\boldsymbol{r}}^i = \boldsymbol{R} + \boldsymbol{A}\left(\bar{\boldsymbol{r}}_0^i + \bar{\boldsymbol{u}}^i\right) \tag{2.4.3}$$

where $\bar{\boldsymbol{r}}^i$, $\bar{\boldsymbol{r}}_0^i$, and $\bar{\boldsymbol{u}}^i$ are, respectively, the deformed position, the undeformed position and the displacement vector of the *i*-th node with respect to the body-attached reference frame $\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$, and $\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{\theta})$ is the orthogonal transformation matrix from the body-attached reference frame to the global reference frame. Note that the bar sign ($\bar{}$) is used to denote quantities that are expressed with respect (and in) to the body-attached frame. Substitution of the approximation introduced in Eq. 2.3.1 into the above equation yields

$$\boldsymbol{r}^i = \boldsymbol{R} + \boldsymbol{A}\left(\bar{\boldsymbol{r}}_0^i + \bar{\boldsymbol{u}}^i\right) \approx \boldsymbol{R} + \boldsymbol{A}\left(\bar{\boldsymbol{r}}_0^i + \bar{\boldsymbol{V}}^i\boldsymbol{q}_f\right) \tag{2.4.4}$$

where in the remainder the '≈' sign will be replaced by the '=' sign for convenience, keeping in mind however the approximation that was introduced by Eq. 2.3.1.

Velocity The velocity of the *i*-th node can be determined by differentiating Eq. 2.4.4 with respect to time:

$$\dot{\boldsymbol{r}}^i = \dot{\boldsymbol{R}} + \dot{\boldsymbol{A}}\bar{\boldsymbol{r}}^i + \boldsymbol{A}\dot{\bar{\boldsymbol{r}}}^i = \dot{\boldsymbol{R}} + \dot{\boldsymbol{A}}\bar{\boldsymbol{r}}^i + \boldsymbol{A}\bar{\boldsymbol{V}}^i\dot{\boldsymbol{q}}_f \tag{2.4.5}$$

where use was made of the fact that $\dot{\bar{\boldsymbol{r}}}_0^i = \boldsymbol{0}$. The time derivative of the transformation matrix $\boldsymbol{A}$ is related to the skew symmetric form of the angular velocity vector $\bar{\boldsymbol{\omega}}$ in the body-attached reference frame as follows (see [6]):

$$\dot{\boldsymbol{A}} = \boldsymbol{A}\tilde{\bar{\boldsymbol{\omega}}} = \boldsymbol{A}\widetilde{\begin{Bmatrix}\omega_1\\ \omega_2\\ \omega_3\end{Bmatrix}} = \boldsymbol{A}\begin{bmatrix}0 & -\bar{\omega}_3 & \bar{\omega}_2\\ \bar{\omega}_3 & 0 & -\bar{\omega}_1\\ -\bar{\omega}_2 & \bar{\omega}_1 & 0\end{bmatrix} \tag{2.4.6}$$

where the tilde sign ($\tilde{}$) denotes the skew symmetric form of a vector, as defined by the above equation. The angular velocities in the body-attached reference frame are related to the time derivatives of the rotational coordinates through the following equation [6]:

$$\bar{\boldsymbol{\omega}} = \bar{\boldsymbol{G}}\dot{\boldsymbol{\theta}} \tag{2.4.7}$$

where $\bar{\boldsymbol{G}} = \bar{\boldsymbol{G}}(\boldsymbol{\theta})$ depends on the rotational parameters (see Appendix C). Using Eq. 2.4.6-2.4.7, Eq. 2.4.5 can be written as

$$\dot{\boldsymbol{r}}^i = \dot{\boldsymbol{R}} + \boldsymbol{A}\tilde{\bar{\boldsymbol{\omega}}}\bar{\boldsymbol{r}}^i + \boldsymbol{A}\bar{\boldsymbol{V}}^i\dot{\boldsymbol{q}}_f = \dot{\boldsymbol{R}} - \boldsymbol{A}\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{\omega}} + \boldsymbol{A}\bar{\boldsymbol{V}}^i\dot{\boldsymbol{q}}_f = \dot{\boldsymbol{R}} - \boldsymbol{A}\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{G}}\dot{\boldsymbol{\theta}} + \boldsymbol{A}\bar{\boldsymbol{V}}^i\dot{\boldsymbol{q}}_f = \boldsymbol{L}^i\dot{\boldsymbol{q}} \quad (2.4.8)$$

where $\dot{\boldsymbol{q}}$ is the time derivative of Eq. 2.4.2 and the matrix $\boldsymbol{L}^i$ is defined as

$$\boldsymbol{L}^i = [\boldsymbol{I}_3 \quad -\boldsymbol{A}\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{G}} \quad \boldsymbol{A}\bar{\boldsymbol{V}}^i] \quad (2.4.9)$$

Kinetic energy The kinetic energy T^i of the *i*-th node of the flexible body is defined by the following formula (see also Eq. 2.1.6):

$$T^i = \frac{1}{2}\dot{\boldsymbol{r}}^{i^T} m^i \dot{\boldsymbol{r}}^i = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{L}^{i^T} m^i \boldsymbol{L}^i \dot{\boldsymbol{q}} = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}^i \dot{\boldsymbol{q}} \quad (2.4.10)$$

where m^i is the (lumped) mass of the *i*-th node and $\boldsymbol{M}^i$ is the nodal mass matrix, which can be written as

$$\boldsymbol{M}^i = \boldsymbol{L}^{i^T} m^i \boldsymbol{L}^i = m^i \begin{bmatrix} \boldsymbol{I}_3 \\ \left(-\boldsymbol{A}\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{G}}\right)^T \\ \left(\boldsymbol{A}\bar{\boldsymbol{V}}^i\right)^T \end{bmatrix} [\boldsymbol{I}_3 \quad -\boldsymbol{A}\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{G}} \quad \boldsymbol{A}\boldsymbol{V}^i] = \begin{bmatrix} \boldsymbol{I}_3 & -\boldsymbol{A}\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{G}} & \boldsymbol{A}\bar{\boldsymbol{V}}^i \\ & -\bar{\boldsymbol{G}}^T\tilde{\bar{\boldsymbol{r}}}^i\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{G}} & \bar{\boldsymbol{G}}^T\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{V}}^i \\ sym. & & \bar{\boldsymbol{V}}^{i^T}\bar{\boldsymbol{V}}^i \end{bmatrix} \quad (2.4.11)$$

where use was made of the orthogonality property of $\boldsymbol{A}$, i.e. $\boldsymbol{A}^T\boldsymbol{A} = \boldsymbol{I}_3$, and of the skew symmetricness of $\tilde{\bar{\boldsymbol{r}}}^i$, i.e. $\tilde{\bar{\boldsymbol{r}}}^{i^T} = -\tilde{\bar{\boldsymbol{r}}}^i$. The kinetic energy of the flexible body is now obtained by summing the kinetic energies of all nodal point masses:

$$T = \sum_{i=1}^{n_n} T^i = \sum_{i=1}^{n_n} \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}^i \dot{\boldsymbol{q}} = \frac{1}{2}\dot{\boldsymbol{q}}^T \left[\sum_{i=1}^{n_n} \boldsymbol{M}^i\right] \dot{\boldsymbol{q}} = \frac{1}{2}\dot{\boldsymbol{q}}^T \boldsymbol{M}\dot{\boldsymbol{q}} \quad (2.4.12)$$

where the mass matrix $\boldsymbol{M}$ is obtained by summing Eq. 2.4.11 for all nodes.

Potential energy In the FFR approach, the expression for the elastic potential energy of the body takes a very simple form since only small linear-elastic displacements contribute to the elastic potential energy. Consequently, the latter is given by

$$U = \frac{1}{2}\boldsymbol{q}_f^T \boldsymbol{K}_{VV} \boldsymbol{q}_f \quad (2.4.13)$$

where $\boldsymbol{K}_{VV}$ is the reduced stiffness matrix as obtained using Eq. 2.3.5, i.e. $\boldsymbol{K}_{VV} = \boldsymbol{V}^T\boldsymbol{K}_{FE}\boldsymbol{V}$, and $\boldsymbol{K}_{FE}$ is the stiffness matrix of the FE model. The above equation can be written in terms of the full vector of generalized coordinates as follows:

$$U = \frac{1}{2}\boldsymbol{q}^T\boldsymbol{K}\boldsymbol{q} = \frac{1}{2}\{\boldsymbol{R}^T \quad \boldsymbol{\theta}^T \quad \boldsymbol{q}_f^T\}\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} \\ sym. & & \boldsymbol{K}_{VV} \end{bmatrix}\begin{Bmatrix} \boldsymbol{R} \\ \boldsymbol{\theta} \\ \boldsymbol{q}_f \end{Bmatrix} \tag{2.4.14}$$

2.4.3 Equations of motion of a flexible body

Equations of motion Having obtained expressions for the kinetic and potential energy of a flexible body, the governing EOMs can be derived by substitution of T (Eq. 2.4.12) and U (Eq. 2.4.13) into Lagrange's equation of motion (Eq. 2.1.4). If the body is subjected to certain constraints described by the algebraic equation $\boldsymbol{g}(\boldsymbol{q}) = \mathbf{0}$, where $\boldsymbol{g}(\boldsymbol{q}) \in \mathbb{R}^{n_\lambda}$ is the vector of n_λ linear independent constraint functions[6], then Lagrange's equation of motion take the following form:

$$\frac{d}{dt}\left(\frac{\partial T}{\partial \dot{\boldsymbol{q}}}\right)^T - \left(\frac{\partial T}{\partial \boldsymbol{q}}\right)^T + \left(\frac{\partial U}{\partial \boldsymbol{q}}\right)^T + \boldsymbol{G}^T\boldsymbol{\lambda} = \boldsymbol{f}_{ext} \tag{2.4.15}$$

where $\boldsymbol{G} = \boldsymbol{g}_{\boldsymbol{q}} = \partial \boldsymbol{g}/\partial \boldsymbol{q} \in \mathbb{R}^{n_q \times n_\lambda}$ is the constraint Jacobian matrix, $\boldsymbol{\lambda} \in \mathbb{R}^{n_\lambda}$ is the associated vector of Lagrange multipliers, and $\boldsymbol{f}_{ext}$ is the vector of generalized external forces. Substitution of Eq. 2.4.12 and Eq. 2.4.13 into the above equation yields

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} + \boldsymbol{G}^T\boldsymbol{\lambda} = \boldsymbol{f}_{ex} + \boldsymbol{f}_v \tag{2.4.16}$$

where the mass and stiffness matrices are defined in Eq. 2.4.11-2.4.12 and Eq. 2.4.14, respectively, and $\boldsymbol{f}_v$ is the quadratic velocity vector, which is defined as

$$\boldsymbol{f}_v = \left[\frac{\partial}{\partial \boldsymbol{q}}(\dot{\boldsymbol{q}}^T\boldsymbol{M}\dot{\boldsymbol{q}})\right]^T - \dot{\boldsymbol{M}}\dot{\boldsymbol{q}} \tag{2.4.17}$$

Substitution of Eq. 2.4.2 and 2.4.11 in the above equation yields

$$\boldsymbol{f}_v = \begin{Bmatrix} \boldsymbol{f}_{v,R} \\ \boldsymbol{f}_{v,\theta} \\ \boldsymbol{f}_{v,f} \end{Bmatrix} = \sum_{i=1}^{n_n} m^i \begin{Bmatrix} -\boldsymbol{A}\tilde{\bar{\boldsymbol{\omega}}}\tilde{\bar{\boldsymbol{\omega}}}\bar{\boldsymbol{r}}^i - 2\boldsymbol{A}\tilde{\bar{\boldsymbol{\omega}}}\bar{\boldsymbol{V}}^i\dot{\boldsymbol{q}}_f + \boldsymbol{A}\tilde{\bar{\boldsymbol{r}}}^i\dot{\bar{\boldsymbol{G}}}\dot{\boldsymbol{\theta}} \\ \bar{\boldsymbol{G}}^T\tilde{\bar{\boldsymbol{\omega}}}\tilde{\bar{\boldsymbol{r}}}^i\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{\omega}} + 2\bar{\boldsymbol{G}}^T\tilde{\bar{\boldsymbol{r}}}^i\tilde{\bar{\boldsymbol{V}}}^i\dot{\boldsymbol{q}}_f\bar{\boldsymbol{\omega}} + \bar{\boldsymbol{G}}^T\tilde{\bar{\boldsymbol{r}}}^i\tilde{\bar{\boldsymbol{r}}}^i\dot{\bar{\boldsymbol{G}}}\dot{\boldsymbol{\theta}} \\ \bar{\boldsymbol{V}}^{i\,T}\tilde{\bar{\boldsymbol{\omega}}}\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{\omega}} + 2\bar{\boldsymbol{V}}^{i\,T}\tilde{\bar{\boldsymbol{V}}}^i\dot{\boldsymbol{q}}_f\bar{\boldsymbol{\omega}} + \bar{\boldsymbol{V}}^{i\,T}\tilde{\bar{\boldsymbol{r}}}^i\dot{\bar{\boldsymbol{G}}}\dot{\boldsymbol{\theta}} \end{Bmatrix} \tag{2.4.18}$$

[6] In this dissertation, only scleronomic constraints are considered, i.e. constraints that depend on the generalized coordinates only, not on time. In the presence of time-dependent (rheonomic) constraints, an additional derivative appears in the time derivative of the constraint equations; however, the methods discussed in this and subsequent chapters do not alter fundamentally.

where $\boldsymbol{f}_{v,R}$, $\boldsymbol{f}_{v,\theta}$, and $\boldsymbol{f}_{v,f}$ represent the quadratic velocity vectors corresponding to the translational, rotational and flexible coordinates, respectively. In the computer implementation of Eq. 2.4.16, the mass matrix (Eq. 2.4.11) and the quadratic velocity vector (Eq. 2.4.18) are expressed in terms of the so-called mass invariants of the flexible bodies. These expressions are presented in Appendix D. Note that when solving Eq. 2.4.16, a viscous damping term $\boldsymbol{D}\dot{\boldsymbol{q}}$, where $\boldsymbol{D}$ represents the system's damping matrix, is generally applied to the left-hand side of the equation to model the internal material damping in the system (see Appendix E).

Eq. 2.4.16 represents a system of Differential-Algebraic Equations (DAEs). While many (mostly implicit) techniques have been developed to solve these DAEs (see Section 2.6.3), it is often desirable to convert these equations into a system of Ordinary Differential Equations (ODEs) by elimination of the constraint equations. This technique will be applied in Chapter 6 and is explained in detail in Appendix B.

Generalized external forces In order to include external forces in the EOMs (Eq. 2.4.16) it is necessary to obtain their generalized counterparts expressed in terms of the FFR kinematics. Such an expression can be obtained by computing the virtual work done by these forces in terms of physical and generalized coordinates, respectively, and equating the two. For a point force $\boldsymbol{f}^i$ acting at the *i*-th node with position vector $\boldsymbol{r}^i$, the virtual work done by the force is

$$(\delta W_{ex})_f^i = \boldsymbol{f}^i \delta \boldsymbol{r}^i = (\boldsymbol{f}_{ex})_f^i \delta \boldsymbol{q} \tag{2.4.19}$$

where the virtual displacement $\delta \boldsymbol{r}^i$ is obtained from Eq. 2.4.9 as

$$\delta \boldsymbol{r}^i = [\boldsymbol{I}_3 \quad -\boldsymbol{A}\tilde{\bar{\boldsymbol{r}}}^i \bar{\boldsymbol{G}} \quad \boldsymbol{A}\bar{\boldsymbol{V}}^i] \begin{Bmatrix} \delta \boldsymbol{R} \\ \delta \boldsymbol{\theta} \\ \delta \boldsymbol{q}_f \end{Bmatrix} = \boldsymbol{L}\delta \boldsymbol{q} \tag{2.4.20}$$

Substitution of this equation into Eq. 2.4.19 yields the following expression for the generalized force corresponding to a point force $\boldsymbol{f}^i$:

$$(\boldsymbol{f}_{ex})_f = \begin{Bmatrix} \boldsymbol{f}^i \\ \bar{\boldsymbol{G}}^T \tilde{\bar{\boldsymbol{r}}}^i \boldsymbol{A}^T \boldsymbol{f}^i \\ \bar{\boldsymbol{V}}^{iT} \boldsymbol{A}^T \boldsymbol{f}^i \end{Bmatrix} \tag{2.4.21}$$

Similarly, the virtual work done by an external moment $\boldsymbol{m}^i$ acting at $\boldsymbol{r}^i$ can be written as

$$(\delta W_{ex})_m^i = \boldsymbol{m}^i \delta \boldsymbol{\theta}^i = (\boldsymbol{f}_{ex})_m^i \delta \boldsymbol{q} \tag{2.4.22}$$

where $\delta\boldsymbol{\theta}^i$ is the virtual rotation at $\boldsymbol{r}^i$, which comprises the virtual rotation of the body-attached reference frame $\delta\boldsymbol{\theta}$ and the virtual rotation at $\boldsymbol{r}^i$ with respect to this frame $\delta\bar{\boldsymbol{\theta}}^i$e, i.e. $\delta\boldsymbol{\theta}^i = \delta\boldsymbol{\theta} + \delta\bar{\boldsymbol{\theta}}^i$. However, since the finite element nodes are assumed to have only translational DOFs, $\delta\bar{\boldsymbol{\theta}}^i = \mathbf{0}$ and $(\boldsymbol{f}_{ex})_m = \{\mathbf{0} \quad \boldsymbol{m}^{i^T} \quad \mathbf{0}\}^T$.

Rigid body motion in a plane This section on the derivation of the equations of motion of flexible bodies is concluded by considering the special case where a three-dimensional flexible body rotates about a fixed axis in space. This case is often encountered in the analysis of e.g. rotating machinery, such as gears and bearings. Let $\boldsymbol{X}_3$ be the axis of rotation, and let θ denote the angle of rotation. Using a Bryant angle formulation, the following expressions are obtained for the transformation matrix $\boldsymbol{A}$ and its time derivative $\dot{\boldsymbol{A}}$:

$$\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{X}_3, \theta) = \begin{bmatrix} c\theta & -s\theta & 0 \\ s\theta & c\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad , \quad \dot{\boldsymbol{A}} = \dot{\theta} \begin{bmatrix} -s\theta & -c\theta & 0 \\ c\theta & -s\theta & 0 \\ 0 & 0 & 0 \end{bmatrix} = \dot{\theta}\boldsymbol{B} \tag{2.4.23}$$

The global position vector of a node *i* and its derivative with respect to time can then be written as (see Eq. 2.4.3 and 2.4.8):

$$\boldsymbol{r}^i = \boldsymbol{R} + \boldsymbol{A}(\theta)\left(\bar{\boldsymbol{r}}_0^i + \bar{\boldsymbol{u}}^i\right) \approx \boldsymbol{R} + \boldsymbol{A}(\theta)\left(\bar{\boldsymbol{r}}_0^i + \bar{\boldsymbol{V}}^i\boldsymbol{q}_f\right) \tag{2.4.24}$$

$$\dot{\boldsymbol{r}}^i = \dot{\boldsymbol{R}} + \dot{\theta}\boldsymbol{B}\,\bar{\boldsymbol{r}}^i + \boldsymbol{A}\bar{\boldsymbol{V}}^i\dot{\boldsymbol{q}}_f = \boldsymbol{L}^i\dot{\boldsymbol{q}} \tag{2.4.25}$$

where $\boldsymbol{L}^i$ is now defined as (see Eq. 2.4.9):

$$\boldsymbol{L}^i = [\boldsymbol{I}_3 \quad \boldsymbol{B}\,\bar{\boldsymbol{r}}^i \quad \boldsymbol{A}\bar{\boldsymbol{V}}^i] \tag{2.4.26}$$

Having obtained an expression for the velocity of an arbitrary node in the FE representation of the flexible body, the equations of motion of the body can now be derived using a similar procedure as outlined in Section 2.4.2-2.4.3. The equations have the familiar form (see Eq. 2.4.16):

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} + \boldsymbol{G}^T\boldsymbol{\lambda} = \boldsymbol{f}_{ex} + \boldsymbol{f}_v \tag{2.4.27}$$

where the stiffness and damping matrices are defined by Eq. 2.4.14 and Eq. E.4, respectively, and the mass matrix and quadratic velocity vector are defined by

$$\boldsymbol{M} = \sum_{i=1}^{n_n} m^i \begin{bmatrix} \boldsymbol{I}_3 & \boldsymbol{B}\bar{\boldsymbol{r}}^i & \boldsymbol{A}\bar{\boldsymbol{V}}^i \\ & \bar{\boldsymbol{r}}^{i^T}\boldsymbol{B}^T\boldsymbol{B}\bar{\boldsymbol{r}}^i & \bar{\boldsymbol{r}}^{i^T}\boldsymbol{B}^T\boldsymbol{A}\bar{\boldsymbol{V}}^i \\ sym. & & \bar{\boldsymbol{V}}^{i^T}\bar{\boldsymbol{V}}^i \end{bmatrix} \tag{2.4.28}$$

$$\boldsymbol{f}_v = \sum_{i=1}^{n_n} m^i \begin{bmatrix} -\left(\dot{\theta}\tilde{\boldsymbol{B}}\bar{\boldsymbol{r}}^i + 2\boldsymbol{B}\bar{\boldsymbol{V}}^i\dot{\boldsymbol{q}}_f\right)\dot{\theta} \\ -\dot{\boldsymbol{q}}_f{}^T\bar{\boldsymbol{V}}^{iT}\left(2\boldsymbol{B}^T\boldsymbol{B}\bar{\boldsymbol{r}}^i\dot{\theta} + \boldsymbol{B}^T\boldsymbol{A}\bar{\boldsymbol{V}}^i\dot{\boldsymbol{q}}_f\right) \\ \bar{\boldsymbol{V}}^{iT}\left(\boldsymbol{B}^T\boldsymbol{B}\bar{\boldsymbol{r}}^i\dot{\theta} - 2\boldsymbol{A}^T\boldsymbol{B}\bar{\boldsymbol{V}}^i\dot{\boldsymbol{q}}_f\right)\dot{\theta} \end{bmatrix} \tag{2.4.29}$$

These expressions are formulated in terms of the body's mass invariants in Appendix D. The generalized external forces due to a point force $\boldsymbol{f}^i$ and a point moment $\boldsymbol{m}^i$ can be obtained using similar principles as discussed above:

$$(\boldsymbol{f}_{ex})_f = \begin{Bmatrix} \boldsymbol{f}^i \\ \bar{\boldsymbol{r}}^{iT}\boldsymbol{B}^T\boldsymbol{f}^i \\ \bar{\boldsymbol{V}}^{iT}\boldsymbol{A}^T\boldsymbol{f}^i \end{Bmatrix} \quad , \quad (\boldsymbol{f}_{ex})_m = \begin{Bmatrix} \boldsymbol{0} \\ \boldsymbol{m}^i \\ \boldsymbol{0} \end{Bmatrix} \tag{2.4.30}$$

The specific form of the EOMs derived in this section is used in Chapters 4 and 5 to model the dynamic contact interactions between a pair of perfectly aligned gears. In Chapter 6, the more general form of Eq. 2.4.16-2.4.18 is used to include the influence of misalignments.

2.5 Treatment of contact interactions

2.5.1 Continuum description of frictionless contact

In the previous sections, the EOMs of linear-elastic bodies undergoing small displacements were derived in strong form using Hamilton's principle (Section 2.1); these equations were then transformed to weak form and discretized in space using the FEM (Section 2.2); finally, large nonlinear rigid body motion was added to the dynamic description of these linear-elastic bodies, resulting in a system of second-order DAEs (Section 2.4). Before discussing the discretization and solution of these equations in time (Section 2.6), the current section investigates how the EOMs of linear-elastic bodies should be extended to account for contact interactions that occur between two flexible bodies.

Note that the focus of this investigation is on modelling contact between flexible bodies that have already been discretized in space using the FEM. Most monographs on computational contact modelling instead introduce the contact conditions (also referred to as the Kuhn-Tucker or Signorini conditions [10]) immediately in the strong form of the equations of motion (Eq. 2.1.10); these equations are then converted to weak form and their solutions are expressed in variational form, thereby maintaining generality of the discussion. For these more comprehensive treatments of contact problems, the interested reader is referred to the excellent monographs of [10] (general discussion, including modern topics), [110] (focus on *explicit* finite element procedures), and [111] (focus on *implicit* finite element procedures).

The contact problem is first introduced in a continuum context. Consider two elastic bodies $\mathcal{B}^b, b = 1,2$, each occupying a volume Ω^b in three-dimensional space with boundary surface Γ^b,

see Fig. 2.9. The boundary Γ^b is divided in three subregions $\Gamma_\sigma^b, \Gamma_u^b$, and Γ_c^b, where Γ_σ^b and Γ_u^b denote, respectively, the regions where boundary loads and displacements are prescribed, and Γ_c^b is the region where contact interactions can occur. When the two bodies come in contact, physics requires that all points $\boldsymbol{r}^b \in \Omega^b$ must not interpenetrate the other body, i.e. $\Omega^1 \cap \Omega^2 = \emptyset$. Additionally, it is assumed that all contact forces excerted by one body on the other are compressive in the direction normal to the contact boundary, thereby neglecting any frictional effects. The influence of frictional effects is taken into account in Appendix F.

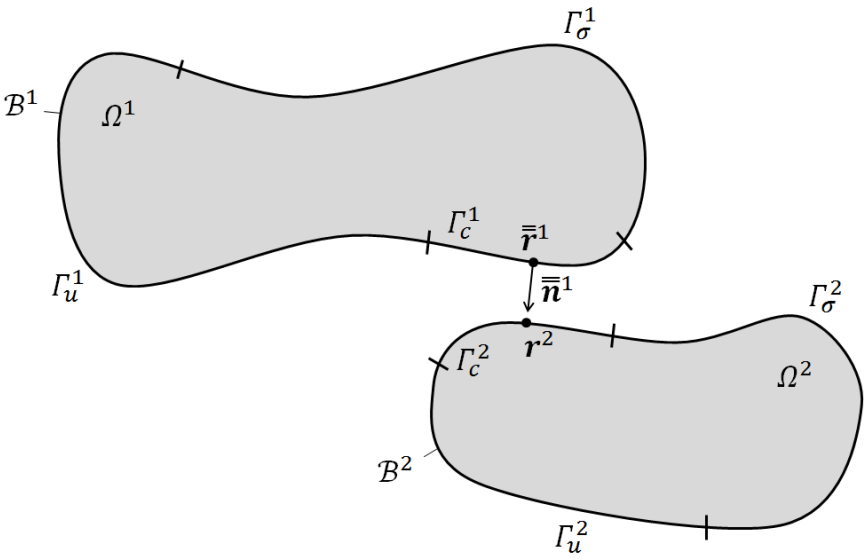


Fig. 2.9 Continuum description of two elastic bodies in contact

In order to describe these contact conditions mathematically, contact constraints should be defined for all points $\boldsymbol{r}^b \in \Gamma_c^b$. Selecting $\mathcal{B}^1$ as the *master* body and picking points $\boldsymbol{r}^2 \in \Gamma_c^2$ from the *slave* body $\mathcal{B}^2$, a set of potential contact points $\bar{\boldsymbol{r}}^1 \in \Gamma_c^1$ can be identified by solving the following minimum distance problem:

$$\bar{\bar{\boldsymbol{r}}}^1(\boldsymbol{r}^2) = \underset{\boldsymbol{r}^1 \in \Gamma_c^1}{arg\ min} \|\boldsymbol{r}^2 - \boldsymbol{r}^1\|_2 \tag{2.5.1}$$

where the double bar sign over a quantity denotes its evaluation at the minimal distance point. For each point $\bar{\bar{\boldsymbol{r}}}^1 \in \Gamma_c^1$, a local surface normal $\bar{\bar{\boldsymbol{n}}}^1$ is defined as the geometric unit normal to the surface at the point $\bar{\bar{\boldsymbol{r}}}^1$. The latter is used to define the *normal gap function* $g_N(\bar{\bar{\boldsymbol{r}}}^1)$ as:

$$g_N(\bar{\bar{\boldsymbol{r}}}^1) = (\boldsymbol{r}^2 - \bar{\bar{\boldsymbol{r}}}^1) \cdot \bar{\bar{\boldsymbol{n}}}^1 \tag{2.5.2}$$

Note that the selection of either body as master or slave doesn't influence the contact constraint definition in the continuous case; in the discretized case, however, this selection can considerably influence the outcome of the simulation.

When the gap function $g_N(\bar{\bar{\boldsymbol{r}}}^1)$ is equal to zero, the two bodies are in contact and contact tractions $\boldsymbol{t}^b$ occur at the points of contact. Newton's law of action and reaction states that the contact tractions $\boldsymbol{t}^1$ on Γ_c^1 are equal to and opposite in direction as $\boldsymbol{t}^2$, which allows one to quantify the tractions on the interface in terms of one traction vector only. Taking $\boldsymbol{t}^1$ as a reference, the component of the contact traction in the direction of the contact surface normal $\bar{\bar{\boldsymbol{n}}}^1$ can be written in terms of the Cauchy stress tensor $\boldsymbol{\sigma}^1$ as:

$$t_N(\bar{\bar{\boldsymbol{r}}}^1) = \boldsymbol{t}^1(\bar{\bar{\boldsymbol{r}}}^1) \cdot \bar{\bar{\boldsymbol{n}}}^1 = (\boldsymbol{\sigma}^1(\bar{\bar{\boldsymbol{r}}}^1) \cdot \bar{\bar{\boldsymbol{n}}}^1) \cdot \bar{\bar{\boldsymbol{n}}}^1 \tag{2.5.3}$$

where t_N is assumed to be negative in compression.

Having obtained expressions for the gap function g_N and the normal contact traction t_N, the so-called Kuhn-Tucker (or Signorini) conditions governing frictionless contact interactions can now be stated as:

$$t_N \leq 0 \tag{2.5.4a}$$

$$g_N \geq 0 \tag{2.5.4b}$$

$$t_N g_N = 0 \tag{2.5.4c}$$

which must hold for all $\boldsymbol{r}^1 \in \Gamma_c^1$ and $\boldsymbol{r}^2 \in \Gamma_c^2$. Eq. 2.5.4a refers to the fact that all contact tractions must be compressive; Eq. 2.5.4b states the impenetrability condition; and Eq. 2.5.4c is the complementarity condition that states that compressive stresses ($t_N < 0$) are only generated in the case where contact occurs, i.e. $g_N = 0$, while the stresses are zero ($t_N = 0$) when the bodies are separated, i.e. $g_N > 0$.

When the Signorini conditions (Eq. 2.5.4) are added to the EOMs of a linear-elastic body (Eq. 2.1.10), the continuum description of the small-deformation contact problem is obtained in strong form. Including these conditions in the semi-discretized formulation of the problem requires discretizing the gap function (Eq. 2.5.2) and enforcing the Signorini conditions on discretized surfaces. These two topics are discussed next.

2.5.2 Discretization of the gap function

2.5.2.1 Overview of contact detection algorithms

The gap function plays a fundamental role in imposing the contact conditions: it identifies the locations at which the bodies come into contact and thereby determines where the impenetrability constraints (Eq. 2.5.4b) are imposed. In a continuum context, the gap function is defined for all points $\boldsymbol{r}^b$ along a smooth contact surface Γ_c^b, allowing Eq. 2.5.4b to be satisfied everywhere within the contact zone. In the case of discretized contact interfaces, however, one has to select a finite

set of points along the discretized contact surfaces at which the gap function and contact conditions are to be evaluated.

Several contact detection algorithms have been developed to select these points and define the associated discretized gap functions [10]. One of the first algorithms to be applied to large displacement contact problems is the Node-To-Surface (NTS) algorithm (also referred to as the Node-To-Segment algorithm in two-dimensional contact analysis [112]). In this algorithm the contact constraints are enforced between a node on the *slave* surface and the corresponding element face on the *master* surface[7], see Fig. 2.10. An important property of the NTS-algorithm is that for hexahedral elements it only passes the contact patch test (which studies the evolution of discretization errors at the contact interface upon mesh refinement) when the two-pass version of the algorithm is used [113]. In this version, the contact conditions are evaluated twice by alternatively treating each surface as slave and master surface. A closely related approach to contact detection is offered by the Gauss Point-To-Surface (GPTS) algorithm, where the contact constraints are enforced at an arbitrary number of Gauss quadrature points on the slave surface [114]. As the single-pass version of the algorithm passes the contact patch test only up to integration error, the more accurate and robust 'two half-pass formulation' is preferred: this is a two-pass algorithm where the contact constraints are enforced only at the surface that is currently treated as the slave surface [115]. The main advantage of this algorithm over the traditional NTS algorithm is that the discretized contact surface is qualitatively well captured even with a low number of finite elements.

Instead of enforcing the contact constraints at a number of nodes or quadrature points only, Surface-To-Surface (STS) algorithms enforce these constraints in an integral manner by integrating the gap function over the contact interface [116, 117, 118, 119]. Given their integral formulation, these algorithms rely on the weak form of the contact problem (see Section 2.2.1). Important features of the majority of STS algorithms include the intermediate contact interface at which the contact quantities are defined and integrated, and the segmentation of the contacting surfaces in the case of three-dimensional contact problems [10]. Among these algorithms, most research efforts nowadays are devoted to the so-called Mortar methods [120, 113, 8]. Owing to their rigorous mathematical background and consistency, they exhibit superior accuracy and stability properties as compared to NTS-based and other STS-based contact approaches. A major drawback of these methods, however, is their high computational cost and complexity[8], mainly stemming from the computation of the so-called Mortar integrals.

[7] In a variational context, one might state that the NTS approach collocates the contact integrals at the nodes instead of integrating them over the full element surface.

[8] Ted Laursen, one of the authors of the Mortar method, was once cited saying that implementing Mortar methods for a general case is a "true nightmare" [197].

2.5.2.2 Node-to-surface contact detection

In dynamic applications where a large number of linear finite elements is required to describe the geometry of the contact surfaces, the computational efficiency and versatility of NTS-based algorithms outweigh the higher accuracy of STS-based contact algorithms. Therefore, in this thesis a two-pass NTS-based contact detection algorithm is used to enforce the contact constraints. A mathematical description of this algorithm is presented in the following paragraphs.

For clarity of exposition, the following notational convention is introduced: let $\boldsymbol{r}^{bi}$ be the position vector of a node i on a body $\mathcal{B}^b$ expressed with respect to a global reference frame $\boldsymbol{X}_1\boldsymbol{X}_2\boldsymbol{X}_3$. A local reference frame $\boldsymbol{x}_1^b\boldsymbol{x}_2^b\boldsymbol{x}_3^b$ is attached to body $\mathcal{B}^b$ with origin at $\boldsymbol{R}^b$ and transformation matrix $\boldsymbol{A}^b$, expressed in terms of the rotational coordinates $\boldsymbol{\theta}^b$ (see Appendix C). In this body-attached reference frame, the node has the undeformed position vector $\bar{\boldsymbol{r}}_0^{bi}$ and displacement vector $\bar{\boldsymbol{u}}^{bi}$, which can be expressed in terms of $\boldsymbol{V}^{bi}$, i.e. the rows of the basis matrix $\boldsymbol{V}^b$ corresponding to node i. With reference to Eq. 2.4.4, the position vector $\boldsymbol{r}^{bi}$ can thus be written as:

$$\boldsymbol{r}^{bi} = \boldsymbol{R}^b + \boldsymbol{A}^b\left(\bar{\boldsymbol{r}}_0^{bi} + \bar{\boldsymbol{u}}^{bi}\right) \approx \boldsymbol{R}^b + \boldsymbol{A}^b\left(\bar{\boldsymbol{r}}_0^{bi} + \bar{\boldsymbol{V}}^{bi}\boldsymbol{q}_f^b\right) \tag{2.5.5}$$

where $\boldsymbol{q}_f^b$ denotes the vector of generalized elastic coordinates of body $\mathcal{B}^b$. In the following, kinematic quantities corresponding to the slave (master) body will be denoted by the superscript '*s*' ('*m*'). Note that when the superscript '*i*' in $\boldsymbol{r}^{bi}$ is omitted, an arbitrary point on body $\mathcal{B}^b$ is referred to.

Let $\mathcal{B}^s$ and $\mathcal{B}^m$ denote, respectively, the discretized slave and master bodies with contact surfaces Γ_c^s and Γ_c^m, and consider a slave node $S^i \in \Gamma_c^s$ with position vector $\boldsymbol{r}^{si}$ in the vicinity of Γ_c^m, see Fig. 2.10. Assume the bodies are discretized using eight-noded solid elements (see Section 2.2.2) and let $\mathcal{F}^{m,i}$ denote the face of a master element in the vicinity of S^i. The geometry of the face $\mathcal{F}^{m,i}$ is described by the location of its corner nodes $M^{l,i}$ $(l = 1, \dots, 4)$ and the element's isoparametric shape functions $N^{l,i}(\boldsymbol{\xi})$, where $\boldsymbol{\xi} = (\xi, \eta, \mu)$ defines the isoparametric mapping. Assuming $\mathcal{F}^{m,i}$ is described by the parameter $\mu = 1$ (see Table 2.1, Section 2.2.2), an arbitrary point $\boldsymbol{r}^{m,i}$ on $\mathcal{F}^{m,i}$ is described by the equation (see Eq. 2.2.4-2.2.5):

$$\boldsymbol{r}^{m,i} = \boldsymbol{r}^{m,i}(\xi, \eta) = \frac{1}{4}\sum_{l=1}^{4} \boldsymbol{r}^{ml,i}\left(1 + \xi\xi^l\right)\left(1 + \eta\eta^l\right) = \frac{1}{4}\sum_{l=1}^{4} N^{l,i}(\xi, \eta)\boldsymbol{r}^{ml,i} \tag{2.5.6}$$

where ξ^l and η^l are the parameters corresponding to the corner nodes $\boldsymbol{r}^{ml,i}$, and $N^{l,i}(\xi, \eta) = \left(1 + \xi\xi^l\right)\left(1 + \eta\eta^l\right)$. The point $\bar{\boldsymbol{r}}^{m,i} \in \mathcal{F}^{m,i}$ that minimizes the distance between S^i and $\mathcal{F}^{m,i}$ is found by solving the minimal distance problem (Eq. 2.5.1), which corresponds to finding the

normal projection of S^i on $\mathcal{F}^{m,i}$. At this point, the tangent vectors are perpendicular to the difference vector $\boldsymbol{g}(\xi,\eta) = \boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}(\xi,\eta)$, which can be expressed as

$$\frac{\partial}{\partial\xi}\boldsymbol{r}^{m,i}(\bar{\bar{\xi}},\bar{\eta})\cdot\left[\boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}(\bar{\bar{\xi}},\bar{\eta})\right] = \boldsymbol{0} \tag{2.5.7a}$$

$$\frac{\partial}{\partial\eta}\boldsymbol{r}^{m,i}(\bar{\bar{\xi}},\bar{\eta})\cdot\left[\boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}(\bar{\bar{\xi}},\bar{\eta})\right] = \boldsymbol{0} \tag{2.5.7b}$$

where $\bar{\bar{\xi}}$ and $\bar{\eta}$ are the parameters of the minimal distance point.

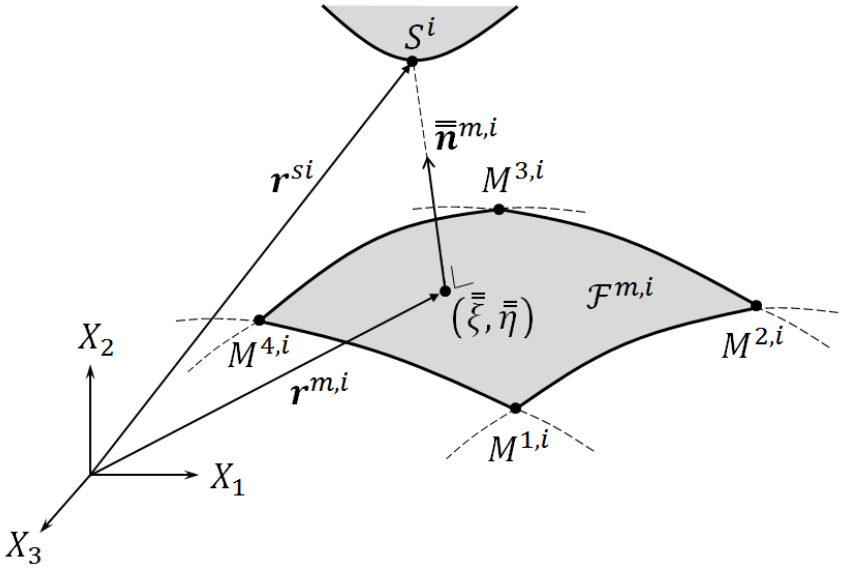


Fig. 2.10 Note-To-Surface contact detection algorithm

Substitution of Eq. 2.5.6 into Eq. 2.5.7 yields:

$$\frac{1}{4}\sum_{l=1}^{4}\boldsymbol{r}^{ml,i}\xi^l\left(1+\bar{\bar{\eta}}\eta^l\right)\cdot\left[\boldsymbol{r}^{si} - \frac{1}{4}\sum_{l=1}^{4}N^{l,i}(\xi,\eta)\boldsymbol{r}^{ml,i}\right] = \boldsymbol{0} \tag{2.5.8a}$$

$$\frac{1}{4}\sum_{l=1}^{4}\boldsymbol{r}^{ml,i}\eta^l\left(1+\bar{\bar{\xi}}\xi^l\right)\cdot\left[\boldsymbol{r}^{si} - \frac{1}{4}\sum_{l=1}^{4}N^{l,i}(\xi,\eta)\boldsymbol{r}^{ml,i}\right] = \boldsymbol{0} \tag{2.5.8b}$$

This equation is solved for $\bar{\bar{\xi}}$ and $\bar{\eta}$ using Newton's method. Once the parameters $\bar{\bar{\xi}}$ and $\bar{\eta}$ are determined, the unit normal vector at the point $\boldsymbol{r}^{m,i}(\bar{\bar{\xi}},\bar{\eta})$ is determined from

$$\bar{\bar{\boldsymbol{n}}}^{m,i} = \boldsymbol{n}^{m,i}(\xi,\eta) = \frac{\boldsymbol{r}^{m,i}_{,\xi}(\bar{\bar{\xi}},\bar{\eta})\times\boldsymbol{r}^{m,i}_{,\eta}(\bar{\bar{\xi}},\bar{\eta})}{\left\|\boldsymbol{r}^{m,i}_{,\xi}(\bar{\bar{\xi}},\bar{\eta})\times\boldsymbol{r}^{m,i}_{,\eta}(\bar{\bar{\xi}},\bar{\eta})\right\|} \tag{2.5.9}$$

where $\boldsymbol{r}_{,\xi}^{m,i} = \partial \boldsymbol{r}^{m,i}/\partial\xi$ and $\boldsymbol{r}_{,\eta}^{m,i} = \partial \boldsymbol{r}^{m,i}/\partial\eta$. When $\left\|\boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}(\bar{\bar{\xi}},\bar{\eta})\right\| \neq 0$ (i.e. S^i is not on $\mathcal{F}^{m,i}$), the following expression can be used to compute the unit normal:

$$\bar{\bar{\boldsymbol{n}}}^{m,i} = \boldsymbol{n}^{m,i}(\xi,\eta) = \frac{\boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}(\bar{\bar{\xi}},\bar{\eta})}{\left\|\boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}(\bar{\bar{\xi}},\bar{\eta})\right\|} \tag{2.5.10}$$

which requires considerably less Floating Point Operations (FLOPs) as compared to Eq. 2.5.9. Finally, having obtained $\bar{\bar{\boldsymbol{n}}}^{m,i}$ and $\boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)$, the gap function (Eq. 2.5.2) can be evaluated:

$$g_N^i\left(\boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)\right) = \left(\boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)\right)\cdot\bar{\bar{\boldsymbol{n}}}^{m,i} \tag{2.5.11}$$

which is then use to enforce the impenetrability constraint (Eq. 2.5.4b). Three approaches for enforcing this constraint are discussed in Section 2.5.3.

2.5.2.3 Derivative of the gap function

Often the derivative of the normal gap function (Eq. 2.5.10) with respect to the vector of generalized coordinates (Eq. 2.4.2) is required (see Section 2.5.3). As the variation of the unit vector $\delta\bar{\bar{\boldsymbol{n}}}^{m,i}$ is perpendicular to the difference vector $\boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)$ [10], the derivative of Eq. 2.5.11 with respect to the vector of generalized coordinates $\boldsymbol{q}$ can be written as

$$\frac{\partial}{\partial\boldsymbol{q}}g_N^i\left(\boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)\right) = \left(\frac{\partial\boldsymbol{r}^{si}}{\partial\boldsymbol{q}} - \frac{\partial\boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)}{\partial\boldsymbol{q}} - \frac{\partial\boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)}{\partial\bar{\bar{\boldsymbol{\xi}}}}\frac{\partial\bar{\bar{\boldsymbol{\xi}}}}{\partial\boldsymbol{q}}\right)\cdot\bar{\bar{\boldsymbol{n}}}^{m,i} \tag{2.5.12}$$

where $\boldsymbol{q}^T = \left\{\boldsymbol{q}^{s^T} \quad \boldsymbol{q}^{m^T}\right\}$ and $\bar{\bar{\boldsymbol{\xi}}} = \left(\bar{\bar{\xi}},\bar{\eta}\right)$. Considering that $\left(\partial\boldsymbol{r}^{m,i}/\partial\bar{\bar{\boldsymbol{\xi}}}\right)\cdot\bar{\bar{\boldsymbol{n}}}^{m,i} = \boldsymbol{0}$ [10], as follows readily from Eq. 2.5.9, the above equation reduces to

$$\frac{\partial}{\partial\boldsymbol{q}}g_N^i\left(\boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)\right) = \left(\frac{\partial\boldsymbol{r}^{si}}{\partial\boldsymbol{q}} - \frac{\partial\boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)}{\partial\boldsymbol{q}}\right)\cdot\bar{\bar{\boldsymbol{n}}}^{m,i} \tag{2.5.13}$$

The position vectors $\boldsymbol{r}^s$ and $\boldsymbol{r}^m$ are given by Eq. 2.5.5, i.e.

$$\boldsymbol{r}^{si} = \boldsymbol{R}^s + \boldsymbol{A}^s\left(\bar{\boldsymbol{r}}_0^{si} + \bar{\boldsymbol{V}}^{si}\boldsymbol{q}_f^s\right) \tag{2.5.14a}$$

$$\boldsymbol{r}^{m,i}\left(\bar{\bar{\xi}},\bar{\eta}\right) = \frac{1}{4}\sum_{l=1}^{4}N^{l,i}\left(\bar{\bar{\xi}},\bar{\eta}\right)\left(\boldsymbol{R}^m + \boldsymbol{A}^m\left(\bar{\boldsymbol{r}}_0^{ml} + \bar{\boldsymbol{V}}^{ml}\boldsymbol{q}_f^m\right)\right) \tag{2.5.14b}$$

The derivatives of these vectors with respect to $\boldsymbol{q}$ are given by

$$\frac{\partial \boldsymbol{r}^{si}}{\partial \boldsymbol{q}} = \begin{bmatrix} \boldsymbol{I}_3 & -\boldsymbol{A}^s \tilde{\bar{\boldsymbol{r}}}^{si} \bar{\boldsymbol{G}}^s & \boldsymbol{A}^s \bar{\boldsymbol{V}}^{si} & \boldsymbol{0}_{3\times n_{q^m}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{L}^{si} & \boldsymbol{0}_{3\times n_{q^m}} \end{bmatrix} \tag{2.5.15a}$$

$$\begin{aligned} \frac{\partial \boldsymbol{r}^{m,i}}{\partial \boldsymbol{q}} &= \frac{1}{4}\sum_{l=1}^{4} N^{l,i}\left(\bar{\bar{\xi}}, \bar{\bar{\eta}}\right) \begin{bmatrix} \boldsymbol{0}_{3\times n_{q^s}} & \boldsymbol{I}_3 & -\boldsymbol{A}^m \tilde{\bar{\boldsymbol{r}}}^{ml} \bar{\boldsymbol{G}}^m & \boldsymbol{A}^m \bar{\boldsymbol{V}}^{ml} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{0}_{3\times n_{q^s}} & \frac{1}{4}\sum_{l=1}^{4} N^{l,i}\left(\bar{\bar{\xi}}, \bar{\bar{\eta}}\right) \boldsymbol{L}^{ml} \end{bmatrix} \end{aligned} \tag{2.5.15b}$$

where n_{q^s} and n_{q^m} denote, respectively, the number of generalized coordinates of the slave and master bodies, and where $\boldsymbol{L}^{si}$ and $\boldsymbol{L}^{ml}$ are defined by Eq. 2.4.9.

2.5.3 Enforcement of the contact constraints

Having obtained the (possible) contact locations through evaluation of the discretized gap function, the contact conditions (Eq. 2.5.4) can be incorporated into the EOMs of the interacting bodies. Different formulations exist to enforce these contact constraints, the most popular being the Lagrange multiplier and penalty methods, which will be discussed next. For an extensive overview of contact constraint enforcement methods, the interested reader is referred to [10].

2.5.3.1 Lagrange multiplier method

In the Lagrange multiplier method, the impenetrability constraint (Eq. 2.5.4b) is treated like any other algebraic constraint ($\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$, see Eq. 2.4.15b), and a Lagrange multiplier $\boldsymbol{\lambda}$ is added to the equations of motion for each slave node that is in contact with the master body. The inequality constraint of Eq. 2.5.4b is transformed to an equality constraint using an *active set strategy*, i.e. a procedure which selects the slave nodes that are in contact at the considered instant. Hence, for a flexible multibody system having no other than contact constraints, the static contact equilibrium[9] can be expressed as:

$$\boldsymbol{K}\boldsymbol{q} + \boldsymbol{G}(\boldsymbol{q})^T \boldsymbol{\lambda} = \boldsymbol{f}_{ex}(\boldsymbol{q}) \tag{2.5.16a}$$

$$g_N^i(\boldsymbol{q}) = 0 \qquad \text{for} \qquad i = 1, \ldots, n_{as} \tag{2.5.16b}$$

where n_{as} denotes the number of slave nodes in the active set. Substitution of Eq. 2.5.12 into the above equation yields the following expression for the constraint terms for the slave and master body, respectively:

[9] The dynamic case will be considered in Section 2.6.

$$\boldsymbol{G}^{s}(\boldsymbol{q})^{T}\boldsymbol{\lambda} = \sum_{i=1}^{n_{as}} \left\{ \begin{bmatrix} \boldsymbol{I}_3 \\ \overline{\boldsymbol{G}}^{s^T} \tilde{\overline{\boldsymbol{r}}}^{si} \boldsymbol{A}^{s} \end{bmatrix} \lambda^{j} \overline{\overline{\boldsymbol{n}}}^{m,i} \right\} \tag{2.5.17a}$$

$$\boldsymbol{G}^{m}(\boldsymbol{q})^{T}\boldsymbol{\lambda} = \sum_{i=1}^{n_{as}} \left\{ \frac{1}{4} \sum_{l=1}^{4} N^{l,i}\left(\bar{\bar{\xi}}, \bar{\bar{\eta}}\right) \begin{bmatrix} \boldsymbol{I}_3 \\ \overline{\boldsymbol{G}}^{m^T} \tilde{\overline{\boldsymbol{r}}}^{ml} \boldsymbol{A}^{m} \end{bmatrix} \lambda^{i}\left(-\overline{\overline{\boldsymbol{n}}}^{m,i}\right) \right\} \tag{2.5.17b}$$

where $\boldsymbol{G}^{b}(\boldsymbol{q}) = \partial \boldsymbol{g}(\boldsymbol{q})/\partial \boldsymbol{q}^{b}$. Comparing the above equations to the equation of the generalized external force (Eq. 2.4.21), one can readily appreciate that λ^{i} corresponds to the magnitude of the unknown normal contact force corresponding to slave node S^{i}, i.e. $\boldsymbol{f}_{c}^{i} = \lambda^{i}\overline{\overline{\boldsymbol{n}}}^{m,i}$. Eq. 2.5.16 is solved using an iterative procedure which involves solving the following system of linearized equations:

$$\begin{bmatrix} \left(\boldsymbol{K} + \boldsymbol{g}_{\boldsymbol{qq}}^{T}\boldsymbol{\lambda} - \boldsymbol{J}_{ex}\right) & \boldsymbol{G}(\boldsymbol{q})^{T} \\ \boldsymbol{G}(\boldsymbol{q}) & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \Delta \boldsymbol{q} \\ \Delta \boldsymbol{\lambda} \end{Bmatrix} = -\begin{Bmatrix} \boldsymbol{K}\boldsymbol{q} + \boldsymbol{G}(\boldsymbol{q})^{T}\boldsymbol{\lambda} - \boldsymbol{f}_{ex}(\boldsymbol{q}) \\ \boldsymbol{g}(\boldsymbol{q}) \end{Bmatrix} = -\begin{Bmatrix} \boldsymbol{r}(\boldsymbol{q}, \boldsymbol{\lambda}) \\ \boldsymbol{g}(\boldsymbol{q}) \end{Bmatrix} \tag{2.5.18}$$

where $\boldsymbol{g}_{\boldsymbol{qq}} = \partial^{2}\boldsymbol{g}(\boldsymbol{q})/\partial(\boldsymbol{q})^{2}$, $\boldsymbol{J}_{ex} = \partial \boldsymbol{f}_{ex}(\boldsymbol{q})/\partial \boldsymbol{q}$ is the Jacobian matrix of the generalized external forces, and $\boldsymbol{r}(\boldsymbol{q}, \boldsymbol{\lambda})$ denotes the residual of the static equilibrium equation.

2.5.3.2 Penalty methods

The Lagrange multiplier method is characterized by the exact fulfilment of the impenetrability constraint $g_{N}^{i}(\boldsymbol{q}) = 0$. To this end, n_{as} additional unknowns (the Lagrange multipliers) are introduced in the EOMs and the contact constraints are enforced explicitly (see Eq. 2.5.18). Penalty methods instead eliminate the contact constraints from the EOMs by sacrificing exact fulfilment of the impenetrability constraint. More specifically, the magnitude of the normal contact forces is computed directly from the gap function (Eq. 2.5.11) as follows:

$$\boldsymbol{f}_{c}^{i} = \epsilon_{N}\langle g_{N}^{i}\rangle \overline{\overline{\boldsymbol{n}}}^{m,i} \tag{2.5.19}$$

where ϵ_{N} is the penalty factor and $\langle\cdot\rangle$ is the Macauley bracket, defined as

$$\langle g_{N}^{i}\rangle \coloneqq \begin{cases} g_{N}^{i} \; if \; g_{N}^{i} \leq 0 \\ 0 \; if \; g_{N}^{i} > 0 \end{cases} \tag{2.5.20}$$

While the solution of the Lagrange multiplier method is exactly recovered when ϵ_{N} tends to infinity, large penalty factors will lead to an ill-conditioned numerical problem, which is why ϵ_{N} is always given a finite value. As a rule of thumb, ϵ_{N} should be selected at least 2-3 orders of magnitude larger than the physical stiffness of the contact interface [121]. Using the definition of Eq. 2.5.19, the static equilibrium equation (Eq. 2.5.16) reduces to:

$$\boldsymbol{K}\boldsymbol{q} = \boldsymbol{f}_{ex}(\boldsymbol{q}) + \boldsymbol{f}_c(\boldsymbol{q}) \tag{2.5.21}$$

where for clarity the vector of generalized contact forces $\boldsymbol{f}_c(\boldsymbol{q})$ was separated from the generalized external force vector $\boldsymbol{f}_{ex}(\boldsymbol{q}, t)$. The former can be partitioned as $\boldsymbol{f}_c^T = \{\boldsymbol{f}_c^{sT} \quad \boldsymbol{f}_c^{mT}\}$, where $\boldsymbol{f}_c^s$ and $\boldsymbol{f}_c^m$ are obtained by substitution of Eq. 2.5.19 in Eq. 2.4.21:

$$\boldsymbol{f}_c^s = \sum_{i=1}^{n_{as}} \boldsymbol{f}_c^{si} = \sum_{i=1}^{n_{as}} \left\{ \begin{bmatrix} \boldsymbol{I}_3 \\ \overline{\boldsymbol{G}}^{sT} \tilde{\overline{\boldsymbol{r}}}^{si} \boldsymbol{A}^s \end{bmatrix} \epsilon_N \langle g_N^i \rangle \overline{\overline{\boldsymbol{n}}}^{m,i} \right\} \tag{2.5.22a}$$

$$\begin{aligned} \boldsymbol{f}_c^m &= \sum_{i=1}^{n_{as}} \boldsymbol{f}_c^{m,i} = \sum_{i=1}^{n_{as}} \left\{ \frac{1}{4} \sum_{l=1}^{4} N^{l,i}(\bar{\bar{\xi}}, \bar{\bar{\eta}}) \boldsymbol{f}_c^{ml,i} \right\} \\ &= \sum_{i=1}^{n_{as}} \left\{ \frac{1}{4} \sum_{l=1}^{4} N^{l,i}(\bar{\bar{\xi}}, \bar{\bar{\eta}}) \begin{bmatrix} \boldsymbol{I}_3 \\ \overline{\boldsymbol{G}}^{mT} \tilde{\overline{\boldsymbol{r}}}^{ml} \boldsymbol{A}^m \end{bmatrix} \epsilon_N \langle g_N^i \rangle \left(-\overline{\overline{\boldsymbol{n}}}^{m,i} \right) \right\} \end{aligned} \tag{2.5.22b}$$

Note, by comparison with Eq. 2.5.17, that the penalty method eliminates the contact constraints in Eq. 2.5.16 by setting the magnitude of the contact force λ^i equal to $\lambda^i = \epsilon_N \langle g_N^i \rangle$. The linearized form of Eq. 2.5.21 is obtained as:

$$(\boldsymbol{K} - \boldsymbol{J}_c - \boldsymbol{J}_{ex}) \Delta \boldsymbol{q} = -\boldsymbol{K}\boldsymbol{q} + \boldsymbol{f}_{ex}(\boldsymbol{q}) + \boldsymbol{f}_c(\boldsymbol{q}) = -\boldsymbol{r}(\boldsymbol{q}) \tag{2.5.23}$$

where $\boldsymbol{J}_c = \partial \boldsymbol{f}_c(\boldsymbol{q}) / \partial \boldsymbol{q}$ is the Jacobian matrix of the generalized contact forces. Using the partitioning of $\boldsymbol{f}_c$ introduced in Eq. 2.5.22, the contact Jacobian $\boldsymbol{J}_c$ can be written as

$$\boldsymbol{J}_c = \frac{\partial \boldsymbol{f}_c(\boldsymbol{q})}{\partial \boldsymbol{q}} = \begin{bmatrix} \dfrac{\partial \boldsymbol{f}_c^s(\boldsymbol{q})}{\partial \boldsymbol{q}^s} & \dfrac{\partial \boldsymbol{f}_c^s(\boldsymbol{q})}{\partial \boldsymbol{q}^m} \\ \dfrac{\partial \boldsymbol{f}_c^m(\boldsymbol{q})}{\partial \boldsymbol{q}^s} & \dfrac{\partial \boldsymbol{f}_c^m(\boldsymbol{q})}{\partial \boldsymbol{q}^m} \end{bmatrix} = \sum_{i=1}^{n_{as}} \begin{bmatrix} \dfrac{\partial \boldsymbol{f}_c^{si}(\boldsymbol{q})}{\partial \boldsymbol{q}^s} & \dfrac{\partial \boldsymbol{f}_c^{si}(\boldsymbol{q})}{\partial \boldsymbol{q}^m} \\ \dfrac{\partial \boldsymbol{f}_c^{m,i}(\boldsymbol{q})}{\partial \boldsymbol{q}^s} & \dfrac{\partial \boldsymbol{f}_c^{m,i}(\boldsymbol{q})}{\partial \boldsymbol{q}^m} \end{bmatrix} \tag{2.5.24}$$

Semi-analytical expressions for the partial derivatives in the above equation can be found in Appendix A.

2.5.3.3 Augmented Lagrange multiplier method

The augmented Lagrangian method can be described as a compromise between Lagrange multiplier and penalty methods, in that it enables exact fulfilment of the impenetrability constraint while using penalty terms to facilitate the convergence of the solution. Using this method, the physical contact force is defined as:

$$\boldsymbol{f}_c^i = \boldsymbol{f}_c^i(\boldsymbol{q}, \lambda^i) = \left(\lambda^i + \epsilon_N \langle g_N^i \rangle\right) \overline{\overline{\boldsymbol{n}}}^{m,i} \tag{2.5.25}$$

such that the static contact equilibrium (Eq. 2.5.21) can be written as

$$\boldsymbol{K}\boldsymbol{q} = \boldsymbol{f}_{ex}(\boldsymbol{q}) + \boldsymbol{f}_c(\boldsymbol{q}, \boldsymbol{\lambda}) \tag{2.5.26}$$

The augmented Lagrange multiplier method forces the gap function g_N^i (and therefore also the penalty term $\epsilon_N \langle g_N^i \rangle$) to zero using an iterative procedure that involves two nested loops. In the inner loop, Eq. 2.5.26 is solved for $\boldsymbol{q}$ assuming a constant value λ_k^i for all Lagrange multipliers, where k denotes the iteration index of the outer loop. The solution of Eq. 2.5.26 involves solving the following system of equations:

$$(\boldsymbol{K} - \boldsymbol{J}_c - \boldsymbol{J}_{ex})\Delta\boldsymbol{q} = -\boldsymbol{K}\boldsymbol{q} + \boldsymbol{f}_{ex}(\boldsymbol{q}) + \boldsymbol{f}_c(\boldsymbol{q}, \boldsymbol{\lambda}_k) = -\boldsymbol{r}(\boldsymbol{q}, \boldsymbol{\lambda}_k) \tag{2.5.27}$$

where $\boldsymbol{\lambda}_k$ is constant. In the outer loop, the value of λ_k^i is updated using the following scheme:

$$\lambda_{k+1}^i = \lambda_k^i + \epsilon_N \langle g_N^i(\boldsymbol{q}_{k+1}) \rangle \tag{2.5.28}$$

and the procedure is repeated until g_N^i is sufficiently small. One can observe that λ_k^i approaches the correct value of the Lagrange multiplier when g_N^i tends to zero. As the converged solution of Eq. 2.5.26 doesn't depend on the value of the penalty factor ϵ_N, it is standard practice in augmented Lagrange iterations to update ϵ_N in order to obtain good convergence, e.g. using the following scheme [10]:

$$\epsilon_{N,k+1} = \begin{cases} 10 \cdot \epsilon_{N,k} & if \quad g_N^i(\boldsymbol{q}_{k+1}) > 0.25 \cdot g_N^i(\boldsymbol{q}_k) \\ \epsilon_{N,k} & if \quad g_N^i(\boldsymbol{q}_{k+1}) \leq 0.25 \cdot g_N^i(\boldsymbol{q}_k) \end{cases} \tag{2.5.29}$$

The procedure described above is due to Uzawa [111].

2.6 Time integration of the equations of motion

In the previous sections, the equations describing the dynamic behavior of elastic bodies undergoing mechanical contact interactions have been derived. The general motion of these bodies is characterized by large nonlinear rigid body motion and superposed linear-elastic deformation. The topic of this final section is the numerical solution of these dynamic equations, which allows one to study the evolution of the system in time given certain initial conditions. In their most general form, the dynamic equations can be written as

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q})^T\boldsymbol{\lambda} = \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \tag{2.6.1a}$$

$$\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0} \tag{2.6.1b}$$

where $\boldsymbol{q} \in \mathbb{R}^{n_q}$ is the vector of generalized system coordinates, $\boldsymbol{M}$ is the configuration-dependent mass matrix, $\boldsymbol{g}(\boldsymbol{q})$ collects all the algebraic constraint equations, including but not limited to contact constraints, $\boldsymbol{G} = \partial \boldsymbol{g}/\partial \boldsymbol{q}$ is the constraint Jacobian, $\boldsymbol{f}$ is the generalized force vector, and $\boldsymbol{\lambda} \in \mathbb{R}^{n_\lambda}$ is the vector of Lagrange multipliers, representing contact and constraint forces. The generalized force vector $\boldsymbol{f}$ is composed of the internal forces $\boldsymbol{f}_{int} = \boldsymbol{K}\boldsymbol{q} + \boldsymbol{D}\dot{\boldsymbol{q}}$, the generalized external forces $\boldsymbol{f}_{ex} = \boldsymbol{f}_{ex}(\boldsymbol{q}, t)$ and the quadratic velocity vector $\boldsymbol{f}_v = \boldsymbol{f}_v(\boldsymbol{q}, \dot{\boldsymbol{q}})$.

Eq. 2.6.1 represents a system of $n_q + n_\lambda$ second-order DAEs in $n_q + n_\lambda$ unknowns. Solving these equations requires discretizing the time interval of interest $\mathcal{I}$ in N increments (or steps) $h_n = t_{n+1} - t_n$ and finding the values of $\boldsymbol{q}$ and $\boldsymbol{\lambda}$ at the discretized times $t_1, \dots, t_N$. Because of the nonlinear constraint equation $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$, implicit (iterative) methods are generally used to solve Eq. 2.6.1. These methods compute the state of the system at a later time by iteratively solving a system of equations involving the current and later state of the system. Implicit methods are discussed in detail in Sections 2.6.2-2.6.3.

In Appendix B, it is shown how generalized coordinate partitioning and penalty methods can be used to eliminate the constraint equations (and the associated Lagrange multipliers) from the EOMs, yielding the following system of equations (expressed in terms of the vector of independent generalized coordinates $\boldsymbol{q}_i$):

$$\boldsymbol{M}(\boldsymbol{q}_i)\ddot{\boldsymbol{q}}_i = \boldsymbol{f}(\boldsymbol{q}_i, \dot{\boldsymbol{q}}_i, t) \tag{2.6.2}$$

In this equation $\boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ now additionally includes the vector of generalized contact forces $\boldsymbol{f}_c = \boldsymbol{f}_c(\boldsymbol{q}, \dot{\boldsymbol{q}})$ and possibly some penalty-regulated constraint forces. Note that the mass matrices and generalized force vectors of Eq. 2.6.1 and Eq. 2.6.2 are not necessarily the same. Eq. 2.6.1 represents a system of n_q semi-discrete second-order ODEs in n_q unknowns that can be solved using either explicit or implicit methods. Explicit methods compute the state of the system at a later time directly from the system's current state, and are discussed in Section 2.6.1. For notational simplicity the subscript *'i'* in $\boldsymbol{q}_i$ will be omitted in this section, keeping in mind however that the vector of *independent* generalized coordinates is referred to.

Compared to explicit methods, implicit methods have a higher cost per time increment owing to the required iterative solution of a system of nonlinear equations at each time step. Due to their iterative nature, implicit methods generally have large regions of absolute stability[10], allowing them in principle to use larger time increments h than explicit methods. Explicit methods

[10] The absolute stability of a numerical integrator is generally expressed with respect to a given *test equation*. For example, in the numerical analysis of ODEs, an often used test equation is the Dahlquist test equation $dy/dt = \lambda y$, with $\lambda < 0$, in which case the absolute stability requirement is expressed as $\|y(t_{n+1})\| < \|y(t_n)\|$.

having a low cost per time increment are therefore generally preferred for dynamic contact problems with many DOFs, for which Newton iterations are expensive and convergence issues are frequent. Implicit methods, on the other hand, become effective candidates for the solution of dynamic contact problems when the dimensions and complexity of these problems are reduced, as will be demonstrated in the next chapters. Therefore, both explicit and implicit algorithms are discussed in the following sections, with a focus on computer implementation and error control. For an in-depth investigation into the various solution methods for ODEs and DAEs, the interested reader is referred to the excellent monographs of [122, 123, 124].
In order to limit the scope of the upcoming overview, a few a priori requirements are imposed on the candidate integration schemes. These are:

- The integrator should have an order of convergence of at least two;
- The integrator should have a large stability region, or at least a computable stability limit;
- The integrator should allow for an efficient estimate of the local truncation error;
- If the integrator is implicit, the number of nonlinear equations to be solved at each time step should not be larger than $(n_q + n_\lambda)$;
- All user-defined parameters of the integrator should relate to physical intuition[11].

These requirements are somewhat less stringent than the recommendations given by e.g. Bathe [125] and Hughes [75, pp. 535-537] in the context of nonlinear structural dynamics. The reason is that the small time increments naturally imposed by dynamic contact problems relax some of these requirements, including the unconditional stability when applied to linear problems [75], and the ability to solve the complete system of dynamic equilibrium equations at the time point of interest [125].

2.6.1 Explicit methods for unconstrained systems

This section discusses the numerical solution of Eq. 2.6.2, which can be written in residual form as

$$\boldsymbol{r}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, t) \coloneqq \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} - \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = \boldsymbol{0} \in \mathbb{R}^{n_q} \tag{2.6.3}$$

Introducing the dependent variable $\boldsymbol{y}(t) \in \mathbb{R}^{2n_q} = \mathbb{R}^{n_y}$ via the identity

$$\boldsymbol{y}(t) \coloneqq \begin{Bmatrix} \boldsymbol{q}(t) \\ \dot{\boldsymbol{q}}(t) \end{Bmatrix} \tag{2.6.4}$$

[11] Examples of such parameters include tolerances on displacements and the amount of numerical dissipation at infinity.

and defining $\dot{\boldsymbol{y}}(t) = d\boldsymbol{y}(t)/dt$, Eq. 2.6.3 can be written as:

$$\dot{\boldsymbol{y}}(t) = \begin{Bmatrix} \dot{\boldsymbol{q}}(t) \\ \ddot{\boldsymbol{q}}(t) \end{Bmatrix} = \begin{Bmatrix} \dot{\boldsymbol{q}}(t) \\ \boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \end{Bmatrix} \coloneqq f(\boldsymbol{y}, t) \tag{2.6.5}$$

which represents a system of $2n_q$ first-order ODEs. Methods that discretize and solve Eq. 2.6.3 are referred to as *direct* methods. They stem from the field of structural dynamics and were originally developed for the analysis of high-dimensional *linear* finite element models. *First-order* methods instead are based on the temporal discretization of Eq. 2.6.5 and were primarily explored in the field of mathematics, imposing no assumptions on the linearity of $f(\boldsymbol{y}, t)$. These methods are generally classified as either single-step or multistep methods, depending on whether the state of the system at a later time is computed from the current state only, or if also a number of previous states are taken into account. First-order methods are discussed in Sections 2.6.1.1-2.6.1.2 and direct methods in Section 2.6.1.3.

In the remainder of this chapter, the following notational convention is used: t_n represents the current time and $t_{n+1} = t_n + h_n$ is the time at the next time step, where h_n is the time increment at t_n. The vector of generalized coordinates $\boldsymbol{q}$ (or the state vector $\boldsymbol{y}$) at time t_n is denoted as $\boldsymbol{q}_n = \boldsymbol{q}(t_n)$ (or $\boldsymbol{y}_n = \boldsymbol{y}(t_n)$), and the right-hand side of Eq. 2.6.5 at time t_n is written as $f_n = f(\boldsymbol{y}_n, t_n)$.

2.6.1.1 Explicit single-step methods

The Runge-Kutta formulas Developed around the 1900s by the German mathematicians C. Runge and M. W. Kutta [126], the Explicit Runge-Kutta (ERK) methods are the most popular explicit single-step integration methods. They are based on the following integral form of Eq. 2.6.5:

$$\boldsymbol{y}(t_{n+1}) - \boldsymbol{y}(t_n) = \int_{t_n}^{t_{n+1}} \dot{\boldsymbol{y}}(t)dt = \int_{t_n}^{t_{n+1}} f(\boldsymbol{y}, t)dt \tag{2.6.6}$$

where the integral is approximated using a numerical quadrature rule. Such rules are generally derived by replacing the integrand by a polynomial that interpolates $f(\boldsymbol{y}, t)$ at certain values within the half-open interval $[t_n, t_{n+1}[$, and integrating the latter exactly:

$$\int_{t_n}^{t_{n+1}} f(\boldsymbol{y}, t)dt \approx \int_{t_n}^{t_{n+1}} \sum_{i=1}^{s} f_i L_i(t)\, dt = \sum_{i=1}^{s} f_i w_i \qquad t_i \in [t_n, t_{n+1}[\tag{2.6.7}$$

where $L_i(t)$ represents the *i*-th interpolating Lagrange polynomial, and $w_i = \int L_i(t)dt$ is the corresponding exact integral. The function evaluations $f_i = f(\boldsymbol{y}_i, t_i)$, where $i = 1 \dots s$, are often referred to as the *stages* of the Runge-Kutta method, with *s* denoting the number of stages. The general s-stage ERK scheme (often denoted as ERKs) can be expressed as

$$\boldsymbol{Y}_1 = f(\boldsymbol{y}_n, t_n) \tag{2.6.8a}$$

$$\boldsymbol{Y}_i = f\left(\boldsymbol{y}_n + h\sum_{j=1}^{i-1} a_{ij}\boldsymbol{Y}_j\,, t_n + c_i h\right) \quad i = 2 \dots s \tag{2.6.8b}$$

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h\sum_{i=1}^{s} b_i \boldsymbol{Y}_i \tag{2.6.8c}$$

where the coefficients are determined such that the highest possible order of convergence for the given number of stages is obtained[12] [122]. Like any other explicit method, the ERK methods are conditionally stable. They preserve linear invariants (e.g. linear momentum) [127], and can be made energy-conserving by projecting the updated state vector onto the manifold of solutions that satisfy the energy invariant [128]. The coefficients of a number of lower-order ERK methods are given in Appendix G.

Error estimation An important aspect of all numerical integration schemes is the availability of a reliable error estimator and an efficient means of controlling the error by adjusting the time increment h. Most adaptive time stepping techniques aim at controlling the local truncation error $\boldsymbol{e}_{n+1} \in \mathbb{R}^{n_y}$, rather than the global error. The former is defined as the error inferred by proceeding from t_n to t_{n+1} assuming $\boldsymbol{y}_n$ was the exact solution, i.e. $\boldsymbol{e}_{n+1} = \bar{\boldsymbol{y}}_{n+1} - \boldsymbol{y}_{n+1}$, with $\bar{\boldsymbol{y}}_n = \boldsymbol{y}_n$, where the bar sign denotes the exact solution. Let the user-defined error tolerance be defined as

$$ETOL = ATOL + max\,(|\boldsymbol{y}_n|, |\boldsymbol{y}_{n+1}|) \cdot RTOL \tag{2.6.9}$$

where *ATOL* and *RTOL* denote, respectively, the absolute and relative error[13]. Then the aim of the increment-size selection strategy might be to select h such that the local errors made at each time step are roughly equal for all time steps, i.e. $|\boldsymbol{e}_{n+1}| \approx ETOL$.

[12] For a numerical integration method to be of order *p*, two conditions have to be fulfilled: first, the method has to be 0-stable, meaning that solutions that start close together remain close over the time interval of interest (for a more quantifiable definition, the interested reader is referred to [122]). Second, the method has to be accurate (or consistent) of order *p*, meaning that the global error ε_n at a time t_n, which measures the difference between the numerical and the exact solution at time t_n, should be of order *p+1*, i.e. $\varepsilon_n = \mathcal{O}(\Delta t^{p+1})$, or $|\varepsilon_n| < C\Delta t^{p+1}$ for some constant C.

[13] If the components of $\boldsymbol{y}$ are very different in magnitude (as is frequently the case when dealing with reduced-order models), an array of tolerances can be considered in which for each component of $\boldsymbol{y}$ an separate absolute and relative error tolerance is defined.

One popular way of estimating the local truncation error is by calculating two approximate solutions $\boldsymbol{y}_{n+1}$ and $\tilde{\boldsymbol{y}}_{n+1}$ at time t_{n+1}, where the former is computed using a *p*-th order integration method (having the global error $\varepsilon_{n+1} = \mathcal{O}(h^{\,p+1})$ and the latter using a method of order $p+1$ (with $\varepsilon_{n+1} = \mathcal{O}(h^{\,p+2})$. Substracting these two solutions yields

$$\tilde{\boldsymbol{y}}_{n+1} - \boldsymbol{y}_{n+1} = \boldsymbol{e}_{n+1} + \mathcal{O}(h^{\,p+2}) \tag{2.6.10}$$

where $\boldsymbol{e}_{n+1} = \mathcal{O}(h^{\,p+1})$ can be considered as a reasonable estimate of the local truncation error of the lower-order method. Since the cost of updating the state vector $\boldsymbol{y}$ is dominated by the evaluation of $f(\boldsymbol{y}, t)$, practical error estimation for ERK methods requires a pair of Runga-Kutta formulas that share as many of these function evaluations as possible. Two of such pairs, which belong to the class of the so-called *embedded* Runge-Kutta methods, are described further on.

Increment-size control The procedure of adapting the increment size using a local error estimator is as follows. If the local error estimator $\boldsymbol{e}_{n+1}$ is larger than the error tolerance, integrating the state vector between t_n and $t_{n+1} = t_n + h_n$ has failed and needs to be redone using a smaller increment size $h_{n,new} < h_n$. As the local truncation error of a method of order *p* scales with h^{p+1}, the new increment size $h_{n,new}$ is selected to satisfy

$$\left(\frac{h_{n,new}}{h_n}\right)^{p+1} |\tilde{\boldsymbol{y}}_{n+1} - \boldsymbol{y}_{n+1}| \approx ETOL \;\longrightarrow\; h_{n,new} = h_n S \left(\frac{ETOL}{|\tilde{\boldsymbol{y}}_{n+1} - \boldsymbol{y}_{n+1}|}\right)^{\frac{1}{p+1}} \tag{2.6.11}$$

where the safety factor *S* accounts for the effects of neglected higher-order terms (usually a value between 0.8 and 0.9 is used). In case component-wise error tolerances $ETOL_i, i = 1 \ldots n_y$ are used, Hairer et al. [123] suggest the following local error measure:

$$err = \sqrt{\frac{1}{n_y} \sum_{i=1}^{n_y} \left[\frac{(\tilde{\boldsymbol{y}}_{n+1} - \boldsymbol{y}_{n+1})_i}{ETOL_i}\right]^2} \tag{2.6.12}$$

in which case Eq. 2.6.11 reduces to $h_{n,new} = h_n S (err)^{\frac{-1}{p+1}}$. When the estimated local error is below the error tolerance (or, equivalently, $err \leq 1$ in Eq. 2.6.12), the current step is accepted. One can then proceed using the current increment size, or increase the step size (using the same formula of Eq. 2.6.11) in case the estimated error is well below the error tolerance. While in principle the higher-order solution $\tilde{\boldsymbol{y}}_{n+1}$ is merely computated with the purpose of estimating the local error, most algorithms based on embedded Runge-Kutta methods use this solution to update the state vector. This technique of proceeding the integration with the higher-order solution is called *local extrapolation*. In Appendix G the coefficients of two lower-order embedded RK methods are given.

2.6.1.2 Explicit multistep methods

The Adams formulas Similar to the Runge-Kutta methods, (linear) multistep methods are based on an approximation of the integrand $f(\boldsymbol{y}, t)$ in Eq. 2.6.6. In the multistep case, however, instead of interpolating the function values at times $t_i \in [t_n, t_{n+1}[$, the interpolant now passes through k function values at times $t_n, t_{n-1}, t_{n-2,\ldots}, t_{n-k+1}$, where k is referred to as the number of *steps* of the method. The advantage of using past information in approximating the integrand is that fewer function evaluations per time step are required to obtain a certain order of convergence. On the other hand, some of the flexibility of single-step methods is lost, which becomes particularly apparent when wanting to change the increment size of a multistep method.

The most commonly used linear multistep methods that are based on the interpolation of past values of $f(\boldsymbol{y}, t)$ are the Adams methods. The general form of a k-step Adams method is given by

$$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h \sum_{i=0}^{k} \beta_i f(\boldsymbol{y}_{n+1-i}, t_{n+1-i}) \tag{2.6.13}$$

where β_i are the method's coefficients. The explicit Adams methods (also known as the Adams-Bashforth methods) have $\beta_0 = 0$, whereas implicit methods (also known as Adams-Moulton methods) have $\beta_0 \neq 0$. Explicit and implicit Adams methods are often combined to form *predictor-corrector* schemes, where an explicit method is used to predict the value of $f(\boldsymbol{y}_{n+1}, t_{n+1})$ which is then used by the implicit method. Such a scheme (often reffered to as a PECE method) can be summarized as:

Predict (P) $$\boldsymbol{y}^p_{n+1} = \boldsymbol{y}_n + h \sum_{i=1}^{k} \beta^p_i f(\boldsymbol{y}_{n+1-i}, t_{n+1-i}) \tag{2.6.14a}$$

Evaluate (E) $$\dot{\boldsymbol{y}}_{n+1} = f\left(\boldsymbol{y}^p_{n+1}, t_{n+1}\right) \tag{2.6.14b}$$

Correct (C) $$\boldsymbol{y}_{n+1} = \boldsymbol{y}_n + h \sum_{i=0}^{k} \beta^c_i f(\boldsymbol{y}_{n+1-i}, t_{n+1-i}) \tag{2.6.14c}$$

Evaluate (E) $$\dot{\boldsymbol{y}}_{n+1} = f(\boldsymbol{y}_{n+1}, t_{n+1}) \tag{2.6.14d}$$

where the superscripts '*p*' and '*c*' refer to the predictor and corrector steps, respectively. The corrector step can be repeated m times to improve the accuracy of the correction, in which case the scheme is symbolized by $\mathrm{P(EC)^mE}$. Although Eq. 2.6.14c is an implicit formula, the overall method is explicit, unless the corrector step is repeated until convergence (in which case Eq. 2.6.14 can be seen as a fixed-point iteration scheme). Adams methods are called *linear* because, unlike e.g. Runge-Kutta methods, the Adams scheme (Eq. 2.6.13) is linear in f. As a consequence, the local truncation error of the method always has the following simple form [122]:

$$\boldsymbol{e}_{n+1} = C_{p+1} h^{p+1} \boldsymbol{y}^{(p+1)}(t_n) + \mathcal{O}(h^{\,p+2}) \tag{2.6.15}$$

where p is the order of convergence of the method, $\boldsymbol{y}^{(p+1)}$ is the *(p+1)*-th derivative of $\boldsymbol{y}$ with respect to time, and C_{p+1} is a computable error constant. The β_i coefficients, orders and error constants of explicit and implicit Adams methods are given in Appendix G. Note here that Adams-Moulton methods require one step less to achieve a certain order of convergence as compared to Adams-Bashforth methods, and have smaller error constants. Given their small stability regions[14] (considerably smaller than e.g. explicit Runge-Kutta methods), the use of Adams-Bashforth methods to solve stiff[15] differential equations is generally discouraged, unless they are combined with an Adams-Moulton method in a PECE scheme. Nevertheless, it should be noted that since Eq. 2.6.14c is generally not iterated to convergence, the order, error, and stability properties of implicit methods not necessarily translate to PECE methods.

Error estimation and increment-size control Among the explicit Adams methods, predictor-corrector schemes are the only valid candidates for solving moderately-stiff ODEs. The error estimate of such schemes can be expressed in terms of the difference between the predictor and corrector solutions. A logical combination of Adams-Bashforth and Adams-Moulton methods would be to pair a *k*-step Bashfort predictor with a *(k-1)*-step Moulton corrector to obtain a PECE method of order $p = k$ requiring two function evaluations per time step. Subtracting the predictor from the corrector, one obtains:

$$\boldsymbol{y}_{n+1} - \boldsymbol{y}^{p}_{n+1} = (C_{p+1} - C^{p}_{p+1}) h^{p+1} \boldsymbol{y}^{(p+1)}(t_n) + \mathcal{O}(h^{\,p+1}) \tag{2.6.16}$$

Then, using Eq. 2.6.15, the following local error estimate for the predictor is found:

$$\boldsymbol{e}_{n+1} = C_{p+1} h^{p+1} \boldsymbol{y}^{(p+1)}(t_n) + \mathcal{O}(h^{\,p+1}) = \frac{C_{p+1}}{C_{p+1} - C^{p}_{p+1}} \left(\boldsymbol{y}_{n+1} - \boldsymbol{y}^{p}_{n+1} \right) \tag{2.6.17}$$

Using the above local error estimate, the new time increment can be determined using a similar reasoning as was used to arrive at Eq. 2.6.11:

$$h_{n,new} = h_{n,old} S \left(\frac{C_{p+1} - C^{p}_{p+1}}{C_{p+1}} \frac{ETOL}{\left| \boldsymbol{y}_{n+1} - \boldsymbol{y}^{p}_{n+1} \right|} \right)^{\frac{1}{p+1}} \tag{2.6.18}$$

[14] For multistep methods the stability regions generally decrease as the order of the method increases.

[15] A differential equation is said to be stiff when a numerical integrator with a finite region of absolute stability is forced to use time increments that are excessively small in relation to the smoothness of the exact solution.

The main difficulty in adapting the increment-size is that in order to proceed, the function values at past times $t_n - jh_{n,new}, j = 1 \dots k-1$, are needed, while what is available are the function values at $t_n - jh_{n,old}$. In the context of explicit multistep methods, two options are available to deal with this difficulty: either an interpolation strategy is adopted, where the missing function values are interpolated based on the available values; or a variable-coefficient strategy is applied, where the coefficients of the Adams methods are recomputed based on the unequally spaced data [129]. While the former approach is simpler with regard to implementation and numerical cost, the variable-coefficient strategy is known to have better stability properties, especially when increment changes are frequent and drastic [122]. Variable-coefficient formulas for Adams-Bashforth-Moulton PECE methods can be found in e.g. [123].

2.6.1.3 Direct explicit integration methods

The Central Difference Formula The central difference method is the most widely used explicit method for direct time integration of the equations of motion in linear and nonlinear structural dynamics. The popularity of the method is mainly due to its second-order accuracy, its computable stability limit, the ease of implementation, and the absence of numerical damping. Direct integration methods such as the central difference method are best analyzed by expressing Eq. 2.6.2 in the familiar expanded form (see Eq. E.5):

$$\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} = \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \tag{2.6.19}$$

where $\boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ collects the generalized external forces (including contact forces) and the quadratic velocity terms. The central difference method is based on the following central difference formula (for constant increment size h):

$$\dot{\boldsymbol{q}}_n = \frac{\boldsymbol{q}_{n+1} - \boldsymbol{q}_{n-1}}{2h} \tag{2.6.20a}$$

$$\ddot{\boldsymbol{q}}_n = \frac{\boldsymbol{q}_{n+1} - 2\boldsymbol{q}_n + \boldsymbol{q}_{n-1}}{h^2} \tag{2.6.20b}$$

Substitution of Eq. 2.6.20 in Eq. 2.6.19 at time t_n yields the following integration scheme:

$$\begin{aligned}\left(\frac{1}{h^2}\boldsymbol{M}(\boldsymbol{q}_n) + \frac{1}{2h}\boldsymbol{D}\right)\boldsymbol{q}_{n+1} \\ = \boldsymbol{f}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n) - \left(\boldsymbol{K} - \frac{2}{h^2}\boldsymbol{M}(\boldsymbol{q}_n)\right)\boldsymbol{q}_n - \left(\frac{1}{h^2}\boldsymbol{M}(\boldsymbol{q}_n) + \frac{1}{2h}\boldsymbol{D}\right)\boldsymbol{q}_{n-1}\end{aligned} \tag{2.6.21}$$

Note that this equations requires a value for the vector of generalized velocities $\dot{\boldsymbol{q}}_n$, which is not available at time t_n (see Eq. 2.6.20). Therefore, the above equation can only be explicitly solved for $\boldsymbol{q}_{n+1}$ when $\boldsymbol{f}$ is velocity-independent, i.e. $\boldsymbol{f} = \boldsymbol{f}(\boldsymbol{q}, t)$. This condition holds e.g. when the

friction forces and quadratic velocity terms can be neglected. In the more general case, a common procedure is to have the generalized velocities lagging half a time increment behind [130]:

$$\dot{\boldsymbol{q}}_{n-1/2} = \frac{\boldsymbol{q}_n - \boldsymbol{q}_{n-1}}{h} \tag{2.6.22}$$

which upon substitution in Eq. 2.6.21 yields the following fully-explicit update formula:

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h\dot{\boldsymbol{q}}_{n-1/2} + h^2\boldsymbol{M}(\boldsymbol{q}_n)^{-1}\left[\boldsymbol{f}\left(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_{n-1/2}, t_n\right) - \boldsymbol{D}\dot{\boldsymbol{q}}_{n-1/2} - \boldsymbol{K}\boldsymbol{q}_n\right] \tag{2.6.23}$$

Although this formula can only guarantee first-order accuracy whereas Eq. 2.6.21 ensures second-order accuracy, the two formulas have almost the same accuracy in the case of small time steps [131]. Starting the method of Eq. 2.6.21 requires $\boldsymbol{q}_{-1}$, which can be computed from the known initial conditions $\boldsymbol{q}_0$ and $\dot{\boldsymbol{q}}_0$:

$$\boldsymbol{q}_{-1} = \boldsymbol{q}_0 - h\dot{\boldsymbol{q}}_0 + \frac{h^2}{2}\ddot{\boldsymbol{q}}_0 \tag{2.6.24}$$

where $\ddot{\boldsymbol{q}}_0$ is obtained from Eq. 2.6.19. Starting the method of Eq. 2.6.23 on the other hand requires $\dot{\boldsymbol{q}}_{-1/2}$, which is obtained from $\dot{\boldsymbol{q}}_{-1/2} = \dot{\boldsymbol{q}}_0 - (h/2)\ddot{\boldsymbol{q}}_0$. The central difference method is conditionally stable and requires the time increment to satisfy [75]:

$$h \leq \frac{2}{\omega_{max}}\sqrt{1+\zeta^2} - \zeta = \frac{T_{min}}{\pi}\sqrt{1+\zeta^2} - \zeta \tag{2.6.25}$$

where ω_{max} is the highest undamped natural frequency in the model and T_{min} the associated period, and ζ is the associated fraction of critical damping. Note that this stability criterion corresponds to requiring that at least π time increments of size h are used to resolve the smallest characteristic period T_{min} of the system.

Error estimation and increment-size control The increment size of central difference methods is generally controlled either by limiting the local truncation error, or by monitoring ω_{max} and adjusting h according to Eq. 2.6.25. The latter is motivated by the observation that for moderate to large finite element models excellent accuracy is obtained with increment sizes just under the stability limit [131, 73]. Given the considerable computational effort that is involved in estimating the current maximal frequency in the response of the system, especially in a reduced-order setting, this dissertation uses the local truncation error to control the increment size. An estimate of the local truncation error is easily obtained by forming the Taylor expansion of the exact solution $\boldsymbol{q}_{n+1}^{ex}$ and subtracting the finite difference approximation:

$$\boldsymbol{e}_{n+1} = \boldsymbol{q}_{n+1}^{ex} - \boldsymbol{q}_{n+1} = \frac{h^3}{12}\dddot{\boldsymbol{q}}_{n+1} + \mathcal{O}(h^4) \approx \frac{h^2}{12}(\ddot{\boldsymbol{q}}_{n+1} - \ddot{\boldsymbol{q}}_n) \tag{2.6.26}$$

As $\ddot{\boldsymbol{q}}_{n+1}$ is approximated using a backward single-step finite difference formula, the increment size h should be sufficiently small for the approximation to be reliable. Using the above error estimate and noting that the rate of convergence of the error is $\mathcal{O}(h^3)$, the new increment size is predicted as

$$h_{n,new} = h_{n,old} S \left(\frac{ETOL}{\|\boldsymbol{e}_{n+1}\|} \right)^{\frac{1}{3}} \tag{2.6.27}$$

where the Eucledian norm of $\boldsymbol{e}_{n+1}$ was used as an error measure. In case component-wise tolerances are defined, a similar procedure as in Eq. 2.6.12 can be followed.

Being essentially a two-step method (the schemes of Eq. 2.6.21 and Eq. 2.6.23 require the vector of generalized coordinates q at times $\mathrm{t_n}$ and $\mathrm{t_{n-1}}$), the central difference integration formulas can be accommodated to increment-size changes either by interpolation of past solutions, or by the use of variable-coefficient formulas. For the central difference scheme with lagging velocities (Eq. 2.6.23), the following variable-coefficient formula is derived in Appendix H:

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h_n \dot{\boldsymbol{q}}_{n-1/2} + \frac{h_n(h_{n-1} + h_n)}{2} \boldsymbol{M}(\boldsymbol{q}_n)^{-1} \left[\boldsymbol{f}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_{n-1/2}, t_n) - \boldsymbol{D}\dot{\boldsymbol{q}}_{n-1/2} - \boldsymbol{K}\boldsymbol{q}_n \right] \tag{2.6.28}$$

where h_n and h_{n-1} are the increment sizes used at the current and previous time step, respectively.

2.6.2 Implicit methods for unconstrained systems

In this section two implicit methods for solving the unconstrained equations of motion (Eq. 2.6.2) will be discussed. These are the Backward Differentiation Formula (BDF) methods and the Newmark method (and its derivatives)[16]. The former belongs to the class of the linear multistep methods, whereas the latter is a direct integration method specially designed for nonlinear problems in structural dynamics.

[16] The implicit variants of the single-step Runge-Kutta methods [140] are not dealt with in this dissertation: although they can achieve higher orders of accuracy for a given number of stages than their explicit counterparts (up to order $p = 2s$), they require the solution of a system of $n_q \cdot s$ nonlinear equations per time increment, which in the context of computational contact dynamics is not competitive with other implicit methods involving only n_q nonlinear equations.

2.6.2.1 Implicit multistep methods

Backward Differentiation Formula Unlike Adams methods, which were derived by *integrating* the polynomial that interpolates past value of $f(\boldsymbol{y}, t)$ (see Eq. 2.6.5), BDF methods are obtained by *differentiating* the polynomial that interpolates past values of $\boldsymbol{y}$ and setting the derivative at t_{n+1} equal to $f(\boldsymbol{y}_{n+1}, t_{n+1})$. The general k-step BDF scheme can be summarized as

$$h\beta_0 f(\boldsymbol{y}_{n+1}, t_{n+1}) = \boldsymbol{y}_{n+1} + \sum_{i=1}^{k} \alpha_i \boldsymbol{y}_{n+1-i} \tag{2.6.29}$$

where the coefficients α_i and β_0 are given in Appendix G. Note that BDF methods have an order of convergence p equal to the number of steps k. As BDF methods are implicit methods, they have large stability regions, although they are not unconditionally stable: the stability regions decrease as the order goes up, and for $p = k = 7$, they become unstable. By evaluating $f(\boldsymbol{y}, t)$ only at the right end of the current step $[t_n, t_{n+1}]$, BDF methods attain the stiff decay property[17], which is a highly desirable feature when solving very stiff ODEs.

Newton's method BDF methods are usually combined with Newton's method for solving Eq. 2.6.29 at each time step. In doing so, the following iteration scheme is obtained:

$$\boldsymbol{y}_{n+1,j+1} = \boldsymbol{y}_{n+1,j} - \left(\boldsymbol{I}_{n_y} - h\beta_0 \left.\frac{\partial f}{\partial \boldsymbol{y}}\right|_j \right)^{-1} \left[h\beta_0 f_{n+1} - \boldsymbol{y}_{n+1} - \sum_{i=1}^{k} \alpha_i \boldsymbol{y}_{n+1-i} \right]_j \tag{2.6.30}$$

where j is the iteration index. Using Eq. 2.6.5, The Jacobian $\partial f / \partial \boldsymbol{y}$ can be written as:

$$\frac{\partial f}{\partial \boldsymbol{y}} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{I}_{n_q} \\ \dfrac{\partial}{\partial \boldsymbol{q}} \left(\boldsymbol{M}(\boldsymbol{q})^{-1} \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \right) & \boldsymbol{M}(\boldsymbol{q})^{-1} \dfrac{\partial}{\partial \dot{\boldsymbol{q}}} \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \end{bmatrix} \tag{2.6.31}$$

The initial guess $\boldsymbol{y}_{n+1,0}$ is usually obtained by evaluating the interpolant passing through past value of $\boldsymbol{y}$ at t_{n+1}. For most applications, the cost spent in computing the Jacobian and solving the linear system (Eq. 2.6.30) dominates the integration effort. Since normally the Jacobian changes very little with respect to time, considerable savings in computational effort can be obtained by recomputing the Jacobian only when deemed necessary, e.g. when the increment size has changed considerably or when the iteration fails to converge. The rate of convergence ρ can be estimated by computing the relative change of the Newton increments $\Delta \boldsymbol{y}$:

[17] Methods with stiff decay have the ability to skip fine-level (i.e. highly oscillating) solution details and still maintain an accurate description of the solution on a coarser level [122].

$$\rho_{j+1} = \frac{\|\Delta \boldsymbol{y}_{n+1,j+1}\|}{\|\Delta \boldsymbol{y}_{n+1,j}\|} \tag{2.6.32}$$

The Jacobian is recomputed when ρ_{j+1} exceeds a certain value (e.g. $\rho_{j+1} = 0.7$). Other estimates of the rate of convergence are available in literature (see e.g. [122]).

Error estimation and increment-size control As BDF methods are linear multistep methods, their local truncation error can be estimated using Eq. 2.6.18, with the error constants given in Appendix G. The *(p+1)*-th derivative of $\boldsymbol{y}$ is usually computed using divided differences [132], which requires the storage of an additional past solution vector $\boldsymbol{y}_{n-p}$. Once an estimate for the local error is obtained, the new increment size is computed using:

$$h_{n,new} = h_{n,old} S \left(\frac{ETOL}{|\boldsymbol{e}_{n+1}|} \right)^{\frac{1}{p+1}} \tag{2.6.33}$$

Two strategies for accommodating the multistep formulas to varying increment sizes were discussed in Section 2.6.1.2, namely the fixed-coefficient and variable-coefficient strategy. Although the latter provides better stability properties than the fixed-coefficient strategy, it is known to limit the reusability of the Jacobian matrix due to the varying leading-coefficient β_0 (see Eq. 2.6.30). For that reason, a third approach called the fixed leading-coefficient strategy is developed as a compromise between fixed-coefficient and variable-coefficient strategies. This strategy comes with a fixed value for the coefficient β_0 but at the cost of an additional term $\beta_1 f_n$, which vanishes when the increment sizes of the last k steps are equal. The derivation of BDF methods for varying increment sizes is discussed in [122, 133].

Direct integration Although The BDF method was originally developed for the numerical integration of nonlinear first-order equations, it can be implemented in direct, second-order form by applying the general BDF formula to the trivial equations $\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}$ and $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}$. In Appendix I, the direct formulation of the second-order BDF integrator is derived, based on the derivation of the Newmark update scheme in the next section.

2.6.2.2 Direct implicit integration methods

Newmark-β formula The main drawback of the central difference method (see Section 2.6.1.3) is its conditional stability, which in principle requires some qualitative knowledge of the system frequencies such that a stable time increment can be selected (see Eq. 2.6.25). The Newmark-β method represents a family of direct integrators that are formulated in terms of two parameters β

and γ that can be selected so as to obtain an unconditionally stable integrator in the linear case[18] [73]. The method is based on the assumption that the generalized velocities and displacements can be written as (see [73] for an interpretation of these expressions in terms of the Taylor expansions of $\dot{\boldsymbol{q}}$ and $\boldsymbol{q}$):

$$\dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}_n + h[(1-\gamma)\ddot{\boldsymbol{q}}_n + \gamma\ddot{\boldsymbol{q}}_{n+1}] \tag{2.6.34a}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h\dot{\boldsymbol{q}}_n + \frac{h^2}{2}[(1-2\beta)\ddot{\boldsymbol{q}}_n + 2\beta\ddot{\boldsymbol{q}}_{n+1}] \tag{2.6.34b}$$

The Newmark-β method is unstable for $\gamma < 1/2$. When $\gamma = 1/2$, the method is unconditionally stable for $\beta \geq 1/4$. Note that the explicit central difference method is retreived for $\gamma = 1/2$ and $\beta = 0$. By setting $\ddot{\boldsymbol{q}}_{n+1} = \boldsymbol{0}$ in the above equations, the *predictor* velocities and displacements are obtained:

$$\dot{\boldsymbol{q}}^*_{n+1} \coloneqq \dot{\boldsymbol{q}}_n + h(1-\gamma)\ddot{\boldsymbol{q}}_n \tag{2.6.35a}$$

$$\boldsymbol{q}^*_{n+1} \coloneqq \boldsymbol{q}_n + h\dot{\boldsymbol{q}}_n + \left(\frac{1}{2}-\beta\right)h^2\ddot{\boldsymbol{q}}_n \tag{2.6.35b}$$

Substitution of these equations into Eq. 2.6.34 allows the vector of generalized accelerations and velocities to be written in terms of the above explicit predictor values and $\boldsymbol{q}_{n+1}$:

$$\ddot{\boldsymbol{q}}_{n+1} = \frac{1}{\beta h^2}(\boldsymbol{q}_{n+1} - \boldsymbol{q}^*_{n+1}) \tag{2.6.36a}$$

$$\dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}^*_{n+1} + \frac{\gamma}{\beta h}(\boldsymbol{q}_{n+1} - \boldsymbol{q}^*_{n+1}) \tag{2.6.36b}$$

which in turn allows the residual form of the EOMs (Eq. 2.6.3) to be expressed in terms of $\boldsymbol{q}_{n+1}$ only:

$$\boldsymbol{r}(\boldsymbol{q}_{n+1}, (\cdot)^*_{n+1}, t_{n+1}) = \boldsymbol{M}(\boldsymbol{q}_{n+1})\ddot{\boldsymbol{q}}_{n+1} - \boldsymbol{f}(\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1}, t_{n+1}) = \boldsymbol{0} \tag{2.6.37}$$

This equation can be solved using Newton's method by iterating on:

$$\boldsymbol{q}_{n+1,k+1} = \boldsymbol{q}_{n+1,k} - \boldsymbol{S}_k^{-1}\boldsymbol{r}(\boldsymbol{q}_{n+1,k}, t_{n+1}) \tag{2.6.38}$$

[18] Note that the stability of an integration method is generally assessed by applying the method to a linear test equation. In the case of direct integration methods, this test equation typically corresponds to the equations of motion of a single-DOF harmonic oscillator (see e.g. [73]).

where $\boldsymbol{S}_k$ is the iteration matrix of the k-th iteration. It is defined as the total derivative of $\boldsymbol{r}$ w.r.t. $\boldsymbol{q}$, i.e.

$$\boldsymbol{S}_k \coloneqq \left.\frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}}\right|_k = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}} + \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{q}}}\frac{\partial \dot{\boldsymbol{q}}}{\partial \boldsymbol{q}} + \boldsymbol{M}\frac{\partial \ddot{\boldsymbol{q}}}{\partial \boldsymbol{q}} = \boldsymbol{K}_T + \boldsymbol{D}_T\frac{\partial \dot{\boldsymbol{q}}}{\partial \boldsymbol{q}} + \boldsymbol{M}\frac{\partial \ddot{\boldsymbol{q}}}{\partial \boldsymbol{q}} \tag{2.6.39}$$

where $\boldsymbol{K}_T$ and $\boldsymbol{D}_T$ are the tangent stiffness and damping matrices, which can be expressed in the formalism of Eq. 2.6.19 as follows:

$$\boldsymbol{K}_T = \frac{\partial \boldsymbol{r}}{\partial \boldsymbol{q}} = \boldsymbol{K} + \frac{\partial}{\partial \boldsymbol{q}}(\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} - \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)) \tag{2.6.40a}$$

$$\boldsymbol{D}_T = \frac{\partial \boldsymbol{r}}{\partial \dot{\boldsymbol{q}}} = \boldsymbol{D} - \frac{\partial}{\partial \dot{\boldsymbol{q}}}\boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \tag{2.6.40b}$$

Finally, the remaining derivatives in Eq. 2.6.39 can be computed from Eq. 2.6.36:

$$\frac{\partial \ddot{\boldsymbol{q}}}{\partial \boldsymbol{q}} = \frac{1}{\beta h^2}\boldsymbol{I}_{n_q} = \beta' \boldsymbol{I}_{n_q} \qquad \text{and} \qquad \frac{\partial \dot{\boldsymbol{q}}}{\partial \boldsymbol{q}} = \frac{\gamma}{\beta h}\boldsymbol{I}_{n_q} = \gamma' \boldsymbol{I}_{n_q} \tag{2.6.41}$$

so that the iteration matrix is given by

$$\boldsymbol{S}_k = \boldsymbol{K}_T + \boldsymbol{D}_T\beta' + \boldsymbol{M}\gamma' \tag{2.6.42}$$

Once the displacement corrections $\Delta\boldsymbol{q}_k = \boldsymbol{q}_{n+1,k+1} - \boldsymbol{q}_{n+1,k}$ are obtained by solving Eq. 2.6.37, the velocity and acceleration updates are found by substitution of $\Delta\boldsymbol{q}_k$ in Eq. 2.6.36:

$$\Delta\ddot{\boldsymbol{q}}_k = \beta'\Delta\boldsymbol{q}_k \qquad \text{and} \qquad \Delta\dot{\boldsymbol{q}}_k = \gamma'\Delta\boldsymbol{q}_k \tag{2.6.43}$$

The Newmark algorithm for updating $\boldsymbol{q}, \dot{\boldsymbol{q}}$ and $\ddot{\boldsymbol{q}}$ with fixed increment size is depicted in Algorithm 2.3. The algorithm first proposed by Newmark is known as the *constant average acceleration method*, and has the parameters $\beta = 1/4$ and $\gamma = 1/2$. This method is second-order accurate, unconditionally stable, and conserves the total energy of the system. Moreover, as the method belongs to the class of the so-called variational integrators [134], it preserves momentum and the symplectic structure of the discretized system. In-depth stability and accuracy analyses of the family of Newmark integrators can be found in [73].

Algorithm 2.3 Newmark method (fixed increment size)

Input: system matrices and vectors $\boldsymbol{M}(\boldsymbol{q}), \boldsymbol{D}, \boldsymbol{K}, \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t)$ and initial conditions $\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0$

Output: solution vectors $\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t)$ for $t \in [t_0, t_1, \dots, t_N]$

1. Initialize $\ddot{\boldsymbol{q}}_0 = \boldsymbol{M}^{-1}(\boldsymbol{f}(\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0, t_0) - \boldsymbol{K}\boldsymbol{q}_0 - \boldsymbol{D}\dot{\boldsymbol{q}}_0)$
2. **for** $t = t_0, t_1, \dots, t_N$ **do**
3. Set $\ddot{\boldsymbol{q}}_{n+1} = \boldsymbol{0}$
4. Predict $\boldsymbol{q}^*_{n+1}$, $\dot{\boldsymbol{q}}^*_{n+1}$ using Eq. 2.6.35
5. **for** $k = 1 \dots itermax$ **do**
6. Compute residual $\boldsymbol{r}_{n+1}$ using Eq. 2.6.37
7. Check for convergence: $\|\boldsymbol{r}_{n+1}\| \leq TOL \Longrightarrow END\ LOOP$
8. Form iteration matrix $\boldsymbol{S}_k$ using Eq. 2.6.39 (if update is needed)
9. Compute displacement corrections using Eq. 2.6.38
10. Correct velocities and accelerations using Eq. 2.6.43
11. **end**
12. **end**

Generalized-α formula While the absence of numerical dissipation in the Newmark-β method is desirable in many applications, it often becomes necessary to have some form of algorithmic damping to filter out the response of high-frequency numerical artefacts [4]. The HHT-method [135] and the generalized-α method [136] both enable algorithmic numerical damping while maintaining second-order accuracy, and are discussed in Appendix J.

Error estimation and increment-size control Controlling the increment size of direct implicit integrators proceeds in a similar fashion as for explicit integrators (see Section 2.6.1.3). The local truncation error is obtained by subtracting the approximate solution $\boldsymbol{q}_{n+1}$ from the Taylor expansion of the exact solution $\boldsymbol{q}^{ex}_{n+1}$, which in case of the Newmark method yields [73]:

$$\boldsymbol{e}_{n+1} = \boldsymbol{q}^{ex}_{n+1} - \boldsymbol{q}_{n+1} = \left(\beta - \frac{1}{6}\right)\frac{h^3}{12}\dddot{\boldsymbol{q}}_{n+1} + \mathcal{O}(h^4) \approx \left(\beta - \frac{1}{6}\right)\frac{h^2}{12}(\ddot{\boldsymbol{q}}_{n+1} - \ddot{\boldsymbol{q}}_n) \qquad (2.6.44)$$

Note again that because of the finite differences approximation of $\dddot{\boldsymbol{q}}_{n+1}$, this error estimate is only reliable when h is already small. Using the above error estimate, the new increment size may be computed using Eq. 2.6.27. Finally, it should be noted that, unlike the central difference method (Eq. 2.6.21), the Newmark method is essentially a single-step method, and therefore no changes are needed to the Newmark formulas (Eq. 2.6.34) to accommodate these to the varying step size.

2.6.3 Half-explicit and implicit methods for constrained systems

2.6.3.1 Alternative formulations of the constrained equations of motion

When the system is subjected to bilateral or unilateral constraints, the equations of motion are represented by a system of second-order DAEs, see Eq. 2.6.1. These equations can be written in residual form as follows:

$$\boldsymbol{r}(\boldsymbol{q},\dot{\boldsymbol{q}},\ddot{\boldsymbol{q}},\boldsymbol{\lambda},t) \coloneqq \boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}} + \boldsymbol{G}(\boldsymbol{q})^T\boldsymbol{\lambda} - \boldsymbol{f}(\boldsymbol{q},\dot{\boldsymbol{q}},t) = \boldsymbol{0} \tag{2.6.45a}$$

$$\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0} \tag{2.6.45b}$$

In the theory of multibody dynamics, it is common practice to introduce the variable $\boldsymbol{v} = \dot{\boldsymbol{q}}$ and write Eq. 2.6.1 in the following first-order form:

$$\dot{\boldsymbol{q}} = \boldsymbol{v} \tag{2.6.46a}$$

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} = \boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t) - \boldsymbol{G}(\boldsymbol{q})^T\boldsymbol{\lambda} \quad \text{(index-3 form)} \tag{2.6.46b}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{q}) \tag{2.6.46c}$$

These equations, as well as Eq. 2.6.1 and Eq. 2.6.45, represent a system of DAEs of index 3, meaning that 3 differentiations of Eq. 2.6.46 must be performed in order to transform it into a system of ODEs in the unknowns $\boldsymbol{q}, \boldsymbol{v}$ and $\boldsymbol{\lambda}$ [137]. The presence of the implicit constraint equation $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$ suggests that implicit integrators should be used to solve Eq. 2.6.46. The first attempts in applying traditional ODE solvers (such as BDF and Implicit Runge-Kutta (IRK) methods) to the above equations met with very limited success, and lead to the publication in 1982 of Petzold's famous article entitled 'DAEs are not ODEs' [138]. The reason for this lack of success is the high sensitivity of the Lagrange multipliers to small, high-frequency perturbations in Eq. 2.6.46, as was demonstrated by Arnold in [139]. Although methods exist that solve the above equation in index-3 form (one of which will be discussed below), in general it has proven to be beneficial to reduce the index of Eq. 2.6.46 by differentiating the constraint equations (Eq. 2.6.46c) with respect to time, i.e.:

$$\boldsymbol{0} = \frac{d}{dt}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{v} \tag{2.6.47a}$$

$$\boldsymbol{0} = \frac{d^2}{dt^2}\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{v}^T\boldsymbol{G}_{\boldsymbol{q}}(\boldsymbol{q})\boldsymbol{v} = \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{v}} + \boldsymbol{\kappa}(\boldsymbol{q},\boldsymbol{v}) \tag{2.6.47b}$$

In the theory of DAEs, the above equations are often referred to as the *hidden constraints* of the index-3 DAE. Replacing the position-level constraint (Eq. 2.6.46c) with the velocity-level constraint (Eq. 2.6.47a), the index-2 formulation of Eq. 2.6.46 is obtained:

$$\dot{\boldsymbol{q}} = \boldsymbol{v} \tag{2.6.48a}$$

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) - \boldsymbol{G}(\boldsymbol{q})^T \boldsymbol{\lambda} \quad \text{(index-2 form)} \tag{2.6.48b}$$

$$\boldsymbol{0} = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{v} \tag{2.6.48c}$$

This formulation is known to be accurately solvable using the RADAU5 integrator, which is an implicit Runge-Kutta method with fifth-order convergence [140]. The index-1 formulation of Eq. 2.6.46 is obtained by replacing Eq. 2.6.46c with the acceleration-level constraint (Eq. 2.6.47b), and can be expressed as the following linear system:

$$\begin{bmatrix} \boldsymbol{I}_{nq} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}(\boldsymbol{q}) & \boldsymbol{G}(\boldsymbol{q})^T \\ \boldsymbol{0} & \boldsymbol{G}(\boldsymbol{q}) & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \dot{\boldsymbol{q}} \\ \dot{\boldsymbol{v}} \\ \boldsymbol{\lambda} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{v} \\ \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) \\ -\boldsymbol{\kappa}(\boldsymbol{q}, \boldsymbol{v}) \end{Bmatrix} \quad \text{(index-1 form)} \tag{2.6.49}$$

A sufficient (but not necessary) condition for the coefficient matrix to be invertible is that the constraint Jacobian $\boldsymbol{G}(\boldsymbol{q})$ is of full rank and the mass matrix $\boldsymbol{M}(\boldsymbol{q})$ is symmetric positive definite. While in principle one more differentiation is required to obtain the ODE form of Eq. 2.6.46 in terms of $\boldsymbol{q}, \boldsymbol{v}$ and $\boldsymbol{\lambda}$, a system of ODEs in terms of $\boldsymbol{q}$ and $\boldsymbol{v}$ can be obtained by solving Eq. 2.6.49 for $\dot{\boldsymbol{v}}$ and $\boldsymbol{\lambda}$ by block Gaussian elimination:

$$\dot{\boldsymbol{v}} = \boldsymbol{M}(\boldsymbol{q})^{-1}[\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) - \boldsymbol{G}(\boldsymbol{q})^T \boldsymbol{\lambda}] \tag{2.6.50a}$$

$$\boldsymbol{\lambda} = [\boldsymbol{G}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{G}(\boldsymbol{q})^T]^{-1}[\boldsymbol{G}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) + \boldsymbol{\kappa}(\boldsymbol{q}, \boldsymbol{v})] \tag{2.6.50b}$$

and substituting Eq. 2.6.50b in Eq. 2.6.50a. The index-1 formulation of the equations of motion (Eq. 2.6.49) can be solved efficiently and robustly with standard implicit ODE solvers like the BDF and Runge-Kutta methods, whereas the index-0 or ODE form of Eq. 2.6.50 can be solved using any (implicit or explicit) ODE solver.

Although *index reduction* greatly improves the solver's robustness, it is known to introduce considerable *drift-off* errors in long-term simulations (see e.g. [141]). Drift-off results from numerical errors (i.e. discretization and round-off errors) in the velocity or acceleration constraints that are integrated over time and lead to significant non-zero errors in the underlying position-constraints. The drift-off increases linearly with time in the index-2 formulation, and quadratically in the index-1 formulation. An early attempt to avoid drift-off in index-1 formulations is due to Baumgarte [142], who substituted the acceleration constraint in Eq. 2.6.49 by a linear combination of the constraints at position-, velocity-, and acceleration-level. However, as no general procedure is available for selecting the weighting coefficients of this linear combination, the practical use of Baumgarte's method is limited. Most index-2 and index-1 formulations nowadays avoid the drift-off effect by monitoring the residual $\|\boldsymbol{g}(\boldsymbol{q}_{n+1})\|$ and *projecting* the solution $\boldsymbol{q}_{n+1}$ (and possibly $\boldsymbol{v}_{n+1}$) on the constraint manifold(s) defined by $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$ (and $\boldsymbol{G}(\boldsymbol{q})\boldsymbol{v} = \boldsymbol{0}$) whenever the residual exceeds a user-defined error bound [143]. Let $\widetilde{\boldsymbol{q}}_{n+1}$ (and $\widetilde{\boldsymbol{v}}_{n+1}$) denote the positions (and velocities)

obtained by the integration scheme at time t_{n+1}; the minimal-distance projection of $\tilde{\boldsymbol{q}}_{n+1}$ on the position-constraint manifold is found by solving the following quadratic minimization problem:

$$\boldsymbol{q}_{n+1} = \underset{\boldsymbol{q}}{arg\,min}\frac{1}{2}(\boldsymbol{q}_{n+1} - \tilde{\boldsymbol{q}}_{n+1})^T \boldsymbol{A}\,(\boldsymbol{q}_{n+1} - \tilde{\boldsymbol{q}}_{n+1}) \tag{2.6.51a}$$

$$\text{subject to:} \quad \boldsymbol{0} = \boldsymbol{g}(\boldsymbol{q}) \tag{2.6.51b}$$

where $\boldsymbol{A}$ is a symmetric positive definite matrix that defines the metric in which the distance is expressed. Usually the mass matrix $\boldsymbol{M}(\boldsymbol{q}_{n+1})$ is selected to solve Eq. 2.6.51 for reasons of efficiency. A widely used approach for finding the projected displacements $\boldsymbol{q}_{n+1}$ was first proposed by Lubich and involves solving the following system of $(n_q + n_\lambda)$ equations [140]:

$$\boldsymbol{0} = \boldsymbol{M}(\boldsymbol{q}_{n+1})(\boldsymbol{q}_{n+1} - \tilde{\boldsymbol{q}}_{n+1}) + \boldsymbol{G}(\boldsymbol{q}_{n+1})^T\boldsymbol{\mu} \tag{2.6.52a}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{q}_{n+1}) \tag{2.6.52b}$$

where $\boldsymbol{\mu} \in \mathbb{R}^{n_\lambda}$ is the vector of Lagrange multipliers used to enforce the constraint equations of Eq. 2.6.52b. Note that the Jacobian matrix required to solve Eq. 2.6.52 is already available in Eq. 2.6.49, yielding a relatively inexpensive projection step (which is the main reason why usually one selects $\boldsymbol{A} = \boldsymbol{M}(\boldsymbol{q}_{n+1})$). With the value $\boldsymbol{q}_{n+1}$ obtained from Eq. 2.6.52, the projected velocities $\boldsymbol{v}_{n+1}$ are obtained from the projection:

$$\boldsymbol{0} = \boldsymbol{M}(\boldsymbol{q}_{n+1})(\boldsymbol{v}_{n+1} - \tilde{\boldsymbol{v}}_{n+1}) + \boldsymbol{G}(\boldsymbol{q}_{n+1})^T\boldsymbol{\mu} \tag{2.6.53a}$$

$$\boldsymbol{0} = \boldsymbol{G}(\boldsymbol{q}_{n+1})\boldsymbol{v}_{n+1} \tag{2.6.53b}$$

which solves a similar quadratic minimization problem as Eq. 2.6.51. Contrary to Eq. 2.6.52, this equation is linear and can thus be solved immediately.

The explicit projection of displacements (and possibly velocities) at the end of certain time intervals causes considerable difficulties in formulating multistep integration schemes, which is why the method is restricted to Runge-Kutta and other single-step methods. In the Gear-Gupta-Leimkuhler formulation [144] (also referred to as the stabilized index-2 formulation), the projection on the constraint manifold ($\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$) is implicitly included by considering the constraints at position- and velocity-level simultaneously, and introducing additional Lagrange multipliers $\boldsymbol{\mu} \in \mathbb{R}^{n_\lambda}$:

$$\dot{\boldsymbol{q}} = \boldsymbol{v} - \boldsymbol{G}(\boldsymbol{q})^T\boldsymbol{\mu} \tag{2.6.54a}$$

$$\boldsymbol{M}(\boldsymbol{q})\dot{\boldsymbol{v}} = \boldsymbol{f}(\boldsymbol{q}, \boldsymbol{v}, t) - \boldsymbol{G}(\boldsymbol{q})^T\boldsymbol{\lambda} \quad \text{(stabilized index-2 form)} \tag{2.6.54b}$$

$$\boldsymbol{0} = \boldsymbol{G}(\boldsymbol{q})\boldsymbol{v} \tag{2.6.54c}$$

$$\boldsymbol{0} = \boldsymbol{g}(\boldsymbol{q}) \tag{2.6.54d}$$

where $\boldsymbol{\mu}$ approaches zero when the solution converges. Unlike the unstabilized index-2 form (Eq. 2.6.48), the above equation can be solved efficiently using both BDF and implicit Runge-Kutta methods, and is not prone to drift-off errors.

Four different formulations of the equations of motion of constrained mechanical systems have been presented above: the familiar index-3 form with constraints on position-level (Eq. 2.6.46), the unstabilized index-2 form with velocity constraints and linear drift-off (Eq. 2.6.48), the index-1 form and the corresponding ODE with acceleration constraints and quadratic drift-off (Eq. 2.6.49-2.6.50), and finally the stabilized index-2 with constraints on both position- and velocity-level and no drift-off (Eq. 2.6.54). Depending on the formulation, different integrators have been developed since the 1980s to solve these equations. Most of these integrators are implicit, however some semi-explicit methods have been developed that are characterized by an explicit treatment of the dynamic equations and an implicit treatment of the constraint equations.

Of particular interest in the field of computational contact mechanics is the semi-explicit method of Carpenter et al [145], which is based on the central difference integrator (Section 2.6.1.3). In their so-called 'forward increment Lagrange multiplier method', a semi-explicit update formula is obtained by relating the linearized contact constraints at time t_{n+1} to the Lagrange multipliers λ_n at time t_n. The generalized displacements are updated using an explicit update formula while at each time step a system of n_λ coupled nonlinear equations must be solved for λ_n. A more general class of half-explicit methods for the solution of (non-stiff) DAEs was developed by Hairer and Brasey [146]. These methods integrate the dynamic equations using an explicit Runge-Kutta solver while enforcing the constraint equations in an implicit fashion. Given the linearity of the velocity constraint, the approach is particularly successful in solving DAEs of index 2 (although it has also been applied to index-1 [147] and index-3 formulations [148]).

When solving industrial-sized dynamic contact problems, explicit integration schemes are generally preferred over implicit schemes due to the high numbers of DOFs involved. The model reduction techniques developed in this dissertation however bring implicit integrators to the fore, even enabling them to outperform their (semi-)explicit counterparts, especially in the analysis of constrained mechanical systems. For that reason, only implicit methods will be considered in this section. For a more comprehensive treatment of the theory and numerical solution of DAEs, the interested reader is referred to [140] and [137].

2.6.3.2 Fully-implicit methods for DAEs of index 1, 2, and 3

Index-1 DAEs The index-1 formulation of the equations of motion can be transformed into ODE form if the so-called Delassus operator $\boldsymbol{G}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{G}(\boldsymbol{q})^T$ is invertible. In that case, substitution of Eq. 2.6.49b into Eq. 2.6.49a yields the following system of ODEs:

$$\dot{\boldsymbol{q}} = \boldsymbol{v} \tag{2.6.55a}$$

$$\dot{\boldsymbol{v}} = \boldsymbol{M}(\boldsymbol{q})^{-1}\big[\boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t) - \boldsymbol{G}(\boldsymbol{q})^T[\boldsymbol{G}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{G}(\boldsymbol{q})^T]^{-1} * [\boldsymbol{G}(\boldsymbol{q})\boldsymbol{M}(\boldsymbol{q})^{-1}\boldsymbol{f}(\boldsymbol{q},\boldsymbol{v},t) + \boldsymbol{\kappa}(\boldsymbol{q},\boldsymbol{v})]\big] \tag{2.6.55b}$$

These equations can be solved using any ODE solver, although generally BDF[19] (Section 2.6.2.1) or IRK methods are used, combined with position and velocity projections (Eq. 2.6.52), or in some particular cases with Baumgarte stabilization. However, evaluating the acceleration constraints, forming the Delassus operator and stabilizing drift-off effects results in an expensive procedure that is seldom worth the effort as compared to higher-index solvers.

Index-2 DAEs By simultaneously enforcing the position and velocity constraints, the GGL or stabilized index-2 formulation (Eq. 2.6.54) eliminates drift-off effects. Furthermore, there is virtually no extra cost involved in evaluating the velocity constraints, as the constraint Jacobian $\boldsymbol{G}(\boldsymbol{q})$ has to be computed anyway when evaluating the dynamic equations. The GGL formulation requires the introduction of an additional vector of n_λ Lagrange multipliers $\boldsymbol{\mu}$, which vanishes analytically as can easily be shown by deriving the position constraints with respect to time and substituting Eq. 2.6.54a:

$$\begin{aligned}\boldsymbol{0} = \frac{d}{dt}\boldsymbol{g}(\boldsymbol{q}) &= \boldsymbol{G}(\boldsymbol{q})\dot{\boldsymbol{q}} = \boldsymbol{G}(\boldsymbol{q})(\boldsymbol{v} - \boldsymbol{G}(\boldsymbol{q})^T\boldsymbol{\mu}) \\ &= \boldsymbol{G}(\boldsymbol{q})\boldsymbol{v} - \boldsymbol{G}(\boldsymbol{q})\boldsymbol{G}(\boldsymbol{q})^T\boldsymbol{\mu} = -\boldsymbol{G}(\boldsymbol{q})\boldsymbol{G}(\boldsymbol{q})^T\boldsymbol{\mu}\end{aligned} \tag{2.6.56}$$

While the first successful attempts in solving Eq. 2.6.54 relied on BDF [144] formulas, direct integration formulas based on Newmark's method (Section 2.6.2.2) received considerable attention in recent years (see e.g. [149]). Schemes based on modified generalized-α schemes are studied in [150] and [151], and were shown to possess second-order accuracy on the level of displacements and velocities, and first-order accuracy on Lagrange multiplier level. Like the earlier BDF methods [144], the Newmark schemes require a system of $2(n_q + n_\lambda)$ equations to be solved at each time step. As this is in contradiction with the integrator requirements expressed at the beginning of Section 2.6, the stabilized index-2 formulation is not considered further.

Index-3 DAEs The first successful reports on applying the Newmark family of integrators to index-3 DAEs appeared in 1989 [152] (see also [153]) and inspired Arnold and Brüls to conduct a convergence study for the index-3 generalized-α scheme in 2006 [136], see Appendix J. Direct substitution of the generalized-α formulas (Eq. J.3) in the index-3 formulation of the EOMs (Eq. 2.6.45) at time t_{n+1} yields:

[19] The most widely used index-0 to 1 integrator is Petzold's DASSL [137], which is based on a variable stepsize variable order fixed leading coefficient implementation of the BDF formulas.

$$\boldsymbol{M}(\boldsymbol{q}_{n+1})\beta'(\boldsymbol{q}_{n+1}-\boldsymbol{q}^*_{n+1})+\boldsymbol{G}(\boldsymbol{q}_{n+1})^T\boldsymbol{\lambda}_{n+1}\ -\boldsymbol{f}(\boldsymbol{q}_{n+1},\dot{\boldsymbol{q}}_{n+1},t_{n+1})=\boldsymbol{0} \tag{2.6.57a}$$

$$\boldsymbol{g}(\boldsymbol{q}_{n+1})=\boldsymbol{0} \tag{2.6.57b}$$

These equations represent a system of n_q+n_λ nonlinear equations that can be solved using Newton's method:

$$\begin{bmatrix}\boldsymbol{M}\beta'+\boldsymbol{D}_T\gamma'+\boldsymbol{K}_T & \boldsymbol{G}^T\\ \boldsymbol{G} & \boldsymbol{0}\end{bmatrix}_{n+1,k}\begin{Bmatrix}\Delta\boldsymbol{q}_k\\ \Delta\boldsymbol{\lambda}_k\end{Bmatrix}=\begin{Bmatrix}\boldsymbol{r}(\boldsymbol{q}_{n+1},\boldsymbol{\lambda}_{n+1})\\ \boldsymbol{g}(\boldsymbol{q}_{n+1})\end{Bmatrix}_k \tag{2.6.58}$$

where k denotes the iteration index, $\Delta\boldsymbol{q}_k$ and $\Delta\boldsymbol{\lambda}_k$ are the Newton increments for the displacements and the Lagrange multipliers, $\boldsymbol{D}_T$ and $\boldsymbol{K}_T$ are the tangent damping and stiffness matrices as defined by Eq. 2.6.40, and the coefficients β' and γ' are defined in Eq. J.7. Furthermore, $\boldsymbol{r}(\boldsymbol{q}_{n+1},\boldsymbol{\lambda}_{n+1})$ is the residue of Eq. 2.6.57a. In the case of small time increments it is advisable to scale the coefficient matrix in the above equations, see e.g. [154]. Once Eq. 2.6.58 is solved, the solutions are updated as follows:

$$\boldsymbol{q}_{n+1,k}=\boldsymbol{q}_{n+1,k-1}+\Delta\boldsymbol{q}_k \tag{2.6.59a}$$

$$\dot{\boldsymbol{q}}_{n+1,k}=\dot{\boldsymbol{q}}_{n+1,k-1}+\gamma'\Delta\boldsymbol{q}_k \tag{2.6.59b}$$

$$\ddot{\boldsymbol{q}}_{n+1,k}=\ddot{\boldsymbol{q}}_{n+1,k-1}+\beta'\Delta\boldsymbol{q}_k \tag{2.6.59c}$$

$$\boldsymbol{\lambda}_{n+1,k}=\boldsymbol{q}_{n+1,k-1}+\Delta\boldsymbol{\lambda}_k \tag{2.6.59d}$$

which is repeated until convergence. The generalized-α procedure is outlined in Algorithm 2.4.

In the convergence analysis of Arnold and Brüls [136] it is demonstrated that the index-3 generalized alpha scheme has second-order convergence for both the displacements $\boldsymbol{q}$ and Lagrange multipliers $\boldsymbol{\lambda}$. While the position constraint can be satisfied to machine precision, there is some drift in the hidden velocity constraints (Eq. 2.6.47a) (see e.g. [155]). Note that the generalized-α method can also be applied to the (unstabalized) index-2 formulation (Eq. 2.6.48) to yield, at each time step, a system of n_q+n_λ nonlinear equations. The time-discretized version of the velocity constraint (Eq. 2.6.47a) at time t_{n+1} is given by:

$$\boldsymbol{G}(\boldsymbol{q}_{n+1})\dot{\boldsymbol{q}}_{n+1}=\boldsymbol{G}(\boldsymbol{q}_{n+1})\left(\dot{\boldsymbol{q}}^*_{n+1}+\gamma'(\boldsymbol{q}_{n+1}-\boldsymbol{q}^*_{n+1})\right)=\boldsymbol{0} \tag{2.6.60}$$

The system of equations formed by combining Eq. 2.6.57a with the above equation is now solved by:

$$\begin{bmatrix}\boldsymbol{M}\beta'+\boldsymbol{D}_T\gamma'+\boldsymbol{K}_T & \boldsymbol{G}^T\\ \boldsymbol{G}_q\dot{\boldsymbol{q}}+\gamma'\boldsymbol{G} & \boldsymbol{0}\end{bmatrix}_{n+1,k}\begin{Bmatrix}\Delta\boldsymbol{q}_k\\ \Delta\boldsymbol{\lambda}_k\end{Bmatrix}=\begin{Bmatrix}\boldsymbol{r}(\boldsymbol{q}_{n+1},\boldsymbol{\lambda}_{n+1})\\ \boldsymbol{G}(\boldsymbol{q}_{n+1})\dot{\boldsymbol{q}}_{n+1}\end{Bmatrix}_k \tag{2.6.61}$$

where $\boldsymbol{G_q} = \partial \boldsymbol{G}/\partial \boldsymbol{q}$. As the position constraints are eliminated from this formulation, it is advisable to monitor the evolution of $\|\boldsymbol{g}(\boldsymbol{q})\|$ and apply a position projection (Eq. 2.6.52) when the latter exceeds a user-specified tolerance.

Algorithm 2.4 Generalized-α method for index-3 DAEs

Input: system matrices and vectors $\boldsymbol{M}(\boldsymbol{q}), \boldsymbol{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, t), \boldsymbol{g}(\boldsymbol{q})$ and initial conditions $\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0, \boldsymbol{\lambda}_0$

Output: solution vectors $\boldsymbol{q}(t), \dot{\boldsymbol{q}}(t), \ddot{\boldsymbol{q}}(t), \boldsymbol{\lambda}(t)$ for $t \in [t_0, t_1, \dots, t_N]$

1. Initialize $\ddot{\boldsymbol{q}}_0 = \boldsymbol{M}^{-1}(\boldsymbol{f}(\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0, t_0) - \boldsymbol{G}^T \boldsymbol{\lambda}_0)$
2. **for** $t = t_1 \dots t_N$ **do**
3. Set $\ddot{\boldsymbol{q}}_i = \boldsymbol{0}$ and $\boldsymbol{\lambda}_i = \boldsymbol{0}$
4. Predict $\boldsymbol{q}_{n+1} = \boldsymbol{q}^*_{n+1}$ and $\dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}^*_{n+1}$ using Eq. J.6
5. **for** $k = 1 \dots itermax$ **do**
6. Compute $\boldsymbol{r}_{n+1} = \boldsymbol{r}(\boldsymbol{q}_{n+1}, \boldsymbol{\lambda}_{n+1})$ and $\boldsymbol{g}_{n+1} = \boldsymbol{g}(\boldsymbol{q}_{n+1})$ using Eq. 2.6.57
7. Check for convergence: $\sqrt{\|\boldsymbol{r}_{n+1}\| + \|\boldsymbol{g}_{n+1}\|} \leq TOL \Longrightarrow END\ LOOP$
8. Form iteration matrix as defined in Eq. 2.6.58 (if update is needed)
9. Compute Newton increments by solving Eq. Eq. 2.6.58
10. Update solution vectors using Eq. 2.6.59
11. **end**
12. Update quasi-accelerations using Eq. J.4
13. **end**

2.7 Summary and conclusion

In this chapter, the space-time discretized Equations Of Motion (EOMs) of a flexible body undergoing large rigid-body motion and subjected to mechanical contact interactions were derived from first principles (see Fig. 2.11). In Section 2.1 the fundamental EOMs of a linear elastic solid were derived from Hamilton's principle and Lagrange's equations. Using Galerkin's weighted residual approach, these EOMs were converted to weak form and spatially discretized using the finite element method in Section 2.2. In Section 2.3 the large numbers of degrees-of-freedom and the high computational complexity of nonlinear functions in these EOMs were reduced using parametric model-order reduction and hyperreduction methods. Large non-linear rigid-body motion was added to the equations in Section 2.4 by combining the reduced-order finite element method with the Floating Frame of Reference Formulation (FFRF), and mechanical contact interactions were taken into account in the resulting EOMs in Section 2.5. Finally, the EOMs were discretized in time using either explicit or implicit integration methods in Section 2.6. These equations are suitable for computer implementation and yield a numerical solution to the initial value problem of continuum mechanics. The modelling framework outlined in this chapter will be used in subsequent chapters to develop and solve the reduced-order EOMs of the dynamic gear contact problem.

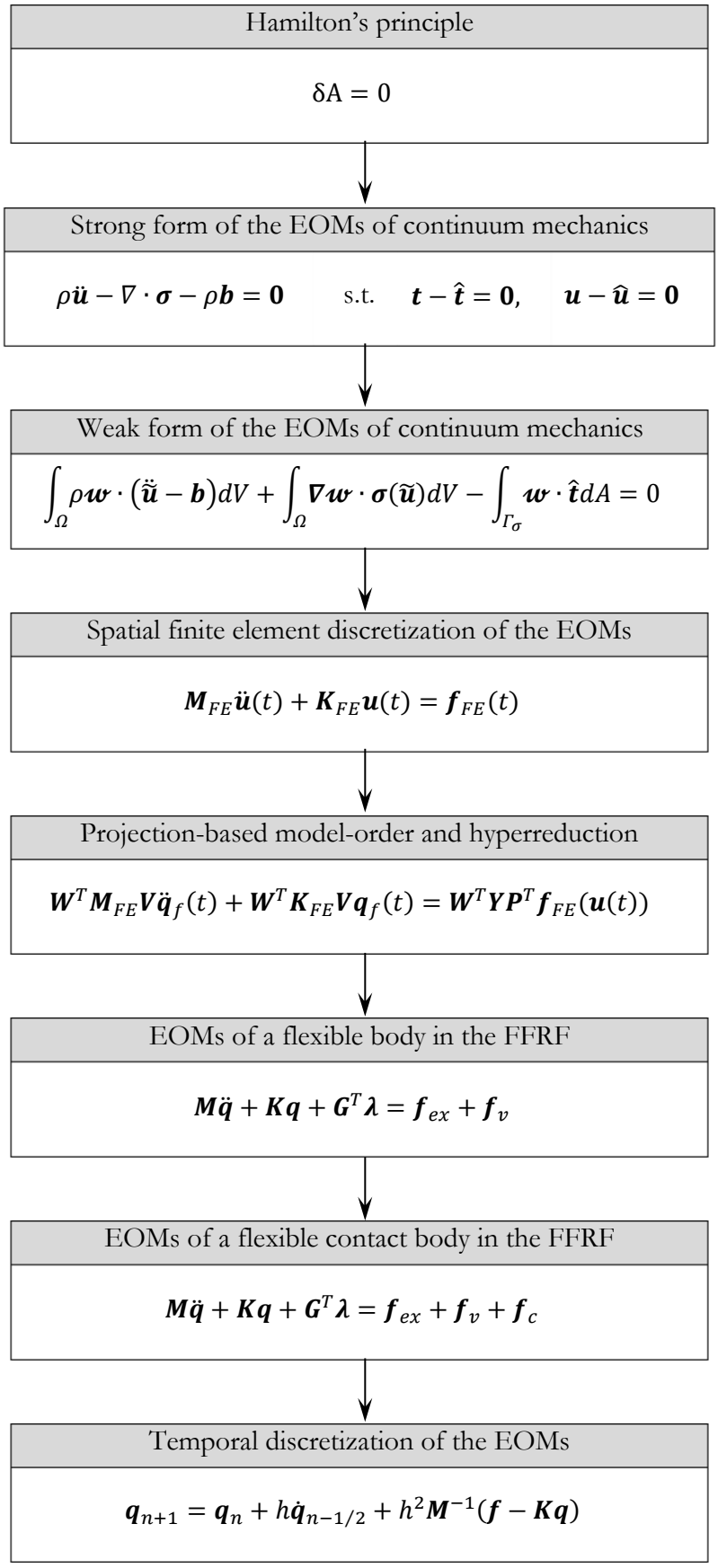


Fig. 2.11 Derivation of the space-time discretized EOMs from first principles

Chapter 3

Description of the dynamic gear contact problem

The main objective of this dissertation is to improve the efficiency of Finite Element (FE) based contact simulations in flexible Multibody System Dynamics (MBSD). The dynamic gear contact problem was selected as a representative case study for two main reasons. First, like many other contact problems in mechanical engineering, gear contact problems are characterized by complex geometries, large rigid-body motions, and continuously varying contact locations. Secondly, of all the many types of machine elements that exist in industry today, gears remain among the most commonly used.

This chapter presents a detailed description of the full-order gear contact model that will be used in subsequent chapters to investigate newly developed model reduction techniques and numerical integrators. In Section 3.1, an overview is given of the essential concepts of involute gear theory. These concepts are considered as the theoretical minimum required to understand the remainder of this dissertation, with more advanced concepts related to the computer implementation of dynamic gear contact models being deferred to the appendices. In Section 3.2 the construction of the reference dynamic gear contact model based on the FE discretization of analytical gear geometries is discussed, and the numerical case studies that will be used in subsequent chapters are introduced. Finally, the reference gear contact model is validated through comparison with nonlinear FE software and state-of-the-art gear contact models in Section 3.3.

3.1 Essential concepts of involute gear geometry

The extreme simplicity of a toothed wheel driving another by means of meshing teeth seems to be in stark contrast with the thousands of books and journal papers that have been written on the subject of gear geometry. Yet the enormous improvements in the performance of geared transmissions during the last century are primarily due to the careful attention given to the shape of gears. In this chapter cylindrical gears with involute tooth profiles are considered. These types of gears represent the majority of gears used in industry nowadays, and their geometry is investigated in great detail in [156, 157, 158]. The summary provided in this section is restricted to the concepts that are relevant for the upcoming chapters.

External cylindrical gears have outward pointing teeth that are either parallel to the gear axis (spur gears) or inclined at a certain angle (helical gears). The main geometric features of an external helical gear are shown in Fig. 3.1. Note that a spur gear is a special case of a helical gear with zero helix angle. The teeth of a gear are numbered in ascending order moving in a clockwise direction and proceeding from tooth 1 until tooth z. The top and bottom land are the outermost and innermost peripheries of the tooth. During normal operation, contact interactions are confined to the involute tooth flank, i.e. the part of the tooth surface that is located between the top land and the root fillet. Two cross-sections are generally used to define gear quantities; these are the transverse and normal sections, which are obtained by sectioning the gear by a plane perpendicular to the gear axis and a plane perpendicular to the tooth flank, respectively. Quantities in the transverse section are denoted by the subscript '*t*', whereas quantities on the normal surface are denoted by the subscript '*n*'. Note that for a spur gear, the transverse and normal sections coincide.

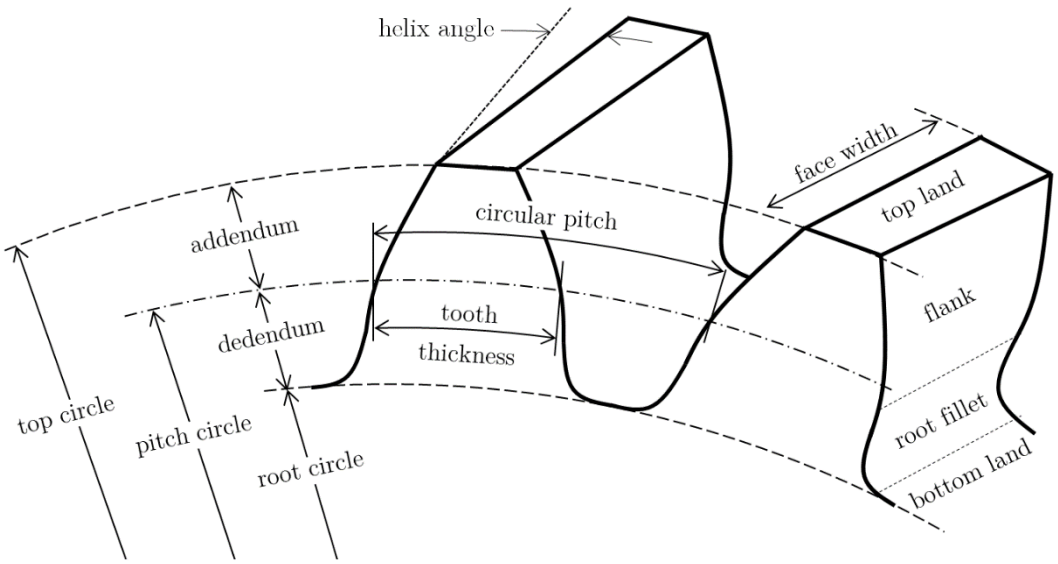


Fig. 3.1 Main geometrical parameters of an external cylindrical gear

When two gears mesh, the contact forces are directed along the Line Of Action (LOA), which is the common tangent to the respective base circles of the engaging gears, see Fig. 3.2. For involute undeformable gears, the line of action is constant and coincides with the path of contact. The pressure angle is defined as the complement of the angle between the line of action and the center line (i.e. the line joining the two centers of the gears), and usually has a value of around 20°. The diameter at which the line of action crosses the center line is referred to as the *pitch diameter* or reference diameter. The distance from one face of a tooth to the corresponding face of an adjacent tooth is generally measured along this diameter, and is referred to as the circular pitch (see Fig. 3.1). This distance is more conveniently expressed in terms of the *modulus*, which is simply defined as the circular pitch divided by π. For a pair of gears to mesh properly, they should have the same

moduli and pressure angles, and their helix angles should be of equal magnitude yet opposite sign. The pitch diameter d, modulus m, helix angle β, and number of teeth z of a gear are related by the equation:

$$d = \frac{m_n z}{\cos \beta} = m_t z \tag{3.1.62}$$

where m_n and m_t denote the module in a normal and transverse section, respectively. Finally, the addendum denotes the radial distance between the pitch circle and the top circle, whereas the dedendum measures the distance between the pitch circle and the root circle.

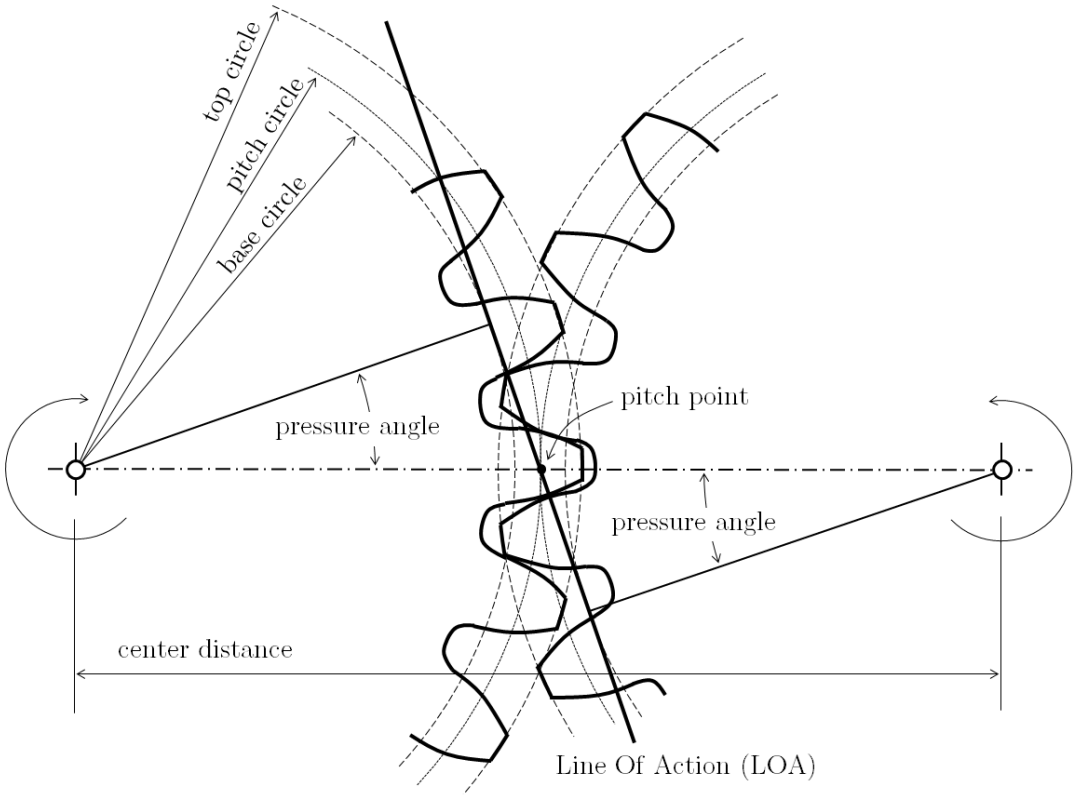


Fig. 3.2 Schematic representation of the line of action of a pair of spur gears

For internal gears, similar concepts and definitions apply as for external gears (see Fig. 3.3). Note however that the top and bottom land of an internal tooth now respectively correspond to the *inner*most and *outer*most peripheries of the tooth, and that the top circle is the smallest of the characteristic circles. Despite the similarities in geometry between internal and external gears, the manufacturing processes of these gears differ substantially, which is why they are treated separately in the upcoming sections.

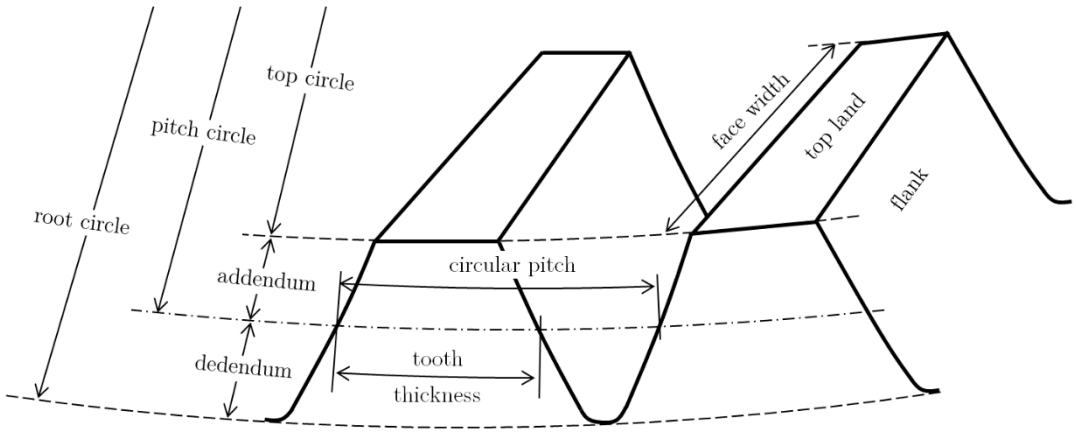


Fig. 3.3 Main geometrical parameters of an internal cylindrical gear

The equations describing the geometry of external and internal cylindrical gears are obtained through a kinematic description of the manufacturing process. These equations form the basis of the FE discretization of the gear geometries and are discussed in detail in Appendix K (for external gears) and Appendix L (for internal gears). In Appendix M the fundamental kinematic relations of involute gear pairs are discussed; these relations are of particular interest for accelerating the contact detection between two engaging discretized tooth flanks (see Section 3.2.4).

3.2 Construction of the dynamic gear contact model

3.2.1 Discretization of the gear geometry

In this dissertation the geometry of cylindrical gears is discretized by the Finite Element Method (FEM; see Section 2.2) using the geometry equations derived in Appendix K and Appendix L. Assuming each gear is rotationally periodic with period $2\pi/z$ (i.e. the gear's angular pitch), the procedure to construct the finite element model of a gear is as follows: first a two-dimensional mesh of a single tooth is constructed based on a number of parameters of which especially the number of elements along the involute tooth profile N_{fef} and along the gear width N_{few} are important; this mesh is then copied and rotated $z-1$ times to obtain a two-dimensional FE mesh of the full gear; finally, the three-dimensional FE model is obtained by extruding (and rotating, in the case of helical gears) the two-dimensional mesh in the direction of the gear axis. The angle of rotation from the front *slice* of the gear to the back slice is given by

$$\beta' = \frac{2b\tan\beta}{d} \tag{3.2.1}$$

where b denotes the face width of the gear. In order to reduce the amount of computer memory required to store the FE models and at the same time limit the pre-processing time of the simulation, a mixed-mesh is applied where only $\lceil\varepsilon\rceil+1$ teeth are finely meshed, while the

remaining teeth are meshed using a coarser mesh (ε is the gear pair's contact ratio and $\lceil \cdot \rceil$ denotes rounding to the next integer). The procedure to perform full-revolution gear contact simulations using this mixed-mesh model while preserving the accuracy of finely meshed gears is described in Chapter 4. The construction of the mixed-mesh FE model is summarized in Fig. 3.4.

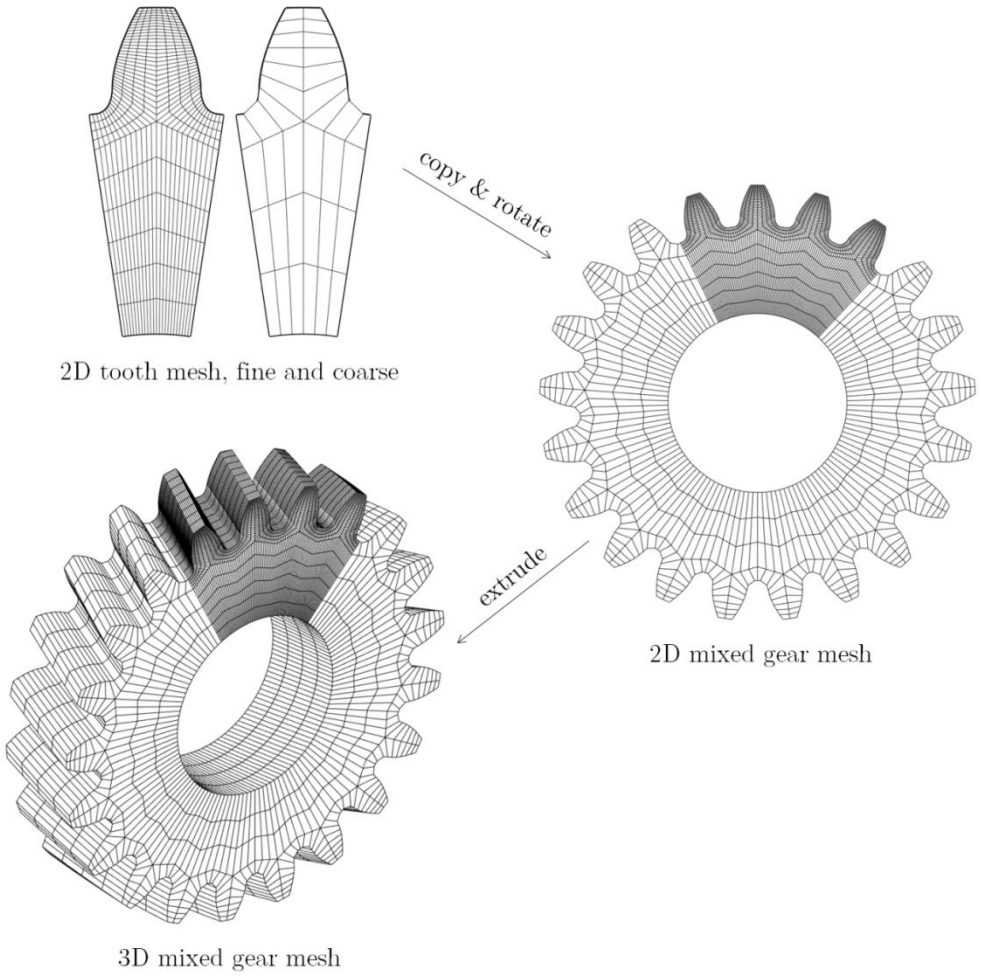


Fig. 3.4 Construction of the three-dimensional FE model of a helical gear with ε between 2 and 3

The element type that is used to discretize the gear geometry is the linear hexahedral 8-noded solid element (see Section 2.2.2). As the boundaries of these elements are composed of straight lines, the discretized involute flank is described by a piecewise-linear approximation of the theoretical involute. This introduces a discretization error that increases as the load applied to the gear is decreased. The error is largest when contact occurs near the root of one of the gears (points A and E in Fig. M.1), where the local radius of curvature is minimal. Let $\Delta u(d)$ denote the normal difference between the theoretical involute and the piecewise-linear approximation as a function of the diameter d, then the gear angle at start of tooth engagement (see Table M.3) will increase by an amount of $\Delta\theta$ according to:

$$\theta_{A1}^{FE} = \theta_{A1} + \Delta\theta_{A1} \approx \theta_{A1} + tan^{-1}\left(\frac{\Delta u_1(d_{Nf1})}{d_{Nf1}}\right) \tag{3.2.2a}$$

$$\theta_{A2}^{FE} = \theta_{A2} + \Delta\theta_{A2} \approx \theta_{A2} + tan^{-1}\left(\frac{\Delta u_2(d_{Na2})}{d_{Na2}}\right) \tag{3.2.2b}$$

Note that since $\varrho_{Nf1} \ll \varrho_{Nf2}$ (see Fig. M.2), it generally holds that $\Delta\theta_{A1} \gg \Delta\theta_{A2}$. The above equation can be used to guide the selection of the number of elements along the flank.

When an accurate representation of the surface contact stresses is necessary, the number of elements along the flank should be adjusted to the sectional width of the contact zone. The Hertzian formula for two elastic cylinders with parallel axes being pressed together can be used to provide a rough estimate for this width [17]. Selecting contact at the pitch point as a reference, the estimated contact width b_c is given by:

$$b_c = 4\sqrt{\frac{T^1}{\pi b_w \varepsilon d}\frac{2\,(1-\nu^2)/E}{1/\varrho^1 + 1/\varrho^2}} \tag{3.2.3}$$

where T^1 is the torque applied to the driving gear, b_w is the active width of the gear pair, ε is the contact ratio, d is the pitch diameter, ν and E are, respectively, the Poission coefficient and modulus of elasticity, and ϱ is the radius of curvature at the pitch point as defined by Eq. M.6.

3.2.2 Construction of the flexible multibody model

Having obtained the coordinates of the FE nodes and the element connectivity relations, the mass and stiffness matrix of the FE model can be computed using the equations developed in Section 2.2. Large rigid-body motion of the FE model is accounted for through the Floating Frame of Reference (FFR) formulation (Section 2.4). To this end, a local reference frame $\boldsymbol{x}_1^i\boldsymbol{x}_2^i\boldsymbol{x}_2^i$ is rigidly attached to the center of the *i*-th gear[20] by means of a rigid MultiPoint Constraint (MPC; RBE2 in Nastran terminology [121]), see Fig. 3.5. The location of the origin of $\boldsymbol{x}_1^i\boldsymbol{x}_2^i\boldsymbol{x}_3^i$ is described in the global reference frame $\boldsymbol{X_1X_2X_3}$ by the position vector $\boldsymbol{R}^i$, and the orientation of $\boldsymbol{x}_1^i\boldsymbol{x}_2^i\boldsymbol{x}_2^i$ w.r.t. $\boldsymbol{X_1X_2X_3}$ is described by the transformation matrix $\boldsymbol{A}^i$, which is expressed in terms of the three-parameter set of Bryant angles (Appendix C). The Equations Of Motion (EOMs) of the *i*-th flexible gear model are derived using Lagrange's equations (Section 2.4.3) and have the following form:

[20] Note that the notational convention of superscripts is changed here with respect to Chapter 2. From here on, the superscript *i* will be used to denote the *i*-th flexible body in the multibody system, whereas the the superscript *j* will be used to refer to the *j*-th node. Therefore, the position of the *j*-th node on the *i*-th flexible body in a multibody system is denoted as $\boldsymbol{r}^{ij}$.

$$\underbrace{\begin{bmatrix} \boldsymbol{M}_{RR} & \boldsymbol{M}_{R\theta} & \boldsymbol{M}_{Rf} \\ & \boldsymbol{M}_{\theta\theta} & \boldsymbol{M}_{\theta f} \\ sym. & & \boldsymbol{M}_{ff} \end{bmatrix}^i}_{\boldsymbol{M}^i} \begin{Bmatrix} \ddot{\boldsymbol{R}}^i \\ \ddot{\boldsymbol{\theta}}^i \\ \ddot{\boldsymbol{q}}_f^i \end{Bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ & \boldsymbol{0} & \boldsymbol{0} \\ sym. & & \boldsymbol{C}_{ff} \end{bmatrix}^i}_{\boldsymbol{C}^i} \begin{Bmatrix} \dot{\boldsymbol{R}}^i \\ \dot{\boldsymbol{\theta}}^i \\ \dot{\boldsymbol{q}}_f^i \end{Bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ & \boldsymbol{0} & \boldsymbol{0} \\ sym. & & \boldsymbol{K}_{ff} \end{bmatrix}^i}_{\boldsymbol{K}^i} \begin{Bmatrix} \boldsymbol{R}^i \\ \boldsymbol{\theta}^i \\ \boldsymbol{q}_f^i \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f}_R^i \\ \boldsymbol{f}_\theta^i \\ \boldsymbol{f}_f^i \end{Bmatrix} \quad (3.2.4)$$

where the submatrices and vectors are defined in Eq. 2.4.11. Note that all submatrices in the above equation are derived from the mass and stiffness matrix of the FE model. The vector of generalized elastic coordinates $\boldsymbol{q}_f^i$ is defined by the Galerkin-based approximation $\boldsymbol{u}^i \approx \boldsymbol{V}^i \boldsymbol{q}_f^i$ (Eq. 2.3.1), where $\boldsymbol{u}^i$ is the vector of nodal coordinates and $\boldsymbol{V}^i$ is the selected reduction basis of the FE model of the *i*-th gear. The reduction basis for the chosen reference model in this dissertation is described in the Section 3.2.3, whereas Chapter 4 proposes a novel type of reduction basis that forms one of the main topics of this dissertation. The contact forces enter into the gear's EOMs either by specifying the active contact constraints as $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$ and adding the Lagrangian constraint term to Eq. 3.2.4 (see Section 2.5.3.1), or simply as external forces (see Sections 2.5.3.2).

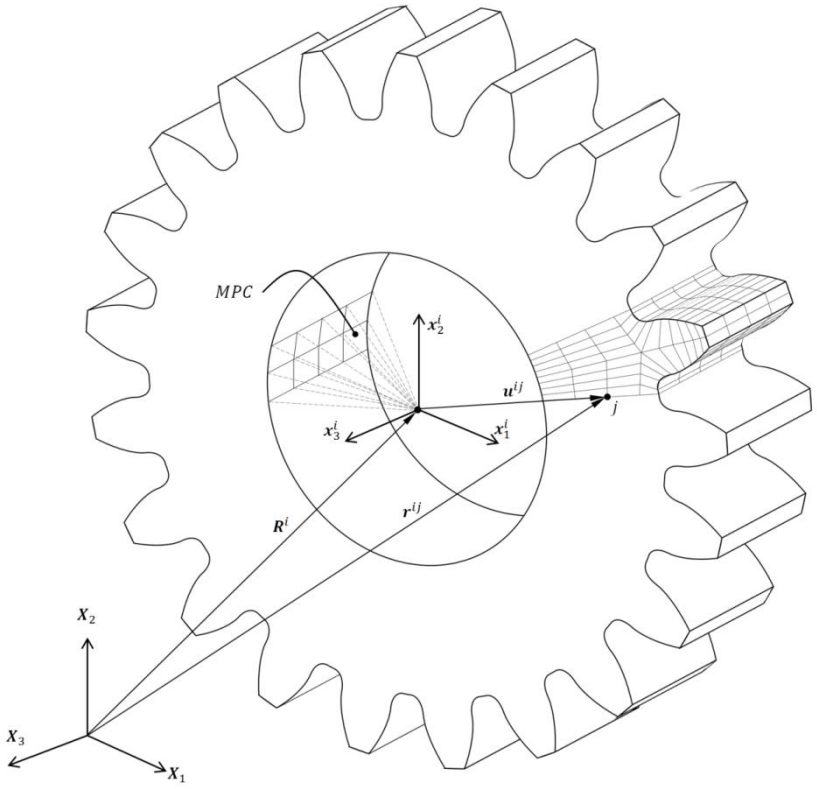


Fig. 3.5 Floating frame of reference description of a flexible gear

Gears are usually either mounted on shafts, or directly supported by rolling element bearings (e.g. in planetary gears). In reality, these shafts and bearings deform under applied loads, causing the gears to misalign. The flexible supports of gears are typically modelled using three-dimensional spring-dampers that are characterized by 6-by-6 stiffness and damping matrices. These spring-

dampers act on the rigid-body coordinates of the gears and produce a *bearing* force $\boldsymbol{f}_b^i$ that is defined as:

$$\boldsymbol{f}_b^i = \begin{Bmatrix} \boldsymbol{f}_{b,R}^i \\ \boldsymbol{f}_{b,\theta}^i \\ \boldsymbol{f}_{b,f}^i \end{Bmatrix} = \underbrace{\begin{bmatrix} \boldsymbol{C}_{b,RR} & \boldsymbol{C}_{b,R\theta} & \boldsymbol{0} \\ & \boldsymbol{C}_{b,\theta\theta} & \boldsymbol{0} \\ sym. & & \boldsymbol{0} \end{bmatrix}^i}_{\boldsymbol{C}_b} \begin{Bmatrix} \Delta\dot{\boldsymbol{R}}^i \\ \Delta\dot{\boldsymbol{\theta}}^i \\ \Delta\dot{\boldsymbol{q}}_f^i \end{Bmatrix} + \underbrace{\begin{bmatrix} \boldsymbol{K}_{b,RR} & \boldsymbol{K}_{b,R\theta} & \boldsymbol{0} \\ & \boldsymbol{K}_{b,\theta\theta} & \boldsymbol{0} \\ sym. & & \boldsymbol{0} \end{bmatrix}^i}_{\boldsymbol{K}_b} \begin{Bmatrix} \Delta\boldsymbol{R}^i \\ \Delta\boldsymbol{\theta}^i \\ \Delta\boldsymbol{q}_f^i \end{Bmatrix} \tag{3.2.5}$$

where $\Delta\boldsymbol{R}^i$, $\Delta\boldsymbol{\theta}^i$, and $\Delta\boldsymbol{q}_f^i$ represent the generalized displacements with respect to the initial conditions. In the $\boldsymbol{K}_b$ and $\boldsymbol{C}_b$ matrices, the entries corresponding to the third Bryant angle are generally zero, denoting unconstrained rotation of the gear about its axis of rotation. For rolling element bearings, the matrices $\boldsymbol{K}_b$ and $\boldsymbol{C}_b$ are usually provided by the bearing manufacturer; if not, they can be computed using e.g. the three-dimensional bearing model described in [22]. For a gear supported by a flexible shaft, the equivalent-spring representation of the shaft can readily be obtained by means of finite elements. To this end, the shaft is successively loaded by three unit loads (along $\boldsymbol{x}_1^i, \boldsymbol{x}_2^i,$ and $\boldsymbol{x}_3^i$) and two unit torques (about $\boldsymbol{x}_1^i$ and $\boldsymbol{x}_2^i$) at the location of the gear bore; from the resulting static displacements and rotations, the shaft's compliance matrix can be assembled, which upon inversion yields the equivalent stiffness matrix of the shaft.

In the particular case where the gear pair is perfectly aligned and supported by rigid shafts or bearings (i.e. $\Delta\boldsymbol{R}^i = \boldsymbol{0}, \phi = \psi = 0$), the EOMs of the *i*-th gear reduce to:

$$\underbrace{\begin{bmatrix} M_{\theta\theta} & \boldsymbol{M}_{\theta f} \\ sym. & \boldsymbol{M}_{ff} \end{bmatrix}^i}_{\boldsymbol{M}^i} \begin{Bmatrix} \ddot{\theta}^i \\ \ddot{\boldsymbol{q}}_f^i \end{Bmatrix} + \underbrace{\begin{bmatrix} 0 & \boldsymbol{0} \\ sym. & \boldsymbol{C}_{ff} \end{bmatrix}^i}_{\boldsymbol{C}^i} \begin{Bmatrix} \dot{\theta}^i \\ \dot{\boldsymbol{q}}_f^i \end{Bmatrix} + \underbrace{\begin{bmatrix} 0 & \boldsymbol{0} \\ sym. & \boldsymbol{K}_{ff} \end{bmatrix}^i}_{\boldsymbol{K}^i} \begin{Bmatrix} \theta^i \\ \boldsymbol{q}_f^i \end{Bmatrix} = \begin{Bmatrix} f_\theta^i \\ \boldsymbol{f}_f^i \end{Bmatrix} \tag{3.2.6}$$

where the submatrices and vectors are defined by Eq. 2.4.28. Chapter 4 deals with the efficient solution of Eq. 3.2.6, while Chapter 6 is concerned with finding a cost-effective solution to Eq. 3.2.4-3.2.5.

3.2.3 Definition of the reference model

The primary objective of this dissertation is to develop a novel Model Order Reduction (MOR) method that yields Reduced-Order Models (ROMs) which are considerably more efficient than what is achievable using state-of-the-art MOR techniques. Therefore, instead of using the unreduced FE model as the reference model in this dissertation[21], it was opted to select a ROM that is constructed using the Component Mode Synthesis (CMS) method (Section 2.3.3.1), which is the standard MOR method in flexible MBSD today.

[21] This option would anyway not be computationally feasible in MATLAB due to memory restrictions for all of the selected case studies.

In the CMS approach, the reduction bases $\boldsymbol{V^1}$ and $\boldsymbol{V^2}$ of the respective gears are composed of two kinds of component modes: one attachment mode $\boldsymbol{\Psi_a^i}$ for each nodal DOF that can be loaded in the course of the simulation, and a limited number of natural vibration modes $\boldsymbol{\Phi^i}$ (see Fig. 3.6), i.e.:

$$\boldsymbol{V^i} = [\boldsymbol{\Psi_a^i} \quad \boldsymbol{\Phi^i}] \tag{3.2.7}$$

In this dissertation, three attachment mode are computed for each node on the flanks of the finely meshed teeth. In doing so, the dimensions of the attachment modes matrix $\boldsymbol{\Psi_a^i}$ of the *i*-th gear are:

$$\boldsymbol{\Psi_a^i} \in \mathbb{R}^{N \times [3N_{fef}N_{few}([\varepsilon]+1)]} \tag{3.2.8}$$

with N the number of nodal DOFs in the full-order finite element model, and N_{fef} and N_{few} are the number of nodes along the involute curve and along the width of the gear. For a gear pair having a contact ratio ε between 2 and 3, selecting $N_{fef} = 30$ and $N_{few} = 10$ yields a total number of 3600 attachment modes. The resulting ROM is said to be *statically complete*, as it yields the exact same solution as the unreduced FE model under static conditions.

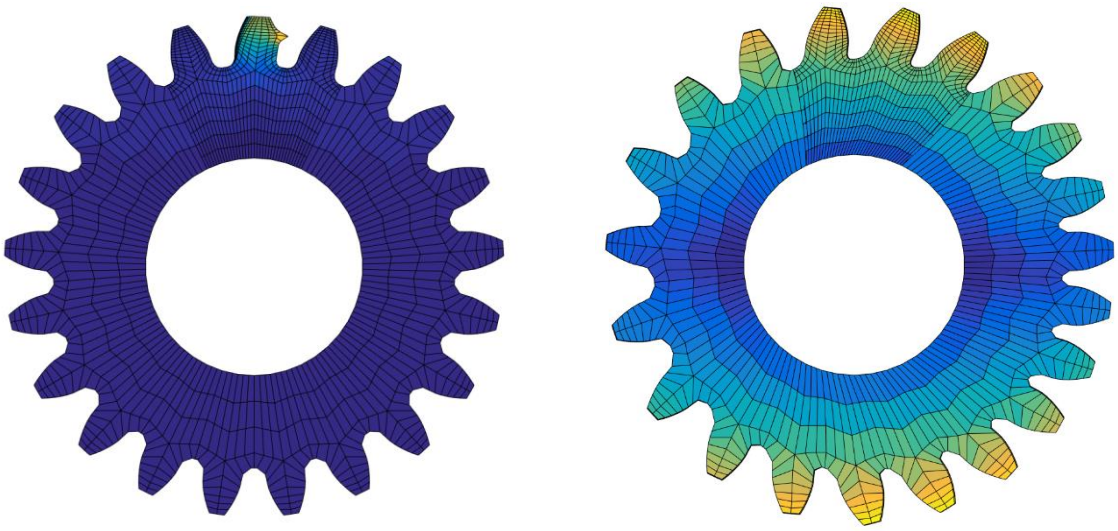


Fig. 3.6 Two example component modes of the reduced-order FE-based gear contact model: an attachment mode (left) and a natural vibration mode (right)

In addition to being statically complete, the CMS-based ROM can be made *dynamically complete* by including a sufficiently large number of natural vibration modes[22]. As there are no a priori means for estimating this number in a reliable way, a series of test runs is conducted in which the number of natural vibration modes is gradually increased until convergence. In Fig. 3.7, a gear pair initially

[22] It is assumed that the number of natural vibration modes is sufficiently large such that the subspace it spans approximately includes the subspace spanned by the inertia relief modes (see Section 2.3.3.1).

at rest is acted upon by an external torque applied to the driving gear; this torque is counteracted by a viscous (i.e. velocity-proportional) resisting torque acting on the driven gear that is tuned such as to achieve a pre-defined angular velocity. High values for both the external torque and pre-defined angular velocity are selected so as to excite the internal dynamics of the gear pair. As can be seen from Fig. 3.7, the Dynamic Transmission Error (DTE, to be defined in Section 3.3) predicted using 100 resp. 1000 natural vibration modes are virtually indistinguishable, with a relative root-mean-square error of less than 0.5% in the time window between 4 and 5 ms (where oscillations are most severe due to the changing number of teeth in contact). In the same time window, excluding all natural vibration modes from the analysis yields a relative root-mean-square error of approximately 18%, whereas the 1% threshold is reached by adding 50 natural vibration modes per gear.

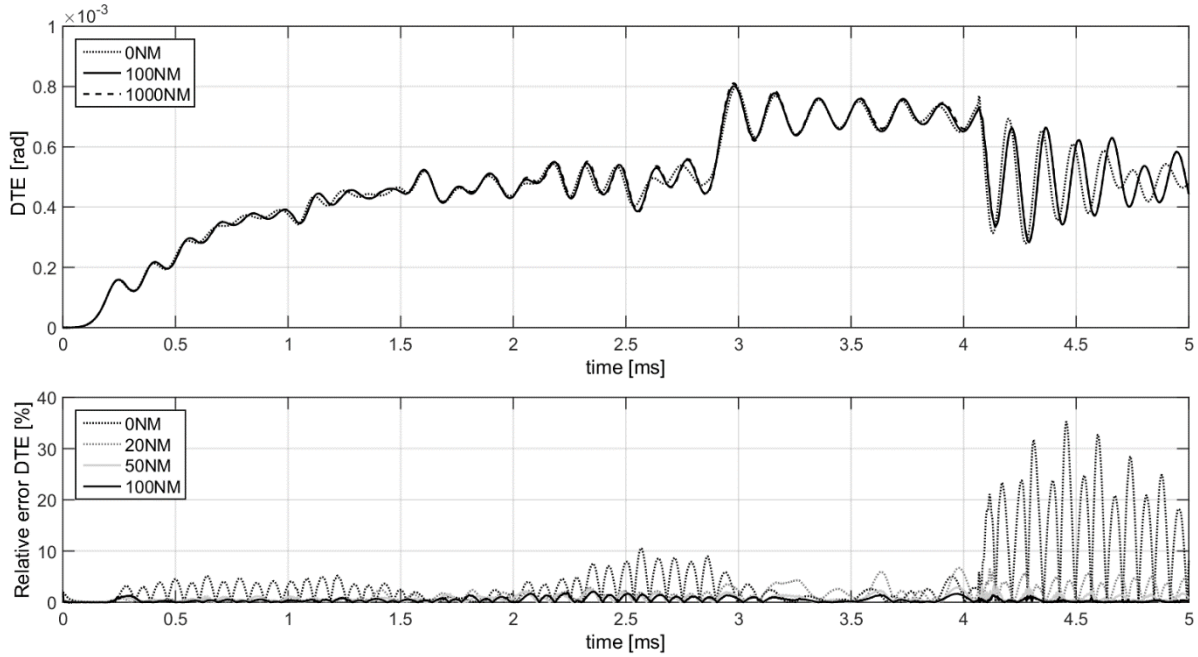


Fig. 3.7 Comparison of the dynamic transmission error (above) and the relative error on the dynamic transmission error for different numbers of natural vibration modes (NM)

Depending on whether or not misalignments are included in the flexibly multibody model (Eq. 3.2.6 vs. 3.2.4), the gear contact model has 1 to 6 rigid DOFs per gear, in addition to 3600 (i.e. the number of attachment modes) + 50 (i.e. the number of natural vibration modes) generalized elastic DOFs per gear. Owing to it static and dynamic completeness, this gear contact model can be assumed to have similar accuracy as the unreduced FE model, and is therefore selected as the primary reference model.

3.2.4 Contact detection algorithm

In this dissertation, a two-pass Node-To-Surface (NTS) contact detection algorithm is used to enforce the contact constraints of Eq. 2.5.4 (see Section 2.5.2). The kinematic relations developed in Appendix M are used to accelerate the process of detecting the contact nodes, leading to the following three-step approach:

1. Identification of the teeth in contact (for gear 1; the same applies to gear 2):
    a. Computation of the teeth angles: $\theta_{1,i} = \theta_1 + (i-1) \cdot \tau_1$ for $i = 1 \dots z$;
    b. Comparison with teeth angles at points A and E: if $\theta_{A1} \leq \theta_{1,i} \leq \theta_{E1}$ then the tooth with angle $\theta_{1,i}$ is in contact.

2. Identification of the *possible* contact nodes (see Fig. 3.8):
    a. Selection of a number of reference transverse sections (*slices)*;
    b. Computation of the contact diameter $d_{y1,i}$ using Eq. M.8;
    c. Computation of the contact width $b_{c,i}$ using a modified version of Eq. 3.2.3: $\varepsilon \leftarrow 1$
    d. Selection of the possible contact nodes based on $d_{1,i}$ and $b_{c,i}$;
    e. Interpolation of the possible contact nodes to the remaining slices.

3. NTS contact search to define the *actual* contact nodes and normal gaps:
    a. Loop over all the master segments;
    b. Loop over all the slave nodes that are linked to the selected master segment;
    c. Projection of the selected slave node on the selected master segment (Eq. 2.5.8);
    d. Computation of the surface normal at the minimal distance point (Eq. 2.5.9);
    e. Computation of the normal gap g_N (Eq. 2.5.11);
    f. Check for penetration: if $g_N < 0$ then slave node is in contact.

Steps 1 and 2 of this procedure are referred to as the *coarse* contact detection phase, while the third step is denoted as the *fine* contact detection. Note that for a perfectly aligned spur gear, it is sufficient to select a single slice in Step 2a, whereas for misaligned or helical gears, multiple slices need to be considered. Safety factors are applied in steps 1b and 2c to ensure a sufficient amount of possible contact nodes are selected. In step 3b, the selection of a subset of slave nodes corresponding to a given master segment is based on the axial coordinates of the slave nodes.

When the penalty approach is used to enforce the contact constraints (Section 2.5.3.2), the NTS algorithm computes the contact forces by multiplying the normal gap g_N with a penalty factor ϵ_N (see 2.5.19)). The penalty factor should be selected as high as possible so as to minimize unphysical penetration of the slave surface into the master surface, yet small enough to avoid ill-conditioning of the problem. Unphysical penetration results in an error in the angular

configuration of the gear, $\Delta\theta$, that can be related to the penalty factor through the following formula:

$$\epsilon_{N,min} \approx \frac{4T}{\Delta\theta_{max}^{1} d^2 \cos\alpha_n \lfloor\varepsilon\rfloor n_{n,c}} \tag{3.2.9}$$

where T denotes the torque applied to the driving gear, d is the gear's pitch diameter, $\lfloor\cdot\rfloor$ denotes rounding to the previous integer, and $n_{n,c}$ is the number of nodes in the contact zone, which depends on the number of elements along the active face and contact width of the gear. In addition, the penalty factor should be roughly 3 orders of magnitude larger than the stiffness of the gear at the contact point. The latter can easily be obtained by applying a unit load in the normal direction to the flank of the gear (say, at a node near the pitch point) and computing the resulting nodal displacement; the inverse of the latter yields the stiffness at the pitch point.

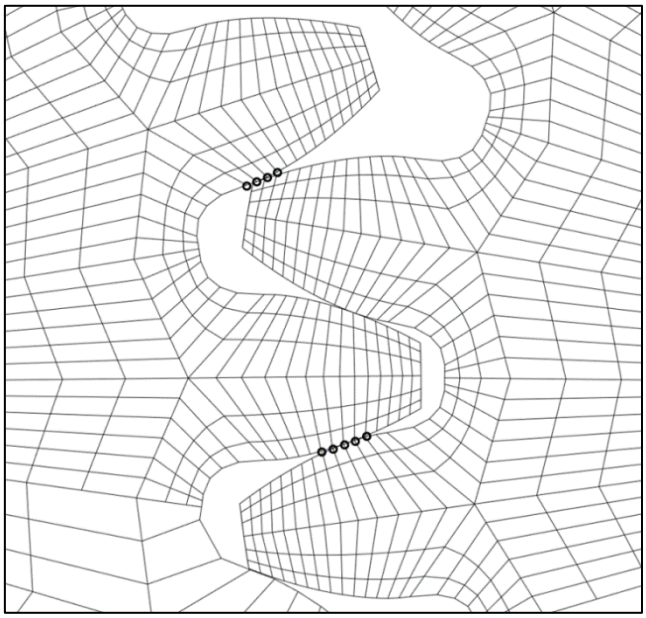


Fig. 3.8 Possible contact nodes (encircled in black) in a transverse section of a helical gear

3.2.5 Selected case studies

Three gear sets will be used throughout this dissertation to validate the newly developed model reduction techniques. These are a spur and helical parallel-axis gear pair and a helical planetary gear set. The parallel-axis spur gear pair consists of two identical spur gears, whereas the parallel-axis helical gear pair consists of two gears having different number of teeth and widths. Depending on the chapter in which they are considered, these gears either have only torsional rigid-body freedom (described by the third Bryant angle), or three-dimensional rigid-body freedom (described by three Cartesian coordinates and three Bryant angles). The main geometrical characteristics of the

parallel-axis gear pairs as well as the load conditions are summarized in Table 3.1. The FE models of the spur and helical parallel-axis gear pairs are shown in Fig. 3.9 and Fig. 3.10, respectively.

	Spur gear pair		**Helical gear pair**	
Parameters	**Pinion**	**Wheel**	**Pinion**	**Wheel**
Number of teeth	21	21	21	28
Normal modulus [mm]	4	4	4	4
Helix angle [deg]	0	0	20	-20
Normal pressure angle [deg]	20	20	20	20
Pitch diameter [mm]	84	84	89.39	119.19
Tip diameter [mm]	92	92	97.39	127.19
Root diameter [mm]	74	74	79.39	109.19
Bore diameter [mm]	30	30	40	40
Radius of rack corner [mm]	1	1	1	1
Face width [mm]	20	20	20	25
Center distance [mm]	84		104.29	
Input torque [Nm]	250	n.a.	250	n.a.

Table 3.1 Parameters of the parallel-axis gear sets

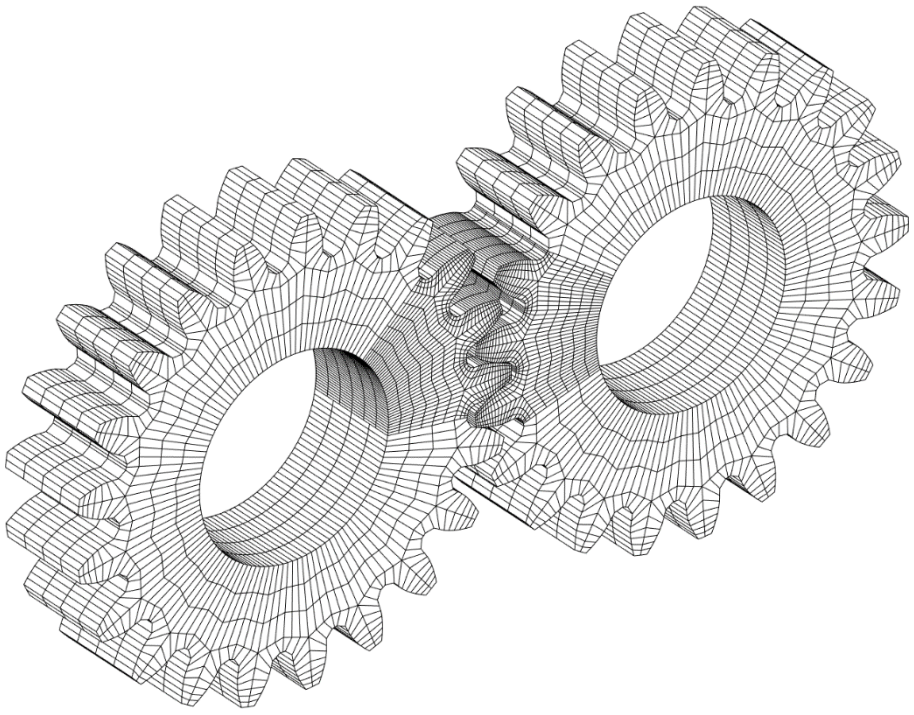


Fig. 3.9 Finite element model of the spur gear pair

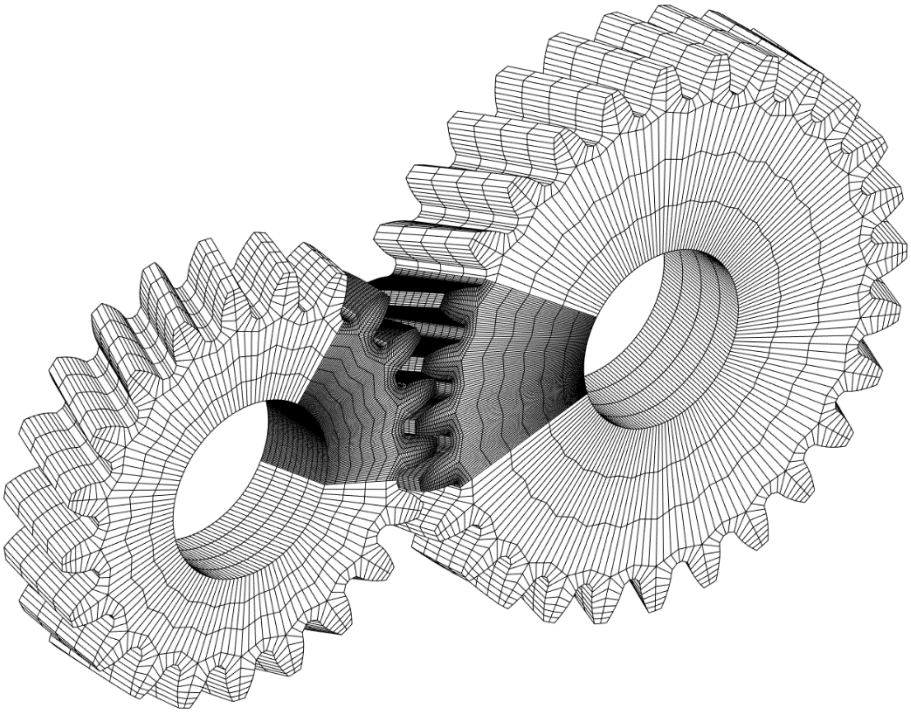


Fig. 3.10 Finite element model of the helical gear pair

The planetary gear set studied in this dissertation is the primary stage of a 750kW wind turbine [159]. It is composed of a ring gear with internal teeth (① in **Fig. 3.11**), a planet carrier (②), three planet gears with external teeth (③), and a sun gear with external teeth (④). The planetary gear set is operated in the fixed-ring configuration: the ring gear is held stationary while the planet carrier is acted upon by an external torque that is counteracted by the torque acting on the sun gear. The main geometrical characteristics of the planetary gear set as well as the load conditions are summarized in Table 3.2, and an FE model of the gear set is shown in Fig. 3.12.

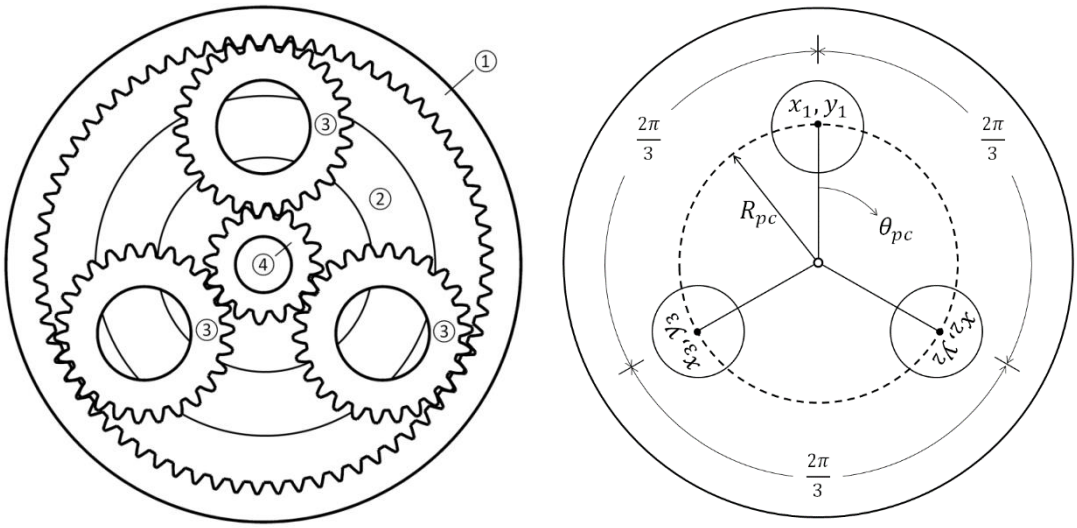


Fig. 3.11 Fixed-ring planetary gear set and its schematic representation

Parameters	Sun	Planet	Ring
Number of teeth	21	39	99
Normal modulus [mm]	10	10	10
Helix angle [deg]	-7.5	7.5	-7.5
Normal pressure angle [deg]	20	20	20
Pitch diameter [mm]	211.81	393.36	998.53
Tip diameter [mm]	235.53	419.36	997.83
Root diameter [mm]	190.53	374.36	1042.8
Bore/outer diameter [mm]	150	290	1170
Radius of rack corner [mm]	0.5	0.5	0.5
Face width [mm]	250	250	250
Center distance [mm]	308		n.a.
Carrier torque [kNm]		360	

Table 3.2 Parameters of the planetary gear set

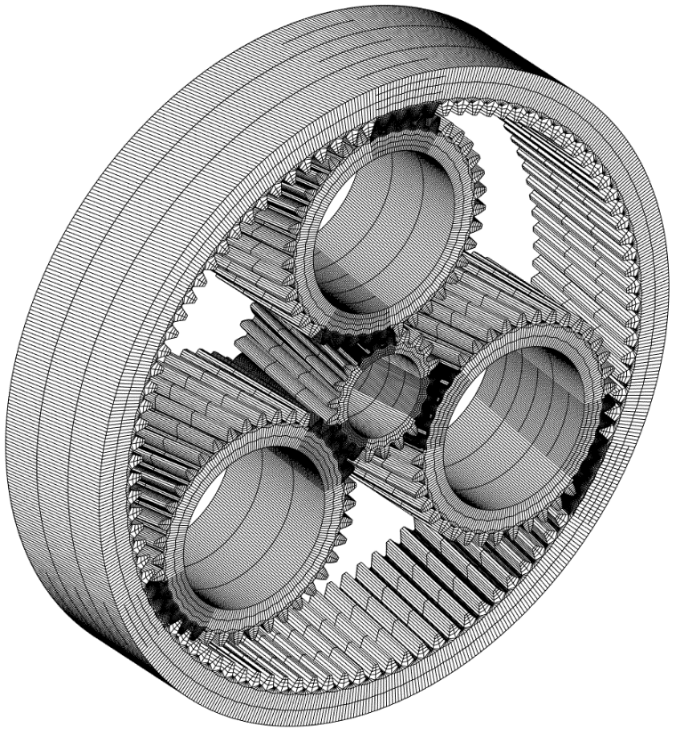


Fig. 3.12 Finite element model of the helical planetary gear set

3.3 Numerical validation of the reference gear contact model

In this final section, the CMS-based reference model is compared with three state-of-the-art gear contact models. An experimental validation of the gear contact model is beyond the scope of this dissertation, however good agreement with test-rig measurements found in literature [160] was demonstrated in an article that was co-authored by the author [161].

3.3.1 Comparison with nonlinear FE software

The primary means for validating the numerical accuracy of the reference model is offered by the non-linear FE package ABAQUS [162], which enables the dynamic simulation of gears using the same FE models as those that underlie the CMS-based reference model. The most important features of dynamic contact simulations in ABAQUS are summarized in Appendix N.3. Two dynamic simulations have been conducted with the parallel-axis gear pairs of Table 3.1, where the pinions rotate at 100 rad/s (≈ 1000 rpm). The comparison between the nonlinear FE model and the CMS-based reference model is based on the Transmission Error (TE) of the gear pair, which is a measure for the elastic deformation of the gears measured at the point of contact and is considered as the primary driver of gear noise and vibrations. The TE is defined as the difference between the actual position of the output gear and the position it would occupy if the geared transmission were perfect [163], and is obtained from

$$TE = \Delta\theta^1 + (z^1/z^2)\Delta\theta^2 \tag{3.3.1}$$

where $\Delta\theta^1$ and $\Delta\theta^2$ are the angular displacement of the pinion and wheel, respectively, and z^1 and z^2 are the respective teeth numbers.

In Fig. 3.13 and Fig. 3.14, the Dynamic TE (DTE) of the spur and helical gear pair as predicted by ABAQUS and the CMS-based reference model are shown. The relative errors between the two models are mainly explained by the difference in modal content and the numerical penalty factors used to enforce the contact constraints (see Appendix N.3). Overall, the CMS-based ROM is in good agreement with the ABAQUS model. The CPU times of the four simulations are shown in Table 3.3, where it should be noted that the CMS-based simulations were conducted in MATLAB, whereas ABAQUS is a highly optimized commercial software package. Nevertheless, the CMS-based model is about one order of magnitude faster than the nonlinear FE model in ABAQUS.

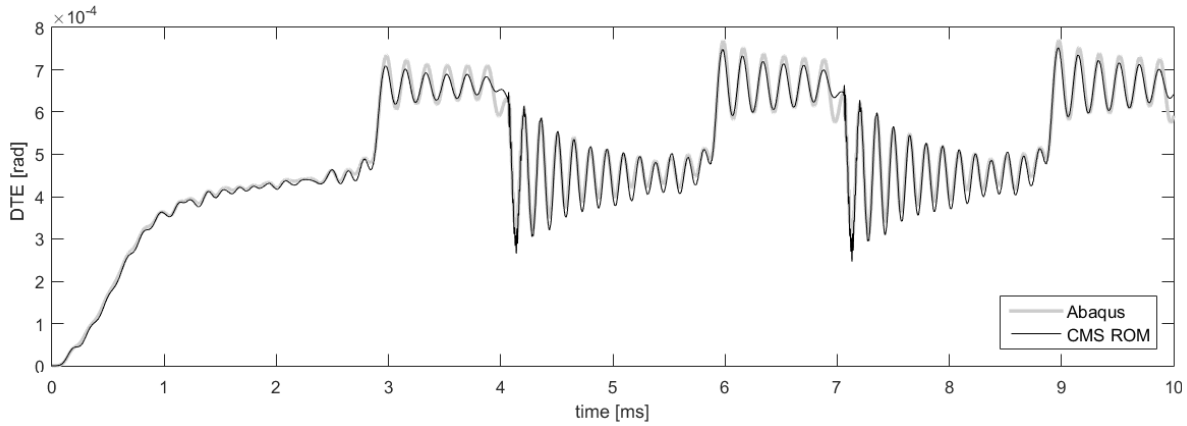


Fig. 3.13 Comparison of the spur gear pair DTE computed using Abaqus and the CMS-based reference model

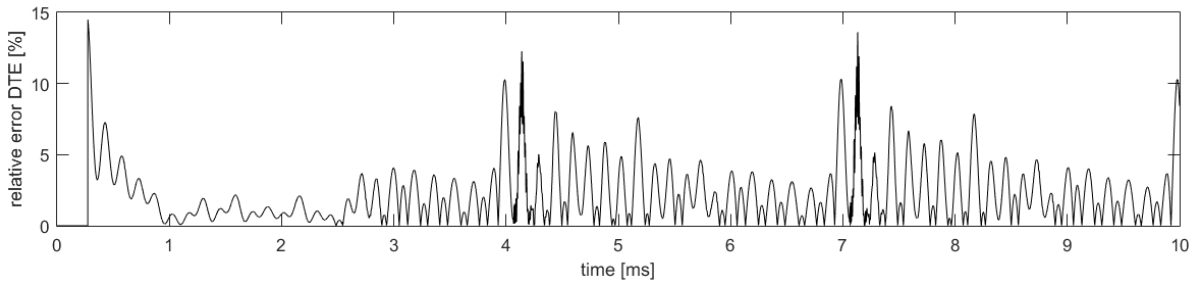


Fig. 3.13 (cont.) Comparison of the spur gear pair DTE computed using Abaqus and the CMS-based reference model

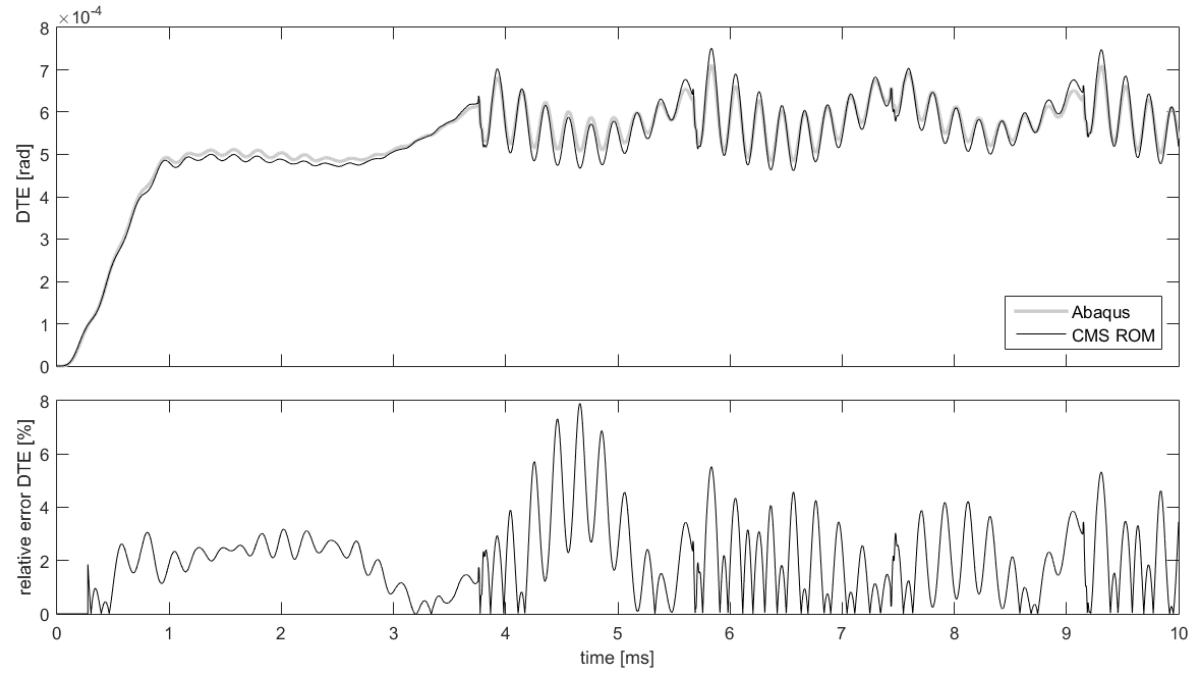


Fig. 3.14 Comparison of the helical gear pair DTE computed using Abaqus and the CMS-based reference model

	Spur		**Helical**	
	Abaqus	**CMS**	**Abaqus**	**CMS**
CPU time [s]	773662	94374	6045587	633024
Number of DOFs	329112	1728	993720	8742
Speed-up factor	1	8.2	1	9.5

Table 3.3 CPU times for Abaqus and the CMS-based reference model

3.3.2 Comparison with state-of-the-art gear contact models

This chapter is concluded with a comparison of the CMS-based reference model with two state-of-the-art gear contact models. The first model is a lumped-parameter model that is based on the gear tooth stiffness function of Kuang [13]. The model is discussed in detail in Appendix N.1 and is comparable to the standard gear contact models that are currently available in most MBSD software packages. The second gear contact model is based on the work of Andersson and Vedmar [19] and represents a compromise between lumped-parameter and distributed-parameter approaches. Its main features are the slicing of the gear width and the separation of local and global deformations, describing the former using classical contact theory and approximating the latter based on a priori FE simulations. This model is described in detail in Appendix N.2 and is comparable to the advanced gear contact models that are currently offered by a select number of MBSD software packages. Both models are based on analytical gear contact theory (see Appendix N) and assume that contact always occurs along the theoretical LOA. The influence of elastic tooth deformations on the location of the LOA is thus not included, nor is the coupling between adjacent teeth. Finally, as these models are quasi-static, the internal dynamics of the gears are neglected.

Fig. 3.15 and Fig. 3.16 the DTE (Eq. 3.3.1) of the spur and helical gear pair as predicted by the aforementioned state-of-the-art gear models and the CMS-based reference model are shown. Note that for simulating the helical gear pair, the Kuang model is coupled with a contact detection approach based on gear tooth slicing (see Fig. N.2). While the main features of the response are captured relatively well by all three models, the local differences between the FE-based reference model and the state-of-the-art gear contact models are considerable. For example, the influence of load on the the gear pair's contact ratio, which increases as the load increases, is only captured by the CMS-based reference model. While it is beyond the scope of this dissertation to investigate the region of applicability of the various state-of-the-art gear models, it is expected that these differences will increase when the gears misalign, transfer more torque, become more lightweight, and/or rotate at higher angular velocities. The CPU times of the various simulations are shown in Table 3.4[23], where speed-up factors of two-to-three orders of magnitude with respect to the CMS-based reference model are observed. While the assumptions underlying the models of Kuang and Andersson-Vedmar are not congruent with the first-principles approach that was promoted in the introduction of this dissertation, these models do set a benchmark for the ROMs-to-come in terms of computational efficiency.

[23] Note that these CPU times are only indicative, as the state-of-the-art gear contact models were implemented in MATLAB in a non-optimized way: for example, for the solution of the system of nonlinear equations that yields the slice contact forces, MATLAB's fsolve.m was used, which is optimized for robustness rather than for speed.

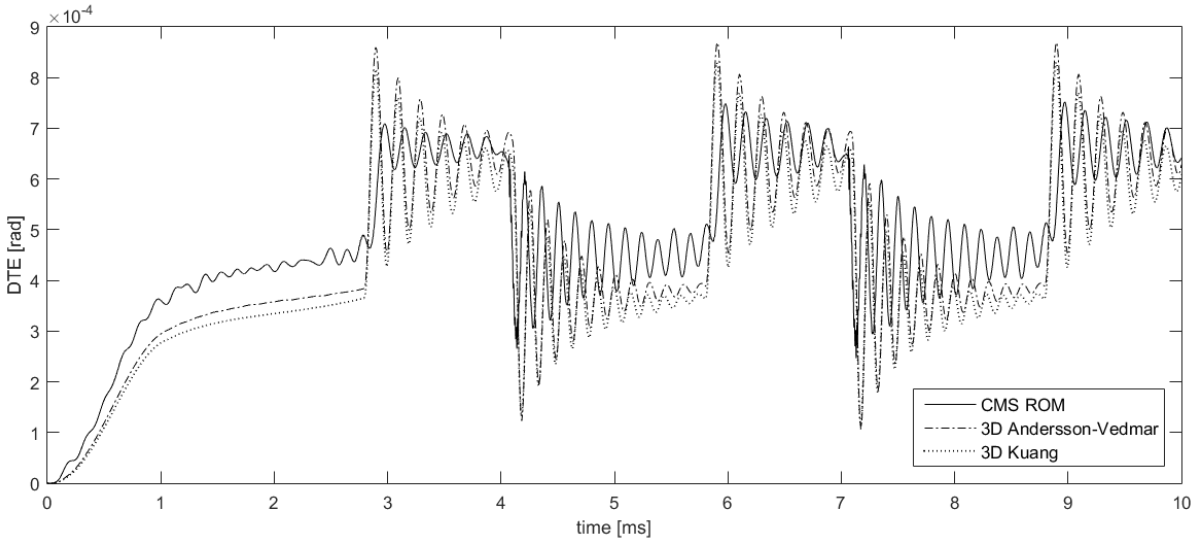


Fig. 3.15 Comparison of the spur gear pair DTE computed using the CMS-based reference model and the gear contact models of Andersson-Vedmar and Kuang

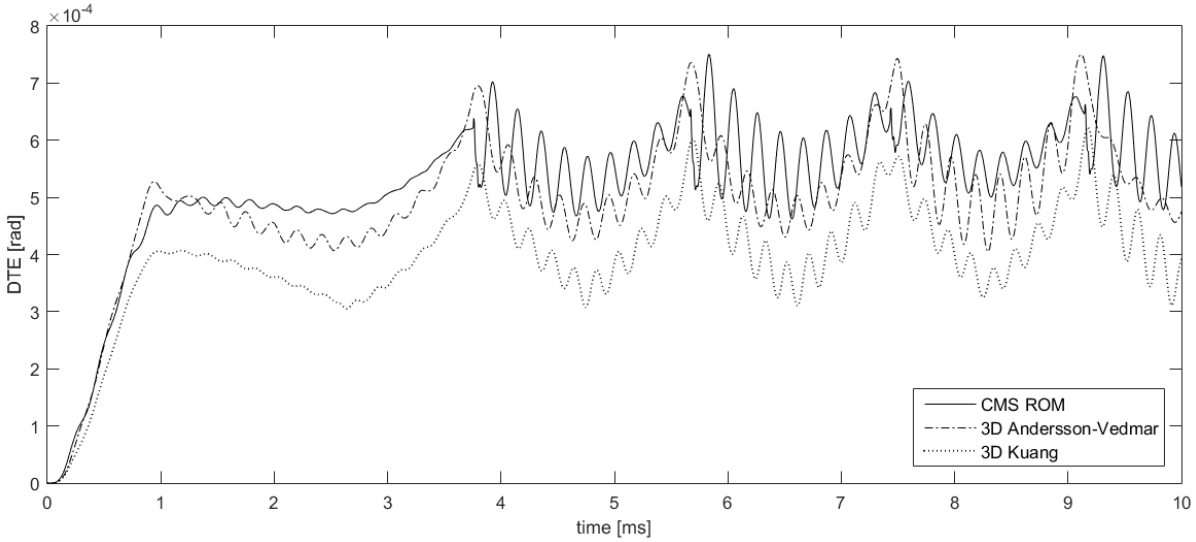


Fig. 3.16 Comparison of the helical gear pair DTE computed using the CMS-based reference model and the gear contact models of Andersson-Vedmar and Kuang

	Spur			**Helical**		
	CMS	**A-V**	**Kuang**	**CMS**	**A-V**	**Kuang**
CPU time [s]	94374	657	223	633024	1753	894
Number of DOFs	1728	2	2	8742	2	2
Speed-up factor	1	144	423	1	361	742

Table 3.4 CPU times for the CMS-based reference model and the gear contact models of Andersson-Vedmar (A-V) and Kuang

3.4 Summary and conclusion

After a concise introduction to the main concepts of involute gear geometry, the focus of this chapter turned to the dynamic gear contact model that is used as a reference throughout this dissertation. While it is referred to as the Full-Order Model (FOM) in subsequent chapters, the reference model is a Reduced-Order Model (ROM) itself, constructed using the Component Mode Synthesis (CMS) approach, which is considered as the state-of-the-art in flexible multibody dynamics. Using the concepts and equations developed in Chapter 2, a mathematical description of the CMS-based reference model was presented. The accuracy of the model was confirmed by comparison with commercial nonlinear FE software, as well as with two state-of-the-art gear contact models. While the CMS-based ROM proved to be substantially faster than the nonlinear FE model, the observed CPU times cannot be reconciled with today's objectives of Computer Aided Engineering (CAE, see Section 1.1), and more efficient Model-Order Reduction (MOR) techniques are needed.

Chapter 4

A novel model-order reduction method for dynamic gear contact problems

In this chapter a novel Model-Order Reduction (MOR) method is presented to efficiently solve contact problems in flexible multibody dynamics. The theoretical foundations of the method derive from the principles outlined in Chapter 2, most notably those underlying Parametric MOR (PMOR; see Section 2.3.2) and the Component Mode Synthesis (CMS) approach (see Section 2.3.3.1). In the current chapter, the method is applied to the analysis of perfectly aligned parallel-axis gears (see Chapter 3 for an in-depth description of the Full-Order Model (FOM)), with particular attention being paid to efficient computer implementation and numerical analysis. The extension of the method to the analysis of misaligned gears and planetary gear trains is dealt with in Chapter 6.

The structure of this chapter is as follows: Section 4.1 introduces the main concepts underlying the method and derives the equations of motion of the Reduced-Order Model (ROM). Section 4.2 deals with the efficient construction of the ROM during the *offline* phase of the simulation, while the numerical solution of the reduced-order equations of motion during the *online* phase is investigated in Section 4.3. Finally, the accuracy and efficiency of the novel MOR method is assessed in Section 4.4 by comparison with the selected reference model (see Section 3.2.3).

4.1 Model reduction based on interpolated contact shapes

4.1.1 Ingredients of the basis matrix

This work considers the dynamic analysis of a pair of gears that are spatially discretized using the Finite Element Method (FEM). It was shown in Section 3.2 that in typical gear contact problems the number N of nodal Degrees Of Freedom (DOFs) in the FE representations of the gears is of the order $N \approx 10^5 - 10^6$. In order to alleviate the computational burden associated with these large-scale FE models, the vector of nodal displacements of the i-th gear, denoted by $\boldsymbol{u}^i \in \mathbb{R}^N$, is approximated in a basis of reduced dimension as follows (see Section 2.3.1, Eq. 2.3.1):

$$\boldsymbol{u}^i \approx \boldsymbol{V}^i \boldsymbol{q}_f^i \tag{4.1.2}$$

where $\boldsymbol{V}^i \in \mathbb{R}^{N\times r}$ is the model's basis matrix and $\boldsymbol{q}_f^i \in \mathbb{R}^r$ is the vector of generalized elastic coordinates. The latter define the components of $\boldsymbol{u}^i$ in the subspace spanned by the columns of $\boldsymbol{V}^i$. The accuracy and effectiveness of Eq. 4.1.2 are determined by the choice of the basis matrix, which distinguishes one MOR technique from another (see Section 2.3).

The CMS approach was introduced in Section 2.3.3.1 as an effective and well-established means for reducing the dimensions of large-scale FE models in structural dynamics, and was applied to the gear contact problem in Section 3.2.3. The basis matrix associated with this approach was defined as (see Eq. 3.2.7):

$$\boldsymbol{V}_{CMS}^i = [\boldsymbol{\Phi}^i \quad \boldsymbol{\Psi}_a^i] \in \mathbb{R}^{N\times(n_k+n_a)} \tag{4.1.3}$$

where the submatrix $\boldsymbol{\Phi}^i \in \mathbb{R}^{N\times n_k}$ represents the set of n_k dynamically reacting eigenvectors, and the submatrix $\boldsymbol{\Psi}_a^i \in \mathbb{R}^{N\times n_a}$ contains one attachment mode for each DOF that can be loaded in the course of the simulation. It was shown in Section 3.2.3 that in gear contact analysis the number n_a of attachment modes in $\boldsymbol{\Psi}_a^i$ is of the order $n_a \approx 10^3 - 10^4$. While this represents a reduction in the number of DOFs by roughly two orders of magnitude as compared to the FOM, the numerical experiments conducted in Section 3.3.1 demonstrated that these numbers are not sufficient for meeting the objectives of this dissertation. For dynamic gear contact analyses to become computationally feasible, a further reduction of the number of DOFs by another two orders of magnitude is desirable.

The current work proposes to modify the traditional CMS approach by replacing the large set of attachment modes $\boldsymbol{\Psi}_a^i$ by a comparatively small set of *contact shapes* $\boldsymbol{\Psi}_c^i \in \mathbb{R}^{N\times n_c}$, where $n_c \ll n_a$. A contact shape is defined as the deformation pattern of a gear obtained from a static contact analysis of the gear pair, constrained at a certain configuration and subjected to certain external forces. In this dissertation, the gear contact shapes associated with an arbitrary configuration *j* are obtained as follows. The driven gear (denoted by the superscript '*2*') is constrained at the angular position θ_j^2 and the angle of the driving gear θ^1 is adjusted accordingly such that the two gears are in contact. An external torque T^1 is then applied to the driving gear and the following system of equations is solved for θ_j^1, $\boldsymbol{\psi}_{c,j}^1$, and $\boldsymbol{\psi}_{c,j}^2$:

$$\begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{K}_{FE}^1 & \mathbf{0} \\ sym. & & \boldsymbol{K}_{FE}^2 \end{bmatrix} \begin{Bmatrix} \theta_j^1 \\ \boldsymbol{\psi}_{c,j}^1 \\ \boldsymbol{\psi}_{c,j}^2 \end{Bmatrix} = \begin{Bmatrix} T^1 \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}_{c,\theta}^1 \\ \boldsymbol{f}_{c,f}^1 \\ \boldsymbol{f}_{c,f}^2 \end{Bmatrix} \tag{4.1.4}$$

where $\boldsymbol{K}_{FE}^1$ and $\boldsymbol{K}_{FE}^2$ are the gears' FE stiffness matrices and $\boldsymbol{f}_c^i$ is the vector of generalized contact forces acting on the *i*-th gear. The solution vectors $\boldsymbol{\psi}_{c,j}^1$ and $\boldsymbol{\psi}_{c,j}^2$ represent the deformation

patterns of the respective gears at static equilibrium. These vectors are referred to as the contact shapes of the gear pair at the configuration (θ_j^1, θ_j^2) subjected to the external torques $(T^1, T^2 = uT^1)$, where u is the transmission ratio of the gear pair. The procedure outlined above is repeated for a number of gear pair configurations (n_c in total) so as to obtain a sufficiently rich reduction basis. The associated basis matrix can be written as:

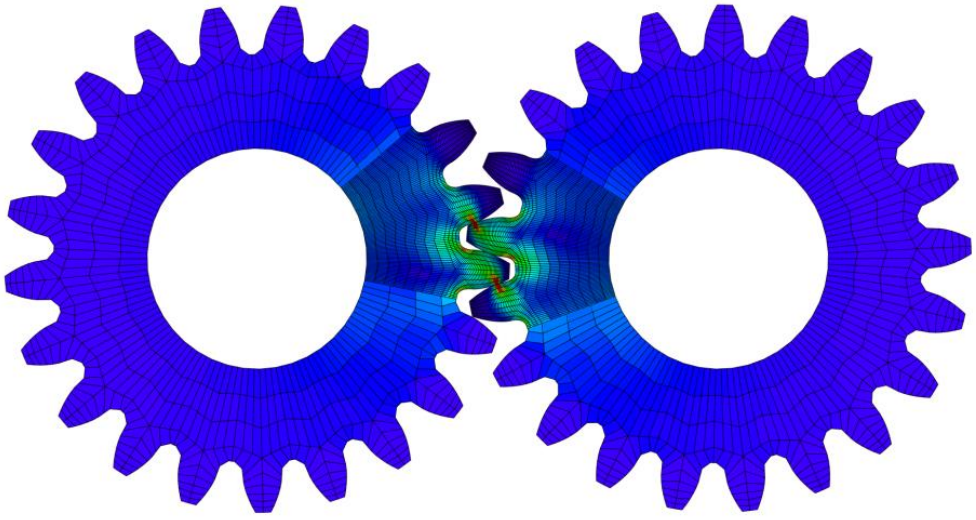
$$\boldsymbol{V}_{CS}^i = [\boldsymbol{\Phi}^i \quad \boldsymbol{\Psi}_c^i] = \left[\boldsymbol{\Phi}^i \quad \boldsymbol{\psi}_{c,1}^i \quad \boldsymbol{\psi}_{c,2}^i \quad \cdots \quad \boldsymbol{\psi}_{c,n_c}^i\right] \in \boldsymbol{R}^{N\times(n_k+n_c)} \tag{4.1.5}$$

where the subscript '*CS*' in $\boldsymbol{V}_{CS}^i$ refers to the Contact Shapes approach. Note that each contact shape in $\boldsymbol{\Psi}_c^i$ can be made mass and stiffness orthogonal with respect to the set of kept eigenmodes $\boldsymbol{\Phi}^i$ using Algorithm 2.1. In doing so, the following identities apply:

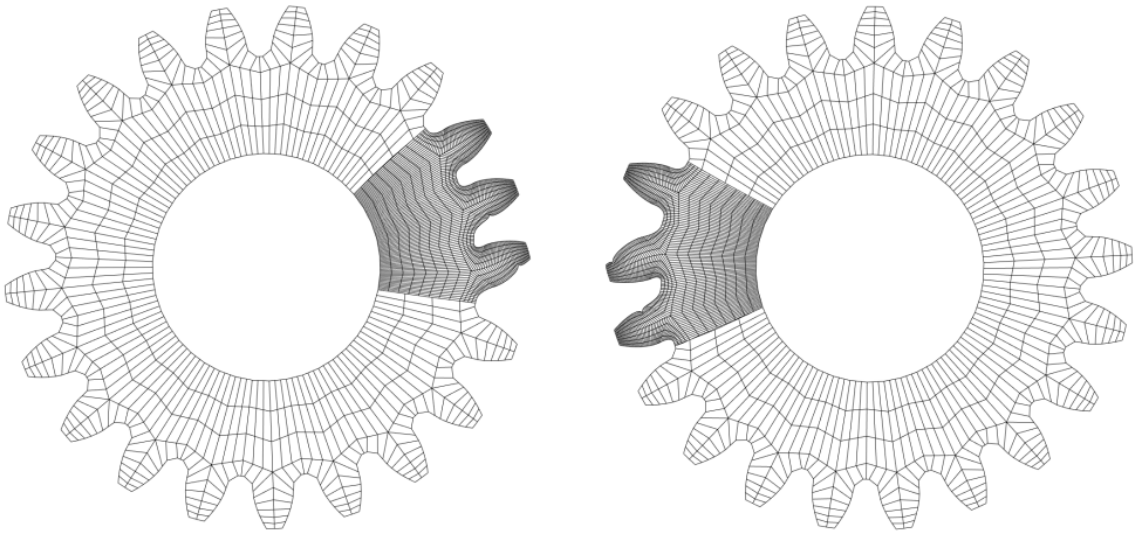
$$\boldsymbol{\Phi}^{i^T}\boldsymbol{M}_{FE}^i\boldsymbol{\Psi}_c^i = \boldsymbol{0} \quad , \quad \boldsymbol{\Phi}^{i^T}\boldsymbol{K}_{FE}^i\boldsymbol{\Psi}_c^i = \boldsymbol{0} \tag{4.1.6}$$

where $\boldsymbol{M}_{FE}^i$ is the FE mass matrix of the *i*-th gear.

As an illustration of the procedure outlined above, Fig. 4.a shows the static equilibrium solution of a spur gear pair at an arbitrary angular configuration. The Von Mises stresses in the gears are displayed so as to indicate the zones of contact. The corresponding contact shapes of the respective gears are shown in Fig. 4.b-c, where the nodal displacements have been magnified 500 times for clarity of illustration. Note that the total tooth deformation is a combination of tooth bending and local surface deformation.



(a)



(b) (c)

Fig. 4.1 Static contact equilibrium (a) and corresponding contact shapes (b-c) of a spur gear pair

4.1.2 Interpolation of the contact shapes

The reduction matrix $\boldsymbol{V}_{CS}^{i}$ of the CS approach (Eq. 4.1.5) consists of n_c contact shapes $\boldsymbol{\psi}_{c,j}^{i}$ that are each associated with a unique gear pair configuration *j*. Instead of projecting the vector of nodal displacements $\boldsymbol{u}^{i}$ on the subspace spanned by the constant basis matrix $\boldsymbol{V}_{CS}^{i}$, a more efficient reduction scheme is obtained when $\boldsymbol{V}_{CS}^{i}$ is continuously adapted based on the current gear pair configuration. In the *Interpolated Contact Shapes* (ICS) approach, the matrix of n_c contact shapes $\boldsymbol{\Psi}_{c}^{i}$ is replaced by a single *interpolated* contact shape $\boldsymbol{\psi}_{c}^{i}$ using the angular position of the driving gear θ^{1} as the interpolation parameter:

$$\boldsymbol{\psi}_{c}^{i} = \boldsymbol{\psi}_{c}^{i}(\theta^{1}) = intp(\theta^{1}, \theta_{1}^{1} \rightarrow \boldsymbol{\psi}_{c,1}^{i}, \theta_{2}^{1} \rightarrow \boldsymbol{\psi}_{c,2}^{i}, \dots, \theta_{n_c}^{1} \rightarrow \boldsymbol{\psi}_{c,n_c}^{i}) \in \mathbb{R}^{N\times 1} \tag{4.1.7}$$

where the operator intp( ) is defined in Section 2.3.2.2. The basis matrix of the ICS approach can thus be written as:

$$\boldsymbol{V}_{ICS}^{i} = [\boldsymbol{\Phi}^{i} \quad \boldsymbol{\psi}_{c}^{i}] = [\boldsymbol{\Phi}^{i} \quad \boldsymbol{\psi}_{c}^{i}(\theta^{1})] \in \mathbb{R}^{N\times(n_k+1)} \tag{4.1.8}$$

Note the matrix of kept eigenmodes $\boldsymbol{\Phi}^{i}$ is not affected by the interpolation approach and thus remains constant throughout the simulation.

The CS and ICS variants of the CMS approach can be considered as Parametric MOR (PMOR) techniques that obtain reduced-order representations of the full-order gear model that is parameterized by the angular configuration of the gear pair. In PMOR terminology (see Section 2.3.2), the CS approach is a *global* approach (Section 2.3.2.1) whereas the ICS approach is a *local* approach (Section 2.3.2.2). Turning to the field of Non-Linear MOR (NLMOR), the ICS approach

has certain characteristics in common with the Global Modal Parameterization (GMP) approach (see Section 2.3.3): while the assumed mode shapes in $\boldsymbol{V}_{ICS}^{i}$ are *component* modes (instead of *system* modes as in GMP), the two techniques rely on the interpolation of mode shapes based on the rigid-body configuration of the multibody system. Whether or not the ICS approach should be classified as a PMOR or NLMOR technique is a semantic discussion.

In order to evaluate Eq. 4.1.7, an interpolation scheme for obtaining the interpolated contact shape(s) must be selected. Valid choices are polynomial or spline interpolation of arbitrary order, possibly preceded by the transformation of the interpolated quantities onto a suitable manifold (see Section 2.3.2.2). A trade-off should be made between accuracy, complexity, and the number of gear pair configurations sampled (the latter is discussed in detail in Section 4.1.4). As will be demonstrated in Section 4.1.3, the order of the interpolating function determines the complexity of the system matrices and the number of mass invariants to be stored. Furthermore, it is desirable to preserve orthogonality of the interpolated contact shapes with respect to the matrix of kept eigenmodes (see Eq. 4.1.6). As a piecewise-linear interpolation of the contact shapes preserves orthogonality and leads to relatively simple expressions for the interpolated system matrices, this scheme will be investigated first. As a trade-off, a relatively large number of gear pair configurations must be sampled during the offline phase of the simulation. In Chapter 5, an advanced interpolation scheme that is closer to the physics of the gear contact problem and requires less parameter samples will be examined.

In a linear interpolation scheme, the interpolated contact shape at an arbitrary gear pair configuration is obtained as follows. Let $\boldsymbol{\psi}_{c,j}^{i}$ and $\boldsymbol{\psi}_{c,j+1}^{i}$ denote the contact shapes corresponding to the gear pair configurations defined by $\theta^1 = \theta_j^1$ and $\theta^1 = \theta_{j+1}^1$, respectively; then the interpolated contact shape at an intermediate angle θ^1, where $\theta_j^1 \geq \theta^1 \geq \theta_{j+1}^1$, is given by the following linear interpolation formula:

$$\boldsymbol{\psi}_c^i(\theta^1) = (1-p)\boldsymbol{\psi}_{c,j}^i + p\boldsymbol{\psi}_{c,j+1}^i \tag{4.1.9}$$

where p is a parameter depending solely on the angular position of the driving gear. The relation between p and θ^1 is linear and can be written explicitly as

$$p = p(\theta^1) = \frac{\theta^1 - \theta_j^1}{\theta_{j+1}^1 - \theta_j^1} = \frac{\theta^1 - \theta_j^1}{\Delta\theta_j^1}\,, \qquad 0 \leq p \leq 1 \tag{4.1.10}$$

where the angular increment $\Delta\theta_j^1$ is defined as $\Delta\theta_j^1 = \theta_{j+1}^1 - \theta_j^1$. The linear interpolation of contact shapes is graphically represented in Fig. 4.2a (for the full gear contact shapes) and **Fig. 4.2**b (for the finely discretized teeth only). Note that for the purpose of illustration the angular increment $\Delta\theta_j^1$ used in this figure is considerably larger than what is recommended.

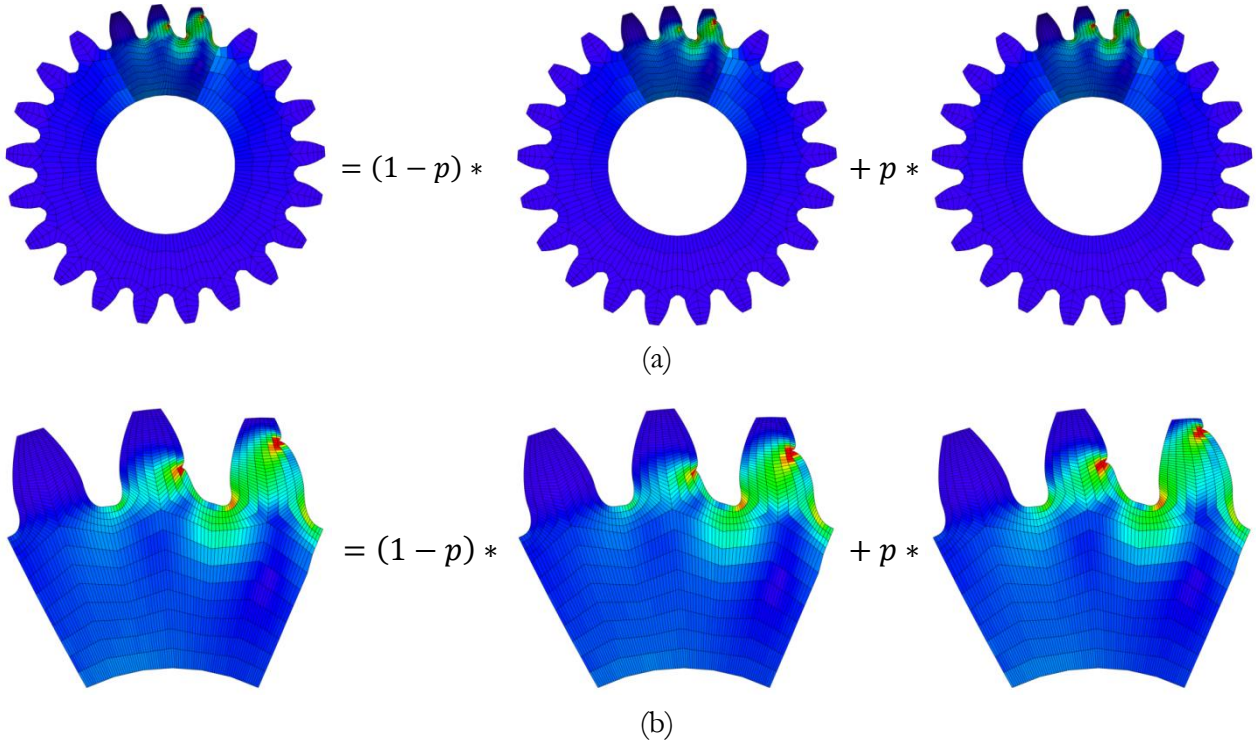


Fig. 4.2 Graphical representation of the linear interpolation of contact shapes

The interpolated contact shape $\boldsymbol{\psi}_c^i(\theta^1)$ is a function of the time-dependent reference angle $\theta^1 = \theta^1(t)$ (see Eq. 4.1.9). Therefore, the first and second derivatives of $\boldsymbol{\psi}_c^i(\theta^1)$ with respect to time are non-zero:

$$\dot{\boldsymbol{\psi}}_c^i = \frac{\partial \boldsymbol{\psi}_c^i}{\partial \theta^1}\dot{\theta}^1 = \boldsymbol{\psi}_\theta^i \dot{\theta}^1 \tag{4.1.11a}$$

$$\ddot{\boldsymbol{\psi}}_c^i = \frac{\partial^2 \boldsymbol{\psi}_c^i}{\partial (\theta^1)^2}\left(\dot{\theta}^1\right)^2 + \frac{\partial \boldsymbol{\psi}_c^i}{\partial \theta^1}\ddot{\theta}^1 = \boldsymbol{\psi}_{\theta\theta}^i\left(\dot{\theta}^1\right)^2 + \boldsymbol{\psi}_\theta^i \ddot{\theta}^1 \tag{3.2.2b}$$

where $\boldsymbol{\psi}_\theta^i = \partial \boldsymbol{\psi}_c^i / \partial \theta^1$. Assuming linear interpolation of the contact shapes, the first term in Eq. 3.2.2b vanishes and one obtains $\ddot{\boldsymbol{\psi}}_c^i = \boldsymbol{\psi}_\theta^i \ddot{\theta}^1$. Substitution of Eq. 4.1.9-4.1.10 into the above equations yields:

$$\dot{\boldsymbol{\psi}}_c^i = \boldsymbol{\psi}_\theta^i \dot{\theta}^1 = \dot{\theta}^1\left(\boldsymbol{\psi}_{c,j+1}^i - \boldsymbol{\psi}_{c,j}^i\right)\left(\Delta\theta_j^1\right)^{-1} = \dot{\theta}^1 \widehat{\boldsymbol{\psi}}_{c,j}^i \left(\Delta\theta_j^1\right)^{-1} \tag{4.1.12a}$$

$$\ddot{\boldsymbol{\psi}}_c^i = \boldsymbol{\psi}_\theta^i \ddot{\theta}^1 = \ddot{\theta}^1\left(\boldsymbol{\psi}_{c,j+1}^i - \boldsymbol{\psi}_{c,j}^i\right)\left(\Delta\theta_j^1\right)^{-1} = \ddot{\theta}^1 \widehat{\boldsymbol{\psi}}_{c,j}^i \left(\Delta\theta_j^1\right)^{-1} \tag{4.1.12b}$$

where $\widehat{\boldsymbol{\psi}}_{c,j}^i = \boldsymbol{\psi}_{c,j+1}^i - \boldsymbol{\psi}_{c,j}^i$ measures the difference between the contact shapes corresponding to θ_{j+1}^1 and θ_j^1, respectively.

In summary, the basis matrix of the ICS approach is composed of a constant set of eigenvectors $\boldsymbol{\Phi}^i$ and a configuration-dependent set of interpolated contact shapes $\boldsymbol{\psi}_c^i(\theta^1)$:

$$\boldsymbol{V}_{ICS}^{i}(\theta^1) = [\boldsymbol{\Phi}^i \quad \boldsymbol{\psi}_c^i(\theta^1)] = \left[\boldsymbol{\Phi}^i \quad \left(1 - p(\theta^1)\right)\boldsymbol{\psi}_{c,j}^i + p(\theta^1)\boldsymbol{\psi}_{c,j+1}^i\right] \in \boldsymbol{R}^{N\times(n_k+1)} \tag{4.1.13}$$

Furthermore, the first and second time derivatives of this matrix are non-zero:

$$\dot{\boldsymbol{V}}_{ICS}^{i}(\theta^1) = \frac{\partial}{\partial\theta^1}\left(\boldsymbol{V}_{ICS}^{i}\right)\dot{\theta}^1 = \dot{\theta}^1\boldsymbol{V}_{\theta}^{i} = \dot{\theta}^1\left[\boldsymbol{0} \quad \widehat{\boldsymbol{\psi}}_{c,j}^{i}\left(\Delta\theta_j^1\right)^{-1}\right] \tag{4.1.14a}$$

$$\ddot{\boldsymbol{V}}_{ICS}^{i}(\theta^1) = \frac{\partial}{\partial\theta^1}\left(\boldsymbol{V}_{ICS}^{i}\right)\ddot{\theta}^1 = \ddot{\theta}^1\boldsymbol{V}_{\theta}^{i} = \ddot{\theta}^1\left[\boldsymbol{0} \quad \widehat{\boldsymbol{\psi}}_{c,j}^{i}\left(\Delta\theta_j^1\right)^{-1}\right] \tag{4.1.14b}$$

The above expressions will give rise to additional terms in the kinetic energy of the reduced-order multibody model (see Eq. 2.4.3), as will be further elaborated on in the next section.

4.1.3 Derivation of the parameterized equations of motion

The equations of motion of reduced-order flexible multibody systems were derived in Section 2.4 and were applied to the dynamic gear contact problem in Section 3.2. One important assumption of the derivation presented in Section 2.4 is that the basis matrices $\boldsymbol{V}^i$ of the reduced-order models remain constant, hence $\dot{\boldsymbol{V}}^i = \ddot{\boldsymbol{V}}^i = \boldsymbol{0}$. For the CS variant of the CMS approach (see Eq. 4.1.5), this assumption holds and the equations presented in Sections 2.4 and 3.2 remain valid. In the ICS approach, however, the derivatives of the basis matrix are non-zero (see Eq. 4.1.14) and give rise to additional terms in the expressions for the kinetic and potential energies of the reduced-order models. In this section the derivation of Section 2.4 is extended to include these extra terms, thereby ensuring the energy-consistent usage of Lagrange's equations of motion (Eq. 2.4.15). As this chapter deals with perfectly aligned gears that rotate about a fixed axis in space, the equations of Section 2.4.3 'Rigid-body motion in a plane' will be dealt with here, assuming further that the linear interpolation scheme of Eq. 4.1.9 is used to form the interpolated contact shapes.

4.1.3.1 Kinematics of a material point

Taking into account Eq. 4.1.13, the global position vector of a node *j* on the *i*-th gear can be written as (see Eq. 2.4.8):

$$\begin{aligned}\boldsymbol{r}^{ij} = \boldsymbol{R}^i + \boldsymbol{A}^i\bar{\boldsymbol{r}}^{ij} &= \boldsymbol{R}^i + \boldsymbol{A}^i\left(\bar{\boldsymbol{r}}_0^{ij} + \bar{\boldsymbol{u}}^{ij}\right) \approx \boldsymbol{R}^i + \boldsymbol{A}^i\left(\bar{\boldsymbol{r}}_0^{ij} + \bar{\boldsymbol{V}}^{ij}\boldsymbol{q}_f^i\right) \\ &= \boldsymbol{R}^i + \boldsymbol{A}^i\left(\theta^i\right)\left(\bar{\boldsymbol{r}}_0^{ij} + \left[\bar{\boldsymbol{\Phi}}^{ij} \quad \bar{\boldsymbol{\psi}}_c^{ij}(\theta^1)\right]\boldsymbol{q}_f^i\right)\end{aligned} \tag{4.1.15}$$

where, as before, the bar sign ($\bar{}$) is used to denote quantities that are expressed with respect to the Floating Frame of Reference (FFR). Note that the subscript *'ICS'* was dropped from $\bar{\boldsymbol{V}}^{ij}$ for clarity of notation, keeping in mind however that the basis matrix of Eq. 4.1.13 is referred to here and in the following. The derivative of this expression with respect to time is obtained using Eq. 4.1.14a:

$$\dot{\boldsymbol{r}}^{ij} = \dot{\theta}^i \boldsymbol{B}^i \bar{\boldsymbol{r}}^{ij} + \boldsymbol{A}^i \left(\dot{\bar{\boldsymbol{V}}}^{ij} \boldsymbol{q}_f^i + \bar{\boldsymbol{V}}^{ij} \dot{\boldsymbol{q}}_f^i \right) = \dot{\theta}^i \boldsymbol{B}^i \bar{\boldsymbol{r}}^{ij} + \boldsymbol{A}^i \left(\dot{\theta}^1 \bar{\boldsymbol{\psi}}_\theta^{ij} \boldsymbol{q}_c^i + \bar{\boldsymbol{V}}^{ij} \dot{\boldsymbol{q}}_f^i \right) \tag{4.1.16}$$

where $\boldsymbol{q}_c^i$ represents the vector of generalized elastic coordinates corresponding to $\boldsymbol{\psi}_c^i(\theta^1)$. Note that since the FFR is rigidly connected to the center of the gear (see Section 3.2.2), the origin of the FFR remains fixed in space and thus $\dot{\boldsymbol{R}}^i = \boldsymbol{0}$. Eq. 4.1.16 can be written in matrix form as follows:

$$\dot{\boldsymbol{r}}^{ij} = \begin{bmatrix} \boldsymbol{B}^i \bar{\boldsymbol{r}}^{ij} + \boldsymbol{A}^i \bar{\boldsymbol{\psi}}_\theta^{ij} \boldsymbol{q}_c^i & \boldsymbol{A}^i \bar{\boldsymbol{V}}^{ij} \end{bmatrix} \begin{Bmatrix} \dot{\theta}^i \\ \dot{\boldsymbol{q}}_f^i \end{Bmatrix} = \boldsymbol{L}_v^{ij} \dot{\boldsymbol{q}}^i \tag{4.1.17}$$

where as in Eq. 3.2.6 the vector of generalized coordinates of the *i*-th gear is defined as $\boldsymbol{q}^{i^T} = \begin{Bmatrix} \theta^i & \boldsymbol{q}_f^{i^T} \end{Bmatrix}$.

4.1.3.2 Kinetic and elastic potential energy of a flexible gear

The kinetic energy T^i of the gear is defined as the sum of the nodal kinetic energies T^{ij}, i.e.

$$T^i = \sum_{j=1}^{n_n} T^{ij} = \sum_{j=1}^{n_n} \frac{1}{2} \left(\boldsymbol{L}_v^{ij} \dot{\boldsymbol{q}}^i \right)^T m^{ij} \left(\boldsymbol{L}_v^{ij} \dot{\boldsymbol{q}}^i \right) = \frac{1}{2} \dot{\boldsymbol{q}}^{i^T} \boldsymbol{M}^i \dot{\boldsymbol{q}}^i \tag{4.1.18}$$

Neglecting terms that are quadratic in the elastic coordinates $\boldsymbol{q}_f^i$, the gear's mass matrix $\boldsymbol{M}^i$ is obtained as

$$\boldsymbol{M}^i = \sum_{j=1}^{n_n} \boldsymbol{L}_v^{ij^T} m^{ij} \boldsymbol{L}_v^{ij} = \sum_{j=1}^{n_n} m^{ij} \begin{bmatrix} \bar{\boldsymbol{r}}^{ij^T} \bar{\boldsymbol{r}}_{xy}^{ij} + 2\bar{\boldsymbol{r}}^{ij^T} \bar{\boldsymbol{\psi}}_{\theta,y-x}^i \boldsymbol{q}_c^i & \bar{\boldsymbol{r}}^{ij^T} \bar{\boldsymbol{V}}_{y-x}^{ij} + \boldsymbol{q}_c^{i^T} \bar{\boldsymbol{\psi}}_\theta^{i^T} \bar{\boldsymbol{V}}^{ij} \\ sym. & \bar{\boldsymbol{V}}^{ij^T} \bar{\boldsymbol{V}}^{ij} \end{bmatrix} \tag{4.1.19}$$

where the subscripts '*xy*' and '*y-x*' are used as introduced in Eq. 2.4.48. Note that in comparison to Eq. 2.4.42, some additional terms appear in the mass matrix; these are due to the fact that the basis matrix depends on the angular configuration of the gear pair, and are indicated in gray.

The potential energy U^i of the gear is obtained as in Eq. 2.4.13, which now takes the following form:

$$U^i = \frac{1}{2} \boldsymbol{q}_f^{i^T} \boldsymbol{K}_{VV}^i \boldsymbol{q}_f^i = \frac{1}{2} \boldsymbol{q}_f^{i^T} \bar{\boldsymbol{V}}^{i^T} \boldsymbol{K}_{FE}^i \bar{\boldsymbol{V}}^i \boldsymbol{q}_f^i = \frac{1}{2} \boldsymbol{q}_f^{i^T} \begin{bmatrix} \bar{\boldsymbol{\Phi}}^{i^T} \boldsymbol{K}_{FE}^i \bar{\boldsymbol{\Phi}}^i & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{\psi}}_c^{i^T} \boldsymbol{K}_{FE}^i \bar{\boldsymbol{\psi}}_c^i \end{bmatrix} \boldsymbol{q}_f^i \tag{4.1.20}$$

As $\boldsymbol{\Phi}^i$ is composed of mass-orthonormalized eigenvectors, the first submatrix in the above equation is a diagonal matrix whose entries are the squares of the natural frequencies of the

eigenvectors in $\boldsymbol{\Phi}^i$. The second submatrix in the above equation can be expressed in terms of the parameter p (see Eq. 4.1.10) by substitution of Eq. 4.1.13:

$$\begin{aligned}\overline{\boldsymbol{\psi}}_c^{i\,T}\boldsymbol{K}_{FE}^i\overline{\boldsymbol{\psi}}_c^i &= \left[(1-p)\overline{\boldsymbol{\psi}}_{c,j}^i + p\overline{\boldsymbol{\psi}}_{c,j+1}^i\right]^T\boldsymbol{K}_{FE}^i\left[(1-p)\overline{\boldsymbol{\psi}}_{c,j}^i + p\overline{\boldsymbol{\psi}}_{c,j+1}^i\right]\\ &= (1-p)^2\overline{\boldsymbol{\psi}}_{c,j}^{i\,T}\boldsymbol{K}_{FE}^i\overline{\boldsymbol{\psi}}_{c,j}^i\\ &\quad + p(1-p)\left(\overline{\boldsymbol{\psi}}_{c,j}^{i\,T}\boldsymbol{K}_{FE}^i\overline{\boldsymbol{\psi}}_{c,j+1}^i + \overline{\boldsymbol{\psi}}_{c,j+1}^{i\,T}\boldsymbol{K}_{FE}^i\overline{\boldsymbol{\psi}}_{c,j}^i\right)\\ &\quad + p^2\overline{\boldsymbol{\psi}}_{c,j+1}^{i\,T}\boldsymbol{K}_{FE}^i\overline{\boldsymbol{\psi}}_{c,j+1}^i\end{aligned} \tag{4.1.21}$$

Note that the linear interpolation of contact shapes results in a quadratic interpolation of the corresponding reduced stiffness matrices. Eq. 4.1.20 can be written in terms of the full vector of generalized coordinates as follows:

$$U^i = \frac{1}{2}\boldsymbol{q}^{i\,T}\boldsymbol{K}^i\boldsymbol{q}^i = \frac{1}{2}\boldsymbol{q}^{i\,T}\begin{bmatrix}0 & \boldsymbol{0}\\ \boldsymbol{0} & \boldsymbol{K}_{VV}^i\end{bmatrix}\boldsymbol{q}_f^i = \frac{1}{2}\boldsymbol{q}^{i\,T}\begin{bmatrix}0 & \boldsymbol{0}\\ \boldsymbol{0} & \overline{\boldsymbol{V}}^{i\,T}\boldsymbol{K}_{FE}^i\overline{\boldsymbol{V}}^i\end{bmatrix}\boldsymbol{q}_f^i \tag{4.1.22}$$

where the gear's stiffness matrix is defined as $\boldsymbol{K}^i = diag\left(0, \boldsymbol{K}_{VV}^i\right)$.

4.1.3.3 Equations of motion of a flexible gear

The EOMs of a flexible gear having a configuration-dependent basis matrix (see Eq. 4.1.13) are obtained by application of Lagrange's unconstrained EOMs:

$$\frac{d}{dt}\left(\frac{\partial T^i}{\partial\dot{\boldsymbol{q}}^i}\right)^T - \left(\frac{\partial T^i}{\partial\boldsymbol{q}^i}\right)^T + \left(\frac{\partial U^i}{\partial\boldsymbol{q}^i}\right)^T = \boldsymbol{f}^i \tag{4.1.23}$$

where $\boldsymbol{f}^i$ is the vector of generalized external and contact forces. Using the general expression of the kinetic energy of Eq. 4.1.18, the first two terms on the left-hand side of Lagrange's equation can be written as

$$\frac{d}{dt}\left(\frac{\partial T^i}{\partial\dot{\boldsymbol{q}}^i}\right)^T - \left(\frac{\partial T^i}{\partial\boldsymbol{q}^i}\right)^T = \boldsymbol{M}^i\ddot{\boldsymbol{q}}^i - \boldsymbol{f}_v^i \tag{4.1.24}$$

where the quadratic velocity vector $\boldsymbol{f}_v^i$, which contains the gyroscopic and Coriolis force components, is defined as in Eq. 2.4.17, leading to

$$\boldsymbol{f}_v^i = -\dot{\boldsymbol{M}}^i\dot{\boldsymbol{q}}^i + \frac{1}{2}\left[\frac{\partial}{\partial\boldsymbol{q}^i}\left(\dot{\boldsymbol{q}}^{i\,T}\boldsymbol{M}^i\dot{\boldsymbol{q}}^i\right)\right] \tag{4.1.25}$$

$$= \sum_{j=1}^{n_n} m^{ij} \begin{Bmatrix} -\dot{\boldsymbol{q}}_f^{i\,T} \bar{\boldsymbol{V}}^{ij\,T} \left(2\bar{\boldsymbol{r}}_{xy}^{ij}\dot{\theta}^i + \bar{\boldsymbol{V}}_{y-x}^{ij}\dot{\boldsymbol{q}}_f^i\right) + 2\dot{\theta}^i \left(-\bar{\boldsymbol{r}}^{ij\,T} \bar{\boldsymbol{\psi}}_{\theta,y-x}^{i}\dot{\boldsymbol{q}}_c^i - \boldsymbol{q}_c^{i\,T} \bar{\boldsymbol{\psi}}_{\theta}^{i\,T} \left[\bar{\boldsymbol{V}}_{\theta}^{ij} - \bar{\boldsymbol{V}}_{y-x}^{ij}\right]\dot{\boldsymbol{q}}_f^i\right) \\ \dot{\theta}^i \bar{\boldsymbol{V}}^{ij\,T} \left(\bar{\boldsymbol{r}}_{xy}^{ij}\dot{\theta}^i + 2\bar{\boldsymbol{V}}_{y-x}^{ij}\dot{\boldsymbol{q}}_f\right) + 2\dot{\theta}^i \left[\bar{\boldsymbol{V}}^{i\,T} \left[\dot{\theta}^i \bar{\boldsymbol{\psi}}_{\theta,y-x}^{i}\boldsymbol{q}_c^i - \bar{\boldsymbol{\psi}}_{\theta}^{i}\dot{\boldsymbol{q}}_c^i\right] - \frac{1}{2}\bar{\boldsymbol{V}}_{\theta}^{ij\,T} \bar{\boldsymbol{V}}^{ij}\dot{\boldsymbol{q}}_f^i\right] \end{Bmatrix}$$

where as in Eq. 4.1.19, terms that are due to the configuration-dependency of the basis matrix are indicated in gray.

The third term on the left-hand side of Eq. 4.1.23 can be obtained by substitution of Eq. 4.1.20 as follows:

$$\left(\frac{\partial U^i}{\partial \boldsymbol{q}^i}\right)^T = \frac{1}{2}\left[\frac{\partial}{\partial \boldsymbol{q}^i}\left(\frac{1}{2}\boldsymbol{q}_f^{i\,T} \boldsymbol{K}_{VV}^i \boldsymbol{q}_f^i\right)\right] = \boldsymbol{K}^i\boldsymbol{q}^i + \begin{Bmatrix} \boldsymbol{q}_f^{i\,T} \bar{\boldsymbol{V}}^{i\,T} \boldsymbol{K}_{FE}^i \bar{\boldsymbol{V}}_{\theta}^i \boldsymbol{q}_f^i \\ \boldsymbol{0} \end{Bmatrix} \approx \boldsymbol{K}^i\boldsymbol{q}^i \tag{4.1.26}$$

where use was made of the assumption that terms that are quadratic in the vector of generalized elastic coordinates $\boldsymbol{q}_f^i$ can be neglected.

Substitution of Eq. 4.1.24 and Eq. 4.1.26 into Eq. 4.1.23 yields the familiar form of the flexible gear's equations of motion:

$$\boldsymbol{M}^i\ddot{\boldsymbol{q}}^i + \boldsymbol{K}^i\boldsymbol{q}^i = \boldsymbol{f}_e^i + \boldsymbol{f}_c^i + \boldsymbol{f}_v^i \tag{4.1.27}$$

where the mass matrix $\boldsymbol{M}^i$ is defined in Eq. 4.1.19, the quadratic velocity vector $\boldsymbol{f}_v^i$ is defined in Eq. 4.1.25, the vector of generalized external forces $\boldsymbol{f}_e^i$ collects the torques and forces applied to the gears, and $\boldsymbol{f}_c^i$ represents the vector of generalized contact forces.

4.1.3.4 Equations of motion in terms of mass invariants

As explained in Section 2.4.3 it is common practice in flexible multibody dynamics to formulate both the mass matrix $\boldsymbol{M}^i$ and quadratic velocity vector $\boldsymbol{f}_v^i$ in terms of a limited number of mass invariants, i.e. matrices that are computed prior to the actual simulation and that contain information regarding the mass distribution of the flexible bodies. As an illustration of this procedure in the case of parametrically reduced FEMs, the mass matrix of Eq. 4.1.19 is expressed here in terms of the gear's mass invariants. For a more elaborate expression of the equations of motion in terms of mass invariants, see Appendix D.

Neglecting terms that are quadratic in the displacements, the gear's mass moment of inertia is given by (see Eq. 4.1.19)

$$\begin{aligned} \boldsymbol{M}_{\theta\theta}^i &= \sum_{j=1}^{n_n} m^{ij}\left[\bar{\boldsymbol{r}}^{ij\,T}\bar{\boldsymbol{r}}_{xy}^{ij} + 2\bar{\boldsymbol{r}}^{ij\,T}\bar{\boldsymbol{\psi}}_{\theta,y-x}^{i}\boldsymbol{q}_c^i\right] \\ &= \sum_{j=1}^{n_n} m^{ij}\left[\left(\bar{\boldsymbol{r}}_0^{ij} + \bar{\boldsymbol{V}}^{ij}\boldsymbol{q}_f^i\right)^T\left(\bar{\boldsymbol{r}}_{0,xy}^{ij} + \bar{\boldsymbol{V}}_{xy}^{ij}\boldsymbol{q}_f^i\right) + 2\left(\bar{\boldsymbol{r}}_0^{ij} + \bar{\boldsymbol{V}}^{ij}\boldsymbol{q}_f^i\right)^T\bar{\boldsymbol{\psi}}_{\theta,y-x}^{i}\boldsymbol{q}_c^i\right] \end{aligned} \tag{4.1.28}$$

$$= \sum_{j=1}^{n_n} m^{ij} \Big[\left(\bar{\boldsymbol{r}}_0^{ij} + \boldsymbol{\Phi}^{ij}\boldsymbol{q}_n^i + \bar{\boldsymbol{\psi}}_c^i \boldsymbol{q}_c^i\right)^T \left(\bar{\boldsymbol{r}}_{0,xy}^{ij} + \boldsymbol{\Phi}_{xy}^{ij}\boldsymbol{q}_n^i + \bar{\boldsymbol{\psi}}_{c,xy}^i \boldsymbol{q}_c^i\right) + 2\left(\bar{\boldsymbol{r}}_0^{ij} + \boldsymbol{\Phi}^{ij}\boldsymbol{q}_n^i + + \bar{\boldsymbol{\psi}}_c^i \boldsymbol{q}_c^i\right)^T \bar{\boldsymbol{\psi}}_{\theta,y-x}^i \boldsymbol{q}_c^i \Big]$$

$$= \sum_{j=1}^{n_n} m^{ij} \left[\bar{\boldsymbol{r}}_0^{ijT} \bar{\boldsymbol{r}}_{0,xy}^{ij} + 2\bar{\boldsymbol{r}}_0^{ijT} \boldsymbol{\Phi}_{xy}^{ij} \boldsymbol{q}_n^i + 2\bar{\boldsymbol{r}}_0^{ijT} \bar{\boldsymbol{\psi}}_{c,xy}^i \boldsymbol{q}_c^i + 2\bar{\boldsymbol{r}}_0^{ijT} \bar{\boldsymbol{\psi}}_{\theta,y-x}^i \boldsymbol{q}_c^i \right]$$

Substitution of Eq. 4.1.9 into the above equation yields

$$\boldsymbol{M}_{\theta\theta}^i = \sum_{j=1}^{n_n} m^{ij} \Big[\bar{\boldsymbol{r}}_0^{ijT} \bar{\boldsymbol{r}}_{0,xy}^{ij} + 2\bar{\boldsymbol{r}}_0^{ijT} \boldsymbol{\Phi}_{xy}^{ij} \boldsymbol{q}_n^i + 2\bar{\boldsymbol{r}}_0^{ijT} \left[(1-p)\bar{\boldsymbol{\psi}}_{c,j}^i + p\bar{\boldsymbol{\psi}}_{c,j+1}^i\right]_{xy} \boldsymbol{q}_c^i + 2\bar{\boldsymbol{r}}_0^{ijT} \bar{\boldsymbol{\psi}}_{\theta,y-x}^i \boldsymbol{q}_c^i \Big] \tag{4.1.29}$$

Note that the summation is carried out over the node index *j* in the superscript, which is not to be confused with the configuration index *j* in the subscript. The above equation can be written in terms of mass invariants as follows

$$\boldsymbol{M}_{\theta\theta}^i = \boldsymbol{M}_2^i + 2\boldsymbol{M}_{3n}^i \boldsymbol{q}_n^i + 2\left[(1-p)\boldsymbol{M}_{3c}^i\{j\} + p\boldsymbol{M}_{3c}^i\{j+1\}\right]\boldsymbol{q}_c^i + 2\widetilde{\boldsymbol{M}}_{3c}^i\{j\}\boldsymbol{q}_c^i \tag{4.1.30}$$

where curled brackets $\{\cdot\}$ indicated the entries to be selected in the designated matrix, and where the mass invariants are defined as (compare with Appendix D.2)

$$\begin{aligned} \boldsymbol{M}_1^i &= \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{r}}_0^{ijT} \bar{\boldsymbol{r}}_{0,xy}^{ij} \in \mathbb{R}^{1\times 1} \qquad \boldsymbol{M}_{3n}^i = \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{r}}_0^{ijT} \bar{\boldsymbol{\Phi}}_{xy}^{ij} \in \mathbb{R}^{1\times n_k} \\ \boldsymbol{M}_{3c}^i &= \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{r}}_0^{ijT} \bar{\boldsymbol{\Psi}}_{c,xy}^i \in \mathbb{R}^{1\times n_c} \quad \widetilde{\boldsymbol{M}}_{3c}^i = \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{r}}_0^{ijT} \widehat{\bar{\boldsymbol{\Psi}}}_c^{ij} \left(\Delta\theta_j^1\right)^{-1} \in \mathbb{R}^{1\times(n_c-1)} \end{aligned} \tag{4.1.31}$$

The matrix $\widehat{\bar{\boldsymbol{\Psi}}}_c^i$ is the contact shapes difference matrix and is defined as

$$\widehat{\bar{\boldsymbol{\Psi}}}_c^i = \left[\left(\bar{\boldsymbol{\psi}}_{c,2}^i - \bar{\boldsymbol{\psi}}_{c,1}^i\right) \quad \left(\bar{\boldsymbol{\psi}}_{c,3}^i - \bar{\boldsymbol{\psi}}_{c,2}^i\right) \quad \dots \quad \left(\bar{\boldsymbol{\psi}}_{c,n_c}^i - \bar{\boldsymbol{\psi}}_{c,n_c-1}^i\right)\right] \in R^{N\times(n_c-1)} \tag{4.1.32}$$

The upper right term in Eq. 4.1.19, which represents the inertial coupling between the rigid body rotation and the elastic deformations of the gears, is given by

$$\begin{aligned} \boldsymbol{M}_{\theta f}^i &= \sum_{j=1}^{n_n} m^{ij} \left[\bar{\boldsymbol{r}}^{ijT} \bar{\boldsymbol{V}}_{y-x}^{ij} + \boldsymbol{q}_c^{iT} \bar{\boldsymbol{\psi}}_\theta^{iT} \bar{\boldsymbol{V}}^{ij}\right] \\ &= \sum_{j=1}^{n_n} m^{ij} \left[\left(\bar{\boldsymbol{r}}_0^{ij} + \bar{\boldsymbol{\Phi}}^{ij}\boldsymbol{q}_n^i + \bar{\boldsymbol{\psi}}_c^i \boldsymbol{q}_c^i\right)^T \left[\bar{\boldsymbol{\Phi}}_{y-x}^{ij} \quad \bar{\boldsymbol{\psi}}_{c,y-x}^i\right] + \boldsymbol{q}_c^{iT} \bar{\boldsymbol{\psi}}_\theta^{iT} \left[\bar{\boldsymbol{\Phi}}^{ij} \quad \bar{\boldsymbol{\psi}}_c^i\right]\right] \\ &= \sum_{j=1}^{n_n} m^{ij} \begin{bmatrix} \left(\bar{\boldsymbol{r}}_0^{ijT} + \boldsymbol{q}_n^{iT}\bar{\boldsymbol{\Phi}}^{ijT} + \boldsymbol{q}_c^{iT}\bar{\boldsymbol{\psi}}_c^{iT}\right)\bar{\boldsymbol{\Phi}}_{y-x}^{ij} & \left(\bar{\boldsymbol{r}}_0^{ijT} + \boldsymbol{q}_n^{iT}\bar{\boldsymbol{\Phi}}^{ijT} + \boldsymbol{q}_c^{iT}\bar{\boldsymbol{\psi}}_c^{iT}\right)\bar{\boldsymbol{\Psi}}_{c,y-x}^i \\ +\boldsymbol{q}_c^{iT}\bar{\boldsymbol{\psi}}_\theta^{iT}\bar{\boldsymbol{\Phi}}^{ij} & +\boldsymbol{q}_c^{iT}\bar{\boldsymbol{\psi}}_\theta^{iT}\bar{\boldsymbol{\psi}}_c^i \end{bmatrix} \end{aligned} \tag{4.1.33}$$

$$\boldsymbol{M}_{\theta f} = \begin{bmatrix} \boldsymbol{M}_{\theta n}^{i} & \boldsymbol{M}_{\theta c}^{i} \end{bmatrix}$$

where the submatrices and are defined by:

$$\boldsymbol{M}_{\theta n}^{i} = \sum_{j=1}^{n_n} m^{ij} \left[\left(\bar{\boldsymbol{r}}_0^{ij^T} + \boldsymbol{q}_n^{i\,T} \bar{\boldsymbol{\Phi}}^{ij^T} + \boldsymbol{q}_c^{i\,T} \bar{\boldsymbol{\psi}}_c^{ij^T} \right) \bar{\boldsymbol{\Phi}}_{y-x}^{ij} + \boldsymbol{q}_c^{i\,T} \bar{\boldsymbol{\psi}}_\theta^{ij^T} \bar{\boldsymbol{\Phi}}^{ij} \right] \tag{4.1.34a}$$

$$\boldsymbol{M}_{\theta c}^{i} = \sum_{j=1}^{n_n} m^{ij} \left[\left(\bar{\boldsymbol{r}}_0^{ij^T} + \boldsymbol{q}_n^{i\,T} \bar{\boldsymbol{\Phi}}^{ij^T} + \boldsymbol{q}_c^{i\,T} \bar{\boldsymbol{\psi}}_c^{ij^T} \right) \bar{\boldsymbol{\psi}}_{c,y-x}^{ij} + \boldsymbol{q}_c^{i\,T} \bar{\boldsymbol{\psi}}_\theta^{ij^T} \bar{\boldsymbol{\psi}}_c^{ij} \right] \tag{4.1.34b}$$

These matrices can be formulated in terms of mass invariants as follows

$$\boldsymbol{M}_{\theta n}^{i} = \boldsymbol{M}_{4n}^{i} + \boldsymbol{q}_n^{i\,T} \boldsymbol{M}_{7nn}^{i} + \boldsymbol{q}_c^{i\,T} \left[(1-p) \boldsymbol{M}_{7sn}^{i}\{j,:\} + p \boldsymbol{M}_{7sn}^{i}\{j+1,:\} \right] + \boldsymbol{q}_c^{i\,T} \widetilde{\boldsymbol{M}}_{7sn}^{i}\{j,:\} \tag{4.1.35a}$$

$$\begin{aligned} \boldsymbol{M}_{\theta c}^{i} = {} & \left[(1-p) \boldsymbol{M}_{4c}^{i}\{j\} + p \boldsymbol{M}_{4c}^{i}\{j+1\} \right] + \boldsymbol{q}_n^{i\,T} \left[(1-p) \boldsymbol{M}_{7nc}^{i}\{:,j\} + p \boldsymbol{M}_{7nc}^{i}\{:,j+1\} \right] \\ & + \boldsymbol{q}_c^{i\,T} \left[(1-p)^2 \boldsymbol{M}_{7cc}^{i}\{j,j\} + p^2 \boldsymbol{M}_{7cc}^{i}\{j+1,j+1\} + p(1 \right. \\ & \left. - p) \left(\boldsymbol{M}_{7cc}^{i}\{j+1,j\} + \boldsymbol{M}_{7cc}^{i}\{j,j+1\} \right) \right] \\ & + \boldsymbol{q}_c^{i\,T} \left[(1-p) \widetilde{\boldsymbol{M}}_{7cc}^{i}\{j,j\} + p \widetilde{\boldsymbol{M}}_{7cc}^{i}\{j,j+1\} \right] \end{aligned} \tag{4.1.35b}$$

where the mass invariants are defined as

$$\begin{aligned} \boldsymbol{M}_{4n}^{i} &= \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{r}}_0^{ij^T} \bar{\boldsymbol{\Phi}}_{y-x}^{ij} \in \mathbb{R}^{1 \times n_k} & \boldsymbol{M}_{4c}^{i} &= \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{r}}_0^{ij^T} \bar{\boldsymbol{\Psi}}_{c,y-x}^{ij} \in \mathbb{R}^{1 \times n_c} \\ \boldsymbol{M}_{7nn}^{i} &= \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{\Phi}}^{ij^T} \bar{\boldsymbol{\Phi}}_{y-x}^{ij} \in \mathbb{R}^{n_k \times n_k} & \boldsymbol{M}_{7nc}^{i} &= \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{\Phi}}^{ij^T} \bar{\boldsymbol{\Psi}}_{c,y-x}^{ij} \in \mathbb{R}^{n_k \times n_c} \\ \boldsymbol{M}_{7cn}^{i} &= \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{\Psi}}_c^{i\,T} \bar{\boldsymbol{\Phi}}_{y-x}^{ij} \in \mathbb{R}^{n_c \times n_k} & \boldsymbol{M}_{7cc}^{i} &= \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{\Psi}}_c^{ij^T} \bar{\boldsymbol{\Psi}}_{c,y-x}^{ij} \in \mathbb{R}^{n_c \times n_c} \\ \widetilde{\boldsymbol{M}}_{7cn}^{i} &= \sum_{j=1}^{n_n} m^{ij} \widehat{\bar{\boldsymbol{\Psi}}}_c^{ij^T} \boldsymbol{\Phi}^{ij} \left(\Delta\theta_j^1 \right)^{-1} & \widetilde{\boldsymbol{M}}_{7cc}^{i} &= \sum_{j=1}^{n_n} m^{ij} \widehat{\bar{\boldsymbol{\Psi}}}_c^{ij^T} \bar{\boldsymbol{\Psi}}_c^{ij} \left(\Delta\theta_j^1 \right)^{-1} \\ & \in \mathbb{R}^{(n_c-1) \times n_k} & & \in \mathbb{R}^{(n_c-1) \times n_c} \end{aligned} \tag{4.1.36}$$

Finally, the submatrix $\boldsymbol{M}_{ff}^{i}$, also referred to as the modal mass matrix, can be written as

$$\boldsymbol{M}_{ff}^{i} = \sum_{j=1}^{n_n} m^{ij} \left[\bar{\boldsymbol{V}}^{ij^T} \bar{\boldsymbol{V}}^{ij} \right] = \sum_{j=1}^{n_n} m^{ij} \begin{bmatrix} \bar{\boldsymbol{\Phi}}^{ij^T} \bar{\boldsymbol{\Phi}}^{ij} & \boldsymbol{0} \\ \boldsymbol{0} & \bar{\boldsymbol{\psi}}_c^{ij^T} \bar{\boldsymbol{\psi}}_c^{ij} \end{bmatrix} = \begin{bmatrix} \boldsymbol{M}_{nn}^{i} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{cc}^{i} \end{bmatrix} \tag{4.1.37}$$

where the submatrices $\boldsymbol{M}_{nn}^{i}$ and $\boldsymbol{M}_{cc}^{i}$ are defined as

$$\boldsymbol{M}_{nn}^{i} = \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{\Phi}}^{ij^T} \bar{\boldsymbol{\Phi}}^{ij} = \boldsymbol{I}_{n_k} \tag{4.1.38a}$$

$$\boldsymbol{M}_{cc}^{i} = \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{\psi}}_{c}^{ij^T} \bar{\boldsymbol{\psi}}_{c}^{ij} = \sum_{j=1}^{n_n} m^{ij} \left[(1-p)\bar{\boldsymbol{\psi}}_{c,j}^{i} + p\bar{\boldsymbol{\psi}}_{c,j+1}^{i}\right]^T \left[(1-p)\bar{\boldsymbol{\psi}}_{c,j}^{i} + p\bar{\boldsymbol{\psi}}_{c,j+1}^{i}\right] \tag{4.1.38b}$$

The latter can be expressed in terms of the mass invariant $\boldsymbol{M}_{8cc}^{i}$ as follows:

$$\begin{aligned} \boldsymbol{M}_{cc}^{i} = \big[&(1-p)^2 \boldsymbol{M}_{8cc}^{i}\{j,j\} + p^2 \boldsymbol{M}_{8cc}^{i}\{j+1,j+1\} + p(1 \\ &- p)\big(\boldsymbol{M}_{8cc}^{i}\{j+1,j\} + \boldsymbol{M}_{8cc}^{i}\{j,j+1\}\big)\big] \end{aligned} \tag{4.1.39}$$

where $\boldsymbol{M}_{8cc}^{i}$ is defined as

$$\boldsymbol{M}_{8cc}^{i} = \sum_{j=1}^{n_n} m^{ij} \bar{\boldsymbol{\Psi}}_{c}^{ij^T} \bar{\boldsymbol{\Psi}}_{c}^{ij} \in R^{n_c \times n_c} \tag{4.1.40}$$

Note that in Eq. 4.1.35 as well as in Eq. 4.1.39 the linear interpolation of contact shapes leads to a quadratic interpolation of mass invariants. This was also observed when interpolating the stiffness invariants in Eq. 4.1.21.

To conclude this section, Eq. 4.1.41 expresses the submatrices of the mass matrix $\boldsymbol{M}^{i}$ in terms of the aforementioned mass invariants when the non-interpolated basis matrix $\boldsymbol{V}_{CS}^{i}$ of Eq. 4.1.5 is used to describe elastic deformation (see also Appendix D.2):

$$\boldsymbol{M}_{\theta\theta}^{i} = \boldsymbol{M}_{2}^{i} + 2\boldsymbol{M}_{4n}^{i}\boldsymbol{q}_{n}^{i} + 2\boldsymbol{M}_{4c}^{i}\boldsymbol{q}_{c}^{i} \tag{4.1.41a}$$

$$\boldsymbol{M}_{\theta n}^{i} = \boldsymbol{M}_{4n}^{i} + \boldsymbol{q}_{n}^{i^T}\boldsymbol{M}_{7nn}^{i} + \boldsymbol{q}_{c}^{i^T}\boldsymbol{M}_{7cn}^{i} \tag{4.1.41b}$$

$$\boldsymbol{M}_{\theta c}^{i} = \boldsymbol{M}_{4c}^{i} + \boldsymbol{q}_{n}^{i^T}\boldsymbol{M}_{7ns}^{i} + \boldsymbol{q}_{c}^{i^T}\boldsymbol{M}_{7cc}^{i} \tag{4.1.41c}$$

$$\boldsymbol{M}_{cc}^{i} = \boldsymbol{M}_{8cc}^{i} \tag{4.1.41d}$$

4.1.4 Selection of the sampling points

One important aspect of the proposed MOR method that hasn't been dealt with thus far is the selection of the sampling points, i.e. the gear pair configurations at which to compute the contact shapes. Two decisions must be made: what angular interval $\left[\theta_{min}^{1}, \theta_{max}^{1}\right]$ to consider and where to locate the sampling points θ_{j}^{1}, for $j = 1, \dots, n_c$, within this interval. The selection of the interval $\left[\theta_{min}^{1}, \theta_{max}^{1}\right]$ is guided by two important observations, namely 1) that most gears are rotationally periodic structures with a period equal to the angular pitch $\tau^{i} = 2\pi/z^{i}$, and 2) that all mass and stiffness invariants of a flexible body are invariant under arbitrary rotations of the vectors involved. As the transmission ratio u of a pair of engaging gears is equal to the ratio of their tooth numbers, it follows from the first observation that the gear pair itself is periodic: when the driving gear travels through one angular pitch τ^{1}, the driven gear travels through $u\tau^{1} = \tau^{2}$ and the gears engage at the exact same tooth flank locations, albeit one tooth further down the line of action. Therefore, it suffices to compute contact shapes on an angular interval of length τ^{1}, i.e. $\left[\theta_{min}^{1}, \theta_{min}^{1} + \tau^{1}\right]$; all

contact shapes that correspond to gear pair configurations outside this interval can be obtained through simple rotation, be it offline or online. Furthermore, by virtue of the second observation, the invariants corresponding to these rotated shapes need not to be recomputed. While the lower limit θ^1_{min} is in principle arbitrary, in the current work the angle at the initial point of contact (see Appendix M) is selected, i.e. $\theta^1_{min} = \theta^1_A$ and $\theta^1_{max} = \theta^1_A + \tau^1$.

Having obtained the angular interval to be sampled, the next question is how to distribute the sampling points across this interval. If an equidistant sampling strategy is opted for, all angular increments are identical, i.e. $\Delta\theta^1_1 = \Delta\theta^1_2 = \cdots = \Delta\theta^1_{n_c-1} = \Delta\theta^1$, where $\Delta\theta^1 = (\theta^1_{max} - \theta^1_{min})/(n_c - 1)$. However, as can be appreciated from the transmission error graphs in Section 3.3, the deformation of a gear is a nonlinear function of the angular configuration of the gear pair, with discontinuities arising whenever teeth enter or leave the zone of contact. In order to properly capture the gear pair's elastic deformation as a function of its angular configuration, the sample points should be distributed across the interval $[\theta^1_{min}, \theta^1_{max}]$ in such a way that regions of high nonlinearity are sampled more.

One way of obtaining such a distribution is by using a Greedy approach (see Section 2.3.2.3). The Greedy algorithm implemented in the current work proceeds as follows. First the angular interval $[\theta^1_{min}, \theta^1_{max}]$ is divided into a large number $n_{c,e} > n_c$ of equidistant sample points. At each sample point a local ROM is constructed, either by concatenating the previously computed contact shapes or by interpolating these shapes, depending on whether the CS or ICS approach is to be used. These local ROMs are used to identify the sample point at which the current ROM yields the largest error (see further). The latter is then updated by evaluating the FOM at the selected sample point, and the process repeats until the desired number of contact shapes are computed. This procedure is summarized in Algorithm 4.1.

The effectiveness of the Greedy algorithm largely depends on the availability of a good measure for the actual error. The latter is defined as

$$\varepsilon^i(\theta^1) = \left\| \boldsymbol{u}^i(\theta^1) - \boldsymbol{V}^i(\theta^1)\boldsymbol{q}^i_f \right\|_2 \tag{4.1.42}$$

where $\|\cdot\|_2$ denotes the Eucledian norm. Obviously, the full-order solution $\boldsymbol{u}^i$ at the parameter realization θ^1 comes at a cost that would render the Greedy algorithm ineffective. Therefore, Algorithm 4.1 uses an error *indicator* that is based on the Eucledian norm of the approximation residual. The latter is obtained by substituting the reduced-order solution into the full-order residual (compare with Eq. 4.1.4):

$$\boldsymbol{r}(\theta^1) = \begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{K}^1_{FE} & \mathbf{0} \\ sym. & & \boldsymbol{K}^2_{FE} \end{bmatrix} \begin{Bmatrix} \theta^1 \\ \boldsymbol{V}^1(\theta^1)\boldsymbol{q}^1_f \\ \boldsymbol{V}^2(\theta^1)\boldsymbol{q}^2_f \end{Bmatrix} - \left(\begin{Bmatrix} T^1 \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}^1_{c,\theta} \\ \boldsymbol{f}^1_{c,f} \\ \boldsymbol{f}^2_{c,f} \end{Bmatrix} \right) \tag{4.1.43}$$

The accuracy of this error indicator was assessed for the spur gear pair (see Section 3.2.4.1) during the first outer loop of Algorithm 4.1 by computing the actual error $\varepsilon(\theta^1)$ for $n_{c,e} = 21$ increments and comparing this with the error indicator $\|\boldsymbol{r}(\theta^1)\|_2$, see Fig. 4.3. It is apparent from this figure that the error indicator follows the same trend as the actual error, and can thus be used to steer the Greedy algorithm.

Algorithm 4.1 Greedy procedure for determining the angular sample distribution

Input: Stiffness matrices $\boldsymbol{K}_{FE}^1$ and $\boldsymbol{K}_{FE}^2$, torque T^1, number of shapes n_c and $n_{c,e}$
Output: Contact shapes $\boldsymbol{\psi}_{c,j}^1$ and $\boldsymbol{\psi}_{c,j}^2$ and associated angles θ_j^1 for $j = 1, \dots, n_c$

Distribute $n_{c,e}$ equidistant sample points across angular interval $\left[\theta_{min}^1, \theta_{max}^1\right]$
Compute contact shapes $\boldsymbol{\psi}_{c,1}^1$ and $\boldsymbol{\psi}_{c,1}^2$ at $\theta_1^1 = \theta_{min}^1$ using Eq. 4.1.4
Compute contact shapes $\boldsymbol{\psi}_{c,n_c}^1$ and $\boldsymbol{\psi}_{c,n_c}^2$ at $\theta_{n_c}^1 = \theta_{max}^1$ using Eq. 4.1.4
Initialize $P = \{1, n_c\}$ and $\boldsymbol{\theta}^1 = \left\{\theta_1^1, \theta_{n_c}^1\right\}$
for $i = 2, \dots, n_c - 1$ **do**
    Construct ROM based on $\boldsymbol{\psi}_{c,j}^1$ and $\boldsymbol{\psi}_{c,j}^2$ for $j = 1, \dots, i-1$
    for $k = 1, \dots, n_{c,e} \setminus P$
        Solve reduced-order static contact problem at θ_k^1
        Compute approximation residual $\boldsymbol{r}_k = \boldsymbol{r}(\theta_k^1)$ using Eq. 4.1.43
    end
    Find index k that maximizes $\|\boldsymbol{r}_k\|_2$
    Set $P \leftarrow \text{sort}[P \quad k]$ and $\boldsymbol{\theta}^1 \leftarrow \text{sort}[\boldsymbol{\theta}^1 \quad \theta_k^1]$
    Compute contact shapes $\boldsymbol{\psi}_{c,i}^1$ and $\boldsymbol{\psi}_{c,i}^2$ at $\theta_i^1 = \theta_k^1$ using Eq. 4.1.4
end

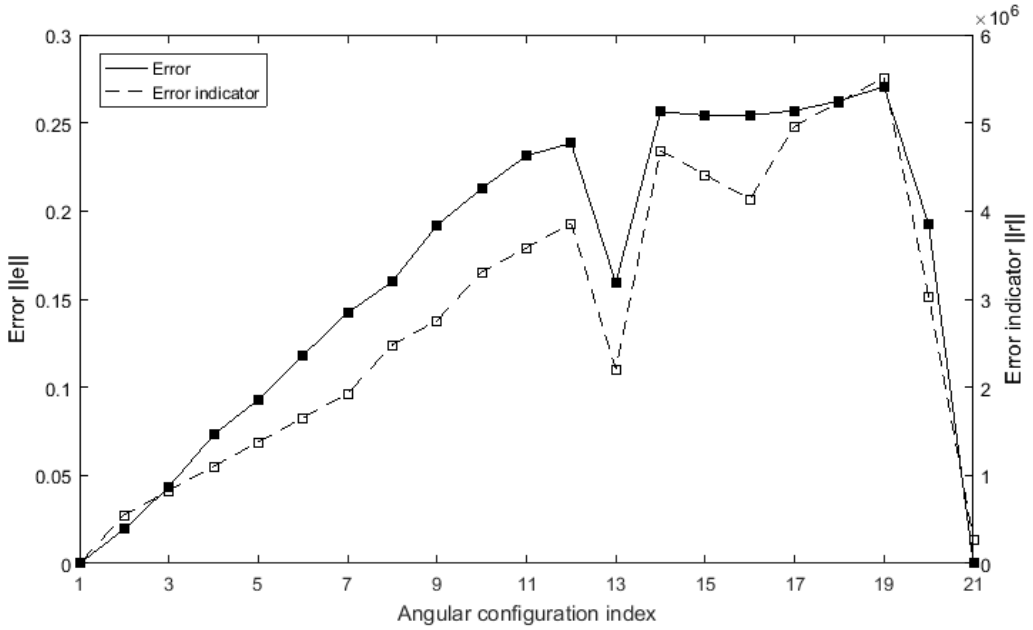


Fig. 4.3 Comparison of the actual error and the error estimator for the spur gear pair, with $n_{c,e} = 21$

4.1.5 Dealing with nonlinear torque dependence

Thus far the nonlinear dependence of the contact shapes $\boldsymbol{\Psi}_c^i$ on the applied torque T^1 (see Eq. 4.1.4) was neglected, assuming the analyzed gear pair operates at a constant torque. However, in gear contact analysis it is often desirable to investigate the dynamic behavior of a pair of gears subjected to time-varying external torques. In this section, the nonlinear torque dependence of a gear's displacement field on the applied torque is briefly investigated using theoretical arguments and numerical experiments. Then a simple extension of the (I)CS approach is proposed to include this nonlinear effect in the ROM.

The nonlinearity in the gear's torque-displacement relationship is caused by the nonlinear dependence of the contact surface area on the applied load (see Section 3.2.1, Eq. 3.2.3). It is a local effect that can be separated from the global *linear* deformation of the gear, as is typically done in semi-analytical gear contact models (see Appendix N.2). A number of formulas are available for computing the local deformation δ of a pair of spur teeth in perfect line contact under the assumptions that the surfaces in contact are perfectly smooth, the elastic limits of the materials are not exceeded, and no frictional forces are active within the contact area. The formula of Weber-Banasheck [164] gives the normal displacement between the tooth surface and a fixed border at a depth h underneath the contact surface (see Appendix N.2, Eq. N.5)), whereas Lundberg [165] gives a formula for the local compression between two parallel-axis cylinders in contact. The latter is given by

$$\delta = \frac{2(1-\nu^2)}{\pi E} q \left[1 + ln \left\{ \frac{\pi E b^2}{(1-\nu^2) q} \left(\frac{\varrho_1 + \varrho_2}{\varrho_1 \varrho_2} \right) \right\} \right] \tag{4.1.44}$$

where E and ν are the material's modulus of elasticity and Poisson coefficient, q is the load per unit width, and b is the active width of the gear pair, and ϱ_1 and ϱ_2 are the radii of curvature at the contact point of the pinion and wheel. In Fig. 4.4, these formula are plotted for the parallel-axis spur gear pair introduced in Section 3.2.5 when one pair of teeth is in contact. The solution of Weber-Banasheck is plotted for a number of h values and approaches the Lundberg solution as h approaches the equivalent radius of curvature at the contact point. In the proximity of the contact surface (i.e. small values of h), nonlinear effects are dominant whereas their influence diminishes as the value of h increases. It is a well-known fact that both solutions can be approximated within engineering accuracy by a power-law function of the form $\delta \approx C q^k$, where C is a constant and k is between 0.9 and 0.925 [166], indicating mild nonlinearity of the load-displacement relationship.

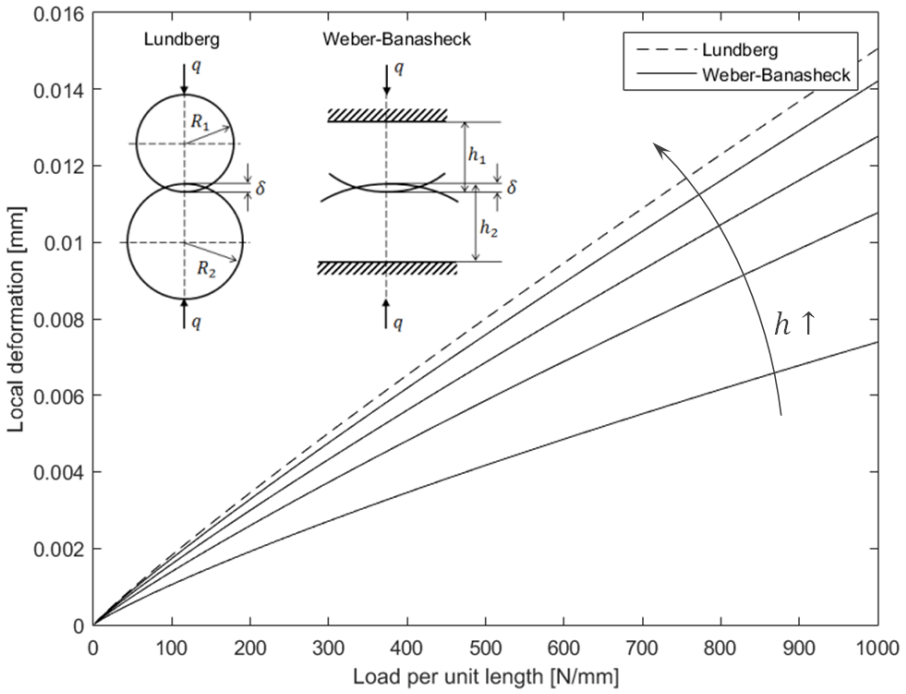
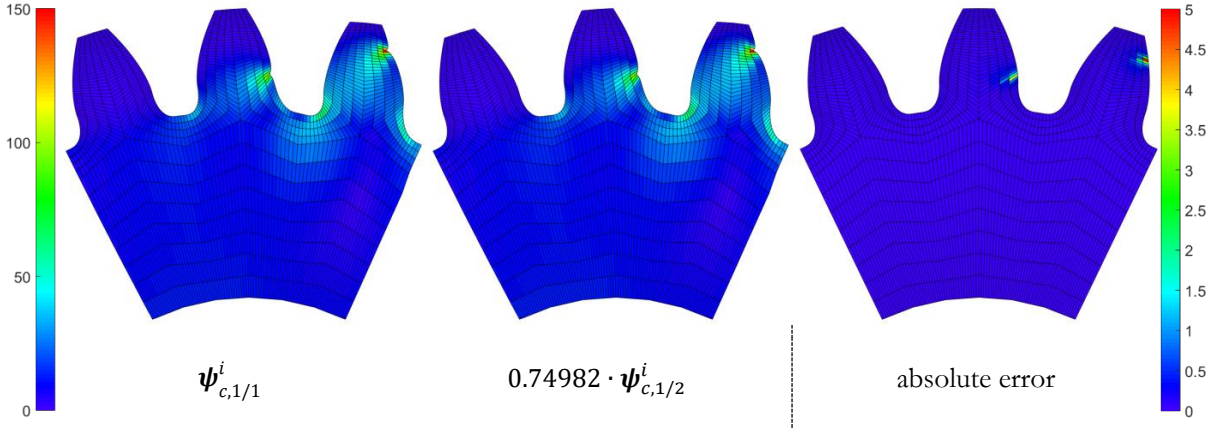


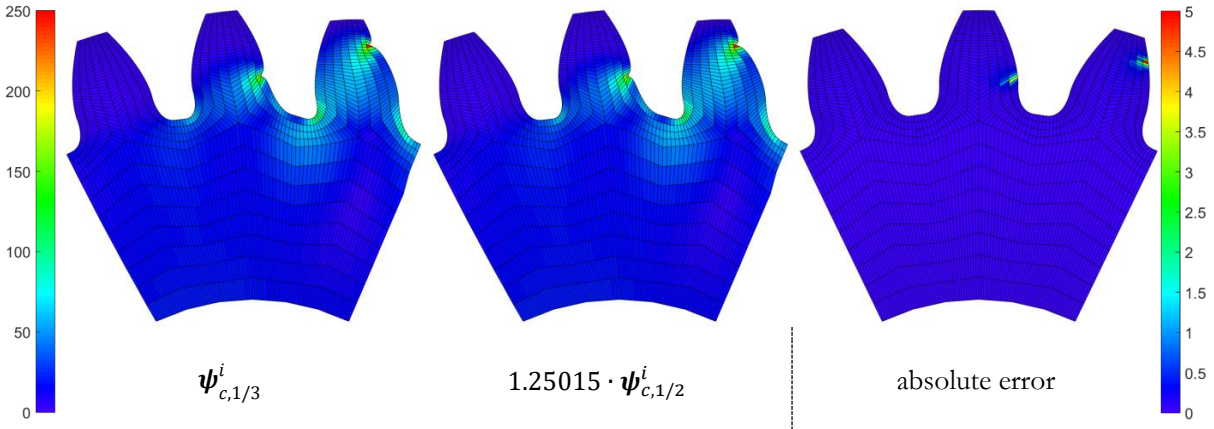
Fig. 4.4 Compression of a pair of spur teeth in perfect line contact as a function of applied load

In order to numerically quantify the degree of nonlinearity in a gear's torque-displacement relationship, the parallel-axis spur gear pair of Section 3.2.5 is considered at an arbitrary configuration θ_1^1 and subjected to four distinct torques $T_1^1 = 150NM$, $T_2^1 = 200NM$, $T_3^1 = 250NM$, and $T_4^1 = 400NM$, yielding the contact shapes $\boldsymbol{\psi}_{c,1/k}^i$ for $k = 1, \ldots, 4$. The first question is whether it is possible to accurately represent $\boldsymbol{\psi}_{c,1/k}^i$ by properly scaling $\boldsymbol{\psi}_{c,1/l}^i$ when $k \neq l$. In Fig. 4.5, the optimal approximation of $\boldsymbol{\psi}_{c,1/1}^i$ (Fig. 4.5a), $\boldsymbol{\psi}_{c,1/3}^i$ (Fig. 4.5b) and $\boldsymbol{\psi}_{c,1/4}^i$ (Fig. 4.5c) by means of a least-squares fit of $\boldsymbol{\psi}_{c,1/2}^i$ are shown: the left, middle and right figures respectively show the actual contact shape, the approximated contact shape, and the absolute difference between the two. From the latter, it is clear that the global deformation of the gear is approximated accurately by linear scaling of $\boldsymbol{\psi}_{c,1/2}^i$, whereas a sizeable error may be induced in the vicinty of the points of contact. For relative small variations of input torque (25% lower and higher in Fig. 4.5a-b), the maximal relatie error is limited (2.7% and 2.1%, respectively); for large torque variations (100% higher in Fig. 4.5c), this error becomes non-negligible (5.2%).

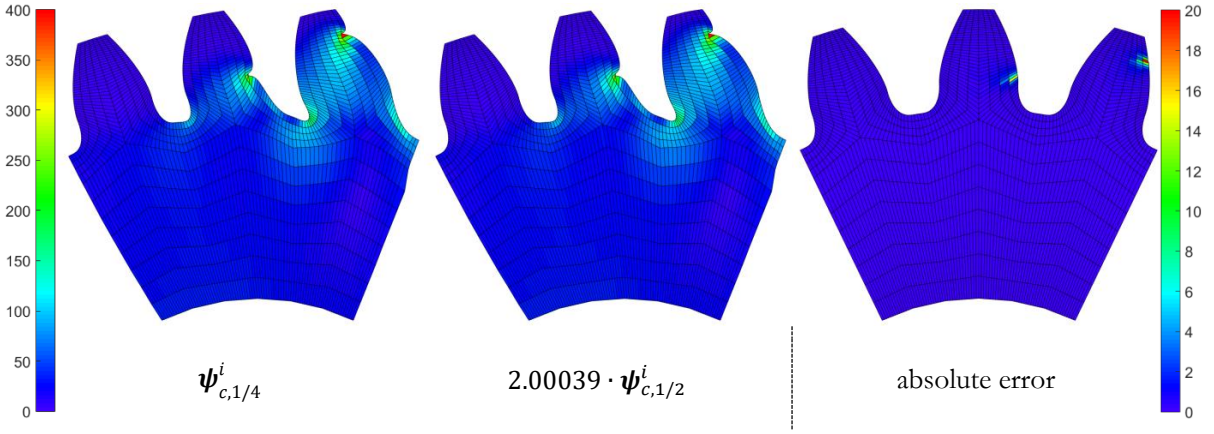
In view of the proposed modification of the (I)CS approach, the next question to be investigated is whether it is possible to obtain a better approximation of a contact shape by linearly interpolating among two contact shapes instead of scaling a single contact shape. In Fig. 4.6, the contact shape $\boldsymbol{\psi}_{c,1/3}^i$ (corresponding to $T_3^1 = 250NM$) is approximated by a linear combination of $\boldsymbol{\psi}_{c,1/1}^i$ (corresponding to $T_1^1 = 150NM$) and $\boldsymbol{\psi}_{c,1/4}^i$ (corresponding to $T_4^1 = 400NM$). The error is again largest in the zones of contact, but the maximum relative error in this case is below 0.05%.



(a) $T^1 = 150NM$, $T^1_1/T^1_2 = 0.75$, error norm = 0.00026, max. relative error = 2.7%

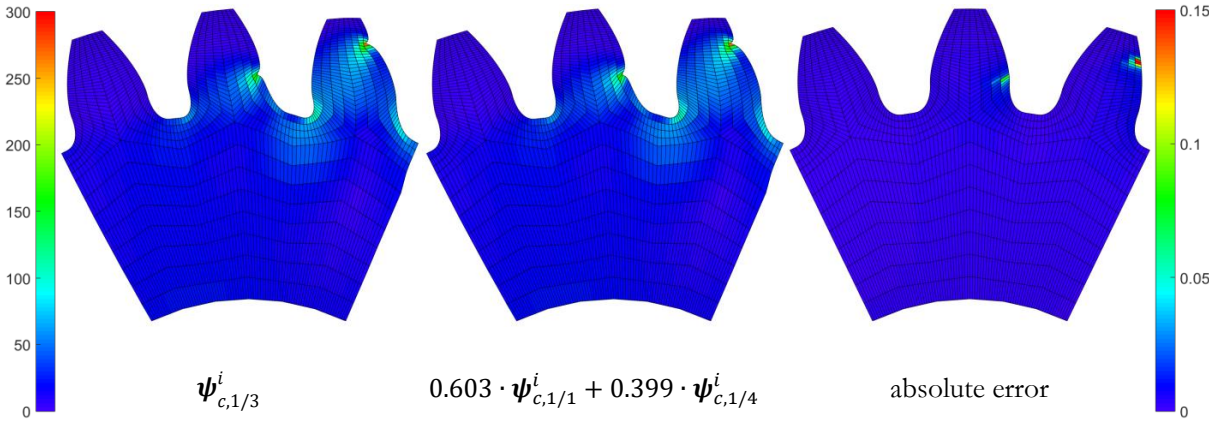


(b) $T^1 = 250NM$, $T^1_3/T^1_2 = 1.25$, error norm = 0.00026, max. relative error = 2.1%



(c) $T^1 = 400NM$, $T^1_4/T^1_2 = 2.00$, error norm = 0.00104, max. relative error = 5.2%

Fig. 4.5 Optimal approximation of $\boldsymbol{\psi}^i_{c,1/1}$, $\boldsymbol{\psi}^i_{c,1/3}$ and $\boldsymbol{\psi}^i_{c,1/4}$ by means of a least-squares fit of $\boldsymbol{\psi}^i_{c,1/2}$. The displayed Von Mises stresses are expressed in [MPa]. Displacements are magnified 500 times.



$T^1 = 250NM$, $T^1_3/T^1_1 = 1.667$, $T^1_3/T^1_4 = 0.625$, error norm = 0.00002 , max. relative error = 0.05%

Fig. 4.6 Optimal approximation of $\boldsymbol{\psi}^i_{c,1/3}$ by means of a least-squares fit of $\left[\boldsymbol{\psi}^i_{c,1/1} \quad \boldsymbol{\psi}^i_{c,1/4}\right]$. The displayed Von Mises stresses are expressed in [MPa]. Displacements are magnified 500 times.

The above numerical experiments have demonstrated that when large torque variations are expected, it may be advisable to enrich the basis matrix $\boldsymbol{V}^{ij}_{(I)CS}$ with contact shapes computed at multiple torque levels. Generally a small number n_T of torque levels is sufficient to capture the torque-displacement nonlinearity, e.g. $n_T = 2$ or 3. In the CS approach, the basis matrix (see Eq. 4.1.5) then becomes

$$\boldsymbol{V}^i_{CS} = [\boldsymbol{\Phi}^i \quad \boldsymbol{\Psi}^i_c] = \left[\boldsymbol{\Phi}^i \quad \boldsymbol{\psi}^i_{c,1/1} \quad \cdots \quad \boldsymbol{\psi}^i_{c,n_c/n_T}\right] \in \boldsymbol{R}^{N\times(n_k+n_c n_T)} \tag{4.1.45}$$

while the basis matrix associated with the ICS approach (see Eq. 4.1.13) becomes

$$\boldsymbol{V}^i_{ICS} = [\boldsymbol{\Phi}^i \quad \boldsymbol{\psi}^i_c] = \left[\boldsymbol{\Phi}^i \quad \boldsymbol{\psi}^i_{c/1}(\theta^1) \quad \cdots \quad \boldsymbol{\psi}^i_{c/n_T}(\theta^1)\right] \in \boldsymbol{R}^{N\times(n_k+n_T)} \tag{4.1.46}$$

where

$$\boldsymbol{\psi}^i_{c/k}(\theta^1) = (1-p)\boldsymbol{\psi}^i_{c,j/k} + p\boldsymbol{\psi}^i_{c,j+1/k} \in \boldsymbol{R}^{N\times 1} \tag{4.1.47}$$

Note that it was opted for not to interpolate contact shapes corresponding to the same angular configuration. The computational cost that would be associated with such a multivariate interpolation scheme (based on both angular configuration and torque level) is not justified by the small reduction of generalized elastic coordinates. The univariate interpolation scheme for $n_T = 3$ is graphically depicted in Fig. 4.7.

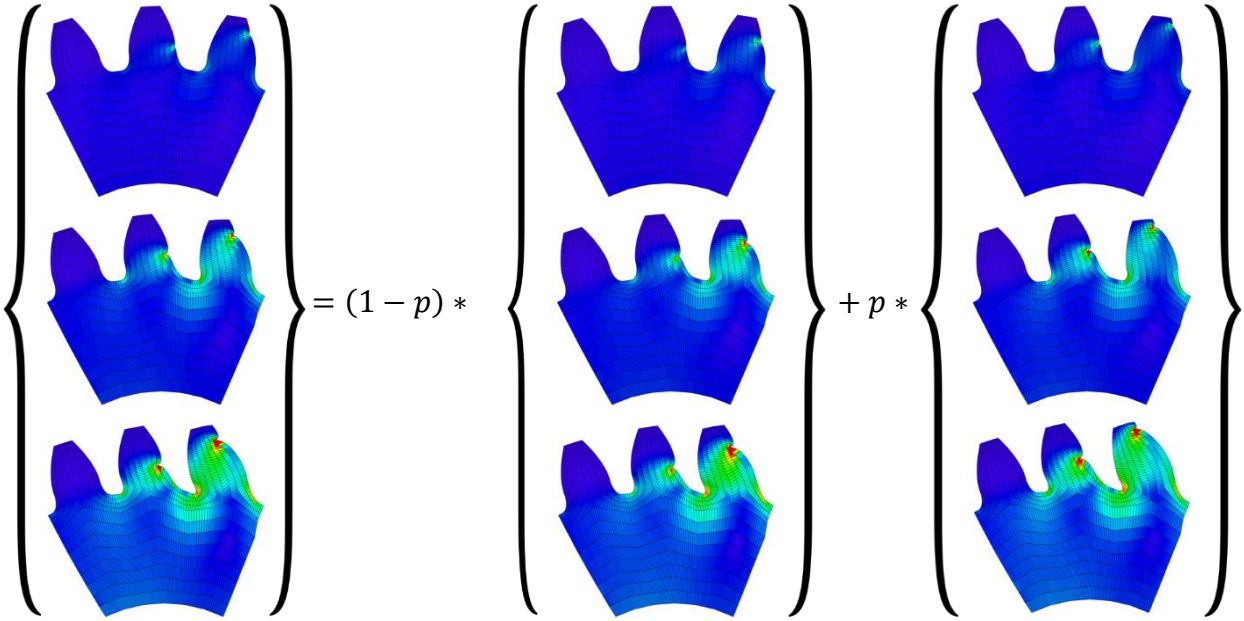


Fig. 4.7 Univariate interpolation scheme in the case of multiple torque levels

4.2 Efficient construction of the reduced-order model

4.2.1 An intermediate reduced-order model

Both the CS and ICS approaches rely on the pre-computation of a set of contact shapes, the number of which typically varies between 10 and 50. As this requires repeatedly solving the large nonlinear system of Eq. 4.1.4 (involving the FOM and the computationally expensive nonlinear contact force vector $\boldsymbol{f}_c^i$), a considerable *offline* cost is to be expected, which can add significantly to the overall analysis cost.

As a first means to reducing this cost, an intermediate ROM (iROM) is used to solve Eq. 4.1.4. As this equation represents a static contact problem with linear internal forces, an exact solution can be obtained by applying a Guyan-like reduction to the flexible DOFs of each of the FE models (see Section 2.3.3.1). The latter are partitioned according to whether or not they can be loaded during the contact analysis, with the first subset comprised of all the DOFs that correspond to nodes on the active flanks of the gears. The attachment modes $\boldsymbol{\Psi}_a^i \in \mathbb{R}^{N \times n_a^i}$ corresponding to these DOFs are computed using Eq. 2.3.27, and the second and third lines of Eq. 4.1.4 are projected onto the lower-dimensional subspace spanned by the columns of $\boldsymbol{\Psi}_a^i$ to obtain:

$$\begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{K}_{aa}^1 & \mathbf{0} \\ sym. & & \boldsymbol{K}_{aa}^2 \end{bmatrix} \begin{Bmatrix} \theta_j^1 \\ \boldsymbol{q}_{a,j}^1 \\ \boldsymbol{q}_{a,j}^2 \end{Bmatrix} = \begin{Bmatrix} T^1 \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}_{c,\theta}^1 \\ \boldsymbol{\Psi}_a^{1\,T} \boldsymbol{f}_{c,f}^1 \\ \boldsymbol{\Psi}_a^{2\,T} \boldsymbol{f}_{c,f}^2 \end{Bmatrix} \tag{4.2.1}$$

where $\boldsymbol{K}_{aa}^{i} = {\boldsymbol{\Psi}_{a}^{i}}^{T} \boldsymbol{K}_{FE}^{i} \boldsymbol{\Psi}_{a}^{i}$ is the projected stiffness matrix of the *i*-th gear, and $\boldsymbol{q}_{a,j}^{i}$ represents the modal DOFs corresponding to the attachment modes $\boldsymbol{\Psi}_{a}^{i}$. A reduced-order yet statically complete solution to Eq. 4.1.4 is now obtained by solving Eq. 4.2.1 and computing the contact shape $\boldsymbol{\psi}_{c,j}^{i}$ from the transformation $\boldsymbol{\psi}_{c,j}^{i} = \boldsymbol{\Psi}_{a}^{i} \boldsymbol{q}_{a,j}^{i}$.

For a gear pair having a contact ratio of ε and a gear flank FE discretization of $N_{fef} \times N_{few}$, where N_{fef} and N_{few} denote the number of elements along the involute curve and the width of the gear, the number of equations in Eq. 4.2.1 is equal to $1 + \lceil \varepsilon \rceil \{2[3(N_{fef} \times N_{few})]\}$. For a helical gear pair having a contact ratio between 3 and 4 and a typical gear flank discretization of (24×10), the number of simultaneous equations to be solved is 5761. While this is substantially lower than the number of equations in Eq. 4.1.4, it still involves a considerable computational cost, especially since the sparsity of the full-order stiffness matrix is lost. In order to further reduce the computational complexity of Eq. 4.2.1, an additional partitioning is performed per angular configuration of the gear pair. Using the analytical equations of Appendix M, the contact zones of all the active tooth flanks in the current angular configuration are identified and the equations corresponding to nodes outside of these zones are eliminated from Eq. 4.2.1. In doing so, the number of equations in the reduced-order equilibrium equations is reduced by an additional factor of 5-10.

4.2.2 Efficient computation of the Jacobian matrix

Eq. 4.2.1 represents a system of nonlinear simultaneous equations that is solved using a Newton-type iterative method. Given an approximation $\boldsymbol{q}_k = \left\{\theta_j^1 \quad {\boldsymbol{q}_{a,j}^1}^T \quad {\boldsymbol{q}_{a,j}^2}^T\right\}_k^T \in \mathbb{R}^{n_q}$ to the solution of Eq. 4.2.1, a better approximation $\boldsymbol{q}_{k+1}$ is obtained using the following formula:

$$\boldsymbol{q}_{k+1} = \boldsymbol{q}_k - \left[\frac{\partial \boldsymbol{r}(\boldsymbol{q}_k)}{\partial \boldsymbol{q}_k}\right]^{-1} \boldsymbol{r}(\boldsymbol{q}_k) = \boldsymbol{q}_k - \boldsymbol{J}(\boldsymbol{q}_k)^{-1} \boldsymbol{r}(\boldsymbol{q}_k) \tag{4.2.2}$$

where $\boldsymbol{r}(\boldsymbol{q}_k) = \boldsymbol{K}\boldsymbol{q}_k - \boldsymbol{f}_{ex} - \boldsymbol{f}_c(\boldsymbol{q}_k)$ represents the residual of Eq. 4.2.1, $\boldsymbol{K} = diag(0, \boldsymbol{K}_{aa}^1, \boldsymbol{K}_{aa}^1)$ is the constant stiffness matrix, and $\boldsymbol{f}_{ex}$ and $\boldsymbol{f}_c$ are the generalized external and contact forces as defined in Eq. 4.2.1. In moving towards a sufficiently accurate approximation to the solution of Eq. 4.2.1, most of the computational efforts are spent on computing and inverting the Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}_k)$. This matrix can be written as:

$$\boldsymbol{J}(\boldsymbol{q}_k) = \left[\frac{\partial \boldsymbol{r}(\boldsymbol{q}_k)}{\partial \boldsymbol{q}_k}\right] = \boldsymbol{K} - \frac{\partial \boldsymbol{f}_c(\boldsymbol{q}_k)}{\partial \boldsymbol{q}_k} = \boldsymbol{K} - \boldsymbol{J}_c(\boldsymbol{q}_k) \tag{4.2.3}$$

where $\boldsymbol{J}_c$ is the Jacobian matrix of the contact forces, or the contact Jacobian. As $\boldsymbol{K}$ is readily available at each iteration, the computational complexity of $\boldsymbol{J}_c$ is largely determinant for the cost

of $\boldsymbol{J}$. Due to the high nonlinearity of the generalized contact forces $\boldsymbol{f}_c(\boldsymbol{q}_k)$, the contact Jacobian $\boldsymbol{J}_c$ is typically computed using a finite differencing scheme. Given a perturbation vector $\Delta\boldsymbol{q}_k$, the *i*-th column of $\boldsymbol{J}_c(\boldsymbol{q}_k)$ is obtained from:

$$\boldsymbol{J}_c[:,i] = \frac{\boldsymbol{f}_c(\boldsymbol{q}_k + \Delta\boldsymbol{q}_k[i]) - \boldsymbol{f}_c(\boldsymbol{q}_k)}{\Delta\boldsymbol{q}_k[i]} \qquad \text{for } i = 1..n_q \tag{4.2.4}$$

where $[:,i]$ denotes the *i*-th column of the associated matrix and $[i]$ refers to the *i*-th element of the associated vector. The computation of $\boldsymbol{J}_c$ thus requires $n_q \approx 10^{2-3}$ re-evaluations of the generalized contact forces $\boldsymbol{f}_c(\boldsymbol{q}_k)$ at each iteration *k*.

Considerable computational savings could be obtained when an analytical expression for $\boldsymbol{J}_c$ was available. In Appendix A, the chain rule of differentiation is applied to the analytical expressions of the generalized contact forces (see Chapter 2, Section 2.5.3.2), to obtain a semi-analytical expression for the contact Jacobian. The expression is *semi*-analytical because, for reasons of efficiency, certain lengthy partial derivatives are approximated using local finite differencing. Notwithstanding the considerable length of the semi-analytical expression, the number of floating point operations (flops) required to evaluate $\boldsymbol{J}_c$ is considerable smaller than the flops involved in Eq. 4.2.4, thus yielding considerable savings in the computation of $\boldsymbol{J}(\boldsymbol{q}_k)$.

Alternatively, the cost of Eq. 4.2.4 can be considerably reduced by assuming that the nodes that are in contact will remain in contact after perturbing the vector of generalized coordinates $\boldsymbol{q}_k$, and that no new nodes will enter the contact zone. In doing so, the computationally intensive contact detection phase, which identifies the slave nodes and master elements that are involved in the contact interaction (see Section 3.2.4), can be omitted and the generalized contact forces $\boldsymbol{f}_c(\boldsymbol{q}_k)$ can be computed directly, based on previously stored node and element indices. In Fig. 4.8, the three aforementioned approaches for computing the Jacobian matrix are compared in terms of computational effort. Note that, depending on the mesh size, the analytical approach is up to 20 times more efficient than the brute force numerical approach of Eq. 4.2.4, whereas the efficient numerical approach (assuming no nodes enter or leave the contact zone during perturbation) is typically 3 times more efficient.

Finally, when the approximation to the solution of Eq. 4.2.1 approaches the real solution, it might not be necessary to re-compute the Jacobian matrix $\boldsymbol{J}(\boldsymbol{q}_k)$ using any of the aforementioned approaches in each iteration. Instead, Broyden's method [167] can be used to approximate $\boldsymbol{J}(\boldsymbol{q}_k)$ based on the previous value of the Jacobian, $\boldsymbol{J}(\boldsymbol{q}_{k-1})$:

$$\boldsymbol{J}(\boldsymbol{q}_k) \approx \boldsymbol{J}(\boldsymbol{q}_{k-1}) + \frac{\Delta\boldsymbol{r}(\boldsymbol{q}_k) - \boldsymbol{J}(\boldsymbol{q}_{k-1})\Delta\boldsymbol{q}_k}{\|\Delta\boldsymbol{q}_k\|^2}\Delta\boldsymbol{q}_k^T \tag{4.2.5}$$

where $\Delta\boldsymbol{r}(\boldsymbol{q}_k) = \boldsymbol{r}(\boldsymbol{q}_k) - \boldsymbol{r}(\boldsymbol{q}_{k-1})$ and $\Delta\boldsymbol{q}_k = \boldsymbol{q}_k - \boldsymbol{q}_{k-1}$. Using the formula of Sherman-Morrison [168], the above equation can be used to directly find an approximation to the inverse of the Jacobian matrix, based on the previous value of the inverse Jacobian matrix, $\boldsymbol{J}^{-1}(\boldsymbol{q}_{k-1})$:

$$\boldsymbol{J}^{-1}(\boldsymbol{q}_k) \approx \boldsymbol{J}^{-1}(\boldsymbol{q}_{k-1}) + \frac{\Delta\boldsymbol{q}_k - \boldsymbol{J}^{-1}(\boldsymbol{q}_{k-1})\Delta\boldsymbol{r}(\boldsymbol{q}_k)}{\Delta\boldsymbol{q}_k^T\boldsymbol{J}^{-1}(\boldsymbol{q}_{k-1})\Delta\boldsymbol{r}(\boldsymbol{q}_k)}\Delta\boldsymbol{q}_k^T\boldsymbol{J}^{-1}(\boldsymbol{q}_{k-1}) \tag{4.2.6}$$

Eq. 4.2.5 merely contains additions and multiplications of quantities that have been computed previously, thus yielding an efficient update of the Jacobian matrix. However, as a mere high-dimensional extension of the secant method, Broyden's method has an order of convergence equal to the golden ratio (≈ 1.618) and is not guaranteed to converge for nonlinear systems of equations. Therefore, in the current work, Newton's method is used during the first iterations until the residual drops below a certain threshold; from then on the algorithm switches to Broyden's method until the solution is converged.

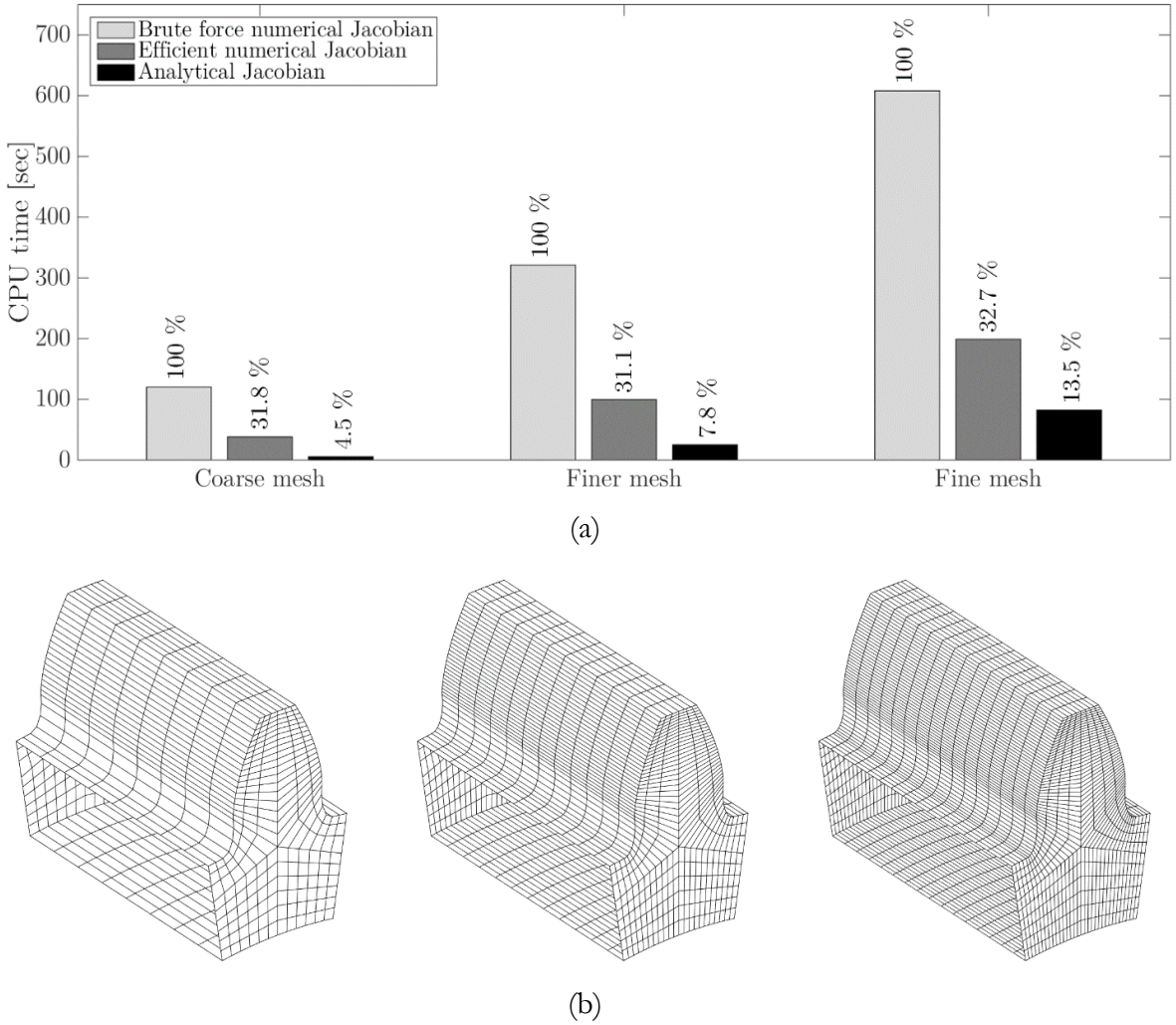


Fig. 4.8 Comparison of the computational efficiency of 3 approaches to compute the Jacobian matrix

4.2.3 A mixed-grid finite element strategy

While the use of an iROM (Eq. 4.2.1) significantly lowers the cost of computing the contact shapes, it does bring about inverting large-dimensional stiffness matrices and storing 10^{3-4} attachment modes. For the helical gear pair presented in Section 3.2.5, storage requirements reach up to several tenths of Gigabytes, leading to significant write-to-memory times or even out-of-memory warnings (e.g. when using MATLAB).

The rotational periodicity of gears can be exploited to reduce the number of DOFs in the full-order FEM and thereby reduce the length of the attachment modes. When computing the global contact shapes, the gear pair's angular configuration is discretized over a single angular pitch of the reference gear (see Section 4.1.4). Therefore, only the teeth that engage as the gear pair moves from its initial configuration to the configuration one angular pitch further should be finely discretized; the remaining teeth merely contribute to the *overall* stiffness of the respective gears, and can thus be discretized using a relatively coarse FE grid. For a pair of parallel-axis external gears, the number of teeth to be finely discretized in each gear is equal to $\lceil \varepsilon \rceil + 1$, where ε is the contact ratio of the gear pair. As in most industrial gears this is only a fraction of the total number of teeth, the storage requirements for the attachment modes as well as the dimensions of the full-order stiffness matrices are significantly reduced. The construction of mixed-grid FE models is discussed in Section 3.2.1.

A number of measures should be taken to enable full-revolution simulations using a mixed-grid FE strategy. Firstly, the contact force algorithm should always operate on a fine grid, no matter which teeth are in engagement. This implies that both the static and dynamic tooth deformations – as described respectively by contact shapes and eigenvectors – should be defined on a fine grid. In Section 4.1.4 it was discussed that when a gear pair travels over one angular pitch and previously unloaded teeth start to engage, the associated contact shapes are rotated over the same amount and the interpolation of contact shapes proceeds with the *rotated* contact shapes, leaving the mass and stiffness invariants unaltered. In doing so, the static deformations of the previously unloaded teeth are described in terms of a fine grid, notwithstanding the coarse discretization that was applied when constructing the FE model. A similar procedure can however not be used for the eigenvectors: as these are not rotationally periodic, they cannot simply be rotated whenever the gear pair travels through an angular pitch. Instead, the eigenvectors should be defined on a fine grid for *all* teeth that can possibly engage in the course of the simulation. An elegant way of computing these fine-grid eigenvectors starting from the stiffness matrix of a single finely meshed tooth can be found in [169], where an algorithm for exact eigenvector extraction for rotationally periodic structures is given.

Secondly, for the interpolation algorithm (see Section 4.1.2) to work, the mixed-grid contact shapes should be made mass- and stiffness orthogonal with respect to the set of eigenvectors. This can be done using Algorithm 2.1 in Section 2.3.3.1, which includes repeatedly solving a reduced eigenvalue problem involving the reduced mass and stiffness matrices $\boldsymbol{M}_{VV} = \boldsymbol{V}^T \boldsymbol{M}_{FE} \boldsymbol{V}$ and $\boldsymbol{K}_{VV} =$

$\boldsymbol{V}^T\boldsymbol{K}_{FE}\boldsymbol{V}$. In these expressions, $\boldsymbol{V} = [\boldsymbol{\Phi} \quad \boldsymbol{\psi}_{c,j}] \in \mathbb{R}^{N\times(n_k+1)}$ is the matrix obtained by concatenating the set of eigenvectors $\boldsymbol{\Phi} \in \mathbb{R}^{N\times n_k}$ and a single contact shape $\boldsymbol{\psi}_{c,j} \in \mathbb{R}^{N\times 1}$, and $\boldsymbol{M}_{FE}$ and $\boldsymbol{K}_{FE}$ represent the mixed-grid FE mass and stiffness matrices. Note that $\boldsymbol{\Phi}$ and $\boldsymbol{\psi}_{c,j}$ should have the same length, whereas thus far only fine-grid eigenvectors and mixed-grid contact shapes were discussed. There are two ways of obtaining the mixed-grid eigenvectors: either they are computed by solving the mixed-grid eigenvalue problem, or they are obtained by fitting the fine-grid eigenvectors onto the mixed grid. The first approach yields eigenvectors that differ considerably in shape from the fine-grid eigenvectors, which is due to the breaking of perfect rotational periodicity in the mixed-grid FEM. The second approach yields eigenvectors that have the same shape as the fine-grid eigenvectors, yet are not mass- and stiffness-orthogonal with respect to the mixed-grid mass and stiffness matrices. While being mathematically dubious, a very good agreement with the fine-grid formulation was obtained by fitting the fine-grid eigenvectors onto the mixed grid and numerically forcing orthogonality of the fitted eigenvectors with respect to the mixed-grid mass and stiffness matrices.

In summary, the mixed-grid FE strategy reduces both storage requirements and CPU time during the offline phase of the simulation. By applying a fine discretization to a small subset of teeth only, the number of integers to be stored decreases drastically, as do the dimensions of the stiffness matrices to be inverted and the matrix-vector products to be computed.

4.2.4 Flow chart of the offline phase of the simulation

Fig. 4.9 shows a flow chart of the ROM construction during the offline phase of the simulation, starting from the full-order mass and stiffness matrices $\boldsymbol{M}_{FE}^i$ and $\boldsymbol{K}_{FE}^i$. In the *component modes phase*, the mixed-grid attachment modes $\boldsymbol{\Psi}_a^i$ and reduced-order stiffness matrices $\boldsymbol{K}_{aa}^i$ are computed through decomposition of the full-order stiffness matrices $\boldsymbol{K}_{FE}^i$ and pre- and post-multiplication of these matrices with $\boldsymbol{\Psi}_a^i$. At the same time, the fine-grid eigenvectors $\boldsymbol{\Phi}_{fine}^i$ are computed by solving the rotationally periodic eigenvalue problem involving the mass and stiffness matrix of a finely discretized tooth. These eigenvectors are then fitted onto the mixed grid using an interpolation scheme based on Delaunay triangulation [170], to obtain the mixed-grid eigenvectors $\boldsymbol{\Phi}_{mixed}^i$. Note that $\boldsymbol{\Phi}_{fine}^i$ is used for computing the tooth deformation during the integration of the equations of motion, whereas $\boldsymbol{\Phi}_{mixed}^i$ is used for residualization of the contact shapes and computation of the mass and stiffness invariants.

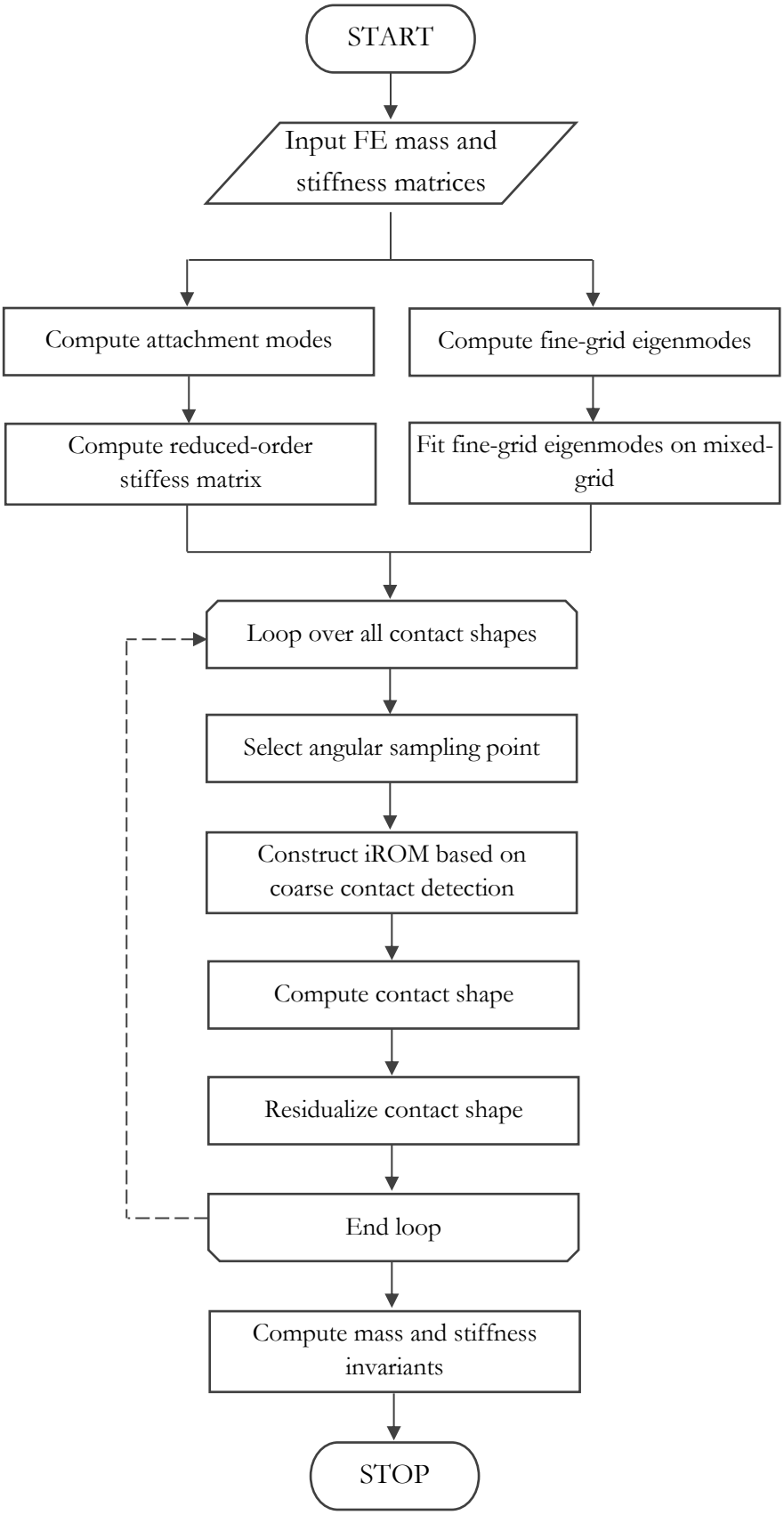


Fig. 4.9 Flow chart of the offline phase of the simulation

In the *contact shapes phase*, Eq. 4.2.1 is solved for a number of angular gear pair configurations, based on the mixed-grid attachment modes and reduced-order stiffness matrices. For each angular configuration, a coarse contact detection (see Section 3.2.4) is performed to identify the possible contact nodes and to eliminate equations corresponding to unloaded DOFs in Eq. 4.2.1. A Newton-type iterative scheme with a line search algorithm [171] and a combination of analytical and Broyden Jacobian updates is used to efficiently solve the remaining equations. Having obtained the mixed-grid contact shapes $\boldsymbol{\Psi}_c^i$, these are made residual with respect to the mixed-grid eigenvectors $\boldsymbol{\Phi}_{mixed}^i$ using Algorithm 2.1 in Section 2.3.3.1, with enforcement of orthogonality of $\boldsymbol{\Phi}_{mixed}^i$ with respect to the mixed-grid mass and stiffness matrices $\boldsymbol{M}_{FE}^i$ and $\boldsymbol{K}_{FE}^i$. Finally, in the *invariants phase*, the mixed-grid mass and stiffness invariants are computed using the equations of Section 4.1.3.3.

4.3 Integration of the equations of motion

4.3.1 Flow chart of the online phase of the simulation

Having constructed the ROM, the reduced-order equations of motion (Eq. 4.1.27) can be integrated in time. The development of an optimal integration scheme specifically tailored to the (I)CS approach is dealt with in Chapter 7. In this chapter, instead, the primary focus is on assessing the influence of the reduced number of DOFs on the numerical complexity of the integration scheme. The latter is measured by counting the number of Floating Point Operations (FLOPs) that are required to propagate the solution in time. To this end, the fixed-increment explicit Central Difference (CD) method (Section 2.6.1.3) is applied to the reduced-order equations of motion. The resulting integration scheme is outlined in Fig. 4.10. Note that the dotted blocks need to be processed only when the ICS approach is used; in the non-interpolated CS approach these are (largely) omitted.

At the beginning of each time increment, three interpolation parameters are computed based on the angular configuration θ^1 of the gear pair. These are the angular pitch parameter τ, the contact shape number *j,* and the interpolation parameter *p*. The angular pitch parameter counts the number of angular pitches travelled by the reference gear and controls the rotation of the contact shapes when $\tau > 1$. The shape number *j* selects the contact shapes $\boldsymbol{\psi}_{c,j}^i$ and $\boldsymbol{\psi}_{c,j+1}^i$ that need to be interpolated to yield the interpolated contact shape $\boldsymbol{\psi}_c^i(\theta^1)$. Finally, the interpolation parameter *p* determines the weighting of each of these shapes according to Eq. 4.1.9. Assuming uniform angular sampling grid is used, where $\Delta\theta^1$ denotes the angular distance between two sampling points, the interpolation parameters are obtained using the following equations:

$$\tau(\theta^1) = \left\lceil \frac{\left|\theta^1 - \theta_1^1\right|}{\tau^1} \right\rceil, \qquad 1 \le \tau \le z^1 \tag{4.3.1}$$

$$j(\theta^1) = \left\lfloor \frac{\left|\theta^1 - \theta_1^1\right| mod\ \tau^1}{\Delta\theta^1} \right\rfloor, \qquad 1 \le j \le n_\theta - 1 \tag{4.3.2}$$

$$p(\theta^1) = \frac{\theta^1 - \theta_j^1}{\theta_{j+1}^1 - \theta_j^1} = \frac{\theta^1 - \theta_j^1}{\Delta\theta^1}, \qquad 0 \le p \le 1 \tag{4.3.3}$$

where $\tau^1 = 2\pi/z^1$ is the angular pitch of the reference gear, $a \bmod b$ gives the remainder after division of a by b, and n_θ is the number of angular sampling points. In the case of a non-uniform sampling grid, Eq. 4.3.2 is replaced by a simple *previous neighbor* algorithm that selects the index of the sampling point θ_j^1 that was last passed by the gear pair.

Having obtained the interpolation parameters, a series of operations is performed to assemble the CDM update equations. The left branch in Fig. 4.10 consists of operations that are expressed solely in terms of the generalized coordinates and the system's mass and stiffness invariants, whereas the right branch operates on the physical grid coordinates and thus requires back-and-forth projections of coordinates and forces.

In the left branch, the stiffness matrix of the gear pair is assembled by interpolating the stiffness invariants using Eq. 4.1.20 and Eq. 4.1.21 based on the current shape number *j* and interpolation parameter *p*. When applying stiffness-proportional damping, the damping matrix is obtained by properly scaling the stiffness matrix based on the lowest eigenfrequencies of the respective gears. The quadratic velocity vector, the *extra* terms (i.e. the terms that are due to the variability of the basis matrix) and the various submatrices of the system's mass matrix are assembled based on the gear pair's (interpolated) mass invariants, as outlined in Section 4.1.3.4. While in general it is not recommended to explicitly compute the inverse of a matrix, the particular structure of the mass matrix $\boldsymbol{M}^i$ combined with the presented MOR scheme allows for an efficient expression of the inverse mass matrix $\boldsymbol{M}^{i^{-1}}$. Using blockwise matrix inversion, the latter is obtained as

$$\boldsymbol{M}^{i^{-1}} = \begin{bmatrix} \boldsymbol{M}_{\theta\theta}^i & \boldsymbol{M}_{\theta f}^i \\ \boldsymbol{M}_{\theta f}^{i\ T} & \boldsymbol{M}_{ff}^i \end{bmatrix}^{-1} = \begin{bmatrix} \breve{\boldsymbol{M}}^{i^{-1}} & -\breve{\boldsymbol{M}}^{i^{-1}} \boldsymbol{M}_{\theta f}^i \boldsymbol{M}_{ff}^{i\ -1} \\ -\boldsymbol{M}_{ff}^{i\ -1} \boldsymbol{M}_{\theta f}^{i\ T} \breve{\boldsymbol{M}}^{i^{-1}} & \boldsymbol{M}_{ff}^{i\ -1} + \boldsymbol{M}_{ff}^{i\ -1} \boldsymbol{M}_{\theta f}^{i\ T} \breve{\boldsymbol{M}}^{i^{-1}} \boldsymbol{M}_{\theta f}^i \boldsymbol{M}_{ff}^{i\ -1} \end{bmatrix} \tag{4.3.4}$$

where $\breve{\boldsymbol{M}}^i$ is a scalar that is defined as

$$\breve{\boldsymbol{M}}^i = \boldsymbol{M}_{\theta\theta}^i - \boldsymbol{M}_{\theta f}^i \boldsymbol{M}_{ff}^{i\ -1} \boldsymbol{M}_{\theta f}^{i\ T} \tag{4.3.5}$$

and where the inverse of $\boldsymbol{M}_{ff}^i$ is equal to:

$$\boldsymbol{M}_{ff}^{i\ -1} = \begin{bmatrix} \boldsymbol{M}_{nn}^i & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{cc}^i \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{I}_{nn} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{M}_{cc}^{i\ -1} \end{bmatrix} \tag{4.3.6}$$

Assuming a single torque was used to compute the contact shapes ($n_T = 1$, see Section 4.1.5), the inverse of $\boldsymbol{M}^i$ is expressed in terms of two scalar inversions and a number of matrix-vector products. Note that in the CS approach, the inverse of $\boldsymbol{M}_{ff}^i$ is precomputed, which further simplifies Eq. 4.3.4.

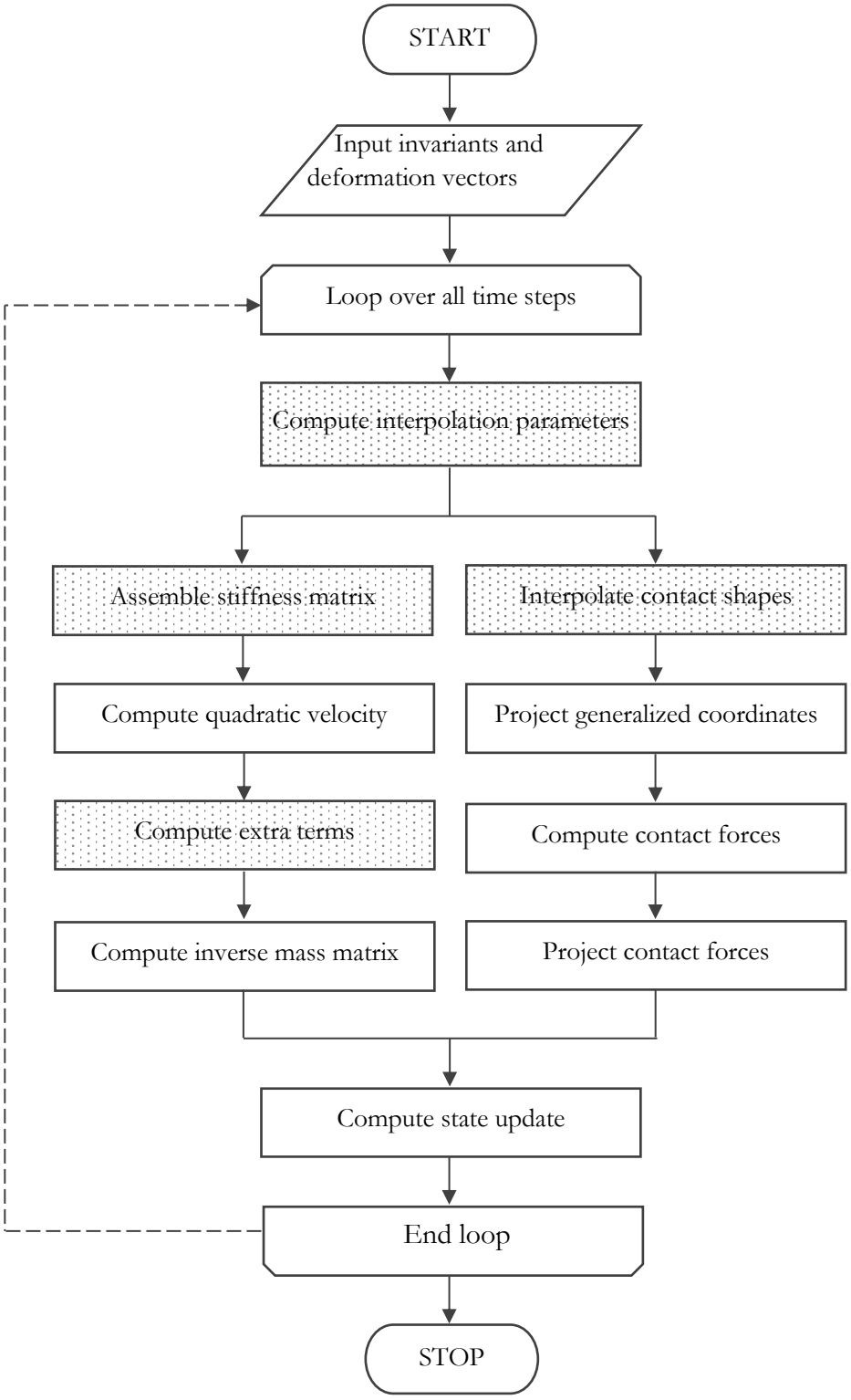


Fig. 4.10 Flow chart of the online phase of the simulation

In the right branch of Fig. 4.10, the generalized contact forces are computed. First the displacement vectors of the active tooth flanks are obtained by selecting the proper rows from the fine-grid eigenvectors and interpolating (and possibly rotating) the proper entries of the contact shapes, based on the three interpolation parameters of Eq. 4.3.1-4.3.3. Then the physical grid coordinates are determined by back-projection of the generalized coordinates using Eq. 4.1.15. Based on these, the physical contact forces are obtained using the sequence of coarse and fine contact detection (see Section 3.2.4), based on a penalty-based Node-To-Surface algorithm (see Section 2.5.3.2). Finally, these contact forces are projected back on the subspace spanned by the current basis matrix, using the equations derived in Section 2.4.3.

Having obtained the system matrices and the vectors of quadratic velocity and generalized contact forces, the CD update equations are constructed. The reference integration scheme used in this dissertation is the fixed-increment explicit CD method with velocities lagging half a time step (see Section 2.6.1.3). The update equations for the dynamic gear contact problem are:

$$\boldsymbol{q}_{n+1}^{i} = \boldsymbol{q}_{n}^{i} + h\dot{\boldsymbol{q}}_{n-1/2}^{i} + h^2 \boldsymbol{M}^{i^{-1}}\left[\boldsymbol{f}_{e}^{i}(t) + \boldsymbol{f}_{v}^{i}\left(\boldsymbol{q}_{n}^{i}, \dot{\boldsymbol{q}}_{n-1/2}^{i}\right) + \boldsymbol{f}_{c}^{i}\left(\boldsymbol{q}_{n}^{i}\right) - \boldsymbol{K}^{i}\boldsymbol{q}_{n}^{i} - \boldsymbol{C}^{i}\dot{\boldsymbol{q}}_{n-1/2}^{i}\right] \tag{4.3.7a}$$

$$\dot{\boldsymbol{q}}_{n+1/2}^{i} = \left(\boldsymbol{q}_{n+1}^{i} - \boldsymbol{q}_{n}^{i}\right)/h \tag{4.3.7b}$$

where the subscript n is the index of the considered time instant, and h is the size of the time increment, which is assumed here to be constant. The integration scheme terminates when all time instants have been processed.

4.3.2 Numerical complexity of the reduced-order integration scheme

In order to assess the influence of the reduced number of DOFs on the numerical complexity of the integration scheme (Eq. 4.3.7), an estimate is made of the number of FLOPs required to process a single time step. In Table 4.1 the FLOP counts for the most computationally expensive operations in Fig. 4.10 are listed. Following the reasoning of [172], a FLOP is assumed to be either a real summation *or* a real multiplication, treating both operations equally. Note that apart from the computation of the physical contact forces, the operations are executed on a gear-per-gear basis; the associated numbers of FLOPs should thus be multiplied by two when assessing the total number of FLOPs. The following symbols are used in Table 3.: n_q, n_f, n_k and n_s respectively denote the number of generalized coordinates, the number of elastic coordinates, the number of retained eigenvectors and the number of static shape vectors (attachment modes or contact shapes) in the reduction basis; N_{fef} and N_{few} respectively represent the number of finite elements along the involute curve and along the width of a tooth, and ε is the contact ratio of the gear pair. Further information on the FLOP counts of the generalized contact force computation can be found in Chapter 8.

Operation	**# FLOPs**
Computation of the elastic and damping forces	$\approx 2\left(2n_f^2 - n_f\right)$
Computation of the quadratic velocity vector	$\approx 8n_f^2 - 4n_f$
Computation of the extra terms	$\approx 8(n_s n_k + n_s^2)$
Computation of the inverse mass matrix (Eq. 4.3.4)	$\approx 13n_f^2 + 4n_f$
Computation of the interpolated contact shapes	$\approx 9\lceil\varepsilon\rceil N_{fef} N_{few}$
Projection of the generalized coordinates	$\approx 3\lceil\varepsilon\rceil n_f N_{fef} N_{few}$
Computation of the physical contact forces	$\approx 20\lceil\varepsilon\rceil N_{fef}^2 N_{few}^2$
Projection of the physical contact forces	$\approx 2\lceil\varepsilon\rceil n_f N_{fef} N_{few}$
Evaluation of the update equations	$\approx 2n_q^2 + 9n_q$

Table 4.1 Number of FLOPs per operation in the explicit CDM integration scheme

In Table 4.2, the estimates of Table 4.1 are applied to four different MOR schemes: the CMS method, which serves as the reference method in this dissertation, the modal transformation method (MOD), the non-interpolated CS approach, and the ICS approach. These methods are applied to the analysis of a helical gear pair having a contact ratio between 2 and 3 (i.e. $\lceil\varepsilon\rceil = 3$) and a FE discretization of $N_{fef} = 30$ and $N_{few} = 12$. The number of retained eigenvectors in the CMS, CS and ICS approaches is $n_k = 50$ per gear, whereas in the MOD approach this number is $n_k = 2000$ per gear [36]. The number of angular sampling points in the (I)CS approach is $n_\theta = 30$, and the number of torques used to compute the contact shapes is $n_T = 1$. As a result, the number of static shape vectors per flexible gear in the CMS, CS and ICS approaches is respectively 3240, 20, and 1.

Although the FLOP counts in Table 4.2 are only indicative, a few important observations can be drawn from these numbers. First, the total number of FLOPs required by the proposed CS and ICS schemes is about 60 times smaller than the number of FLOPs required by the standard CMS method. Second, in the CMS approach only 8% of the overall CPU time is spent on computing the generalized contact forces, whereas in the CS and ICS approach this number increases to 97% and 98.5%, respectively. Clearly, the contact force evaluation becomes the computational bottleneck in the latter two approaches, and further gains in computational efficiency have to be sought here (see Chapter 8). Finally, the computation of the additional terms and the interpolation of invariants and shape vectors results in a negligible increase of the amount of FLOPs, occupying a mere 0.2% of the overall CPU time in the ICS approach.

	# MFLOPs in example			
Operation	**CMS**	**MOD**	**CS**	**ICS**
Comp. of elastic & damping forces	87 (14%)	32 (13%)	0.05 (0.6%)	0.02 (0.24%)
Comp. of quadratic velocity vector	173 (28%)	64 (26%)	0.10 (1.1%)	0.04 (0.49%)
Comp. of extra terms	n.a.	n.a.	n.a.	0.00 (0.01%)
Comp. of inverse mass matrix	281 (45%)	104 (42%)	0.17 (1.9%)	0.07 (0.8%)
Comp. of interp. contact shapes	n.a.	n.a.	n.a.	0.02 (0.2%)
Proj. of generalized coordinates	21 (2.5%)	13 (5%)	0.52 (5.8%)	0.33 (3.9%)
Comp. of physical contact forces	7.78 (1.2%)	7.78 (3%)	7.78 (86.5%)	7.78 (91.6%)
Proj. of physical contact forces	14 (2.3%)	9 (4%)	0.35 (3.8%)	0.22 (2.6%)
Evaluation of update equations	43 (7%)	16 (7%)	0.03 (0.3%)	0.01 (0.1%)
Total	628 (100%)	245 (100%)	8.99 (100%)	8.49 (100%)

Table 4.2 Comparison of FLOP counts in four MOR schemes (comp. = computation; proj. = projection, interp. = interpolated)

4.4 Validation and numerical experiments

4.4.1 Numerical validation of the proposed MOR scheme

In the previous section, the numerical efficiency of the proposed (I)CS approach was assessed by estimating the number of FLOPs required to integrate the reduced-order EOMs. In this section, the numerical accuracy of the (I)CS approach is the main concern. Numerical accuracy is understood here as a relative concept: it is defined with respect to a given reference method that is used to solve the same underlying EOMs. As such, it is not concerned with the physical reality (e.g. in the form of test-rig measurements), nor with mathematical models studying the same phenomena using different governing equations (e.g. nonlinear FE models in Abaqus or semi-analytical lumped parameter models, see Section 3.3 and Appendix N).

The reference method in this dissertation is the CMS approach (Section 2.3.3.1 and 3.2.3) combined with the explicit CD method with fixed increment size (Eq. 4.3.7). The maximal increment size of the CD method is limited by the stability criterion $h \leq 2/\omega_{max}$ (Section 2.6.1.3), where ω_{max} is the largest angular eigenfrequency of the system. The latter can be determined prior to the actual simulation by linearizing the CMS model about a static equilibrium configuration $\{\boldsymbol{q}_0, \dot{\boldsymbol{q}}_0\}$ and solving the corresponding eigenvalue problem $\det(\boldsymbol{K}_t - \omega^2 \boldsymbol{M}) = 0$. In the presence of spectral damping (see Appendix E), the stability criterion of the CD method becomes

$$h \leq \frac{2}{\omega_{max}} \left(\sqrt{1 + \xi_{max}^2} - \xi_{max} \right) \tag{4.4.1}$$

where ξ_{max} is the fraction of critical damping at ω_{max}. In this dissertation stiffness-proportional damping is applied, where the fraction of critical damping at the lowest eigenfrequency ω_{min} is set at $\xi = 0.01$. Hence the above stability criterion can be expressed as

$$h \leq \frac{2}{\omega_{max}}\left(\sqrt{1+\xi^2\left(\frac{\omega_{max}}{\omega_{min}}\right)^2}-\xi\frac{\omega_{max}}{\omega_{min}}\right) \tag{4.4.2}$$

Note that even for lightly damped systems, the factor between brackets can be considerably smaller than 1 if the frequency content of the problem is large.

Aside from stability considerations, the increment size of an explicit method can be limited by ill-conditioning of the update equations (see Eq 4.3.7). In order to avoid ill-conditioning of the mass matrix, the attachment modes in the CMS approach and the contact shapes in the CS approach are mass-orthogonalized using the following Gram-Schmidt procedure:

$$\tilde{\boldsymbol{v}}_i = \boldsymbol{v}_i - \sum_{j=1}^{i-1}\left(\tilde{\boldsymbol{v}}_j \boldsymbol{M} \boldsymbol{v}_i\right)\tilde{\boldsymbol{v}}_j \tag{4.4.3}$$

where $\tilde{\boldsymbol{v}}_i$ and $\boldsymbol{v}_i$ are the *i*-th column vector in the mass-orthogonalized and original matrix of attachment modes and contact shapes, respectively. In the CMS approach, this mass-orthogonalization is performed on a tooth-per-tooth basis, thus preserving the relationship between mass-orthogonalized attachment modes and the teeth they are associated with. This enables an efficient implementation of the CMS approach, where at each time increment the quasi-static contribution of the reduction basis consists merely of the (mass-orthogonalized) attachment modes corresponding to the *active* teeth, as proposed in [43].

In Table 4.3 the simulation parameters of two dynamic simulations are listed. The spur and helical gear pairs that are used in these simulations are described in Section 3.2.5, Table 3.1. Both gear pairs operate at moderate angular velocities: 100 rad/s or roughly 1000 rpm for the spur gear pair and 178 rad/s or roughly 1770 rpm for the helical gear pair. In each simulation an external torque is applied to the pinion while a counteracting viscous torque is applied to the wheel shaft so as to attain a predefined steady-state angular velocity. Both simulations were performed using the four MOR schemes of Table 4.2, where the maximal stable time increment required by the CMS scheme is used for all the schemes. The main parameters of the four ROMs are given in Table 4.4 and Table 4.5. Note that the reported CPU times follow the same trend as observed in Table 4.2, with speed-up factors of approximately 65 for the ICS approach as compared to the CMS scheme. This increase in computational efficiency is solely due to the reduced number of FLOPs required to evaluate the system's update equations (Eq. 4.3.7).

The Dynamic Transmission Errors (DTEs, see Section 3.3) of the spur and helical gear pairs are shown in Fig. 4.11 and Fig. 4.12. Note that the CMS and CS predictions are nearly identical, whereas minor errors are introduced by the ICS scheme near the transition points, i.e. the points where the number of teeth in contact changes (see Section 5.1 for a discussion of this phenomenon). Nevertheless, the relative errors of both the CS and ICS approach (measured with respect to the CMS approach) are below 5% throughout the simulated time interval[24].

Parameters	**Spur**	**Helical**
Number of elements along involute curve	24	24
Number of elements along tooth width	6	12
Number of nodes per tooth flank	175	325
Number of nodes in FE model pinion	90.720	132.132
Critical damping lowest eigenfrequency	1%	1%
Stable time increment	5e-9 s	2.5e-9 s
Applied torque	250 Nm	250 Nm
Steady-state angular velocity pinion	100 rad/s	178 rad/s
Penalty factor	5e8 N/m	5e8 N/m
Simulation time	10 ms	10 ms

Table 4.3 Simulation parameters of the dynamic simulations

Parameter	**CMS**	**MOD**	**CS**	**ICS**
Number of static modes per gear	2025	0	30	1
Number of angular samples in (I)CS	n.a.	n.a.	30	30
Number of torques in (I)CS	n.a.	n.a.	1	1
Number of eigenvectors	50	1000	50	50
Number of DOFs per gear	2076	1001	81	32
Lowest linearized eigenfrequency	3.71e3 Hz	3.82e3 Hz	3.72e3 Hz	3.68e3 Hz
Highest linearized eigenfrequency	3.95e6 Hz	3.29e5 Hz	6.12e5 Hz	3.74e5 Hz
Pre-processing CPU time	1148 s	4935 s	1496 s	1423 s
Processing CPU time	354224 s	112090 s	5480 s	5124 s
Speed-up factor w.r.t. CMS	1	3.2	65	65

Table 4.4 Model parameters of the four ROMs in the spur gear simulation

[24] Note that the relative error, which is used as the primary error metric throughout this dissertation, is very sensitive to phasing differences, which is why two curves with nearly identical amplitudes but small phasing differences can lead to high relative errors.

Parameter	CMS	MOD	CS	ICS
Number of static modes per gear	3900	0	30	1
Number of angular samples in (I)CS	n.a.	n.a.	30	30
Number of torques in (I)CS	n.a.	n.a.	1	1
Number of eigenvectors	50	1000	50	50
Number of DOFs per gear	3951	1001	81	32
Lowest linearized eigenfrequency	3.24e3 Hz	3.37e3 Hz	3.27e3 Hz	3.22e3 Hz
Highest linearized eigenfrequency	4.78e6 Hz	5.28e5 Hz	7.03e5 Hz	5.80e6 Hz
Pre-processing CPU time	1945 s	7648 s	2776 s	2648 s
Processing CPU time	1783547 s	475911 s	28946 s	26075 s
Speed-up factor w.r.t. CMS	1	3.8	61	68

Table 4.5 Model parameters of the four ROMs in the helical gear simulation

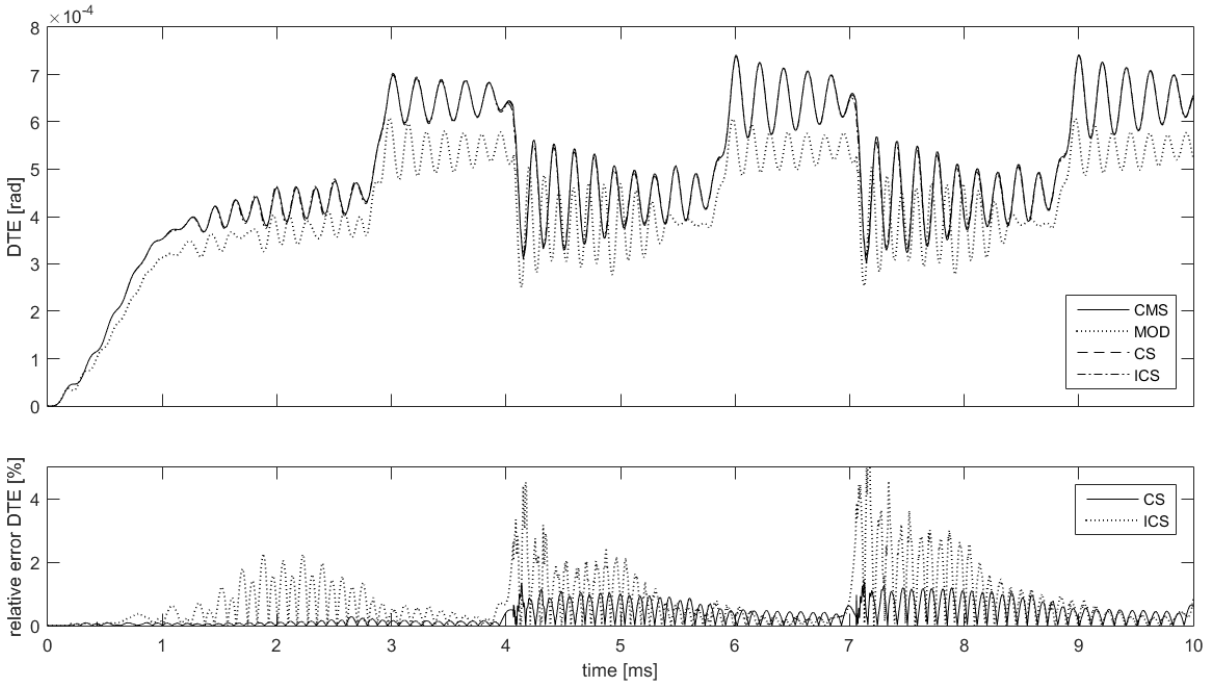


Fig. 4.11 Comparison of the DTE predictions of a spur gear pair by the four MOR techniques

The accuracy of the various ROMs is further assessed by comparing the stress fields predicted by the four MOR methods. Fig. 4.13 shows the evolution of normal stresses at two points along the flank: one near the pitch point (left figure), and one near the root (right figure). Both the CS and ICS approach yield normal stresses that are very close to the prediction of the CMS reference model, with relative errors of less than 4% in the critical zone. Notwithstanding its acceptable performance in predicting integrated quantities such as DTE, the modal transformation method (MOD) fails to accurately predict the gear pair's stress field. Despite the large number of eigenvectors in its reduction basis, the modal ROM is not capable of resolving local contact stresses (see left figure). To compensate for this lack of local deformation, there is more global deformation in the modal ROM, as is evidenced by the higher root stresses (see right figure).

Similar observations are made based on Fig. 4.14, where the Von Mises stress predictions of the four MOR schemes are plotted at a particular instant (t = 2 ms) in the simulation.

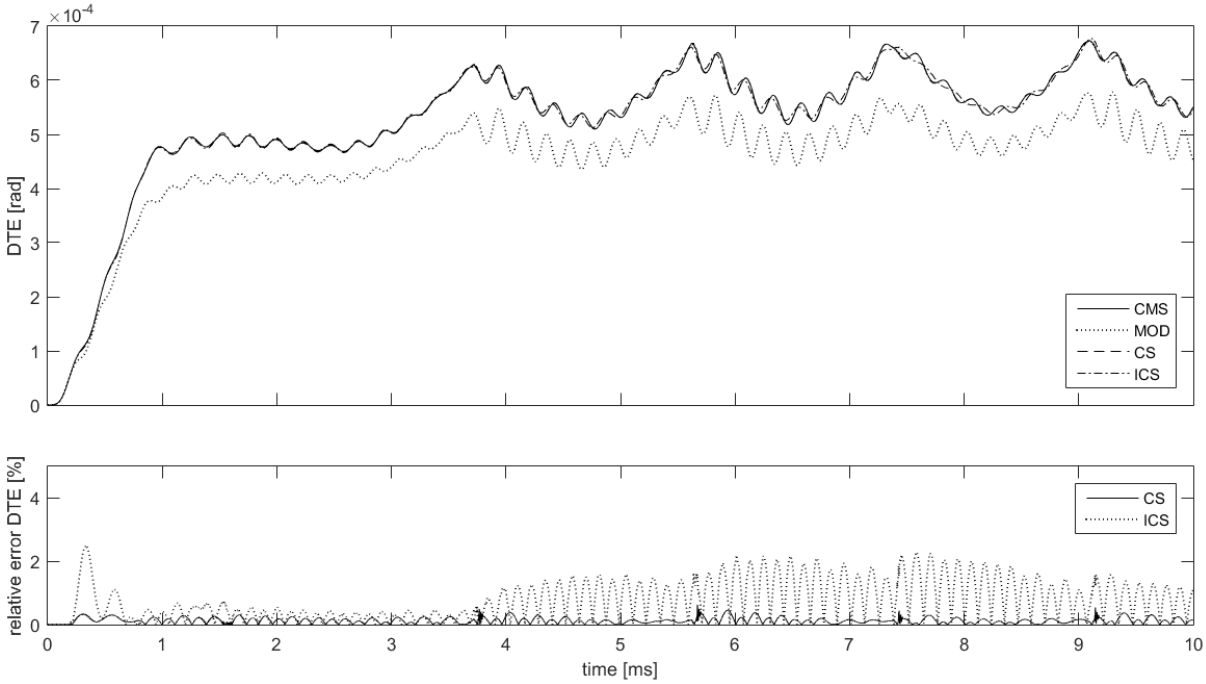


Fig. 4.12 Comparison of the DTE predictions of a helical gear pair by the four MOR techniques

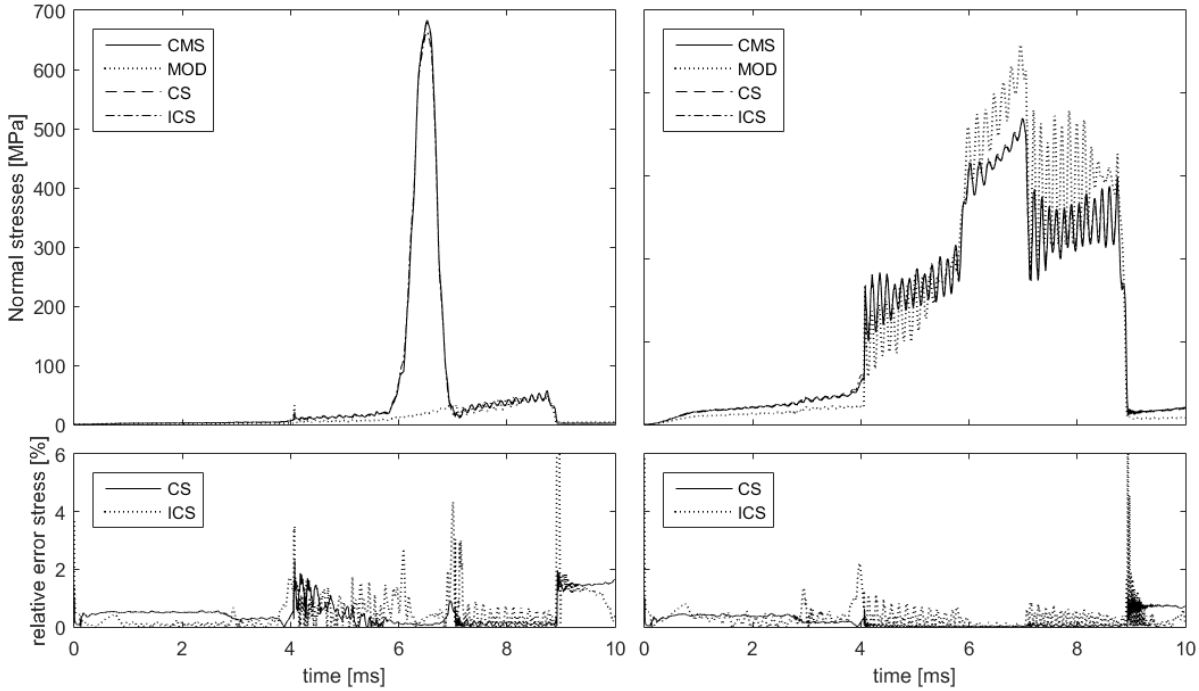


Fig. 4.13 Comparison of the contact stress predictions of a spur gear pair by the four discussed MOR techniques

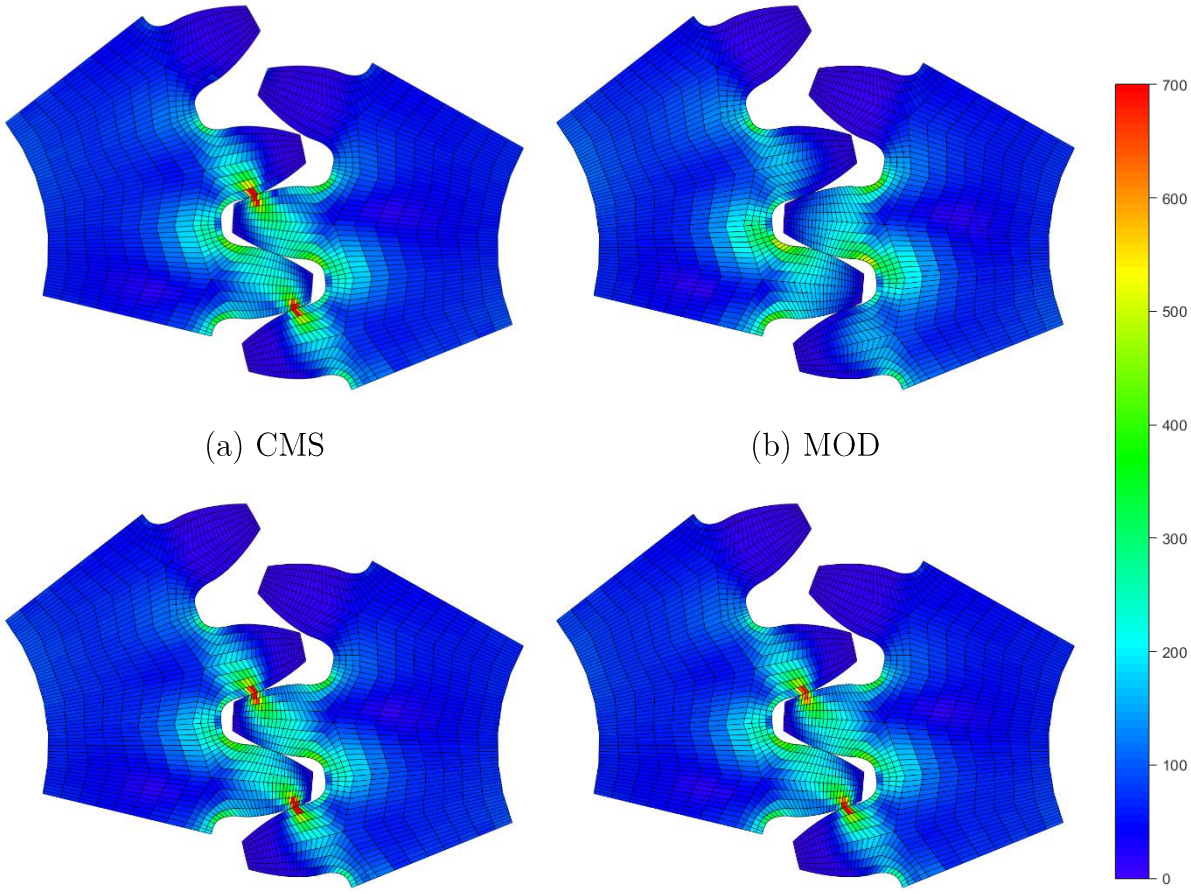


Fig. 4.14 Comparison of the Von Mises stress predictions of a spur gear pair at t = 2 ms by the four discussed MOR techniques

Summarizing, the (I)CS approach yields displacement and stress results with similar accuracy as the standard CMS method, yet requiring less than 2% of the computational effort (when applying the same numerical integrator). The full potential of the method in terms of computational efficiency will however only be revealed in Chapter 7, where the ICS approach is combined with an optimal integration scheme.

4.4.2 Numerical investigation of the additional terms in the ICS approach

In the previous section, the reduced-order EOMs of the ICS approach were solved taking into account all the terms that are due to the variability of the reduction basis (see Section 4.1.3.3). In this section, the influence of these additional terms on the solution accuracy is evaluated in function of the gear pair's operating conditions.

To this end, the spur gear pair of Table 3.1 is simulated at three different operating speeds corresponding to the low-speed (250 rpm), moderate-speed (1000 rpm) and high-speed regime (5000 rpm), see Fig. 4.15-Fig. 4.18. At each of these operating speeds, the DTE is computed while respectively neglecting the additional terms (left figures), the quadratic velocity forces (middle

figures), and both (right figures). These predictions are then compared with the DTE predictions of the *all-inclusive* ICS model.

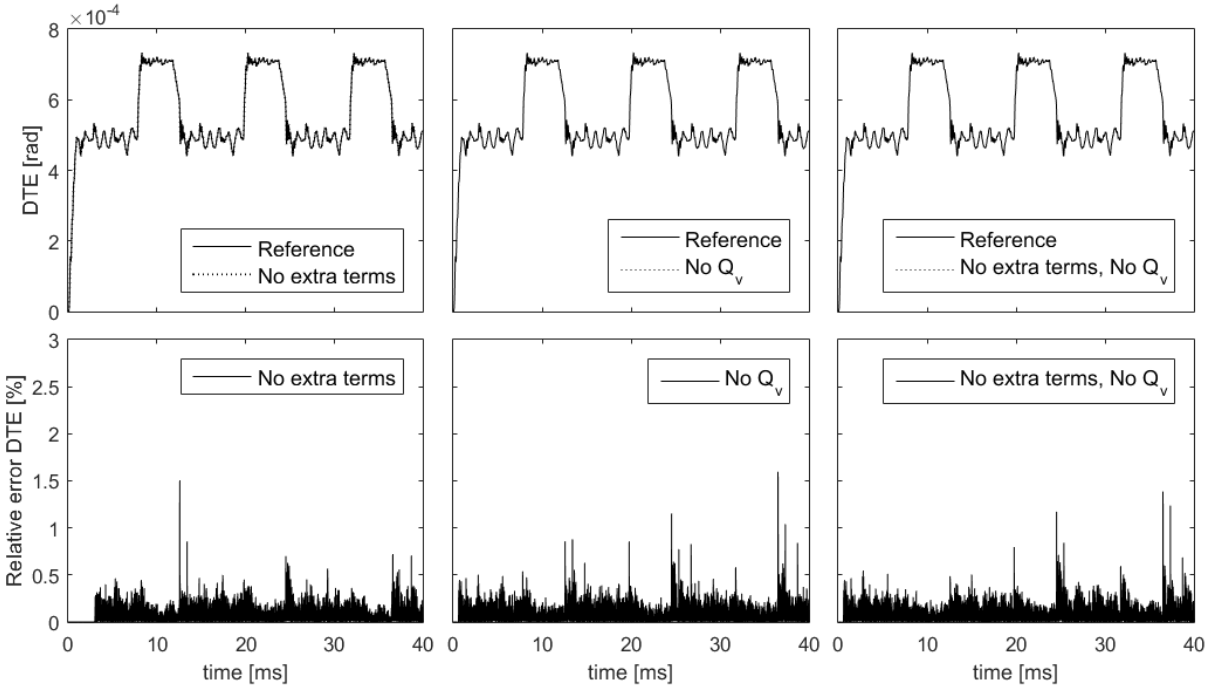


Fig. 4.15 Influence of extra terms and quadratic velocity terms on the accuracy of the predicted DTE at low speed

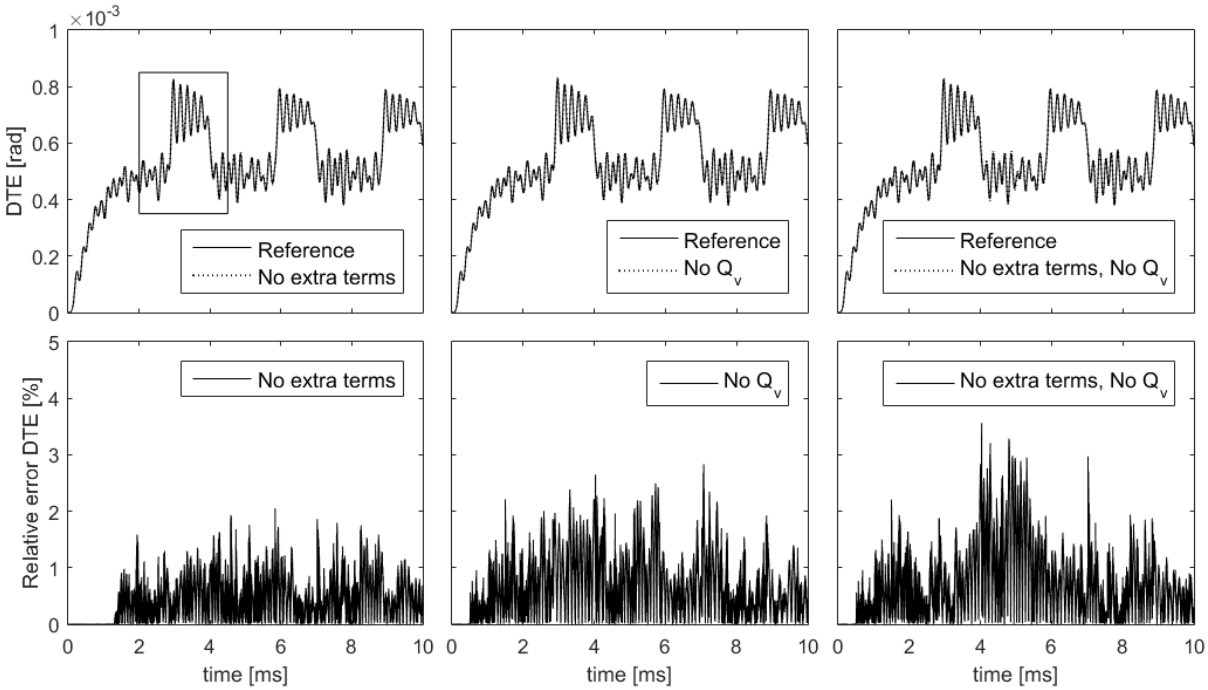


Fig. 4.16 Influence of extra terms and quadratic velocity terms on the accuracy of the predicted DTE at moderate speed

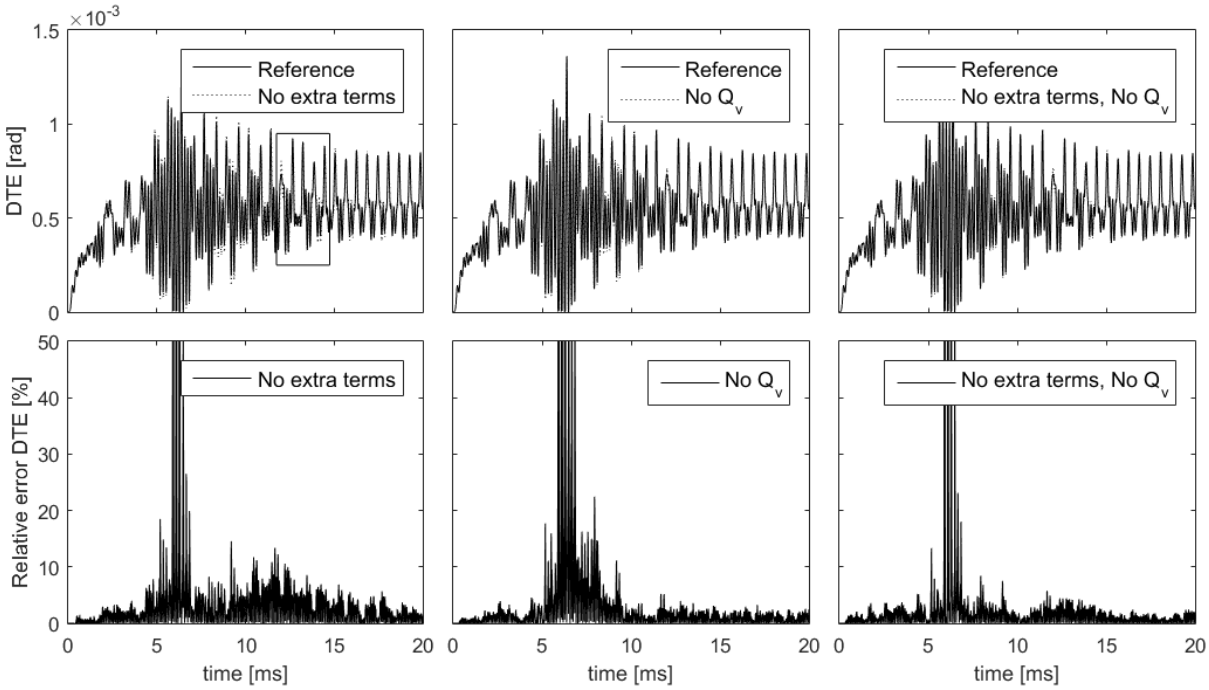


Fig. 4.17 Influence of extra terms and quadratic velocity terms on the accuracy of the predicted DTE at high speed

Note that as the high-speed gear pair accelerates during the first 10 ms of the simulation (Fig. 4.17), the gear pair's meshing frequency crosses the lowest eigenfrequency of the gear pair, causing significant DTE oscillations. While the amplitudes of these oscillations are predicted well by all three models, small phasing differences cause substantial relative errors, which vanish as the angular velocity is increased beyond the excited eigenfrequency.

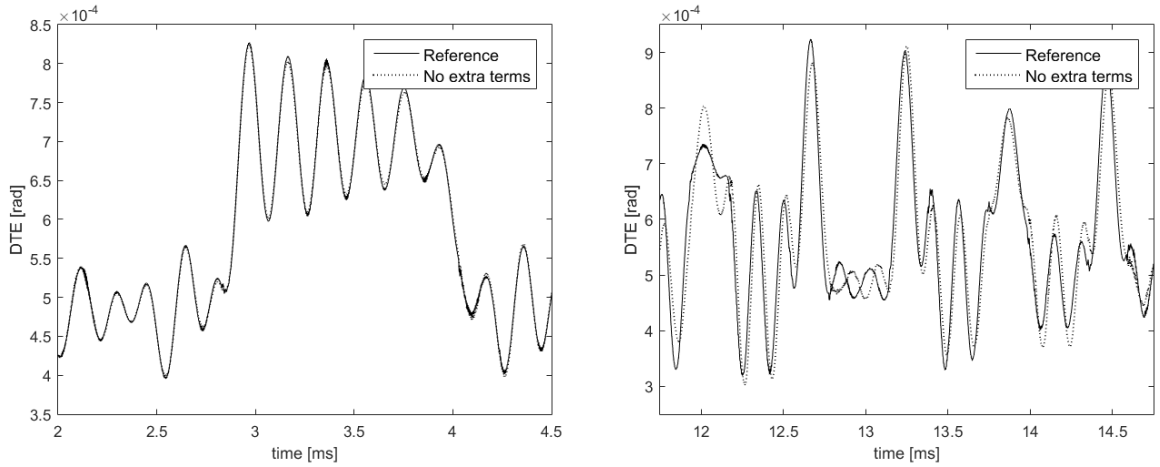


Fig. 4.18 Detail graphs from the upper left graphs in Fig. 4.16 (left) and Fig. 4.17 (right)

The simulation results above indicate that the influence of the additional terms is of the same order of magnitude as the quadratic velocity vector, and only affects the solution accuracy at elevated angular velocities (see right graph in Fig. 4.18). Example applications where these terms might become important are lightweight and very high-speed (~10000 rpm or more) gearboxes. While it

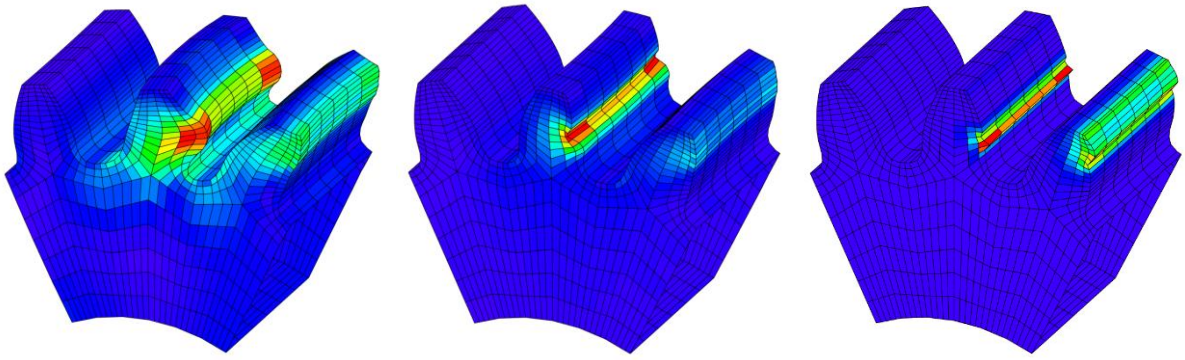
is safe to say that for engineering purposes (e.g. gear microgeometry design) the influence of the additional terms can be neglected in most analyses, it should be noted that the relative cost of these terms is anyway limited (only 0.2% of the overall cost, see Table 4.2).

4.4.3 Comparison with the Proper Orthogonal Decomposition method

As a final numerical experiment, the CS approach is compared with the Proper Orthogonal Decomposition (POD) method, with which it has some similarities (see Section 2.3.3.2). In the POD method, the EOMs are projected on the subspace spanned by the dominant left-singular vectors of a matrix of solution snapshots that are collected during a full-order *training* simulation. The (I)CS approach can be seen as a way to eliminate the expensive training phase of the POD method by making certain assumptions about what the dominant deformation patterns of the gears will be. More specifically, the CS approach assumes that these dominant patterns closely resemble a linear combination of eigenmodes and static contact deformation patterns computed at various angular configurations.

The POD model used in this investigation is derived from a snapshot matrix of back-projected displacement vectors that are collected during a dynamic gear contact simulation based on the CMS approach (see Table 4.3, column 'sim 1' for the model parameters). A training simulation of 4 ms is conducted – covering one angular pitch of rotation – with snapshots collected each 0.005 ms, yielding a total of 800 snapshots. In order to promote consistency with the CS approach (and to enable full-revolution simulations, see further), the eigenmode contribution is eliminated from the snapshot matrix prior to its singular value decomposition. Furthermore the number of retained Proper Orthogonal Modes (POMs, see Fig. 4.19) is set to 30, being the number of contact shapes used in the CS-based ROM. These POVs are augmented with 50 eigenvectors, yielding a total of 80 generalized elastic DOFs per gear. Analogous to the (I)CS approach, full-revolution simulations using the POD model are enabled by proper rotation of the POMs each time the gear pair travels through an angular pitch (see Section 4.1.4).

Fig. 4.20 shows the predicted DTE of the spur gear pair of Table 3.1 that is subjected to a torque of 250 Nm, which is the torque used for constructing both the CS and POD models. In order to excite the CS and POD models outside their comfort zone, a torque ripple with an amplitude of 100 Nm is applied between t = 13 ms and t = 15 ms (see top figure in Fig. 4.20). Between t = 0 ms and t = 4 ms (i.e. the span of the POD training simulation), the POD model outperforms the CS model. From t = 4 ms onwards, the results of the POD model progressively degrade, with the CS model clearly outperforming the POD model during and right after the torque ripple (see Fig. 4.21). Note that the CS model is remarkably robust with respect to torque variations, despite it being constructed based on a single torque value (see also Section 4.1.5; more experiments related to the performance of the ICS approach in the case of largely varying torques can be found in [173]).



(a) 1st POM (b) 2nd POM (c) 5th POM

Fig. 4.19 Three Proper Orthogonal Modes (POMs) of the spur gear

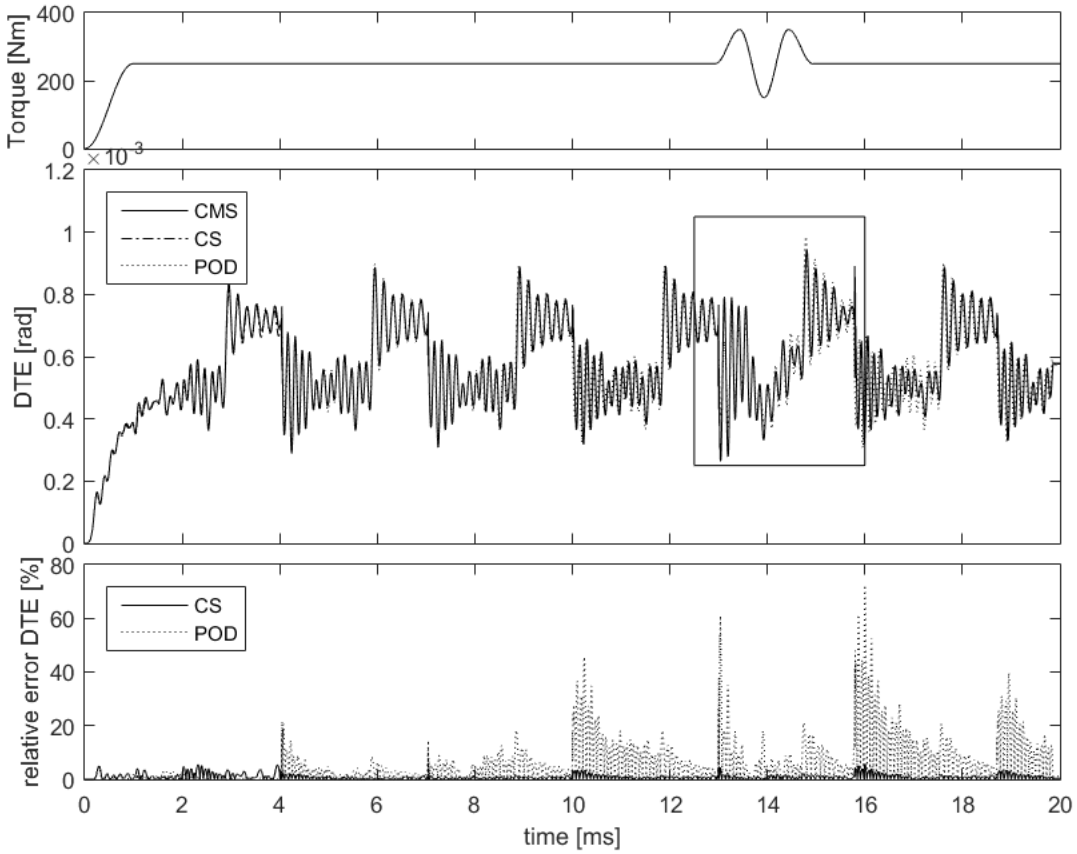


Fig. 4.20 Comparison of the DTE predictions of a spur gear pair during a torque ripple by the CMS, CS and POD method

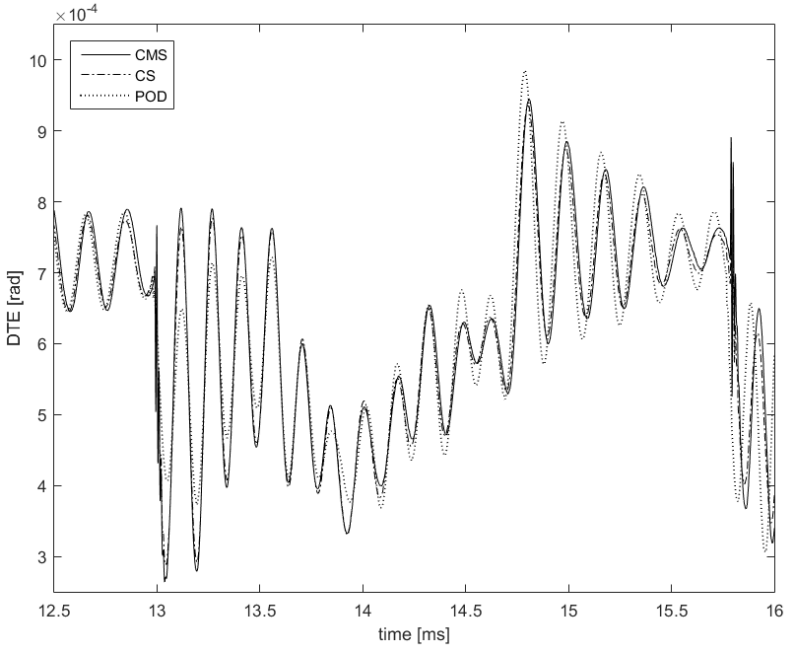


Fig. 4.21 Detail from the DTE curve of Fig. 4.20

Despite the slightly better overall performance of the CS model in the torque ripple experiment, the POD model does a more than decent job at predicting the DTE of the spur gear pair. The added value of the (I)CS approach is that it provides accurate DTE predictions without the need of an expensive training phase. Constructing the POD model took 22578s, whereas the CS model was constructed in less than 480s on the same machine. In addition to its superior training efficiency, the natural frequencies of the contact shapes are substantially larger than the frequencies of the POMs, resulting in larger stable time increments for the CS scheme. This is demonstrated in Fig. 4.22, where the time increments selected by an error-controlled central difference integrator are compared for the CS and POD methods. While the cost per explicit time increment is the same for both methods, the CS method allows using time increments that are on average 6 times larger. In the ICS method, instead, time increments for the same integrator are roughly ten times larger (see Fig. 7.3).

4.5 Summary and conclusion

In this chapter, a novel Model-Order Reduction (MOR) method was presented for the efficient solution of Finite-Element (FE) based contact problems in flexible multibody dynamics. The method was introduced in the context of dynamic gear contact analysis, but can readily be applied to the simulation of bearings, cam-follower mechanisms, road-tire interactions, and other machine components and mechanisms that are characterized by a certain degree of predictability in the way the components interact. The presented method belongs to the class of projection-based MOR methods and can be seen as a parametric variant of the Component Mode Synthesis (CMS) approach. The reduction basis of the method is composed of a truncated set of eigenmodes

augmented with a configuration-dependent set of so-called contact shapes, which are the component-level deformation patterns of the mechanism obtained from a series of a priori static contact analyses. The computation of these contact shapes during the offline phase of the simulation is accelerated through the use of an intermediate Reduced-Order Model (ROM), an analytical contact Jacobian, and a hybrid mesh topology. The variability of the set of contact shapes gives rise to additional terms in the equations of motion, but these were shown to be negligible in most analyses, and cheap to evaluate otherwise. The method's computational complexity per explicit time increment was shown to be two orders of magnitude smaller than standard CMS methods, whereas a relative error of no more than 5% was observed in all of the performed dynamic analyses.

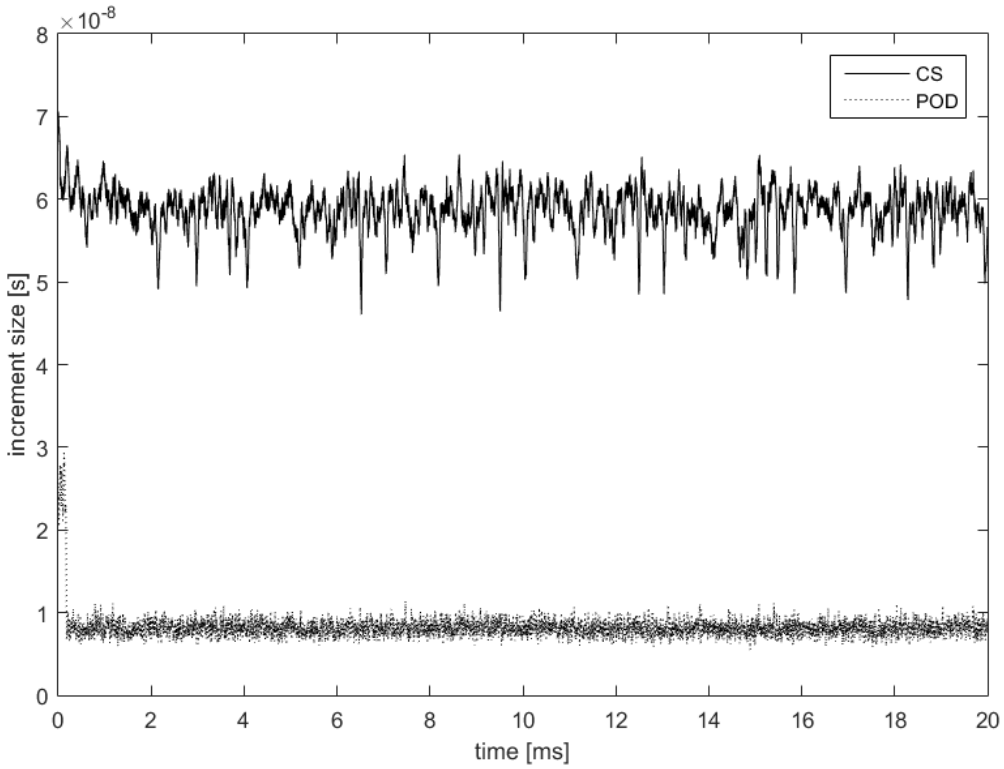


Fig. 4.22 Comparison of time increments used by the adaptive central difference method

Chapter 5

Methodological improvements of the novel MOR method

The novel Model Order Reduction (MOR) method introduced in Chapter 4 is optimal in the sense that it uses the minimal number of generalized elastic coordinates to capture the deformation effects that are characteristic of sliding contact interactions. At the same time, the algorithmic complexity of the method is comparable to that of standard Component Mode Synthesis (CMS) methods, especially when linear interpolation is used (Eq. 4.1.9) and the influence of the additional terms in the Equations Of Motion (EOMs) is neglected (see Section 4.4.2). As such, the method offers a unique combination of accuracy and computational efficiency that is hard to surpass.

Nonetheless, the method suffers from a number of shortcomings that might become apparent under certain circumstances, such as when the angular discretization of the gear pair configuration is necessarily coarse or when the gears experience substantial dynamic excitation, causing the actual contact conditions to differ significantly from the static conditions. In this chapter, the influence of these shortcomings is investigated and a number of modifications to the method of Chapter 4 are introduced. The *advanced* version of the method eliminates the aforementioned shortcomings by slightly increasing the number of Degrees Of Freedom (DOFs) and introducing a novel, physics-based interpolation scheme.

The structure of this chapter is as follows: Section 5.1 investigates the two main shortcomings of the Interpolated Contact Shapes (ICS) approach that was introduced in Chapter 4 (Section 4.1.2). In Section 5.2, a variant of the *contact shape vector* is introduced that forms the basis of an improved reduced-order gear contact model, while Section 5.3. deals with the accurate, physics-based interpolation of these shape vectors. Finally, in Section 5.4 the improved method is validated through comparison with the Full-Order Model (FOM).

5.1 Shortcomings of the interpolated contact shapes approach

The ICS approach was introduced in Section 4.1.2 and is based on the following definition of the basis matrix:

$$\boldsymbol{V}_{ICS}^{i} = [\boldsymbol{\Phi}^{i} \quad \boldsymbol{\psi}_{c}^{i}] = [\boldsymbol{\Phi}^{i} \quad \boldsymbol{\psi}_{c}^{i}(\theta^{1})] \in \mathbb{R}^{N\times(n_k+1)} \tag{5.1.4}$$

where $\boldsymbol{\Phi}^{\mathrm{i}}$ represents a constant set of eigenvectors and $\boldsymbol{\psi}_{c}^{i}(\theta^{1})$ is obtained by linearly interpolating among two adjacent contact shapes $\boldsymbol{\psi}_{\mathrm{c,j}}^{\mathrm{i}}$ and $\boldsymbol{\psi}_{\mathrm{c,j+1}}^{\mathrm{i}}$, *i.e.*

$$\boldsymbol{\psi}_{c}^{i}(\theta^{1}) = (1-p)\boldsymbol{\psi}_{c,j}^{i} + p\boldsymbol{\psi}_{c,j+1}^{i} \tag{5.1.5}$$

The interpolation parameter p is defined in Eq. 4.1.9 and depends linearly on the angular configuration of the gear pair. The two possible shortcomings of this approach are related to the way contact shapes are defined (see Eq. 4.1.4) and the way they are interpolated (Eq. 5.1.5), respectively. The following two sections discuss these possible shortcomings and the conditions in which they may appear in detail.

5.1.1 The interdependence of quasi-static tooth deformations

In Section 4.1.1, a contact shape of a gear is defined as the gear's deformation pattern resulting from a static contact analysis at a particular angular configuration and subjected to a particular torque. By projection of the gear pair's EOMs on a basis that is composed of these contact shapes, the quasi-static behavior of one tooth of the gear is inherently coupled to the deformation of the other teeth in mesh.

One case where this inherent coupling of the quasi-static deformations of teeth can be troublesome is when the gears experience substantial dynamic excitation, causing the actual contact conditions to differ significantly from the static conditions. Consider for example the eigenmode that is displayed in Fig. 5.1. In this vibration mode, adjacent teeth oscillate in opposite in-plane directions. When this mode is excited, it might result in a redistribution of tooth contact forces among the teeth in contact, causing the tooth deformations to behave differently than in static conditions. This redistribution will be partially accommodated for by the set of eigenvectors in the basis matrix, but not entirely: local tooth deformations, for example, are solely expressed in terms of interpolated contact shapes.

Another, more frequent case where the coupling of tooth deformations might jeopardize the solution accuracy is when a transition occurs between the number of teeth in contact. Consider a pair of spur gears having a contact ratio between 1 and 2, meaning the gear pair alternates between having 1 and 2 pair of teeth in contact. The evolution of the number of teeth in contact and the load shared by the various teeth is best represented by a Load Sharing Factor (LSF) plot, as shown in Fig. 5.2.

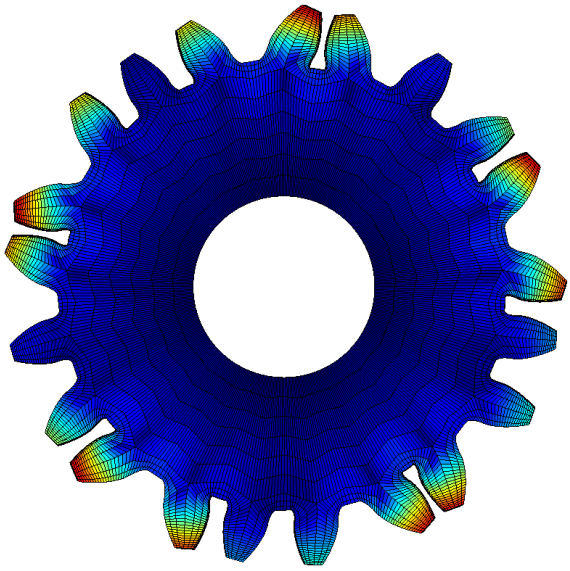


Fig. 5.1 A spur gear eigenmode in which adjacent teeth oscillate in opposite directions

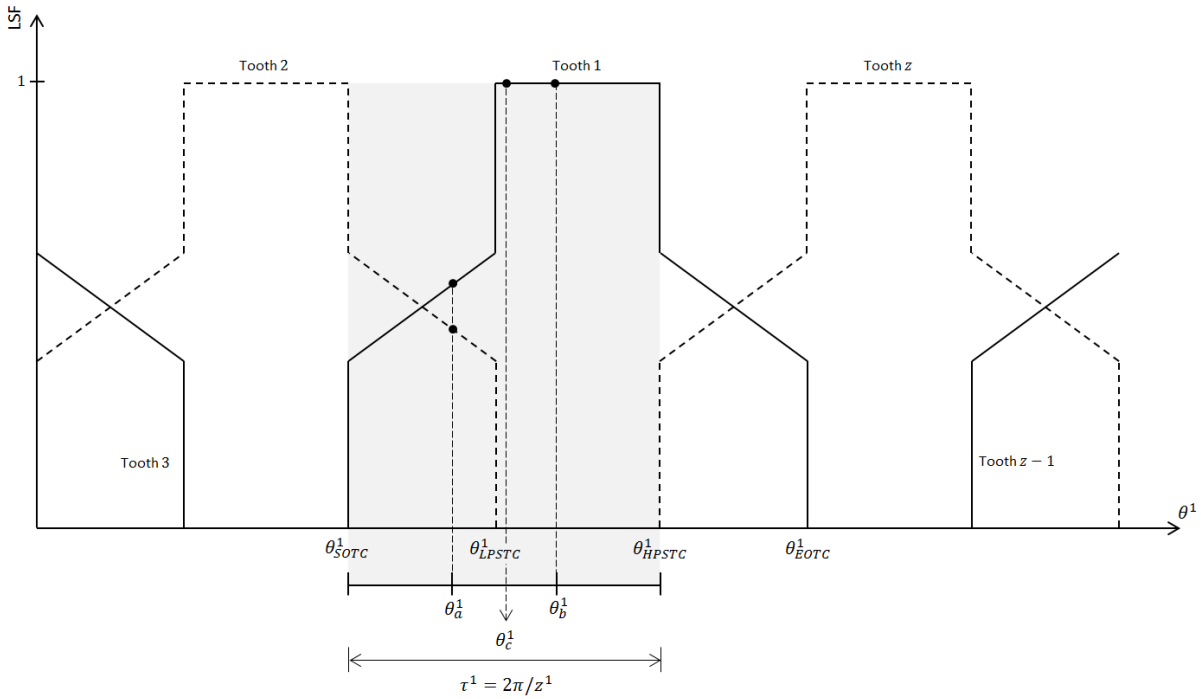


Fig. 5.2 A typical Load Sharing Factor (LSF) plot of a spur gear pair having a contact ratio between 1 and 2

The LSF measures the fraction of the applied torque that is carrier per teeth; if only one pair of teeth is in contact, the teeth in question each have a LSF of 1. In Fig. 5.2, the LSF is plotted as a function of the reference gear's rotation angle θ^1. Four important angles are identified with reference to Tooth 1:

- the Start Of Tooth Contact (SOTC) angle θ^1_{SOTC},
- the Lowest Point of Single Tooth Contact (LPSTC) angle θ^1_{LPSTC},

- the Highest Point of Single Tooth Contact (HPSTC) angle θ^1_{HPSTC}, and
- the End Of Tooth Contact (EOTC) angle θ^1_{EOTC}.

The shaded area in Fig. 5.2, which spans one angular pitch τ^1 and extends from the angle at which Tooth 1 starts to engage (θ^1_{SOTC}) to the angle at which Tooth z starts to engage (θ^1_{HPSTC}), is the angular interval that is discretized when computing the contact shapes (see Section 4.1.4). Counting from the left, the last angular configuration where Teeth 1 and 2 are in contact is denoted by θ^1_a, while the first angular configuration where only Tooth 1 is loaded is denoted by θ^1_b. The contact shapes corresponding to these angles are shown in Fig. 5.3a and Fig. 5.3b, respectively. When the gear pair is at an angular configuration midway between θ^1_a and θ^1_b, it is clear from Fig. 5.2 that at this point Tooth 2 is no longer in contact. However, the interpolated contact shape corresponding to θ^1_c contains some tooth bending and local deformation near the tip of Tooth 2, as can be seen in Fig. 5.3c. This of course does not make physical sense. While the influence of this error may be minimized by locating the angles θ^1_a and θ^1_b as close as possible to the transition point, the exact location of this point under dynamic conditions might be hard to define.

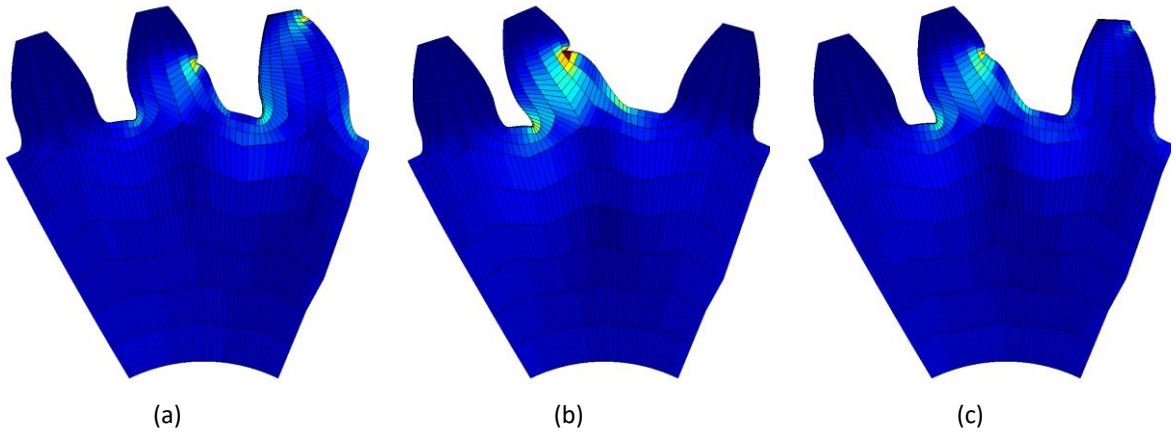


Fig. 5.3 Contact shapes corresponding to the LSF plot of Fig. 5.2, with a) the contact shape corresponding to θ^1_a, b) the contact shape corresponding to θ^1_b, c) the interpolated contact shape corresponding to θ^1_c. Note that the tooth in the middle is tooth 1, with teeth 2 and z to the right and left of it, respectively.

5.1.2 The unphysical interpolation of local deformations

The second shortcoming of the ICS approach is that the interpolation of contact shapes only yields the desired interpolated shape in the limit of a fine angular discretization. While the *global* deformation of a gear is well captured by the interpolation scheme no matter how coarse the angular discretization (owing to the principle of Saint-Venant, see Appendix N.2), the same is not true for the local deformation. The latter is illustrated in Fig. 5.4, where the curves represent the local displacements u of a contact surface that extends along the x-axis. The top curve $\boldsymbol{\psi}_a$ represents the local contact shape corresponding to a contact load applied at $\boldsymbol{x}_a$, while the second

curve $\boldsymbol{\psi}_b$ corresponds to a load applied at x_b. Using the linear interpolation approach, the response of the contact surface to a load that acts at x_c midway between x_a and x_b would be described in terms of the interpolated local contact shape $\boldsymbol{\psi}_c$ (third curve in Fig. 5.4), which differs considerably from the intended local contact shape $\boldsymbol{\psi}_{c,ideal}$ (bottom curve in Fig. 5.4). Clearly the deformation pattern of $\boldsymbol{\psi}_c$ does not make physical sense.

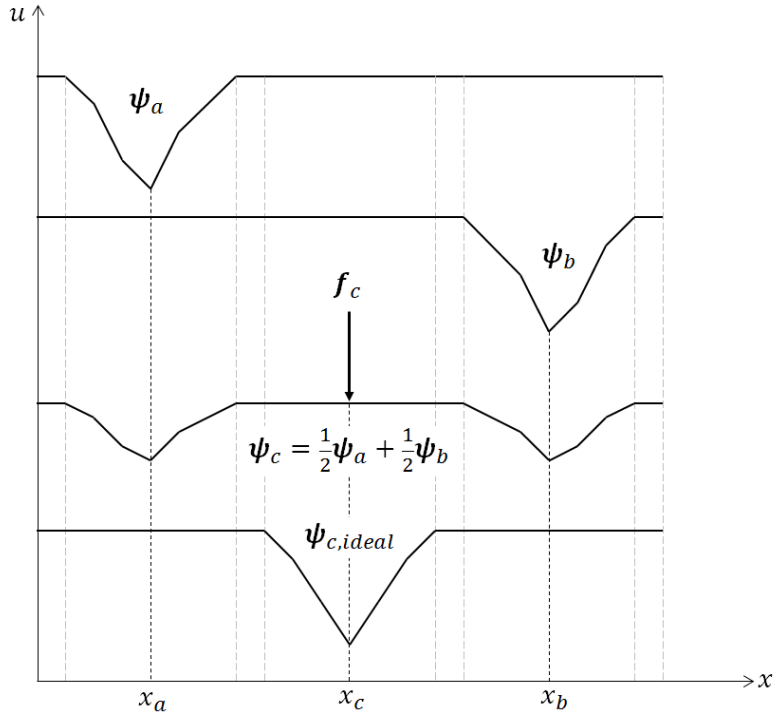


Fig. 5.4 Schematic representation of the linear interpolation of local deformations gone wrong

The same effect is shown in Fig. 5.5, where the linear interpolation of two adjacent but widely spaced contact shapes $\boldsymbol{\psi}_a$ and $\boldsymbol{\psi}_b$ results in an interpolated contact shape $\boldsymbol{\psi}_c$ that bears little resemblance to the correct contact shape $\boldsymbol{\psi}_{c,ideal}$, with the exception of the global tooth bending. The result of this local error is that the contact stiffness of the gear is increased when contact occurs in-between two adjacent sampling points θ_j^1 and θ_{j-1}^1.This effect becomes apparent in the Static Transmission Error (STE) of the reduced-order gear pair: Fig. 5.6 shows the STE curves of a spur gear pair obtained by the interpolated contact shapes approach where subsequently 10 and 40 equidistant sampling points are used. While these curves coincide at the common sampling points, there is a clear unphysical stiffening effect in-between sampling points when a coarse sampling grid is used, which can be especially troublesome in dynamic simulations.

5.1.3 Concluding remarks

In this section on the shortcomings of the ICS approach, the author deliberately impersonated the Devil's advocate. It took a series of far-fetched thought experiments and ill-considered simulation

settings to uncover these shortcomings, which were displayed in their worst possible form. Without disrupting this chapter's storyline and notwithstanding the possibility of these shortcomings to surface under certain circumstances, it is expected that these occasions are rare as long as reasonable simulations parameters are selected.

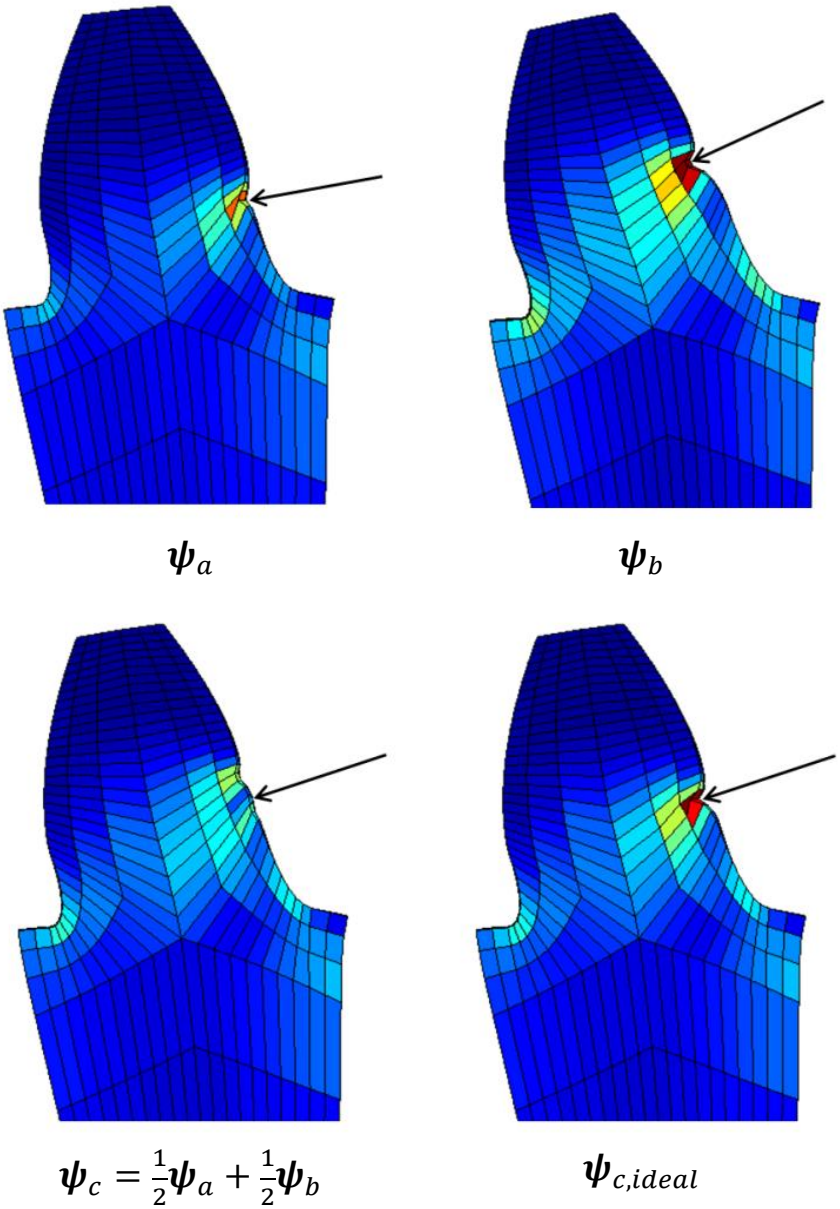


Fig. 5.5 Linear interpolation of adjacent but widely spaced contact shapes

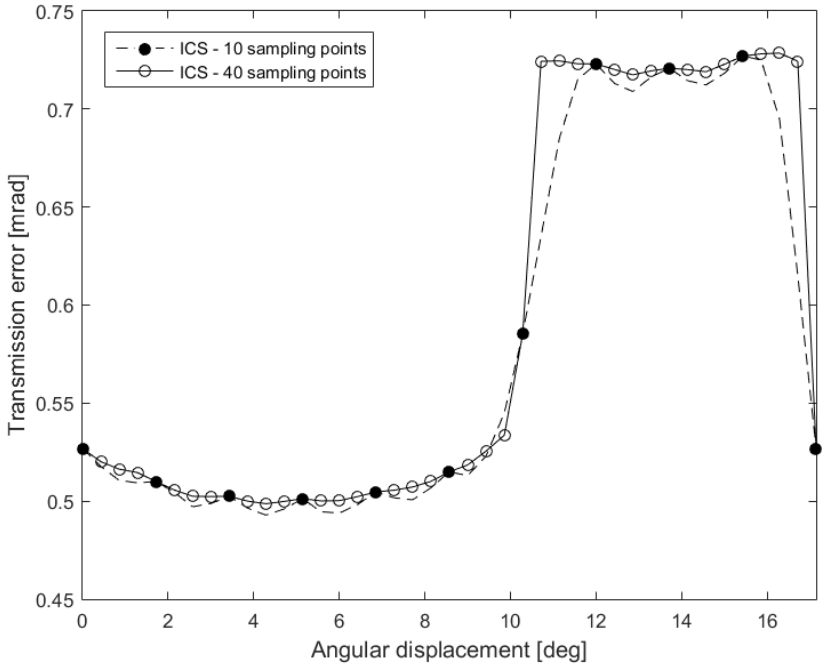


Fig. 5.6 Comparison of the static transmission error computed using the interpolated contact shapes approach with different numbers of contact shapes

With regard to the interdependence of tooth deformations (Section 5.1.1), it is expected that the dynamic excitation of natural vibration modes can only significantly alter contact force distributions in the case of lightly-loaded gears. Such gears are encountered in the numerical analysis of gear rattle, i.e. a noise phenomenon that is dominated by inertial and damping effects – less so by geometrical effects – and that is typically studied using lumped instead of distributed parameter models. Furthermore, the erroneous behavior of the interpolated contact shapes approach when teeth enter or leave the gear mesh can be alleviated by proper distribution of the sampling points, or simply by using a sufficiently fine sampling grid (see Section 4.1.4). Finally, it should be noted that as the contact ratio of the gear pair or the helix angles of the gears increase, the transition effect is less pronounced as the one shown in Section 5.1.1.

The unphysical interpolation of local displacements in the interpolated contact shapes approach is indeed troublesome when a coarse sampling grid is used, which can e.g. be motivated by limited pre-processing resources. However, it is clear from Fig. 5.7 (compare to Fig. 5.4) that as the spacing of sampling points (in casu: x_a and x_b) is reduced, linearly interpolating the contact shapes will yield an interpolated contact shape $\boldsymbol{\psi}_c$ that closely resembles the intended contact shape $\boldsymbol{\psi}_{c,ideal}$ (indicated in dashed lines). In a finite element context, this can be quantified as follows: as long as the centers of pressure of two contact shapes are within one element length of each other, the interpolated contact shape will be virtually indistinguishable from the intended contact shape. This leads to the following rule of thumb: the number of sampling points within the reference interval (see e.g. the shaded area in Fig. 5.2) should be roughly equal to the number of elements along the tooth flank to achieve optimal interpolation.

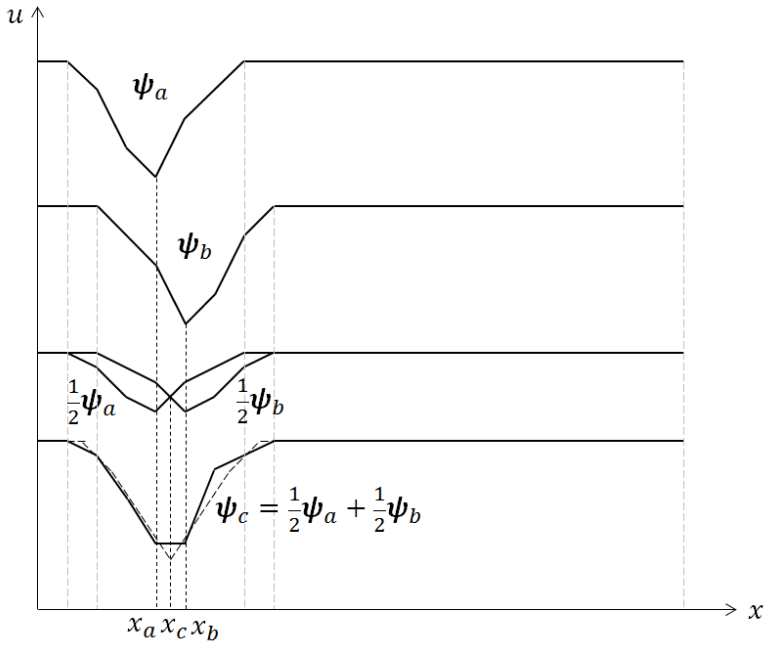


Fig. 5.7 Schematic representation of the linear interpolation of local deformations gone right

5.2 Model reduction based on interpolated *tooth* contact shapes

5.2.1 Definition and computation of tooth contact shapes

In order to eliminate the error related to the interdependence of quasi-static tooth deformations (Section 5.1.1), this section introduces a modified version of the ICS approach, called the Interpolated Tooth Contact Shapes (ITCS) approach. As its name suggest, the ITCS approach replaces the interpolation of *contact shapes* in the basis matrix $\boldsymbol{V}$ by the interpolation of *tooth contact shapes*. In analogy with the contact shapes of Chapter 4, a tooth contact shape is defined as the deformation pattern of a gear obtained from a static contact analysis *considering only a single pair of teeth is in contact* (see Fig. 5.8). The procedure to compute these shapes is the same as outlined in Section 4.1.1 (Eq. 4.1.4), with two minor differences: 1) contact forces are computed only for the selected tooth pair, leaving the remaining teeth in mesh unloaded, and 2) the torque applied to the driving gear is equal to the torque that is carried by the selected tooth pair during normal operation of the gear pair. The latter can be determined from the gear pair's LSF plot or by using a simple lumped-parameter gear contact model (see e.g. Appendix N). The LSF value of a spur gear tooth having a contact ratio between 1 and 2 is mathematically expressed in function of the rotation angle θ^1 by the ISO Standard [15]:

$$LSF(\theta^1) = \begin{cases} \frac{1}{3} + \frac{1}{3}\frac{tan\,\theta^1}{tan\,\theta^1_{LPSTC}} & \theta^1_{SOTC} \le \theta^1 \le \theta^1_{LPSTC} \\ 1 & \theta^1_{LPSTC} \le \theta^1 \le \theta^1_{HPSTC} \\ \frac{1}{3} + \frac{1}{3}\frac{tan\,\theta^1_{EOTC} - tan\,\theta^1}{tan\,\theta^1_{EOTC} - tan\,\theta^1_{LPSTC}} & \theta^1_{HPSTC} \le \theta^1 \le \theta^1_{EOTC} \end{cases} \quad (5.2.1)$$

Note that this formula is only approximately correct, as the influence of microgeometry and the applied torque is left out of the equation. However, an approximate value for the gear's LSF suffices for the proper computation of tooth contact shapes, as will be shown later.

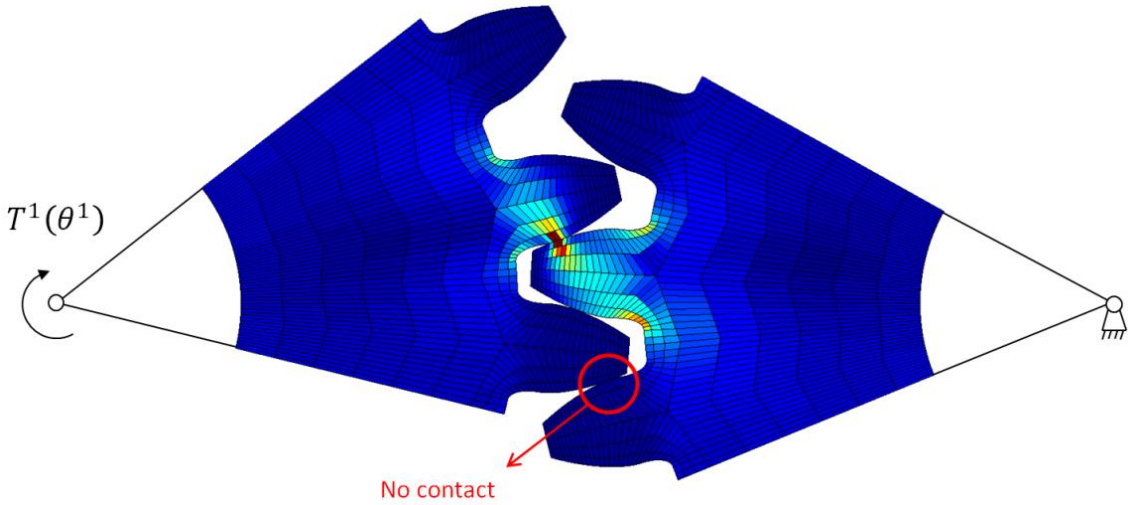


Fig. 5.8 The computation of a spur gear's tooth contact shapes

With regard to the selection of sampling points, a somewhat different strategy as compared to the ICS approach is adopted (Section 4.1.4). In the ICS approach, the angular pitch $\tau^1 = 2\pi/z^1$ of the reference gear is considered as the fundamental periodic unit of the gear pair and the sampling interval is selected as $[\theta^1_{SOTC}, \theta^1_{SOTC} + \tau^1] = [\theta^1_{SOTC}, \theta^1_{HPSTC}]$ (see Fig. 5.2). Note that this interval covers only a portion of the full tooth engagement cycle. In the ITCS approach, instead, the full tooth engagement cycle of one tooth pair $[\theta^1_{SOTC}, \theta^1_{EOTC}]$ is selected as the sampling interval. This interval coincides with the shaded area in Fig. 5.9 and its length is equal to the transverse angle of transmission φ^1_a of the reference gear. The selection of the sampling points within the interval $[\theta^1_{SOTC}, \theta^1_{EOTC}]$ proceeds as explained in Section 4.1.4, using either an equidistant sampling grid or a strategy based on a Greedy algorithm.

Note that while a larger angular interval needs to be sampled in the ITCS approach ($\varphi^1_a > \tau^1$), the computation of tooth contact shapes is substantially more efficient than the computation of contact shapes, owing to the near-quadratic dependence of the CPU time on the number of teeth pairs that are simultaneously in contact. Due to the rotational periodicity of a gear, it suffices to compute the tooth contact shapes for one tooth pair only: the tooth contact shapes corresponding to the other teeth can be obtained through rotation of the former whenever necessary.

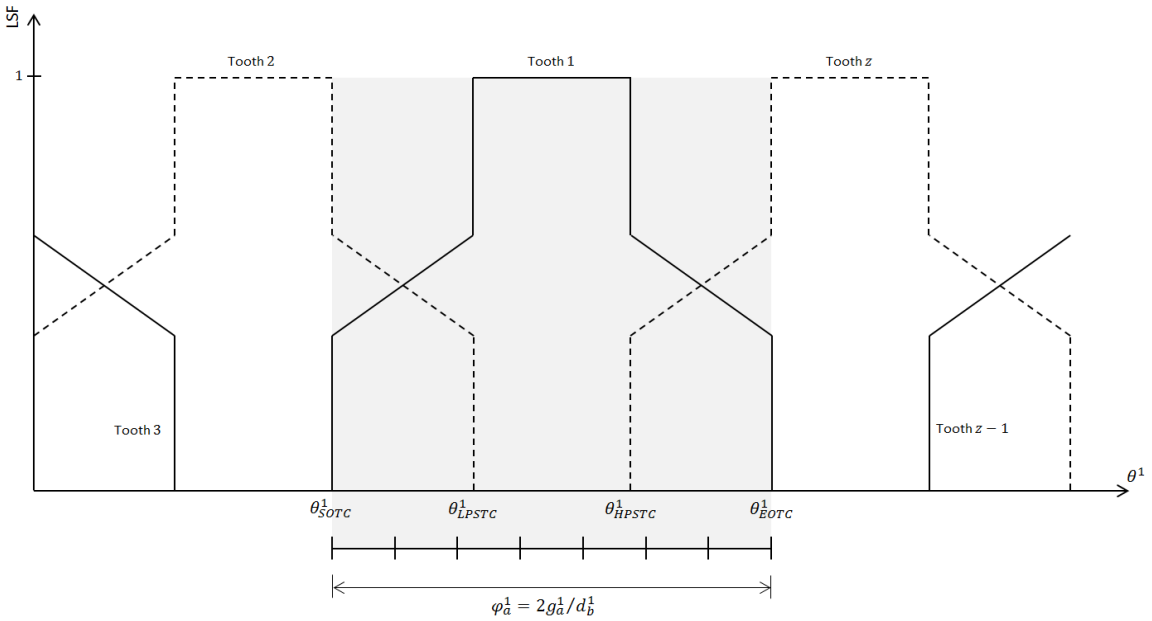


Fig. 5.9 Load Sharing Factor (LSF) plot of a spur gear pair having a contact ratio between 1 and 2

This section on the definition of tooth contact shapes is concluded by pointing out another desirable feature of these shapes, namely that, as compared to regular contact shapes, their deformation patterns are less dependent on the applied torque. This is demonstrated in Fig. 5.10, where two tooth contact shapes are compared: a tooth contact shape computed at 100% torque multiplied by 1.5, and a tooth contact shape computed at 150% torque. As can be seen in this figure, the maximal relative error between these shapes is 0.7%. Because of this mild torque dependence, it suffices to use an approximate value for the LSF of the teeth when computing the tooth contact shapes. Moreover, the ITCS approach will be less susceptible to torque oscillations than the ICS approach when only a single reference torque is used to build the ROM (see Section 4.1.5).

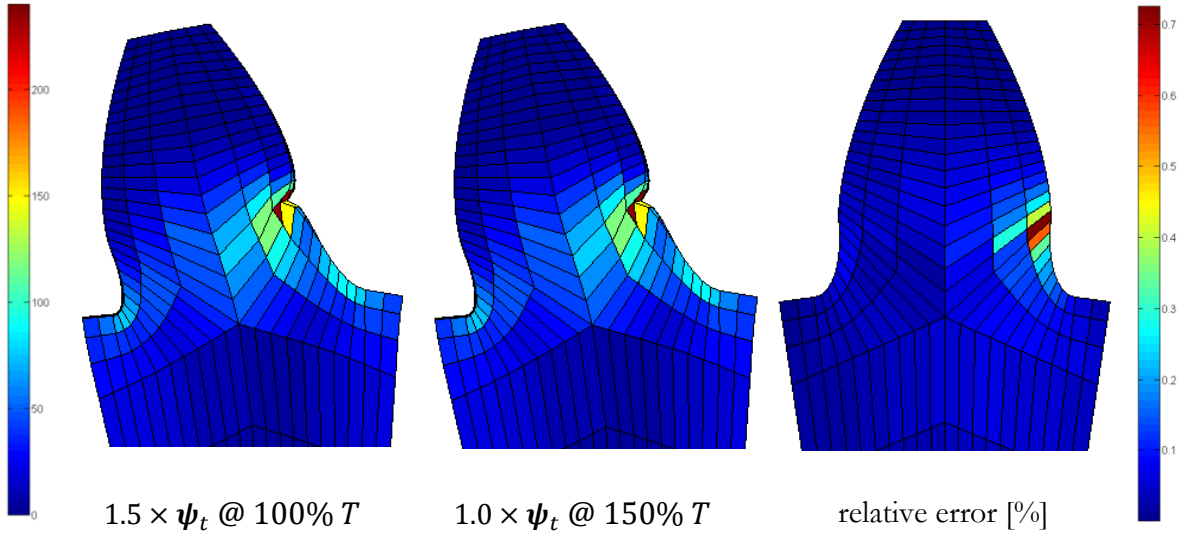


Fig. 5.10 The mild torque dependence of tooth contact shapes

5.2.2 Interpolation of the tooth contact shapes

Similar to the ICS approach, tooth contact shapes are interpolated based on the angular configuration of the gear pair. However, instead of having a single contact shape per gear, the ITCS approach assigns an interpolated tooth contact shape to each tooth that enters the contact zone. In order to keep the number of generalized elastic DOFs constant, the response of a gear pair having a contact ratio of ε will always be described in terms of $\lceil\varepsilon\rceil$ interpolated contact shapes per gear. Each of these shapes has its own dependent parameter p_l, where the subscript $l = l_1, \dots, l_{\lceil\varepsilon\rceil}$ is the active tooth number. All the parameters p_l, including the time-varying active tooth numbers, ultimately depend on the single independent parameter θ^1, i.e. the angular configuration of the reference gear. Denoting the interpolated tooth contact shape associated with the *l*-th tooth by $\boldsymbol{\psi}_{tl}^i$, the basis matrix of the ITCS approach is given by

$$\boldsymbol{V}_{ITCS}^i = [\boldsymbol{\Phi}^i \quad \boldsymbol{\Psi}_t^i(\theta^1)] = \left[\boldsymbol{\Phi}^i \quad \boldsymbol{\psi}_{tl_1}^i(\theta^1) \quad \dots \quad \boldsymbol{\psi}_{tl_{\lceil\varepsilon\rceil}}^i(\theta^1)\right] \in \mathbb{R}^{N\times(n_k+\lceil\varepsilon\rceil)} \tag{5.2.2}$$

where $\boldsymbol{\Psi}_t^i(\theta^1)$ represents the matrix collecting all active interpolated tooth contact shapes. Using a linear interpolation scheme, as in Eq. 4.1.9, the interpolated tooth contact shape $\boldsymbol{\psi}_{tl}^i$ is obtained as

$$\boldsymbol{\psi}_{tl}^i(\theta^i) = \left[1 - p_l^i(\theta^1)\right]\boldsymbol{\psi}_{tl,j}^i + p_l^i(\theta^1)\boldsymbol{\psi}_{tl,j+1}^i \tag{5.2.3}$$

where the subscript *j* refers to the *j*-th angular configuration or sampling point within the interval $[\theta_{SOTC}^1, \theta_{EOTC}^1]$. Three kinds of parameters need to be determined in the above equations:

- The active tooth numbers $l_1, \dots, l_{\lceil\varepsilon\rceil}$ for both the driving and driven gear;
- The tooth contact shape numbers $j_1, \dots, j_{\lceil\varepsilon\rceil}$;
- The interpolation parameters $p_1, \dots, p_{\lceil\varepsilon\rceil}$.

Each of these parameters depends on the angular configuration θ^1 of the reference gear. Note that while the active tooth numbers differ for the driving and driven gear, the shape numbers and interpolation parameters are the same for both gears. The determination of these parameters is less straight-forward than in the ICS approach, and is discussed here for spur gear pairs having a contact ratio between 1 and 2. A similar reasoning is followed when extending this approach to spur gear pairs having higher contact ratios or helical gear pairs.

Consider again the LSF plot of the driving gear of a spur gear pair, as shown in Fig. 5.11. The origin of the abscissa is arbitrarily set at $\theta_{SOTC}^1 - (\theta_{HPSTC}^1 - \theta_{LPSTC}^1)/2$, which is considered to be the starting point of the analysis. This point marks the beginning of the interval in which Tooth 1 and Tooth 2 are in the (numerical) contact zone, as is indicated by the thick arrows underneath

the LSF plot. Whenever the gear rotates over one angular pitch, one tooth leaves the contact zone while another tooth enters. Each tooth that enters the contact zone will remain there during two full angular pitches before being replaced by another tooth. For the spur gear pair of Fig. 5.11, the active tooth numbers l_1^i and l_2^i are given by the following sequence of equations:

$$l_1^i = 1 \mp 2 \cdot rem\left(\left\lfloor \frac{\theta^i}{2\tau^i} \right\rfloor, z^i\right); \qquad l_1^i = rem(l_1^i, z^i);\ if\ \ l_1^i \le 0\ \ then\ \ l_1^i \leftarrow l_1^i + z^i; \quad (5.2.4a)$$

$$l_2^i = 2 \mp 2 \cdot rem\left(\left\lfloor \frac{\theta^i}{2\tau^i} + \frac{1}{2} \right\rfloor, z^i\right);\ \ l_2^i = rem(l_2^i, z^i);\ if\ \ l_2^i \le 0\ \ then\ \ l_2^i \leftarrow l_2^i + z^i \quad (5.2.4b)$$

where the '$-$' holds when $i = 1$ and the '$+$' is used when $i = 2$, and where $\mathrm{rem}(a, b)$ yields the remainder of a/b. Table 5. gives the sequence of active tooth numbers obtained using these expressions for the spur gear pair of Fig. 5.11. Note that by placing the (numerical) tooth transition point in the middle of the Single Tooth Contact (STC) interval, it is assured that the interpolated contact shape corresponding to the tooth number leaving the contact shape is no longer excited at the transition point.

Gear rotation θ^1		Driving gear		Driven gear	
from	until	l_1^1	l_2^1	l_1^2	l_2^2
0	τ^1	1	2	1	z^2
τ^1	$2\tau^1$	1	z^1	1	2
$2\tau^1$	$3\tau^1$	$\mathrm{z}^1 - 1$	z^1	3	2
$3\tau^1$	$4\tau^1$	$\mathrm{z}^1 - 1$	$\mathrm{z}^1 - 2$	3	4
$4\tau^1$	$5\tau^1$	$\mathrm{z}^1 - 3$	$\mathrm{z}^1 - 2$	5	4
$5\tau^1$	$6\tau^1$	$\mathrm{z}^1 - 3$	$\mathrm{z}^1 - 4$	5	6
...	...	...	...	...	...

Table 5.1 Evolution of tooth contact pairs as a function of θ^1

Having obtained the active tooth numbers l_1 and l_2 for both gears, the corresponding tooth contact shape numbers j_l for $l = 1{,}2$ are determined. Considering an equidistant sampling grid of n_c samples, the shape numbers are given by

$$j_l = \begin{cases} 1 & rem(\theta^1 + \delta_{l2}\tau^1, 2\tau^1) < \theta^1_{SOTC} \\ \left\lfloor \left(\dfrac{rem(\theta^1 + \delta_{l2}\tau^1, 2\tau^1) - \theta^1_{SOTC}}{\theta^1_{EOTC} - \theta^1_{SOTC}} \right) n_c \right\rfloor & \theta^1_{SOTC} \le rem(\theta^1 + \delta_{l2}\tau^1, 2\tau^1) \le \theta^1_{EOTC} \\ n_c & rem(\theta^1 + \delta_{l2}\tau^1, 2\tau^1) > \theta^1_{EOTC} \end{cases} \quad (5.2.5)$$

where the Kronecker delta function $\delta_{l2} = 1$ if $l = 2$ and zero otherwise. Finally, the interpolation parameters p_l for $l = 1,2$ corresponding to these shape numbers are given by

$$p_l = \begin{cases} 0 & rem(\theta^1 + \delta_{l2}\tau^1, 2\tau^1) < \theta^1_{SOTC} \\ \left(\dfrac{rem(\theta^1 + \delta_{l2}\tau^1, 2\tau^1) - \theta^1_j}{\theta^1_{j+1} - \theta^1_j} \right) & \theta^1_{SOTC} \le rem(\theta^1 + \delta_{l2}\tau^1, 2\tau^1) \le \theta^1_{EOTC} \\ 1 & rem(\theta^1 + \delta_{l2}\tau^1, 2\tau^1) > \theta^1_{EOTC} \end{cases} \quad (5.2.6)$$

where the angles θ^1_j and θ^1_{j+1} refer to the j-th and $(j+1)$-th angular configurations at which the interpolated tooth contact shapes $\boldsymbol{\psi}^i_{tl,j}$ and $\boldsymbol{\psi}^i_{tl,j+1}$ were computed. As an illustration of the above expressions, Fig. 5.12 shows the evolution of the shape numbers (Eq. 5.2.5) and interpolation parameters (Eq. 5.2.6) as a function of the LSF plot of the spur gear of Table 3.1 with $z^1 = 21$ and $n_c = 20$.

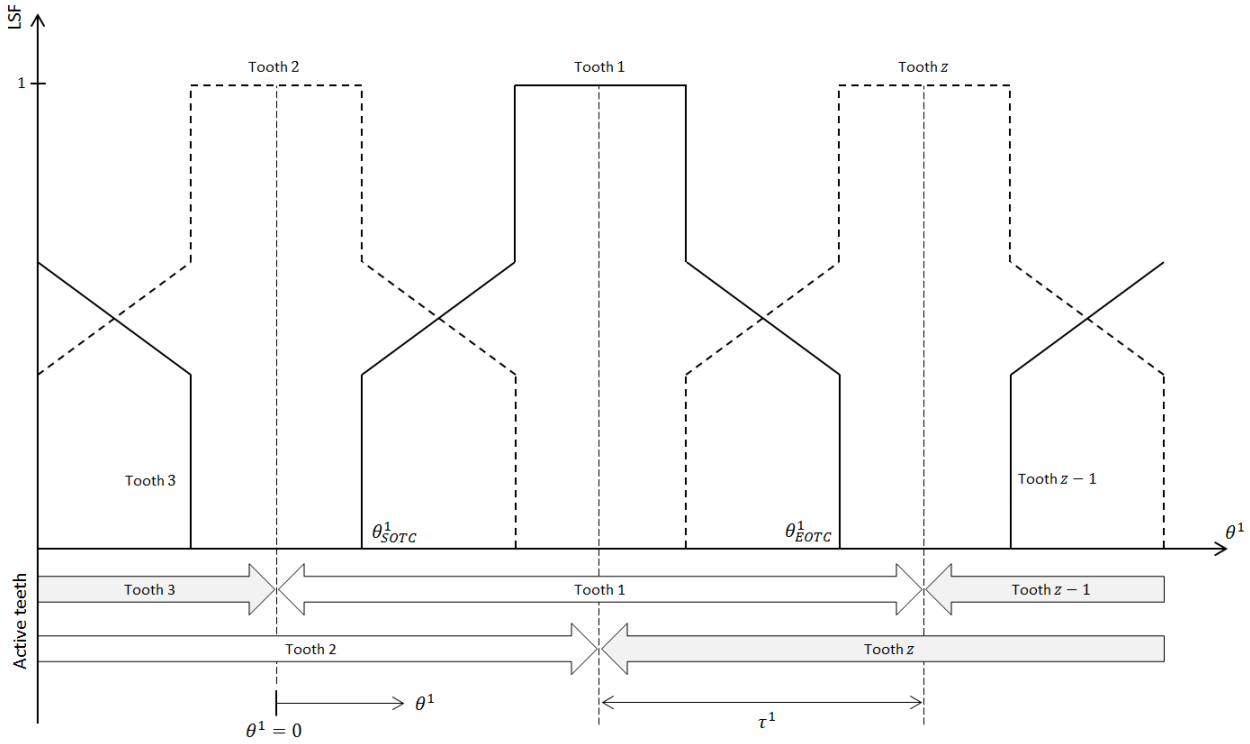


Fig. 5.11 Load Sharing Factor (LSF) plot of a spur gear pair having a contact ratio between 1 and 2. The active teeth are indicated below the LSF plot.

In summary, the ITCS approach is based on a basis matrix $\boldsymbol{V}_{ITCS}^{i}$ that consists of a fixed set of eigenvectors augmented with a fixed number of configuration-dependent tooth contact shapes. A generalized elastic DOF is assigned to each of these shapes, with the meaning of these DOFs changing discontinuously as teeth enter and leave the contact zone and tooth contact shapes are replaced by other tooth contact shapes. Given the linear dependence of the basis matrix $\boldsymbol{V}_{ITCS}^{i}$ on the angular configuration of the reference gear θ^1, the EOMs of the ITCS approach are the same as for the ICS approach (see Section 4.1.3).

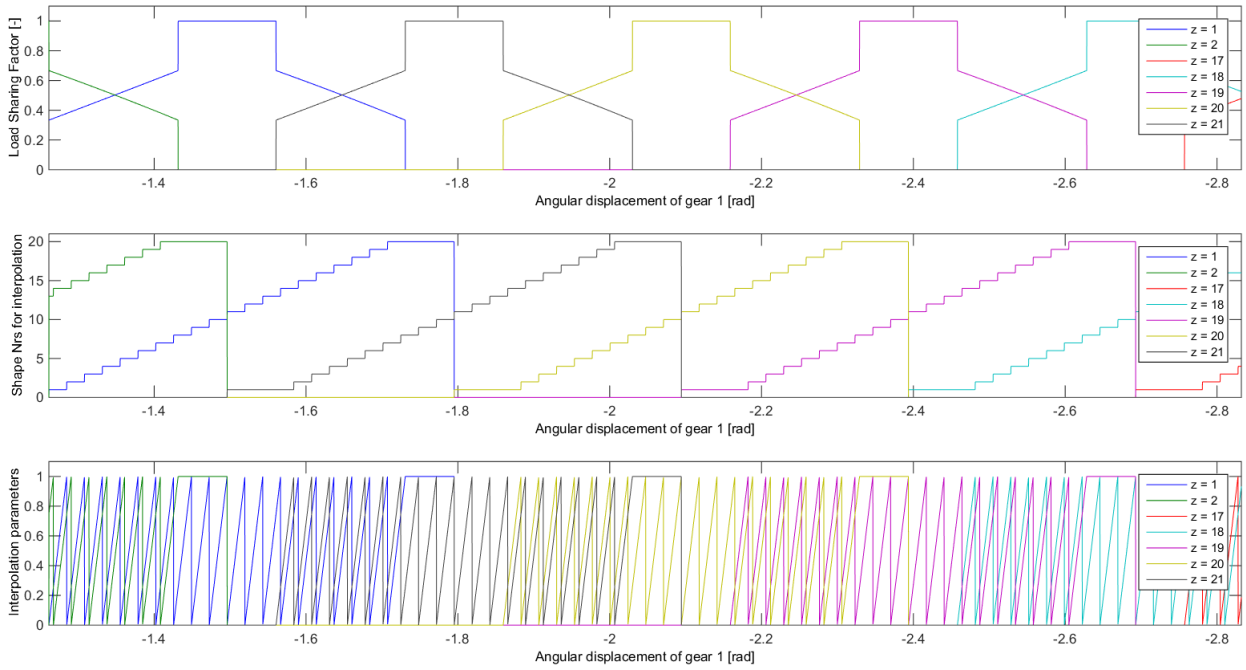


Fig. 5.12 Evolution of shape numbers and interpolation parameters for a spur gear pair having 21 teeth in function of the LSF plot

5.3 Global interpolation and local sliding of the tooth contact shapes

5.3.1 Separation of global and local tooth deformation

In Section 5.1.2 it was observed that the linear interpolation of local tooth deformations leads to unphysical deformations if the sampling grid that is used to compute the contact shapes is too coarse. In Fig. 5.13, the unphysical interpolation of local deformations that was schematically represented in Fig. 5.4 is replaced by what can intuitively be considered as the ideal interpolation scheme. It consists of two steps: first the local deformations $\boldsymbol{\psi}_a$ and $\boldsymbol{\psi}_b$ are displaced to the actual point of contact $\boldsymbol{x}_c$ yielding the shifted deformation shapes $\overrightarrow{\boldsymbol{\psi}}_a$ and $\overleftarrow{\boldsymbol{\psi}}_b$; then a weighted sum of these displaced local deformations yields the ideal interpolated local deformation.

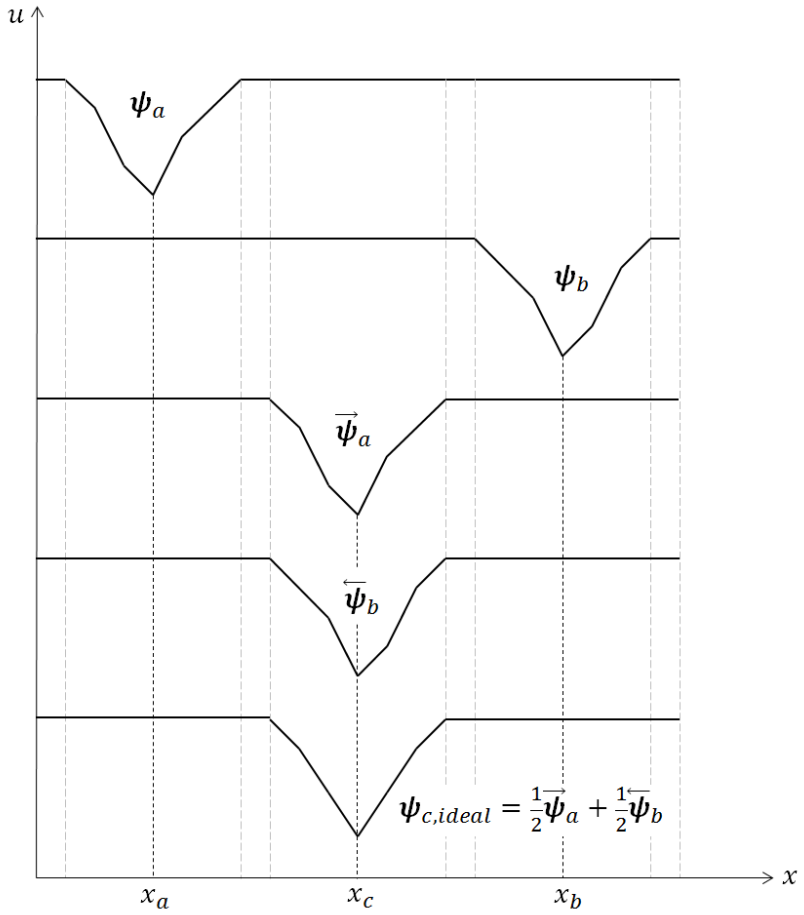


Fig. 5.13 Schematic representation of the ideal interpolation of local deformations

With regard to the interpolation of gear contact shapes, it was concluded in Section 5.1.2 that the global deformation of a gear is well-captured by linearly interpolating the contact shapes, whereas the local deformation is not. If a local sliding approach as the one depicted in Fig. 5.13 is to be applied to the gear contact problem, the global and local contributions to the (tooth) contact shapes should be separated prior to the interpolation. This is achieved using the following four-step approach:

1. Compute the (tooth) contact shapes and store the contact forces $\boldsymbol{f}_c^i$ at equilibrium (Fig. 5.14a);

2. Construct a tooth finite element model by clamping all nodes but the ones in the active half-width of the selected tooth (Fig. 5.14b);

3. Compute the local deformation by applying the stored contact forces $\boldsymbol{f}_c^i$ to the tooth half-width model (Fig. 5.14c);

4. Compute the global deformation by subtracting the local deformation from the global deformation (Fig. 5.14d).

Note that this procedure to separate the local from the global deformations bears resemblance to the approach of Andersson-Vedmar [19] (See Appendix N.2).

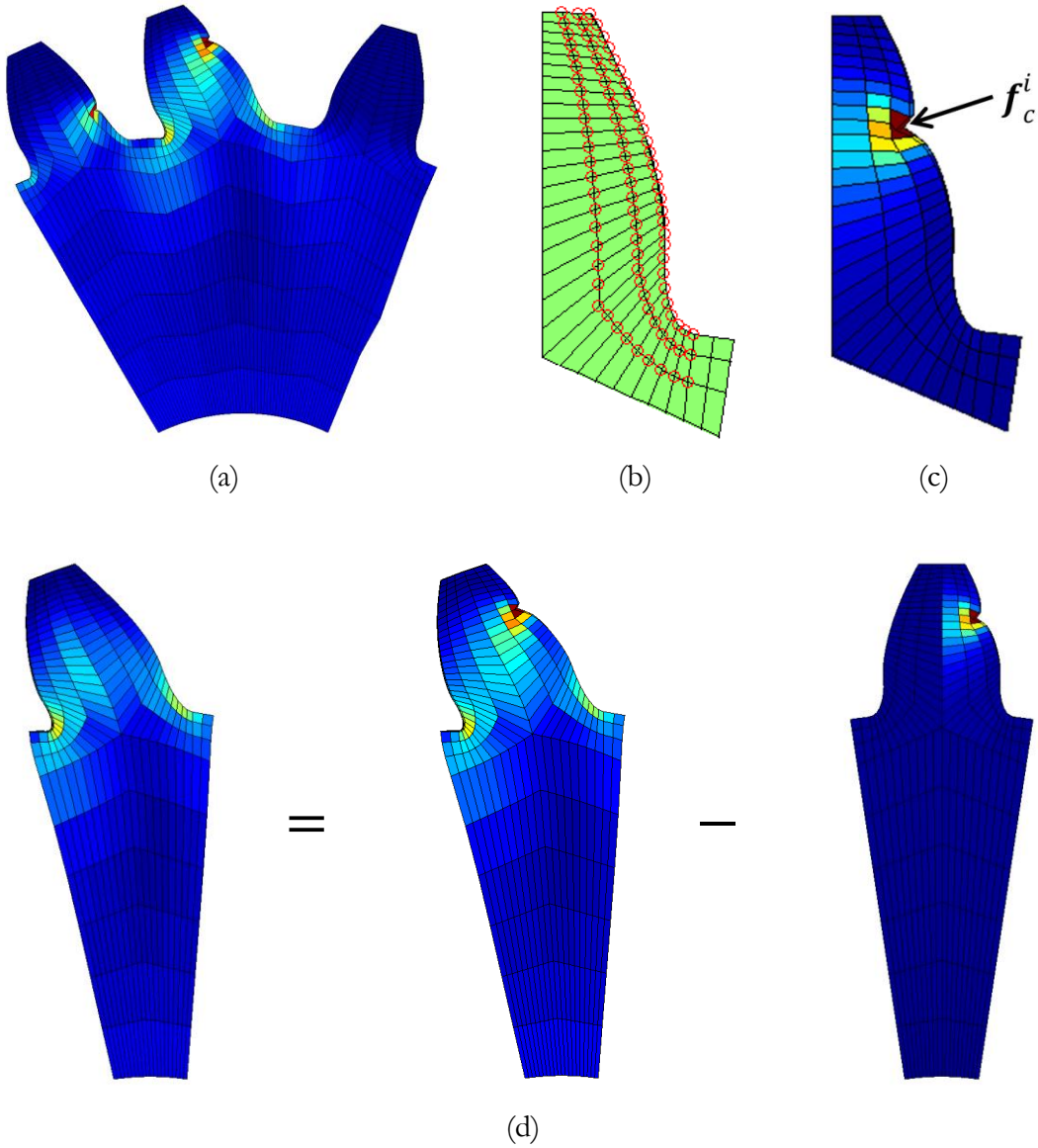


Fig. 5.14 Separation of global and local deformations in a spur gear's (tooth) contact shapes

5.3.2 Sliding the local tooth deformations

Having separated the global and local contributions to the (tooth) contact shapes, an improved interpolation scheme, based on Fig. 5.13, can be developed for the reduced-order gear contact model. Global deformations are interpolated using the linear interpolation scheme of Eq. 5.2.3

without modifications. Local deformations, instead, are first displaced to the actual point of contact before applying linear interpolation. The displacement of local tooth deformation is based on the radii of contact of the (tooth) contact shapes and the actual radius of contact (Eq. M.6 in Appendix M). Obviously, sliding local deformations in a discontinuous FE setting is not as straightforward as in a continuous setting. In order to demonstrate the general sliding strategy, the simplified example of Fig. 5.15 is considered. In this figure, a one-dimensional FE model is considered with nodes on $x = 1,2,\dots,15$. A contact shape $\boldsymbol{\psi}_a$ is computed for a contact force acting at $x = 4.5$, and at the considered instant the actual contact force is applied at $x = 10.25$. By shifting the contact shape 5 nodes to the right, the deformation pattern $\vec{\boldsymbol{\psi}}_a^5$ corresponding to a contact load at $x = 9.5$ is obtained, whereas a shift of 6 nodes to the right yields the deformation pattern $\vec{\boldsymbol{\psi}}_a^6$ corresponding to a contact load at $x = 10.5$. The correct slided contact shape $\vec{\boldsymbol{\psi}}_a$ is obtained by computing the weighted sum of $\vec{\boldsymbol{\psi}}_a^5$ and $\vec{\boldsymbol{\psi}}_a^6$ based on the respective contact force locations, i.e.

$$\vec{\boldsymbol{\psi}}_a = \left(\frac{10.5 - 10.25}{10.5 - 9.5}\right)\vec{\boldsymbol{\psi}}_a^5 + \left[1 - \left(\frac{10.5 - 10.25}{10.5 - 9.5}\right)\right]\vec{\boldsymbol{\psi}}_a^6 = \frac{1}{4}\vec{\boldsymbol{\psi}}_a^5 + \frac{3}{4}\vec{\boldsymbol{\psi}}_a^6 \tag{5.3.1}$$

The slided contact shape $\vec{\boldsymbol{\psi}}_a$ is considered to be the best approximation of sliding the original contact shape $\boldsymbol{\psi}_a$ over a distance of 5.75 to the right in a discontinuous setting.

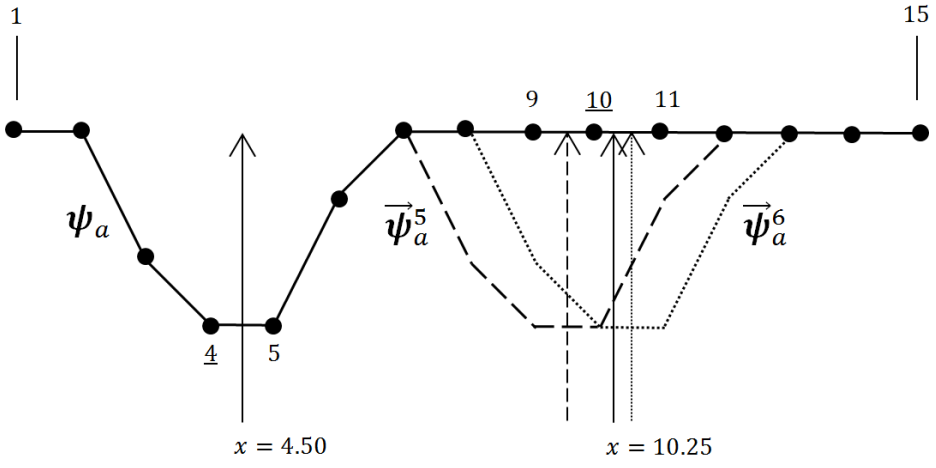


Fig. 5.15 Simplified example of sliding local deformations in a finite element setting

Returning to the three-dimensional gear contact problem, the approach for sliding local tooth deformations is similar as the procedure outlined above. When shifting the local displacements, the full grid of tooth flank displacements is shifted upwards or downwards with respect to the root of the tooth, setting empty entries in the shifted grid equal to zero and discarding entries that fall out of the grid. In Fig. 5.16 the concept of sliding local tooth displacements is illustrated for a coarse sampling grid where the direct linear interpolation of local (tooth) contact shapes yields to

unphysical results (Fig. 5.16a). The effect of sliding local tooth deformations is graphically illustrated in Fig. 5.16b, and the improved accuracy of the novel interpolation scheme is demonstrated in Fig. 5.16c-d.

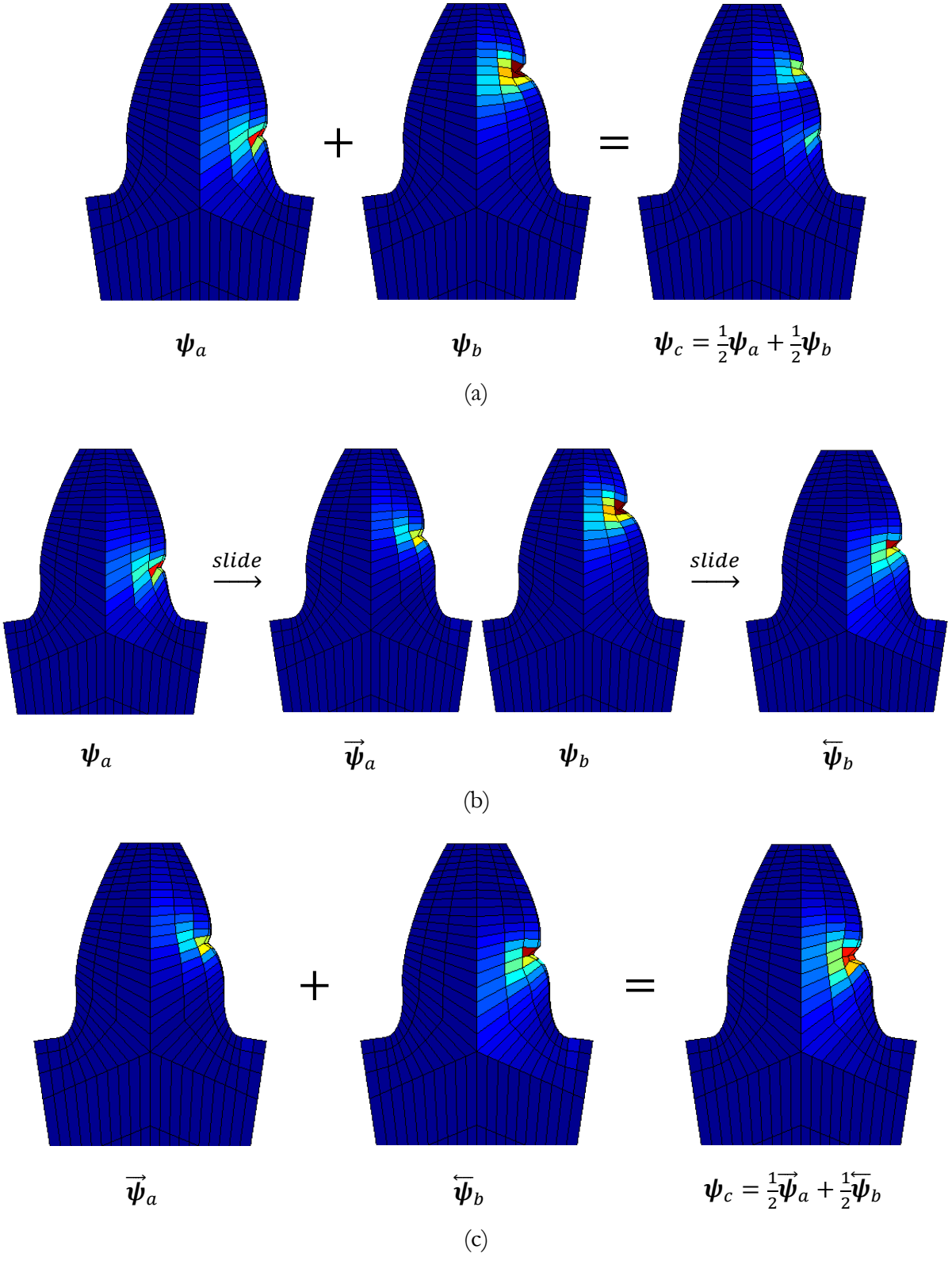
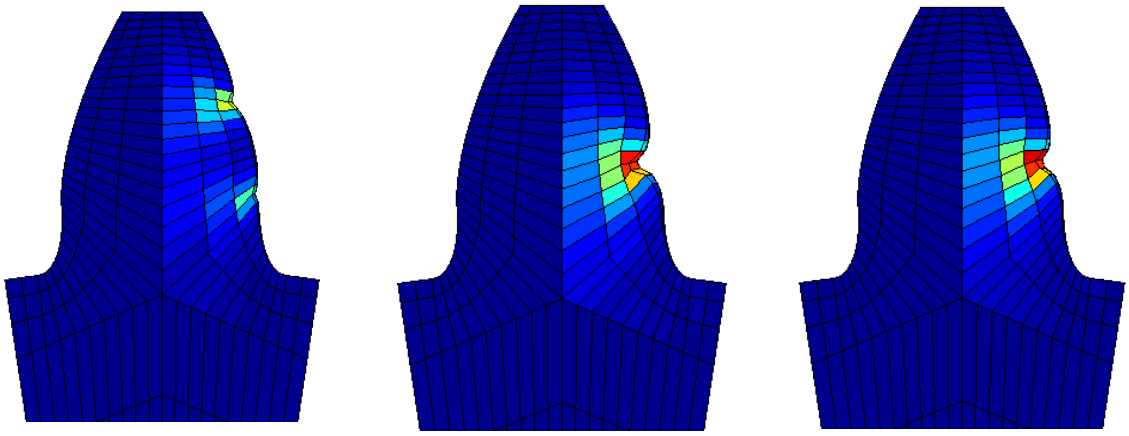


Fig. 5.16 Sliding of local tooth deformations in a spur gear



Direct linear interpolation Local sliding + interpolation Exact contact shape

(d)

Fig. 5.16 (cont.) Sliding of local tooth deformations in a spur gear

By sliding the local tooth deformations, the interpolated (tooth) contact shapes closely approximates the theoretical (tooth) contact shapes, as is illustrated in Fig. 5.16. However, the EOMs of a reduced-order gear pair were derived in Chapter 4 assuming the interpolated contact shapes are obtained by direct application of the linear interpolation formula, Eq. 4.1.9. There it was observed that the linear interpolation of contact shapes results in a quadratic interpolation of certain mass and stiffness invariants (Eq. 4.1.21 and Eq. 4.1.39). Given the importance of the quasi-static elastic restoring forces in flexible contact problems, it is crucial for the accuracy of the simulation that the interpolated reduced-order stiffness matrix $\boldsymbol{K}_{tt}^{i} = \boldsymbol{\Psi}_{t}^{i\,T}\boldsymbol{K}_{FE}^{i}\boldsymbol{\Psi}_{t}^{i}$ is well represented by the interpolation scheme. Consider the first entry of this matrix, corresponding to the interpolated tooth contact shape $\boldsymbol{\psi}_{t1}^{i}$. The direct linear interpolation of this matrix leads to the following expression for $\boldsymbol{K}_{tt}^{i}\{1,1\}$:

$$\begin{aligned}\boldsymbol{\psi}_{t1}^{i\,T}\boldsymbol{K}_{FE}^{i}\boldsymbol{\psi}_{t1}^{i} = {} & (1-p)^2\boldsymbol{\psi}_{t1,j}^{i\,T}\boldsymbol{K}_{FE}^{i}\boldsymbol{\psi}_{t1,j}^{i} \\ & + p(1-p)\left(\boldsymbol{\psi}_{t1,j}^{i\,T}\boldsymbol{K}_{FE}^{i}\boldsymbol{\psi}_{t1,j+1}^{i} + \boldsymbol{\psi}_{t1,j+1}^{i\,T}\boldsymbol{K}_{FE}^{i}\boldsymbol{\psi}_{t1,j}^{i}\right) \\ & + p^2\boldsymbol{\psi}_{t1,j+1}^{i\,T}\boldsymbol{K}_{FE}^{i}\boldsymbol{\psi}_{t1,j+1}^{i}\end{aligned} \tag{5.3.2}$$

The above expression is plotted in Fig. 5.17 as a function of θ^1 (going from θ_{SOTC}^1 until θ_{EOTC}^1) for two ROMs that are constructed using equidistant sampling grids of $n_s = 5, 10, 20$ and 40 sampling points, respectively. It can be observed in this figure that the closer the interpolated (tooth) contact shapes resemble the theoretical (tooth) contact shapes (i.e. the more sampling points that are used), the closer the values of $\boldsymbol{K}_{tt}^{i}\{1,1\}$ are to a smooth curve running through all the sampling points. As sliding local tooth deformations yields (tooth) contact shapes that approximate the theoretical (tooth) contact shapes very closely, the value of $\boldsymbol{K}_{tt}^{i}\{1,1\}$ should evolve according to this smooth curve, and not as prescribed by the quadratic interpolation scheme of Eq. 5.3.2, as was confirmed

by numerical experiments conducted by the author. Similar observations hold for all interpolated mass and stiffness invariants involving the interpolated (tooth) contact shapes. As no analytical equations are available for these smooth curves, they need to be determined by curve-fitting prior to the actual simulation.

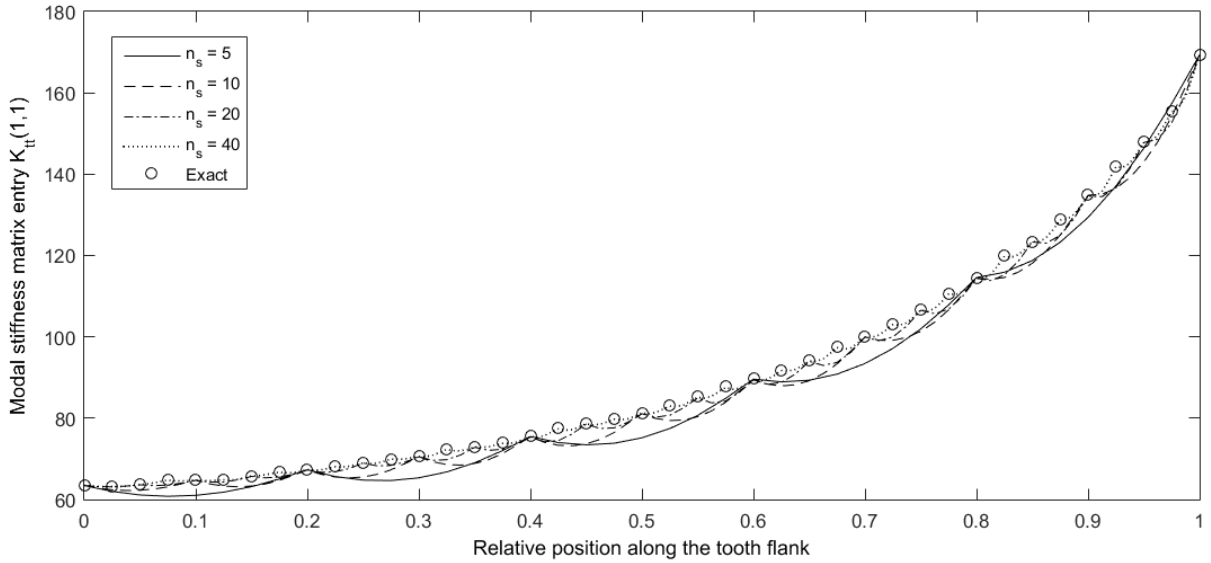


Fig. 5.17 Evolution of the first element of the reduced-order stiffness matrix $\boldsymbol{K}_{tt}^{i}$

5.4 Numerical validation

In this section, the ITCS approach with local sliding is compared with the ICS approach with direct linear interpolation, as introduced in Chapter 4. To this end, the spur gear simulations of Section 4.4.1 are repeated using the simulation parameters outlined in Table 4.3, with the exception that both the ICS and ITCS models are constructed using a coarse equidistant angular sampling grid with $n_s = 10$. The main simulation parameters are summarized in Table 5.2. Note that the stable time increment is taken from the CMS-based reference model, and is thus not the stable time increment for the I(T)CS models. With regard to computational efficiency, the ITCS approach is somewhat less performant than the ICS approach (respectively 71 and 55 times faster than the CMS-based reference model), which is due to the comparatively high cost of interpolating the tooth contact shapes (see Section 5.3), and the small increase in number of DOFs in the ITCS model.

The Dynamic Transmission Error (DTE, see Section 3.3) of the spur gear pair is shown in Fig. 5.18. The DTE predicted by the ICS approach exhibits relatively large errors whereas for the ITCS approach the relative error is below 5% throughout the entire simulation interval. The superior accuracy of the ITCS approach really becomes apparent when inspecting the normal stresses in the contact zone (see left graph in Fig. 5.19): for a point located midway between the points of contact corresponding to adjacent contact shapes, the local contact stresses are

significantly underestimated by the ICS model. In the example considered in Fig. 5.19, the peak stresses are underestimated by roughly 200 MPa or approximately 30%. The ITCS approach instead is capable of predicting local stresses across the entire gear flank surface with high accuracy. In accordance with Saint Venant's principle, the normal stresses in the root are well captured by both the ICS and ITCS approach (see right graph in Fig. 5.19). Similar conclusions can be drawn from Fig. 5.20, where the Von Mises stress of the spur gear pair are plotted at a particular instant during the simulation (t = 1.8 ms) where the reference angle of the gear pair is midway between two angular sampling points. Note that in the ICS model, owing to the linear interpolation of contact shapes on a coarse sampling grid, the zone of local deformation is spread out over a larger area than in the CMS and ITCS approaches, resulting in lower maximal stresses.

Parameters	**ICS**	**ITCS**
Gear pair	Spur	
Number of angular samples	10	10
Number of torques	1	1
Number of eigenvectors	50	50
Number of DOFs in gear pair model	102	104
Stable time increment	5e-9 s	
Applied torque	250 Nm	
Steady-state angular velocity pinion	100 rad/s	
Penalty factor	5e8 N/m	
Single core CPU time CMS	354224 s	
Single core CPU times	4954 s	6420 s
Speed-up factor w.r.t. CMS	71	55

Table 5.2 Simulation parameters of the dynamic simulations

Summarizing, the ITCS approach yields displacement and stress results with similar accuracy as the standard CMS method, and with superior accuracy as compared to the ICS approach in the case of coarse sampling grids. The price to pay is a small increase in the number of DOFs (proportional to the contact ratio of the gear pair) and added algorithmic complexity in the interpolation of the contact shapes, which results in an increase in overall computational cost of 10-20%.

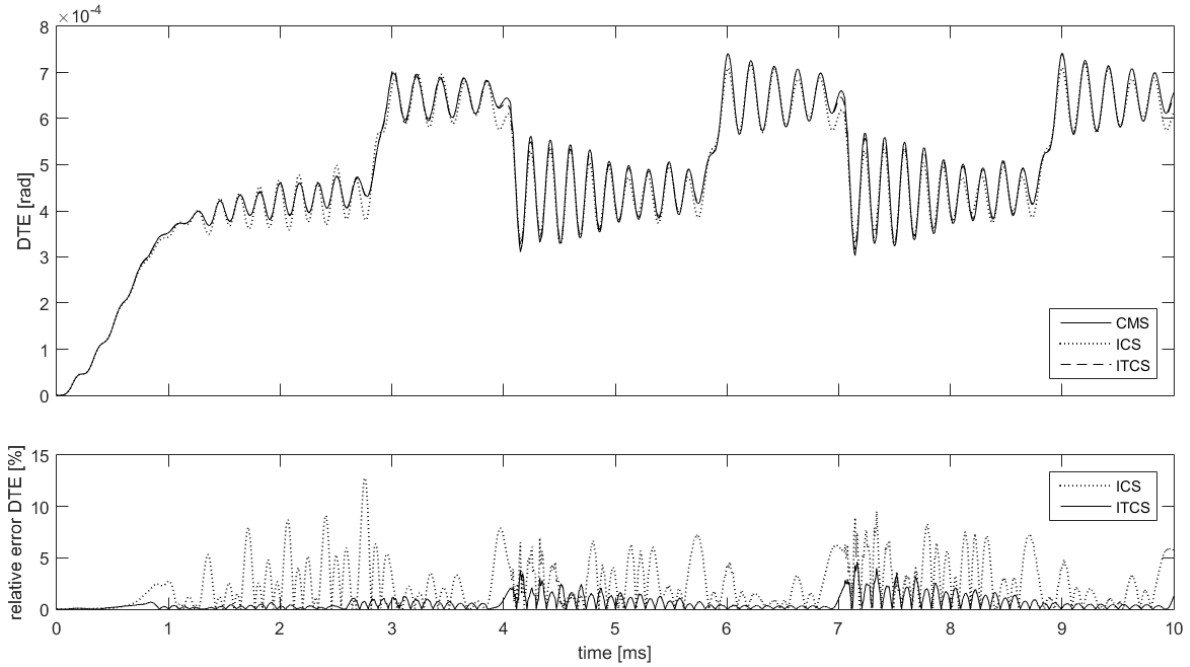


Fig. 5.18 Comparison of the DTE predictions of a spur gear pair by the CMS, ICS, and ITCS models

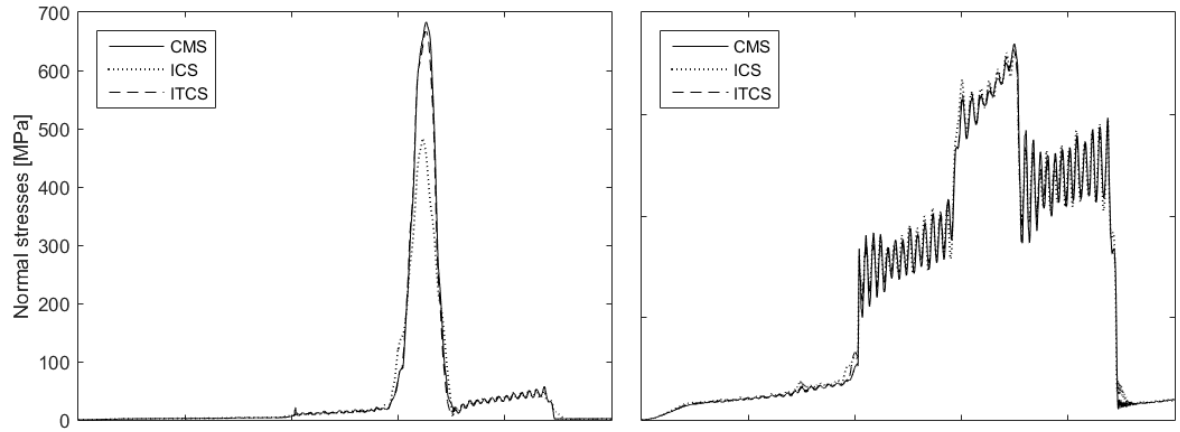
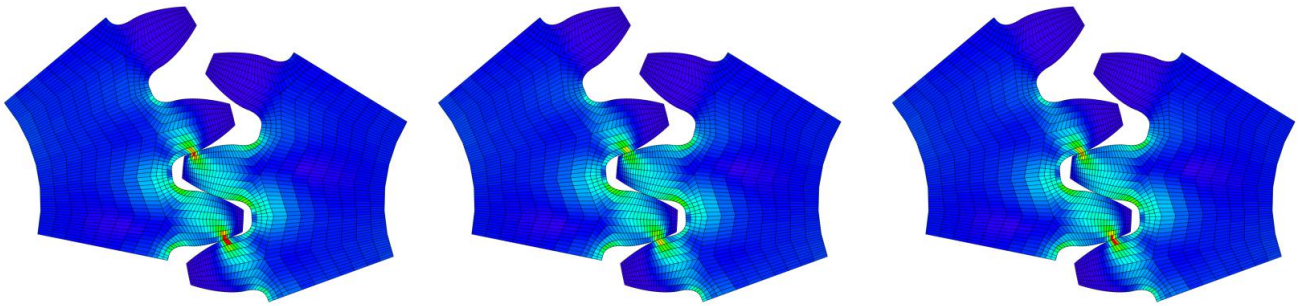


Fig. 5.19 Comparison of the contact stress predictions of a spur gear pair by the three ROMs



(a) CMS (b) ICS (c) ITCS

Fig. 5.20 Comparison of the Von Mises stress predictions of a spur gear pair at t = 1.8 ms by the CMS, ICS, and ITCS ROMs. The color scaling is the same as in Fig. 4.14.

5.5 Summary and conclusion

In this chapter, the main shortcomings of the Interpolated Contact Shapes (ICS) approach presented in Chapter 4 were investigated both conceptually and numerically. In the ICS approach, the quasi-static elastic response of a pair of interacting flexible bodies is described by interpolating among a set of precomputed contact shapes based on the relative configuration of the bodies. When for one reason or another the sampling of the configuration space is coarse, the polynomial interpolation of adjacent contact shapes might lead to interpolated shapes that are physically meaningless, especially when the trajectory of the interacting bodies is discontinuous (e.g. in the case of a pair of spur gears, where the number of teeth in contact varies discontinuously). Moreover, in the ICS approach the influence of spatially separated contact zones is combined into a single shape vector, which can be too restrictive a modelling approach when dynamic effects prevail. For these reasons, the current chapter presents a modified version of the ICS approach that is based on the interpolation of so-called *tooth contact shapes*. Instead of interpolating linearly among these shapes, the local contribution of the shapes is separated from the global contribution, interpolating the latter using a linear scheme and *sliding* the local deformations based on the actual point of contact, thus yielding interpolated tooth contact shapes that are physically meaningful no matter how coarse the sampling of the configuration space. Notwithstanding the somewhat higher computational cost of the Interpolated Tooth Contact Shapes (ITCS) approach, the superior accuracy of the method in the case of a coarse angular sampling and the improved representation of torque nonlinearities were demonstrated through a series of numerical experiments.

Chapter 6

Extensions of the novel MOR method

A novel Model-Order Reduction (MOR) method for dynamic gear contact problems was introduced in Chapter 4 assuming the gears are perfectly aligned, rotate about fixed axes in space, and engage with only one other gear. In this chapter, the method is extended to include misaligned gears that can rotate and translate in three-dimensional space as well as engage with multiple gears at the same time. While the main principles of the original method remain valid under these circumstances, a number of topics, including the efficient sampling of high-dimensional parameter spaces and the interpolation of contact shapes in a multi-mesh setting, need to be properly investigated.

This chapter is organized as follows: in Section 6.1 the Interpolated Contact Shapes (ICS) approach (see Chapter 4) is applied to the misaligned gear contact problem. The main challenges in adapting the ICS approach to this problem are the cost-effective sampling of the misalignment parameters and the proper interpolation of the contact shapes. Section 6.2 then discusses the application of the ICS approach to the dynamic analysis of planetary gear sets. Here the main difficulty is in designing an interpolation scheme that accounts for the multitude of simultaneous gear contacts. At the end of each section, the modified ICS approaches are validated by comparison with reference simulations. Finally, Section 6.3 briefly discusses the use of the ICS approach in a quasi-static gear contact force element for use in general flexible multibody simulations.

6.1 Model reduction of misaligned gear contact problems

6.1.1 Preliminary considerations on gear misalignment

When gears misalign, this can result in shifts of contact force distributions, moving peak stresses to the edges of the tooth flanks, and may ultimately lead to increased gear wear and noise. Gear misalignment can be caused by manufacturing and assembly errors or by elastic deflections of the gear pair's supporting structures. A distinction is made between parallel misalignment and angular misalignment. The former results in a change in center distance between the gear shafts, thereby slightly altering the contact ratio of the gear pair while leaving the shape of the contact zone essentially unchanged. Angular misalignment is further differentiated into misalignment *parallel* and

perpendicular to the plane of action, see Fig. 6.1. Parallel angular (or yaw) misalignment shifts the contact loads to the side of the face width by increasing tooth separation at one side and reducing it at the other, yet without changing the contact area and contact ratio. In the case of perpendicular angular (or pitch) misalignment, on the other hand, the contact zone becomes skewed and the contact area and contact ratio of the gear pair are reduced. Both types of angular misalignment will generally result in increased tooth contact and bending stresses.

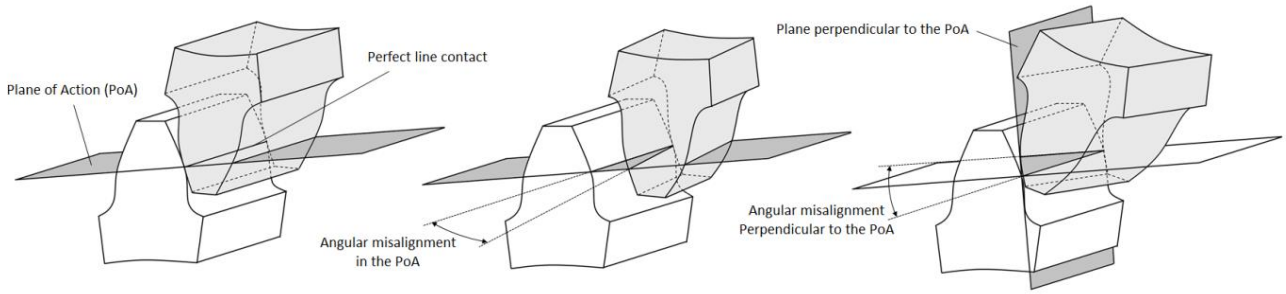


Fig. 6.1 Angular misalignment in and perpendicular to the Plane of Action (PoA)

Without loss of generality of the proposed method, this dissertation only considers gear misalignments that are caused by elastic deflections of the supporting structures. A simple geared transmission is shown in Fig. 6.2 and consists of a pinion and wheel that are supported by flexible shafts. The shafts are supported at their outer ends by rigid or flexible bearings and are initially perfectly aligned. When a torque is applied to the pinion shaft, the reaction forces at the centers of the gears will cause the shafts to deform, leading to gear pair misalignment. For spur gear pairs and symmetric mounting ($z_g/L = 0.5$), the shaft deformations will solely result in parallel misalignment (increase in center distance), whereas for helical gears and off-centered spur gears ($z_g/L \neq 0.5$) parallel as well as angular misalignment will occur.

In the presented analysis of the flexibly supported gear pair, the shafts are modelled using linear hexahedral finite elements. Rigid Multi-Point Constraints (MPCs) are used to constrain the nodes that coincide with the gear and bearing locations. In order to simplify the analysis, the Finite Element (FE) models of the shafts are excluded from the actual simulations. Each gear is instead supported by a so-called bushing element that is defined by a 6-by-6 symmetric stiffness matrix $\boldsymbol{K}_b^i$ (see Section 3.2.2, Eq. 3.2.5). This matrix represents the stiffness of the gear's supporting structure and is computed by applying unit loads and torques to the gear-shaft MPC and inverting the thus-obtained compliance matrix. In dynamic analyses, the inertial properties of the shafts are included in the definition of the mass invariants of the gears. In doing so, the system of Fig. 6.2 is reduced to a spatial two-body contact problem.

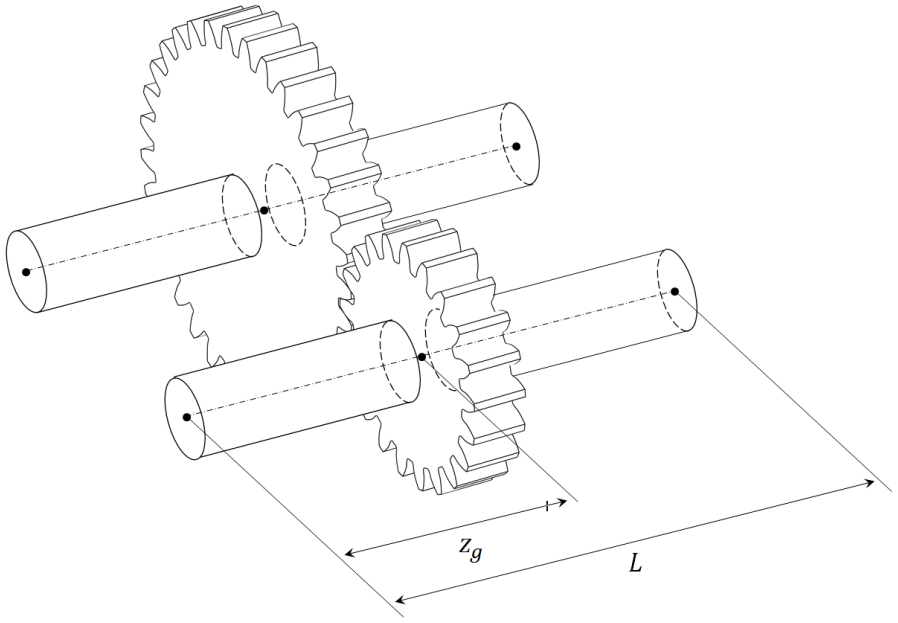


Fig. 6.2 A spur gear pair supported by flexible shafts

6.1.2 General outline of the solution approach

The equations governing the dynamic behavior of misaligned gears are discussed in Section 3.2.2 (Eq. 3.2.4). Contrary to perfectly aligned gears, which are described in terms of a single rigid-body coordinate, θ^i, the configuration of misaligned gears is expressed in terms of the full vector of rigid-body or reference coordinates. Having selected the three-parameter set of Bryant angles $\boldsymbol{\theta}^i = \{\boldsymbol{\phi}^i, \boldsymbol{\psi}^i, \boldsymbol{\theta}^i\}$ to parameterize rotations (see Appendix C), the full vector of reference coordinates $\boldsymbol{q}_r^i$ of the *i*-th gear is given by:

$$\boldsymbol{q}_r^i = \{\boldsymbol{R}^{iT} \quad \boldsymbol{\theta}^{iT}\}^T = \{X_1^i, X_2^i, X_3^i, \boldsymbol{\phi}^i, \boldsymbol{\psi}^i, \boldsymbol{\theta}^i\}^T \tag{6.1.3}$$

where $\boldsymbol{R}^i = \{X_1^i, X_2^i, X_3^i\}^T$ is the vector of Cartesian coordinates. Parallel misalignment occurs whenever the relative displacements in the $\boldsymbol{X}_1\boldsymbol{X}_2$ are non-zero, i.e. $\Delta X_1^i \neq 0$ or $\Delta X_2^i \neq 0$, axial misalignment occurs when $\Delta X_3^i \neq 0$, and angular misalignment occurs when $\Delta\boldsymbol{\phi}^i \neq 0$ or $\Delta\boldsymbol{\psi}^i \neq 0$. As before, the third Bryant angle $\boldsymbol{\theta}^i$ measures the global rotation of the *i*-th gear about its axis of rotation.

In order to efficiently solve the misaligned gear contact problem, the perfectly-aligned ICS approach is extended to the case of misaligned gears. This extension is complicated by the fact that the number of parameters describing the configuration of each gear increases from one to six: three parameters describing translational misalignment ($\Delta X_1^i, \Delta X_2^i, \Delta X_3^i$), two parameters describing angular misalignment ($\Delta\boldsymbol{\phi}^i, \Delta\boldsymbol{\psi}^i$), and a sixth parameter ($\boldsymbol{\theta}^i$) representing the global rotation of the gear pair. In order to accurately capture the displacements and strains resulting from these misalignments, the set of contact shapes $\boldsymbol{\Psi}_c^i$ should be enriched accordingly, i.e. with a sufficiently

rich set of *misaligned* contact shapes. Analogous to the perfectly aligned contact shapes, each misaligned contact shape is obtained by evaluating the Full-Order Model (FOM) of the flexible gear pair for a selected combination of configuration parameters (and external forces). An appropriate choice of the locations of these parameter combinations in the parameter domain is fundamental to accurately capture the behavior of the gear pair in the entire domain.

In Section 2.3.2.3 it was argued that when the number of parameters is large, sampling methods that are problem-aware and adaptive should be selected over physics-based, uniform or random sampling methods. The selection of parameter samples in these methods is guided by the underlying system behavior and proceeds sequentially in an adaptive manner, based on a quantifiable measure of the comprehensiveness of the selected set of parameter samples. One such method, the Greedy sampling method [84], was introduced in Chapter 4 (Section 4.1.4) as an effective means to identify the gear pair configurations that contribute most to the set of contact shapes $\boldsymbol{\Psi}_c^i$ in terms of added information. Starting from a base Reduced-Order Model (ROM), the Greedy method progressively enriches the ROM by determining the parameter values at which the current ROM leads to the largest estimated error, evaluating the FOM for those parameter values and updating the ROM accordingly. Given the favorable properties of the method even when large numbers of parameters need to be sampled, the Greedy approach will also be applied to the misaligned gear contact problem, see Section 6.1.3.

The Greedy approach uses the ROM to identify its weak spots within the parameter domain. Therefore, a definition of the ROM is required, including a definition of the interpolation scheme that is used to construct the ROM. In the perfectly-aligned ICS approach, contact shapes are interpolated based on the angular configuration of the gear pair, i.e.

$$\boldsymbol{\psi}_c^i(\theta^1) = [1 - p(\theta^1)]\boldsymbol{\psi}_{c,j}^i + p(\theta^1)\boldsymbol{\psi}_{c,j+1}^i \tag{6.1.4}$$

where θ^1 is the current angular configuration of the reference gear, p is the interpolation parameter, and $\boldsymbol{\psi}_{c,j}^i$ and $\boldsymbol{\psi}_{c,j+1}^i$ are the contact shapes computed at the adjacent angular configurations θ_j^1 and θ_{j+1}^1. The number of sampled angular configurations in this approach is typically somewhere between 10 and 50. The aim of interpolating the corresponding contact shapes is to capture the majority of the entailed information by a single DOF. A similar approach could be used in the misaligned case, where the misaligned contact shapes are interpolated based on the misalignment parameters to obtain a single interpolated contact shape and keep the number of DOFs minimal. However, most multivariate interpolation schemes becomes mathematically intractable as soon as the number of interpolation parameters reaches beyond two. Therefore, in this dissertation it is opted to interpolate the (misaligned) contact shapes based on the angular configuration θ^1 only, essentially sticking to the interpolation scheme of Eq. 6.1.4 where $\boldsymbol{\psi}_{c,j}^i$ and $\boldsymbol{\psi}_{c,j+1}^i$ are replaced by matrices containing all (misaligned) contact shapes computed at the angular configurations θ_j^1 and θ_{j+1}^1 (see Fig. 6.3). In terms of the global versus local ROM discussion of

Section 2.3.2, the misaligned ICS approach is local in terms of angular configuration but global with regard to misalignments. The price to pay for keeping the interpolation scheme simple is an increase in the number of elastic DOFs from 1 to $(1 + n_m)$ per gear, where n_m is the number of misalignment shapes per angular configuration[25]. However, as will be demonstrated in Section 6.1.3, through proper usage of matrix compression techniques, n_m can be kept reasonably small so as not to jeopardize the computational efficiency of the misaligned ICS approach.

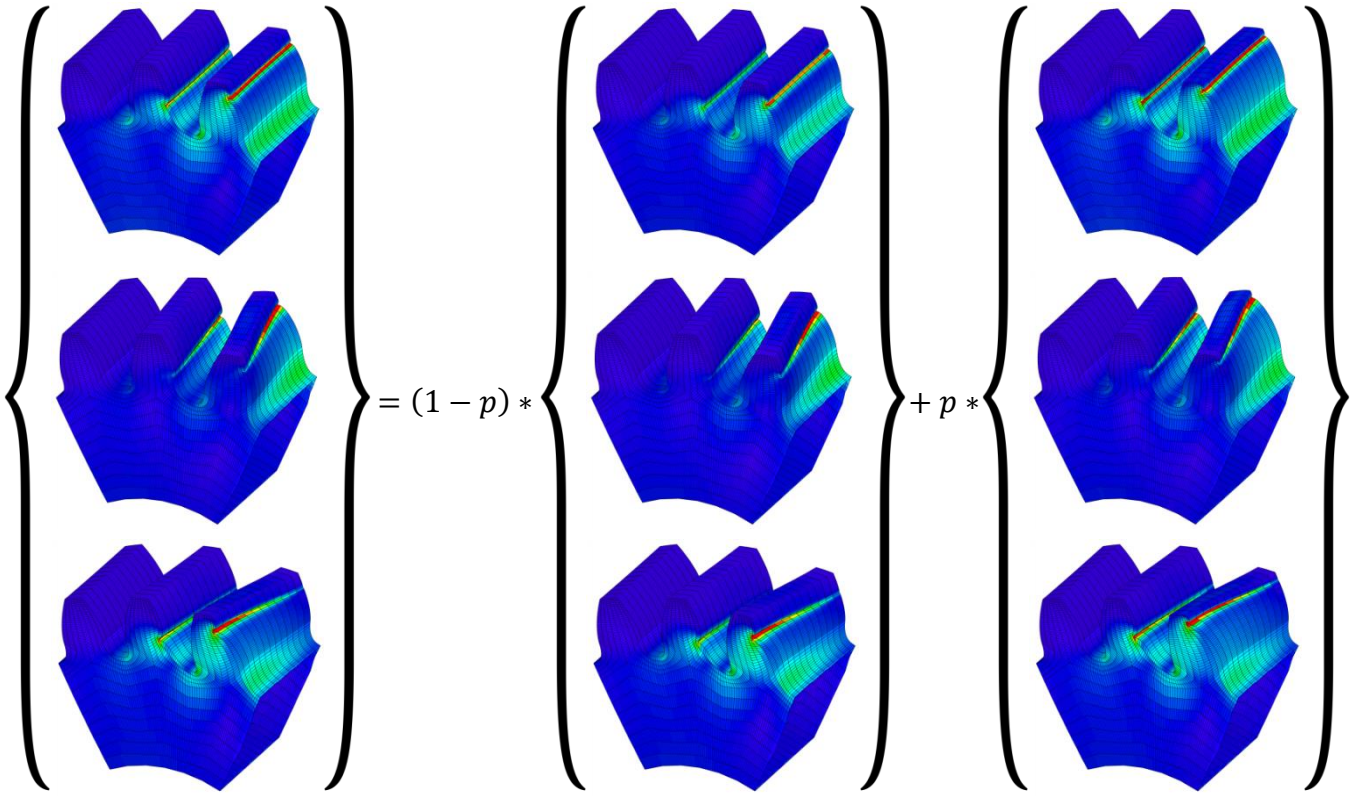


Fig. 6.3 Interpolation of (misaligned) contact shapes in the misaligned ICS approach

Having selected a suitable interpolation scheme and a procedure for training the ROM, the reduced-order Equations Of Motion (EOMs) of a misaligned gear pair can be established. In Section 2.4, the EOMs of spatial flexible multibody systems are derived assuming the basis matrices $\boldsymbol{V}^i$ of the flexible bodies are constant matrices (i.e. $\dot{\boldsymbol{V}}^i = \boldsymbol{0}$). In Chapter 4, this assumption is abandoned for the case of planar rigid body motion, and the influence of the $\dot{\boldsymbol{V}}^i$-proportional terms in the EOMs is investigated (see Section 4.1.3 and Section 4.4.2). A similar investigation for the case of general rigid-body motion in three dimensions is out of the scope of this chapter, but proceeds in similar fashion as outlined in the aforementioned sections. Given the great similarity between the interpolation schemes used in the current chapter and in Chapter 4 (Eq. 6.1.4), it is

[25] Or from n_T to $n_T(1 + n_m)$ if n_T torque levels are considered, see Section 4.1.5.

expected that the same conclusions as those drawn in Chapter 4 apply here, and that likewise the influence of the $\dot{\boldsymbol{V}}^i$-proportional terms can be neglected in most practical applications.

6.1.3 Optimal training of the reduced-order model

6.1.3.1 Three kinds of contact shapes

The first step in solving the misaligned gear contact problem is training the parametric ROM. The aim of this step is to collect a sufficiently rich set of contact shapes for representing quasi-static gear contact deformations during the simulation. Analogous to the perfectly-aligned ICS approach, these contact shapes are obtained by solving a series of static contact problems involving the FOM of the gear pair. For a flexibly supported gear pair that may misalign under the applications of external forces (see e.g. Fig. 6.2), three kinds of contact shapes are considered. These are the perfectly aligned, the rigidly misaligned, and the flexibly misaligned contact shapes.

The basic procedure for computing these contact shapes is the same, namely 1) constraining the driven gear at a selected angle of rotation θ_j^2, 2) adjusting the angle of rotation of the driving gear θ^1 accordingly such that the two gears start to engage, 3) applying an external torque T^1 to the driving gear, and 4) solving the resulting system of equilibrium equations to find the static displacement vectors of the gears. The difference between the three kinds of shapes lies in the treatment of the rigid-body coordinates, as shown in Table 6.1. The *perfectly aligned* contact shapes, which were introduced in Section 4.1.1, are obtained by perfectly aligning the gear pair and constraining all rigid-body coordinates except for the angle of rotation of the driving gear, θ^1. The system of equilibrium equations that subsequently needs to be solved is given by Eq. 4.1.4 and is repeated here for clarity (refer to Section 4.1.1 for the definition of the symbols):

$$\begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{K}_{FE}^1 & \mathbf{0} \\ sym. & & \boldsymbol{K}_{FE}^2 \end{bmatrix} \begin{Bmatrix} \theta_j^1 \\ \boldsymbol{\psi}_{c,j}^1 \\ \boldsymbol{\psi}_{c,j}^2 \end{Bmatrix} = \begin{Bmatrix} T^1 \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}_{c,\theta}^1 \\ \boldsymbol{f}_{c,f}^1 \\ \boldsymbol{f}_{c,f}^2 \end{Bmatrix} \tag{6.1.5}$$

The same system of equations is solved for obtaining the *rigidly misaligned* contact shapes, except that this time the gear pair is constrained in a misaligned configuration. Note that in that case the assumption of planar rigid-body motion is no longer valid ($x_3^i \neq 0$ and/or $\boldsymbol{\phi}^i \neq 0$ and/or $\psi^i \neq 0$). Therefore, instead of using the simplified expressions for the rotation matrix (Eq. 2.4.23) and generalized external force vector (Eq. 2.4.30), the general expressions of Eq. C.2 and Eq. 2.4.21 need to be used.

The third kind of contact shape – the *flexibly misaligned* contact shape – is obtained by considering the flexibly supported gear pair. Instead of imposing gear pair misalignments through constraints, the gears are allowed to misalign under the action of the externally applied torque, in accordance with the elastic deflection of their supports. The system of equilibrium equations that needs to be solved is given by

$$\begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & \boldsymbol{K}_{FE}^{1} & \mathbf{0} & \mathbf{0} \\ & & & \mathbf{0} & \mathbf{0} \\ sym. & & & & \boldsymbol{K}_{FE}^{2} \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_{r,j}^{1}\{1:5\} \\ \theta_j^1 \\ \boldsymbol{\psi}_{c,j}^{1} \\ \boldsymbol{q}_{r,j}^{2}\{1:5\} \\ \boldsymbol{\psi}_{c,j}^{2} \end{Bmatrix} = \begin{Bmatrix} \mathbf{0} \\ T^1 \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}_{c,r}^{1}\{1:5\} \\ \boldsymbol{f}_{c,\theta}^{1} \\ \boldsymbol{f}_{c,f}^{1} \\ \boldsymbol{f}_{c,r}^{2}\{1:5\} \\ \boldsymbol{f}_{c,f}^{2} \end{Bmatrix} - \begin{Bmatrix} \boldsymbol{f}_{b,r}^{1}\{1:5\} \\ 0 \\ \mathbf{0} \\ \boldsymbol{f}_{b,r}^{2}\{1:5\} \\ \mathbf{0} \end{Bmatrix} \quad (6.1.6)$$

where $\boldsymbol{q}_r^i$ is the vector of rigid-body or reference coordinates of the *i*-th gear, $\boldsymbol{f}_{b,r}^i = \boldsymbol{K}_b^i \Delta \boldsymbol{q}_r^i$ are the bushing forces acting on the center of that gear (see Eq. 3.2.5), and the curled brackets { } indicate the intended entries of the associated vector.

Contact shape	Constrained rigid-body coordinates	# rigid-body DOFs	Origin of misalignment
Perfectly aligned	All but θ^1	1	None
Rigidly misaligned	All but θ^1	1	Imposed
Flexibly misaligned	Only θ^2	11	Elastic deflection support

Table 6.1 Comparison of different kinds of contact shapes

6.1.3.2 Training of the parametric ROM using a Greedy approach

Following the distinction between rigidly and flexibly misaligned contact shapes, there are two versions of the misaligned ICS approach. In the ICS approach based on flexibly misaligned contact shapes, the interpolation scheme is the same as for the perfectly aligned ICS approach (Eq. 6.1.4) and the training of the parametric ROM follows the same reasoning as outlined in Section 4.1.4. Such an approach can e.g. be used when it is assumed that the misalignments that occur under static conditions will be similar to the dynamic misalignments.

In the ICS approach based on rigidly misaligned contact shapes, a number of n_m misaligned contact shapes is computed for each angular configuration θ_j^1 that is considered during the training of the ROM. During simulation, all contact shapes (misaligned as well as perfectly aligned) are interpolated based on the angular configuration parameter θ^1, as will be discussed in Section 6.1.4. For this interpolation to make physical sense, the interpolated contact shapes should share the same misalignment parameters $\Delta X_1^i, \Delta X_2^i, \Delta X_3^i, \Delta\phi^i$, and $\Delta\psi^i$. Denoting a given combination of misalignment parameters by Δ_k, the main challenge is in determining the set of misalignment parameter combinations $\Delta_k, k = 1, \dots, n_m$ that yields a robust ROM at an affordable training cost. In comes the Greedy sampling approach.

Given an initial set of k misaligned contact shapes, the Greedy sampling approach offers an effective means to identify the combination of misalignment parameters Δ_{k+1} that contributes

most to the set of k shapes. In the discrete variant of the Greedy approach, Δ_{k+1} is identified by performing an exhaustive search over a finite set of preselected parameter combinations. At each of these parameter combinations, the error of the current ROM (i.e. the ROM that is constructed based on the set of k misaligned contact shapes) is estimated and the parameter combination yielding the largest error is selected as Δ_{k+1}. The FOM is then evaluated at this parameter combination to compute the $(k+1)$-th misaligned contact shape, and the process is repeated with the updated ROM.

Evaluating the current ROM at a given parameter combination is relatively cheap as the number of elastic DOFs is small ($k \ll 100$ per gear) and the system of $(1+2k)$ static equilibrium equations needs to be solved only once. Nevertheless, in order to minimize the cost of the Greedy search algorithm, it is important to minimize the number of candidate parameter combinations. Physical insight into gear misalignment can be of great aid in properly selecting the misalignment parameters. In the introductory section it was discussed that parallel misalignment only slightly alters the contact area; therefore, a small number of samples in the $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ directions may suffice, if any at all (note that this also depends on the ability of the FE mesh to resolve small shifts in contact area). With regard to angular misalignments, pitch misalignments are known to have a bigger impact on the contact area than yaw misalignments; therefore it is reasonable to include a larger number of samples in the pitch direction of the misalignment space. Finally, axial misalignment is negligible in most applications and can thus be neglected in the training of the ROM. The orders of magnitude of the misalignment amplitudes can be estimated using the applied torque, the corresponding reaction forces at the centers of the gears, and the stiffness matrices of the supporting shafts.

Algorithm 6.1 gives a general overview of the Greedy-based training of the ROM. The inputs of the algorithm are the FE stiffness matrices $\boldsymbol{K}_{FE}^{i}$ and the full-factorial matrix of candidate parameter combinations Δ_{ff}. The latter is a $(n_{ff} \times 5)$ matrix whose rows are all the possible misalignment parameter combinations, i.e.

$$\Delta_{ff} = \begin{Bmatrix} \Delta_{ff,1} \\ \Delta_{ff,2} \\ \vdots \end{Bmatrix} = \begin{bmatrix} \Delta X_{1,1} & \Delta X_{2,1} & \Delta X_{3,1} & \Delta\phi_1 & \Delta\psi_1 \\ \Delta X_{1,2} & \Delta X_{2,2} & \Delta X_{3,2} & \Delta\phi_2 & \Delta\psi_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \tag{6.1.7}$$

Note that the misalignment parameters are defined for the gear pair as a whole, and not separately for the individual gears, which would lead to 10 instead of 5 misalignment parameters. In applying these misalignments to the individual gears, it is assumed that the gear pair misalignments are shared equally between the two gears (see Eq. 6.1.10). The goal of the Greedy sampling approach is to select a number of $n_m \ll n_{ff}$ parameter combinations that yields a ROM that is robust with respect to parameter variations. These parameter combinations are stored in the $(n_m \times 5)$ matrix of selected parameter combinations Δ that is initialized by setting it equal to the perfectly aligned condition $\Delta_0 = (0,0,0,0,0)$ (i.e. all misalignments are zero).

The algorithm loops over all angular configurations $\theta_j^1, j = 1, \dots, n_a$, where the locations of θ_j^1 within the sampling interval are determined as outlined in Section 4.1.4. For each angular configuration, the FOM (Eq. 6.1.6) is evaluated at all the parameter combinations in Δ and a ROM is constructed based on the obtained (misaligned) contact shapes $\boldsymbol{\Psi}_{c,j}^i$. Next, the performance of this ROM is assessed by computing the reduced-order error estimator at all the remaining parameter combinations in Δ_{ff}. Analogous to Eq. 4.1.43, this error estimator is obtained by substitution of the reduced-order solution into the full-order residual:

$$\left\|\boldsymbol{r}(\theta_j^1, \Delta_{ff,l})\right\|_2 = \left\| \begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{K}_{FE}^1 & \mathbf{0} \\ sym. & & \boldsymbol{K}_{FE}^2 \end{bmatrix} \begin{Bmatrix} \theta_j^1 \\ \boldsymbol{\Psi}_{c,j}^1 \boldsymbol{q}_c^1 \\ \boldsymbol{\Psi}_{c,j}^2 \boldsymbol{q}_c^2 \end{Bmatrix} - \left(\begin{Bmatrix} T^1 \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}_{c,\theta}^1 \\ \boldsymbol{f}_{c,f}^1 \\ \boldsymbol{f}_{c,f}^2 \end{Bmatrix} \right) \right\| \tag{6.1.8}$$

where $\boldsymbol{q}_c^1$ and $\boldsymbol{q}_c^2$ are obtained by solving the reduced-order version of Eq. 6.1.5:

$$\begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} \\ & {\boldsymbol{\Psi}_{c,j}^1}^T \boldsymbol{K}_{FE}^1 \boldsymbol{\Psi}_{c,j}^1 & \mathbf{0} \\ sym. & & {\boldsymbol{\Psi}_{c,j}^2}^T \boldsymbol{K}_{FE}^2 \boldsymbol{\Psi}_{c,j}^2 \end{bmatrix} \begin{Bmatrix} \theta_j^1 \\ \boldsymbol{q}_c^1 \\ \boldsymbol{q}_c^2 \end{Bmatrix} - \left(\begin{Bmatrix} T^1 \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}_{c,\theta}^1 \\ {\boldsymbol{\Psi}_{c,j}^1}^T \boldsymbol{f}_{c,f}^1 \\ {\boldsymbol{\Psi}_{c,j}^1}^T \boldsymbol{f}_{c,f}^2 \end{Bmatrix} \right) \tag{6.1.9}$$

subjected to the following misalignment constraints:

$$X_1^i = \frac{(-1)^i}{2} \Delta X_{1,l} \ , \ X_2^i = \frac{(-1)^i}{2} \Delta X_{2,l} \ , \ X_3^i = \frac{(-1)^i}{2} \Delta X_{3,l} \ , \\ \phi^i = \frac{(-1)^i}{2} \Delta \phi_l \ , \ \psi^i = \frac{(-1)^i}{2} \Delta \psi_l \tag{6.1.10}$$

If the maximum estimated error is larger than the maximum allowable error *tol*, the parameter combination $\Delta_{ff,l}$ yielding the largest estimated error is added to the matrix of selected parameter combinations Δ. Then the FOM is evaluated at the corresponding misalignment configuration for the current and all previous angular configurations $\theta_j^1, j = 1, .., j$, before the algorithm continues. In that way it is guaranteed that the same combinations of misalignment parameters are used at all angular configurations θ_j^1, while making sure that the obtained ROM is robust for all the possible parameter combinations in Δ_{ff} across the entire angular sampling interval $\left[\theta_{min}^1, \theta_{max}^1\right]$.

Algorithm 6.1 terminates once the estimated error for all parameter combinations is below tol, which in this dissertation is defined as a fraction τ of the error estimator for the perfectly aligned gear pair, i.e. $tol = \tau \left\|\boldsymbol{r}(\theta_j^1, \Delta_0)\right\|_2$. In Table 6.2, the number of misalignment parameter variations (including the zero value) and the total number of (misaligned) contact shapes computed for the spur and helical gear pairs of Table 3.1 are shown. For the spur gear pair, 7 (misaligned) contact shapes were computed out of 36 possible parameter combinations, whereas for the helical

gear pair 13 contact shapes were computed out of 288 combinations. Note that the fraction τ was set to 0.1, which was observed to yield a good compromise between accuracy and number of shapes (see also Section 6.1.5).

Algorithm 6.1 Greedy-based training of the ROM in the misaligned ICS approach

Input: finite element stiffness matrices $\boldsymbol{K}_{FE}^{i}$, candidate parameter combinations Δ_{ff}

Output: set of (misaligned) contact shapes $\boldsymbol{\Psi}_{c}^{i} = \left[\boldsymbol{\Psi}_{c,1}^{i} \quad \boldsymbol{\Psi}_{c,2}^{i} \quad \dots \quad \boldsymbol{\Psi}_{c,n_a}^{i}\right]$

0. initialize: $\Delta = \Delta_0 = (0,0,0,0,0)$
1. **for** $j = 1 \dots n_\theta$ **do**
2. select next sampling point θ_j^1 (see Section 4.1.4)
3. initialize contact shapes matrix $\boldsymbol{\Psi}_{c,j}^{i} = [\,]$
4. **for** $k = 1 \dots \text{size}(\Delta,1)$ **do**
5. solve Eq. 6.1.6 for $\theta^1 = \theta_j^1$ and $\Delta_k = \Delta\{k,:\}$ to obtain $\boldsymbol{\psi}_{c,j,k}^{i}$
6. assemble contact shapes matrix $\boldsymbol{\Psi}_{c,j}^{i} = [\boldsymbol{\Psi}_{c,j}^{i} \quad \boldsymbol{\psi}_{c,j,k}^{i}]$
7. **end**
8. construct current ROM: $\boldsymbol{K}_{cc,j}^{i} = {\boldsymbol{\Psi}_{c,j}^{i}}^{T} \boldsymbol{K}_{FE}^{i} \boldsymbol{\Psi}_{c,j}^{i}$
9. **for** $l = 1 \dots \text{size}(\Delta_{ff}, 1)$ **do**
10. Evaluate Eq. 6.1.8 for θ_j^1 and $\Delta_{ff,l}$ to obtain $\left\|\boldsymbol{r}(\theta_j^1, \Delta_{ff,l})\right\|_2$
11. **end**
12. **if** $\max\left(\left\|\boldsymbol{r}(\theta_j^1, \Delta_{ff,l})\right\|_2\right) < tol$
13. add $\Delta_{ff,l}$ that maximizes error to Δ: $\Delta = [\Delta\,;\, \Delta_{ff,l}]$
14. remove $\Delta_{ff,l}$ from Δ_{ff}
15. return to step 1.
16. **end**
17. **end**

Gear pair	$\#\Delta X_1$	$\#\Delta X_2$	$\#\Delta X_3$	$\#\Delta\phi$	$\#\Delta\psi$	Full fact.	τ	$1+n_m$
Spur	2	2	1	3	3	36	0.1	7
Helical	3	3	2	4	4	288	0.1	13

Table 6.2 Number of misaligned contact shapes computed using Algorithm 6.1 for two example gear pairs

6.1.4 Interpolation of the misaligned contact shapes

In Section 6.1.3.1, three kinds of contact shapes were discussed: the perfectly aligned contact shape (see Section 4.1.1), the flexibly misaligned contact shape (Eq. 6.1.6) and the rigidly misaligned contact shape (Eq. 6.1.5 and Eq. 6.1.10). Depending on which type of contact shape is used, a

different variant of the ICS approach is obtained, as shown in the first three rows of Table 6.3. The Perfectly aligned ICS (PICS) approach and the Misaligned ICS approach based on Flexibly misaligned shapes (F-MICS) both use the linear interpolation scheme of Eq. 6.1.4 to obtain the interpolated contact shape(s). When used for solving the misaligned dynamics gear contact problem, there are no algorithmic differences between the PICS and F-MICS approach apart from the definition of the misaligned contact shapes.

ICS method	**Contact shapes**	**Parameters**	**Interpolation scheme**	**# DOFs**
PICS	Perfectly aligned	θ^1	Local in θ^1	1
F-MICS	Flexibly misaligned	θ^1	Local in θ^1	1
R-MICS	Rigidly misaligned	$\Delta X_1, \Delta X_2, \Delta X_3,$ $\Delta\phi, \Delta\psi, \theta^1$	Local in θ^1, global in other parameters	$1+n_m$
S-MICS	SVD of Rigidly misaligned	$\Delta X_1, \Delta X_2, \Delta X_3,$ $\Delta\phi, \Delta\psi, \theta^1$	Local in θ^1, global in other parameters	$r \ll 1 + n_m$

Table 6.3 Comparison of different kinds of ICS approaches

The Misaligned ICS approach based on Rigidly misaligned contact shapes (R-MICS) requires a somewhat different interpolation scheme than the other two approaches, due to the multitude of misaligned contact shapes that are computed at each angular configuration θ_j^1. As was explained in Section 6.1.2, for reasons of computational and mathematical complexity it is opted not to use a multivariate interpolation scheme to interpolate all misaligned contact shapes based on the actual misalignment parameters. Instead, all misaligned contact shapes are interpolated based on the single parameter θ^1. This only makes physical sense when the various sets of misaligned contact shapes computed at the different angular configurations θ_j^1 correspond to the same misalignment parameter combinations, as is ensured by the Greedy-based training algorithm discussed in the previous section. In that case, the interpolation scheme of Eq. 6.1.4 is replaced by the following variant:

$$\boldsymbol{\Psi}_c^i(\theta^1) = [1 - p(\theta^1)]\boldsymbol{\Psi}_{c,j}^i + p(\theta^1)\boldsymbol{\Psi}_{c,j+1}^i \in \mathbb{R}^{N\times(1+n_m)} \tag{6.1.11}$$

where $\boldsymbol{\Psi}_c^i$ is the matrix of interpolated misaligned contact shapes of the *i*-th gear, and $\boldsymbol{\Psi}_{c,j}^i$ and $\boldsymbol{\Psi}_{c,j+1}^i$ are the matrices of misaligned contact shapes corresponding to the angular configurations θ_j^1 and θ_{j+1}^1. The interpolation parameter $p(\theta^1)$ is the same as in the PICS approach and is given by Eq. 4.1.10.

The matrix of interpolated misaligned contact shapes is composed of one perfectly aligned contact shape and n_m misaligned contact shapes, resulting in $(1 + n_m)$ elastic DOFs (in addition to the n_k modal elastic DOFs). In the numerical examples discussed at the end of the previous section, it was shown that the number n_m can be as high as 20, leading to a significant increase in the number of DOFs as compared to the PICS and F-MICS approaches. One very effective way to compress the dimensions of a high-dimensional matrix is by using the Singular Value Decomposition (SVD, see Section 2.3.3.2). Applying this decomposition to $\boldsymbol{\Psi}_{c,j}^{i}$ yields:

$$\boldsymbol{\Psi}_{c,j}^{i} = \boldsymbol{U}_{c,j}^{i}\boldsymbol{\Sigma}_{c,j}^{i}{\boldsymbol{D}_{c,j}^{i}}^{T} \tag{6.1.12}$$

where the columns of the unitary matrices $\boldsymbol{U}_{c,j}^{i} \in \mathbb{R}^{N\times N}$ and $\boldsymbol{D}_{c,j}^{i} \in \mathbb{R}^{(1+n_m)\times(1+n_m)}$ denote the left- and right-singular vectors of $\boldsymbol{\Psi}_{c,j}^{i}$, respectively, and $\boldsymbol{\Sigma}_{c,j}^{i} \in \mathbb{R}^{N\times(1+n_m)}$ is a diagonal rectangular matrix whose diagonal entries are the singular values $\sigma_{c,j,k}^{i}$, for $k = 1, \dots, (1 + n_m)$, of $\boldsymbol{\Psi}_{c,j}^{i}$, with $\sigma_{c,j,1}^{i} \geq \sigma_{c,j,2}^{i} \geq \cdots \geq \sigma_{c,j,1+n_m}^{i}$. Let $\overline{\boldsymbol{U}}_{c,j}^{i}$, $\overline{\boldsymbol{\Sigma}}_{c,j}^{i}$ and ${\overline{\boldsymbol{D}}_{c,j}^{i}}^{T}$ denote the matrices that are obtained by retaining only the first *r* columns of $\boldsymbol{U}_{c,j}^{i}$ and the first *r* rows of $\boldsymbol{\Sigma}_{c,j}^{i}$ and ${\boldsymbol{D}_{c,j}^{i}}^{T}$, respectively, then the Theorem of Eckart-Young states that the optimal *r*-rank approximation of $\boldsymbol{\Psi}_{c,j}^{i}$ in the Frobenius norm is given by $\overline{\boldsymbol{\Psi}}_{c,j}^{i} = \overline{\boldsymbol{U}}_{c,j}^{i}\overline{\boldsymbol{\Sigma}}_{c,j}^{i}{\overline{\boldsymbol{D}}_{c,j}^{i}}^{T}$, and that the Frobenius norm of the difference between $\boldsymbol{\Psi}_{c,j}^{i}$ and $\overline{\boldsymbol{\Psi}}_{c,j}^{i}$ is given by the sum of the discarded squared singular values [49]. The optimal *r*-rank approximation of the subspace spanned by the columns of $\boldsymbol{\Psi}_{c,j}^{i}$ is given by the span of the *r* left-singular vectors in $\boldsymbol{U}_{c,j}^{i}$ corresponding to the *r* largest singular values of $\boldsymbol{\Psi}_{c,j}^{i}$. Consequently, if *r* is chosen sufficiently high so as to yield an accurate representation of $\boldsymbol{\Psi}_{c,j}^{i}$, yet small enough such that $r \ll 1 + n_m$, the dimensions of the ROM can be significantly reduced by using the misaligned singular vectors $\overline{\boldsymbol{U}}_{c,j}^{i}$ instead of $\boldsymbol{\Psi}_{c,j}^{i}$ to represent quasi-static tooth deformations, and thus by replacing Eq. 6.1.11 by :

$$\overline{\boldsymbol{U}}_{c}^{i}(\theta^1) = [1 - p(\theta^1)]\overline{\boldsymbol{U}}_{c,j}^{i} + p(\theta^1)\overline{\boldsymbol{U}}_{c,j+1}^{i} \in \mathbb{R}^{N\times r} \tag{6.1.13}$$

where $\overline{\boldsymbol{U}}_{c}^{i}(\theta^1)$ is the matrix of interpolated misaligned singular vectors. The thus obtained ICS approach is referred to as the Misaligned ICS approach based on Singular vectors (S-MICS, see last row in Table 6.3). For the spur gear pair of Table 6.2, Fig. 6.4 shows the three dominant left-singular vectors at a particular angular configuration. Note that due to the symmetric sampling of misalignments, the dominant singular vector closely resembles the perfectly aligned contact shape.

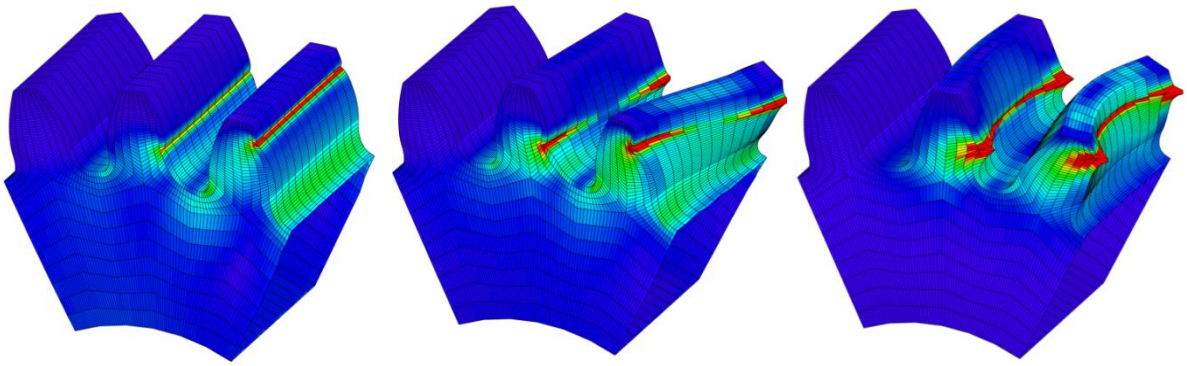


Fig. 6.4 Left-singular vectors corresponding to the largest singular values of the SVD of the matrix of misaligned contact shapes of a spur gear

While the linear interpolation of misaligned contact shape matrices $\boldsymbol{\Psi}_{c,j}^{i}$ and $\boldsymbol{\Psi}_{c,j+1}^{i}$ in Eq. 6.1.11 is meaningful given their common set of misalignment parameter combinations, it is not clear whether this commonality is preserved after performing the SVDs of $\boldsymbol{\Psi}_{c,j}^{i}$ and $\boldsymbol{\Psi}_{c,j+1}^{i}$. In order to improve the consistency of the interpolation scheme, as well as to avoid mode crossing phenomena, a *congruence transformation* can be applied to all $\bar{\boldsymbol{U}}_{c,j}^{i}$ prior to the actual online interpolation. Given a basis matrix $\boldsymbol{V}_j \in \mathbb{R}^{N\times r}$, the congruence transformation of $\boldsymbol{V}_j$ is given by $\tilde{\boldsymbol{V}}_j = \boldsymbol{V}_j\boldsymbol{T}_j$ where $\boldsymbol{T}_j \in \mathbb{R}^{r\times r}$ is a square matrix that replaces each column in $\boldsymbol{V}_j$ by linear combinations of its columns. In the S-MICS approach, a modified version of the congruence transformation proposed by Amsallem & Farhat [55] (see Section 2.3.2.2) is implemented. In this version, the goal is to find the transformation matrix $\boldsymbol{T}_j$ for each angular configuration θ_j^1 that minimizes the Frobenius norm of the difference between the transformed matrix of misaligned singular vectors at the next configuration, $\tilde{\boldsymbol{U}}_{c,j+1}^{i}$, and the matrix of misalignment singular vectors at the current configuration, $\bar{\boldsymbol{U}}_{c,j}^{i}$, i.e.

$$\boldsymbol{T}_j = arg\,min\left\|\bar{\boldsymbol{U}}_{c,j+1}^{i}\boldsymbol{T}_j - \bar{\boldsymbol{U}}_{c,j}^{i}\right\|_F^2 \qquad \text{subject to } \boldsymbol{T}_j^T\boldsymbol{T}_j = \boldsymbol{I}_r \tag{6.1.14}$$

This minimization problem is equivalent to optimizing the Modal Assurance Criterion (MAC) between $\tilde{\boldsymbol{U}}_{c,j+1}^{i}$ and $\bar{\boldsymbol{U}}_{c,j}^{i}$ (see [78]), and is analytically solved by the SVD of the matrix $\bar{\boldsymbol{U}}_{c,j+1}^{i\,T}\bar{\boldsymbol{U}}_{c,j}^{i} = \boldsymbol{U}_{c,jj}^{i}\boldsymbol{\Sigma}_{c,jj}^{i}\boldsymbol{D}_{c,jj}^{i\,T} \in \mathbb{R}^{r\times r}$ [174]:

$$\boldsymbol{T}_j = \boldsymbol{U}_{c,jj}^{i}\boldsymbol{D}_{c,jj}^{i\,T} \in \mathbb{R}^{r\times r} \tag{6.1.15}$$

The general procedure for obtaining the transformed matrices of misaligned singular vectors $\tilde{\boldsymbol{U}}_{c,j}^{i}$ is outlined in Algorithm 6.2. Note that the determination of *r* in Step 4 of the algorithm is based on the Frobenius norm of $\boldsymbol{\Psi}_{c,j}^{i}$, which can be interpreted as a measure of the elastic potential

energy stored in the misaligned contact shapes. A possible way of selecting r is then to find the minimal value of r that captures the bulk of this energy in the r-rank approximations of all the misaligned contact shape matrices. Finally, it is worth mentioning that in all of the considered study cases, the transformation matrices are nearly-diagonal matrices with diagonal values close to 1 or -1 and relatively small off-diagonal terms (<0.01), indicating that, notwithstanding possible sign changes, the singular vectors corresponding to adjacent angular configurations are already expressed in nearly-optimal bases.

6.1.5 Numerical validation

In order to validate the proposed extension of the ICS approach to the case of misaligned gears, the parallel-axis gear pairs of Table 3.1 are mounted on flexible shafts and a series of dynamic simulations is conducted using the ICS approaches of Table 6.3, in addition to the CMS reference approach and the modal transformation method. The simulation parameters are the same as the ones listed in Table 4.3, including the parameter settings for the fixed-increment central difference integrator. In each of these simulations the geometrical parameters of the shaft are $L = 200$ mm, $\emptyset = 30$ mm, and $z_g/L = 0.3$ (see Fig. 6.2). The stiffness matrices of the shafts are obtained as outlined in Section 6.1.1 (see also Section 3.2.2), and the damping matrices are obtaining by multiplying the diagonal terms of the stiffness matrix by 1e-5, setting the off-diagonal terms to zero (this matrix accounts for the internal damping in the shaft and the elasto-hydrodynamic damping in the supporting bearings). The main parameters of the MOR schemes and the associated simulation times for the spur and helical gear pairs are shown in Table 6.4 and Table 6.5. In the R-MICS and S-MICS approaches, the greedy algorithm (Algorithm 6.1) was used to selected the misalignment parameter combinations, using the settings of Table 6.2. In the S-MICS approach, the number of retained singular values was 3 for both the spur and helical gear pair example. It can be observed from the tables below that similar gains in computational efficiency as in Chapter 4 are obtained, with speed-up factors between 60 and 70 due solely to the reduced number of FLOPs required in forming the update equations (the influence of the reduced frequency content of the ROMs is investigated in Chapter 7).

The Dynamic Transmission Errors (DTEs) of the spur and helical gear pairs are shown in Fig. 6.5 and Fig. 6.6, respectively. While the angular misalignment of the gear pairs is relatively small ($\phi \approx 0.03°$ and $\psi \approx 0.01°$ for the spur gear pair), the Perfectly-aligned ICS (PICS) approach fails to correctly capture the tooth deformations. Both the Misaligned ICS (MICS) approaches on the other hand are in very good agreement with the CMS-based reference model, with relative errors of no more than 3% in both simulations.

Algorithm 6.2 Congruence transformation of misaligned contact shapes

Input: misaligned contact shape matrices $\boldsymbol{\Psi}_{c,j}^{i}$ for $j = 1, \dots, n_\theta$

Output: congruence transformations of the singular vectors $\widetilde{\boldsymbol{U}}_{c,j}^{i}$ for $j = 1, \dots, n_\theta$

1. **for** $j = 1 \dots n_\theta$ **do**
2. compute SVD of $\boldsymbol{\Psi}_{c,j}^{i} = \boldsymbol{U}_{c,j}^{i} \boldsymbol{\Sigma}_{c,j}^{i} {\boldsymbol{D}_{c,j}^{i}}^{T}$
3. **end**
4. find r based on the Frobenius norms of $\boldsymbol{\Psi}_{c,j}^{i}$ and $\overline{\boldsymbol{\Psi}}_{c,j}^{i}$ for $j = 1, \dots, n_\theta$
5. truncate the left-singular vectors to obtain $\overline{\boldsymbol{U}}_{c,j}^{i} = \boldsymbol{U}_{c,j}^{i}\{:, 1{:}r\}$
6. set $\widetilde{\boldsymbol{U}}_{c,1}^{i} = \overline{\boldsymbol{U}}_{c,1}^{i}$
7. **for** $j = 1 \dots n_\theta - 1$ **do**
8. compute ${\overline{\boldsymbol{U}}_{c,j+1}^{i}}^{T} \overline{\boldsymbol{U}}_{c,j}^{i} = \boldsymbol{U}_{c,jj}^{i} \boldsymbol{\Sigma}_{c,jj}^{i} {\boldsymbol{D}_{c,jj}^{i}}^{T}$
9. compute $\boldsymbol{T}_j = \boldsymbol{U}_{c,jj}^{i} {\boldsymbol{D}_{c,jj}^{i}}^{T}$
10. perform congruence transformation: $\widetilde{\boldsymbol{U}}_{c,j+1}^{i} = \overline{\boldsymbol{U}}_{c,j+1}^{i} \boldsymbol{T}_j$
11. **end**

Parameters	**CMS**	**MOD**	**PICS**	**R-MICS**	**S-MICS**
Number of angular samples	n.a.	n.a.	20	20	20
Number of torques	n.a.	n.a.	1	1	1
Number of contact shapes per angle	n.a.	n.a.	1	7	3
Number of eigenvectors	50	1000	50	50	50
Number of DOFs per gear	2081	1006	57	63	59
Pre-processing CPU time	1148 s	4935 s	1423 s	1867 s	1877 s
Processing CPU time	463527 s	125473 s	7227 s	7765 s	7345 s
Speed-up factor w.r.t. CMS	1	3.7	64	60	63

Table 6.4 Parameters and CPU times of the dynamic misaligned spur gear pair simulations

Parameters	**CMS**	**MOD**	**PICS**	**R-MICS**	**S-MICS**
Number of angular samples	n.a.	n.a.	20	20	20
Number of torques	n.a.	n.a.	1	1	1
Number of contact shapes per angle	n.a.	n.a.	1	13	3
Number of eigenvectors	50	1000	50	50	50
Number of DOFs in driving gear	3956	1006	57	69	59
Pre-processing CPU time	1945 s	7648 s	2648 s	3275 s	3291 s
Processing CPU time	2043291 s	634922 s	29761 s	32462 s	30482 s
Speed-up factor w.r.t. CMS	1	3.2	69	63	67

Table 6.5 Parameters and CPU times of the dynamic misaligned helical gear pair simulations

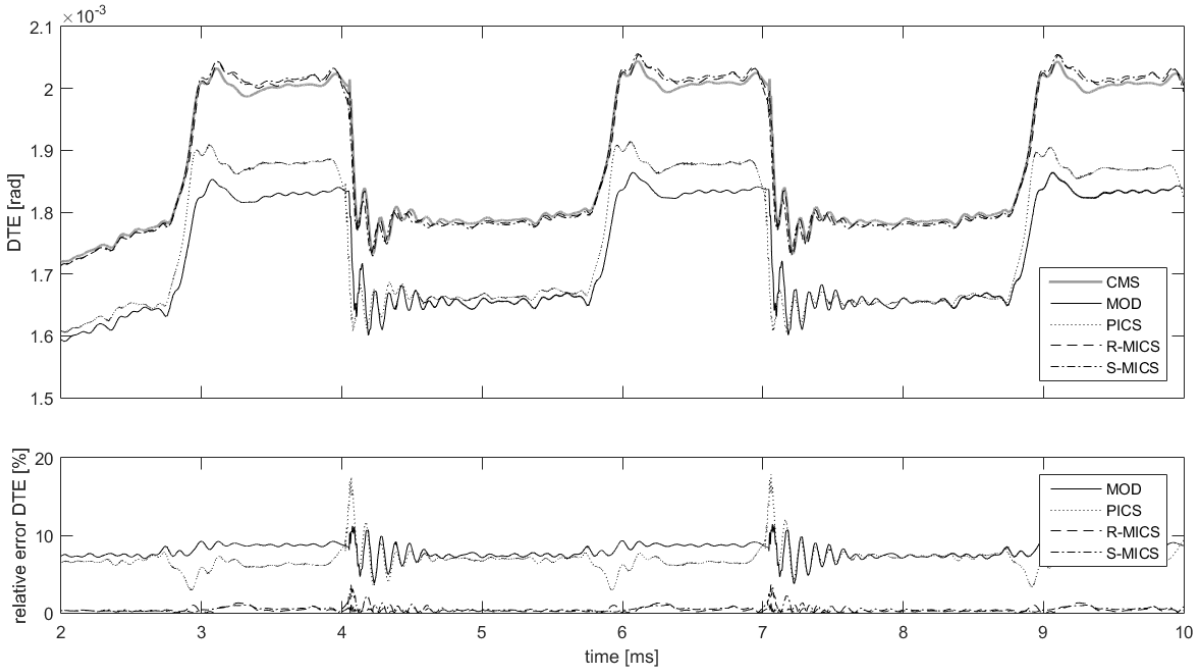


Fig. 6.5 Comparison of the DTE predictions of a misaligned spur gear pair by the MOR schemes of Table 6.4

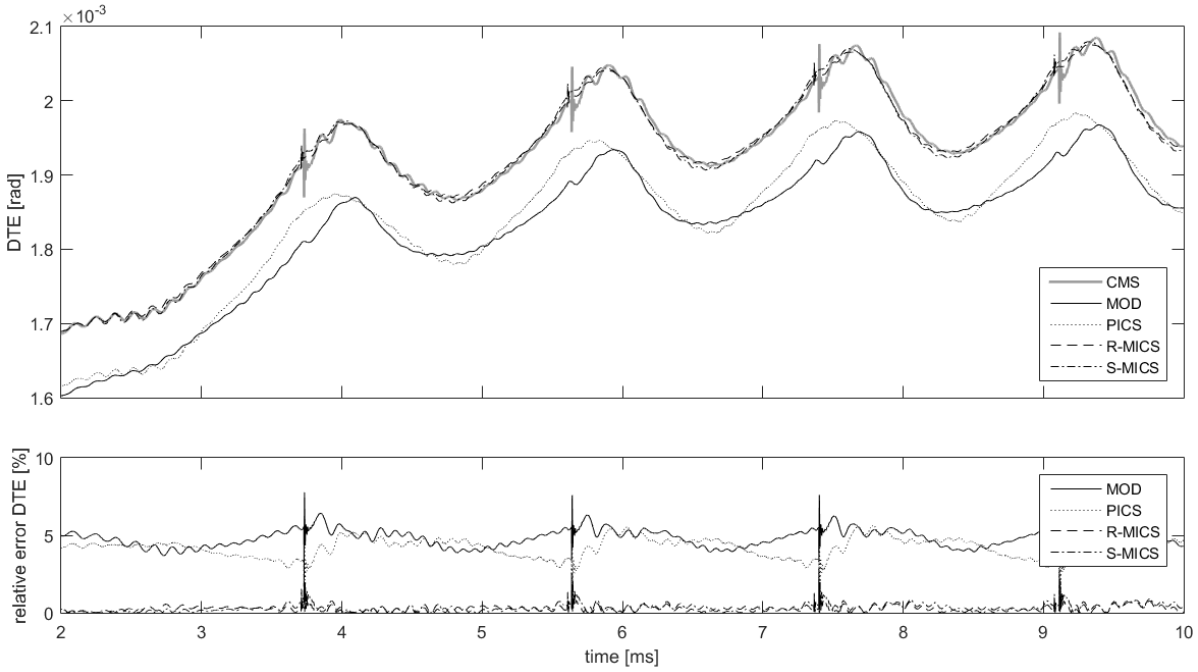


Fig. 6.6 Comparison of the DTE predictions of misaligned helical gear pair by the MOR schemes of Table 6.5

The accuracy of the various ROMs is further investigated by observing the evolution of the normal surface stresses in Fig. 6.7 and Fig. 6.8. In Fig. 6.7 the evolution of normal stresses at a point halfway along the tooth flank is plotted for the leftmost and rightmost cross-sections of the spur gear pair. Due to the gear pair's angular misalignment, the normal stresses are concentrated at the right side of the flank, as is accurately predicted by both of the MICS approaches. As was observed in Section 4.4.1, the MOD approach fails to resolve local deformations despite the multitude of eigenvectors in the reduction basis, leading to large errors on stress level. Not

surprisingly, the PICS approach predicts equal stresses at both ends of the tooth flank and underestimates peak stresses by roughly 40%.

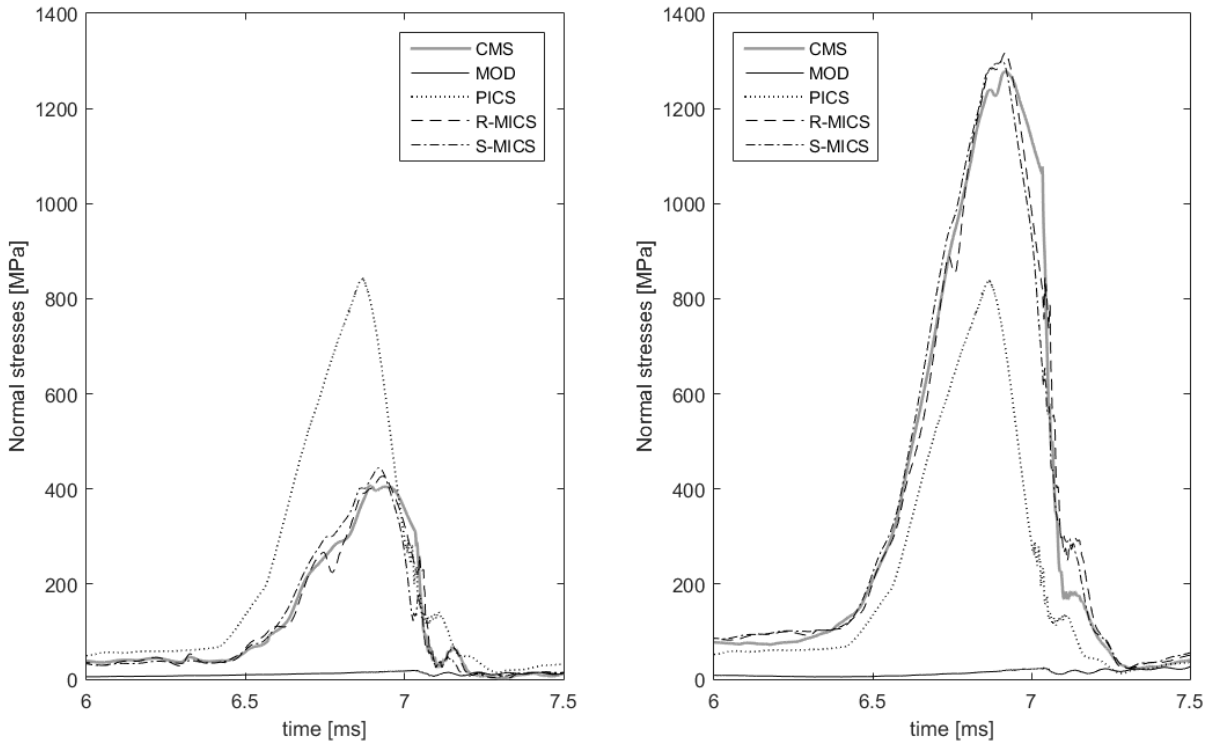


Fig. 6.7 Comparison of the contact stress predictions at a point along the flank of a misaligned spur gear pair by the MOR schemes of Table 6.5; the left figure shows the stresses at the outer left side of the tooth flank; the right figure shows the stresses at the outer right side of the tooth flank

In Fig. 6.8 the evolution of normal stresses at a point along the root is shown for the leftmost and rightmost cross-sections of the spur gear pair. The R-MICS and S-MICS approach are again in very good agreement with the CMS-based reference model. The MOD approach overestimates the peak root stresses as a means to compensate for the lack of local deformation near the point of contact, as was also observed in Section 4.4.1. The PICS approach again fails to capture the correct left-right stress balance and significantly underestimates the peak root stresses. The same conclusions can be drawn from Fig. 6.9, where the Von Mises stresses of the spur gear are plotted at time t = 5 ms.

Summarizing, the Misaligned ICS approach yields displacement and stress results with similar accuracy as the standard CMS method, yet requiring less than 2% of the computational effort (when applying the same numerical integrator). As the S-MICS approach is shown to have similar accuracy as the R-MICS approach, this method becomes the reference MOR method for misaligned contact problems. Note that as compared to the perfectly-aligned ICS approach of Chapter 4, the number of DOFs in the S-MICS approach was increased by no more than 4 (out of 118) in the examples of Table 6.4 and Table 6.5, thereby demonstrating that the curse of dimensionality has little effect on the ICS approach.

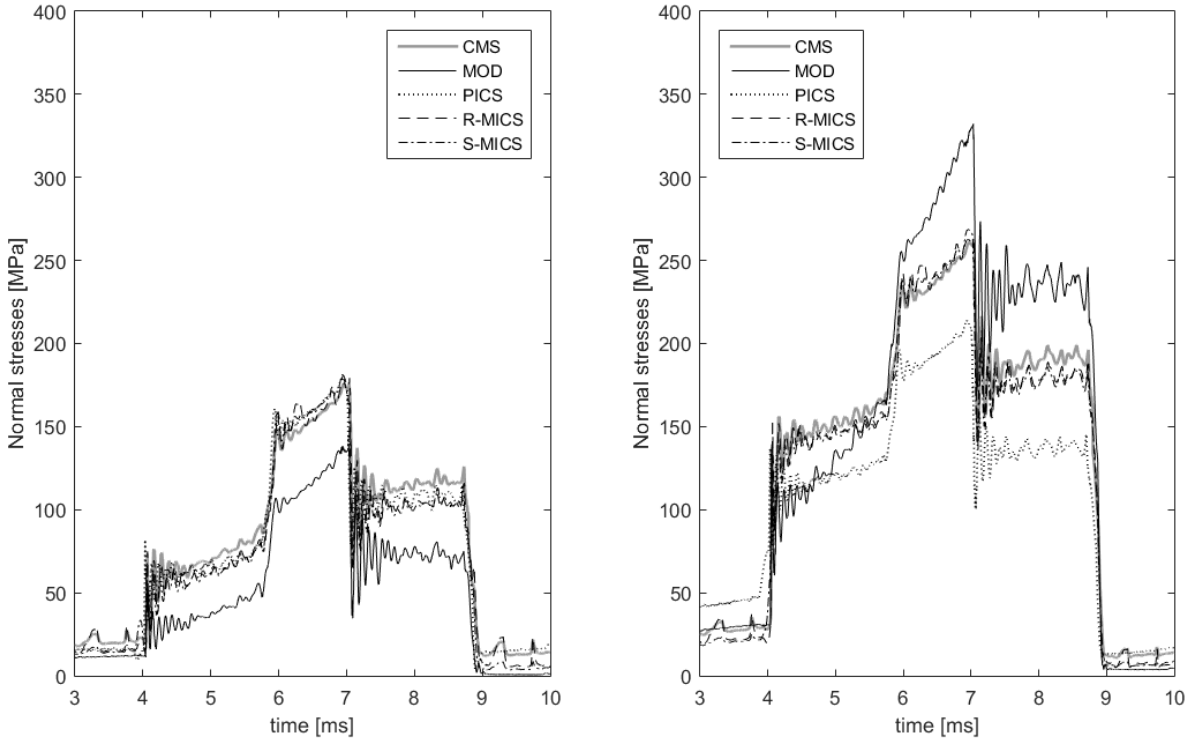


Fig. 6.8 Comparison of the contact stress predictions at a point in the root of a misaligned spur gear pair by the MOR schemes of Table 6.5; the left figure shows the stresses at the outer left side of the tooth root; the right figure shows the stresses at the outer right side of the tooth root

6.2 Model-order reduction of planetary gear sets

6.2.1 Preliminary considerations and solution outline

Planetary gear sets are preferred over parallel-axis gear trains when the available space and permissible weight of the transmission are small compared to the intended speed ratio and input torque. They attain a higher power transmission efficiency than parallel-axis gears and a superior load distribution by spreading the torque over multiple planet gears, thereby increasing the total contact surface of the transmission. Despite the complexity of their design, planetary gear sets are ubiquitous in wind turbine, helicopter and automotive transmissions.

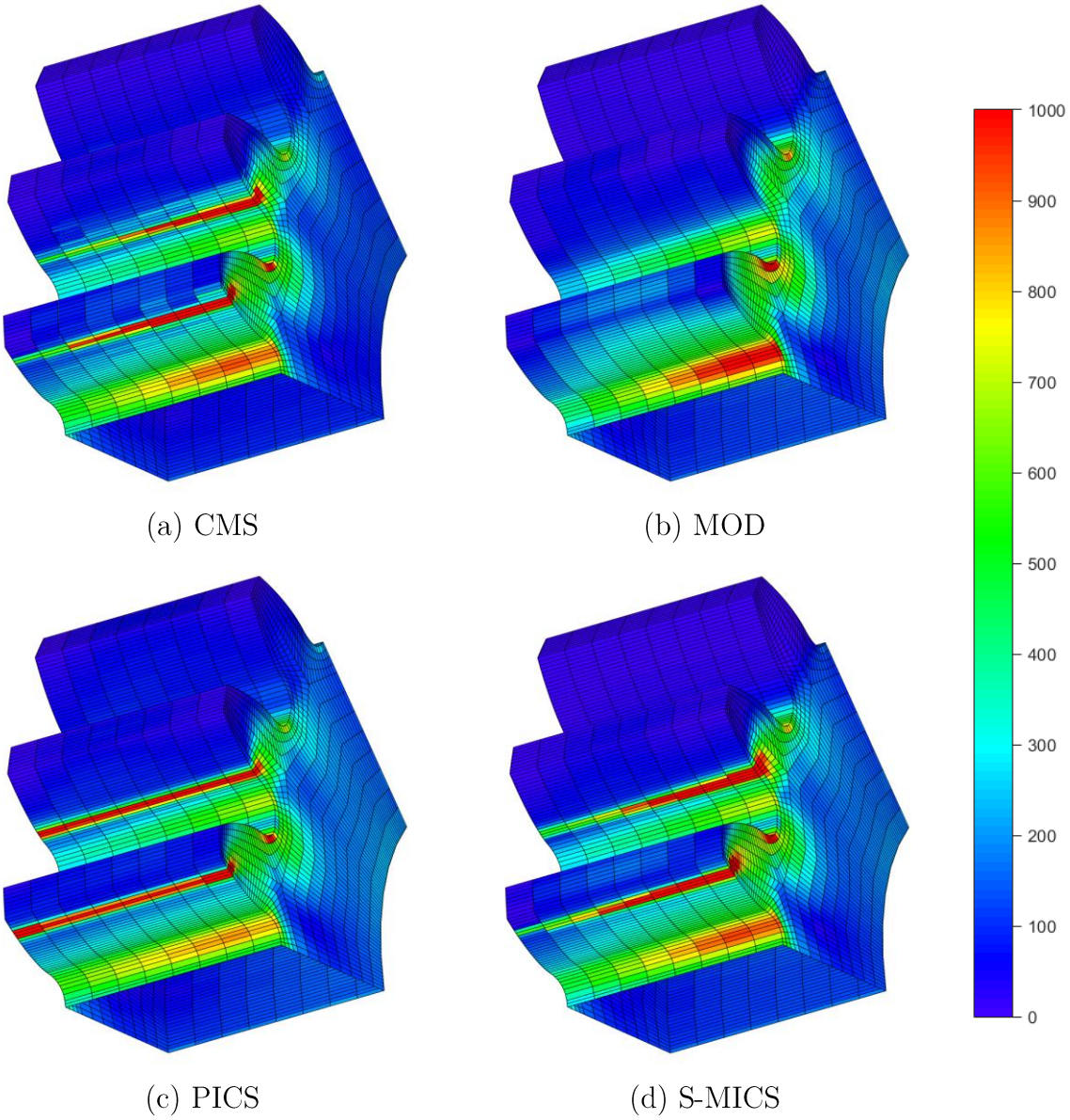


Fig. 6.9 Comparison of the predicted stress distributions in the misaligned spur gear pair at t = 5 ms by the MOR schemes of Table 6.5

In this section, the MOR method presented in Chapter 4 to efficiently solve parallel-axis gear contact problems is extended to the case of planetary gears. The scope of this investigation is limited to the analysis of perfectly aligned gears, allowing to focus on the new challenges posed by the planetary gear application. These challenges are the development of an Interpolated Contact Shapes (ICS) approach for gears that simultaneously engage with multiple gears, and the efficient solution of the Differential-Algebraic Equations (DAEs) that describe their dynamic behavior.

The planetary gear model that is considered in this dissertation was briefly introduced in Section 3.2.5 and it schematically represented in Fig. 6.10. The vector of generalized coordinates describing the configuration of this gear set is given by:

$$\boldsymbol{q}^T = \{\underbrace{\theta^c}_{\boldsymbol{q}^c} \quad \underbrace{\theta^s \quad \boldsymbol{q}_f^{s\,T}}_{\boldsymbol{q}^s} \quad \underbrace{X_1^1 \quad X_2^1 \quad \theta^1 \quad \boldsymbol{q}_f^{1\,T}}_{\boldsymbol{q}^1} \quad \underbrace{X_1^2 \quad X_2^2 \quad \theta^2 \quad \boldsymbol{q}_f^{2\,T}}_{\boldsymbol{q}^2} \quad \underbrace{X_1^3 \quad X_2^3 \quad \theta^3 \quad \boldsymbol{q}_f^{3\,T}}_{\boldsymbol{q}^3} \quad \underbrace{\boldsymbol{q}_f^{r\,T}}_{\boldsymbol{q}^r}\} \tag{6.2.1}$$

where the superscripts '*r*', '*c*', '*s*,' '1', '2', and '3' refer to the ring gear (1, see Fig. 6.10), planet carrier (2), sun gear (4), first, second and third planet gear (3), respectively. It can be observed from this vector that the ring gear is held stationary, the planet carrier is modelled as a rigid body, the sun gear rotates about a fixed axis in space, and the planet gears undergo large rigid-body motion as they orbit the sun gear. Given the rigidity of the planet carrier, the rigid-body translations X_1^p and X_2^p of the planet gears are completely determined by the rotation of the planet carrier, as is expressed by the following constraint equation:

$$\boldsymbol{g}(\boldsymbol{q}) = \begin{cases} X_1^p - r_p^c \, sin\left(\theta^c + (p-1)\dfrac{2\pi}{3}\right) = 0 \\ \\ X_2^p - r_p^c \, cos\left(\theta^c + (p-1)\dfrac{2\pi}{3}\right) = 0 \end{cases} \qquad for\ p = 1 \ldots 3 \tag{6.2.2}$$

where r_p^c is the pitch radius of the planet carrier, i.e. the distance from the center of the carrier to the center of the planet gears.

In most geared wind turbines, the aerodynamic torque generated by the turbine rotor is applied to the planet carrier and transmitted to the sun gear, which is ultimately connected to the power generator. In helicopter gearboxes, where planetary gear sets are typically composed of five instead of three planet gears, the turboshaft engine torque is applied to the sun gear and transmitted to the planet carrier that is directly connected to the helicopter blades. When the planet carrier is used as input shaft, the transmission and torque ratio of carrier-sun pair is given by

$$\frac{\omega^s}{\omega^c} = 1 + \frac{z^r}{z^s} \qquad \frac{T^s}{T^c} = \left(1 + \frac{z^r}{z^s}\right)^{-1} = \frac{z^s}{z^r + z^s} \tag{6.2.3}$$

where $\omega^s = \dot{\theta}^s$ and $\omega^c = \dot{\theta}^c$ are the angular velocities of the sun gear and planet carrier, T^c and T^s are the input and output torques applied to the carrier and sun gear, and z^s and z^r are the tooth numbers of the sun and ring gear, respectively. Note that the planet carrier and sun gear rotate in the same direction. Other important transmission and torque ratios are the carrier-planet transmission ratio and the carrier-ring torque ratio:

$$\frac{\omega^p}{\omega^c} = 1 - \frac{z^r}{z^p} = -\frac{z^r - z^p}{z^p} \qquad \frac{T^r}{T^c} = \frac{z^r}{z^r + z^s} \tag{6.2.4}$$

where $\omega^p = \dot{\theta}^p$ is the angular velocity of the planet gears and T^r is the reaction torque of the ring gear. Note that the carrier-sun ratios are not influenced by the number of teeth of the planet gears;

in fact these do not carry any torque, they are merely used as force transmitters. The above relationships will prove to be important in the development of an ICS approach for planetary gears.

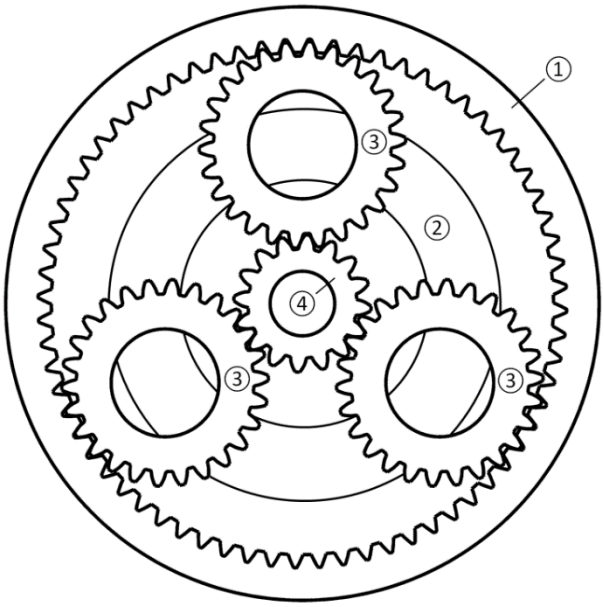


Fig. 6.10 Schematic representation of the planetary gear model

They are essentially two ways to construct an ICS approach for planetary gears in the spirit of Chapter 4. In the ICS approach for parallel-axis gears, contact shapes are computed by applying a constant torque to the input gear, fixing the rotation of the output gear and computing the resulting gear deformations; during simulation these contact shapes are then interpolated based on the rotation angle of the input gear. Similarly, planetary gear contact shapes can be obtained by applying a constant torque to the planet carrier, fixing the rotation of the sun gear and computing the resulting deformations of the planet, sun and ring gears; then during simulation these contact shapes are interpolated based on the rotation angle of the planet carrier. This approach will be referred to as the ICS-I approach.

Whereas the ICS-I approach deals with six gear meshes at the same time (three planet-ring meshes and three planet-sun meshes), the second approach – further referred to as the ICS-II approach – considers one gear mesh at the time. Contact shapes are obtained by isolating the sun-planet and ring-planet gear pairs and computing the gear deformations as in the parallel-axis ICS approach; then for each gear pair during simulation these contact shapes are interpolated based on the rotation angle of the designated reference gear. Both the ICS-I and ICS-II approaches are discussed and numerically validated in the upcoming sections. As in Section 6.1, the influence of the configuration-dependency of the basis matrix $\boldsymbol{V}^i$ on the EOMs will not be explicitly investigated (for a derivation of these equations in the case of planar rigid-body motion, including all $\dot{\boldsymbol{V}}^i$–proportional terms, see [175]).

6.2.2 Computation and interpolation of the contact shapes

6.2.2.1 The ICS-I approach: a multi-mesh approach

In the ICS-I approach, the contact shapes of the sun, ring and planet gears are computed by fixing the rotation angle θ^s of the sun gear, applying a torque T^c to the planet carrier, and computing the resulting gear deformations. The system of equilibrium equations that needs to be solved when the planet carrier is at an angle θ_j^c can be written as

$$\underbrace{\begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{K}_{FE}^s & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & & \boldsymbol{K}_{FE}^1 & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & & & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & & & & \boldsymbol{K}_{FE}^2 & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & & & & & & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & sym. & & & & & & \boldsymbol{K}_{FE}^3 & \mathbf{0} \\ & & & & & & & & \boldsymbol{K}_{FE}^r \end{bmatrix}}_{\boldsymbol{K}} \begin{Bmatrix} \theta_j^c \\ \boldsymbol{\psi}_{c,j}^s \\ \boldsymbol{q}_{r,j}^1 \\ \boldsymbol{\psi}_{c,j}^1 \\ \boldsymbol{q}_{r,j}^2 \\ \boldsymbol{\psi}_{c,j}^2 \\ \boldsymbol{q}_{r,j}^3 \\ \boldsymbol{\psi}_{c,j}^3 \\ \boldsymbol{\psi}_{c,j}^r \end{Bmatrix} = \begin{Bmatrix} T^c \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}_{c,\theta}^c \\ \boldsymbol{f}_{c,f}^s \\ \boldsymbol{f}_{c,r}^1 \\ \boldsymbol{f}_{c,f}^1 \\ \boldsymbol{f}_{c,r}^2 \\ \boldsymbol{f}_{c,f}^2 \\ \boldsymbol{f}_{c,r}^3 \\ \boldsymbol{f}_{c,f}^3 \\ \boldsymbol{f}_{c,f}^r \end{Bmatrix} \tag{6.2.5}$$

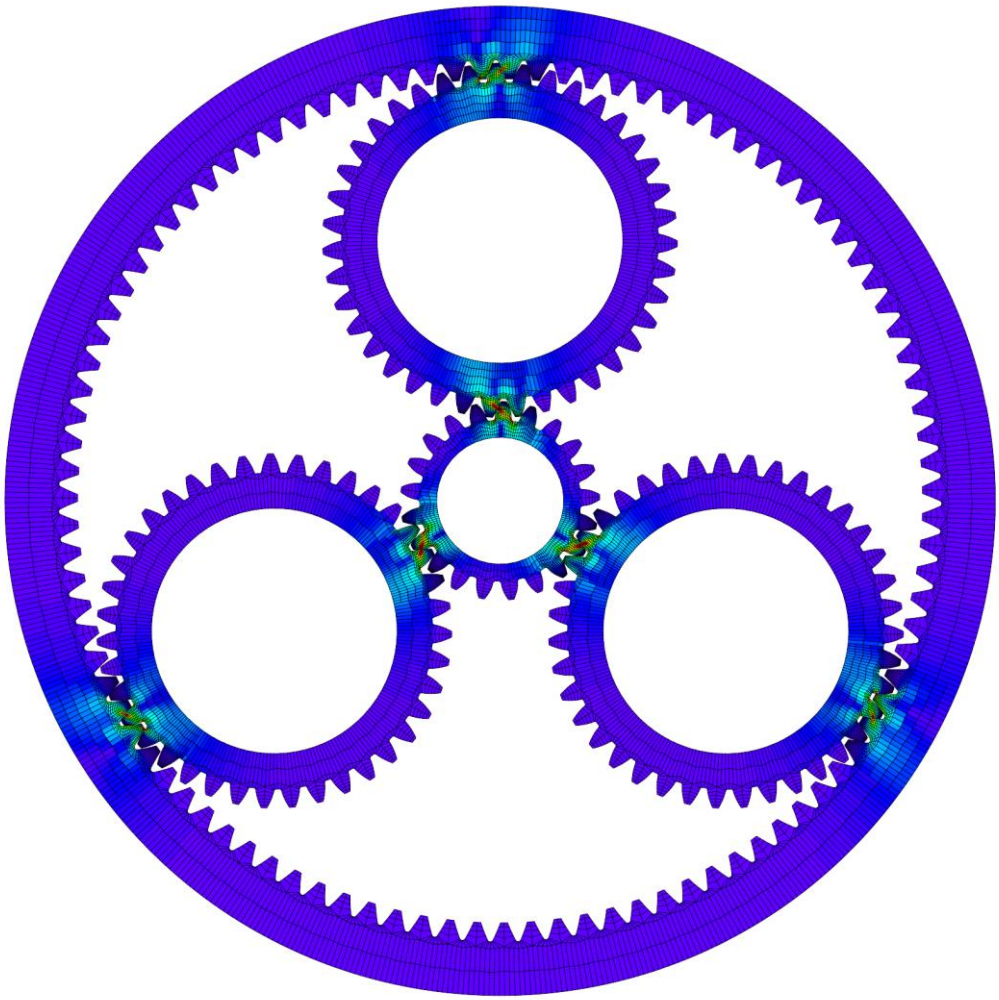
subjected to $\boldsymbol{g}(\boldsymbol{q})$ (see Eq. 6.2.2), where $\boldsymbol{q}_r^p = \{X_1^p, X_2^p, \theta^p\}$ is the vector of rigid-body coordinates of the *p*-th planet gear. This equation is solved using the following iteration matrix:

$$\boldsymbol{J} := \begin{bmatrix} \boldsymbol{K} - \partial \boldsymbol{f}_c / \partial \boldsymbol{q} & \boldsymbol{G}(\boldsymbol{q})^T \\ \boldsymbol{G}(\boldsymbol{q}) & \mathbf{0} \end{bmatrix} \tag{6.2.6}$$

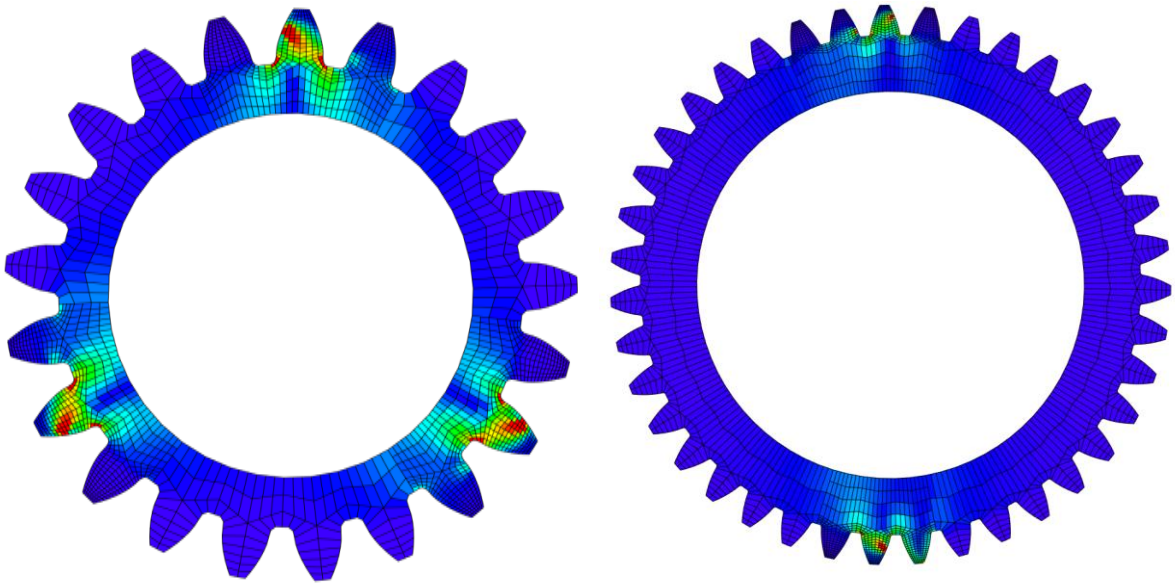
where the constraint Jacobian $\boldsymbol{G}(\boldsymbol{q}) = \partial \boldsymbol{g}(\boldsymbol{q}) / \partial \boldsymbol{q}$ is obtained by deriving Eq. 6.2.2 w.r.t. $\boldsymbol{q}$:

$$\boldsymbol{G}(\boldsymbol{q}) = \begin{bmatrix} -r_p^c \cos\theta^c & 0 & \mathbf{0} & 1 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ r_p^c \sin\theta^c & 0 & \mathbf{0} & 0 & 1 & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ -r_p^c \cos(\theta^c + 2\pi/3) & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & 1 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ r_p^c \sin(\theta^c + 2\pi/3) & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & 0 & 1 & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ -r_p^c \cos(\theta^c + 4\pi/3) & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & 1 & 0 & 0 & \mathbf{0} & \mathbf{0} \\ r_p^c \sin(\theta^c + 4\pi/3) & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & \mathbf{0} & 0 & 1 & 0 & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{6.2.7}$$

Note that in Eq. 6.2.6 the second column of $\boldsymbol{G}(\boldsymbol{q})$ should be removed as the fixed rotation angle of the sun gear is eliminated from the vector of generalized coordinates $\boldsymbol{q}$ in Eq. 6.2.5. In Fig. 6.11, the multi-mesh contact shapes of the planetary gear set introduced in Section 3.2.5 (Table 3.2) are shown (note that this is a helical gear set, with only the left-most slice shown in the figure below).



(a)



(b) (c)

Fig. 6.11 Planetary gear contact shapes in the ICS-I approach: (a) static contact equilibrium, (b) contact shape sun gear, (c) contact shape planet gear

In the form presented in Eq. 6.2.5, the system of static equilibrium equations requires considerable computational resources for its solution. Not only is the planetary gear model comprised of five finite element models, but there are six simultaneous gear contact meshes to be considered. In Chapter 4, several ways of accelerating the computation of contact shapes were discussed, including the analytical derivation of the contact force Jacobian $\partial \boldsymbol{f}_c/\partial \boldsymbol{q}$, the use of an intermediate ROM based on attachment modes, and a mixed-grid formulation that improves memory usage as well as computational complexity. In addition to these measures, the computation of the contact Jacobian can be accelerated by using parallel processing techniques to simultaneously compute the contact forces in the six independent gear meshes, as well as by neglecting certain submatrices that represent weak coupling between generalized coordinates (e.g. the mutual influence of the ring-planet gear meshes can safely be neglected when computing the contact Jacobian).

An important aspect of computing contact shapes is determining the locations of the sampling points θ_j^c. In the parallel-axis ICS approach, the angular interval to be sampled was narrowed down considerably by exploiting the rotational periodicity of a parallel-axis gear pair, limiting the length of the sampling interval to $\tau^1 = 2\pi/z^1$, where τ^1 and z^1 are the angular pitch and the number of teeth of the driving gear. The question then is whether there exists a fundamental periodic unit in a planetary gear set that can be used to minimize the length of the sampling interval, and thereby limit the number of contact shapes that need to be computed. As it turns out, the configuration of a planetary gear set is geometrically indistinguishable from its original configuration when the planet carrier rotates over the angular pitch of the ring gear, i.e. $\Delta\theta^c = \tau^r = 2\pi/z^r$. This can be observed by substituting the ring's angular pitch in the incremental form of Eq. 6.2.3:

$$\Delta\theta^s = \Delta\theta^c\left(1+\frac{z^r}{z^s}\right) = \frac{2\pi}{z^r}\left(1+\frac{z^r}{z^s}\right) = \frac{2\pi}{z^r}+\frac{2\pi}{z^s} = \tau^r+\tau^s \tag{6.2.8}$$

The above equation states that when the planet carrier rotates over τ^r, the sun gear follows this rotation (τ^r) adding its angular pitch τ^s. Likewise, substituting $\Delta\theta^c = \tau^r$ in the planet-carrier transmission ratio yields

$$\Delta\theta^p = \Delta\theta^c\left(1-\frac{z^r}{z^p}\right) = \frac{2\pi}{z^r}\left(1-\frac{z^r}{z^p}\right) = \frac{2\pi}{z^r}-\frac{2\pi}{z^p} = \tau^r-\tau^p \tag{6.2.9}$$

which corresponds to a planet rotation over τ^r in the direction of $\Delta\theta^c$ followed by a rotation over the planet's angular pitch τ^p in the opposite direction. For an observer travelling along with the planet carrier, there is no geometrical difference between the configuration of the gear pair at $\Delta\theta^c = 0$ and $\Delta\theta^c = \tau^r$ (assuming the crown of teeth as well as the gear blanks are rotationally periodic, and notwithstanding the fact that the teeth numbers of the engaging teeth have shifted by one unit). Hence, the length of the interval that should be sampled during the contact shapes

computation is τ^r. The determination of the sampling points θ_j^c within this interval proceeds as explained in Section 4.1.4.

During simulation, the contact shapes of the various gears are interpolated based on the angular configuration of the planet carrier θ^c, using the well-known linear interpolation formula:

$$\boldsymbol{\psi}_c^i(\theta^c) = [1 - p(\theta^c)]\boldsymbol{\psi}_{c,j}^i + p(\theta^c)\boldsymbol{\psi}_{c,j+1}^i \qquad \text{for } i = s,1,2,3,r \tag{6.2.10}$$

where $p(\theta^c)$ is the interpolation parameter that is defined similarly as in Chapter 4:

$$p(\theta^c) = \frac{\theta^c - \theta_j^c}{\theta_{j+1}^c - \theta_j^c} = \frac{\theta^c - \theta_j^c}{\Delta\theta_j^c}\,, \qquad 0 \le p \le 1 \tag{6.2.11}$$

When the planet carrier travels over one angular pitch of the ring gear, the contact shapes corresponding to the sun, ring and planet gears are rotated over the angular pitches τ^s, τ^r, and τ^p, respectively. This transformation can either be carried out online or offline based on the expected number of pitches travelled by the carrier. As the mass and stiffness invariants of flexible bodies are invariant under rotations, it suffices to simply rotate the displacements of the nodes on the active tooth flanks of the gears, as these are required to evaluate the generalized contact forces.

6.2.2.2 The ICS-II approach: a single-mesh approach

While the ICS-I approach leads to the least amount of elastic DOFs per gear and comes with an interpolation strategy that is relatively easy to implement, it suffers from three main drawbacks, namely:

- The computation of contact shapes involves six simultaneous gear contact meshes, five coupled finite element models, and six constraint equations;
- The interpolation of contact shapes is based on the configuration of the planet carrier alone and neglects the influence of dynamic gear ratio changes;
- The deformations of gear meshes within one gear (three meshes in the sun and ring gears and two meshes in the planet gears) are inherently coupled.

The ICS-II approach eliminates these drawbacks by considering one gear mesh at the time and by interpolating contact shapes on a pair-per-pair base[26]. The two pairs of gears that need to be

[26] By now, the attentive reader should have noticed that the ICS-II approach is to ICS-I what the ITCS approach (Chapter 4) for parallel-axis gears is to the ICS approach of Chapter 4.

considered when computing the contact shapes are the sun-planet and ring-planet pairs. As the planet gears in a planetary ring gear do not carry torque, these gears are held fixed while a torque is applied to the sun and ring gear, respectively. The equilibrium equations that need to be solved are similar as those encountered when dealing with parallel-axis gears:

$$\begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{K}_{FE}^{s} & \mathbf{0} \\ sym. & & \boldsymbol{K}_{FE}^{p} \end{bmatrix} \begin{Bmatrix} \theta_j^s \\ \boldsymbol{\psi}_{c,j}^{s} \\ \boldsymbol{\psi}_{c,j}^{ps} \end{Bmatrix} = \begin{Bmatrix} T^s \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}_{c,\theta}^{s} \\ \boldsymbol{f}_{c,f}^{s} \\ \boldsymbol{f}_{c,f}^{p} \end{Bmatrix} \tag{6.2.12a}$$

$$\begin{bmatrix} 0 & \mathbf{0} & \mathbf{0} \\ & \boldsymbol{K}_{FE}^{r} & \mathbf{0} \\ sym. & & \boldsymbol{K}_{FE}^{p} \end{bmatrix} \begin{Bmatrix} \theta_j^r \\ \boldsymbol{\psi}_{c,j}^{r} \\ \boldsymbol{\psi}_{c,j}^{pr} \end{Bmatrix} = \begin{Bmatrix} T^r \\ \mathbf{0} \\ \mathbf{0} \end{Bmatrix} + \begin{Bmatrix} \boldsymbol{f}_{c,\theta}^{r} \\ \boldsymbol{f}_{c,f}^{r} \\ \boldsymbol{f}_{c,f}^{p} \end{Bmatrix} \tag{6.2.12b}$$

where $\boldsymbol{\psi}_{c,j}^{ps}$ and $\boldsymbol{\psi}_{c,j}^{pr}$ are the contact shapes of the planet gear computed by considering the sun-planet and ring-planet pairs, respectively. Assuming perfect load sharing among the planet gears, the torques applied to the sun and ring gear can be expressed in terms of the planet carrier torque as follows:

$$T^s = \frac{T^c}{3}\frac{z^s}{z^r + z^s} \qquad\qquad T^r = \frac{T^c}{3}\frac{z^r}{z^r + z^s} \tag{6.2.13}$$

As in the parallel-axis ICS approach, the length of the angular interval that needs to be sampled is equal to the angular pitch of the driving gear, in casu: those of the ring and sun gear, respectively. Note that as compared to the ICS-I approach, two considerably smaller systems of unconstrained static equilibrium equations involving a single gear mesh need to be solved, leading to a reduced offline CPU cost. The computation of the planetary gear contact shapes in the ICS-II approach is shown in Fig. 6.12.

During the actual simulation, the obtained contact shapes are interpolated on a pair-per-pair base, with the planet gear as the reference gear in each pair. Instead of summing contact shapes belonging to the same gear, an elastic DOF is assigned to each contact shape, thereby allowing the algorithm to weigh the contribution of each contact shape based on the actual contact conditions. In doing so, the number of elastic DOFs used to represent quasi-static gear deformations equals the number of simultaneous gear meshes, in casu: three for the sun and ring gears and two for the planet gears. Using the familiar linear interpolation scheme, the matrices of interpolated contact shapes for the sun, ring and planet gears are obtained as

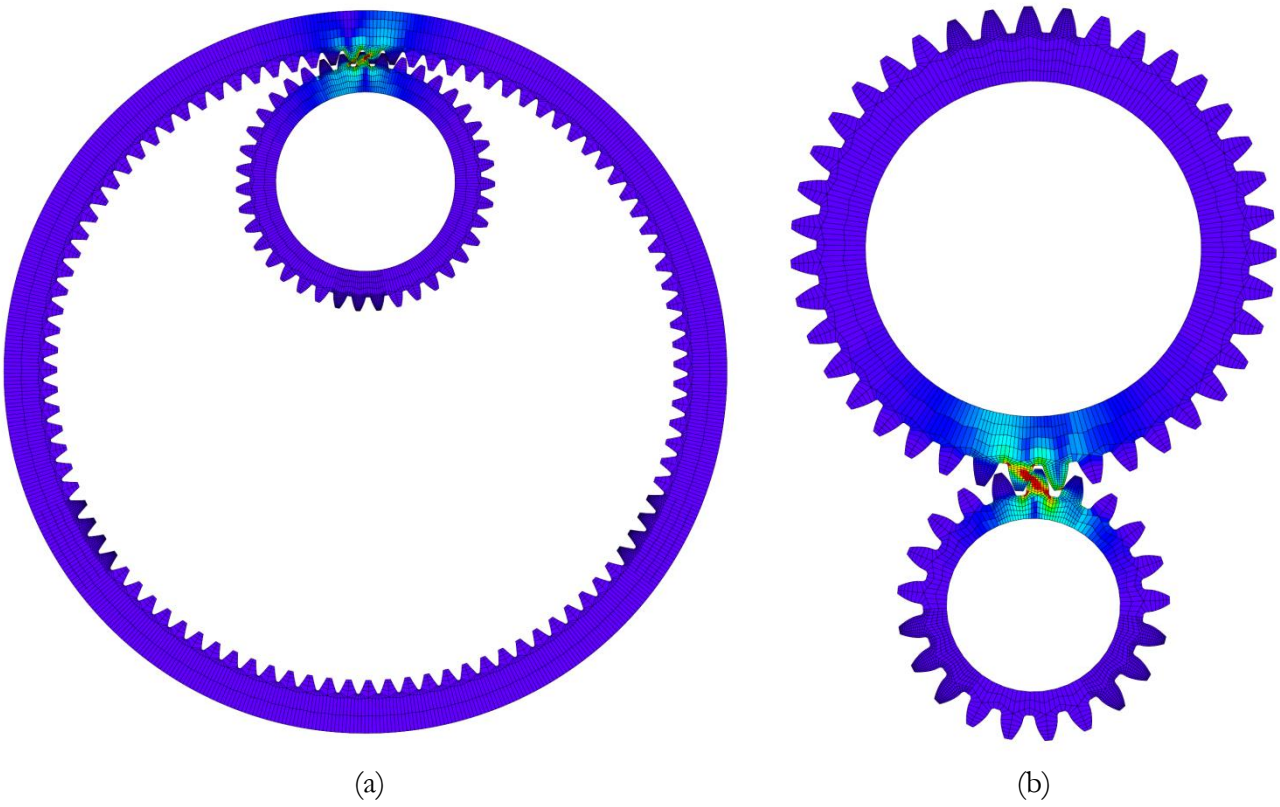
$$\boldsymbol{\Psi}_c^i(\theta^1,\theta^2,\theta^3,\theta^c) = \begin{Bmatrix} \left([1-p_1^i(\theta^1,\theta^c)]\boldsymbol{\psi}_{c,j_{1i}}^i + p_1^i(\theta^1,\theta^c)\boldsymbol{\psi}_{c,j_{1i}+1}^i\right)^T \\ \left([1-p_2^i(\theta^2,\theta^c)]\boldsymbol{\psi}_{c,j_{2i}}^i + p_2^i(\theta^2,\theta^c)\boldsymbol{\psi}_{c,j_{2i}+1}^i\right)^T \\ \left([1-p_3^i(\theta^3,\theta^c)]\boldsymbol{\psi}_{c,j_{3i}}^i + p_3^i(\theta^3,\theta^c)\boldsymbol{\psi}_{c,j_{3i}+1}^i\right)^T \end{Bmatrix}^T \quad \text{for } i = s,r \tag{6.2.14a}$$

$$\boldsymbol{\Psi}_c^p(\theta^p,\theta^c)=\begin{Bmatrix}\left(\left[1-p_p^s(\theta^p,\theta^c)\right]\boldsymbol{\psi}_{c,j_{ps}}^{ps}+p_p^s(\theta^p,\theta^c)\boldsymbol{\psi}_{c,j_{pr}+1}^{ps}\right)^T\\ \left(\left[1-p_p^r(\theta^p,\theta^c)\right]\boldsymbol{\psi}_{c,j_{pr}}^{pr}+p_p^r(\theta^p,\theta^c)\boldsymbol{\psi}_{c,j_{ps}+1}^{pr}\right)^T\end{Bmatrix}^T \qquad \text{for } p=1,2,3 \quad (6.2.14b)$$

where the interpolation parameters $p_p^i(\theta^p,\theta^c)$ are defined as

$$p_p^i(\theta^p,\theta^c)=\frac{[rem(\Delta\theta^p-\Delta\theta^c,\tau^p]-\Delta\theta_{j_{pi}}^p}{\Delta\theta_{j_{pi}+1}^p-\Delta\theta_{j_pi}^p} \qquad \text{for } i=s,r \text{ and } p=1,2,3 \qquad (6.2.15)$$

In this equation, $\theta_{j_{pi}}^p$ is the angle of the p-th planet gear in the planet-sun ($i=s$) or planet-ring ($i=r$) gear pair at the j_{pi}-th sampling point. The shape numbers j_{pi}, for $i=s,r$ and $p=1,2,3$, depend on the relative angle of the p-th planet gear with respect to the planet carrier, $\theta^p-\theta^c$.



(a) (b)

Fig. 6.12 Planetary gear contact shapes in the ICS-II approach: (a) static contact equilibria ring gear – planet gear, (b) sun gear – planet gear

From Eq. 6.2.14-6.2.15 it is clear that the interpolation of contact shapes in the ICS-II approach is considerably more involved as in the ICS-I approach. No fewer than 6 interpolation parameters p_p^i and 6 shape numbers j_{pi} need to be determined for computing the interpolated contact shapes, while the number of elastic DOFs per gear is increased as compared to the ICS-I approach. On the other hand, the ICS-II approach offers greater versatility due to the decoupling

of gear mesh deformations and gear ratio's, in addition to a lower offline computational cost. A full comparison of the ICS-I and ICS-II approach is given in Table 6.6.

ICS method	contact shapes	independent parameters	# dependent parameters	# elastic DOFs
ICS-I	system-level	θ^c	2 (p, j)	5
ICS-II	pair-by-pair	$\theta^1, \theta^2, \theta^3, \theta^c$	12 (p_p^i, j_{pi})	12

Table 6.6 Comparison of different kinds of ICS approaches for planetary gears

6.2.3 Elimination of the constraint equations

The dynamic behavior of the planetary gear set is described by a system of index-3 DAEs of the following general form (neglecting generalized damping forces):

$$\begin{aligned} \boldsymbol{M\ddot{q}} + \boldsymbol{Kq} + \boldsymbol{G}^T\boldsymbol{\lambda} &= \boldsymbol{f}_e + \boldsymbol{f}_c + \boldsymbol{f}_v \\ \boldsymbol{g}(\boldsymbol{q}) &= \boldsymbol{0} \end{aligned} \tag{6.2.16}$$

where $\boldsymbol{\lambda} \in \mathbb{R}^6$ is the vector of Lagrange multipliers associated with the constraint equations $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$, and $\boldsymbol{G}^T\boldsymbol{\lambda}$ is the vector of associated reaction forces. One solution approach is to integrate these DAEs using one of the fully-implicit integration schemes discussed in Section 2.6.3.2. Another approach is to first convert the system of DAEs to ODE form by eliminating the constraint equations (see Appendix B), and to integrate the resulting system of unconstrained ODEs by one of the ODE integrators discussed in Sections 2.6.1 and 2.6.2. In transient contact analyses, implicit integration schemes are incompatible with the CMS approach due to the high cost of the linear solves that need to be performed at each time step. Therefore in this chapter Eq. 6.2.16 is converted to ODE form and integrated using the central difference integrator that was selected as the reference integrator in Section 4.3. The implicit solution of Eq. 6.2.16 will be considered in Chapter 7.

A first step in eliminating the 6 constraint equations is the partitioning of the vector of generalized coordinates $\boldsymbol{q} \in \mathbb{R}^{n_q}$ into a vector of independent and dependent coordinates, $\boldsymbol{q}_i \in \mathbb{R}^{n_q-6}$ and $\boldsymbol{q}_d \in \mathbb{R}^6$. Selecting the rigid-body planet displacements as dependent coordinates, Eq. 6.2.1 can be rearranged as follows:

$$\boldsymbol{q}^T = \left\{ \underbrace{\theta^c \quad \theta^s \quad \boldsymbol{q}_f^{s\,T} \quad \theta^1 \quad \boldsymbol{q}_f^{1\,T} \quad \theta^2 \quad \boldsymbol{q}_f^{2\,T} \quad \theta^3 \quad \boldsymbol{q}_f^{3\,T} \quad \boldsymbol{q}_f^{r\,T}}_{\boldsymbol{q}_i^T} \quad \underbrace{X_1^1 \quad X_2^1 \quad X_1^2 \quad X_2^2 \quad X_1^3 \quad X_2^3}_{\boldsymbol{q}_d^T} \right\} \tag{6.2.17}$$

Applying the same permutations and partitioning to Eq. 6.2.16, this equation can be written in partitioned form as

$$\begin{bmatrix} \boldsymbol{M}_{ii} & \boldsymbol{M}_{id} \\ \boldsymbol{M}_{id} & \boldsymbol{M}_{dd} \end{bmatrix} \begin{Bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{q}}_d \end{Bmatrix} + \begin{bmatrix} \boldsymbol{K}_{ii} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_i \\ \boldsymbol{q}_d \end{Bmatrix} + [\boldsymbol{G}_i \quad \boldsymbol{G}_d]^T \boldsymbol{\lambda} = \begin{Bmatrix} (\boldsymbol{f}_e)_i \\ \boldsymbol{0} \end{Bmatrix} + \begin{Bmatrix} (\boldsymbol{f}_c)_i \\ (\boldsymbol{f}_c)_d \end{Bmatrix} + \begin{Bmatrix} (\boldsymbol{f}_v)_i \\ (\boldsymbol{f}_v)_d \end{Bmatrix} \qquad (6.2.18)$$
$$\boldsymbol{g}(\boldsymbol{q}_i, \boldsymbol{q}_d) = \boldsymbol{0}$$

where $\boldsymbol{G}_i$ and $\boldsymbol{G}_d$ can be obtained from Eq. 6.2.7 as

$$\boldsymbol{G}_i = \begin{bmatrix} -r_p^c \cos\theta^c & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} \\ r_p^c \sin\theta^c & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} \\ -r_p^c \cos(\theta^c + 2\pi/3) & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} \\ r_p^c \sin(\theta^c + 2\pi/3) & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} \\ -r_p^c \cos(\theta^c + 4\pi/3) & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} \\ r_p^c \sin(\theta^c + 4\pi/3) & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} & 0 & \boldsymbol{0} \end{bmatrix}, \qquad \boldsymbol{G}_d = \boldsymbol{I}_6 \qquad (6.2.19)$$

Using the generalized coordinates partitioning method (see Appendix B), the vector of generalized accelerations $\ddot{\boldsymbol{q}}$ can be expressed in terms of the vector of independent generalized accelerations using Eq. B.4:

$$\ddot{\boldsymbol{q}} = \begin{Bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{q}}_d \end{Bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n_q - 6} \\ -\boldsymbol{G}_i \end{bmatrix} \ddot{\boldsymbol{q}}_i - \begin{Bmatrix} \boldsymbol{0} \\ (\boldsymbol{G}\dot{\boldsymbol{q}})_{\boldsymbol{q}} \dot{\boldsymbol{q}} \end{Bmatrix} = \boldsymbol{T}_{di} \ddot{\boldsymbol{q}}_i - \begin{Bmatrix} \boldsymbol{0} \\ \boldsymbol{\kappa} \end{Bmatrix} \qquad (6.2.20)$$

where for the planetary gear set $\boldsymbol{\kappa}$ is defined as

$$\boldsymbol{\kappa} = r_p^c (\theta^c)^2 \begin{Bmatrix} \sin\theta^c \\ \cos\theta^c \\ \sin(\theta^c + 2\pi/3) \\ \cos(\theta^c + 2\pi/3) \\ \sin(\theta^c + 4\pi/3) \\ \cos(\theta^c + 4\pi/3) \end{Bmatrix} \qquad (6.2.21)$$

Substitution of Eq. B.4 in Eq. 6.2.18 and pre-multiplication of the latter with $\boldsymbol{T}_{di}^T$ eliminates the Lagrange multipliers ($\boldsymbol{T}_{di}\boldsymbol{G}^T = \boldsymbol{0}$) and uncouples the dependent and independent coordinates in the EOMs:

$$\widetilde{\boldsymbol{M}}_{ii} \ddot{\boldsymbol{q}}_i + \boldsymbol{K}_{ii} \boldsymbol{q}_i = (\boldsymbol{f}_e)_i + (\boldsymbol{f}_c)_i + (\boldsymbol{f}_v)_i - \boldsymbol{G}_i^T [(\boldsymbol{f}_c)_d + (\boldsymbol{f}_v)_d + m^p \boldsymbol{\kappa}] + \boldsymbol{M}_{id} \boldsymbol{\kappa} \qquad (6.2.22)$$

where the mass matrix $\widetilde{\boldsymbol{M}}_{ii}$ is defined as

$$\widetilde{\boldsymbol{M}}_{ii} = \boldsymbol{M}_{ii} + \begin{bmatrix} \boldsymbol{G}_i^T \boldsymbol{M}_{dd} \boldsymbol{G}_i & -\boldsymbol{G}_i^T \boldsymbol{M}_{di} \\ -\boldsymbol{M}_{id} \boldsymbol{G}_i & \boldsymbol{0} \end{bmatrix} = \begin{bmatrix} 3m^p \left(r_p^c\right)^2 & -\boldsymbol{G}_i^T \boldsymbol{M}_{di} \\ -\boldsymbol{M}_{id} \boldsymbol{G}_i & \boldsymbol{0} \end{bmatrix} \tag{6.2.23}$$

Eq. 6.2.22 represents a system of $n_q - 6$ coupled ODEs in $\boldsymbol{q}_i$ that can be solved using any ODE integrator. Note that the vector of generalized contact forces $\boldsymbol{f}_c$ is a function of the full vector of generalized coordinates $\boldsymbol{q}$; hence once $\boldsymbol{q}_i$ is obtained, the vector of dependent generalized coordinates $\boldsymbol{q}_d$ needs to be determined from $\boldsymbol{g}(\boldsymbol{q}_i, \boldsymbol{q}_d) = \boldsymbol{0}$. As the constraint equations are linear in $\boldsymbol{q}_d$ (see Eq. 6.2.2), this does not require an iterative solution procedure.

6.2.4 Numerical validation

The performance of the multi-mesh ICS-I and ICS-II approaches are assessed by comparison with the CMS-based reference ROM and a modal ROM. For this analysis, the helical planetary gear set that was introduced in Section 3.2.5 (Table 3.2) is simulated. The parameters of this simulation are shown in Table 6.7. The planet carrier is acted upon by a torque of 360 kNm (corresponding to the nominal aerodynamic torque acting on the NREL GRC 750 kW wind turbine [159]), and is counteracted by a viscous torque acting on the sun gear representing the influence of the electrical generator. At t = 100 ms, a 50-ms torque ripple is applied to the sun gear, thereby simulating the behavior of the planetary gear set during a symmetrical voltage dip (see [176] for a full-drivetrain simulation of this event using commercial flexible multibody software). Note that in order to guarantee computational tractability of the CMS-based reference model, a very coarse FE mesh had to be used for these simulations, thereby sacrificing somewhat the absolute accuracy of the simulation. The parameters of the various MOR schemes are shown in Table 6.8. A speed-up of roughly 90 times is observed for both the ICS-I and ICS-II approaches with respect to the CMS-based reference model. Note that while the online cost of the ICS-II model is somewhat higher than the cost of the ICS-I model, it makes up for that during the construction of the ROM, which requires less CPU time than in the ICS-I approach.

The DTE of the carrier-sun pair is shown in Fig. 6.13. Consistent with the observations made in Section 4.4.1, both ICS approaches are in very good agreement with the CMS-based ROM, while the modal ROM (MOD) considerably underestimates the DTE due to its inability of resolving local tooth deformations. During the torque ripple (t = 100 – 150 ms) the ICS-II approach is shown to outperform the ICS-I approach, which is due to the increased flexibility of the former in representing tooth contact deformations. The high accuracy of the ICS approaches with respect to the CMS approach is confirmed in Fig. 6.14, where the Von Mises stresses of the sun gear are shown at t = 110 ms. These findings confirm the claimed versatility of the ICS approach in dealing with multi-mesh gear trains.

Parameters	Value
Gear set	Helical, planetary
Number of elements along involute curve	14
Number of elements along tooth width	6
Applied torque planet carrier	360 kNm
Angular velocity planet carrier	22.4 rpm
Steady-state torque sun gear	63 kNm
Angular velocity sun gear	128 rpm
Start voltage dip	0.1 s
End voltage dip	0.15 s
Torque ripple amplitude	50 kNm

Table 6.7 Simulation parameters of the planetary gear set voltage dip simulation

Parameters	CMS	MOD	ICS-I	ICS-II
Number of angular samples	n.a.	n.a.	30	30
Number of torques	n.a.	n.a.	1	1
Number of eigenvectors per gear	50	500	50	50
Number of DOFs in planetary gear model	10839	2505	260	267
Single core pre-processing CPU time [s]	4045	22587	7375	5347
Single core processing CPU time [s]	3788231	946721	42564	43618
Speed-up factor w.r.t. CMS	1	4	89	87

Table 6.8 Parameters and CPU times of the various MOR schemes

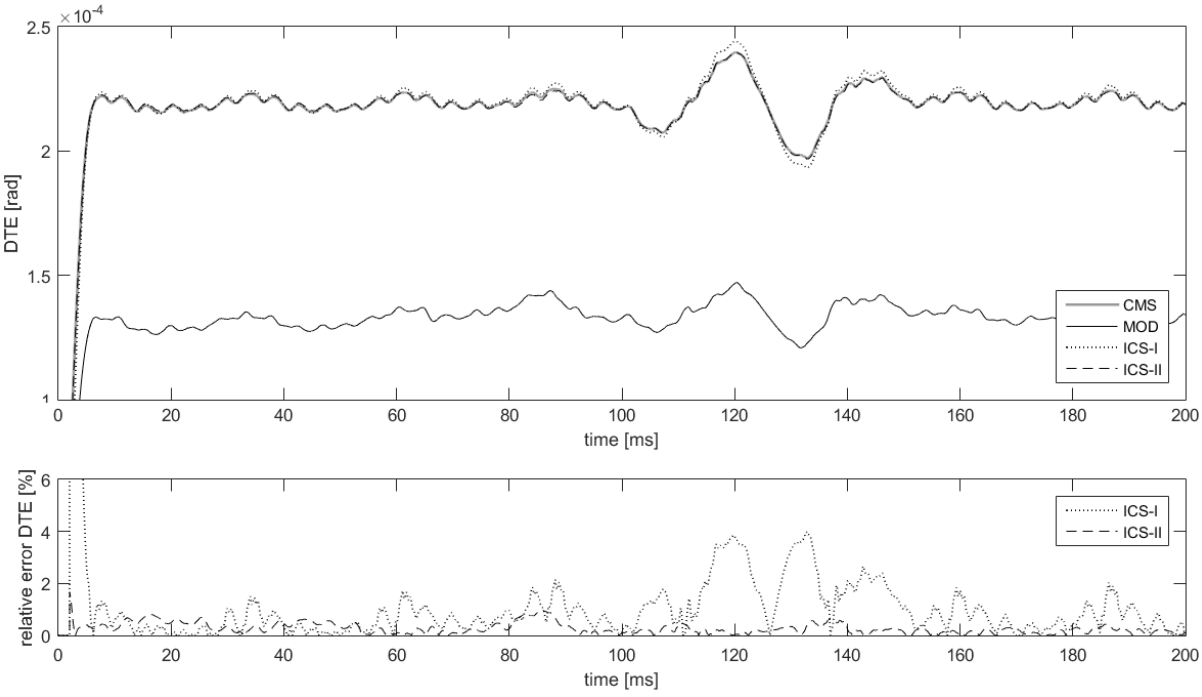


Fig. 6.13 Comparison of DTE predictions for the GRC planetary gear set by the MOR schemes of Table 6.8

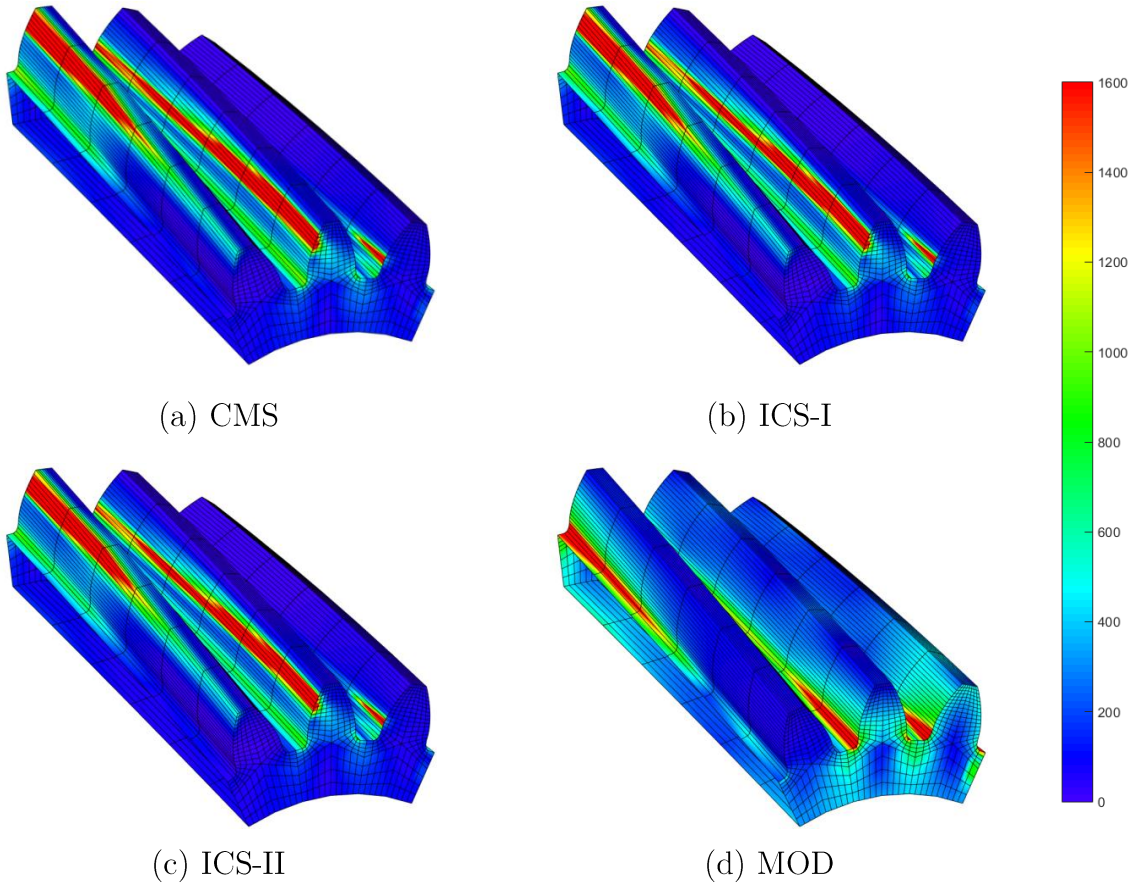


Fig. 6.14 Comparison of the predicted stress distributions in the GRC planetary gear set at t = 110 ms by the MOR schemes of Table 6.8

6.3 A quasi-static gear contact force element for flexible multibody simulations

6.3.1 Positioning of the quasi-static gear contact force element

This chapter on extensions of the ICS approach is concluded with a brief discussion of a novel quasi-static Gear Contact Force Element (GCFE). Advanced GCFEs represent an active area of research and development in most commercial multibody simulation environments (e.g. LMS Virtual.Lab Motion (Siemens PLM Software), Simpack (Dassault Systèmes), Adams (MSC Software), RecurDyn (FunctionBay)). Initially the computational efficiency and robustness of such GCFEs was the main concern, leading to relatively simple lumped parameter models that have proven useful in the early design stage and system-level analysis of mechanical transmissions (see Section 1.2.1 and Appendix N). With increasing computational resources and ever-stricter drivetrain Noise, Vibration and Harshness (NVH) regulations, Original Equipment Manufacturers (OEMs) started to push for more advanced gear modelling capabilities that should ultimately

enable combined component- and system-level optimization. While computational efficiency remains a key concern, accuracy is no longer blindly sacrificed.

Improved accuracy is especially desirable in the representation of the stiffness of the interacting components and in the distribution of the contact loads across the contact surfaces. The most advanced solutions that are currently available in commercial software packages combine FE-based pre-processing features with semi-analytical contact solutions and geometric slicing for improved stiffness and contact force representations (see Appendix N.2). As such, these models are at the border between lumped and distributed parameter approaches. One feature of the original lumped parameter approach that is however preserved in all of the current GCFEs is the computation of contact forces, which for reasons of efficiency is based on analytical gear kinematics and undeformed gear geometry. As a result, off-line of action effects are not taken into account and additional approximations are necessary when gears misalign. Furthermore, tooth coupling effects are generally neglected, and often also the coupling between different contact slices is omitted. Finally, none of the advanced GCFEs available today (including the one presented shortly) take into account the internal dynamics of gears, as this would introduce additional (elastic) DOFs in the multibody system.

While GCFEs based on slicing and semi-analytical contact solutions continue to be valuable in a multitude of practical applications, there can be occasions when accurate contact kinematics are important beyond the approximations of semi-analytical GCFEs (e.g. when the operating torques are high with respect to the load capacity of the gears, or when elastic deformations are large). If in addition the internal dynamic effects are expected to be negligible, the FE-based GCFE presented hereafter offers an effective alternative.

6.3.2 Implementation of the novel gear contact force element

A schematic representation of the novel GCFE is shown in Fig. 6.15. The inputs of the element are the rigid-body coordinates of the pinion and wheel, $\boldsymbol{q}_r^1$ and $\boldsymbol{q}_r^2$. These are composed of three translational coordinates and three or four rotational coordinates per gear, depending on which parameterization is used (see Appendix C). If necessary, the latter are transformed to Bryant angles, which is the parameterization used internally by the GCFE. The outputs of the element are the generalized contact forces acting on the body-attached reference frames of the wheel and pinion, $(\boldsymbol{f}_c^1)_r$ and $(\boldsymbol{f}_c^2)_r$. Behind the scenes, a reduced-order gear contact model based on the ICS approach is used to compute the contact forces. The ROM is trained using the principles outlined in Sections 6.1 and 6.2 (for misaligned gears, the S-MICS approach is used, see Section 6.1.4). No natural vibration modes are included in the reduction basis and the interpolated (misaligned) contact shapes are assumed to respond quasi-statically.

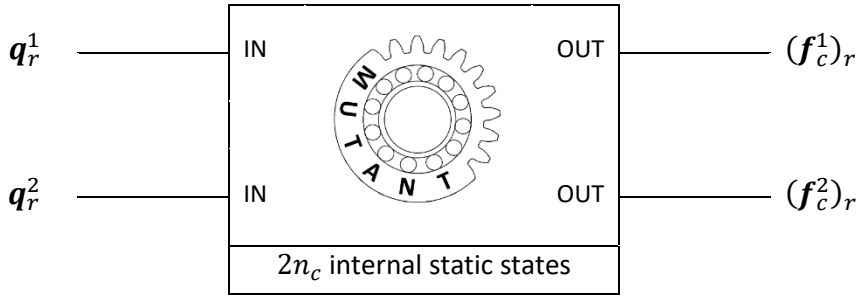


Fig. 6.15 Schematic representation of the novel Gear Contact Force Element

The generalized contact forces are determined by solving a non-linear system of equations expressing static equilibrium between the generalized elastic contact forces and the internal (elastic) forces of the gears, i.e.

$$\boldsymbol{r}(\boldsymbol{q}_c^1, \boldsymbol{q}_c^2) = \begin{bmatrix} \boldsymbol{K}_{cc}^1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{K}_{cc}^2 \end{bmatrix} \begin{Bmatrix} \boldsymbol{q}_c^1 \\ \boldsymbol{q}_c^2 \end{Bmatrix} - \begin{Bmatrix} (\boldsymbol{f}_c^1(\boldsymbol{q}_c^1, \boldsymbol{q}_c^2))_c \\ (\boldsymbol{f}_c^2(\boldsymbol{q}_c^1, \boldsymbol{q}_c^2))_c \end{Bmatrix} = \boldsymbol{0} \tag{6.3.1}$$

where $\boldsymbol{q}_c^i$ is the vector of generalized elastic coordinates of the *i*-th gear corresponding to the set of interpolated contact shapes, and $\boldsymbol{K}_{cc}^i$ and $\left(\boldsymbol{f}_c^i\right)_c$ are the corresponding reduced stiffness matrix and generalized elastic contact forces, respectively. The above system of equations is solved using a Newton-Raphson scheme keeping the rigid body coordinates $\boldsymbol{q}_r^1$ and $\boldsymbol{q}_r^2$ fixed. The generalized contact forces corresponding to these coordinates, $(\boldsymbol{f}_c^1)_r$ and $(\boldsymbol{f}_c^2)_r$, are obtained as a by-product of the solution of Eq. 6.3.1. While convergence is faster when the Newton-Raphson scheme is initiated with the solutions $\boldsymbol{q}_{c,n}^1$ and $\boldsymbol{q}_{c,n}^2$ of the previous time step, zero initial conditions work as well and it is thus not necessary to include a memory-block in the GCFE. Note that the generalized elastic coordinates $\boldsymbol{q}_c^1$ and $\boldsymbol{q}_c^2$ can be used during post-processing to assess the quasi-static contact and tooth bending stresses and displacements.

As the number of internal states in the GCFE is very small ($n_c = 1 \ldots 3$ in typical gear contact simulations), the CPU cost of the element is largely determined by the cost of computing the contact forces and evaluating the contact force Jacobian. This cost can be considerably reduced through the use of the analytical contact Jacobian (see Section 4.2.2) and by applying hyperreduction techniques (see Chapter 8). In semi-analytical GCFEs with coupled slices, instead, the cost is largely determined by the solution of the system of nonlinear equations that describes the compression of the various slices. In the next section, a first comparison of CPU times between the novel GCFE and a semi-analytical GCFE is conducted.

6.3.3 Numerical validation

As a preliminary validation of the proposed GCFE, the spur gear pair of Table 3.1 (see also Section 4.4.1) is analyzed using three gear contact models. The first model, further referred to as 'Mutant – dynamic', is based on the ICS approach using the parameters of Table 4.3. It accounts for internal gear dynamics by including 50 eigenvectors per gear in the reduced-order bases and by dynamically coupling torsional and elastic DOFs. The second model is the GCFE that is obtained from the latter by eliminating all the eigenvectors and retaining only the interpolated contact shapes, which are assumed to respond quasi-statically. The third model, which is derived from the same FE models as the Mutant models, is a semi-analytical gear contact model that is based on the approach of Andersson & Vedmar [19] (see Appendix N.2). The model is representative for the advanced gear solutions that are currently available in commercial multibody software (see Section 6.3.1). The main characteristics of these models are shown in Table 6.9, where N_{slice} refers to the number of slices used in the Andersson-Vedmar approach. Note that the required online CPU time of the Andersson-Vedmar-based GCFE is bracketed as the current implementation was not optimized for efficiency (it uses MATLAB's *fsolve.m* instead of a dedicated solver for obtaining the slice forces).

A torque of 500 Nm is applied to the driving gear and a resisting viscous torque is applied to the driven gear so as to attain an angular velocity of 250 rpm and 1000 rpm, respectively (the CPU times in Table 6.9 correspond to the higher-speed simulation). All models were solved using the optimized Newmark integrator (see Chapter 7) and their DTE predictions are shown in Fig. 6.16 and Fig. 6.17. For low to moderate velocities, the dynamic and quasi-static Mutant models are in good agreement, with relative errors well below 10%. Note that while the model of Andersson-Vedmar yields a reasonable prediction of the average and peak-to-peak value of the DTE, it fails to capture the effect of the applied load on the contact ratio. At elevated velocities, the differences between dynamic and quasi-static Mutant progressively increase, with relative errors reaching up to 20% at 1000 rpm. The frequency of the DTE oscillations, on the other hand, is captured relatively well by all three models (see Fig. 6.17), as these oscillations are primarily due to rotational inertial effects.

Parameters	**Mutant - dynamic**	**Mutant - GCFE**	**Andersson-Vedmar**
Flexibility description	FE	FE	FE + analytical
Contact detection	FE	FE	analytical
# DOFs per gear	52	1	1
# static internal states	0	1	$N_{slice} = 6$
Single core pre-processing CPU time	1423 s	481 s	408 s
Single core processing CPU time	167 s	63 s	(245 s)

Table 6.9 Parameters and CPU times of the three gear contact force elements

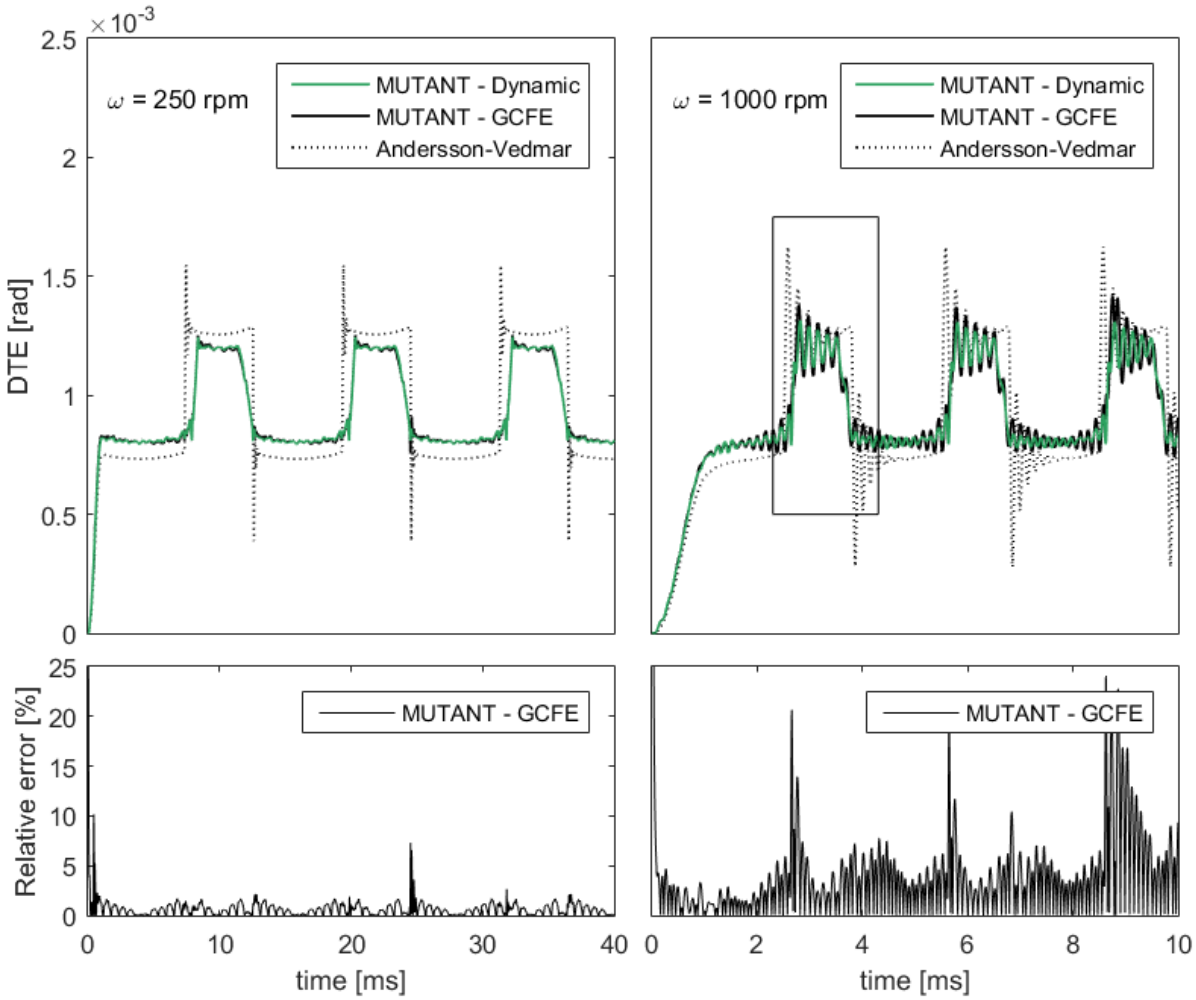


Fig. 6.16 Comparison of DTE predictions by three gear contact force elements

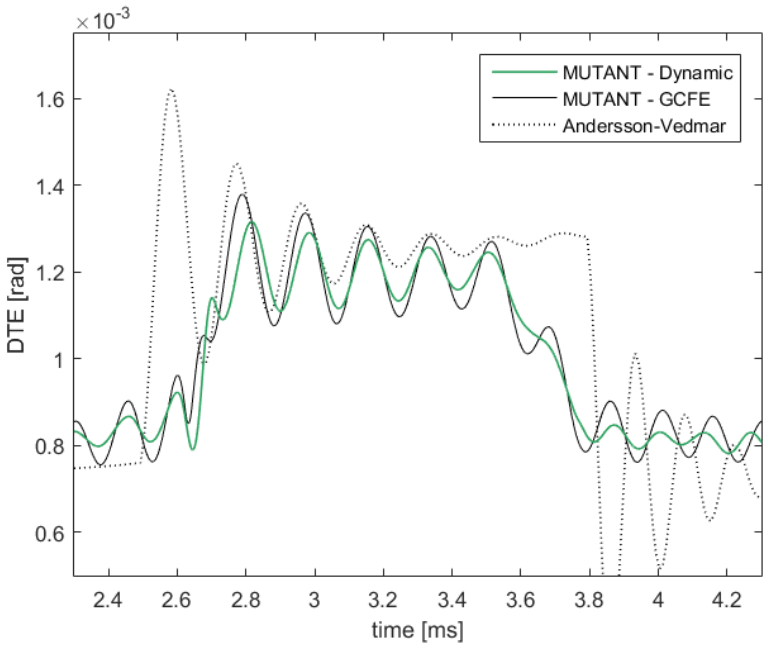


Fig. 6.17 Detail of the DTE prediction corresponding to the rectangle in Fig. 6.16

Summarizing, the ICS-based quasi-static GCFE offers an effective means for including flexible gear models in multibody simulations when the influence of internal gear dynamics is small while at the same time high accuracy is required for the contact detection and stiffness representation. When instead the meshing frequencies of the gears approach the natural frequencies of the gear pair, or when vibrations are transmitted into the gearbox from external sources, the use of dynamic gear contact models should be carefully considered.

6.4 Summary and conclusion

In this chapter, the Interpolated Contact Shapes (ICS) approach that was introduced in Chapter 4 was extended to the analysis of misaligned gears and multi-mesh gear trains, including planetary gear sets. The main challenge in developing a *Misaligned* ICS approach consists of effectively training the Reduced-Order Model (ROM). To this end, a greedy-based sampling algorithm was designed to efficiently sample the high-dimensional misalignment parameter space while at the same time ensuring consistent interpolation of the misaligned contact shapes. Through numerical experiments it was shown that by slightly increasing the number of generalized elastic Degrees-Of-Freedom (DOFs), the misaligned ICS approach yields ROMs with similar accuracy and efficiency as the ROMs presented in Chapter 4. The versatility of the ICS approach was further demonstrated by applying the method to the analysis of planetary gear sets, which are characterized by multiple simultaneous tooth contacts. Two natural extensions of the ICS approach for parallel-axis gear pairs were proposed and were shown to yield similar gains in computational efficiency. Finally, a novel quasi-static gear contact force element for flexible multibody simulations was proposed based on the misaligned ICS approach. Owing to its small number of static internal states and its first-principles-based contact detection, the force element has the potential of earning a rightful place among state-of-the-art commercial gear contact force elements.

Chapter 7

Efficient integration of the reduced-order equations of motion

In the previous three chapters, a novel Model-Order Reduction (MOR) method was presented for reducing the dimensions of Finite-Element (FE) based contact problems in flexible multibody dynamics. By judiciously choosing the ingredients of the reduction basis (see Section 4.1.1), the method was shown to reduce the number of Degrees Of Freedom (DOFs) in the original Full-Order Model (FOM) with several orders of magnitude, from as much as one million to a few tens or hundreds. The influence of lowering the number of DOFs is two-fold: first, it reduces the number of Floating Point Operations (FLOPs) that are required to evaluate the inertial, internal, and quadratic velocity forces, which can be expressed directly in terms of the vectors of reduced coordinates, see Section 4.3.2. Secondly, it increases the efficiency of the numerical integration scheme, either by filtering out high-frequency components from the Equations Of Motion (EOMs), thereby increasing the size of the stable time increment in explicit integrators, or by lowering the cost of the linear solves performed at each Newton iteration in implicit integration schemes. This secondary effect is the main focus of the current chapter.

Thus far in this dissertation, the explicit Central Difference (CD) method was used as the reference integration scheme for solving the EOMs of a pair of engaging flexible gears. It is common practice in FE-based contact problems to choose explicit over implicit schemes due to the lower cost per time increment of the former. The size of the increment is in general not only constrained by the integrator's stability conditions, but also by the accuracy requirements of the non-smooth contact problem at hand. As a results, the ability of implicit integrators to compensate for their expensive Newton iterations by using larger time increments is limited. While it is difficult to theoretically quantify the difference in complexity between explicit and implicit schemes, the latter require roughly $n_q/n_{iter,Jac}$ times more FLOPs than explicit schemes, where n_q is the number of DOFs and $n_{iter,Jac}$ is the average number of iterations before the Jacobian matrix needs to be updated. It follows that implicit integration schemes can only compete with explicit schemes when the accuracy requirements of the problem allow the implicit scheme to use time increments that are roughly $n_q/n_{iter,Jac}$ larger than what is dictated by the stability bounds of the explicit integrator. It is clear that by applying MOR, this threshold $n_q/n_{iter,Jac}$ is reduced and implicit schemes might be brought to the fore. One of the aims of this chapter is to verify whether this is

indeed the case for the novel MOR method, and if so, how such an implicit integrator can be optimized for the dynamic gear contact problem.

Other aspects of numerical time integration that are investigated in the current chapter include the influence of the order of accuracy, the use of adaptive time stepping, and the stability of the integrator. The structure of this chapter is as follows: in Section 7.1, the EOMs that need to be solved are analyzed and the relationship between time increment size and the frequency content of these EOMs is investigated. In Section 7.2, a technique for eliminating the artificial high-frequency components of the EOMs is applied to the dynamic gear contact problem. In Section 7.3, an overview is given of the various time integration schemes that are investigated in search for an optimal time integration scheme for reduced-order contact problems. Finally, in Section 7.4 some numerical experiments are conducted and the optimal time integration scheme is selected.

7.1 Preliminary considerations on time increment selection

7.1.1 Partitioned form of the equations of motion

The EOMs of a flexible gear with fixed center of rotation were derived in Section 2.4.3 (see also Section 3.2.2). In Section 4.1.3, these equations were extended to include the effect of a configuration-dependent basis matrix (see Eq. 4.1.27). Neglecting generalized damping forces, the EOMs have the following general form:

$$\boldsymbol{M}^i\ddot{\boldsymbol{q}}^i + \boldsymbol{K}^i\boldsymbol{q}^i = \boldsymbol{f}_e^i + \boldsymbol{f}_c^i + \boldsymbol{f}_v^i \tag{7.1.1}$$

where $\boldsymbol{M}^i$ is the mass matrix of the i-th gear, $\boldsymbol{K}^i$ is the stiffness matrix, $\boldsymbol{f}_e^i$, $\boldsymbol{f}_c^i$ and $\boldsymbol{f}_v^i$ are the vectors of generalized external, contact and quadratic velocity forces, respectively, and $\boldsymbol{q}^i$ is the vector of generalized coordinates. The latter can be partitioned as:

$$\boldsymbol{q}^i = \begin{Bmatrix} \theta^i \\ \boldsymbol{q}_f^i \end{Bmatrix} = \begin{Bmatrix} \theta^i \\ \boldsymbol{q}_n^i \\ \boldsymbol{q}_c^i \end{Bmatrix} \tag{7.1.2}$$

where $\boldsymbol{q}_n^i$ and $\boldsymbol{q}_c^i$ represent the vectors of generalized elastic coordinates corresponding to the eigenvectors $\boldsymbol{\Phi}^i$ and contact shapes $\boldsymbol{\Psi}_c^i$. Applying this partitioning to Eq. 6.2. and taking into account the orthogonality of the contact shapes $\boldsymbol{\Psi}_c^i$ with respect to the set of kept eigenvectors $\boldsymbol{\Phi}^i$ (see Eq. 4.1.6), the following equations are obtained:

$$\begin{bmatrix} M_{\theta\theta} & \boldsymbol{M}_{\theta n} & \boldsymbol{M}_{\theta c} \\ & \boldsymbol{I}_{nn} & \boldsymbol{0} \\ sym. & & \boldsymbol{M}_{cc} \end{bmatrix}^i \begin{Bmatrix} \ddot{\theta}^i \\ \ddot{\boldsymbol{q}}_n^i \\ \ddot{\boldsymbol{q}}_c^i \end{Bmatrix} + \begin{bmatrix} 0 & \boldsymbol{0} & \boldsymbol{0} \\ & \boldsymbol{\Lambda}_{nn} & \boldsymbol{0} \\ sym. & & \boldsymbol{K}_{cc} \end{bmatrix}^i \begin{Bmatrix} \theta^i \\ \boldsymbol{q}_n^i \\ \boldsymbol{q}_c^i \end{Bmatrix} = \begin{Bmatrix} (\boldsymbol{f}_e^i)_\theta \\ \boldsymbol{0} \\ \boldsymbol{0} \end{Bmatrix} + \begin{Bmatrix} (\boldsymbol{f}_c^i)_\theta \\ (\boldsymbol{f}_c^i)_n \\ (\boldsymbol{f}_c^i)_c \end{Bmatrix} + \begin{Bmatrix} (\boldsymbol{f}_v^i)_\theta \\ (\boldsymbol{f}_v^i)_n \\ (\boldsymbol{f}_v^i)_c \end{Bmatrix} \tag{7.1.3}$$

where $\boldsymbol{I}_{nn}$ is a n_k-dimensional identity matrix and $\boldsymbol{\Lambda}_{nn}$ is a diagonal matrix whose diagonal entries are the squared eigenvalues corresponding to the eigenvectors in $\boldsymbol{\Phi}^i$. Note that when the ICS approach is applied with $n_T = 1$, the submatrices $\boldsymbol{M}_{cc}^i$, $\boldsymbol{K}_{cc}^i$ and $\boldsymbol{M}_{\theta c}^i$ are scalars. When instead the non-interpolated CS approach is used with mass-orthonormalized contact shapes (see Eq. 4.4.3), the submatrix $\boldsymbol{M}_{cc}^i$ is an identity matrix, $\boldsymbol{I}_{cc}$.

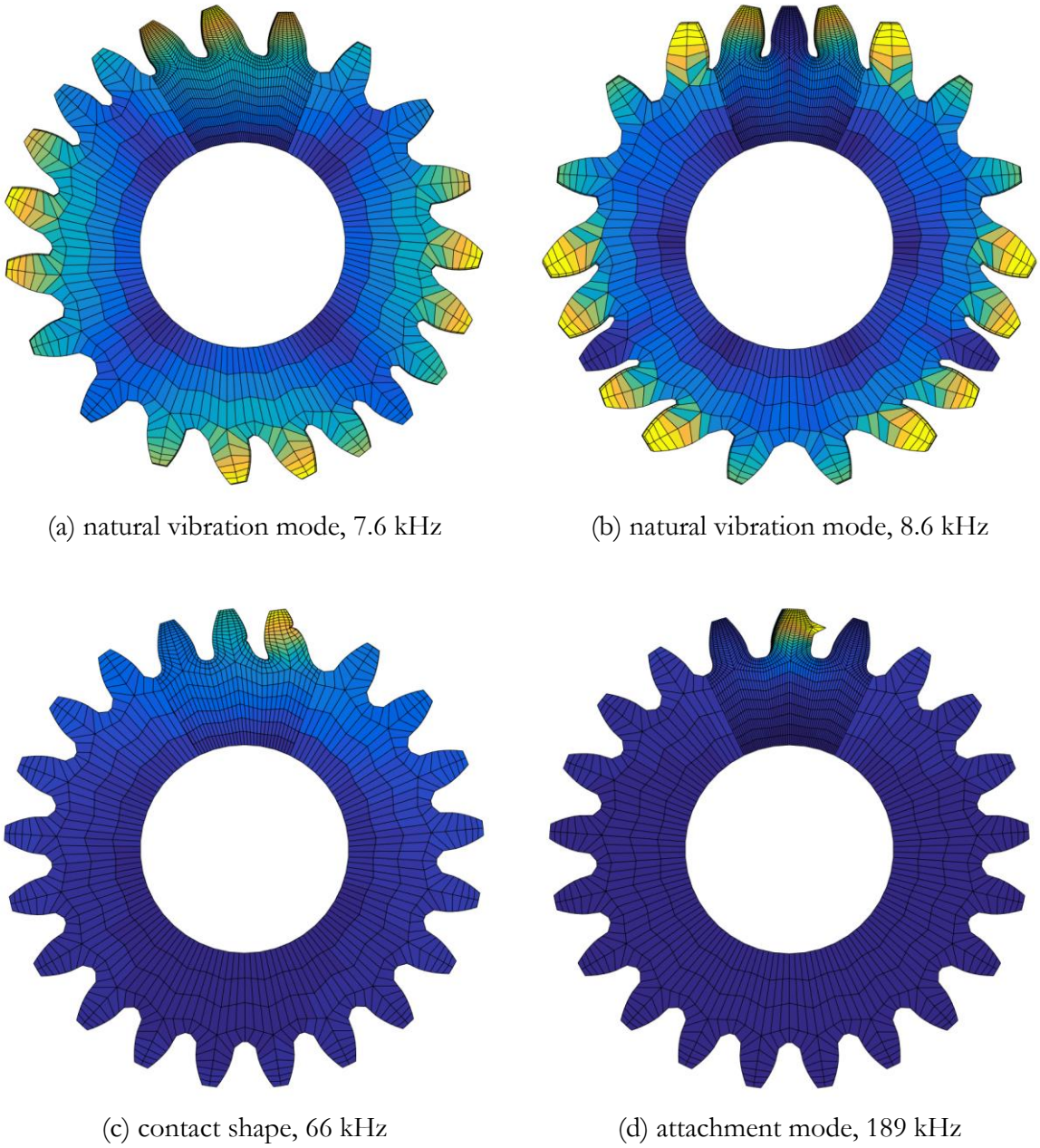
7.1.2 Frequency content of the equations of motion

Eq. 7.1.3 represents a coupled system of nonlinear second-order Ordinary Differential Equations (ODEs) in $\boldsymbol{q}^i$. The numerical solution of such equations is discussed in Sections 2.6.1 and 2.6.2, which respectively deal with explicit and implicit integrators. Regardless of the kind of integrator that is used to solve Eq. 7.1.3, the accuracy and stability of the integrator depend on the frequency content of the problem, or more specifically on the ratio of the selected time increment Δt to the shortest natural period of the problem.

A common way to gain insight into the dynamics and natural periods of the gear contact problem is by linearizing Eq. 7.1.3 about a static equilibrium solution and solving the corresponding generalized eigenvalue problem to find the gear pair's eigenvalues and eigenvectors. The latter consist of a combination of the gear pair's torsional vibration modes and the elastic vibration modes of the individual gears. The frequency of the former is largely determined by the ratio of the linearized gear mesh stiffness to the gear pair's mass moment of inertia, while the latter depend on the (pseudo-)eigenfrequencies of the component modes that are used to describe the elastic behavior of the individual gears. Global deformation modes – such as the set of kept natural vibration modes $\boldsymbol{\Phi}^i$ – correspond to relatively low frequencies, whereas local deformation modes – such as the attachment modes $\boldsymbol{\Psi}_a^i$ or the contact shapes $\boldsymbol{\Psi}_c^i$ – correspond to comparatively high frequencies, see Fig. 7.. The size of the increment Δt is thus to a large extend determined by the shape vectors $\boldsymbol{\Psi}_a^i$ or $\boldsymbol{\Psi}_c^i$.

Very often the dynamic responses of high-frequency modes in a numerical model are physically meaningless: they are a result of the spatial discretization of the problem and/or of the artificial separation of global and local deformation modes. A common way to suppress instabilities caused by these high-frequency modes is to dampen their response either by including frequency-proportional damping into the EOMs, or by using a numerical integration scheme that artificially adds numerical damping, such as the generalized-α solver (see Appendix J)[27]. Another approach eliminates the high-frequency components altogether from the EOMs, assuming these components only contribute to the solution quasi-statically. The latter approach is investigated and applied to the dynamic gear contact problem in Section 7.2.

[27] In recent work by Dewalque et al. [220] it is argued that in fact both are needed to correctly stabilize the high-frequency content of the solution.



(a) natural vibration mode, 7.6 kHz

(b) natural vibration mode, 8.6 kHz

(c) contact shape, 66 kHz

(d) attachment mode, 189 kHz

Fig. 7.1 Comparison of the (pseudo-)eigenfrequencies of the component modes of a spur gear

7.1.3 Accuracy and stability in dynamic contact problems

The primary concern when selecting a time increment Δt is that it yields a numerically stable time integration scheme. Following the definition of Géradin [177], a numerical integration scheme is stable if there exists an increment $h_{stable} > 0$ such that for any $h \in [0, h_{stable}]$ a finite variation of the coordinate vector $\boldsymbol{q}^i(t_n) = \boldsymbol{q}_n^i$ at time t_n induces only a non-increasing variation of $\boldsymbol{q}^i(t_{n+1}) = \boldsymbol{q}_{n+1}^i$ calculated at the next time step $t_{n+1} = t_n + h$. The accuracy of the solution is of secondary

concern and relates to the desired temporal resolution of the solution, $h_{accurate}$. Obviously, the accuracy of the solution can only be controlled within the interval $[0, h_{stable}]$ and the increment size is ultimately given by $h = \min(h_{stable}, h_{accurate})$.

Explicit integration schemes are known to be conditionally stable, whereas certain implicit integrators, such as the Newmark-β and generalized-α methods, are said to be *unconditionally* stable. It is a common misconception that the stability of these methods is *in general* not affected by the size of the time increment, and that the latter can be selected merely based on accuracy considerations. The unconditional stability statement is only generally valid for linear problems. In the nonlinear case, no general statements about the stability of implicit integrators can be made (unless the schemes are energy preserving or decaying, see [178]). Linearized stability analysis may be used to derive the necessary conditions for stability, but in implicit schemes these may not be sufficient [179] (in explicit schemes on the other hand they generally are, which is why the formula for the stable time increment of the CDM (see Eq. 4.4.1) works so well). Amongst others, the stability of implicit integration schemes is limited by the relatively small convergence radius of the Newton-Raphson scheme that is used to iteratively solve the nonlinear dynamic equilibrium equations at each time step. Therefore, selecting an adequate time increment h is as critical in implicit as in explicit analyses.

In nonlinear analysis, and in dynamic contact analyses in particular, selecting an a priori fixed time increment size h will lead to suboptimal integration of the EOMs. If $h \in [0, h_{stable}]$ is too large, some higher-frequency contributions to the solution might be filtered out thus missing important solution details. If instead h is unnecessary small, precious CPU resources will be wasted. Fixed increment sizes are especially troublesome in the presence of discontinuities in the system's response, such as at the onset of tooth engagement, as they require local adaptation of the increment size in order to maintain constant accuracy throughout the solution process. For these reasons, an adaptive time stepping algorithm is applied for optimally integrating the EOMs. Increment size adjustments are generally based on some measure of the local integration error that is induced by the integration scheme. With the exception of the CDM, where increment size control tends to be based on stability requirements, most adaptive integrators thus control accuracy, not stability. Hence, if the user-defined error tolerance is such that $h_{accurate} < h_{stable}$, the integration process will fail.

7.2 Elimination of the high-frequency solution components

7.2.1 Dynamic decoupling of the contact shapes

In Section 7.1.2, it was argued that the dynamic response of high-frequency modes in projection-based ROMs is often unphysical and might lead to spurious oscillations. One way to deal with these numerical artefacts and thereby suppress the associated instabilities is to eliminate the high-frequency components from the EOMs. Two conditions need to be fulfilled for this approach to be applicable. First, the equations corresponding to the high-frequency modes – in casu: the

contact shapes $\boldsymbol{\Psi}_c^i$ – need to be dynamically decoupled from the other equations. Second, the inertia effects associated with these modes should be negligable.

In Eq. 7.1.3, the generalized elastic coordinates $\boldsymbol{q}_c^i$ are dynamically coupled with the rigid body rotation θ^i through the inertial forces $\boldsymbol{M}_{\theta c}^i \ddot{\boldsymbol{q}}_c^i$. Using Appendix D.2 and Eq. 7.1.3, the inertial forces associated with the rigid-body rotation of the gear can be expressed in terms of the mass invariants as follows (neglecting the additional terms in the ICS approach):

$$\begin{aligned}\left(\boldsymbol{f}_l^i\right)_\theta = & \underbrace{\boldsymbol{M}_2^i \ddot{\theta}^i + 2\ddot{\theta}^i \boldsymbol{M}_{3n}^i \boldsymbol{q}_n^i + 2\ddot{\theta}^i \boldsymbol{M}_{3c}^i \boldsymbol{q}_c^i + \ddot{\theta}^i {\boldsymbol{q}_n^i}^T \boldsymbol{M}_{6n}^i \boldsymbol{q}_n^i + \ddot{\theta}^i {\boldsymbol{q}_c^i}^T \boldsymbol{M}_{6c}^i \boldsymbol{q}_c^i}_{M_{\theta\theta}^i \ddot{\theta}^i} \\ & + \underbrace{\boldsymbol{M}_{4n}^i \ddot{\boldsymbol{q}}_n^i + {\boldsymbol{q}_n^i}^T \boldsymbol{M}_{7nn}^i \ddot{\boldsymbol{q}}_n^i + {\boldsymbol{q}_c^i}^T \boldsymbol{M}_{7sn}^i \ddot{\boldsymbol{q}}_n^i}_{\boldsymbol{M}_{\theta n}^i \ddot{\boldsymbol{q}}_n^i} + \underbrace{\boldsymbol{M}_{4c}^i \ddot{\boldsymbol{q}}_c^i + {\boldsymbol{q}_n^i}^T \boldsymbol{M}_{7nc}^i \ddot{\boldsymbol{q}}_c^i + {\boldsymbol{q}_c^i}^T \boldsymbol{M}_{7cc}^i \ddot{\boldsymbol{q}}_c^i}_{\boldsymbol{M}_{\theta c}^i \ddot{\boldsymbol{q}}_c^i}\end{aligned} \tag{7.2.1}$$

where $\boldsymbol{M}_{kl}^i, k = 3,4,6,7$ and $l = n, c, nc, cn, nn, cc$ are the gear's mass invariants as defined in Appendix D.2. For the dynamic spur gear simulation that was discussed in Section 4.4.1 (see Table 4.3), the relative contribution of the eleven terms in Eq. 7.2.1 is investigated in Fig. 7.2. This figure shows that the inertial forces $\boldsymbol{M}_{\theta c}^i \ddot{\boldsymbol{q}}_c^i$ are negligible in comparison with the inertial forces $\boldsymbol{M}_{\theta n}^i \ddot{\boldsymbol{q}}_n^i$ that are associated with the lower-frequency natural vibration modes $\boldsymbol{\Phi}^i$.

Following a similar reasoning as outlined in [180], this observation can be supported by theoretical considerations based on the separation between local and global deformations. Assuming the accelerations $\ddot{\boldsymbol{q}}_n^i$ and $\ddot{\boldsymbol{q}}_c^i$ are of similar magnitude, the aforementioned proposition is equivalent to

$$\left\|\boldsymbol{M}_{\theta n}^i\right\| \gg \left\|\boldsymbol{M}_{\theta c}^i\right\| \tag{7.2.2}$$

where $\|\cdot\|$ is a matrix norm that is obtained by summing the absolute values of all the entries of the designated matrix. Eq. 4.1.34 gives an expression for the matrices in the above equation. Both equations involve the same vector quantities multiplied with either $\boldsymbol{\Phi}^i$ or $\boldsymbol{\Psi}_c^i$. It is to be expected that the summation along all the nodes $j = 1..n_n$ of a vector multiplied with *global* deformation vectors (in casu: $\boldsymbol{\Phi}^i$) is considerably bigger than the nodal summation of the product involving *local* deformation vectors (in casu: $\boldsymbol{\Psi}_c^i$). This justifies the assumption that $\left\|\boldsymbol{M}_{\theta c}^i\right\|$ is negligible in comparison with $\left\|\boldsymbol{M}_{\theta n}^i\right\|$ (Eq. 7.2.2) and allows Eq. 4.1.34 to be approximated by

$$\boldsymbol{M}_{\theta n}^i \approx \sum_{j=1}^{n_n} m^{ij} \left(\bar{\boldsymbol{r}}_0^{ij^T} + {\boldsymbol{q}_n^i}^T \bar{\boldsymbol{\Phi}}^{ij^T}\right) \bar{\boldsymbol{\Phi}}_{y-x}^{ij} \tag{7.2.3a}$$

$$\boldsymbol{M}_{\theta c}^i \approx \boldsymbol{0} \tag{7.2.3b}$$

Note that while the inertial forces $\boldsymbol{M}_{\theta c}^i \ddot{\boldsymbol{q}}_c^i$ can be neglected in the first line of Eq. 7.1.3, the gear's mass moment of inertia $\boldsymbol{M}_{\theta\theta}^i$ still depends on $\boldsymbol{q}_c^i$. Complete decoupling of $\boldsymbol{q}_c^i$ and θ^i in the mass

matrix can be obtained by assuming that the influence of local deformations (represented by $\boldsymbol{\Psi}_c^i$) on $\boldsymbol{M}_{\theta\theta}^i$ is negligible in comparison with the influence of global deformations. In doing so, the expression for the mass moment of inertia (see Eq. 4.1.28) reduces to:

$$\boldsymbol{M}_{\theta\theta}^i \approx \sum_{j=1}^{n_n} m^{ij} \left[\bar{\boldsymbol{r}}_0^{ij^T} \bar{\boldsymbol{r}}_{0,xy}^{ij} + 2\bar{\boldsymbol{r}}_0^{ij^T} \bar{\boldsymbol{\Phi}}_{xy}^{ij} \boldsymbol{q}_n^i \right] \tag{7.2.4}$$

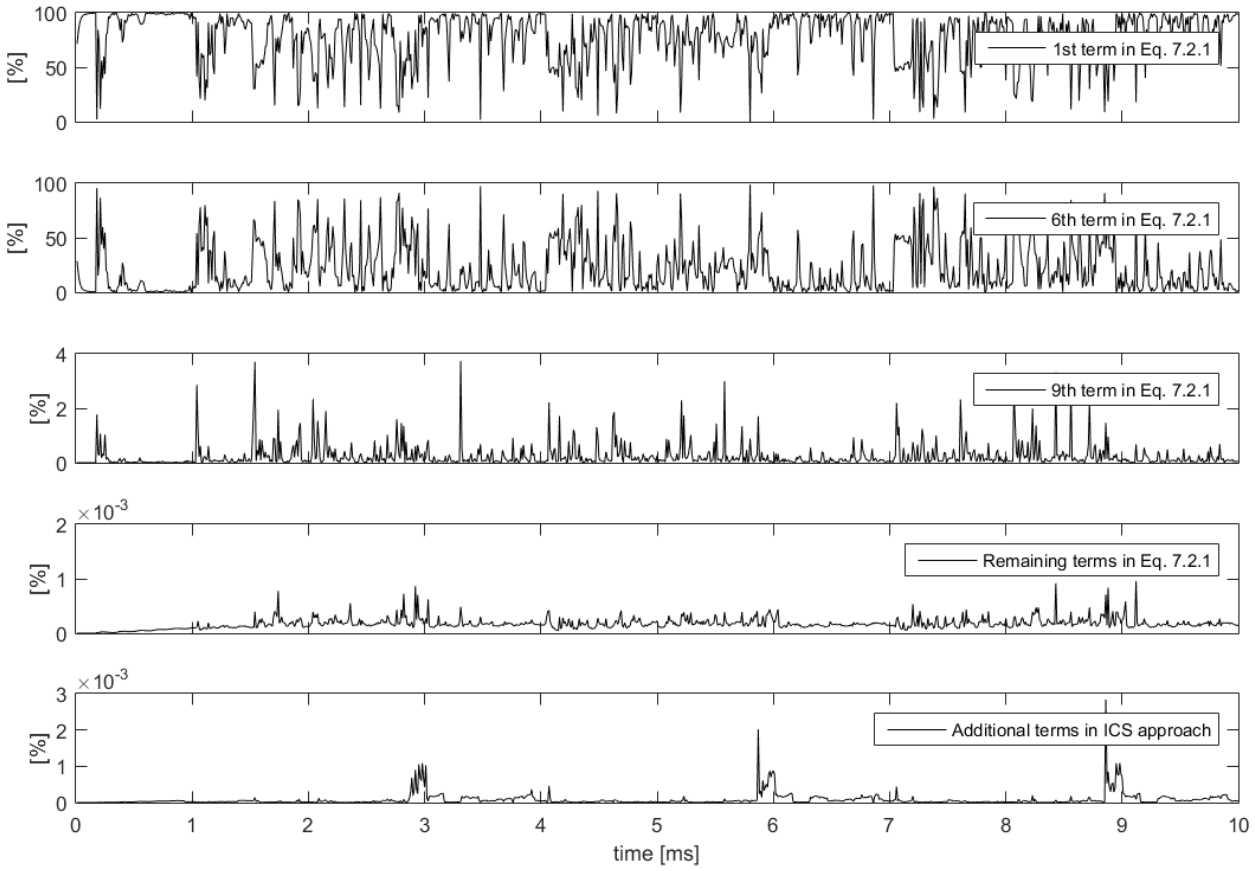


Fig. 7.2 Relative contribution of the terms in Eq. 7.2.1 to the inertial forces associated with the gear's rigid-body rotation

Following a similar reasoning for the quadratic velocity vector (see Eq. 4.1.25), the components of this vector simplify to

$$\left(\boldsymbol{f}_v^i\right)_\theta \approx \sum_{j=1}^{n_n} m^{ij} \left[-\dot{\boldsymbol{q}}_n^{i\,T} \bar{\boldsymbol{\Phi}}^{ij^T} \left(2\dot{\theta}^i \bar{\boldsymbol{r}}_{0,xy}^{ij} + 2\dot{\theta}^i \bar{\boldsymbol{\Phi}}_{xy}^{ij} \boldsymbol{q}_n^i + \bar{\boldsymbol{\Phi}}_{y-x}^{ij} \dot{\boldsymbol{q}}_n^i \right) \right] \tag{7.2.5a}$$

$$\left(\boldsymbol{f}_v^i\right)_n \approx \sum_{j=1}^{n_n} m^{ij} \left[\dot{\theta}^i \bar{\boldsymbol{\Phi}}^{ij^T} \left(\dot{\theta}^i \bar{\boldsymbol{r}}_{0,xy}^{ij} + \dot{\theta}^i \bar{\boldsymbol{\Phi}}_{xy}^{ij} \boldsymbol{q}_n^i + 2\bar{\boldsymbol{\Phi}}_{y-x}^{ij} \dot{\boldsymbol{q}}_n^i \right) \right] \tag{7.2.5b}$$

$$\left(\boldsymbol{f}_v^i\right)_c \approx \boldsymbol{0} \tag{7.2.5c}$$

Summarizing, using the approximations introduced in Eq. 7.2.3-7.2.5, the EOMs of the *i*-th flexible gear (Eq. 7.1.3) simplify to

$$\begin{bmatrix} M_{\theta\theta} & \boldsymbol{M}_{\theta n} & \boldsymbol{0} \\ & \boldsymbol{I}_{nn} & \boldsymbol{0} \\ sym. & & \boldsymbol{M}_{cc} \end{bmatrix}^i \begin{Bmatrix} \ddot{\theta}^i \\ \ddot{\boldsymbol{q}}_n^i \\ \ddot{\boldsymbol{q}}_c^i \end{Bmatrix} + \begin{bmatrix} 0 & \boldsymbol{0} & \boldsymbol{0} \\ & \boldsymbol{\Lambda}_{nn} & \boldsymbol{0} \\ sym. & & \boldsymbol{K}_{cc} \end{bmatrix}^i \begin{Bmatrix} \theta^i \\ \boldsymbol{q}_n^i \\ \boldsymbol{q}_c^i \end{Bmatrix} = \begin{Bmatrix} (\boldsymbol{f}_e^i)_\theta \\ \boldsymbol{0} \\ \boldsymbol{0} \end{Bmatrix} + \begin{Bmatrix} (\boldsymbol{f}_c^i)_\theta \\ (\boldsymbol{f}_c^i)_n \\ (\boldsymbol{f}_c^i)_c \end{Bmatrix} + \begin{Bmatrix} (\boldsymbol{f}_v^i)_\theta \\ (\boldsymbol{f}_v^i)_n \\ \boldsymbol{0} \end{Bmatrix} \tag{7.2.6}$$

Note that the equations for θ^i and $\boldsymbol{q}_n^i$ are dynamically decoupled from $\boldsymbol{q}_c^i$; however, the equations remain statically coupled through the vector of generalized contact forces $\boldsymbol{f}_c^i$, which depends on the full vector of generalized coordinates $\boldsymbol{q}^i$.

7.2.2 Quasi-static consideration of the contact shapes

The second condition that needs to be fulfilled for the quasi-static approach to be applicable, is that the inertia effects associated with the contact shapes are negligible. As the pseudo-eigenfrequencies associated with these shapes are typically significantly higher than the largest eigenvalue considered in $\boldsymbol{\Phi}^i$, they will not be dynamically excited in the frequency range of interest; hence they will respond quasi-statically and the associated inertia effects can be neglected. This observation allows the last line of Eq. 7.2.6 to be solved for the generalized elastic coordinates $\boldsymbol{q}_c^i$:

$$\boldsymbol{q}_c^i = {\boldsymbol{K}_{cc}^i}^{-1} \left(\boldsymbol{f}_c^i\right)_c \tag{7.2.7}$$

given that the reduced-order stiffness matrix $\boldsymbol{K}_{cc}^i$ is invertible. Using this equation, the simplified equations of motion (Eq. 7.2.6) reduce to

$$\begin{bmatrix} M_{\theta\theta} & \boldsymbol{M}_{\theta n} \\ sym. & \boldsymbol{I}_{nn} \end{bmatrix}^i \begin{Bmatrix} \ddot{\theta}^i \\ \ddot{\boldsymbol{q}}_n^i \end{Bmatrix} + \begin{bmatrix} 0 & \boldsymbol{0} \\ sym. & \boldsymbol{\Lambda}_{nn} \end{bmatrix}^i \begin{Bmatrix} \theta^i \\ \boldsymbol{q}_n^i \end{Bmatrix} = \begin{Bmatrix} \left(\boldsymbol{f}_e^i\right)_\theta \\ \boldsymbol{0} \end{Bmatrix} + \begin{Bmatrix} \left(\boldsymbol{f}_c^i\right)_\theta \\ \left(\boldsymbol{f}_c^i\right)_n \end{Bmatrix} + \begin{Bmatrix} \left(\boldsymbol{f}_v^i\right)_\theta \\ \left(\boldsymbol{f}_v^i\right)_n \end{Bmatrix} \tag{7.2.8}$$

and are subjected to Eq. 7.2.7. By eliminating $\boldsymbol{q}_c^i$ from the gear pair's EOMs, the frequency content of the dynamic gear contact problem is reduced and a larger time increment Δt can be selected for solving Eq. 7.2.8. The price to be paid, however, is that the response of the gear pair is now described by a system of Differential-Algebraic Equations (DAEs) and thus involves the solution of a set of nonlinear algebraic equations in each step of the time integration. The following sections deal with the development of optimal integration schemes for solving the ODEs of Eq. 7.1.3 and the DAEs of Eq. 7.2.7-7.2.8.

7.3 Numerical integration schemes for the dynamic gear contact problem

Having discussed the EOMs that need to be solved and the factors that influence the selection of an adequate time increment Δt, the aim of the next two sections is to identify the optimal integration scheme for the dynamic gear contact problem, taking into account the distinctive features of the novel MOR method. Four types of integration schemes are taken into consideration: explicit and implicit schemes for solving the ODEs of Eq. 6.2., and semi-explicit and fully-implicit schemes for solving the DAEs of Eq. 7.2.7-7.2.8. In the current section an overview is given of the candidate integrators, their main characteristics and some of their algorithmic particularities. In Section 7.4 these integrators are put to the test and the optimal integration scheme is selected.

7.3.1 Explicit integration of the equations of motion

In Section 2.6.1, three explicit integrators for solving Eq. 6.2. are discussed: the Runge-Kutta (RK) method (Section 2.6.1.1), the Adams PECE (APECE) method (Section 2.6.1.2) and the Central Difference (CD) method (Section 2.6.1.3). The RK and APECE methods are general-purpose explicit integrators for non-stiff ODEs in first-order from (see Eq. 2.6.5). The CD method is instead specifically designed for solving problems in structural dynamics and operates directly on the second-order form of Eq. 6.2.. Assuming the mass matrix $\boldsymbol{M}^i$ is invertible (as is always the case for Eq. 6.2.) and introducing the variable $\boldsymbol{v}^i = \dot{\boldsymbol{q}}^i$, Eq. 6.2. can be transformed to first-order form as follows:

$$\begin{Bmatrix} \dot{\boldsymbol{q}}^i \\ \dot{\boldsymbol{v}}^i \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{v}^i \\ \left[\boldsymbol{M}^i(\boldsymbol{q}^i)\right]^{-1} \boldsymbol{f}^i(t, \boldsymbol{q}^i, \boldsymbol{v}^i) \end{Bmatrix} = f(t, \boldsymbol{q}^i, \boldsymbol{v}^i) \tag{7.3.1}$$

where the generalized force vector $\boldsymbol{f}^i$ is defined as

$$\boldsymbol{f}^i(t, \boldsymbol{q}^i, \boldsymbol{v}^i) \equiv \boldsymbol{f}_e^i(t) + \boldsymbol{f}_c^i(\boldsymbol{q}^i) + \boldsymbol{f}_v^i(\boldsymbol{q}^i, \boldsymbol{v}^i) - \boldsymbol{K}^i \boldsymbol{q}^i - \boldsymbol{D}^i \boldsymbol{v}^i \tag{7.3.2}$$

and where $f(t, \boldsymbol{q}^i, \boldsymbol{v}^i)$ is referred to as the Right-Hand Side (RHS) of Eq. 7.3.1. Note that in comparison with Eq. 6.2., the generalized viscous damping forces (see Appendix E) are included in the problem description.

In computational contact mechanics, it is desirable to have an integration scheme with an order of convergence of at least two (see the introduction of Section 2.6). In single-step methods such as the RK method, the order of convergence scales with the number of RHS evaluations carried out per time step, whereas for multistep methods such as the APECE method, the order is generally dependent on the number of past solution or RHS values that are used to advance the

solution in time. The CD method can be seen as a two-step method with a fixed order of convergence of two (or approximately two, as in Eq. 2.6.23). Note that since for stiff problems the increment size of explicit integrators is determined by stability rather than accuracy requirements, there is no computational advantage to be expected from increasing the order of convergence past second order, unless this simultaneously increases the stability region of the integrator. The latter is the case for the RK method, but not for Adams methods, whose regions of stability decrease as the order of the method increases.

Another desirable feature of the envisioned integration scheme is the availability of a computable estimate of the local truncation error, such that it can be used for adaptive time stepping. While the latter will generally result in a higher cost per time increment, it has the benefit of using larger time increments when the solution behaves smoothly and smaller time increments when discontinuities occur or high-frequency modes are excited, which can lead to considerable overall time savings. While error estimation is generally more costly in single-step methods, they offer greater flexibility in changing the increment size than multi-step methods. The latter come with cheap error estimators yet require either variable-coefficient update formulas or interpolation strategies for adjusting the increment size.

Taking into account the above considerations, the following five explicit integration schemes are investigated in the current work:

- the $1^{st}(2^{nd})$ order Fehlberg RK method (RK-F, see Appendix G.1);
- the $2^{nd}(3^{rd})$ order Bogacki-Shampine RK method (RK-BS, see Appendix G.1);
- the 2^{nd} order Adams PECE method (APECE-2, see Appendix G.2);
- the 4^{th} order Adams PECE method (APECE-4, see Appendix G.2);
- the 2^{nd} order CD method (see Section 2.6.1.3).

A comparison of these methods in terms of order of convergence, number of function evaluations per update, number of steps used in the update formula, error estimation and stability is given in Table 7.1. While the Fehlberg RK method progresses the solution using a second-order accurate approximation, the increment size is based on an error estimate for the embedded first-order method, which is why depending on the source the method is classified as either a first- or second-order method. The influence of the order of convergence is assessed by comparing Fehlberg's method with the higher-order method of Bogacki-Shampine. A similar exercise is conducted by comparing the 2^{nd} and 4^{th} order APECE methods: contrary to the pair of RK methods, however, the higher order of convergence doesn't come at the price of additional function evaluations, yet results in reduced stability properties. All integrators have narrow regions of absolute stability which cannot be quantified exactly unless with reference to a selected linear test equation; therefore, only qualitative assessments are made in Table 7.1. Note that the order of convergence of the CD method is indicated as ~2, which is due to the fact that the integration scheme with velocities lagging half a time increment is used (see Eq. 2.6.23).

The numerical implementation of the above integration schemes is discussed in Section 2.6.1. Some additional remarks are in place here. In both of the embedded RK schemes, the higher-order approximation is used to progress the solution and the FSAL property is used to minimize the number of RHS evaluations per time step. In the Adams PECE methods, the same order of convergence is used for both the Adams-Bashforth predictor and the Adams-Moulton corrector, so as to guarantee the order is conserved in the PECE scheme. For this reason, the APECE-2 integrator that combines a first-order Adams-Bashforth predictor with a second-order Adams-Moulton corrector, thus yielding a larger stability region [181], was discarded. In the CD scheme, Eq. 2.6.23 is used to advance the solution with generalized velocities lagging half a time increment, and the local truncation error is estimated based on the Taylor expansion of the exact solution (see Eq. 2.6.26). In all multistep schemes (including the CD scheme), the time increment size is adjusted using a variable-coefficient strategy.

Integrator	Order	# RHS evaluations per step	# steps in update formula	Error estimation based on…	Stability
RK-F	1(2)	2	single-step (1)	higher-order approximation	< RK-BS ~ APECE-2
RK-BS	2(3)	3	single-step (1)	higher-order approximation	> RK-F > APECE
APECE-2	2	2	multistep (2)	difference corrector and predictor	> APECE-4 ~ RK-F
APECE-4	4	2	multistep (4)	difference corrector and predictor	< APECE-2 < RK
CD	~2	1	multistep (2)	jerk approximation	< RK < APECE

Table 7.1 Comparison of the selected explicit integration schemes

7.3.2 Semi-explicit integration of the equations of motion

Instead of solving Eq. 6.2. in terms of the full vector of generalized coordinates $\boldsymbol{q}^i$, it was shown in Section 7.2 that, under the appropriate assumptions, the generalized elastic coordinates $\boldsymbol{q}_c^i$ can be eliminated from the gear's EOMs, thereby reducing the dimensions of the associated system matrices and lowering the frequency content of the problem (see Eq. 7.2.7-7.2.8). The price to pay, however, is that the equations describing the dynamics of the flexible gear pair are described in

terms of a system of DAEs. Introducing the variables $\boldsymbol{q}_d^i = \left\{\theta^{i^T} \quad \boldsymbol{q}_n^{i\,T}\right\}^T$ and $\boldsymbol{v} = \dot{\boldsymbol{q}}_d^i$, this system can be written in first-order form as

$$\dot{\boldsymbol{q}}_d^i = \boldsymbol{v}^i \tag{7.3.3a}$$

$$\dot{\boldsymbol{v}}^i = \left[\boldsymbol{M}^i\left(\boldsymbol{q}_d^i\right)\right]^{-1}\boldsymbol{f}^i\left(t, \boldsymbol{q}_d^i, \boldsymbol{q}_c^i, \boldsymbol{v}^i\right) = f\left(t, \boldsymbol{q}_d^i, \boldsymbol{q}_c^i, \boldsymbol{v}^i\right) \tag{7.3.3b}$$

$$\boldsymbol{0} = \boldsymbol{q}_c^i - \boldsymbol{K}_{cc}^{i\,-1}\left(\boldsymbol{f}_c^i(\boldsymbol{q}_d^i, \boldsymbol{q}_c^i)\right)_c = g\left(\boldsymbol{q}_d^i, \boldsymbol{q}_c^i\right) \tag{7.3.3c}$$

where f and g represent the RHS and constraint functions, respectively.

The above system of DAEs is of a particular form (index 0), which can be solved relatively straightforwardly using explicit integration schemes. As an example, consider the forward Euler scheme applied to Eq. 7.3.3, where the constraint equation (Eq. 7.3.3c) is satisfied at the next time step, *n+1*:

$$\boldsymbol{q}_{d,n+1}^i = \boldsymbol{q}_{d,n}^i + h\boldsymbol{v}_n^i \tag{7.3.4a}$$

$$\boldsymbol{v}_{n+1}^i = \boldsymbol{v}_n^i + hf\left(t_n, \boldsymbol{q}_{d,n}^i, \boldsymbol{q}_{c,n}^i, \boldsymbol{v}_n^i\right) \tag{7.3.4b}$$

$$\boldsymbol{0} = g\left(\boldsymbol{q}_{d,n+1}^i, \boldsymbol{q}_{c,n+1}^i\right) \tag{7.3.4c}$$

Eq. 7.3.4a-b are explicit equations that can readily be solved for $\boldsymbol{q}_{d,n+1}^i$ and $\boldsymbol{v}_{n+1}^i$, whereas Eq. 7.3.4c is an implicit equation in terms of $\boldsymbol{q}_{c,n+1}^i$ that can be solved using a Newton-type iterative method. When the Contact Shapes (CS) approach (Eq. 4.1.5) is used, this is a system of n_c ($\approx$ 20..50) simultaneous equations, whereas in the Interpolated Contact Shapes (ICS) approach (Eq. 4.1.8), only n_T ($\approx$ 1..3) equations need to be solved. A similar strategy as outlined for the forward Euler method can be applied to each of the explicit integration schemes in Table 7.1. Note that since Eq. 7.3.4c (or, equivalently, Eq. 7.3.3c) is evaluated at the next time step, *n+1*, it is no longer necessary to compute the vector of generalized contact forces when evaluating f at the next time step.

7.3.3 Implicit integration of the equations of motion

Despite their superior stability properties, implicit integration schemes are rarely used to solve dynamic FE-based contact problems. In most real-world applications, the benefits of using larger time increments are nullified by the high cost of computing the system Jacobian and solving the associated system of nonlinear equations. However, when the number of DOFs and/or the computational complexity of the problem are significantly reduced, implicit integration schemes come to the fore.

In Section 2.6.2, two classes of implicit integrators are discussed for solving Eq. 6.2.: the Backward Differentiation Formula (BDF, Section 2.6.2.1) and the family of Newmark integrators

(Section 2.6.2.2), including the Newmark-β method (NMK- β, where the parameters $\beta = 0.25$ and $\gamma = 0.5$ are selected) and the Generalized-α method (GEN- α). As with explicit integration schemes, an order of convergence of at least two and a reliable error estimator are mandatory for solving problems in structural dynamics. The order of convergence of BDF methods is equal to the number of steps, whereas the Newmark family of integrators has a fixed order of convergence of exactly two. The highest-order BDF method that exhibits A-stability is the second-order BDF method (BDF-2), with the higher-order methods having progressively smaller stability regions. The Newmark methods are said to be unconditionally stable, although as with all quantitative statements about stability, this strictly only holds when applied to linear problems. BDF methods have the stiff decay property (i.e. they numerically dissipate high-frequency components in the solution), yet their spectral radius depends on the selected increment size. The NMK-β method exhibits no numerical dissipation (the same holds for the explicit CD method), whereas in the GEN- α scheme the dissipation of high-frequency oscillations can be algorithmically controlled. While long-term energy conservation is of lesser importance in transient contact dynamics, it is mentioned here that the family of Newmark integrators conserves energy in a symplectic fashion, while the BDF methods are not energy-conservative. Being multi-step methods, changing the increment size in BDF methods requires either interpolating past solution vectors or recomputing the coefficients of the update formula. The family of implicit Newmark integrators are all single-step methods with straight-forward increment size changes.

Taking into account the above considerations, the following four implicit integration schemes are investigated in the current work:

- the 2nd order BDF method (BDF-2, see Appendix I);
- the 4th order BDF method (BDF-4, see Appendix G.3);
- the Newmark-β method (NMK- β, see Section 2.6.2.2);
- the Generalized-α method (GEN- α, see Appendix J).

A comparison of these methods in terms of order of convergence, number of steps used in the update formula, error estimation, stability and dissipative properties is given in Table 7.2.

Integrator	Order	# steps in update formula	Error estimation based on…	Stability	High-freq. numerical dissipation
BDF-2	2	multistep (2)	3rd derivative approximation	A-stability > BDF-4	Uncontrollable
BDF-4	4	multistep (4)	5th derivative approximation	< BDF-2 < NMK	Uncontrollable
NMK-β	2	single-step (1)	jerk approximation	"unconditional" > BDF	None
GEN-α	2	single-step (1)	jerk approximation	"unconditional" > BDF	Algorithmically controllable

Table 7.2 Comparison of the selected implicit integration schemes

The fourth-order BDF method is included in the analysis to assess the influence of the order of convergence on the computational cost of the solution. In both BDF integrators the direct formulation (see Appendix I) is used, where the initial values of the multistep update formulas are computed using a central difference scheme. Adaptive time stepping in the BDF schemes is based on variable-coefficient update formulas and solely controls the local truncation error, not the numerical dissipation. The dynamic equilibrium equations are solved using a Newton-Raphson iterative scheme with Jacobian updates whenever convergence is slow or when the increment size has changed considerably (see Section 2.6.2). Whether the numerical or analytical contact Jacobian is used depends on the number of generalized elastic DOFs and the number of elements in the contact zones of the gears. As all of the integrators are fully implicit, quasi-static consideration of the contact shapes (Eq. 7.2.7-7.2.8) can easily be added to the schemes.

7.4 Validation and numerical experiments

7.4.1 Dynamic simulation of a parallel-axis spur gear pair

7.4.1.1 Explicit integration schemes

The numerical integrators selected in Section 7.3.1-7.3.3 are put to the test by solving the reduced-order EOMs of the spur gear pair that was analyzed in Section 4.4.1. The Interpolated Contact Shapes (ICS) approach is used to construct the ROM of the spur gear pair. The simulation parameters as well as the parameters of the ICS-based ROM are given in Table 4.3. All numerical integrators are equipped with adaptive time step control, based on an estimation of the local truncation error (see Section 2.6). The parameters for time step control are summarized in Table 7.3. Note that for the CD method, however, the integration tolerances are set ten times smaller

than in Table 7.3 to account for the reduced reliability of the CD's local error estimator (see Eq. 2.6.26).

Parameters	**Value**
Absolute integration tolerance (ATOL)	1e-8
Relative integration tolerance (RTOL)	1e-6
Relative tolerance Newton-Raphson (ITOL)	1e-10
Position constraint tolerance	1e-8
Maximal time step	1e-5 s
Minimal time step	5e-10 s
Maximal step size increase per step	5
Maximal step size decrease per step	0.1
Safety factor	0.9

Table 7.3 Parameters for the time step control

The main diagnostics of the explicit integration schemes in the numerical solution of the spur gear EOMs (Eq. 6.2.) are shown in Table 7.5. The Root-Mean-Square (RMS) error on the predicted Dynamic Transmission Error (DTE) is evaluated by comparison with the reference integrator, which is a fixed-increment Newmark integrator with very small increment size (Δt = 1e-9 s). The evolution of the increment sizes as a function of simulation time is shown in Fig. 7.3, where a moving mean with a window of 1000 is used to smoothen the data. Note that owing to the reduced frequency-content of the ICS-based ROM, the average increment sizes are considerably larger than the stable time increment for the CMS-based reference simulation (see Table 4.3), with corresponding gains in computational efficiency. With regard to the influence of the order of convergence, it can be observed that while the higher-order RK-BS method uses larger time increments than the RK-F method, this does not result in reduced CPU times due to the increased number of RHS evaluations. In APECE methods, on the other hand, increasing the order does not come at the price of additional RHS evaluations, and the fourth-order APECE scheme is observed to be more efficient than its second-order counterpart. Note that while the average increment sizes of the RK-F and APECE-4 methods are of similar magnitude, the RK-F has fewer rejected steps and allows for a simpler adjustment of increment sizes, which results in superior CPU timings. The higher-order methods come with smaller RMS errors and therefore might outperform the lower-order methods for stricter integration tolerances; however the settings of Table 7.3 proved to be sufficient for the dynamic gear contact problem. Finally the CD method, with its close-to-second-order accuracy at the cost of a single RHS evaluation per step, proves to be most effective explicit integrator for the considered application in terms of CPU time, closely followed by the RK-F method. The latter observation is of particular relevance in the case of

multiphysical and/or system-level simulations, where first- and second-order systems of ODEs have to be solved simultaneously.

Integrator	# accepted steps	# rejected steps	# RHS evaluations	Average increment size	CPU time	DTE RMS error
RK-F	33637	3838	74950	2.97e-7 s	349 s	1.7e-8 rad
RK-BS	27174	4323	94491	3.68e-7 s	452 s	9.2e-9 rad
APECE-2	59160	8891	127211	1.69e-7 s	894 s	3.3e-8 rad
APECE-4	31847	7503	78700	3.14e-7 s	512 s	6.7e-9 rad
CD	102880	22300	125180	9.72e-8 s	321 s	5.8e-7 rad

Table 7.4 Diagnostics of the selected explicit integration schemes

In order to further improve the computational efficiency of the CD method, it is combined with the approach outlined in Section 7.2. The diagnostics of the CD method with Quasi-Static consideration of the contact shapes (CD-QS) are shown in Table 7.5. Owing to the required iterative solution of Eq. 7.2.7, the number of RHS evaluations is larger in the CD-QS method; however this is compensated for by the use of larger time increments than in the standard CD method, making the CD-QS method more efficient overall. The assumptions that were introduced in Section 7.2 (Eq. 7.2.2-7.2.5) are validated by comparing the DTE predictions of the CD and CD-QS methods in Fig. 7.4. While the local relative errors in this figure are limited (< 5%), the DTE RMS error of the CD-QS integrator with respect to the reference integrator is considerable (see Table 7.5), yet acceptable for the considered application.

Integrator	# accepted steps	# rejected steps	# RHS evaluations	Average increment size	CPU time	DTE RMS error
CD	102880	22300	125180	9.72e-8 s	321 s	5.8e-7 rad
CD-QS	28653	5122	56697	3.49-7 s	182 s	6.4e-5 rad

Table 7.5 Diagnostics of the CD method with and without quasi-static consideration of the contact shapes

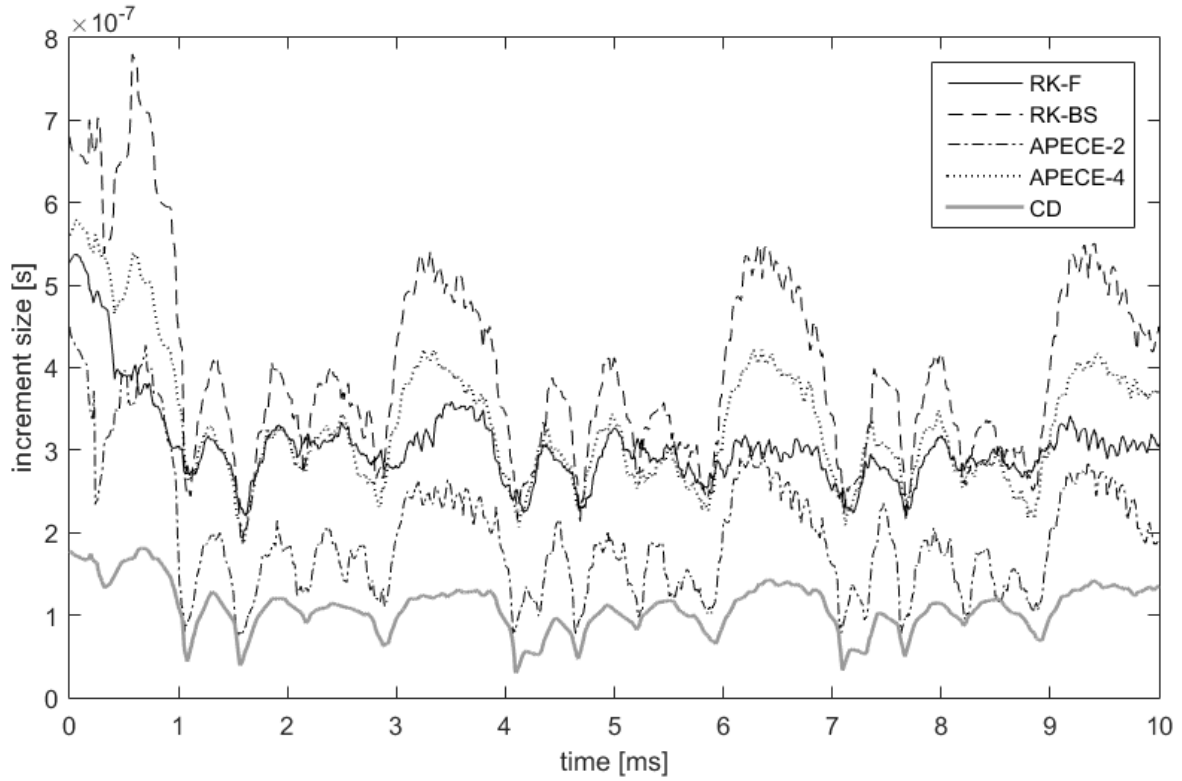


Fig. 7.3 Evolution of the time increment sizes of the explicit integration schemes

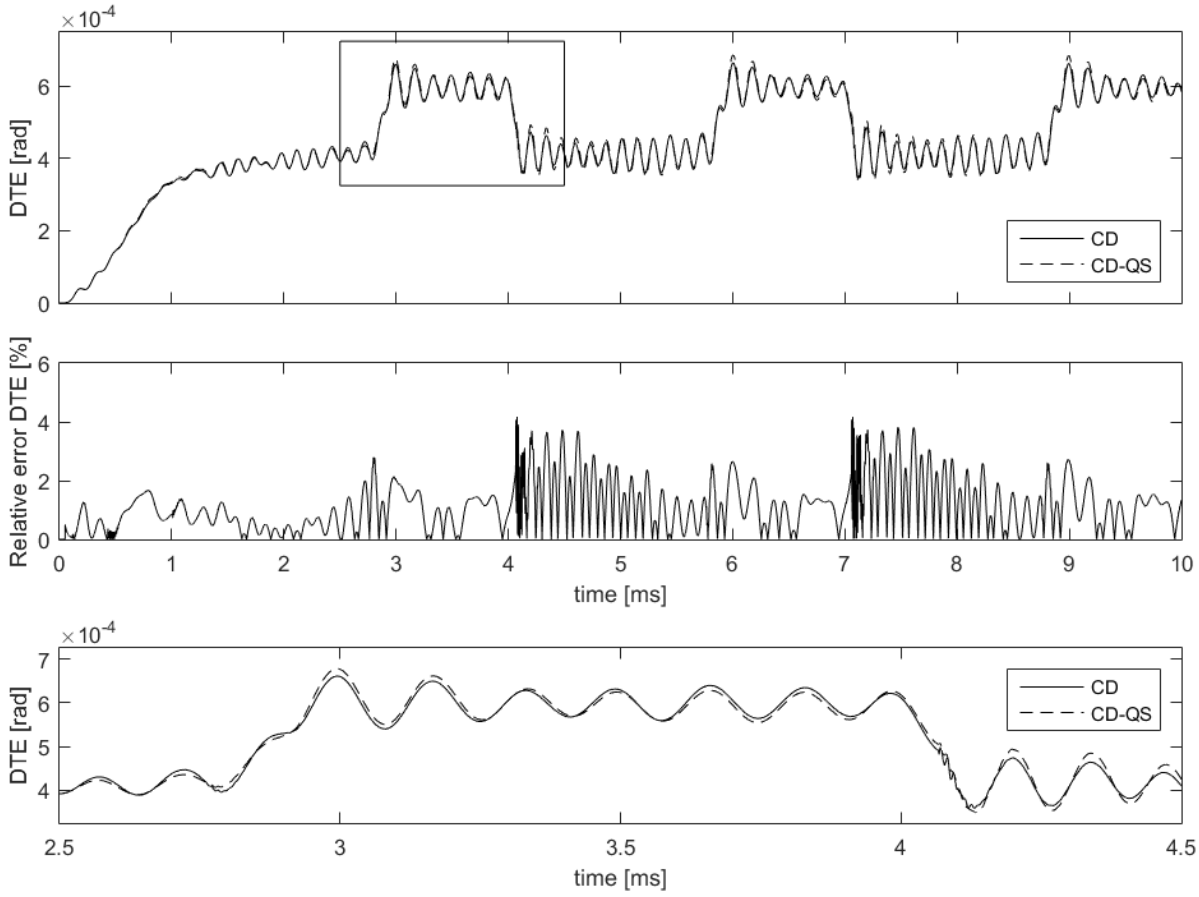


Fig. 7.4 Comparison of the DTE predictions by the CD and CD-QS methods

7.4.1.2 Implicit integration schemes

Analogous to the explicit integrators in Section 7.4.1.1, the main diagnostics of the implicit integration schemes are shown in Table 7.6. The evolution of the increment sizes as a function of simulation time is shown in Fig. 7.5. Contrary to the PECE methods of Table 7.5, increasing the order of convergence of the BDF integrator doesn't result in reduced CPU times, which is mainly due to the lower stability properties of the higher-order BDF methods. The influence of adding algorithmic numerical dissipation is limited, as is evidenced by the diagnostics of the GEN-α integrator, where the spectral radius is set to 0.9. For progressively smaller spectral radii the average increment size decreases slightly, resulting in minor increases in overall CPU time. In the dynamic gear contact problem with the parameter settings of Table 7.3, the Newmark family of integrators outperforms the BDF-2 integrator. For less stringent tolerances, however, the BDF-2 method is superior to the Newmark methods in terms of computational efficiency. This can be especially interesting when conducting long-time simulations where global gear pair characteristics are the main concern.

Integr.	# accepted steps	# rejected steps	# RHS eval.	# Jacobian updates	# iterations	Average increment size	CPU time	DTE RMS error
BDF-2	3036	188	48531	216	26067	3.29e-6 s	285 s	6.6e-8 rad
BDF-4	3559	475	82317	270	54237	2.81e-6 s	486 s	2.8e-8 rad
NMK-β	1544	68	37096	246	11512	6.48e-6 s	244 s	1.1e-8 rad
GEN-α	1650	65	35364	258	11860	6.29e-6 s	253 s	1.2e-8 rad

Table 7.6 Diagnostics of the selected implicit integration schemes

In order to further increase the computational efficiency of the Newmark integrator, four variants of the standard Newmark scheme are investigated in Table 7.7. These are the NMK method with Quasi-Static consideration of the contact shapes (NMK-QS, see Section 7.2), the NMK method with Broyden updates (NMK-B, see Section 4.2.2), the NMK method with analytical contact Jacobian (NMK-AJ, see Section 4.2.2 and Appendix A), and a combination of Broyden and analytical Jacobian updates (NMK-BAJ). In the NMK-B method, the full numerical Jacobian is only used when the Jacobian update has not lead to convergence. In the NMK-AJ the full tangent stiffness matrix of the gear pair is computed analytically each time the Jacobian is required. In the NMK-B row, a Jacobian update is counted whenever the full numerical Jacobian (requiring $n_q = 104$ RHS evaluations) is evaluated, but not when Broyden's update is used. In the NMK-AJ and NMK-BAJ rows, a Jacobian update is counted whenever the analytical Jacobian is evaluated. While strictly this involves no additional RHS evaluations, the complexity of forming the analytical

contact Jacobian surpasses that of one RHS evaluation and might even surpass the complexity of computing the full Jacobian numerically[28].

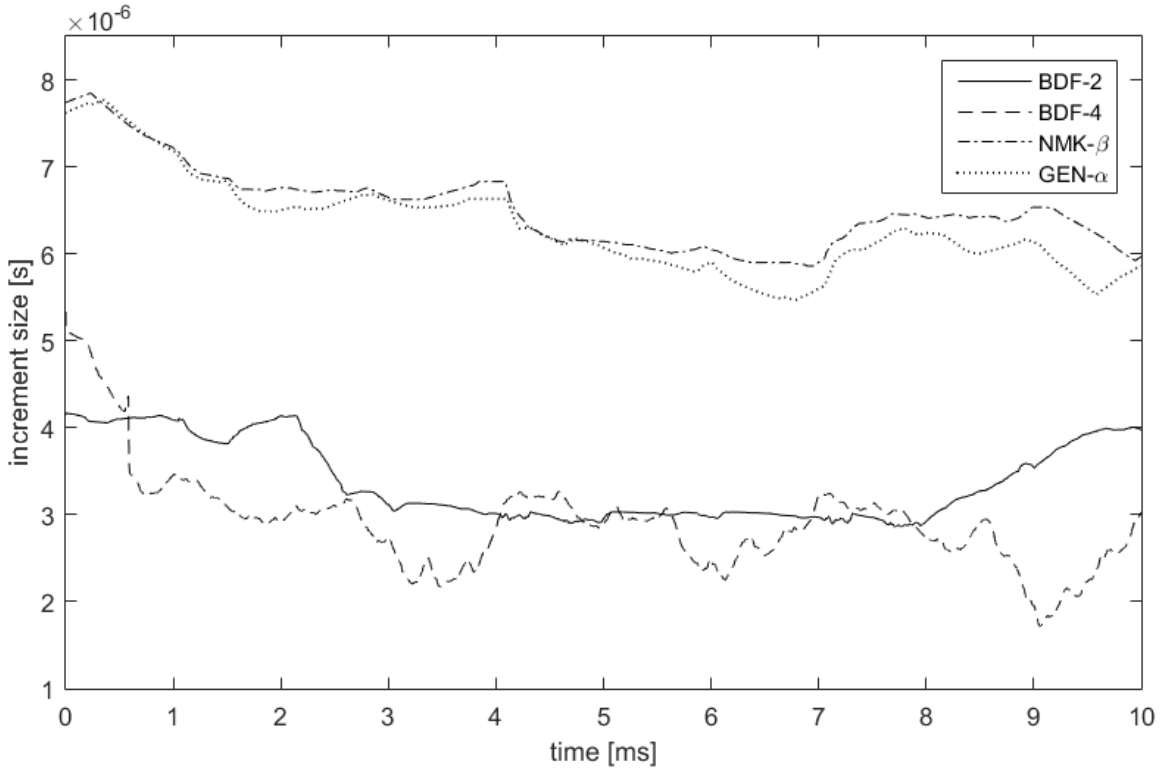


Fig. 7.5 Evolution of the time increment sizes of the implicit integration schemes

Contrary to the explicit CD method, eliminating the high-frequency solution components from the NMK scheme results in only minor decreases of CPU time (~10%). Taking into consideration the solution error that is introduced by this approach (see Fig. 7.4), the NMK-QS integrator is abandoned in the current work. The optimal implicit integrator is obtained by combining analytical Jacobian updates with Broyden's updates, which results in a reduction of the overall CPU time by roughly 40%, making the NMK-BAJ integrator approximately 20% more efficient than the semi-explicit CD-QS integrator. It is important to note though that, in the absence of hyperreduction (see Chapter 8), these conclusions are susceptible to the resolution of the FE model that is used to discretize the gear pair. Further investigations in terms of FE mesh density are necessary to generalize these conclusions.

[28] As the analytical contact Jacobian is obtained as a summation of physical contact force contributions (see Appendix A), it can become more expensive than the numerical Jacobian when the FE discretization is very fine and the number of active contact nodes becomes large. Note that this is rarely the case when using the analytical contact Jacobian to compute the contact shapes, as the number of active contact nodes in that case is generally smaller than the number of DOFs in the FOM or intermediate ROM, see Section 4.2.

Integrator	# accepted steps	# rejected steps	# RHS eval.	# Jacobian updates	# iterations	Average increment size	CPU time	DTE RMS error
NMK	1544	68	37096	246	11512	6.48e-6 s	244 s	1.1e-8 rad
NMK-QS	1236	18	34867	228	11155	8.10e-6 s	221 s	5.4e-6 rad
NMK-B	1535	78	29230	89	19974	6.51e-6 s	204 s	1.1e-8 rad
NMK-AJ	1512	63	10274	231	10274	6.61e-6 s	164 s	1.2e-8 rad
NMK-BAJ	1542	76	15278	92	15278	6.48e-6 s	147 s	1.2e-8 rad

Table 7.7 Diagnostics of the Newmark variants

7.4.2 Dynamic simulation of a planetary gear set

Section 6.2 deals with the numerical solution of the reduced-order EOMs of planetary gear sets. In these equations, the planet carrier that holds the planet gears in place is modelled using a number of constraint equations which turn the EOMs into a system of Differential-Algebraic Equations (DAEs). In order to avoid implicit integration in the computation of the CMS-based reference solution, these EOMs were transformed into a system of ODEs by eliminating the constraint equations based on the procedure outlined in Appendix B (see Section 6.2.3). This transformation comes with a considerable number of additional FLOPs (see Eq. 6.2.22-6.2.23) that have to be performed at each time step and that compromise the efficiency of the numerical solution from the onset.

Inspired by the good overall performance of implicit integration schemes when combined with the ICS-approach, it is investigated whether implicit integrators for solving the planetary gear set's DAEs (Eq. 6.2.18) can be competitive with or even outperform the explicit integrators that solve the gear set's ODEs (Eq. 6.2.22). In Table 7.8 the optimal explicit and implicit integration schemes as identified in Section 7.4.1 are applied to the planetary gear contact problem that was dealt with in Section 6.2.4 (see Table 6.7 and Table 6.8 for the simulation and ROM parameters; note that the ICS-II approach was used to construct the planetary gear ROM). The table below shows that the optimized Newmark integrator is close to twice as efficient in solving the planetary gear DAEs as the CD-QS integrator that solves the underlying ODEs. It is important to note though that due to timing restrictions the FE models used in the planetary gear example are excessively coarse, and it is highly questionable whether the same relative difference in CPU timing will be observed for finer FE models. Nevertheless, the experiment confirms the merits of implicit integrators for the solution of reduced-order industrial-sized transient contact problems.

Integrator	# accepted steps	# rejected steps	# RHS evaluations	# Jacobian updates	# iterations	Average increment size	CPU time
CD-QS	175575	29262	204837	n.a.	n.a.	1.14e-6 s	2027 s
NMK-BAJ	4695	134	117626	304	36458	4.26e-5 s	1191 s

Table 7.8 Diagnostics of the CD and NMK methods for the solution of the EOMs of a constrained planetary gear set

7.5 Summary and conclusion

In this chapter the selection of an optimal integration scheme for solving the reduced-order equations of motion of a pair of flexible gears was investigated in light of the Interpolated Contact Shapes (ICS) approach. Owing to the significant dimensional reduction that is enabled by this approach, both explicit and implicit integrators were brought to the fore. A preliminary selection among well-established numerical integrators was made based on the order of convergence and the availability of an efficient yet accurate error estimate. Using the gear contact problems introduced in Chapters 4 and 6 as study cases, the main diagnostics of the selected integrators were compared. The optimal integration scheme for the ICS-based ROM was shown to be the adaptive Newmark integrator combined with the analytical contact Jacobian (see Section 4.2.2) and Broyden's method for approximating the system's Jacobian. Among the explicit integrators, the central difference method with velocities lagging half a time step and quasi-static consideration of the contact shapes proved to be the most performant in terms of CPU time. Compared to the reference integration scheme that was used in Chapters 4 to 6, the use of these optimal integration schemes enabled a reduction of the overall simulation time with a factor of 20 and more.

Chapter 8

Efficient computation of the reduced-order contact forces

In the introductory chapter of this dissertation, three main reasons were listed for the high computational cost of Finite-Element (FE) based contact problems: the large number of Degrees Of Freedom (DOFs), the small characteristic time lengths, and the complexity of computing the contact forces. The novel Model-Order Reduction (MOR) method introduced in Chapter 4 (and improved and extended in Chapters 5 and 6) was shown to reduce the dimensions of the Full-Order Model (FOM) with several orders of magnitude. As a result of this comparatively small amount of DOFs, a class of numerical integrators was brought to the fore in Chapter 7 that would otherwise not have had a place in transient contact dynamics. Owing to their larger stability regions, these integrators are less susceptible to the sub-microsecond time scales of FE-based contact problems and operate on time increments that are much larger than what would have been possible in the FOM.

Having dealt with the first two reasons behind the high cost of FE-based contact problems, the aim of this final chapter is to investigate and reduce the complexity of computing the generalized contact forces. In Section 4.3.2, the numerical complexity of the proposed Interpolated Contact Shapes (ICS) approach was investigated in terms of the number of Floating Point Operations (FLOPs) required to form the system's update equations (Eq. 4.3.7). This investigation revealed that in dynamic analyses up to 98% of the CPU time is spent on evaluating and projecting the contact forces (see Table 4.2). This means that any speed-up in computing these forces will directly translate in similar gains in overall simulation efficiency.

In this chapter two state-of-the-art hyperreduction schemes (see Section 2.3.4) for reducing the complexity of high-dimensional nonlinear problems will be extended and applied to the dynamic gear contact problem. In Section 8.1 a comprehensive investigation of the full-order and reduced-order contact forces is conducted in preparation of Sections 8.2 and 8.3. These sections deal with the extension of the Discrete Empirical Interpolation Method (DEIM, see Section 2.3.4.1) and the Energy-Conserving Weighting and Sampling (ECSW) method (see Section 2.3.4.2) to the analysis of FE-based contact problems. Particular attention is given to the conservation of linear momentum in the hyperreduced ROM, and to the design of hyperreduction algorithms in a parametric MOR context. Finally in Section 8.4 the two hyperreduction schemes are validated by comparison with the reference model and an assessment of the overall computational efficiency is made.

8.1 Preliminary considerations on the nonlinear contact forces

8.1.1 Computation of the full-order contact forces

A proper understanding of the mechanics behind the full- and reduced-order contact force computation is a prerequisite for increasing the efficiency of enforcing the contact constraints. In this section, a comprehensive analysis of the algorithm that is used to compute the nonlinear contact forces is conducted in preparation of the analysis of hyperreduction techniques in Section 8.2 and 8.3.

In this dissertation, as in the majority of FE-based dynamic contact problems, contact constraints are enforced using a penalty approach combined with a Node-To-Surface (NTS) algorithm (see Section 2.5.2.2). In every contact interaction between two flexible bodies, one body is designated as the *master* body and the other as the *slave* body, where the nodes of the *slave* body are said to penetrate the element faces of the *master* body. The robustness of the NTS algorithm is significantly increased by applying a two-pass strategy where each body is alternately treated as master and slave and the resulting contact forces are averaged. In the following, these concepts are applied to the dynamic gear contact problem.

Let $\boldsymbol{f}_c^{it} \in \mathbb{R}^{N_f^i}$ denote the vector of physical contact forces associated with the active flank of the *t*-th tooth of the *i*-th gear. When a two-pass algorithm is used to compute the contact forces, $\boldsymbol{f}_c^{it}$ is obtained as

$$\boldsymbol{f}_c^{it} = \frac{1}{2}\boldsymbol{f}_c^{it,s} + \frac{1}{2}\boldsymbol{f}_c^{it,m} \tag{8.1.5}$$

where $\boldsymbol{f}_c^{it,s}$ and $\boldsymbol{f}_c^{it,m}$ are the vectors of contact forces obtained by consecutively treating gear *i* as slave and master, respectively. The total vector of physical contact forces $\boldsymbol{f}_c^{t}$ associated with the *t*-th engaging *tooth pair* is obtained by vertically stacking $\boldsymbol{f}_c^{1t}$ and $\boldsymbol{f}_c^{2t}$, and can be written as

$$\boldsymbol{f}_c^{t} = \begin{Bmatrix} \boldsymbol{f}_c^{1t} \\ \boldsymbol{f}_c^{2t} \end{Bmatrix} = \frac{1}{2}\underbrace{\begin{Bmatrix} \boldsymbol{f}_c^{1t,s} \\ \boldsymbol{f}_c^{2t,m} \end{Bmatrix}}_{1^{st}\ pass} + \frac{1}{2}\underbrace{\begin{Bmatrix} \boldsymbol{f}_c^{1t,m} \\ \boldsymbol{f}_c^{2t,s} \end{Bmatrix}}_{2^{nd}\ pass} = \frac{1}{2}\boldsymbol{f}_c^{t,sm} + \frac{1}{2}\boldsymbol{f}_c^{t,ms} \tag{8.1.6}$$

Note that during the first pass the pinion ($i = 1$) is treated as slave body and the wheel ($i = 2$) as master body, while during the second pass this ordering is reversed.

In the convention of the NTS approach, *slave* nodes penetrate *master* elements. Therefore, slave forces $\boldsymbol{f}_c^{it,s}$ are said to be pure nodal forces, whereas master forces $\boldsymbol{f}_c^{it,m}$ are elemental forces in the sense that they are obtained by distributing point forces across the corner nodes of the penetrated element faces (see Fig. 2.10). Furthermore, slave nodes can penetrate a master element

only once, whereas master elements can be penetrated by multiple slave nodes at the time. In view of the hyperreduction schemes developed in the next two sections, it is important to note that this implies that each nodal contact force on the slave gear is the result of a single contact interaction between that node and an element of the master gear, whereas nodal contact forces on the master gear can be the result of multiple contact interactions involving all adjacent master elements and an a priori unknown number of slave nodes.

The full-order contact force algorithm takes as input the (deformed) grid coordinates $\boldsymbol{r}^{sj}$ and $\boldsymbol{r}^{mj}$ of the master and slave nodes that are likely to be in contact (see Section 3.2.4). An outer loop is then performed over all master elements and an inner loop over all slave nodes to detect the master-slave interactions by solving a minimum distance problem (see Section 2.5.2.2). Whenever a slave node penetrates a master element, the nodal contact forces are computed using Eq. 2.5.6-2.5.11. The computational complexity of the NTS algorithm is measured by counting the number of FLOPs, where a FLOP is assumed to be either a real multiplication or a real summation. Denoting N_{fef} and N_{few} as the number of finite elements along the involute curve of a tooth and along the width of the gear, the (worst-case) numbers of FLOPs in an efficient implementation of a three-dimensional NTS algorithm are shown in Table 3..

Operation	**# FLOPs**
Single slave node in contact with one element	≈ 350
Loop over N_{sl} slave nodes to detect contact with one element	$\approx 100 + 200N_{sl}$
Inner loop over all slave nodes to detect contact with one element	$\approx 50N_{fef}N_{few}$
Outer loop over all master elements to detect master-slave contact	$\approx 10N_{fef}^2N_{few}^2$
Two-pass NTS algorithm to detect contact between one pair of teeth	$\approx 20N_{fef}^2N_{few}^2$

Table 8.1 Number of FLOPs in a three-dimensional NTS algorithm

For a typical finite element discretization, where $N_{fef} \approx 30 - 50$ and $N_{few} \approx 10 - 20$, the number of FLOPs varies between roughly 2e6 and 2e7 FLOPs. When executed on a dual-core Intel ® Core ™ i5-3320M Processor having a clock speed of 2.6 GHz and capable of processing 4 FLOPs per cycle (which amounts to roughly 20 GigaFLOPs per second), the number of contact interactions that can in theory be processed per second lies between 1000 and 10.000. This might seem like a lot, but factor in a time increment of $\Delta t = 10^{-8}s$ in an explicit single-stage integration scheme and a contact ratio between 3 and 4, and it will take the processor between 10 and 100 hours to perform a 1 second simulation (i.e. four to five orders of magnitude slower than real-time); that is, considering only the computation of full-order contact forces.

8.1.2 Computation of the reduced-order contact forces

The NTS algorithm described in the previous section operates on the grid coordinates of the discretized master and slave bodies and computes nodal contact forces in the physical domain. In order to integrate this algorithm in a reduced-order setting, the input and output variables need to be transformed from the physical (full-order) domain to the reduced-order (or modal) domain.

The physical grid coordinates $\boldsymbol{r}^{ij}$ of the master and slave body are obtained from the respective vectors of generalized coordinates as follows (see Eq. 2.5.14):

$$\boldsymbol{r}^{ij} = \boldsymbol{R}^i + \boldsymbol{A}^i(\boldsymbol{\theta}^i)\left(\bar{\boldsymbol{r}}_0^{ij} + \bar{\boldsymbol{V}}^{ij}\boldsymbol{q}_f^i\right) \tag{8.1.7}$$

where $\boldsymbol{R}^i$ and $\boldsymbol{\theta}^i$ denote the global position and orientation of the local reference frame of the *i*-th body, $\boldsymbol{q}_f^i$ is the vector of generalized elastic coordinates, $\bar{\boldsymbol{r}}_0^{ij}$ is the local undeformed position vector of the *j*-th node on the *i*-th body and $\bar{\boldsymbol{V}}^{ij} \in \mathbb{R}^{3\times n_f}$ is the restriction of the basis matrix to the rows corresponding to this node. The number of FLOPs required to compute this projection from the physical to the modal domain for both the master and slave bodies is roughly equal to $500 + 3n_f N_{fef} N_{few}$ (counting 20 FLOPs for trigonometric function evaluations).

Having obtained the nodal contact forces (see previous section), these are projected back to the reduced-order domain to obtain the reduced-order or generalized contact forces $\boldsymbol{Q}_c^i$ (see Section 2.4.3):

$$\boldsymbol{Q}_c^i = \sum_j \boldsymbol{Q}_c^{ij} = \sum_j \left\{ \begin{bmatrix} \boldsymbol{I}_3 \\ \bar{\boldsymbol{G}}^{i\,T}\tilde{\bar{\boldsymbol{r}}}^{ij}\boldsymbol{A}^{i\,T} \\ \bar{\boldsymbol{V}}^{ij\,T}\boldsymbol{A}^{i\,T} \end{bmatrix} \boldsymbol{f}_c^{ij} \right\} = \sum_j \mathbb{V}^{ij\,T}\boldsymbol{f}_c^{ij} \tag{8.1.8}$$

where the matrix $\mathbb{V}^{ij}$ is defined as

$$\mathbb{V}^{ij\,T} = [\boldsymbol{I}_3 \quad -\boldsymbol{A}^i\tilde{\bar{\boldsymbol{r}}}^{ij}\bar{\boldsymbol{G}}^i \quad \boldsymbol{A}^i\bar{\boldsymbol{V}}^{ij}]^T = \begin{bmatrix} \boldsymbol{I}_3 \\ \bar{\boldsymbol{G}}^{i\,T}\tilde{\bar{\boldsymbol{r}}}^{ij}\boldsymbol{A}^{i\,T} \\ \bar{\boldsymbol{V}}^{ij\,T}\boldsymbol{A}^{i\,T} \end{bmatrix} \tag{8.1.9}$$

and where $\bar{\boldsymbol{G}}^i$ is the transformation matrix of body *i* expressing the angular velocities $\bar{\boldsymbol{\omega}}^i$ in terms of the time derivatives of the rotational coordinates $\dot{\boldsymbol{\theta}}^i$ (see Appendix C). The number of FLOPs required to transform the nodal contact forces to generalized contact forces, assuming one fourth of the contact candidates are actually in contact, is roughly equal to $2n_f N_{fef} N_{few}$

In summary, the above equations express the vector of generalized contact forces $\boldsymbol{Q}_c^i$ in terms of the vector of generalized coordinates $\boldsymbol{q}$. The relationship $\boldsymbol{Q}_c^i = \boldsymbol{Q}_c^i(\boldsymbol{q})$ is graphically represented in Fig. 8.1. An overview of the numbers of FLOPs involved in computing $\boldsymbol{Q}_c^i$ starting from $\boldsymbol{q}$ is given in Table 8.2. Note that despite the reduced dimensions of $\boldsymbol{Q}_c^i \in \mathbb{R}^{n_q \ll N}$, the

numerical complexity of $\boldsymbol{Q}_c^i$ still scales with the dimensions of the original FOM ($\approx 20N_{fef}^2 N_{few}^2$). This considerably limits the computational efficiency of the ROM. In order to remove the dependence of the ROM's complexity on the size of the underlying FOM and to further increase its efficiency, a second level of approximation is required. This is where the field of hyperreduction comes into play.

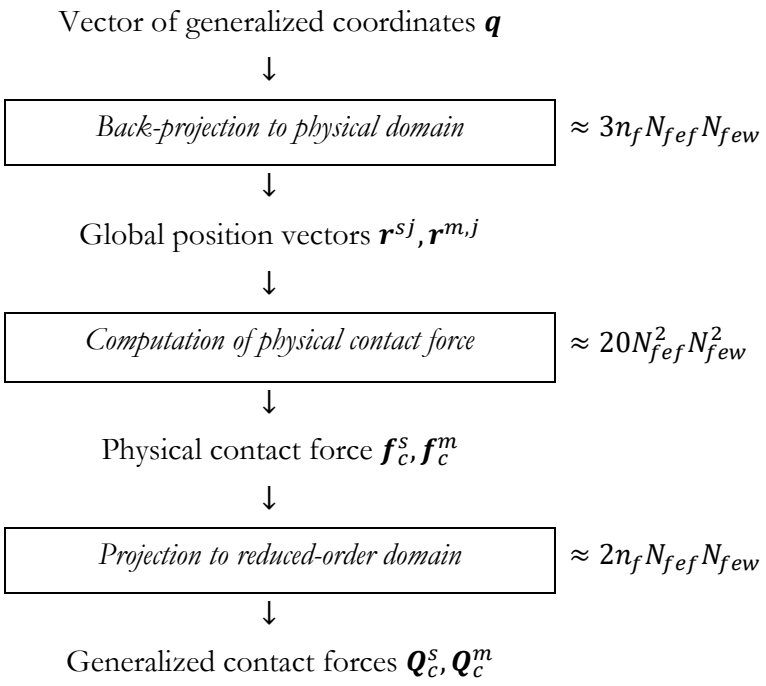


Fig. 8.1 Subspace projections in the generalized contact force computation

Operation	**# FLOPs**
Computation of grid coordinates of a single node	$\approx 18 + 6n_f$
Computation of grid coordinates of all candidate slave and master nodes	$\approx 3n_f N_{fef} N_{few}$
Two-pass NTS algorithm to detect contact between one pair of teeth	$\approx 20N_{fef}^2 N_{few}^2$
Back-projection of a single nodal contact force	$\approx 140 + 2n_f$
Back-projection of all nodal contact forces	$\approx 2n_f N_{fef} N_{few}$

Table 8.2 Number of FLOPs in computing the generalized contact forces

8.1.3 Hyperreduction of dynamic contact problems

Most if not all successful hyperreduction schemes rely on the selective evaluation of a subset of components of the *full-order* nonlinear function. They therefore require the back-and-forth projection of at least some components of the nonlinear function and/or generalized coordinates

between the reduced- and full order subspaces. The Discrete Empirical Interpolation Method (DEIM [66], see Section 2.3.4.1) and the Gauss-Newton method with Approximated Tensors (GNAT [105]) approximate the full-order nonlinear function in the physical domain before projecting it in the reduced-order subspace, whereas the Energy-Conserving Sampling and Weighting method (ECSW [67], see Section 2.3.4.2) directly approximates the reduced-order nonlinear function via a comparatively small number of back-and-forth subspace projections on element level.

While the DEIM and ECSW method are designed for reducing the complexity of general nonlinear Finite Element Models (FEMs), they are commonly applied for approximating nonlinear internal forces where the nonlinearity is due to geometric and/or material nonlinearities (see Section 1.2.5 and the references therein). In these applications, it is common for the nonlinear function to be composed of element-wise contributions involving *all* the elements in the full-order FEM. The high numerical complexity of the nonlinear function in these applications originates primarily from the multitude of elements, rather than from the cost of evaluating the nonlinear function at the element-level. Since evaluating this function at the element-level typically only requires knowledge of the element's nodal DOFs, the aforementioned hyperreduction schemes can achieve considerable computational savings by expressing the nonlinear function in terms of a comparatively small number of elemental contributions.

The same is not true for multibody contact problems. In these problems, the nonlinear function to be approximated – in casu: the vector of nonlinear contact forces – is composed of expensive element-wise contributions involving only a small subset of elements in the full-order FEM (i.e. the elements in the contact zones of the interacting bodies). Potential computational savings are further reduced by the fact that the nonlinear contact forces at the element-level not only depend on the element's nodal DOFs, but also on an a priori undetermined set of DOFs belonging to the other body. Finally, because of the continuous relative motion of the interacting bodies, the contact force patterns and locations change considerably during the simulation. This poses significant challenges to the standard DEIM and ECSW schemes, which fail to identify a sufficiently small set of elemental contributions that can be used to reconstruct the nonlinear contact force vector.

Apart from the challenges in developing a hyperreduction scheme that achieves a considerable reduction in the number of FLOPs, there is one important conservation law that should be preserved by the hyperreduction scheme, namely the law of conservation of linear momentum or Newton's third law: action equals reaction. Any violation of this law in a static or dynamic contact analysis will result in loss of solution stability and will inevitably lead to convergence failure. The principle of action and reaction implies that the sum of all master forces along any direction should equal the sum of all slave forces in that same direction. In Section 8.2 and 8.3, it is investigated how the standard DEIM and ECSW need to be adapted to construct efficient and momentum-conserving hyperreduced ROMs in the field of computational contact dynamics.

8.2 Hyperreduction using the Discrete Empirical Interpolation Method

8.2.1 A DEIM scheme for contact problems

A general nonlinear function $\boldsymbol{f}(t) \in \mathbb{R}^N$ is approximated by the DEIM using the following linear superposition (see Section 2.3.4.1):

$$\boldsymbol{f}(t) \approx \boldsymbol{Y}\boldsymbol{c}(t) = \underbrace{\boldsymbol{Y}(\boldsymbol{P}^T\boldsymbol{Y})^{-1}}_{\textit{precomputed}}\underbrace{\boldsymbol{P}^T\boldsymbol{f}(t)}_{\textit{selective evaluation}} \tag{8.2.1}$$

where $\boldsymbol{Y} = [\boldsymbol{y}_1 \quad \cdots \quad \boldsymbol{y}_h] \in \mathbb{R}^{N\times h}$ is the projection basis obtained by compressing a matrix of precomputed function snapshots, $\boldsymbol{c}(t) \in \mathbb{R}^h$ is the corresponding coefficient vector, and $\boldsymbol{P} \in \mathbb{R}^{N\times h}$ is a Boolean matrix that selects the entries of $\boldsymbol{f}(t)$ that need to be evaluated. The boolean matrix $\boldsymbol{P}$ is constructed using a greedy algorithm (Algorithm 2.2) that operates on the projection basis $\boldsymbol{Y}$. Note that $\boldsymbol{P}$ is only used here as a notational aid; in practical implementations of the DEIM, an index vector – referred to as the vector of DEIM indices – replaces $\boldsymbol{P}$. In FE analyses where the nonlinear function is composed of element-wise contributions, it is usually more efficient to apply the DEIM in unassembled form, in which case Eq. 8.2.1 takes the following form:

$$\boldsymbol{f}(t) \approx \boldsymbol{Y}\boldsymbol{c}(t) = \underbrace{\boldsymbol{Y}({}^u\boldsymbol{P}^T{}^u\boldsymbol{Y})^{-1}}_{\textit{precomputed}}\underbrace{{}^u\boldsymbol{P}^T{}^u\boldsymbol{f}(t)}_{\textit{selective evaluation}} \tag{8.2.2}$$

where the left-superscript 'u' refers to unassembled vectors and matrices. The unassembled nonlinear function ${}^u\boldsymbol{f}(t)$ is obtained by vertically stacking the unassembled elemental contributions, i.e.

$${}^u\boldsymbol{f}(t) = \begin{Bmatrix} \boldsymbol{f}^{e_1}(t) \\ \vdots \\ \boldsymbol{f}^{e_{n_e}}(t) \end{Bmatrix} \tag{8.2.3}$$

where $\boldsymbol{f}^{e_e}$ denotes the elemental contribution of the e-th element and n_e is the number of contribution elements. The assembled nonlinear function $\boldsymbol{f}(t)$ is obtained from the unassembled function by a regular FE assembly of element contributions, and the same holds for the unassembled projection vectors, i.e.

$$\boldsymbol{f}(t) = \texttt{assemble}({}^u\boldsymbol{f}(t)) \qquad \text{and} \qquad \boldsymbol{Y} = \texttt{assemble}({}^u\boldsymbol{Y}) \tag{8.2.4}$$

The aim of this section is to apply Eq. 8.2.1 and/or Eq. 8.2.2 to the approximation of the full-order contact force in dynamic gear contact problems.

The modified DEIM scheme used in this dissertation to approximate the vector of physical contact forces $\boldsymbol{f}_c^t$ proceeds as follows. During the offline phase of the simulation, a number of snapshots of slave and master contact forces are collected for both gears, where each pass of the two-pass NTS algorithm is treated separately. The slave forces are collected in assembled form whereas the master forces, being elemental forces (see Section 8.1.1), are collected in unassembled form. In order to preserve Newton's third law, the slave and master forces corresponding to a single snapshot are stacked vertically in one vector and are added to the unassembled snapshot matrix ${}^{U}\boldsymbol{X}_c^{t,sm}$ (first pass) or ${}^{U}\boldsymbol{X}_c^{t,ms}$ (second pass). These matrices are decomposed using the Singular Value Decomposition (SVD) to obtain the DEIM projection matrices ${}^{u}\boldsymbol{Y}_c^{t,sm}$ and ${}^{u}\boldsymbol{Y}_c^{t,ms}$:

$$ {}^{u}\boldsymbol{Y}_c^{t,sm} = \left[{}^{u}\boldsymbol{y}_{c,1}^{t,sm} \quad \cdots \quad {}^{u}\boldsymbol{y}_{c,h}^{t,sm} \right] = \begin{bmatrix} {}^{u}\boldsymbol{Y}_c^{1t,s} \\ {}^{u}\boldsymbol{Y}_c^{2t,m} \end{bmatrix} \tag{8.2.5} $$

and likewise for ${}^{u}\boldsymbol{Y}_c^{t,ms}$. The first N_f^1 rows of these matrices correspond to contact forces acting on the t-th active tooth of the first gear, and the next N_f^2 rows correspond to contact forces on the t-th active tooth of the second gear. The SVD preserves the force equilibrium between DOFs corresponding to master and slave forces, as will be demonstrated in Section 8.2.4. The Boolean matrices $\boldsymbol{P}^{t,sm}$ and $\boldsymbol{P}^{t,ms}$ (or, equivalently, the associated DEIM indices) are constructed by applying the greedy algorithm to the (compressed) snapshots, i.e.

$$ \boldsymbol{P}^{t,sm} = \texttt{algorithm2.2}({}^{u}\boldsymbol{Y}_c^{t,sm}) \tag{8.2.6} $$

and likewise for ${}^{u}\boldsymbol{P}^{t,ms}$. These boolean matrices identify the slave and master contact forces that need to be evaluated for constructing the weighting coefficients associated with ${}^{u}\boldsymbol{Y}_c^{t,sm}$ and ${}^{u}\boldsymbol{Y}_c^{t,ms}$. During the online phase of the simulation, the contact forces $\boldsymbol{f}_c^{t,sm}$ and $\boldsymbol{f}_c^{t,ms}$ are approximated by

$$ \boldsymbol{f}_c^{t,sm} \approx \underbrace{\boldsymbol{Y}_c^{t,sm}\left(\boldsymbol{P}^{t,sm\,T}\,{}^{u}\boldsymbol{Y}_c^{1t,s}\right)^{-1}}_{\textit{precomputed}} \underbrace{\boldsymbol{P}^{t,sm\,T}\,{}^{u}\boldsymbol{f}_c^{t,sm}}_{\textit{selective evaluation}} \tag{8.2.7} $$

and likewise for $\boldsymbol{f}_c^{t,ms}$, where the assembled vectors and matrices are obtained by applying Eq. 8.2.4. Finally, the full vector of physical contact forces associated with the t-th engaging tooth pair is obtained using Eq. 8.1.6.

8.2.2 Local hyperreduction using the modified DEIM scheme

Eq. 8.2.5-8.2.7 lay out the general outline of the modified DEIM, yet two main issues remain unsolved. The first relates to the construction of the matrix of contact force snapshots. One approach would be to conduct a training simulation using the (unhyperreduced) ROM or FOM, storing all the contact forces and applying the Proper Orthogonal Decomposition (POD, see Section 2.3.3.2) method to identify the dominant directions in the snapshots matrix. A cheaper alternative would be to simply store the contact forces computed during the offline computation of (misaligned) contact shapes (see Section 6.1) at various gear pair configurations, assuming the dynamic contact forces can be approximated accurately by a superposition of static contact forces. Given the importance of a low offline simulation cost, the latter approach is adapted in this dissertation. Note that also in this approach the force snapshots can be collected using the ROM instead of the FOM, in which case they come at virtually no cost.

The second unresolved issue relates to the dependence of the pattern and location of the contact force distribution on the angular configuration of the gear pair. In line with the discussion regarding local versus global parametric ROMs (see Section 2.3.2), two variants of the modified DEIM can be designed. In the global variant, the contact force snapshot matrix contains data covering the full range of angular configurations and the Boolean matrices remain valid throughout the full tooth engagement cycle. However, the large number of projection vectors that are then required to accurately approximate contact forces at all possible angular configurations is detrimental to the efficiency of the global variant. In the local variant, the projection bases $\boldsymbol{Y}_c^{t,sm}$ and $\boldsymbol{Y}_c^{t,ms}$ as well as the associated Boolean matrices depend on the current angular configuration of the gear pair. However, contrary to parametric Reduced-Order Bases (ROBs), it is not possible to interpolate Boolean matrices (or, equivalently, to interpolate DEIM indices). There are two ways to work around this, referred to as the localDEIM-I and the localDEIM-II approach.

In the localDEIM-I approach, the DEIM indices and projection vectors are computed for each angular configuration separately. Whenever the gear pair is at an intermediate angular configuration $\theta_j^1 < \theta^1 < \theta_{j+1}^1$, the contact forces are approximated twice using the DEIM ingredients of the adjacent sampling points θ_j^1 and θ_{j+1}^1, respectively, and the results are interpolated based on the angular configuration of the gear pair. Based on Eq. 8.2.7, the localDEIM-I approach can be summarized as:

$$\begin{aligned}\boldsymbol{f}_c^{t,sm} \approx \left(1 - p(\theta^1)\right) \cdot \boldsymbol{Y}_{c,j}^{t,sm} \left(\boldsymbol{P}_j^{t,sm^T u} \boldsymbol{Y}_{c,j}^{t,sm}\right)^{-1} \boldsymbol{P}_j^{t,sm^T u} \boldsymbol{f}_c^{t,ms} \\ + \; p(\theta^1) \cdot \boldsymbol{Y}_{c,j+1}^{t,sm} \left(\boldsymbol{P}_{j+1}^{t,sm^T u} \boldsymbol{Y}_{c,j+1}^{1t,s}\right)^{-1} \boldsymbol{P}_{j+1}^{t,sm^T u} \boldsymbol{f}_c^{t,ms}\end{aligned} \tag{8.2.8}$$

where the interpolation parameter p is as defined in Chapter 4, Section 4.1.2. As will be demonstrated by means of a conceptual example in Section 8.2.4, interpolating contact forces based on the angular configuration of the gear pair only leads to accurate results when the number

of angular configuration samples is very large, which is why the localDEIM-I approach is discarded in this dissertation in favor of the localDEIM-II approach.

In the localDEIM-II approach, the DEIM projection bases and Boolean matrices are computed by considering two adjacent angular configurations θ_j^1 and θ_{j+1}^1 at the time, i.e.

$$\boldsymbol{Y}_{c,j}^{t,sm} = \mathrm{svd}\left(\left[\boldsymbol{X}_{c,j}^{t,sm} \quad \boldsymbol{X}_{c,j+1}^{t,sm}\right]\right) \qquad \text{and} \qquad \boldsymbol{P}_j^{t,sm} = \texttt{algorithm2.2}\left({}^{u}\boldsymbol{Y}_{c,j}^{t,sm}\right) \tag{8.2.9}$$

where $\mathrm{svd}(\square, h)$ computes the left-singular vectors of $\square$, and $\boldsymbol{X}_{c,j}^{t,sm}$ and $\boldsymbol{X}_{c,j+1}^{t,sm}$ are the snapshot matrices of physical contact forces computed at the angular configurations θ_j^1 and θ_{j+1}^1, respectively. During simulation, these projection bases and Boolean matrices are substituted whenever the gear pair is in a new $\left[\theta_j^1, \theta_{j+1}^1\right]$ interval. One might say this is a global approach on a local level, or a local clustering approach. For moderate numbers of angular configuration samples, the localDEIM-II outperforms the localDEIM-I approach for comparable values of h, as will be demonstrated in Section 8.2.4.

8.2.3 Error-based selection of the DEIM indices

The standard algorithm to select the DEIM indices (Algorithm 2.2) is based on a predetermined set of h projection vectors $\boldsymbol{y}_i$ that are typically obtained by forming the SVD of the selected snapshot matrix and retaining only the first h left-singular vectors. The algorithm terminates when h DEIM indices have been identified. While the relative magnitudes of the singular values of the snapshot matrix can steer the selection of an adequate number of retained left-singular values, they lack a direct connection with the accuracy of the approximation. In order to fully automate the DEIM algorithm and eliminate all decisions based on the user's personal assessment, an error-based selection of the DEIM indices is needed.

In the original article on the DEIM [66] a conservative a posteriori bound on the error of the DEIM is given. Given a nonlinear function $\boldsymbol{f}$ and its DEIM approximation $\hat{\boldsymbol{f}}$ as given by Eq. 8.2.1, the error bound for $\hat{\boldsymbol{f}}$ is given by:

$$\left\|\boldsymbol{f} - \hat{\boldsymbol{f}}\right\|_2 \leq \|(\boldsymbol{P}^T\boldsymbol{Y})^{-1}\|_2 \|(\boldsymbol{I} - \boldsymbol{Y}\boldsymbol{Y}^T)\boldsymbol{f}\|_2 \tag{8.2.10}$$

Furthermore, it has been proven that when $\boldsymbol{f}$ is nearly in the range of the snapshot matrix, the second term in the above product can be approximated by

$$\|(\boldsymbol{I} - \boldsymbol{Y}\boldsymbol{Y}^T)\boldsymbol{f}\|_2 \approx \sigma_{h+1} \tag{8.2.11}$$

where σ_{h+1} is the largest singular value corresponding to the discarded set of singular vectors. Eq. 8.2.10-8.2.11 thus provide a cheap bound on the error introduced by the DEIM, as it only requires the inversion of an $h \times h$ matrix, a 2-norm and a scalar multiplication.

Algorithm 8.1 shows a modification to the original DEIM algorithm where the above error bound is included to steer the greedy selection of DEIM indices. The error bound is defined in terms of the relative error $\tau = \|\boldsymbol{f} - \hat{\boldsymbol{f}}\|_2 / \|\boldsymbol{f}\|_2$, which allows to replace the left-hand side of Eq. 8.2.10 by $\tau\|\boldsymbol{f}_{ref}\|_2$, where $\boldsymbol{f}_{ref}$ is a selected reference snapshot (e.g. the snapshot with the highest 2-norm, or in the case of the gear contact problem the vector of perfectly aligned contact forces). The performance of the modified DEIM selection scheme is illustrated in the next section.

Algorithm 8.1 Error-based procedure for computing the Unassembled DEIM ingredients

Input: Snapshot matrix ${}^u\boldsymbol{X} \in \mathbb{R}^{N\times m}$, reference snapshot ${}^u\boldsymbol{f}_{ref}$, relative error tolerance τ

Output: projection basis $\boldsymbol{Y} \in \mathbb{R}^{N\times h}$, DEIM indices $\boldsymbol{\rho} = \{\rho_i\} \in \mathbb{R}^h$, Boolean matrix $\boldsymbol{P} \in \mathbb{R}^{N\times h}$

1. Compute SVD of ${}^u\boldsymbol{X}$: $[{}^u\boldsymbol{U}, \sim, \sim] = \text{svd}({}^u\boldsymbol{X}, '\text{econ}')$, where ${}^u\boldsymbol{U} = [{}^u\boldsymbol{u}_1 \quad \cdots \quad {}^u\boldsymbol{u}_m]$
2. Find index of maximum value in $|\boldsymbol{y}_1|$ and set equal to ρ_1
3. ${}^u\boldsymbol{Y} \leftarrow [{}^u\boldsymbol{u}_1]$, $\boldsymbol{P} \leftarrow [\boldsymbol{e}_{\rho_1}]$, $\boldsymbol{\rho} \leftarrow \{\rho_1\}$
4. Compute error bound: $\varepsilon = \|(\boldsymbol{P}^T{}^u\boldsymbol{Y})^{-1}\|_2 \|(\boldsymbol{I} - {}^u\boldsymbol{Y}{}^u\boldsymbol{Y}^T){}^u\boldsymbol{f}_{ref}\|_2$
5. Initialize counter: $i = 1$
6. **while** $\varepsilon \geq \tau\|{}^u\boldsymbol{f}_{ref}\|_2$ **do**
7. Increment counter: $i \leftarrow i + 1$
8. Solve $(\boldsymbol{P}^T{}^u\boldsymbol{Y}_{i-1})\boldsymbol{c} = \boldsymbol{P}^T{}^u\boldsymbol{u}_i$ for $\boldsymbol{c}$
9. Compute the residual vector $\boldsymbol{r} = {}^u\boldsymbol{u}_i - {}^u\boldsymbol{Y}_{i-1}\boldsymbol{c}$
10. Find index of maximum value in $|\boldsymbol{r}|$ and set equal to ρ_i
11. ${}^u\boldsymbol{Y} \leftarrow [{}^u\boldsymbol{Y} \quad \boldsymbol{u}_i]$, $\boldsymbol{P} \leftarrow [\boldsymbol{P} \quad \boldsymbol{e}_{\rho_1}]$, $\boldsymbol{\rho} \leftarrow \{\boldsymbol{\rho} \quad \rho_i\}$
12. Compute error bound $\varepsilon = \|(\boldsymbol{P}^T{}^u\boldsymbol{Y})^{-1}\|_2 \|(\boldsymbol{I} - {}^u\boldsymbol{Y}{}^u\boldsymbol{Y}^T){}^u\boldsymbol{f}_{ref}\|_2$
13. **end**
14. Assemble $\boldsymbol{Y} = \texttt{assemble}({}^u\boldsymbol{Y})$

8.2.4 A conceptual example

In this section, the modified DEIM scheme outlined above is illustrated conceptually for a gear pair composed of two identical spur gears (see Section 3.2.5). The gear pair is fixed at a certain angular configuration θ_j^1 and 16 misaligned contact shapes are computed using the greedy procedure discussed in Section 6.1.3.2, while storing the resulting physical contact forces at each misalignment configuration. Slave and master contact forces are computed separately, so for each tooth pair in contact there are four vectors of physical contact forces: the two vectors of slave contact forces acting on the tooth flanks of the pinion and wheel, respectively, and the two vectors of master contact forces.

In Fig. 8.2 to Fig. 8.4 some of the contact force distributions that are involved in constructing and validating the hyperreduced model are schematically represented. These figures are composed as follows: the horizontal grid represents the projected FE surface mesh of the interacting flank of one of the active teeth of the selected gear. Elements that are indicated in grey refer to the

elements that are selected by the coarse contact detection algorithm (see Section 3.2.4). For these elements (and the associated FE nodes), a series of minimal distance problems is solved to detect contact between master elements and slave nodes. All FE nodes that are involved in a contact interaction are marked in black; the arrows that depart from them represent the associated normal contact forces. Slave nodes that are selected by the DEIM algorithm are marked in red, whereas the elected master elements are shown in red. Note that in the figures below, the contact force distributions corresponding to a single pass are shown, whereas the DEIM nodes and elements correspond to both passes, their maximal number being $2h$, if no overlap between nodes and/or elements occurs.

In Fig. 8.2a and Fig. 8.2b, the slave and master contact force distributions acting on the first active tooth of the driving (= slave) and driven (= master) gear are plotted at a particular angular configuration θ_j^1. Three alignment conditions are considered in this figure: one perfectly aligned and two misaligned configurations, resulting from a combination of angular misalignments and center distance variations. Each of these pair-wise contact force distributions (slave + master) is collected in a contact force snapshot matrix that is associated with that particular angular configuration.

Having assembled the snapshot matrices at all sampled angular configurations $\theta_j^1, j = 1, \dots, n_s$, $n_s - 1$ pairs of snapshot matrices corresponding to adjacent configurations θ_j^1 and θ_{j+1}^1 are combined and decomposed using the SVD to obtain the left-singular vectors and singular values of the combined snapshot matrix. The left-singular vectors corresponding to the three largest singular values are shown in Fig. 8.3 for the slave (a) and master (b) gear. By comparing Fig. 8.3a and Fig. 8.3b, one can qualitatively verify that the SVD indeed satisfies Newton's third law. Based on these singular vectors, and on the relative error tolerance of $\tau = 0.02$, the DEIM indices are computed by execution of Algorithm 8.1. These indices refer to slave nodes and (unassembled) master elements that need to be queried for approximating the nonlinear contact forces.

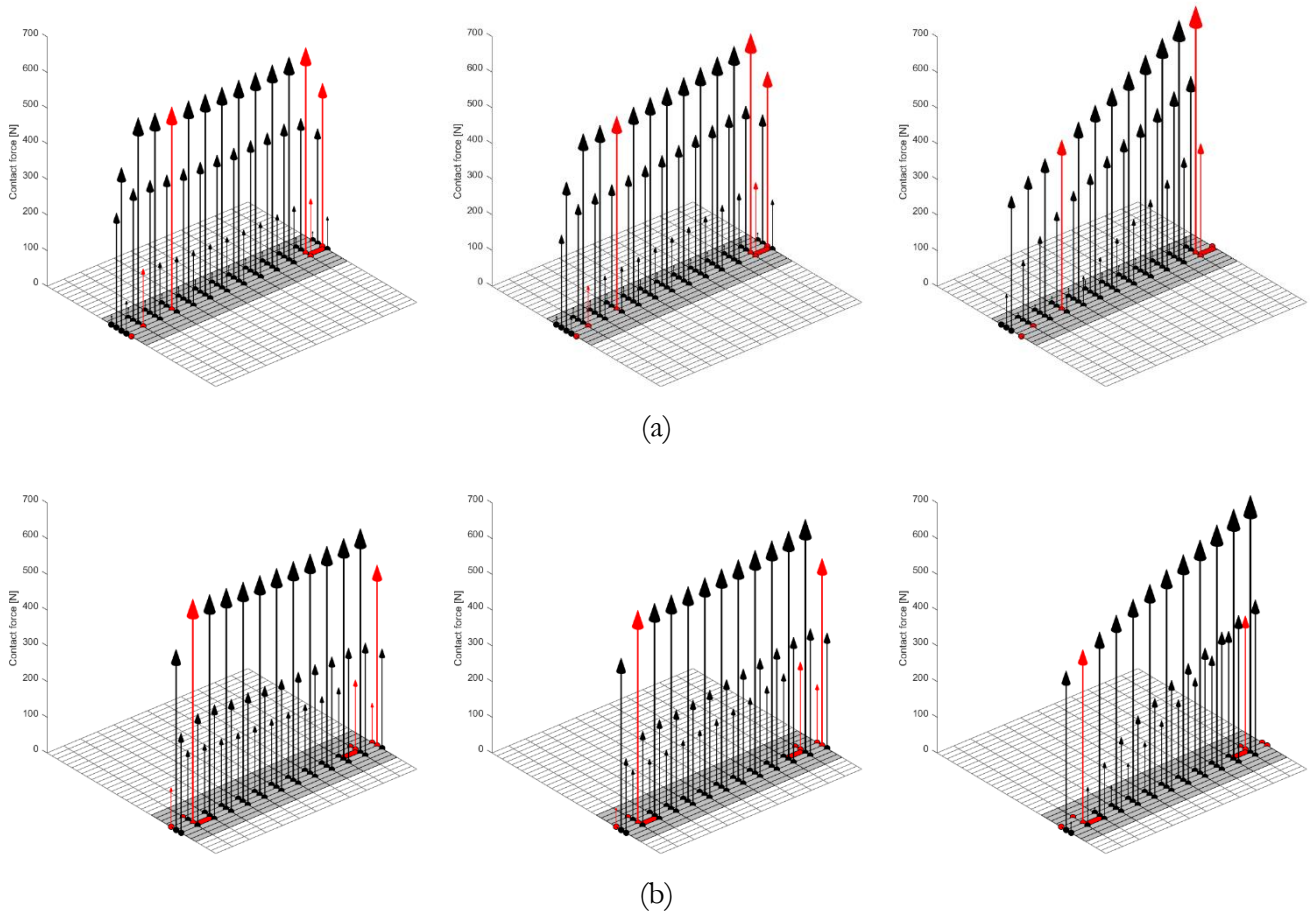


Fig. 8.2 Slave (a) and master (b) contact force distributions in perfectly aligned and misaligned conditions

The performance of the localDEIM-II approach (see Section 8.2.2, Eq. 8.2.9) is assessed by computing the static contact force distributions at an angular configuration θ^1 midway between θ_j^1 and θ_{j+1}^1 when a moderate sampling of $n_s = 20$ is used. The figures on the left of Fig. 8.4 show the unreduced contact force distributions on the first active tooth of the slave and master gear, respectively. The middle figures show the DEIM approximation obtained using the localDEIM-II approach, and the right figures show the error made by this approximation with respect to the unreduced solution. Note that the maximum relative error is well below the 2% error bound that was imposed on the DEIM algorithm. Furthermore, note that while the DEIM approximation of the unassembled nonlinear function is exact in the queried unassembled function entries, this not necessarily shows in the assembled version of the approximated function. Finally it should be noted that in addition to the high robustness of the localDEIM-II approach with regard to changes in angular configuration, the DEIM scheme exhibits excellent robustness with regard to changes in alignment and loading conditions.

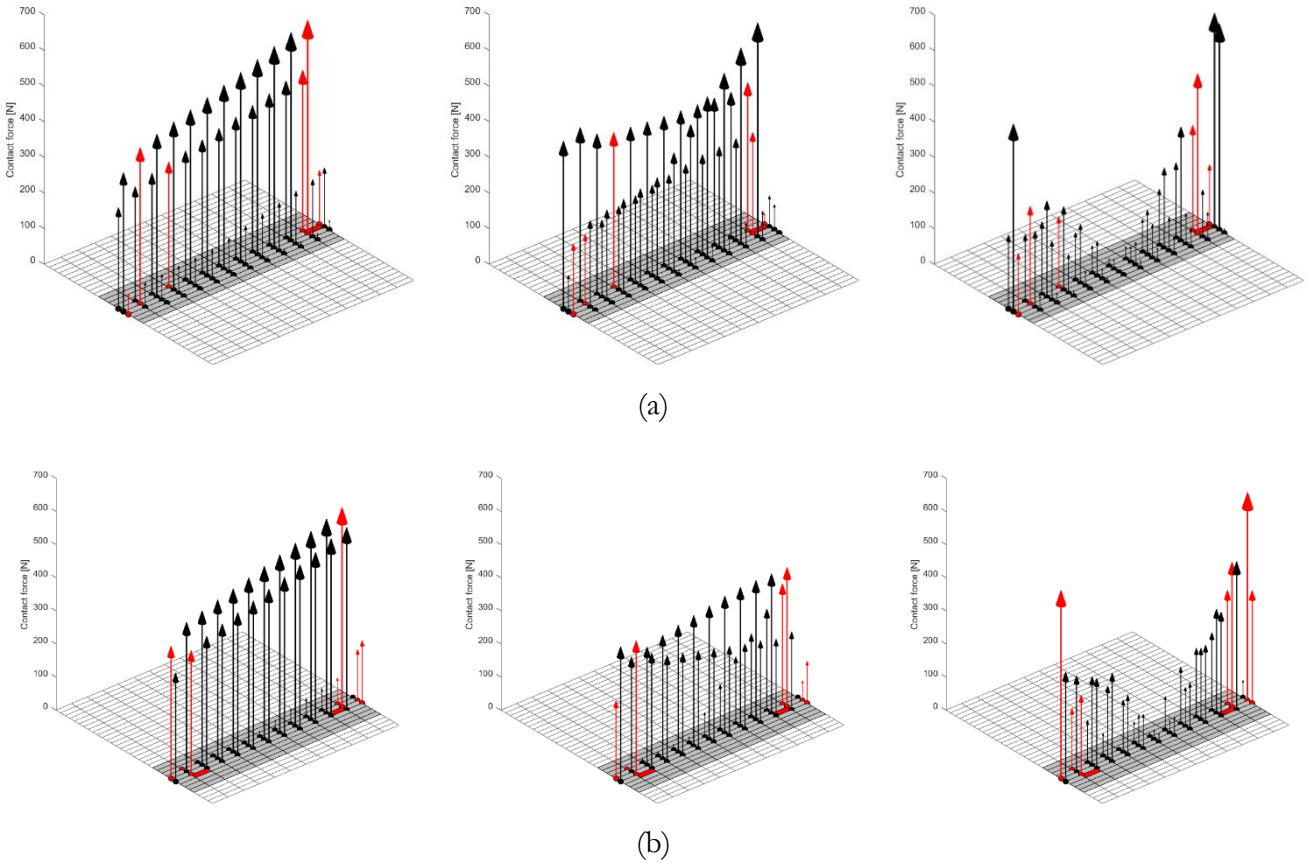


Fig. 8.3 Singular vectors of contact force snapshot matrix corresponding to the slave (a) and master (b) gear

The above exercise was repeated for both the localDEIM-I and localDEIM-II approaches for various number of samples, in casu $n_s = 10 - 50$ with steps of 10 and $n_s = 100$. Each time the hyperreduced models were trained with a relative error tolerance of 0.02 and with $\boldsymbol{f}_{ref}$ equal to the perfectly aligned contact force distribution at one (localDEIM-I) or both (localDEIM-II) of the adjacent angular configurations. Fig. 8.5 shows the evolution of the relative error of the DEIM approximation of the contact force distribution midway in the considered angular interval. It is readily apparent that the localDEIM-II approach outperforms the localDEIM-I approach when it comes to approximating a contact force distribution at an unsampled angular configuration.

Finally, in Fig. 8.6 the reduced meshes of a spur and helical gear as computed by the modified DEIM scheme are shown. Element faces that are indicated in gray refer to the elements that are retained by the coarse contact detection algorithm (see Section 3.2.4); nodes and elements that are indicated in red refer to the DOFs that are associated with the DEIM indices: the contact forces acting on these quantities need to be computed for forming the DEIM approximation of Eq. 8.2.7.

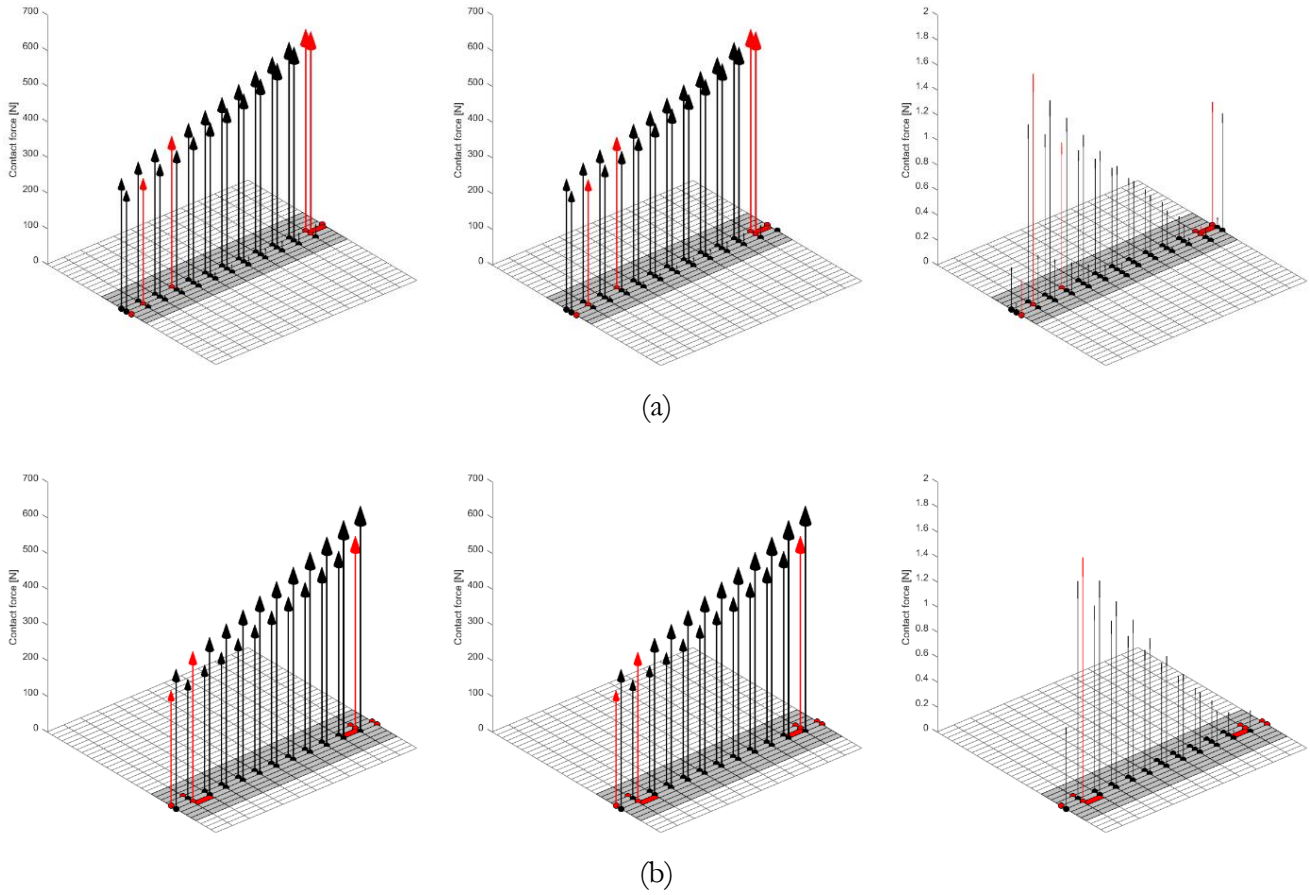


Fig. 8.4 Assessment of the localDEIM-II scheme for approximating the contact force distribution at an intermediate angle

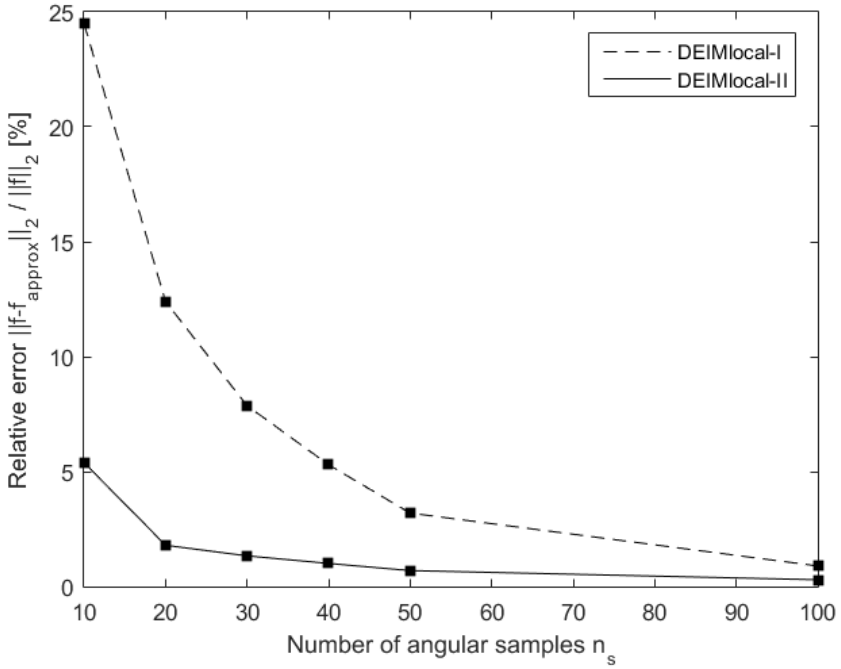
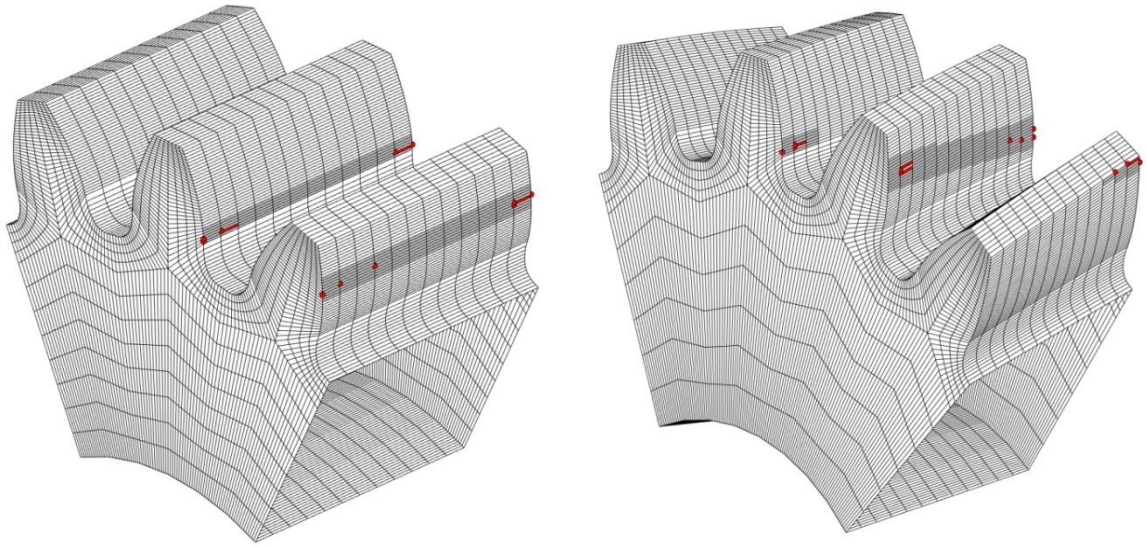


Fig. 8.5 Evolution of the approximation error of the localDEIM-I and localDEIM-II approaches as a function of number of angular samples



(a) (b)

Fig. 8.6 Reduced mesh in the modified DEIM scheme of a spur (a) and helical (b) gear

8.2.5 *Evaluation of the numerical complexity*

The efficiency of the modified DEIM scheme depends to a large extend on the efficiency of the selective computation of slave contact forces. While the DEIM indices indicate which nodal and/or elemental contact forces have to be computed on the tooth flank of the reference gear, it is not clear a priori which are the elements and/or nodes of the interacting tooth flank that are involved in this computation. An extensive search over the other flank's elements and/or nodes could jeopardize (although not nullify) the potential savings of the DEIM scheme. In this dissertation, instead, for each slave node and/or master element selected by the DEIM scheme, a list of candidate contact elements and/or nodes is constructed based on a predefined influence ellipse (defined in terms of the gear width, tooth height, and FE discretization) and sorted in descending order of proximity, so that the master-slave pair that is most likely to be in contact is checked first. In the examples shown in Fig. 8.6, the number of contact candidates for each slave node or master element varies between 4 and 8, where on average between 1.6 and 1.9 candidate pairs are checked. It should be noted, however, that such a confined search would lead to considerable savings even in the absence of hyperreduction[29].

The influence of the modified DEIM scheme on the number of FLOPs required to process a single tooth pair engagement can be assessed based on Table 8.1. As it turns out, the number of slave nodes and/or master elements selected by the error-based DEIM scheme is independent of the FE discretization. Instead, it is inversely proportional to the relative error tolerance and the

[29] The main benefit of hyperreduction in this regard is that it makes this bookkeeping task manageable. Moreover, given the unassembled formulation of the hyperreduction scheme, the selective evaluation of master element forces in a hyperreduced model is considerably less evolved than in the absence of hyperreduction.

angular distance between two sampled angular configurations. A conservative estimate of the number of FLOPs required to compute the selected entries of the full-order contact force vector is obtained by making the following assumptions: first, the number of minimal distance problems solved per selected entry is four (instead of the 1.6 or 1.9 reported above); secondly the selected entries all correspond to slave contact forces, as is it is more expensive to loop over four master elements keeping the slave node constant, than to loop over four slave nodes while keeping the master element constant. Using these assumptions and Table 8.1, the number of FLOPs required for the selective evaluation of the contact force vector is approximately $3000h$, where h is the predefined or error-based number of DEIM projection vectors and indices. For the example of Fig. 8.6, $h = 12$ and the number of FLOPs is 36 000. This is approximately two orders of magnitude smaller than the numbers reported on in Section 8.1.

There are additional computational savings in the selective projection of generalized coordinates to the physical domain; however in a worst-case estimation based on the aforementioned confined contact pairs search, these savings are limited. The cost of averaging the two contact passes and back-projecting the full-order contact forces remains the same as in the absence of hyperreduction. Finally, there is an additional cost associated with forming the DEIM approximation (Eq. 8.2.1) that requires approximately $hN_{fef}N_{few}$ FLOPs. An overview of the number of FLOPs involved in computing the generalized contact forces with and without hyperreduction is shown in Table 8.3. The example used in this table is the spur gear pair of Fig. 8.6, which has the values $N_{fef} = 35, N_{few} = 12, n_f = 100$ and $h = 12$. Notice that computing the vector of physical contact forces in the absence of hyperreduction occupies 95% of the CPU time, whereas in the hyperreduced model only 24% of the CPU time is spent on forming the contact force vector, whereas 76% is spent on subspace projections. Finally, the number of FLOPs involved in approximating the vector of generalized contact forces in the hyperreduced model (160.2 kFLOPs) is about 22.5 times smaller than in the absence of hyperreduction (3.7 MFLOPs). The actual performance of the DEIM scheme in a dynamic gear contact analysis will be assessed in Section 8.4.

	# FLOPs in theory		**# kFLOPs in example**	
Operation	**no hyperred.**	**hyperred.**	**no hyperred.**	**hyperred.**
Projection coordinates	$\approx 3n_f N_{fef} N_{few}$	$\approx n_f N_{fef} N_{few}$	126 (3%)	42 (25%)
Comp. of contact forces	$\approx 20 N_{fef}^2 N_{few}^2$	$\approx 3000h$	3528 (95%)	36 (21%)
Constr. of DEIM approx.	n.a.	$\approx hN_{fef}N_{few}$	n.a.	5 (3%)
Back-proj. contact forces	$\approx 2n_f N_{fef} N_{few}$	$\approx 2n_f N_{fef} N_{few}$	84 (2%)	84 (51%)
Total			3738 (100%)	166.2 (100%)

Table 8.3 Number of FLOPs in computing the generalized contact forces in the DEIM scheme

8.3 Hyperreduction using the Energy-Conservative Sampling and Weighting method

8.3.1 A local ECSW scheme for contact problems

The starting point of the ECSW method is the element-wise construction of the reduced-order nonlinear function $\boldsymbol{f}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) \in \mathbb{R}^r$, and its approximation by a weighted summation of a subset of elemental contributions, i.e.

$$\boldsymbol{f}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = \sum_{e=1}^{n_e} \boldsymbol{V}^{e^T} \boldsymbol{f}(\boldsymbol{V}^e \boldsymbol{q}, \boldsymbol{V}^e \dot{\boldsymbol{q}}, t) \approx \sum_{E_r} \xi^e \, \boldsymbol{V}^{e^T} \boldsymbol{f}(\boldsymbol{V}^e \boldsymbol{q}, \boldsymbol{V}^e \dot{\boldsymbol{q}}, t) \tag{8.3.1}$$

where $\boldsymbol{V}^e \in \mathbb{R}^{N_e \times r}$ is the restriction of the basis matrix $\boldsymbol{V} \in \mathbb{R}^{N \times r}$ to the rows associated with element *e*, n_e is the number of elements involved in evaluating the full-order nonlinear function $\boldsymbol{f}$, N_e is the number of DOFs associated with the element *e*, E_r is the set of element indices of the so-called *reduced mesh* (i.e. these are the elements whose contribution to $\boldsymbol{f}_r$ is weighted to obtain an accurate approximation of $\boldsymbol{f}_r$) and ξ^e is the weighting factor associated with element *e*. The reduced mesh and element weighting factors are obtained by finding a sufficiently accurate solution of a sparse Non-Negative Least Squares (NNLS) problem, as discussed in Section 2.3.4.2 and outlined in Algorithm 2.3.

In view of the distinction between *nodal* slave contact forces and *elemental* master contact forces (see Section 8.1), it makes sense to formulate the fundamental ECSW formula (Eq. 8.3.1) in terms of nodal instead of elemental contributions, i.e.

$$\boldsymbol{f}_r(\boldsymbol{q}, \dot{\boldsymbol{q}}, t) = \sum_{j=1}^{n_n} \boldsymbol{V}^{j^T} \boldsymbol{f}(\boldsymbol{V}^j \boldsymbol{q}, \boldsymbol{V}^j \dot{\boldsymbol{q}}, t) \approx \sum_{\mathcal{E}_r} \xi^j \, \boldsymbol{V}^{j^T} \boldsymbol{f}(\boldsymbol{V}^j \boldsymbol{q}, \boldsymbol{V}^j \dot{\boldsymbol{q}}, t) \tag{8.3.2}$$

where $\boldsymbol{V}^j \in \mathbb{R}^{N_n \times r}$ is the restriction of the basis matrix $\boldsymbol{V} \in \mathbb{R}^{N \times r}$ to the rows associated with node *j*, n_n is the number of nodes involved in evaluating the full-order nonlinear function $\boldsymbol{f}$, N_n is the number of DOFs associated with the node *j*, $\mathcal{E}_r$ is the set of node indices in the reduced mesh and ξ^j is the weighting factor associated with node *j*. The procedure to determine the reduced mesh $\mathcal{E}_r$ and the associated node weighting factors ξ^j is the same as for the elemental ECSW formulation, except that sparse NNLS problem (Algorithm 2.3) is solved using unassembled snapshots of the nodal instead of elemental contributions.

The application of the ECSW method to the dynamic gear contact problem proceeds as follows. Referring to Eq. 8.1.6 and Eq. 8.1.8, the generalized contact force vector can be written as

$$\boldsymbol{Q}_c = \begin{Bmatrix} \boldsymbol{Q}_c^1 \\ \boldsymbol{Q}_c^2 \end{Bmatrix} = \sum_{t=1}^{n_{tic}} \mathbb{V}^{t^T} \boldsymbol{f}_c^t = \underbrace{\frac{1}{2}\sum_{t=1}^{n_{tic}} \mathbb{V}^{t,sm^T} \begin{Bmatrix} \boldsymbol{f}_c^{t1,s} \\ \boldsymbol{f}_c^{t2,m} \end{Bmatrix}}_{1^{st}\ pass} + \underbrace{\frac{1}{2}\sum_{t=1}^{n_{tic}} \mathbb{V}^{t,ms^T} \begin{Bmatrix} \boldsymbol{f}_c^{t1,m} \\ \boldsymbol{f}_c^{t2,s} \end{Bmatrix}}_{2^{nd}\ pass} \tag{8.3.3}$$

where n_{tic} is the number of Teeth In Contact and $\mathbb{V}^{t,sm}$ is the projection matrix associated with the contact force vector $\boldsymbol{f}_c^{t,ms}$ (see Eq. 8.1.9). Decomposing the above formula in terms of elemental contributions yields

$$\boldsymbol{Q}_c = \underbrace{\frac{1}{2}\sum_{t=1}^{n_{tic}}\sum_{e=1}^{n_{e,t}} [\mathbb{V}^{t1e,s^T} \quad \mathbb{V}^{t2e,m^T}] \begin{Bmatrix} \boldsymbol{f}_c^{t1e,s} \\ \boldsymbol{f}_c^{t2e,m} \end{Bmatrix}}_{1^{st}\ pass} + \underbrace{\frac{1}{2}\sum_{t=1}^{n_{tic}}\sum_{e=1}^{n_{e,t}} [\mathbb{V}^{t1e,m^T} \quad \mathbb{V}^{t2e,s^T}] \begin{Bmatrix} \boldsymbol{f}_c^{t1e,m} \\ \boldsymbol{f}_c^{t2e,s} \end{Bmatrix}}_{2^{nd}\ pass} \tag{8.3.4}$$

where $n_{e,t}$ is the number of elements per tooth that contribute to the vector of contact forces. Note that the decomposition of the contact force in elemental contributions strictly only makes sense for the vectors of master contact forces, which are indeed elemental forces. Therefore, the above formula should be read as follows: in the first pass, where the wheel ($i = 2$) is the master gear, $\boldsymbol{f}_c^{2te,m}$ represents the unassembled elemental contribution to $\boldsymbol{f}_c^{2t,m}$ due to the *e*-th element of the considered tooth, and $\boldsymbol{f}_c^{1te,s}$ represents the nodal contributions to $\boldsymbol{f}_c^{1t,s}$ corresponding to the slave nodes that are in contact with the *e*-th element of the master gear. Using this convention, it is guaranteed that Newton's third law is preserved on the elemental level. An equivalent formula in terms of nodal contributions is obtained by replacing the superscripts '*e*' by '*n*' in the above formula, i.e.

$$\boldsymbol{Q}_c = \underbrace{\frac{1}{2}\sum_{t=1}^{n_{tic}}\sum_{n=1}^{n_{n,t}} [\mathbb{V}^{t1n,s^T} \quad \mathbb{V}^{t2n,m^T}] \begin{Bmatrix} \boldsymbol{f}_c^{t1n,s} \\ \boldsymbol{f}_c^{t2n,m} \end{Bmatrix}}_{1^{st}\ pass} + \underbrace{\frac{1}{2}\sum_{t=1}^{n_{tic}}\sum_{n=1}^{n_{n,t}} [\mathbb{V}^{t1n,m^T} \quad \mathbb{V}^{t2n,s^T}] \begin{Bmatrix} \boldsymbol{f}_c^{1tn,m} \\ \boldsymbol{f}_c^{2tn,s} \end{Bmatrix}}_{2^{nd}\ pass} \tag{8.3.5}$$

where this time the slave gear is the reference gear, and $\boldsymbol{f}_c^{2tn,m}$ represents the elemental contributions to $\boldsymbol{f}_c^{2t,m}$ due to the element that is penetrated by the *n*-th node of the slave gear.

In the proposed ECSW scheme, each pass is approximated separately whereas the contributions of the different teeth in contact are approximated altogether. The ECSW approximations of the first pass based on the elemental and nodal formulations can respectively be written as (dropping the '*sm*' superscripts)

$$\boldsymbol{Q}_c = \sum_{e=1}^{n_e} \boldsymbol{Q}_c^e = \sum_{e=1}^{n_e} [\mathbb{V}^{1e^T} \quad \mathbb{V}^{2e^T}] \begin{Bmatrix} \boldsymbol{f}_c^{1e} \\ \boldsymbol{f}_c^{2e} \end{Bmatrix} \approx \sum_{E_r} \xi^e [\mathbb{V}^{1e^T} \quad \mathbb{V}^{2e^T}] \begin{Bmatrix} \boldsymbol{f}_c^{1se} \\ \boldsymbol{f}_c^{2me} \end{Bmatrix} \tag{8.3.6}$$

where the first two summations are carried out over all $n_e = n_{tic} \cdot n_{e,t}$ elements that are in contact (across all active teeth) and the third summation is carried out over all elements $e \in E_r$ in the reduced mesh. In the nodal formulation, this equation becomes

$$\boldsymbol{Q}_c = \sum_{e=1}^{n_n} \boldsymbol{Q}_c^n = \sum_{n=1}^{n_n} [\mathbb{V}^{1n^T} \quad \mathbb{V}^{2n^T}] \begin{Bmatrix} \boldsymbol{f}_c^{1n} \\ \boldsymbol{f}_c^{2n} \end{Bmatrix} \approx \sum_{\mathcal{E}_r} \xi^n [\mathbb{V}^{1n^T} \quad \mathbb{V}^{2n^T}] \begin{Bmatrix} \boldsymbol{f}_c^{1n} \\ \boldsymbol{f}_c^{2n} \end{Bmatrix} \tag{8.3.7}$$

where the first two summations are carried out over all $n_n = n_{tic} \cdot n_{n,t}$ nodes and the third summation is carrier out over all nodes $n \in \mathcal{E}_r$ in the reduced mesh.

The reduced meshes E_r and $\mathcal{E}_r$ are determined by applying algorithm 2.3 (see Section 2.3.4.2) to the unassembled snapshot matrix $\boldsymbol{G}$ and the corresponding assembled snapshot vector $\boldsymbol{b} =$ `sum(`$\boldsymbol{G}$`, 'columns')`. Analogous to the DEIM scheme outlined in Section 8.2, the snapshot matrix consists of (generalized and unassembled) contact force distributions computed at various gear pair alignment conditions. As in the localDEIM-II approach, two adjacent angular configurations are considered at the time, thus yielding a localECSW-II approach that is based on the discontinuous switching of ECSW sampling indices and weights based on the angular configuration of the gear pair. The snapshot matrix $\boldsymbol{G}$ is thus composed as follows:

$$\boldsymbol{G}^T = \begin{bmatrix} \boldsymbol{G}_j^T & \boldsymbol{G}_{j+1}^T \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \boldsymbol{G}_{j,1}^T & \cdots & \boldsymbol{G}_{j,n_m}^T \end{bmatrix} & \begin{bmatrix} \boldsymbol{G}_{j+1,1}^T & \cdots & \boldsymbol{G}_{j+1,n_m}^T \end{bmatrix} \end{bmatrix} \tag{8.3.8}$$

where $\boldsymbol{G}_j$ and $\boldsymbol{G}_{j+1}$ are the snapshot matrices corresponding to the adjacent angular configurations θ_j^1 and θ_{j+1}^1, and where $\boldsymbol{G}_{j,k}$ is the snapshot matrix corresponding to the *j*-th angular configuration and the *k*-th considered alignment condition. In the elemental and nodal ECSW formulations, the latter is respectively defined as

$$\boldsymbol{G}_{j,k} = \begin{bmatrix} \left(\boldsymbol{Q}_c^1\right)_{j,k} & \cdots & \left(\boldsymbol{Q}_c^e\right)_{j,k} & \cdots & \left(\boldsymbol{Q}_c^{n_e}\right)_{j,k} \end{bmatrix} \tag{8.3.9a}$$

$$\boldsymbol{G}_{j,k} = \begin{bmatrix} \left(\boldsymbol{Q}_c^1\right)_{j,k} & \cdots & \left(\boldsymbol{Q}_c^n\right)_{j,k} & \cdots & \left(\boldsymbol{Q}_c^{n_n}\right)_{j,k} \end{bmatrix} \tag{8.3.9b}$$

The number of rows of the unassembled snapshot matrix is thus equal to $2n_q n_m$ whereas the number of columns is equal to n_e or n_n, depending on whether the elemental or nodal approach is used. In the next two sections, the robustness and efficiency of the localECSW scheme is investigated by applying Eq. 8.3.6-8.3.7 to an example gear pair.

8.3.2 A conceptual example

The ECSW hyperreduction method directly approximates the vector of generalized contact forces $\boldsymbol{Q}_c = \mathbb{V}^T \boldsymbol{f}_c$. Due to the singularity of the projector $\mathbb{V}\mathbb{V}^T$, a direct comparison of non-hyperreduced

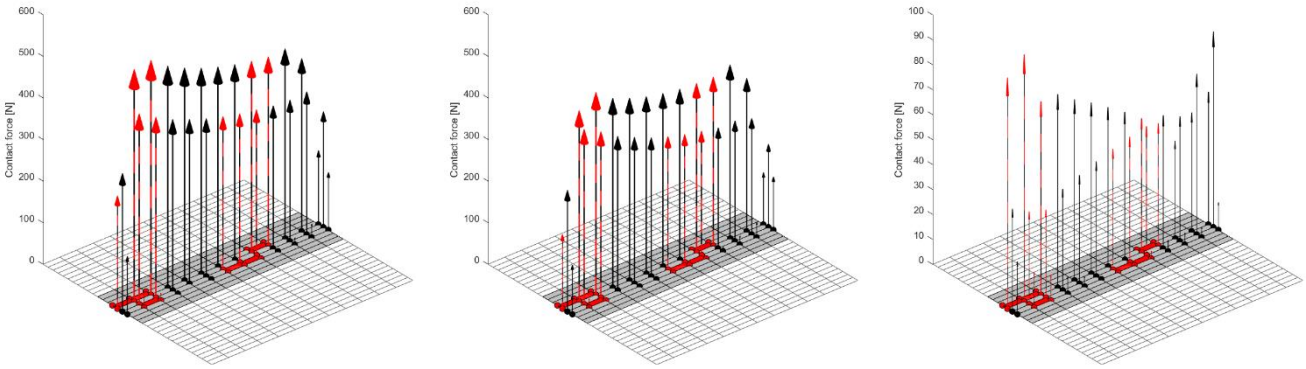
and hyperreduced contact force distributions as in Section 8.2.4 is not possible. In order to still enable a physical interpretation of the approximation error, the generalized contact forces $\boldsymbol{Q}_c^i$ are applied as external forces to the respective gear ROMs while fixing their rigid body motion; the obtained reduced-order displacement field ($\boldsymbol{q}_f^i$) is then projected to the physical domain ($\boldsymbol{u}_f^i = \boldsymbol{V}^i\boldsymbol{q}_f^i$), and when multiplied with the full-order stiffness matrix $\boldsymbol{K}_{FE}^i$ yields the vector of physical contact forces $\boldsymbol{f}_c^i$, i.e.

$$\boldsymbol{f}_c^i = \boldsymbol{K}_{FE}^i\boldsymbol{V}^i\left(\boldsymbol{K}_{ff}^i\right)^{-1}\boldsymbol{Q}_c^i \tag{8.3.10}$$

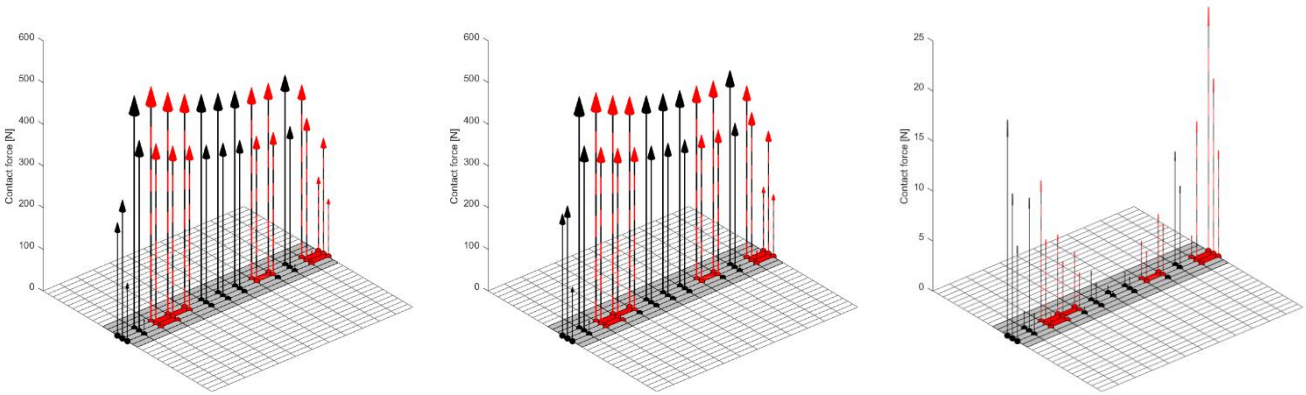
where $\boldsymbol{K}_{ff}^i$ is the reduced-order stiffness matrix of the *i*-th ROM, i.e. $\boldsymbol{K}_{ff}^i = \boldsymbol{V}^{i^T}\boldsymbol{K}_{FE}^i\boldsymbol{V}^i$. In the figures below, all contact force distributions, including the non-hyperreduced distributions, are passed through the above formula.

In Fig. 8.7, the numerical experiment of Fig. 8.4 is repeated for the elemental ECSW approach: the hyperreduced model is trained using the contact force snapshots corresponding to the adjacent angular configurations θ_j^1 and θ_{j+1}^1, and the error of the approximation is measured at an intermediate configuration θ^1 midway between θ_j^1 and θ_{j+1}^1. Analogous to Fig. 8.4, the figures on the left, in the middle and on the right respectively show the reference contact force distribution on the master gear, its ECSW approximation, and the absolute error between the two. The arrows associated with the corner nodes of the selected elements are dashed in black and red to indicate that these nodal forces only need to be evaluated for the *unassembled* element that is marked in red, which is considerably less complicated than computing the assembled element forces.

The number of sampled angular gear pair configurations in Fig. 8.7a and Fig. 8.7b is $n_\theta = 20$ and $n_\theta = 50$. Despite the good robustness of the elemental ECSW scheme with regard to load and alignment variations, it is readily clear from Fig. 8.7a that the method is not very robust with respect to changes in angular configuration. For that reason, the relative error tolerance is reduced to $\tau = 0.01$ and the number of angular samples is increased to 50 in Fig. 8.7b. While increasing the number of angular samples increases both the offline CPU cost and the online memory consumption, these are negligable in view of the overall CPU and memory costs. After all, the cost of training the hyperreduced model can be significantly reduced by using the ROM instead of the FOM, and the storage requirements for the ECSW method are very small: no more than a sparse $(n_q \times 1)$ vector of weighting factors per angular sample (i.e. a few hundreds of bytes at most).



(a) $n_\theta = 20$

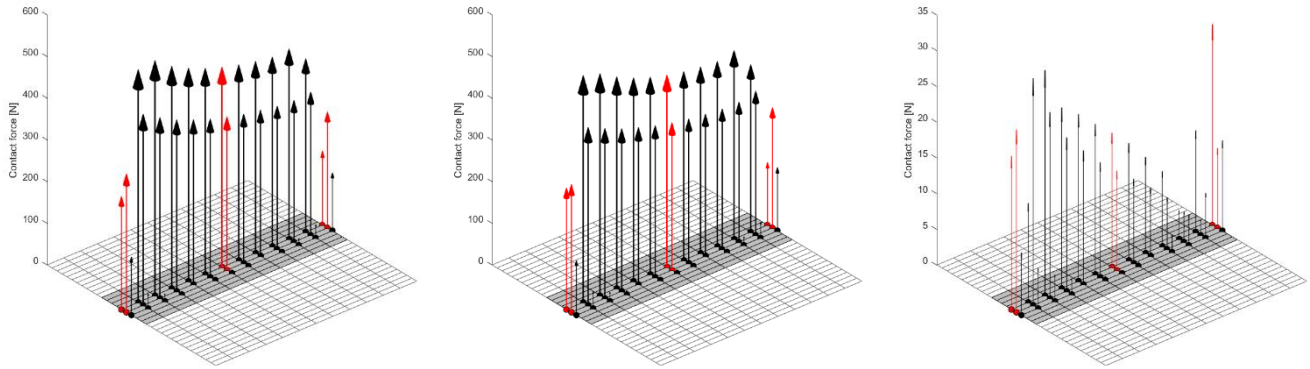


(b) $n_\theta = 50$

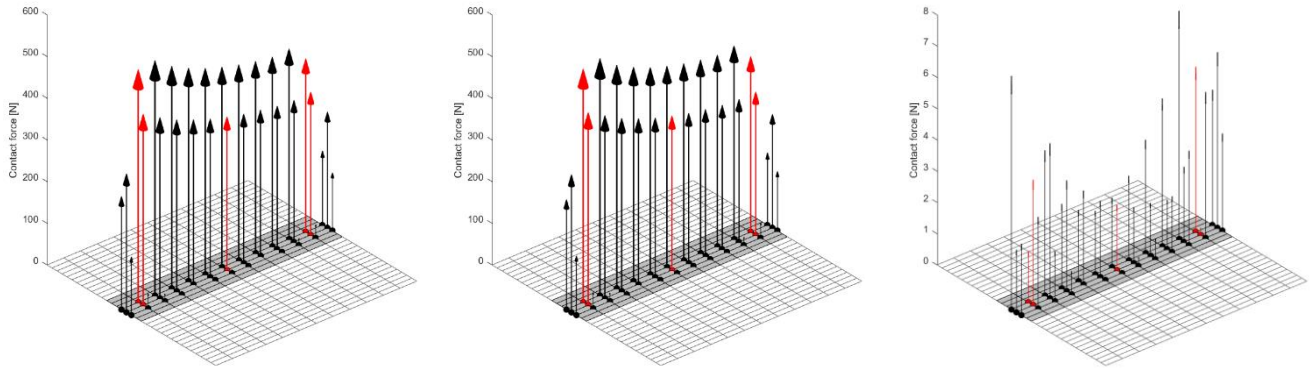
Fig. 8.7 Assessment of the elemental ECSW scheme for approximating the contact force distribution at an intermediate angle

The limited robustness of the ECSW method to variations in angular configuration is also observed in the nodal variant, albeit to a lesser extent. In Fig. 8.8, the numerical experiment of Fig. 8.7 is repeated for the nodal ECSW approach, using the same relative error tolerance and number of angular samples. The higher robustness of the nodal ECSW approach is also clear in Fig. 8.9, where the evolution of the relative error of the generalized contact forces at the intermediate angular configuration is shown in function of the number of angular samples. Overall, both ECSW approaches require a higher angular sampling than the DEIM approach outlined in Section 8.2.

Finally, in Fig. 8.10 and Fig. 8.11 the reduced meshes of a spur and helical gear as computed by the elemental and nodal ECSW schemes are shown for a relative error tolerance of $\tau = 0.01$ and an angular sampling of $n_\theta = 50$.



(a) $n_\theta = 20$



(b) $n_\theta = 50$

Fig. 8.8 Assessment of the nodal ECSW scheme for approximating the contact force distribution at an intermediate angle

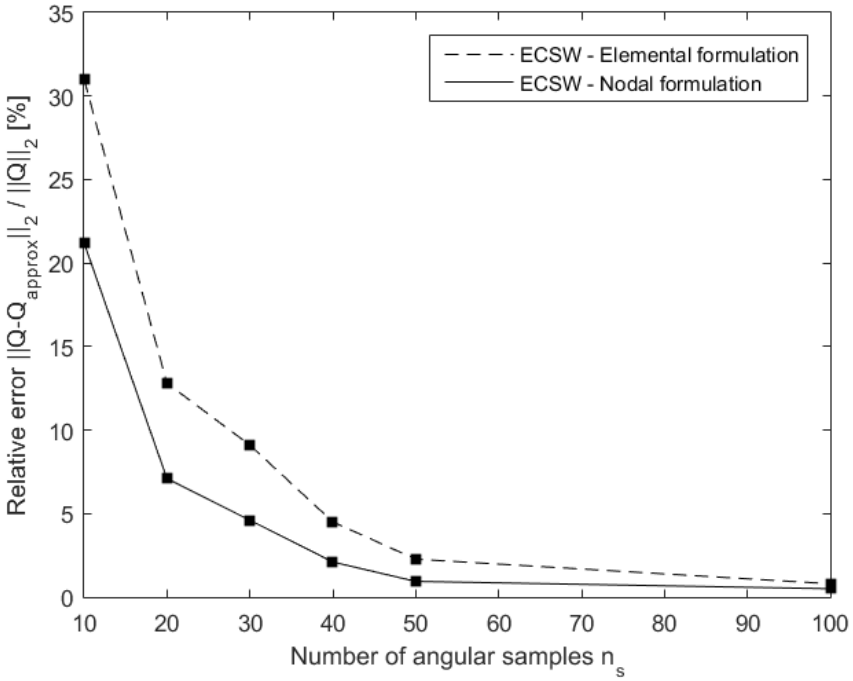
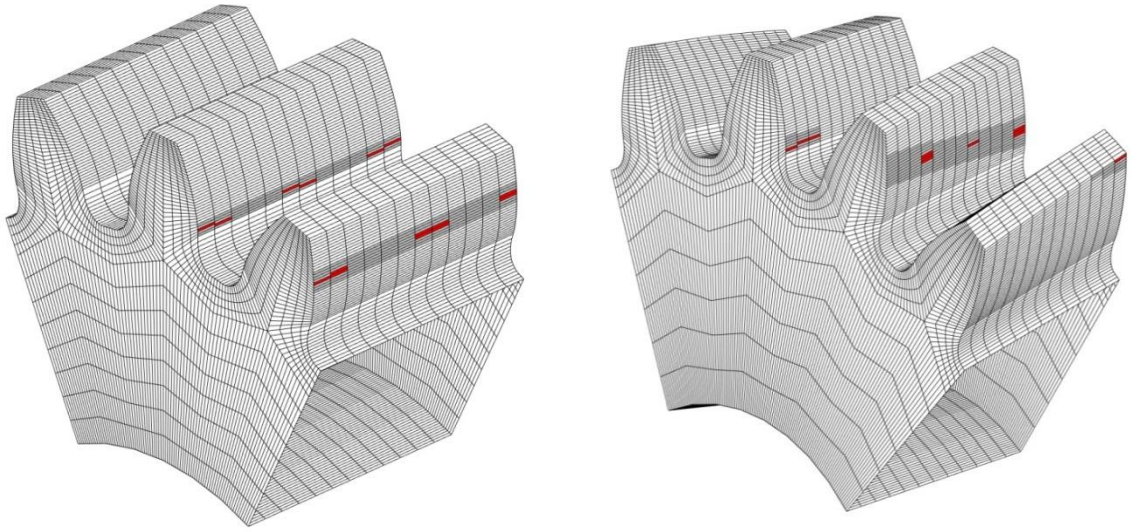
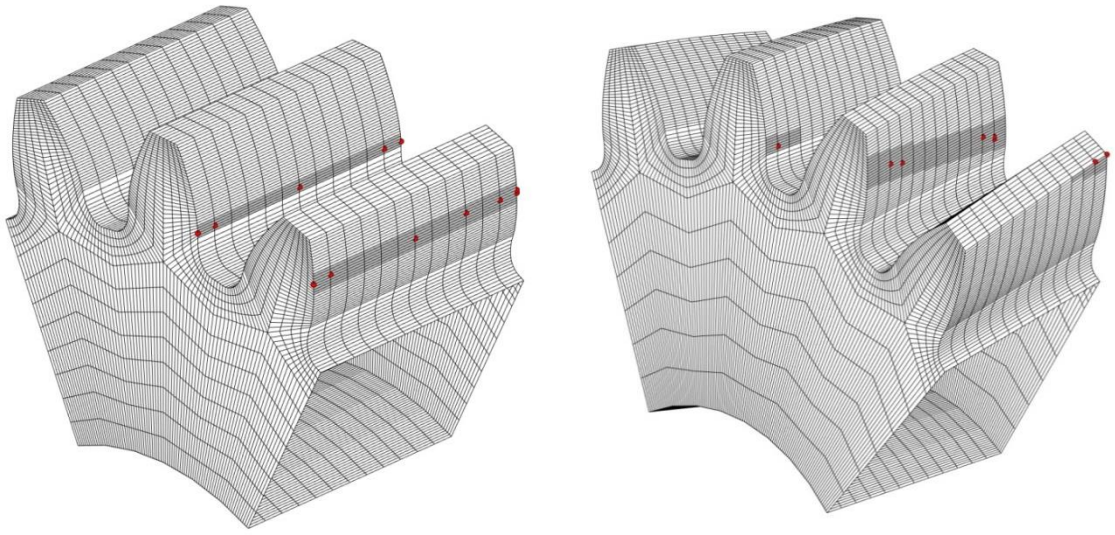


Fig. 8.9 Evolution of the approximation error of the ECSW approaches as a function of number of angular samples



(a) (b)

Fig. 8.10 Reduced mesh in the elemental ECSW scheme of a spur (a) and helical (b) gear



(a) (b)

Fig. 8.11 Reduced mesh in the nodal ECSW scheme of a spur (a) and helical (b) gear

8.3.3 Evaluation of the numerical complexity

The computational complexity of the ECSW algorithm is measured in Table 8.4 by counting the number of FLOPs required to form the projections, evaluate the selected entries of the contact force vector, and form the hyperreduction approximation. The example used in Table 8.4 is the spur gear pair of Fig. 8.10 (hyperreduced using the nodal ECSW approach), which has the values $N_{fef} = 35, N_{few} = 12, n_f = 100$ and $h = 11$. Contrary to the DEIM scheme, the ECSW approximation is constructed *after* back-projecting the contact forces, thus permitting a selective back-projection of generalized contact forces. Thus results in a reduction of the back-projection cost of about 75% with respect to the DEIM scheme, making the overall ECSW scheme 40%

more efficient than the DEIM scheme and about 35 times more efficient than the non-hyperreduced ROM.

	# FLOPs in theory		# kFLOPs in example	
Operation	**no hyperred.**	**hyperred.**	**no hyperred.**	**hyperred.**
Projection coordinates	$\approx 3n_f N_{fef} N_{few}$	$\approx n_f N_{fef} N_{few}$	126 (3%)	42 (43%)
Comp. of contact forces	$\approx 20 N_{fef}^2 N_{few}^2$	$\approx 3000h$	3528 (95%)	33 (33%)
Back-proj. contact forces	$\approx 2n_f N_{fef} N_{few}$	$\approx 20hn_f$	84 (2%)	22 (22%)
Constr. of ECSW approx.	n.a.	$\approx 2hn_f$	n.a.	2.2 (2%)
Total			3738 (100%)	99.2 (100%)

Table 8.4 Number of FLOPs in computing the generalized contact forces in the ECSW scheme

8.4 Numerical validation

In order to assess the online performance of the modified DEIM and ECSW hyperreduction schemes, the misaligned spur gear contact problem of Section 6.1.5 is reconsidered. The simulation parameters and main characteristics of the gear pair's FE representation are summarized in Table 4.3. The ROM of the gear pair is constructed using the S-MICS approach (see Section 6.1.3) with the parameters of Table 6.4. According to the FLOP counts of Table 4.2 roughly 90% of the CPU time is spent on computing the contact forces in the spur gear contact problem of Table 4.3; hence the maximum achievable speed-up factor for this problem is 10. The DEIM and ECSW hyperreduced models are trained based on the MICS-based ROM, as discussed in Section 8.2.2 and Section 8.3.1. An overview of the main characteristics of these models is given in Table 8.5. Owing to the limited robustness of the ECSW scheme with regard to changes in contact force location, the number of angular samples used to train the ECSW-based ROM was set to 50, whereas in the DEIM scheme 20 angular samples are sufficient. Note however from the table below that the cost of training the hyperreduced models is negligible once the ICS-based ROM is constructed. The EOMs of the gear pair are solved twice: first using the fixed-increment Central Difference (CD) method that was used in Section 6.1.5; then using the optimized Newmark scheme that was investigated in Section 7.4.1.2. As compared to the non-hyperreduced ROM, computational speed-up factors between 7 and 8 are observed for the DEIM- and ECSW-based ROMs. Higher computational savings can be expected when considering larger FE models (e.g. for the spur gear pair of Fig. 8.10 the maximum achievable speed-up factor is 50).

Parameters	No hyper.	DEIM	ECSW
Model-order reduction approach	S-MICS	S-MICS	S-MICS
Hyperreduction approach	no	DEIM	ECSW
Number of angular samples in hyperreduction	n.a.	20	50
Number of DOFs per gear	59	59	59
Number of elements / nodes in reduced mesh	n.a.	9	11
Single core pre-processing CPU time	1877 s	1954 s	2118 s
Single core processing CPU time (CD)	7345 s	1009 s	941 s
Single core processing CPU time (NMK)	318 s	63 s	57 s
Speed-up factor w.r.t. No hyper. (CD)	1	7.3	7.8

Table 8.5 Parameters and CPU times of the dynamic misaligned spur gear pair simulations

The excellent relative accuracy of the hyperreduced models is demonstrated in Fig. 8.12 to Fig. 8.14. In Fig. 8.12 the predicted DTE of the gear pair is shown, with relative errors well below 5% except at the transition points, where small amplitude and fazing differences can be observed. The same holds for the normal contact force at the 21th teeth of the driving gear, which is shown in Fig. 8.13. Finally, analogous to Fig. 6.9, Fig. 8.14 shows the predicted stress distributions in the active teeth at t = 5 ms. The excellent agreement between the different stress fields is partly explained by the fact that the underlying ROMs all share the same modal displacement shapes.

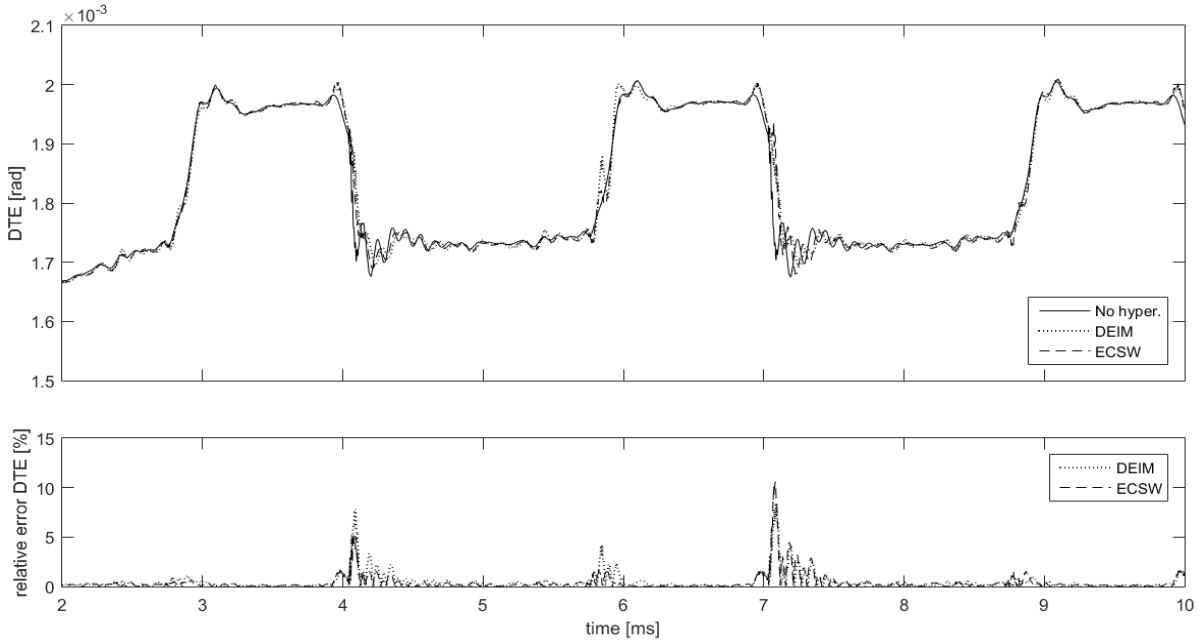


Fig. 8.12 Comparison of the DTE predictions of a misaligned spur gear pair by the hyperreduced models

As a last comparison and a final note in this dissertation, consider the reported CPU times for the CMS-based reference model in Table 6.4. Conducting a 10 ms misaligned gear contact analysis using this model and a fixed-increment central difference integrator took 463527 s of single-core CPU time. Having reduced the number of DOFs, optimized the numerical integrator, and reduced the number of FLOPs in the contact force evaluation, that same 10 ms simulation now takes 57 s (see Table 8.5). That is a speed-up of 8132 times.

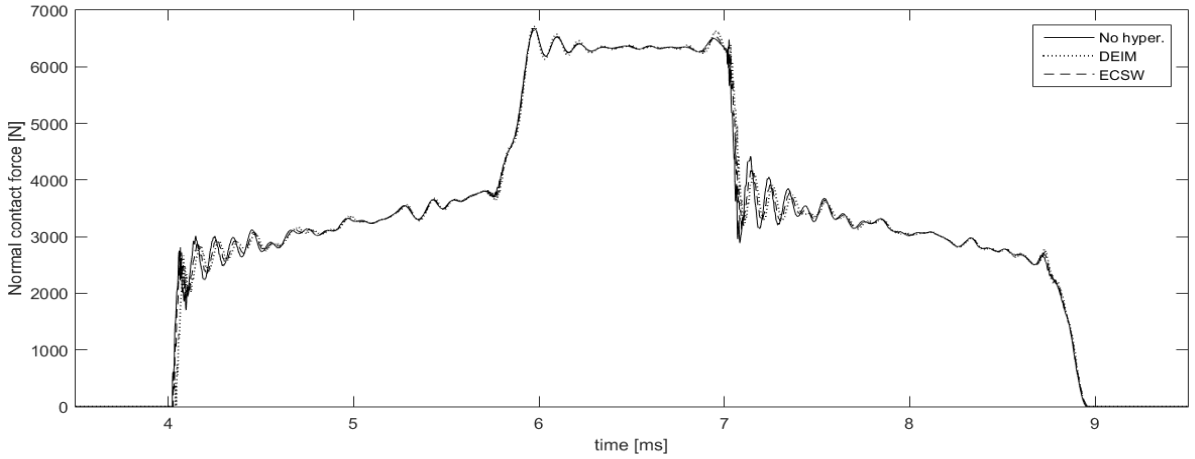
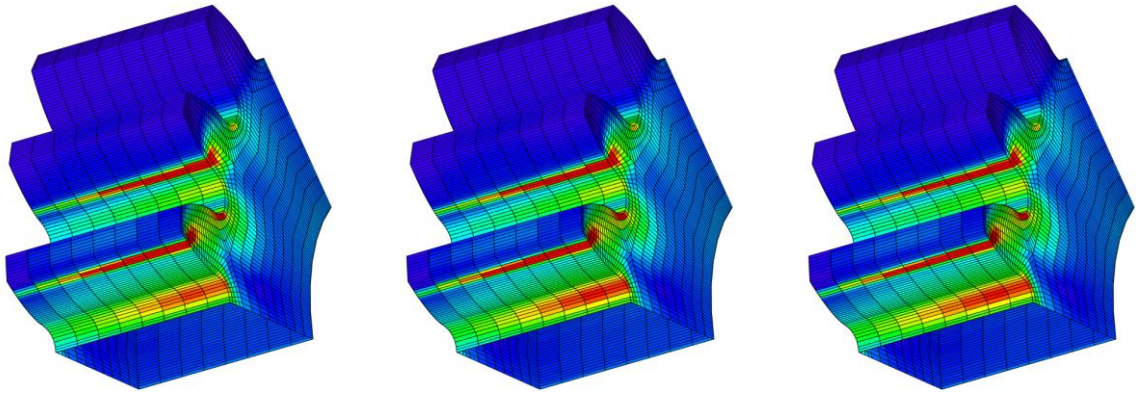


Fig. 8.13 Comparison of the contact force predictions of a misaligned spur gear pair



(a) No hyperreduction (b) DEIM (c) ECSW

Fig. 8.14 Comparison of the predicted Von Mises stresses at t = 5 ms in a misaligned spur gear pair

8.5 Summary and conclusion

In this chapter, the last of the three computational bottlenecks of finite-element based contact problems was dealt with: the computational complexity of the nonlinear contact forces. Having reduced the number of degrees of freedom and optimized the numerical integration scheme, up to 98% of the simulation time is spent on computing the contact forces. Following a comprehensive analysis of the complexity of the full-order and generalized contact forces, a second level approximation referred to as hyperreduction was applied to the dynamic gear contact

problem. Two hyperreduction schemes were developed based on the Discrete Empirical Interpolation Method (DEIM) and the Energy-Conserving Sampling and Weighting (ECSW) method, respectively, with particular focus on two main challenges: the development of effective hyperreduction schemes in the context of parametric model reduction, and the preservation of Newton's third law of motion. The two hyperreduction schemes were developed using a number of conceptual examples and their numerical efficiency was assessed by solving the dynamic misaligned gear contact problems that were introduced in Chapter 6. While the DEIM scheme proved to be more robust to changes in contact force locations, the ECSW scheme results in higher computational savings. Overall, both the DEIM and ECSW schemes were shown to reduce the CPU cost of the non-hyperreduced contact problem with roughly one order of magnitude.

Chapter 9

Conclusions and prospects for future research

The research presented in this dissertation is concerned with the efficient solution of dynamic contact problems by means of distributed-parameter methods. Such problems are encountered in the computer-aided design of mechanical systems, where contact interactions are ubiquitous and often vital to the proper functioning of the system. With weight reduction and increased operating frequencies among the primary objectives of present-day engineering, a dynamic description of these problems becomes increasingly important.

Distributed-parameter methods such as the Finite Element Method (FEM) are often combined with the Flexible Multibody Dynamics (FMBD) approach to construct mathematical models of the interacting components when their geometries are complex and/or deformable. These models however come at a high computational cost due to three main reasons: the high number of Degrees Of Freedom (DOFs) involved, the large number of time increments needed to numerically solve the time-dependent equilibrium equations, and finally the high number of Floating Point Operations (FLOPs) required to evaluate the nonlinear contact forces.

In this dissertation an attempt was made to mitigate the computational burden of dynamic contact problems by tackling all three of these issues. While the methods and procedures discussed in this dissertation apply to a broad class of contact problems, the dynamic gear contact problem was selected as a common theme throughout this text. In this final chapter, a summary of the main contributions and conclusions of the current work is presented (Section 9.1). Despite the significant strides that have been made in lowering the cost of dynamic contact simulations, the quest for computational efficiency in numerical modelling continues: some prospects for future research are discussed in Section 9.2.

9.1 Conclusions

9.1.1 A novel model-order reduction scheme for dynamic contact problems

Projection-based Model Order Reduction (MOR) is a powerful tool that enables the systematic generation of cost-efficient representations of large-scale dynamical models by projecting the system's state vector onto a lower-dimensional subspace. Traditional MOR methods such as the

Component Mode Synthesis (CMS) approach, the Proper Orthogonal Decomposition (POD) method, balanced truncation and Krylov-subspace methods have repeatedly proven their worth in a great variety of applications. Yet when faced with a dynamic contact problem – especially when sliding contact interactions are at play – their effectiveness rapidly degrades due to the variable topology of the interacting components and the associated multitude of potential interface DOFs.

The first and central contribution of this dissertation is the development of a novel MOR method that is specifically tailored to the solution of Finite-Element (FE) based contact problems in flexible multibody dynamics. The method belongs to the class of projection-based MOR methods and can be seen as a parametric variant of the CMS approach. By judiciously choosing the ingredients of the reduction basis, the method was shown to reduce the number of DOFs in the Full-Order Model (FOM) with several orders of magnitude, from as much as one million to a few tens or hundreds. The reduction basis is composed of a truncated set of eigenmodes augmented with a configuration-dependent set of so-called contact shapes. These contact shapes are the component-level deformation patterns of the mechanism obtained from a series of a priori static contact analyses. The computation of these shapes during the offline phase of the simulation is accelerated through the use of an intermediate ROM, an analytical contact Jacobian, and a hybrid mesh topology. One of the main advantages of the method is that the dimensions of the ROM are independent of the number of DOFs in the FOM, which is a desirable feature in the context of computational contact mechanics where highly refined FE discretizations are common. The variability of the set of contact shapes leads to additional terms in the equations of motion, but these were shown to be negligible in most analyses, and cheap to evaluate otherwise. The method's computational complexity per explicit time increment was shown to be two orders of magnitude smaller than standard CMS methods while maintaining a similar level of accuracy.

The method was introduced in the context of dynamic gear contact analysis, but can readily be applied to the simulation of bearings, cam-follower mechanisms, road-tire interactions, and other machine components and mechanisms that are characterized by a certain degree of predictability in the way the components interact. Having developed and validated the method for the perfectly-aligned gear contact problem, the method was extended to the case of spatially misaligned gear pairs and multi-mesh gear trains, leaving the basic idea underlying the original method unaltered. Particular attention was paid to three of the method's fundamental concepts: the accurate interpolation of contact shapes when the offline sampling of configuration parameters is coarse, the efficient training of the ROM when the dimensionality of the contact parameters is high, and the definition of contact shapes in a multi-contact setting where a single component engages with multiple components at the same time. Under all of these circumstances, the method proved to be robust and reliable, yielding ROMs that are both efficient to construct and to evaluate.

9.1.2 Selection of an optimal integration scheme

The selection of a proper integration scheme is an often overlooked yet essential aspect of efficiently solving dynamic contact problems. In industrial-sized contact analyses it is common

practice to choose explicit over implicit schemes. While the superior stability properties of iterative schemes allow for considerably larger time increments, this generally does not result in reduced simulation times due to the expensive linearization of high-dimensional nonlinear FE models. Explicit integration schemes instead enable relatively cheap state updates, yet they require small time increments to yield a stable solution, especially when the frequency content of the solution is high.

In this dissertation the selection of an optimal integration scheme was investigated in light of the novel MOR method. The influence of lowering the number of DOFs on the performance of numerical integrators is two-fold: in explicit integration schemes it filters out high-frequency components from the equations of motion, thereby increasing the size of the stable time increments. In implicit integration schemes, it lowers the cost of the linear solves performed at each Newton iteration by reducing the number of states that need to be perturbed. Owing to the significant dimensional reduction that is enabled by the novel MOR method, both explicit and implicit integrators are brought to the fore.

The optimal integration scheme was selected by comparing the performance of a number of well-established explicit and implicit numerical integrators in solving the reduced-order gear contact problem. A preliminary selection was made based on the order of convergence and the availability of an efficient yet reliable estimate of the local truncation error. The latter proved to be particularly important in FE-based contact simulations, where increment size changes of multiple orders of magnitude within a single simulation are not uncommon. An order of convergence of two was shown to be sufficient in dynamic contact analyses, with higher orders of convergence being favorable only if it comes with greater stability. Among the candidate explicit integrators are the Runge-Kutta methods, Adams-Moulton predictor-corrector methods, and the central difference method. The candidate implicit integrators are the backward differentiation formula's and the family of Newmark integrators.

Three kinds of equations were solved using these integrators: the Ordinary Differential Equations (ODEs) resulting from the semi-discretization and reduction of the dynamic equilibrium equations, the mixed Differential-Algebraic Equations (DAEs) resulting from the quasi-static consideration of high-frequency solution components, and the DAEs resulting from the enforcement of algebraic constraints on the dynamic equilibrium equations, as encountered in the modelling of planetary gear sets with rigid planet carriers. The optimal integrator was shown to be the second-order adaptive Newmark integrator combined with the analytical contact Jacobian and Broyden's method for approximating the system's Jacobian. The central difference method with velocities lagging half a time step and with quasi-static consideration of the contact shapes proved to be competitive in terms of required CPU time, yet comes with a less reliable local error estimator and an order of convergence that is close to but not exactly two. Compared to the fixed-increment central difference integrator that was used as the reference integration scheme in the first part of the dissertation, these optimal integrators enabled a reduction of the overall simulation time with a factor of 20 and more.

9.1.3 Hyperreduction of the nonlinear contact forces

Having reduced the number of DOFs and optimized the numerical integration scheme that solves the reduced-order equations of motion, up to 98% of the simulation time is spent on enforcing the contact constraints. This is because projection-based MOR techniques only reduce the computational complexity of linear operators, whereas the enforcement of contact constraints is a highly nonlinear procedure that operates on the full-order model coordinates. As in the majority of the FE-based contact analyses, this work uses a penalty approach combined with a Node-To-Surface (NTS) contact detection algorithm to enforce the contact constraints. The computational cost of enforcing these constraints scales quadratically with the number of finite elements used to discretize the contact surfaces, thereby partially jeopardizing the order-reduction that was brought about by the novel MOR method.

In order to remove the dependence of the ROM's complexity on the dimensions of the underlying FOM and to further increase its efficiency, this work investigated a second level of approximation generally referred to as hyperreduction. Most if not all successful hyperreduction schemes rely on the selective evaluation of a subset of components of the nonlinear function they aim to approximate. The Discrete Empirical Interpolation Method (DEIM) approximates the full-order nonlinear function in the physical domain before projecting it in the reduced-order subspace, whereas the Energy-Conserving Sampling and Weighting (ECSW) method directly approximates the reduced-order nonlinear function via a comparatively small number of back-and-forth subspace projections on element level.

This dissertation investigated the usage of both the DEIM and ECSW hyperreduction schemes to approximate the vector of nonlinear contact forces. Three main challenges are addressed in this manuscript. The first involves the development of efficient hyperreduction schemes in the context of parametric model reduction, where the continuous interpolation of contact shapes conflicts with the discontinuous interpolation of collocation indices. The second challenge relates to the preservation of Newton's third law of motion, which imposes constraints on the training procedure that is used to construct the hyperreduced ROM. Finally, the third challenge is of algorithmic nature and is concerned with ensuring that the selective evaluation of contact pairs doesn't get caught up in the tangle of interdependencies that is characteristic of FE-based contact force computations.

Of the two developed hyperreduction schemes, the ECSW-based scheme proved to be about 40% more efficient than the DEIM scheme, which is primarily due to the reduced cost of the involved subspace projections. The DEIM scheme on the other hand offers greater robustness with respect to changes in configuration parameters, allowing a coarser sampling of the parameter domain. The number of contact pairs that need to be evaluated is approximately the same in both schemes and is independent of the dimensions of the FOM. The only FE dependency that is left in the hyperreduced model originates from the back-and-forth projection of contact forces, which comes at a cost that scales linearly with the number of finite elements on the contact surfaces. A distinctive feature of the two developed hyperreduction schemes is that the hyperreduced ROMs

are trained during the computation of the contact shapes, and therefore do not imply additional offline training costs. In the investigated numerical examples, the cost of computing the nonlinear contact forces was reduced with a factor of roughly 22 in the modified DEIM scheme and 35 in the ECSW-based scheme. Overall, these factors yield a computational gain of roughly one order of magnitude as compared to the non-hyperreduced ROM.

9.1.4 Overall conclusion

This dissertation investigated the three main reasons behind the high computational cost of FE-based dynamic contact simulations. For these kind of simulations to be practically useful in the emerging fields of virtual prototyping and digital twin creation, the overall simulation cost needs to be reduced with orders of magnitude that are beyond the capabilities of standard MOR techniques. Drawing from the fields of parametric MOR, hyperreduction and adaptive numerical integration, this dissertation reduced the overall simulation cost of dynamic contact simulations with roughly four orders of magnitude as compared to standard reduction techniques, without jeopardizing the first-principles approach of finite-element modelling. While the real-time simulation of mechanical systems whose inner workings rely on complex contact interactions remains a challenge yet to be met, this work moved these kinds of simulations from the state- or university-owned supercomputing clusters to the mechanical engineer's personal computer.

9.2 Prospects for future research

Research never ends, inevitably making the presentation of research results as a closed whole somewhat artificial. New ideas and insights rarely come without new challenges and further research opportunities. In the following five sections, the most promising research directions that have emerged from the presented work are discussed. Some of these directions are currently being explored by the author and his colleagues, others are waiting to be embarked on.

9.2.1 Online adaptive model reduction

Most model-order and hyperreduction schemes that are designed for the efficient analysis of nonlinear and/or parametric systems rely on an often expensive training phase to construct the reduced-order models. While the current work invested considerable efforts in reducing the cost of this training phase, the accuracy of the obtained ROMs remains dependent on the comprehensiveness of the training scenario's and on the validity of the training assumptions. In online adaptive model reduction, the ROM is constructed and progressively adapted based on model information that is gathered at previous time steps. While adaptive *order* reduction is restricted by the expensive projection of system matrices, the emerging field of hyperreduction seems particularly amenable for online adaptation.

Two main requirements need to be fulfilled for adaptive hyperreduction to be practically feasible: the online adaptation of hyperreduced models should invoke only a small number of floating point operations, and the accuracy of the current hyperreduction approximation should be continuously monitored in a cost-effective way (much like the local truncation error is monitored in adaptive time integration). In the context of FE-based contact analysis, the DEIM hyperreduction scheme is a good candidate for adaptive model reduction: it has proven to be robust with respect to changes in configuration parameters, and it approximates the physical instead of the reduced-order contact force, thereby paving the way for an efficient error estimator (based on the superfluous yet inevitable evaluation of non-collocated contact pairs). Furthermore, the central algorithm behind the DEIM is the singular value decomposition, for which a number of efficient low-rank updates have been formulated in literature.

Adaptive hyperreduction has the potential of mitigating some of the training assumptions that were introduced in the current work. Moreover, in the context of parametric hyperreduction adaptive schemes are likely to select fewer collocation points, thereby increasing the overall efficiency of the contact force evaluation.

9.2.2 Surface smoothing in a model reduction setting

This dissertation concentrated on reducing the computational cost of FE-based contact simulations while preserving as much as possible the accuracy of the FOM. In the FE solution of contact problems, the interacting surfaces are typically discretized using Lagrange elements exhibiting C^0 continuity at the inter-element boundaries, which in an isoparametric context applies to both the geometry and the displacement field. This non-smooth discretization can lead to convergence issues in the analysis of sliding contact problems, as well as to highly oscillatory contact forces even when convergence is achieved.

The novel MOR method presented in this dissertations allows for an efficient procedure to smoothen the interacting contact surfaces and to alleviate the effects of C^0–continuous surface discretizations. The idea is to globally interpolate the nodes on the contact surfaces by B-spline surfaces that are at least C^1–continuous and to use these smooth surface descriptions to enforce the contact constraints. Owing to the linear dependence between B-spline surface descriptions and the associated B-spline control points, it suffices to compute these control points only once prior to the actual simulation for all the shape vectors in the reduction basis. During the online phase of the simulation, the smoothed surface representations are obtained by computing the weighted sum of the precomputed control points based on the modal participation factors of the respective shape vectors.

To fully exploit the potential of this surface smoothing approach, the NTS contact detection algorithm that was used in the current work needs to be replaced by a Segment-To-Segment (STS) algorithm, as was demonstrated by the author in a series of numerical experiments. These STS algorithms offer greater robustness and accuracy than their NTS counterparts, yet they come at a considerable price, especially when mortar formulations are used. However, a number of

countermeasures can be taken to reduce the cost of these methods in dynamic contact simulations: firstly, in the FE-based simulation of gears or bearings, the predictable kinematics and structured lay-out of the FE meshes can be exploited to reduce the effort of finding and approximating the overlap regions, thereby lowering the cost of evaluating the mortar integrals. Secondly, inspired by the hyperreduction schemes that were developed in this dissertation for NTS-based contact force computations, a STS-based hyperreduced contact force algorithm could enable smoother, more realistic contact forces at a reasonable price.

9.2.3 Semi-analytical reduced-order contact modelling

The evaluation of contact forces in a FE context remains the main computational bottleneck in reduced-order dynamic contact analysis. Even when hyperreduction is applied, the number of floating point operations involved in evaluating the contact forces dominates the assembly of the generalized force residuals, and the artificial oscillations induced by C^0–continuous surface discretizations force adaptive time integrators to use relatively small time increments. A good compromise between first-principles modelling and computational efficiency can be found in combining semi-analytical contact force computations with the MOR techniques presented in this dissertation.

The main features of such a semi-analytical, reduced-order modelling approach applied to the dynamic gear contact problem could be the following. The global-local consideration of elastic displacements that was popularized by Andersson & Vedmar [19] (see also Section 5.3) is used to separate the local displacement from the global deformation in the definition of the contact shapes. The global surface displacements represented by the *globalized* contact shapes and the set of retained natural vibration modes are globally interpolated using smooth surfaces (e.g. the B-spline surfaces introduced in Section 9.2.2) to obtain an analytical description of the global displacements in terms of the generalized elastic coordinates of the gears. The FE representations of the gears are reduced using the interpolated contact shapes approach, yielding a minimal set of elastic coordinates. The FE-based contact detection is replaced by a kinematic contact detection algorithm along the lines of the lumped-parameter models discussed in Section 3.3.1, where the global elastic displacement of the reduced-order gear models are obtained by finding the analytical distance between the perfect involute profile and the deformed surface-smoothed gear contact surface along the gear pair's line of action. The local overlap between globally deformed gear surfaces is used to compute the resulting contact stress distributions based on Hertzian contact theory. Finally, these stress distributions are integrated over the discretized contact surfaces using Gaussian quadrature to obtain the FE contact forces.

The advantage of the proposed modelling approach is that the need of an exhaustive contact search is eliminated, while at the same time the numerical stiffness of the problem is reduced by allowing local contact penetration. Furthermore, as the contact force evaluation is based on single-point intersections of smooth analytical surface descriptions, no artificial oscillations are introduced through the contact forces. A related approach that combines local Hertzian contact

theory with FE-based contact detection was recently published in an article that is co-authored by the author of this dissertation [161].

9.2.4 Lubricated contact dynamics and multiphysical model reduction

The performance of mechanical systems whose inner workings rely on contact interactions is significantly influenced by the lubrication of the contact surfaces. At the fundamental level, the governing equations of lubricant theory are the Navier-Stokes equations, with the two-dimensional Reynolds equations yielding accurate approximations of fluid pressure distributions in thin viscous fluid films. While the numerical solution of the semi-discretized Reynolds equations is relatively inexpensive in comparison with solving the three-dimensional continuum equations, the lubricant film should be taken into account when computing the contact shapes. What is called for is a multiphysical model reduction framework that offers a holistic consideration of two physically distinct domains to identify the optimal reduction basis for each domain separately or for the two domains combined. Assuming fully-flooded conditions, the computation of solid-to-solid contact forces is replaced by evaluating fluid pressure distributions, which poses new challenges to the field of hyperreduction. The availability of efficient lubricated contact models are expected to further close the gap between numerical simulations and experimental measurements, especially at high operating velocities, where the damping effects of lubricant films have a significant influence on the dynamic behavior of the interacting components.

9.2.5 Towards system-level simulation

The ultimate goal of reduced-order contact modelling is to bring efficient contact models towards the system-level design and analysis of mechatronic drivetrains and systems. The development of a FE-based gear contact force element in Section 6.3 was a first step towards the exploitation of the methods developed in this dissertation in system-level analysis. Apart from the never-ending quest for more efficient model formulations, a number of additional system-level challenges should be addressed in the upcoming years. First, the large variety of numerical (contact) models that is currently available calls for a clear classification – assisted by both numerical and experimental investigations – based on the applications in which these models perform best (in terms of required accuracy versus efficiency). Second, the coupling of three-dimensional distributed-parameter models with lower-dimensional mechatronic building blocks needs to be optimized both from the software point of view, as with regard to the accurate transfer of information at the model interfaces. Thirdly, the accurate solution of system-level models calls for error-controlled asynchronous integrators that are able to efficiently cope with the largely varying time scales of these models.

Appendix A

Derivation of the analytical contact Jacobian

The generalized contact force vector $\boldsymbol{Q}_c{}^T = \{\boldsymbol{Q}_c^{1^T} \quad \boldsymbol{Q}_c^{2^T}\} \in \mathbb{R}^N$ acting between two flexible bodies (denoted by the superscripts 1 and 2) is a function of the vector of generalized coordinates $\boldsymbol{q}^T = \{\boldsymbol{q}^{1^T} \quad \boldsymbol{q}^{2^T}\} \in \mathbb{R}^N$. It can be written as a summation of the generalized contact forces $\boldsymbol{Q}_{c,j}^i$ corresponding to the physical contact forces $\boldsymbol{f}_{c,j}^i, j = 1,..,n_F$ using the equations developed in Section 2.4.3:

$$\boldsymbol{Q}_c^i = \sum_{j=1}^{n_F} \boldsymbol{Q}_{c,j}^i = \sum_{j=1}^{n_F} \begin{Bmatrix} \boldsymbol{f}_{c,j}^i \\ \overline{\boldsymbol{G}}^{i^T} \widetilde{\overline{\boldsymbol{u}}}^{ij} \boldsymbol{A}^{i^T} \boldsymbol{f}_{c,j}^i \\ \overline{\boldsymbol{V}}^{ij^T} \boldsymbol{A}^{i^T} \boldsymbol{f}_{c,j}^i \end{Bmatrix} \tag{A.1}$$

where $\overline{\boldsymbol{u}}^{ij} = \overline{\boldsymbol{u}}_0^{ij} + \overline{\boldsymbol{V}}^{ij} \boldsymbol{q}_f^i$ is the position vector in the local reference frame to the *j*-th point of contact, and the transformation matrices $\boldsymbol{A}^i$ (from the local body reference frame to the global reference frame) and $\overline{\boldsymbol{G}}^i$ are defined in 0. The aim of this appendix is to obtain an analytical expression for the derivative of Eq. A.1 with respect to the vector of generalized coordinates $\boldsymbol{q}$, also referred to as the contact Jacobian. The availability of an analytical contact Jacobian can greatly accelerate the computation of contact shapes (see Section 4.2.2) and the solution of the equations of motion using implicit methods (see Section 7.4.1.2).

The Jacobian of the generalized contact forces can be decomposed as follows:

$$\boldsymbol{J}_c = \frac{\partial \boldsymbol{Q}_c}{\partial \boldsymbol{q}} = \begin{bmatrix} \frac{\partial \boldsymbol{Q}_c^1}{\partial \boldsymbol{q}^1} & \frac{\partial \boldsymbol{Q}_c^1}{\partial \boldsymbol{q}^2} \\ \frac{\partial \boldsymbol{Q}_c^2}{\partial \boldsymbol{q}^1} & \frac{\partial \boldsymbol{Q}_c^2}{\partial \boldsymbol{q}^2} \end{bmatrix} \tag{A.2}$$

where an expression for the partial derivatives can be obtained from Eq. A.1:

$$\frac{\partial \boldsymbol{Q}_c^k}{\partial \boldsymbol{q}^i} = \sum_{j=1}^{n_F} \frac{\partial \boldsymbol{Q}_{c,j}^k}{\partial \boldsymbol{q}^i} = \sum_{j=1}^{n_F} \boldsymbol{J}_{c,j}^{k,i} = \sum_{j=1}^{n_F} \frac{\partial}{\partial \boldsymbol{q}^i} \begin{Bmatrix} \boldsymbol{f}_{c,j}^k \\ \overline{\boldsymbol{G}}^{k^T} \widetilde{\overline{\boldsymbol{u}}}^{kj} \boldsymbol{A}^{k^T} \boldsymbol{f}_{c,j}^k \\ \overline{\boldsymbol{V}}^{kj^T} \boldsymbol{A}^{k^T} \boldsymbol{f}_{c,j}^k \end{Bmatrix} \tag{A.3}$$

The contact Jacobian $\boldsymbol{J}_{c,j}^{k,i} = \partial \boldsymbol{Q}_{c,j}^{k} / \partial \boldsymbol{q}^{i}$ corresponding to the physical contact force $\boldsymbol{f}_{c,j}^{k}$ can be written in terms of these generalized coordinates as follows:

$$\boldsymbol{J}_{c,j}^{k,i} = \begin{bmatrix} \frac{\partial \boldsymbol{f}_{c,j}^{k}}{\partial \boldsymbol{R}^{i}} & \frac{\partial \boldsymbol{f}_{c,j}^{k}}{\partial \boldsymbol{\theta}^{i}} & \frac{\partial \boldsymbol{f}_{c,j}^{k}}{\partial \boldsymbol{q}_{f}^{i}} \\ \bar{\boldsymbol{G}}^{k^T} \tilde{\bar{\boldsymbol{u}}}^{kj} \boldsymbol{A}^{k^T} \frac{\partial \boldsymbol{f}_{c,j}^{k}}{\partial \boldsymbol{R}^{i}} & \frac{\partial \left(\bar{\boldsymbol{G}}^{k^T} \tilde{\bar{\boldsymbol{u}}}^{kj} \boldsymbol{A}^{k^T} \right)}{\partial \boldsymbol{\theta}^{i}} \boldsymbol{f}_{c,j}^{k} + \bar{\boldsymbol{G}}^{k^T} \tilde{\bar{\boldsymbol{u}}}^{kj} \boldsymbol{A}^{k^T} \frac{\partial \boldsymbol{f}_{c,j}^{k}}{\partial \boldsymbol{\theta}^{i}} & \bar{\boldsymbol{G}}^{k^T} \tilde{\bar{\boldsymbol{V}}}^{kj} \boldsymbol{A}^{k^T} \boldsymbol{f}_{c,i} + \bar{\boldsymbol{G}}^{k^T} \tilde{\bar{\boldsymbol{u}}}^{kj} \boldsymbol{A}^{k^T} \frac{\partial \boldsymbol{f}_{c,j}^{k}}{\partial \boldsymbol{q}_{f}^{i}} \\ \bar{\boldsymbol{V}}^{kj^T} \boldsymbol{A}^{k^T} \frac{\partial \boldsymbol{f}_{c,j}^{k}}{\partial \boldsymbol{R}^{i}} & \bar{\boldsymbol{V}}^{kj^T} \frac{\partial \boldsymbol{A}^{k^T}}{\partial \boldsymbol{\theta}^{i}} \boldsymbol{f}_{c,j}^{k} + \bar{\boldsymbol{V}}^{kj^T} \boldsymbol{A}^{k^T} \frac{\partial \boldsymbol{f}_{c,j}^{k}}{\partial \boldsymbol{\theta}^{i}} & \bar{\boldsymbol{V}}^{kj^T} \boldsymbol{A}^{k^T} \frac{\partial \boldsymbol{f}_{c,j}^{k}}{\partial \boldsymbol{q}_{f}^{i}} \end{bmatrix} \quad \text{(A.4)}$$

This matrix includes the Jacobian matrix of the physical contact force $\boldsymbol{f}_{c,j}^{k}$ w.r.t. the vector of generalized coordinates $\boldsymbol{q}^{i}$, i.e. the physical contact Jacobian $\boldsymbol{J}_{f,j}^{k,i}$, which will be computed in Section A.1. It is assumed that a master-slave, node-to-segment approach is used to compute the contact forces, in which the first body is the master body and the second body is the slave body.

A.1 Derivative of the physical contact force

In the master body, which is assumed to be discretized using eight-noded solid elements (see Section 2.2.2), the contact force is distributed over the four corner nodes of the penetrated element surface. The locations of these corner nodes are represented in the global reference frame by the position vectors $\boldsymbol{x}^{1l}, l = 1 \dots 4$. The weighting factors for the contact force distribution are the so-called iso-geometric parameters ξ_c^1 and η_c^1 that define the physical point of contact, $\boldsymbol{x}_c^1$, according to:

$$\boldsymbol{x}_c^1 = \boldsymbol{x}^1(\xi_c^1, \eta_c^1) = \frac{1}{4} \sum_{l=1}^{4} \boldsymbol{x}^{1l} \left(1 + \xi_l^1 \xi_c^1\right) \left(1 + \eta_l^1 \eta_c^1\right) \quad \text{(A.5)}$$

where ξ_l^1 and η_l^1 are the iso-geometric parameters corresponding to the corner nodes. Accordingly, the repartitioned contact force at the l-th surface node can be written as

$$\boldsymbol{f}_{c,j}^{1,l} = \frac{1}{4} \left(1 + \xi_l^1 \xi_c^1\right) \left(1 + \eta_l^1 \eta_c^1\right) \boldsymbol{F}_{c,j}^{1} = w^l(\xi_c^1, \eta_c^1) \boldsymbol{F}_{c,j}^{1} = w_c^l \boldsymbol{F}_{c,j}^{1} \quad \text{(A.6)}$$

where w_c^l is the weighting factor and $\boldsymbol{f}_{c,j}^{1}$ is the equivalent physical contact force. The computation of this force is given by (see Section 2.5.3.2):

$$\boldsymbol{f}_{c,j}^{1} = \boldsymbol{n}^1(\xi_c^1, \eta_c^1) \cdot c_p g_N(\xi_c^1, \eta_c^1) = \boldsymbol{n}_c^1 \cdot c_p g_N(\xi_c^1, \eta_c^1) \quad \text{(A.7)}$$

where $\boldsymbol{n}_c^1$ is the unit normal at the contact point, c_p is the penalty factor, and the gap function is given by (see Section 2.5.2)

$$g_N = \boldsymbol{n}_c^1 \cdot (\boldsymbol{x}^1(\xi_c^1, \eta_c^1) - \boldsymbol{s}) = \boldsymbol{n}_c^1 \cdot (\boldsymbol{x}_c^1 - \boldsymbol{s}^2) \tag{A.8}$$

where $\boldsymbol{s}$ is the absolute position vector of the slave node penetrating the master segment. Substitution of this equation in Eq. A.7 yields:

$$\boldsymbol{f}_{c,j}^1 = \boldsymbol{n}^1(\xi_c^1, \eta_c^1) \cdot c_p\big(\boldsymbol{n}_c^1 \cdot (\boldsymbol{x}_c^1 - \boldsymbol{s}^2)\big) = c_p \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T} (\boldsymbol{x}_c^1 - \boldsymbol{s}^2) \tag{A.9}$$

so that the partitioned contact force equals:

$$\boldsymbol{f}_{c,j}^{1,l} = \frac{c_p}{4}\big(1 + \xi_l^1 \xi_c^1\big)\big(1 + \eta_l^1 \eta_c^1\big)\Big[\boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T} (\boldsymbol{x}_c^1 - \boldsymbol{s}^2)\Big] = c_p w_c^l \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T} (\boldsymbol{x}_c^1 - \boldsymbol{s}^2) \tag{A.10}$$

Finally, the unit normal vector $\boldsymbol{n}_c^1$ is given by (see Section 2.5.2)

$$\boldsymbol{n}_c^1 = \frac{\boldsymbol{N}_c^1}{\|\boldsymbol{N}_c^1\|} = \frac{\boldsymbol{x}_{,\xi}(\xi_c^1, \eta_c^1) \times \boldsymbol{x}_{,\eta}(\xi_c^1, \eta_c^1)}{\big\|\boldsymbol{x}_{,\xi}(\xi_c^1, \eta_c^1) \times \boldsymbol{x}_{,\eta}(\xi_c^1, \eta_c^1)\big\|} \tag{A.11}$$

where subscript ',' means 'differentiated with respect to'.

The physical contact Jacobian $\boldsymbol{J}_{f,j}^{1,l}$, corresponding to the partitioned contact force $\boldsymbol{f}_{c,j}^{1,l}$ can be written as follows:

$$\boldsymbol{J}_{f,j}^{1,l} = \frac{\partial \boldsymbol{f}_{c,j}^{1,l}}{\partial \boldsymbol{q}^i} = \left[\frac{\partial \boldsymbol{f}_{c,j}^{1,l}}{\partial \boldsymbol{q}^1} \quad \frac{\partial \boldsymbol{f}_{c,j}^{1,l}}{\partial \boldsymbol{q}^2}\right] \tag{A.12}$$

Substitution of Eq. A.10 in the first derivative in this equation yields:

$$\frac{\partial \boldsymbol{f}_{c,j}^{1,l}}{\partial \boldsymbol{q}^1} = c_p \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T} (\boldsymbol{x}_c^1 - \boldsymbol{s}^2) \frac{\partial w_c^l}{\partial \boldsymbol{q}^1} + c_p w_c^l \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{q}^1} (\boldsymbol{x}_c^1 - \boldsymbol{s}^2) + c_p w_c^l \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T} \frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{q}^1} \tag{A.13}$$

Using the chain rule of differentiation, the following expressions for these derivatives are obtained:

$$\frac{\partial w_c^l}{\partial \boldsymbol{q}^1} = \frac{\partial w_c^l}{\partial \xi_c^1}\frac{\partial \xi_c^1}{\partial \boldsymbol{q}^1} + \frac{\partial w_c^l}{\partial \eta_c^1}\frac{\partial \eta_c^1}{\partial \boldsymbol{q}^1} = \frac{\partial w_c^l}{\partial \xi_c^1}\sum_{l=1}^{4}\left(\frac{\partial \xi_c}{\partial \boldsymbol{x}^{1l}}\frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1}\right) + \frac{\partial w_c^l}{\partial \eta_c^1}\sum_{l=1}^{4}\left(\frac{\partial \eta_c^1}{\partial \boldsymbol{x}^{1l}}\frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1}\right) \tag{A.14}$$

$$\frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{q}^1} = \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{n}_c^1}\frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{q}^1} = \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{n}_c^1}\frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{N}_c^1}\frac{\partial \boldsymbol{N}_c^1}{\partial \boldsymbol{q}^1} \tag{A.15}$$

$$= \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1T}}{\partial \boldsymbol{n}_c^1} \frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{N}_c^1} \left\{ \frac{\partial \boldsymbol{N}_c^1}{\partial \xi_c^1} \frac{\partial \xi_c^1}{\partial \boldsymbol{q}^1} + \frac{\partial \boldsymbol{N}_c^1}{\partial \eta_c^1} \frac{\partial \eta_c^1}{\partial \boldsymbol{q}^1} + \sum_{l=1}^{4} \left(\frac{\partial \boldsymbol{N}_c^1}{\partial \boldsymbol{x}^{1l}} \frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} \right) \right\}$$

$$= \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1T}}{\partial \boldsymbol{n}_c^1} \frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{N}_c^1} \left\{ \frac{\partial \boldsymbol{N}_c^1}{\partial \xi_c^1} \sum_{l=1}^{4} \left(\frac{\partial \xi_c^1}{\partial \boldsymbol{x}^{1l}} \frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} \right) + \frac{\partial \boldsymbol{N}_c^1}{\partial \eta_c^1} \sum_{l=1}^{4} \left(\frac{\partial \eta_c^1}{\partial \boldsymbol{x}^{1l}} \frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} \right) + \sum_{l=1}^{4} \left(\frac{\partial \boldsymbol{N}_c^1}{\partial \boldsymbol{x}^{1l}} \frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} \right) \right\}$$

$$\frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{q}^1} = \frac{\partial \boldsymbol{x}_c^1}{\partial \xi_c^1} \frac{\partial \xi_c^1}{\partial \boldsymbol{q}^1} + \frac{\partial \boldsymbol{x}_c^1}{\partial \eta_c^1} \frac{\partial \eta_c^1}{\partial \boldsymbol{q}^1} + \sum_{l=1}^{4} \left(\frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{x}^{1l}} \frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} \right)$$

$$= \frac{\partial \boldsymbol{x}_c}{\partial \xi_c^1} \sum_{l=1}^{4} \left(\frac{\partial \xi_c^1}{\partial \boldsymbol{x}^{1l}} \frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} \right) + \frac{\partial \boldsymbol{x}_c}{\partial \eta_c^1} \sum_{l=1}^{4} \left(\frac{\partial \eta_c^1}{\partial \boldsymbol{x}^{1l}} \frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} \right) + \sum_{l=1}^{4} \left(\frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{x}^{1l}} \frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} \right) \tag{A.16}$$

These equations contain two very complex derivatives, namely $\partial \xi_c^1 / \partial \boldsymbol{x}^{1l}$ and $\partial \eta_c^1 / \partial \boldsymbol{x}^{1l}$. An explicit exact analytical derivation of these derivatives is not possible, due to the implicit nature of the iso-geometric parameters ξ_c^1 and η_c^1. An efficient approximation of these derivatives can be found using finite differences

$$\frac{\partial \xi_c^1}{\partial \boldsymbol{x}^{1l}} \approx \left[\frac{\xi_c^1(\boldsymbol{x}^{1l} + \Delta x_1^{1l}) - \xi_c^1}{\Delta x_1^{1l}} \quad \frac{\xi_c^1(\boldsymbol{x}^{1l} + \Delta x_2^{1l}) - \xi_c^1}{\Delta x_2^{1l}} \quad \frac{\xi_c^1(\boldsymbol{x}^{1l} + \Delta x_2^{1l}) - \xi_c^1}{\Delta x_3^{1l}} \right] \tag{A.17}$$

where $\xi_c^1(\boldsymbol{x}^{1l} + \Delta x_m^{1l})$ is computed numerically (using e.g. a single iteration of a Newton-Raphson scheme) by perturbing the *m*-th component of the *l*-th position vector. The computation of the aforementioned two partial derivatives will then require 12 evaluations of the new iso-geometric contact parameters, yet the computational cost of this numerical approximation is expected to be of similar order of magnitude as the analytical approximation. The remaining partial derivatives in Eq. A.14-A.16 are relatively straight-forward. The derivatives of the contact point w.r.t. the iso-geometric contact parameters ξ_c and η_c can easily be obtained using Eq. A.5:

$$\frac{\partial \boldsymbol{x}_c^1}{\partial \xi_c^1} = \frac{1}{4} \sum_{l=1}^{4} \boldsymbol{x}^{1l} \xi_l^1 \left(1 + \eta_l^1 \eta_c^1\right) \tag{A.18}$$

$$\frac{\partial \boldsymbol{x}_c^1}{\partial \eta_c} = \frac{1}{4} \sum_{l=1}^{4} \boldsymbol{x}^{1l} \eta_l^1 \left(1 + \xi_l^1 \xi_c^1\right) \tag{A.19}$$

The derivatives of the weighting factors w.r.t. the iso-geometric parameters can be obtained from Eq. A.6 as

$$\frac{\partial w_c^l}{\partial \xi_c^1} = \frac{1}{4} \xi_l^1 \left(1 + \eta_l^1 \eta_c^1\right) \tag{A.20}$$

$$\frac{\partial w_c^l}{\partial \eta_c^1} = \frac{1}{4}\eta_l^1\left(1 + \xi_l^1\xi_c^1\right) \tag{A.21}$$

The derivative of $\boldsymbol{n}_c^1\boldsymbol{n}_c^{1^T}$ w.r.t. the unit normal vector $\boldsymbol{n}_c^1$ can be written as follows using the Kronecker delta product:

$$\frac{\partial \boldsymbol{n}_c^1\boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{n}_c^1} = \boldsymbol{I}_{3\times3} \otimes \boldsymbol{n}_c^1 + Vec(\boldsymbol{I}_{3\times3}) \otimes \boldsymbol{n}_c^{1^T} \tag{A.22}$$

or in matrix notation:

$$\frac{\partial \boldsymbol{n}_c^1\boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{n}_c^1} = \begin{bmatrix} 2\boldsymbol{n}_{c,x}^1 & \boldsymbol{n}_{c,y}^1 & \boldsymbol{n}_{c,z}^1 & \vdots & 0 & \boldsymbol{n}_{c,x}^1 & 0 & \vdots & 0 & 0 & \boldsymbol{n}_{c,x}^1 \\ \boldsymbol{n}_{c,y}^1 & 0 & 0 & \vdots & \boldsymbol{n}_{c,x}^1 & 2\boldsymbol{n}_{c,y}^1 & \boldsymbol{n}_{c,z}^1 & \vdots & 0 & 0 & \boldsymbol{n}_{c,y}^1 \\ \boldsymbol{n}_{c,z}^1 & 0 & 0 & \vdots & 0 & \boldsymbol{n}_{c,z}^1 & 0 & \vdots & \boldsymbol{n}_{c,x}^1 & \boldsymbol{n}_{c,y}^1 & 2\boldsymbol{n}_{c,z}^1 \end{bmatrix} \tag{A.23}$$

The derivative of the unit normal vector $\boldsymbol{n}_c^1$ w.r.t. its magnitude $\boldsymbol{N}_c^1$ can be obtained using the quotient rule of differentiation:

$$\begin{aligned} \frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{N}_c^1} = \frac{\partial}{\partial \boldsymbol{N}_c^1}\left(\frac{\boldsymbol{N}_c^1}{\|\boldsymbol{N}_c^1\|}\right) = \frac{\partial}{\partial \boldsymbol{N}_c^1}\left(\frac{\boldsymbol{N}_c^1}{\sqrt{\boldsymbol{N}_c^{1^T}\boldsymbol{N}_c^1}}\right) = \frac{\boldsymbol{I}_{3\times3}}{\|\boldsymbol{N}_c^1\|} - \frac{\boldsymbol{N}_c^1\boldsymbol{N}_c^{1^T}}{\|\boldsymbol{N}_c^1\|^3} \\ = \frac{1}{\|\boldsymbol{N}_c^1\|}\left(\boldsymbol{I}_{3\times3} - \boldsymbol{n}_c^1\boldsymbol{n}_c^{1^T}\right) \end{aligned} \tag{A.24}$$

The derivatives of the normal vector w.r.t. ξ_c^1 and η_c^1 can be obtained using Eq. A.11:

$$\frac{\partial \boldsymbol{N}_c^1}{\partial \xi^1}(\xi_c^1, \eta_c^1) = \boldsymbol{x}_{,\xi}(\xi_c^1, \eta_c^1) \times \boldsymbol{x}_{,\eta\xi}(\xi_c^1, \eta_c^1) \tag{A.25}$$

$$\frac{\partial \boldsymbol{N}_c^1}{\partial \eta^1}(\xi_c^1, \eta_c^1) = \boldsymbol{x}_{,\xi\eta}(\xi_c^1, \eta_c^1) \times \boldsymbol{x}_{,\eta}(\xi_c^1, \eta_c^1) \tag{A.26}$$

where use was made of the fact that for bilinear shape functions, the second derivatives are zero. The derivatives of the normal vector w.r.t. the corner node position vector $\boldsymbol{x}^{1l}$ can be obtained from the same equation and are given by

$$\begin{aligned} \frac{\partial \boldsymbol{N}_c^1}{\partial \boldsymbol{x}^{1l}} = \frac{\partial \boldsymbol{x}_{,\xi}}{\partial \boldsymbol{x}^{1l}} \times \boldsymbol{x}_{,\eta} + \boldsymbol{x}_{,\xi} \times \frac{\partial \boldsymbol{x}_{,\eta}}{\partial \boldsymbol{x}^{1l}} \\ = \frac{\xi_l^1\left(1 + \eta_l^1\eta_c^1\right)}{4}\boldsymbol{I}_{3\times3} \times \boldsymbol{x}_{,\eta} + \boldsymbol{x}_{,\xi} \times \frac{\eta_l^1\left(1 + \xi_l^1\xi_c^1\right)}{4}\boldsymbol{I}_{3\times3} \end{aligned} \tag{A.27}$$

Next, the derivative of the physical contact point w.r.t. the *l*-th corner node position vector is obtained from Eq. A.5 as

$$\frac{\partial \boldsymbol{x}_c}{\partial \boldsymbol{x}^{1l}} = \frac{\partial \boldsymbol{x}(\xi_c^1, \eta_c^1)}{\partial \boldsymbol{x}^{1l}} = \frac{1}{4}\boldsymbol{I}_{3\times 3}\left(1 + \xi_l^1 \xi_c^1\right)\left(1 + \eta_l^1 \eta_c^1\right) \tag{A.28}$$

Finally, the derivative of the *l*-th corner node position vector w.r.t. the vector of generalized coordinates needs to be determined. The position vector $\boldsymbol{x}^{1l}$ can be expressed in terms of the generalized coordinates of the flexible body as follows:

$$\boldsymbol{x}^{1l} = \boldsymbol{R}^1 + \boldsymbol{A}^1 \overline{\boldsymbol{u}}^{1l} = \boldsymbol{R}^1 + \boldsymbol{A}^1\left(\overline{\boldsymbol{u}}_0^{1l} + \overline{\boldsymbol{u}}_f^{1l}\right) = \boldsymbol{R}^1 + \boldsymbol{A}^1\left(\overline{\boldsymbol{u}}_{0,l}^{1l} + \overline{\boldsymbol{\Psi}}^{1l}\boldsymbol{\eta}^1\right) \tag{A.29}$$

Then, the derivative of $\boldsymbol{x}^{1l}$ w.r.t. the vector of generalized coordinates is obtained as

$$\frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} = \begin{bmatrix} \boldsymbol{I} & \boldsymbol{A}_{\boldsymbol{\theta}}^1 \overline{\boldsymbol{u}}^{l1} & \boldsymbol{A}^1 \overline{\boldsymbol{\Psi}}^{l1} \end{bmatrix} = \begin{bmatrix} \boldsymbol{I} & -\boldsymbol{A}^1 \widetilde{\overline{\boldsymbol{u}}}^{l1} \overline{\boldsymbol{G}}^1 & \boldsymbol{A}^1 \overline{\boldsymbol{\Psi}}^{l1} \end{bmatrix} \tag{A.30}$$

Analogous to Eq. A.13, the second derivative in Eq. A.12 can be written as:

$$\begin{aligned} \frac{\partial \boldsymbol{f}_{c,j}^{1,l}}{\partial \boldsymbol{q}^2} = {} & c_p \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1\,T}(\boldsymbol{x}_c^1 - \boldsymbol{s}^2)\frac{\partial w_c^l}{\partial \boldsymbol{q}^2} + c_p w_c^l \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1\,T}}{\partial \boldsymbol{q}^2}(\boldsymbol{x}_c^1 - \boldsymbol{s}^2) \\ & + c_p w_c^l \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1\,T}\left(\frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{q}^2} - \frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2}\right) \end{aligned} \tag{A.31}$$

Using the chain rule of differentiation, the following expressions for these derivatives are obtained:

$$\frac{\partial w_c^l}{\partial \boldsymbol{q}^2} = \frac{\partial w_c^l}{\partial \xi_c^1}\frac{\partial \xi_c^1}{\partial \boldsymbol{q}^2} + \frac{\partial w_c^l}{\partial \eta_c^1}\frac{\partial \eta_c^1}{\partial \boldsymbol{q}^2} = \frac{\partial w_c^l}{\partial \xi_c^1}\frac{\partial \xi_c^1}{\partial \boldsymbol{s}^2}\frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2} + \frac{\partial w_c^l}{\partial \eta_c^1}\frac{\partial \eta_c^1}{\partial \boldsymbol{s}^2}\frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2} \tag{A.32}$$

$$\begin{aligned} \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1\,T}}{\partial \boldsymbol{q}^2} &= \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1\,T}}{\partial \boldsymbol{n}_c^1}\frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{N}_c^1}\frac{\partial \boldsymbol{N}_c^1}{\partial \boldsymbol{q}^2} = \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1\,T}}{\partial \boldsymbol{n}_c^1}\frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{N}_c^1}\left\{\frac{\partial \boldsymbol{N}_c^1}{\partial \xi_c^1}\frac{\partial \xi_c^1}{\partial \boldsymbol{q}^2} + \frac{\partial \boldsymbol{N}_c^1}{\partial \eta_c^1}\frac{\partial \eta_c^1}{\partial \boldsymbol{q}^2}\right\} \\ &= \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1\,T}}{\partial \boldsymbol{n}_c^1}\frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{N}_c^1}\left\{\frac{\partial \boldsymbol{N}_c^1}{\partial \xi_c^1}\frac{\partial \xi_c}{\partial \boldsymbol{s}^2}\frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2} + \frac{\partial \boldsymbol{N}_c^1}{\partial \eta_c^1}\frac{\partial \eta_c^1}{\partial \boldsymbol{s}^2}\frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2}\right\} \end{aligned} \tag{A.33}$$

$$\frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{q}^2} = \frac{\partial \boldsymbol{x}_c^1}{\partial \xi_c^1}\frac{\partial \xi_c^1}{\partial \boldsymbol{q}^2} + \frac{\partial \boldsymbol{x}_c^1}{\partial \eta_c^1}\frac{\partial \eta_c^1}{\partial \boldsymbol{q}^2} = \frac{\partial \boldsymbol{x}_c^1}{\partial \xi_c^1}\frac{\partial \xi_c}{\partial \boldsymbol{s}^2}\frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2} + \frac{\partial \boldsymbol{x}_c^1}{\partial \eta_c^1}\frac{\partial \eta_c^1}{\partial \boldsymbol{s}^2}\frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2} \tag{A.34}$$

The partial derivatives in these equations are computed using similar equations as are used to evaluate Eq. A.14-A.16. The partial derivatives of the isogeometric contact parameters $\partial \xi_c^1 / \partial \boldsymbol{s}^2$

and $\partial\eta_c^1/\partial \boldsymbol{s}^2$ are again obtained from numerical finite differences (three Newton-Raphson evaluations in total). The derivative of the slave node position vector $\boldsymbol{s}^2$ w.r.t. the vector of generalized coordinates of the slave body is obtained as in Eq. A.30:

$$\frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2} = [\boldsymbol{I} \quad \boldsymbol{A}_{\boldsymbol{\theta}}^2 \overline{\boldsymbol{u}}^2 \quad \boldsymbol{A}^2 \overline{\boldsymbol{\Psi}}^2] = [\boldsymbol{I} \quad -\boldsymbol{A}^2 \widetilde{\overline{\boldsymbol{u}}}^2 \overline{\boldsymbol{G}}^2 \quad \boldsymbol{A}^2 \overline{\boldsymbol{\Psi}}^2] \tag{A.35}$$

In the slave body, the contact force is applied to the slave node in the opposite direction of the contact normal $\boldsymbol{n}_c^1$. Therefore, we have the equality:

$$\boldsymbol{f}_{c,j}^2 = -\boldsymbol{f}_{c,j}^1 = -\boldsymbol{n}^1(\xi_c^1, \eta_c^1) \cdot c_p\left(\boldsymbol{n}_c^1 \cdot (\boldsymbol{x}_c^1 - \boldsymbol{s}^2)\right) = -c_p \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T} (\boldsymbol{x}_c^1 - \boldsymbol{s}^2) \tag{A.36}$$

The physical contact Jacobian corresponding to this force can be written as:

$$\boldsymbol{J}_{f,j}^{2,i} = \frac{\partial \boldsymbol{f}_{c,j}^2}{\partial \boldsymbol{q}^i} = \left[\frac{\partial \boldsymbol{f}_{c,j}^{2,l}}{\partial \boldsymbol{q}^1} \quad \frac{\partial \boldsymbol{f}_{c,j}^{2,l}}{\partial \boldsymbol{q}^2}\right] \tag{A.37}$$

Substitution of Eq. A.36 in this equation yields:

$$\frac{\partial \boldsymbol{f}_{c,j}^2}{\partial \boldsymbol{q}^1} = -c_p \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{q}^1} (\boldsymbol{x}_c^1 - \boldsymbol{s}^2) - c_p \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T} \frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{q}^1} \tag{A.38}$$

$$\frac{\partial \boldsymbol{f}_{c,j}^2}{\partial \boldsymbol{q}^2} = -c_p \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{q}^2} (\boldsymbol{x}_c^1 - \boldsymbol{s}^2) - c_p \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T} \left(\frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{q}^2} - \frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2}\right) \tag{A.39}$$

Where the partial derivatives have been determined above.

A.2 Derivative of the generalized contact force

Having obtained the physical contact Jacobian, the generalized contact Jacobian can be computed using Eq. A.4. This equation contains two derivatives w.r.t. the angular coordinates, which in this dissertation are the Bryant angles (see Appendix C). These derivatives are computed in this section. The first derivative in Eq. A.4 is

$$\frac{\partial\left(\overline{\boldsymbol{G}}^T \widetilde{\overline{\boldsymbol{u}}}^i \boldsymbol{A}^T\right)}{\partial \boldsymbol{\theta}} \tag{A.40}$$

which is a derivative of a 3×3 matrix w.r.t. a 3×1 vector, thus yielding a $3 \times 3 \times 3$ array that is composed as:

$$\frac{\partial\left(\overline{\boldsymbol{G}}^T\widetilde{\overline{\boldsymbol{u}}}^i\boldsymbol{A}^T\right)}{\partial\boldsymbol{\theta}}=\left[\frac{\partial\left(\overline{\boldsymbol{G}}^T\widetilde{\overline{\boldsymbol{u}}}^i\boldsymbol{A}^T\right)}{\partial\phi}\ \vdots\ \frac{\partial\left(\overline{\boldsymbol{G}}^T\widetilde{\overline{\boldsymbol{u}}}^i\boldsymbol{A}^T\right)}{\partial\psi}\ \vdots\ \frac{\partial\left(\overline{\boldsymbol{G}}^T\widetilde{\overline{\boldsymbol{u}}}^i\boldsymbol{A}^T\right)}{\partial\theta}\right] \tag{A.41}$$

Using the small-angle approximation (see Appendix C), the $\overline{\boldsymbol{G}}$ and $\boldsymbol{A}$ matrices are approximated by

$$\overline{\boldsymbol{G}}\approx\begin{bmatrix}cos\,\theta & sin\,\theta & 0\\ -sin\,\theta & cos\,\theta & 0\\ \psi & 0 & 1\end{bmatrix},\qquad \boldsymbol{A}\approx\begin{bmatrix}cos\,\theta & -sin\,\theta & \psi\\ sin\,\theta & cos\,\theta & -\phi\\ -\psi\,cos\,\theta+\phi\,sin\,\theta & \psi\,sin\,\theta+\phi\,cos\,\theta & 1\end{bmatrix} \tag{A.42}$$

The matrix product $\overline{\boldsymbol{G}}^T\widetilde{\overline{\boldsymbol{u}}}^i\boldsymbol{A}^T$ can then be computed from (taking into account the small-angle approximation):

$$\begin{aligned}&\overline{\boldsymbol{G}}^T\widetilde{\overline{\boldsymbol{u}}}^i\boldsymbol{A}^T\\ &=\begin{bmatrix}cos\,\theta & sin\,\theta & 0\\ -sin\,\theta & cos\,\theta & 0\\ \psi & 0 & 1\end{bmatrix}^T\begin{bmatrix}0 & -\overline{u}_3^i & \overline{u}_2^i\\ \overline{u}_3^i & 0 & -\overline{u}_1^i\\ -\overline{u}_2^i & \overline{u}_1^i & 0\end{bmatrix}\begin{bmatrix}cos\,\theta & -sin\,\theta & \psi\\ sin\,\theta & cos\,\theta & -\phi\\ -\psi\,cos\,\theta+\phi\,sin\,\theta & \psi\,sin\,\theta+\phi\,cos\,\theta & 1\end{bmatrix}^T\\ &=\begin{bmatrix}0 & \overline{u}_1^i(\psi c\theta-\phi s\theta)-\overline{u}_2^i(\psi s\theta+\phi c\theta)-\overline{u}_3^i & \overline{u}_1^i s\theta+\overline{u}_2^i c\theta-\overline{u}_3^i\phi\\ -\overline{u}_1^i\psi c\theta+\overline{u}_2^i\psi s\theta+\overline{u}_3^i & \overline{u}_1^i\phi c\theta-\overline{u}_2^i\phi s\theta & -\overline{u}_1^i c\theta+\overline{u}_2^i s\theta-\overline{u}_3^i\psi\\ -\overline{u}_1^i s\theta-\overline{u}_2^i c\theta & \overline{u}_1^i c\theta-\overline{u}_2^i s\theta & \overline{u}_1^i(\psi s\theta+\phi c\theta)+\overline{u}_2^i(\psi c\theta-\phi s\theta)\end{bmatrix}\end{aligned} \tag{A.43}$$

The derivatives in Eq. A.41 are now obtained by differentiating Eq. A.43 w.r.t. the respective Bryant angles:

$$\frac{\partial\left(\overline{\boldsymbol{G}}^T\widetilde{\overline{\boldsymbol{u}}}^i\boldsymbol{A}^T\right)}{\partial\phi}=\begin{bmatrix}0 & -\overline{u}_1^i\,sin\,\theta-\overline{u}_2^i\,cos\,\theta & -\overline{u}_3^i\\ 0 & \overline{u}_1^i\,cos\,\theta-\overline{u}_2^i\,sin\,\theta & 0\\ 0 & 0 & \overline{u}_1^i\,cos\,\theta-\overline{u}_2^i\,sin\,\theta\end{bmatrix} \tag{A.44}$$

$$\frac{\partial\left(\overline{\boldsymbol{G}}^T\widetilde{\overline{\boldsymbol{u}}}^i\boldsymbol{A}^T\right)}{\partial\psi}=\begin{bmatrix}0 & \overline{u}_1^i\,cos\,\theta-\overline{u}_2^i\,sin\,\theta & 0\\ -\overline{u}_1^i c\theta+\overline{u}_2^i s\theta & 0 & -\overline{u}_3^i\\ 0 & 0 & \overline{u}_1^i\,sin\,\theta+\overline{u}_2^i\,cos\,\theta\end{bmatrix} \tag{A.45}$$

$$\begin{aligned}&\frac{\partial\left(\overline{\boldsymbol{G}}^T\widetilde{\overline{\boldsymbol{u}}}^i\boldsymbol{A}^T\right)}{\partial\theta}\\ &=\begin{bmatrix}0 & -\overline{u}_1^i(\psi s\theta+\phi c\theta)-\overline{u}_2^i(\psi c\theta-\phi s\theta) & \overline{u}_1^i c\theta-\overline{u}_2^i s\theta\\ \overline{u}_1^i\psi s\theta+\overline{u}_2^i\psi c\theta & -\overline{u}_1^i\phi s\theta-\overline{u}_2^i\phi c\theta & \overline{u}_1^i s\theta+\overline{u}_2^i c\theta\\ -\overline{u}_1^i c\theta+\overline{u}_2^i s\theta & -\overline{u}_1^i s\theta-\overline{u}_2^i c\theta & \overline{u}_1^i(\psi c\theta-\phi s\theta)-\overline{u}_2^i(\psi s\theta+\phi c\theta)\end{bmatrix}\end{aligned} \tag{A.46}$$

The second derivative in Eq. A.4 is

$$\frac{\partial \boldsymbol{A}^T}{\partial \boldsymbol{\theta}} = \left[\frac{\partial \boldsymbol{A}^T}{\partial \phi} \;\vdots\; \frac{\partial \boldsymbol{A}^T}{\partial \psi} \;\vdots\; \frac{\partial \boldsymbol{A}^T}{\partial \theta}\right] = \left[\begin{bmatrix} 0 & 0 & \sin\theta \\ 0 & 0 & \cos\theta \\ 0 & -1 & 0 \end{bmatrix} \vdots \begin{bmatrix} 0 & 0 & -\cos\theta \\ 0 & 0 & \sin\theta \\ 1 & 0 & 0 \end{bmatrix} \vdots \begin{bmatrix} -\sin\theta & \cos\theta & \psi\sin\theta + \phi\cos\theta \\ -\cos\theta & -\sin\theta & \psi\cos\theta - \phi\sin\theta \\ 0 & 0 & 0 \end{bmatrix}\right] \tag{A.47}$$

A.3 Further approximations to simplify the contact Jacobian

One approximation that is often applied in contact problems based on quadrilateral segments (such as the face of an eight-noded hexahedral solid element), is that, even though this segment is not flat, the normal direction to this segment is considered as constant and equal to the exact normal computed at the center of the segment ($\xi = \eta = 0$) [182]. Using this approximation, Eq. A.15, A.31 and A.39 reduce to:

$$\frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{q}^1} = \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{n}_c^1}\frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{q}^1} = \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{n}_c^1}\frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{N}_c^1}\frac{\partial \boldsymbol{N}_c^1}{\partial \boldsymbol{q}^1} = \frac{\partial \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}}{\partial \boldsymbol{n}_c^1}\frac{\partial \boldsymbol{n}_c^1}{\partial \boldsymbol{N}_c^1}\sum_{l=1}^{4}\left(\frac{\partial \boldsymbol{N}_c^1}{\partial \boldsymbol{x}^{1l}}\frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1}\right) \tag{A.48}$$

$$\frac{\partial \boldsymbol{f}_{c,j}^{1,l}}{\partial \boldsymbol{q}^2} = c_p \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}(\boldsymbol{x}_c^1 - \boldsymbol{s}^2)\frac{\partial w_c^l}{\partial \boldsymbol{q}^2} + c_p w_c^l \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}\left(\frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{q}^2} - \frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2}\right) \tag{A.49}$$

$$\frac{\partial \boldsymbol{f}_{c,j}^{2}}{\partial \boldsymbol{q}^2} = -c_p \boldsymbol{n}_c^1 \boldsymbol{n}_c^{1^T}\left(\frac{\partial \boldsymbol{x}_c^1}{\partial \boldsymbol{q}^2} - \frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2}\right) \tag{A.50}$$

A.4 Analytical contact Jacobian for planar rigid body rotation

In section 2.4.3, the equations of motion of flexible multibody systems having only one rotational (and zero translational) rigid body degrees of freedom are derived. The analytical contact Jacobian for such a system can be obtained using the equations presented in the previous sections. In the case of planar rigid body rotation, the vector of generalized contact forces (Eq. A.1) is given by

$$\boldsymbol{Q}_c^i = \sum_{j=1}^{n_F} \boldsymbol{Q}_{c,j}^i = \sum_{j=1}^{n_F} \begin{Bmatrix} \overline{\boldsymbol{u}}^{ij^T} \boldsymbol{B}^{i^T} \boldsymbol{f}_{c,j}^i \\ \overline{\boldsymbol{V}}^{ij^T} \boldsymbol{A}^{i^T} \boldsymbol{f}_{c,j}^i \end{Bmatrix} \tag{A.51}$$

where the matrix $\boldsymbol{B}^i$ is given by

$$\boldsymbol{B}^i \equiv \begin{bmatrix} -\sin\theta & -\cos\theta & 0 \\ \cos\theta & -\sin\theta & 0 \\ 0 & 0 & 0 \end{bmatrix}^i \tag{A.52}$$

The derivative of the vector w.r.t. the generalized coordinates of the multibody system (Eq. A.4) is

$$\boldsymbol{J}_{c,j}^{k,i} = \frac{\partial \boldsymbol{Q}_{c,j}^k}{\partial \boldsymbol{q}^i} = \begin{bmatrix} \overline{\boldsymbol{u}}^{kj^T} \boldsymbol{B}_\theta^{k^T} \boldsymbol{f}_{c,j}^k + \overline{\boldsymbol{u}}^{kj^T} \boldsymbol{B}^{k^T} \dfrac{\partial \boldsymbol{f}_{c,j}^k}{\partial \theta^i} & \overline{\boldsymbol{V}}^{kj^T} \boldsymbol{B}^{k^T} \boldsymbol{f}_{c,j}^k + \overline{\boldsymbol{u}}^{kj^T} \boldsymbol{B}^{k^T} \dfrac{\partial \boldsymbol{f}_{c,j}^k}{\partial \boldsymbol{q}_f^i} \\ \overline{\boldsymbol{V}}^{kj^T} \boldsymbol{B}^{k^T} \boldsymbol{F}_{c,j}^k + \overline{\boldsymbol{V}}^{kj^T} \boldsymbol{A}^{k^T} \dfrac{\partial \boldsymbol{f}_{c,j}^k}{\partial \theta^i} & \overline{\boldsymbol{V}}^{kj^T} \boldsymbol{A}^{k^T} \dfrac{\partial \boldsymbol{f}_{c,j}^k}{\partial \boldsymbol{q}_f^i} \end{bmatrix} \tag{A.53}$$

where the derivative of $\boldsymbol{B}^k$ w.r.t. the rigid body rotation angle θ is

$$\boldsymbol{B}_\theta^k \equiv \begin{bmatrix} -\cos\theta & \sin\theta & 0 \\ -\sin\theta & -\cos\theta & 0 \\ 0 & 0 & 0 \end{bmatrix}^k \tag{A.54}$$

Finally, the derivatives of the master and slave node indices w.r.t. the generalized coordinates (Eq. A.30 and A.35) reduce to:

$$\frac{\partial \boldsymbol{x}^{1l}}{\partial \boldsymbol{q}^1} = [\boldsymbol{B}^1 \overline{\boldsymbol{u}}^{l1} \quad \boldsymbol{A}^1 \overline{\boldsymbol{V}}^{l1}] \quad , \quad \frac{\partial \boldsymbol{s}^2}{\partial \boldsymbol{q}^2} = [\boldsymbol{B}^2 \overline{\boldsymbol{u}}^2 \quad \boldsymbol{A}^2 \overline{\boldsymbol{V}}^2] \tag{A.55}$$

Appendix B

Elimination of the constraint equations

The dynamics of a flexible body that is subjected to certain constraints $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0} \in \mathbb{R}^{n_\lambda}$ is described by a coupled system of differential and algebraic equations, see Eq. 2.4.16. While many techniques have been developed to solve these DAEs (see Section 2.6), it is often desirable to convert Eq. 2.4.16 into a system of Ordinary Differential Equations (ODEs) by elimination of the constraint equations. The latter can be achieved using the method of Generalized Coordinate Partitioning (GCP) [183]. Assuming the constraint equations $\boldsymbol{g} = \boldsymbol{0}$ are linearly independent (and therefore $\mathrm{rank}(\boldsymbol{G}) = n_\lambda$, where $\boldsymbol{G} = \boldsymbol{g}_{\boldsymbol{q}}$), it is possible to identify $n_q - n_\lambda$ independent coordinates and partition the vector of generalized coordinates accordingly:

$$\boldsymbol{q} = \begin{Bmatrix} \boldsymbol{q}_i \\ \boldsymbol{q}_d \end{Bmatrix} \tag{B.1}$$

where $\boldsymbol{q}_i$ and $\boldsymbol{q}_d$ are, respectively, the vectors of independent and dependent coordinates. In general-purpose multibody software, this partitioning is usually performed automatically by factorizing the constraint Jacobian (see e.g. [184], Appendix B, where the LU-factorization of $\boldsymbol{G}$ is used). The relation between $\boldsymbol{q}_i$ and $\boldsymbol{q}_d$ is given by the (generally nonlinear) constraint equation $\boldsymbol{g}(\boldsymbol{q}_i, \boldsymbol{q}_d) = \boldsymbol{0}$. The relations between the independent and dependent velocities and accelerations can be obtained by deriving the constraint equations with respect to time. Using the partitioning of Eq. B.1, the first and second time derivatives of $\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0}$ can be written as

$$\boldsymbol{g}_{\boldsymbol{q}}\dot{\boldsymbol{q}} = \boldsymbol{g}_{\boldsymbol{q}_i}\dot{\boldsymbol{q}}_i + \boldsymbol{g}_{\boldsymbol{q}_d}\dot{\boldsymbol{q}}_d = \boldsymbol{0} \tag{B.2a}$$

$$\left(\boldsymbol{g}_{\boldsymbol{q}}\dot{\boldsymbol{q}}\right)_{\boldsymbol{q}}\dot{\boldsymbol{q}} + \boldsymbol{g}_{\boldsymbol{q}}\ddot{\boldsymbol{q}} = \left(\boldsymbol{g}_{\boldsymbol{q}}\dot{\boldsymbol{q}}\right)_{\boldsymbol{q}}\dot{\boldsymbol{q}} + \boldsymbol{g}_{\boldsymbol{q}_i}\ddot{\boldsymbol{q}}_i + \boldsymbol{g}_{\boldsymbol{q}_d}\ddot{\boldsymbol{q}}_d = \boldsymbol{\kappa} + \boldsymbol{g}_{\boldsymbol{q}_i}\ddot{\boldsymbol{q}}_i + \boldsymbol{g}_{\boldsymbol{q}_d}\ddot{\boldsymbol{q}}_d = \boldsymbol{0} \tag{B.2b}$$

where $\boldsymbol{\kappa} = \left(\boldsymbol{g}_{\boldsymbol{q}}\dot{\boldsymbol{q}}\right)_{\boldsymbol{q}}\dot{\boldsymbol{q}}$. From these equations, the following expressions for the dependent velocities and accelerations are obtained:

$$\dot{\boldsymbol{q}}_d = -\boldsymbol{g}_{\boldsymbol{q}_d}^{-1}\boldsymbol{g}_{\boldsymbol{q}_i}\dot{\boldsymbol{q}}_i = \boldsymbol{g}_{di}\dot{\boldsymbol{q}}_i \tag{B.3a}$$

$$\ddot{\boldsymbol{q}}_d = -\boldsymbol{g}_{\boldsymbol{q}_d}^{-1}\left(\boldsymbol{\kappa} + \boldsymbol{g}_{\boldsymbol{q}_i}\ddot{\boldsymbol{q}}_i\right) = -\boldsymbol{g}_{\boldsymbol{q}_d}^{-1}\boldsymbol{g}_{\boldsymbol{q}_i}\ddot{\boldsymbol{q}}_i - \boldsymbol{g}_{\boldsymbol{q}_d}^{-1}\boldsymbol{\kappa} \tag{B.3b}$$

Eq. B.3 allows the total vector of generalized accelerations $\ddot{\boldsymbol{q}}$ to be written in terms of the independent accelerations $\ddot{\boldsymbol{q}}_i$ as follows:

$$\ddot{\boldsymbol{q}} = \begin{Bmatrix} \ddot{\boldsymbol{q}}_i \\ \ddot{\boldsymbol{q}}_d \end{Bmatrix} = \begin{bmatrix} \boldsymbol{I}_{n_q - n_\lambda} \\ -\boldsymbol{g}_{\boldsymbol{q}_d}^{-1} \boldsymbol{g}_{\boldsymbol{q}_i} \end{bmatrix} \ddot{\boldsymbol{q}}_i + \begin{Bmatrix} \boldsymbol{0} \\ -\boldsymbol{g}_{\boldsymbol{q}_d}^{-1} \boldsymbol{\kappa} \end{Bmatrix} = \boldsymbol{T}_{di} \ddot{\boldsymbol{q}}_i + \bar{\boldsymbol{\kappa}} \tag{B.4}$$

where

$$\boldsymbol{T}_{di} = \begin{bmatrix} \boldsymbol{I}_{n_q - n_\lambda} \\ -\boldsymbol{g}_{\boldsymbol{q}_d}^{-1} \boldsymbol{g}_{\boldsymbol{q}_i} \end{bmatrix} \in \mathbb{R}^{n_q \times (n_q - n_\lambda)} \quad , \quad \bar{\boldsymbol{\kappa}} = \begin{Bmatrix} \boldsymbol{0} \\ -\boldsymbol{g}_{\boldsymbol{q}_d}^{-1} \boldsymbol{\kappa} \end{Bmatrix} \in \mathbb{R}^{n_q} \tag{B.5}$$

Assuming the vector of generalized coordinates is arranged according to the coordinate partitioning of Eq. B.1, Eq. B.4 can be substituted in Eq. 2.4.16 to yield

$$\boldsymbol{M}(\boldsymbol{T}_{di} \ddot{\boldsymbol{q}}_i + \bar{\boldsymbol{\kappa}}) + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} + \boldsymbol{G}^T \boldsymbol{\lambda} = \boldsymbol{f}_{ex} + \boldsymbol{f}_v \tag{B.6a}$$

$$\boldsymbol{g}(\boldsymbol{q}) = \boldsymbol{0} \tag{B.6b}$$

Pre-multiplying this equation by $\boldsymbol{T}_{di}^T$ and rearranging terms yields

$$\boldsymbol{T}_{di}^T \boldsymbol{M} \boldsymbol{T}_{di} \ddot{\boldsymbol{q}}_i + \boldsymbol{T}_{di}^T \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{T}_{di}^T \boldsymbol{K}\boldsymbol{q} + \boldsymbol{T}_{di}^T \boldsymbol{G}^T \boldsymbol{\lambda} = \boldsymbol{T}_{di}^T (\boldsymbol{f}_{ex} + \boldsymbol{f}_v) - \boldsymbol{T}_{di}^T \boldsymbol{M} \bar{\boldsymbol{\kappa}} \tag{B.7}$$

where $\boldsymbol{T}_{di}^T \boldsymbol{G}^T$ can be shown to be a null matrix, thereby eliminating the Lagrange multipliers from the above equations. This allows Eq. B.7 to be written as:

$$\boldsymbol{M}_{ii} \ddot{\boldsymbol{q}}_i = \boldsymbol{T}_{di}^T (\boldsymbol{f}_{ex} + \boldsymbol{f}_v - \boldsymbol{K}\boldsymbol{q} - \boldsymbol{D}\dot{\boldsymbol{q}} - \boldsymbol{M}\bar{\boldsymbol{\kappa}}) \tag{B.8}$$

which has the form of a system of $n_q - n_\lambda$ ODEs.

Although the numerical solution of ODEs is considerably less involved then the solution of a system of DAEs, solving Eq. B.8 can be troublesome for three reasons. First of all, the mass matrix $\boldsymbol{M}_{ii} = \boldsymbol{T}_{di}^T \boldsymbol{M} \boldsymbol{T}_{di}$ is a full matrix whereas $\boldsymbol{M}$ is a sparse matrix and thus can be inverted more efficiently. Secondly, Eq. B.8 requires more matrix multiplications and inversions then Eq. 2.6.45, most of which cannot be performed offline. Finally, evaluating the right-hand side of Eq. B.8 requires the full vector of generalized coordinates $\boldsymbol{q} = \{\boldsymbol{q}_i^T \quad \boldsymbol{q}_d^T\}^T$ and velocities $\dot{\boldsymbol{q}} = \{\dot{\boldsymbol{q}}_i^T \quad \dot{\boldsymbol{q}}_d^T\}^T$. The former is obtained by iteratively solving the constraint equation $\boldsymbol{g}(\boldsymbol{q}_i, \boldsymbol{q}_d) = \boldsymbol{0}$, e.g. using Newton's method:

$$\boldsymbol{q}_d^{k+1} = \boldsymbol{q}_d^k - \boldsymbol{g}_{\boldsymbol{q}_d}^{-1} \boldsymbol{g}(\boldsymbol{q}_i, \boldsymbol{q}_d^k) \tag{B.9}$$

The full vector of generalized velocities, on the other hand, is obtained using Eq. B.2b.

Appendix C

Parameterization of rotations

In Section 2.4 the specific choice of the rotational parameters that describe the orientation of the body-attached reference frame was left undefined. In order to define these parameters, note that any rotation of a vector $\bar{\boldsymbol{r}}$ in space is uniquely determined by its axis of rotation $\boldsymbol{e}_\vartheta$ and the rotation angle ϑ about this axis. The latter is expressed by the Rodrigues formula [6]:

$$\boldsymbol{r} = \left[\boldsymbol{I}_3 + \tilde{\boldsymbol{e}}_\vartheta \, sin\,\vartheta + 2(\tilde{\boldsymbol{e}}_\vartheta)^2 \, sin^2\frac{\vartheta}{2}\right]\bar{\boldsymbol{r}} = \boldsymbol{A}(\boldsymbol{e}_\vartheta, \vartheta)\bar{\boldsymbol{r}} \tag{C.1}$$

where $\boldsymbol{r}$ is the rotated vector and $\boldsymbol{A} \in \mathbb{R}^{3\times3}$ is the associated transformation matrix (see Eq. 2.4.3). Since $\boldsymbol{e}_\vartheta$ is a unit vector (and therefore uniquely defined by specifying two of its components), the transformation matrix $\boldsymbol{A}$ can be expressed in terms of three independent parameters. Various choices of parameter sets are possible, including Euler parameters, Euler angles, Rodriguez parameters, Bryant angles, and direction cosines. Among these, the Euler parameters and Bryant angles are of particular importance to the current work.

Bryant angles define the orientation of a body in space by specifying three successive rotations about three axes, where the angles of rotation are referred to as the Bryant angles. With reference to Fig. C.1, the first rotation is carried out counterclockwise about the $\boldsymbol{X}_1$ axis through an angle $\boldsymbol{\phi}$, resulting in the reference system $\boldsymbol{x}_1''\boldsymbol{x}_2''\boldsymbol{x}_3''$; the second rotation produces the reference system $\boldsymbol{x}_1'\boldsymbol{x}_2'\boldsymbol{x}_3'$ by counterclockwise rotation about $\boldsymbol{x}_2''$ through an angle $\boldsymbol{\psi}$; finally, the body-attached reference system $\boldsymbol{x}_1\boldsymbol{x}_2\boldsymbol{x}_3$ is obtained by counterclockwise rotation about $\boldsymbol{x}_3'$ through an angle θ. Let $\boldsymbol{A}_1 = \boldsymbol{A}(\boldsymbol{X}_1, \boldsymbol{\phi})$, $\boldsymbol{A}_2 = \boldsymbol{A}(\boldsymbol{x}_2'', \boldsymbol{\psi})$, and $\boldsymbol{A}_3 = \boldsymbol{A}(\boldsymbol{x}_3', \theta)$ denote the transformation matrices of the respective rotations, then the total transformation matrix can be obtained from Eq. C.1 as

$$\boldsymbol{A} = \boldsymbol{A}(\boldsymbol{X}_1, \boldsymbol{\phi})\boldsymbol{A}(\boldsymbol{x}_2'', \boldsymbol{\psi})\boldsymbol{A}(\boldsymbol{x}_3', \theta) = \begin{bmatrix} c\boldsymbol{\psi}c\theta & -c\boldsymbol{\psi}s\theta & s\boldsymbol{\psi} \\ c\boldsymbol{\phi}s\theta + s\boldsymbol{\phi}s\boldsymbol{\psi}c\theta & c\boldsymbol{\phi}c\theta - s\boldsymbol{\phi}s\boldsymbol{\psi}s\theta & -s\boldsymbol{\phi}c\boldsymbol{\psi} \\ s\boldsymbol{\phi}s\theta - c\boldsymbol{\phi}s\boldsymbol{\psi}c\theta & s\boldsymbol{\phi}c\theta + c\boldsymbol{\phi}s\boldsymbol{\psi}s\theta & c\boldsymbol{\phi}c\boldsymbol{\psi} \end{bmatrix} \tag{C.2}$$

where '*s*' denotes the sine and '*c*' the cosine of the angle. The relationship between the angular velocity $\bar{\boldsymbol{\omega}}$ and the time derivatives of the Bryant angles $\boldsymbol{\theta} = \{\boldsymbol{\phi}, \boldsymbol{\psi}, \theta\}$ (Eq. 2.81) is given by (see [184]):

$$\bar{\boldsymbol{\omega}} = \bar{\boldsymbol{G}}\dot{\boldsymbol{\theta}} = \begin{bmatrix} c\phi c\theta & s\theta & 0 \\ -c\psi s\theta & c\theta & 0 \\ s\psi & 0 & 1 \end{bmatrix} \begin{Bmatrix} \dot{\phi} \\ \dot{\psi} \\ \dot{\theta} \end{Bmatrix} \approx \begin{Bmatrix} \dot{\phi} \\ \dot{\psi} \\ \dot{\theta} \end{Bmatrix} \tag{C.3}$$

where the approximation sign '≈' holds only in the case of infinitesimal rotations.

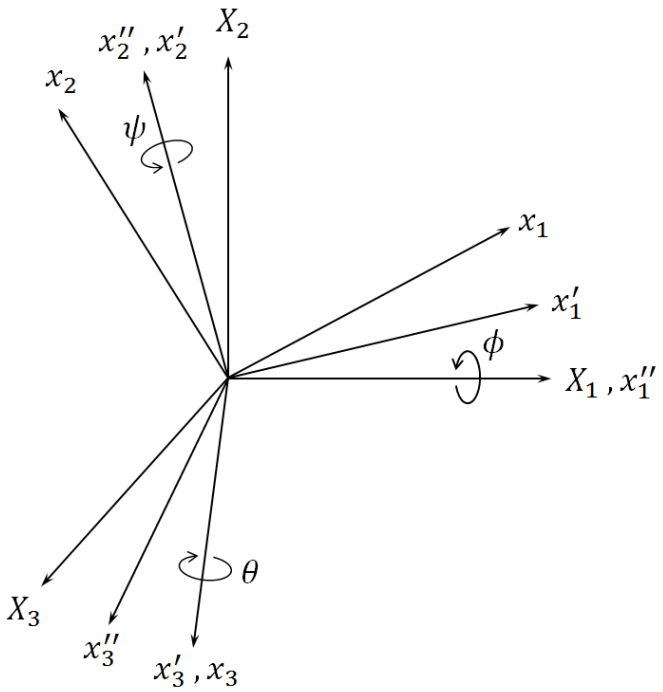


Fig. C.1 Bryant rotation angles convention

Note from Eq. C.3 that $\bar{\boldsymbol{G}}$ becomes singular when $\phi = \pi/2$. In that case, the axes of the first and third rotations in the Bryant sequence coincide, so that the rotation angles ϕ and θ become indistinguishable. Similar singularity problems are encountered for Euler angles and other three-parameter sets of rotational coordinates. In order to circumvent such singularity conditions, the redundant set of Euler parameters $\boldsymbol{\theta} = \{e_0, e_1, e_2, e_3\}$ can be used to parameterize rotations. Euler parameters have the advantage of yielding simpler expressions for the transformation matrix and quadratic velocity vector, while at the same time avoiding singularities in the configuration of the body. These advantages however come at the price of an algebraic constraint equation $\boldsymbol{g}(\boldsymbol{\theta}) = \boldsymbol{\theta}^T\boldsymbol{\theta} - 1 = 0$ that needs to be satisfied at all times, thereby complicating the numerical solution of the body's EOMs (Eq. 2.4.16).

Appendix D

Equations of motion in terms of mass invariants

D.1 General three-dimensional motion

In Section 2.4, the EOMs of a spatial flexible body are derived and expressed in terms of the body's mass matrix (Eq. 2.4.11) and quadratic velocity vector (Eq. 2.4.18). These quantities are formulated in terms of summations that need to be carried out over all the nodes in the FE representation of the flexible body. In the numerical implementation of the EOMs, these summations are computed only once prior to the actual simulation. In doing so, the mass matrix and quadratic velocity vector can be formulated in terms of these so-called 'mass invariants' as follows:

$$\boldsymbol{M} = \begin{bmatrix} m\boldsymbol{I}_3 & -\boldsymbol{A}\left(\boldsymbol{M}_2 + \boldsymbol{M}_7\boldsymbol{q}_f\right)\overline{\boldsymbol{G}} & \boldsymbol{A}\boldsymbol{M}_6 \\ & -\overline{\boldsymbol{G}}^T\left[\boldsymbol{M}_3 + \boldsymbol{q}_f\left(\boldsymbol{M}_5 + \boldsymbol{M}_{10}\boldsymbol{q}_f\right)\right]\overline{\boldsymbol{G}} & \overline{\boldsymbol{G}}^T\left[\boldsymbol{M}_4 + \boldsymbol{M}_9\boldsymbol{q}_f\right] \\ sym. & & \boldsymbol{M}_8 \end{bmatrix} \tag{D.1}$$

$$\boldsymbol{f}_v = \begin{Bmatrix} -\boldsymbol{A}\tilde{\bar{\boldsymbol{\omega}}}\tilde{\bar{\boldsymbol{\omega}}}\left(\boldsymbol{M}_1 + \boldsymbol{M}_6\boldsymbol{q}_f\right) - 2\boldsymbol{A}\tilde{\bar{\boldsymbol{\omega}}}\boldsymbol{M}_6\dot{\boldsymbol{q}}_f + \boldsymbol{A}\left(\boldsymbol{M}_2 + \boldsymbol{M}_7\boldsymbol{q}_f\right)\dot{\overline{\boldsymbol{G}}}\dot{\boldsymbol{\theta}} \\ \overline{\boldsymbol{G}}^T\tilde{\bar{\boldsymbol{\omega}}}\left[\boldsymbol{M}_3 + \boldsymbol{q}_f\left(\boldsymbol{M}_5 + \boldsymbol{M}_{10}\boldsymbol{q}_f\right)\right]\bar{\boldsymbol{\omega}} + 2\overline{\boldsymbol{G}}^T\left[\boldsymbol{M}_{51} + \boldsymbol{M}_{10}\boldsymbol{q}_f\right]\dot{\boldsymbol{q}}_f\bar{\boldsymbol{\omega}} + \overline{\boldsymbol{G}}^T\left[\boldsymbol{M}_3 + \boldsymbol{q}_f\left(\boldsymbol{M}_5 + \boldsymbol{M}_{10}\boldsymbol{q}_f\right)\right]\dot{\overline{\boldsymbol{G}}}\dot{\boldsymbol{\theta}} \\ \boldsymbol{V}^{i\,T}\tilde{\bar{\boldsymbol{\omega}}}\tilde{\bar{\boldsymbol{r}}}^i\bar{\boldsymbol{\omega}} + 2\boldsymbol{V}^{i\,T}\tilde{\overline{\boldsymbol{V}}}^i\dot{\boldsymbol{q}}_f\bar{\boldsymbol{\omega}} + \boldsymbol{V}^{i\,T}\tilde{\bar{\boldsymbol{r}}}^i\dot{\overline{\boldsymbol{G}}}\dot{\boldsymbol{\theta}} \end{Bmatrix} \tag{D.2}$$

where m is the mass of the body and the mass invariants are defined as

$$\begin{aligned}
\boldsymbol{M}_1 &= \sum_{i=1}^{n_n} m^i\bar{\boldsymbol{r}}_0^i \in \mathbb{R}^{3\times 1} & \boldsymbol{M}_6 &= \sum_{i=1}^{n_n} m^i\overline{\boldsymbol{V}}^i \in \mathbb{R}^{3\times n_f} \\
\boldsymbol{M}_2 &= \sum_{i=1}^{n_n} m^i\tilde{\bar{\boldsymbol{r}}}_0^i \in \mathbb{R}^{3\times 3} & \boldsymbol{M}_7 &= \sum_{i=1}^{n_n} m^i\tilde{\overline{\boldsymbol{V}}}^i \in \mathbb{R}^{3\times 3\times n_f} \\
\boldsymbol{M}_3 &= \sum_{i=1}^{n_n} m^i\tilde{\bar{\boldsymbol{r}}}_0^i\tilde{\bar{\boldsymbol{r}}}_0^i \in \mathbb{R}^{3\times 3} & \boldsymbol{M}_8 &= \sum_{i=1}^{n_n} m^i\overline{\boldsymbol{V}}^{i\,T}\overline{\boldsymbol{V}}^i \in \mathbb{R}^{n_f\times n_f} \\
\boldsymbol{M}_4 &= \sum_{i=1}^{n_n} m^i\tilde{\bar{\boldsymbol{r}}}_0^i\overline{\boldsymbol{V}}^i \in \mathbb{R}^{3\times n_f} & \boldsymbol{M}_9 &= \sum_{i=1}^{n_n} m^i\tilde{\overline{\boldsymbol{V}}}^i\overline{\boldsymbol{V}}^i \in \mathbb{R}^{3\times n_f\times n_f}
\end{aligned} \tag{D.3}$$

$$\boldsymbol{M}_5 = \sum_{i=1}^{n_n} m^i \left(\tilde{\bar{\boldsymbol{r}}}_0^i \tilde{\bar{\boldsymbol{V}}}^i + \tilde{\bar{\boldsymbol{V}}}^i \tilde{\bar{\boldsymbol{r}}}_0^i\right) \in \mathbb{R}^{3\times 3\times n_f} \quad \boldsymbol{M}_{10} = \sum_{i=1}^{n_n} m^i \tilde{\bar{\boldsymbol{V}}}^i \tilde{\bar{\boldsymbol{V}}}^i \in \mathbb{R}^{3\times 3\times n_f\times n_f}$$

D.2 Rigid-body motion in a plane

When the flexible body's rigid body motion is constrained to a plane, the body's mass matrix (Eq. 2.4.28) and quadratic velocity vector (Eq. 2.4.29) can be expressed in terms of mass invariants as follows:

$$\boldsymbol{M} = \begin{bmatrix} m\boldsymbol{I}_3 & \boldsymbol{B}\left(\boldsymbol{M}_1^\theta + \boldsymbol{M}_5^\theta \boldsymbol{q}_f\right) & \boldsymbol{A}\boldsymbol{M}_5^\theta \\ & \boldsymbol{M}_2^\theta + 2\boldsymbol{M}_3^\theta \boldsymbol{q}_f + \boldsymbol{q}_f^T \boldsymbol{M}_6^\theta \boldsymbol{q}_f & \boldsymbol{M}_4^\theta + \boldsymbol{q}_f^T \boldsymbol{M}_7^\theta \\ sym. & & \boldsymbol{M}_8^\theta \end{bmatrix} \tag{D.4}$$

$$\boldsymbol{f}_v = \begin{Bmatrix} -\left(\dot{\theta}\right)^2 \tilde{\boldsymbol{B}}\left(\boldsymbol{M}_1^\theta + \boldsymbol{M}_5^\theta \boldsymbol{q}_f\right) - 2\dot{\theta}\boldsymbol{B}\boldsymbol{M}_5^\theta \dot{\boldsymbol{q}}_f \\ -2\dot{\theta}\left[\dot{\boldsymbol{q}}_f^T \left(\boldsymbol{M}_3^{\theta\,T} + \boldsymbol{M}_6^\theta \boldsymbol{q}_f\right)\right] - \dot{\boldsymbol{q}}_f^T \boldsymbol{M}_7^\theta \dot{\boldsymbol{q}}_f \\ \left(\dot{\theta}\right)^2 \left(\boldsymbol{M}_3^{\theta\,T} + \boldsymbol{M}_6^\theta \boldsymbol{q}_f\right) + 2\dot{\theta}\boldsymbol{M}_7^\theta \dot{\boldsymbol{q}}_f \end{Bmatrix} \tag{D.5}$$

where the mass invariants are defined as

$$\begin{aligned} \boldsymbol{M}_1^\theta &= \sum_{i=1}^{n_n} m^i \bar{\boldsymbol{r}}_0^i \in \mathbb{R}^{3\times 1} & \boldsymbol{M}_5^\theta &= \sum_{i=1}^{n_n} m^i \bar{\boldsymbol{V}}^i \in \mathbb{R}^{3\times n_f} \\ \boldsymbol{M}_2^\theta &= \sum_{i=1}^{n_n} m^i \bar{\boldsymbol{r}}_0^{i\,T} \bar{\boldsymbol{r}}_{0,xy}^i \in \mathbb{R}^1 & \boldsymbol{M}_6^\theta &= \sum_{i=1}^{n_n} m^i \bar{\boldsymbol{V}}^{i\,T} \bar{\boldsymbol{V}}_{xy}^i \in \mathbb{R}^{n_f\times n_f} \\ \boldsymbol{M}_3^\theta &= \sum_{i=1}^{n_n} m^i \bar{\boldsymbol{r}}_0^{i\,T} \bar{\boldsymbol{V}}_{xy}^i \in \mathbb{R}^{1\times n_f} & \boldsymbol{M}_7^\theta &= \sum_{i=1}^{n_n} m^i \bar{\boldsymbol{V}}^{i\,T} \bar{\boldsymbol{V}}_{y-x}^i \in \mathbb{R}^{n_f\times n_f} \\ \boldsymbol{M}_4^\theta &= \sum_{i=1}^{n_n} m^i \bar{\boldsymbol{r}}_0^{i\,T} \bar{\boldsymbol{V}}_{y-x}^i \in \mathbb{R}^{1\times n_f} & \boldsymbol{M}_8^\theta &= \sum_{i=1}^{n_n} m^i \bar{\boldsymbol{V}}^{i\,T} \bar{\boldsymbol{V}}^i \in \mathbb{R}^{n_f\times n_f} \end{aligned} \tag{D.6}$$

In these definitions, the subscript 'xy' denotes setting the third component of the associated vector to zero, while the subscript '$y-x$' means setting the third component to zero, inverting the sign of the first component and swapping the first and second components. These actions represent the pre-multiplications of a vector with $\boldsymbol{B}^T\boldsymbol{B}$ and $\boldsymbol{B}^T\boldsymbol{A}$, respectively:

$$\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{r} = \begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{Bmatrix}r_x\\ r_y\\ r_z\end{Bmatrix} = \begin{Bmatrix}r_x\\ r_y\\ 0\end{Bmatrix} = \boldsymbol{r}_{xy}\ ,$$
$$\boldsymbol{B}^T\boldsymbol{A}\boldsymbol{r} = \begin{bmatrix}0 & 1 & 0\\ -1 & 0 & 0\\ 0 & 0 & 0\end{bmatrix}\begin{Bmatrix}r_x\\ r_y\\ r_z\end{Bmatrix} = \begin{Bmatrix}r_y\\ -r_x\\ 0\end{Bmatrix} = \boldsymbol{r}_{y-x} \qquad \text{(D.7)}$$

Appendix E

Internal damping forces

When materials deform, internal frictional effects cause the material to absorb and dissipate energy. In engineering analysis, it is often assumed that this material damping can be represented by viscous (i.e. velocity-proportional) damping. In doing so, the finite element equations of motion (Eq. 2.2.2) are extended as follows:

$$\boldsymbol{M}_{FE}\ddot{\boldsymbol{u}}(t) + \boldsymbol{D}_{FE}\dot{\boldsymbol{u}}(t) + \boldsymbol{K}_{FE}\boldsymbol{u}(t) = \boldsymbol{f}_{FE}(t) \tag{E.1}$$

where $\boldsymbol{D}_{FE}$ denotes the finite element damping matrix. In principle this matrix could be derived by finite element procedures analogous to those employed in Section 2.2 for deriving the mass and stiffness matrices (see Eq. 2.2.9). However, the physical laws that govern this damping behavior are not sufficiently well understood to allow for such an approach, and furthermore, damping is more naturally defined at the component level rather than at the element level. A popular method to define the damping matrix $\boldsymbol{D}_{FE}$ is to employ proportional or Rayleigh damping, which is defined by

$$\boldsymbol{D}_{FE} = \alpha\boldsymbol{M}_{FE} + \beta\boldsymbol{K}_{FE} \tag{E.2}$$

The constants α and β are chosen so as to produce specified modal damping factors for two selected vibration modes of the FE model (see e.g. [4]):

$$\alpha = \frac{2\omega_1\omega_2}{\omega_2^2 - \omega_1^2}(\zeta_1\omega_2 - \zeta_2\omega_1) \quad , \quad \beta = \frac{2}{\omega_2^2 - \omega_1^2}(\zeta_2\omega_2 - \zeta_1\omega_1) \tag{E.3}$$

where ω_1 and ω_2 are the frequencies of the selected vibration modes and ζ_1 and ζ_2 are the corresponding modal damping factors. Mass-proportional damping is characterized by a decrease in damping as the frequency increases, while stiffness-proportional damping increases linearly with frequency. This is depicted in Fig. E.1.

As the damping matrix acts only on the generalized elastic velocities, the viscous damping force exerted on a flexible body undergoing large nonlinear rigid-body motion with superposed elastic deformation can be written as:

$$\boldsymbol{f}_d = \boldsymbol{D}\dot{\boldsymbol{q}} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} \\ & \mathbf{0} & \mathbf{0} \\ sym. & & \boldsymbol{D}_{VV} \end{bmatrix} \begin{Bmatrix} \boldsymbol{R} \\ \boldsymbol{\theta} \\ \boldsymbol{q}_f \end{Bmatrix} \tag{E.4}$$

where $\boldsymbol{D}_{VV} = \boldsymbol{V}^T \boldsymbol{D}_{FE} \boldsymbol{V}$, and $\boldsymbol{V}$ is the body's basis matrix (see Eq. 2.3.1). Taking into account this viscous damping term, the general form of the equations of motion (Eq. 2.4.16) can now be written in the following form:

$$\boldsymbol{M}\ddot{\boldsymbol{q}} + \boldsymbol{D}\dot{\boldsymbol{q}} + \boldsymbol{K}\boldsymbol{q} + \boldsymbol{G}^T\boldsymbol{\lambda} = \boldsymbol{f}_{ex} + \boldsymbol{f}_v \tag{E.5}$$

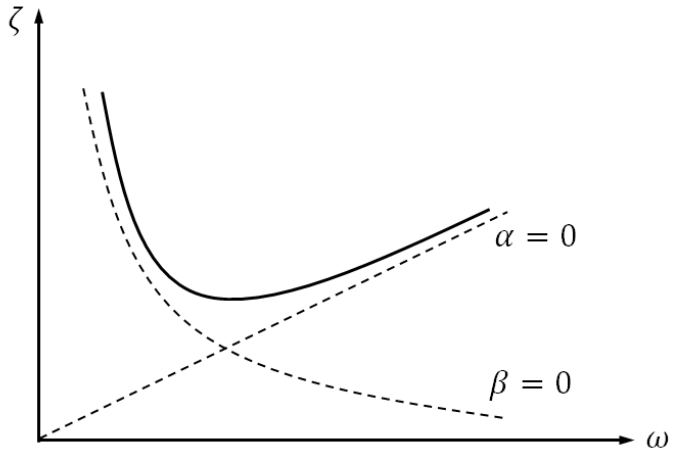


Fig. E.1 Rayleigh damping as a function of $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ coefficients

Appendix F

Frictional contact forces

In Section 2.5, frictional effects resulting from the relative tangential motion between two contact surfaces have been neglected. These effects, as well as the closely related topics of adhesion, wear, and lubrication, are investigated in the field of tribology, which attracted famous physicists like Aristotle, da Vinci, Coulomb and Tabor. Despite its long-standing history, many frictional phenomena have to this day not been completely understood. As these phenomena result from the interplay of mechanical, chemical and electromagnetic effects at the atomic level, first-principles modelling of friction is generally not feasible. Instead, heuristic approaches are used to describe frictional effects at the macroscopic level. In this section, the mathematical treatment of dry (i.e. non-lubricated) frictional contact interactions in mechanical systems is briefly discussed. For a comprehensive overview of the matter, the reader is referred to [185, 186, 187].

When frictional effects in the contact interface are considered, two cases can be distinguished: when there is no relative tangential motion between the two contact surfaces, the contact is said to be *adhesive* and the contact surfaces are said to be *sticking*; when instead the contact surfaces move in a tangential direction with respect to each other, they are said to be in *sliding* contact. The transition from *sticking* to *slipping* takes place once the tangential stress $\boldsymbol{t}_T$ in the contact interface exceed a certain limit. During slipping, the tangential stress is proportional to the normal contact stress $\boldsymbol{t}_N$ and acts in the opposite direction to the relative sliding velocity $\dot{\boldsymbol{g}}_T$ (to be defined later on). This is classically described by Coulomb's friction law, which can be expressed as

$$\boldsymbol{t}_T = -\mu_D |t_N| \frac{\dot{\boldsymbol{g}}_T}{\|\dot{\boldsymbol{g}}_T\|} \qquad if \qquad \|\boldsymbol{t}_T\| > \mu_S |t_N| \tag{F.1}$$

where μ_S and μ_D denote, respectively, the static and dynamic friction coefficients. In general, the friction coefficients depend on different parameters like surface rougness, normal contact pressure t_N, temperature, and relative sliding velocity $\dot{\boldsymbol{g}}_T$. A popular heuristic friction law that relates the static and dynamic friction coefficients is given by

$$\mu(\dot{\boldsymbol{g}}_T) = \mu_D + (\mu_S - \mu_D) e^{-c\|\dot{\boldsymbol{g}}_T\|} \tag{F.2}$$

where c is some constant. Note that the static friction coefficient is generally higher than the dynamic coefficient for the same materials. For steel-to-steel contact, the friction coefficients typically range from 0.8 (static friction) to 0.4 (dynamic friction) [188].

The contact conditions for tangential contact are presented here for the discretized contact problem (see Section 2.5.2). Consider a slave node $S^i \in \Gamma_c^s$ with position vector $\boldsymbol{r}^{si}$ that is in contact with the master surface Γ_c^m at the location $\boldsymbol{r}^{m,i}(\bar{\bar{\xi}}, \bar{\bar{\eta}})$. In the case of *sticking* contact, it is required that the relative displacement in the tangential direction between S^i and the minimal distance point is zero, i.e.

$$\boldsymbol{g}_{T,stick}^i(\bar{\bar{\xi}}, \bar{\bar{\eta}}) = \boldsymbol{g}^i(\bar{\bar{\xi}}, \bar{\bar{\eta}}) - g_N^i \bar{\bar{\boldsymbol{n}}}^{m,i} = \left(\boldsymbol{I}_3 - \bar{\bar{\boldsymbol{n}}}^{m,i} \otimes \bar{\bar{\boldsymbol{n}}}^{m,i}\right)\left[\boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}(\bar{\bar{\xi}}, \bar{\bar{\eta}})\right] = \boldsymbol{0} \qquad \text{(F.3)}$$

where $\otimes$ denotes the tensor product. Note that the above equation for the tangential gap function only holds in the case of sticking contact. As the normal contact constraint enforces zero displacement in the normal direction as well (see Eq. 2.5.4b), there is formally no need to distinguish between normal and tangential directions, and the constraint governing sticking contact can simply be written as:

$$\boldsymbol{g}(\bar{\bar{\xi}}, \bar{\bar{\eta}}) = \boldsymbol{r}^{si} - \boldsymbol{r}^{m,i}(\bar{\bar{\xi}}, \bar{\bar{\eta}}) = \boldsymbol{0} \qquad \text{(F.4)}$$

Usually, however, the normal and tangential gap functions are treated separately, for reasons that will be described shortly.

In the case of *sliding* contact, one must obtain an expression for the relative sliding velocity $\dot{\boldsymbol{g}}_T$, see Eq. F.1. Tangential relative slip is characterized by the continuous motion of the minimal distance parameters $\bar{\bar{\boldsymbol{\xi}}} = (\bar{\bar{\xi}}, \bar{\bar{\eta}})$ on the master surface. For a slave node S^i in sliding contact, the relative sliding velocity can thus be written as

$$\dot{\boldsymbol{g}}_T^i = \frac{\partial \boldsymbol{r}^{m,i}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \dot{\bar{\bar{\boldsymbol{\xi}}}} = \frac{\partial \boldsymbol{r}^{m,i}(\boldsymbol{\xi})}{\partial \xi} \dot{\bar{\bar{\xi}}} + \frac{\partial \boldsymbol{r}^{m,i}(\boldsymbol{\xi})}{\partial \eta} \dot{\bar{\bar{\eta}}} \qquad \text{(F.5)}$$

where $\dot{\bar{\bar{\boldsymbol{\xi}}}} = d\bar{\bar{\boldsymbol{\xi}}}/dt$. The tangential gap function for sliding contact is obtained by integrating the above expression from the onset of sliding contact, denote by t_0, to the current time t:

$$\boldsymbol{g}_{T,slip}^i = \int_{t_0}^{t} \frac{\partial \boldsymbol{r}^{m,i}(\boldsymbol{\xi})}{\partial \boldsymbol{\xi}} \dot{\bar{\bar{\boldsymbol{\xi}}}}\, dt \qquad \text{(F.6)}$$

An expression for the time derivative of $\bar{\bar{\boldsymbol{\xi}}}$ can be obtained by deriving Eq. 2.5.7 with respect to time:

$$\frac{d}{dt}\left(\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\cdot\left[\boldsymbol{r}^{si}-\boldsymbol{r}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\right]\right)$$
$$=\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\cdot\left[\dot{\boldsymbol{r}}^{si}-\dot{\boldsymbol{r}}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})-\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\dot{\bar{\bar{\boldsymbol{\xi}}}}\right] \tag{F.7}$$
$$+\left[\dot{\boldsymbol{r}}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})+\boldsymbol{r}_{,\xi\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\dot{\bar{\bar{\boldsymbol{\xi}}}}\right]\cdot\left[\boldsymbol{r}^{si}-\boldsymbol{r}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\right]=\boldsymbol{0}$$

Rearranging terms and noting that $\left(\boldsymbol{r}^{si}-\boldsymbol{r}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\right)=g_N^i\bar{\bar{\boldsymbol{n}}}^{m,i}$, the above equation is written as:

$$\left[\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\cdot\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})-\boldsymbol{r}_{,\xi\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\cdot g_N^i\bar{\bar{\boldsymbol{n}}}^{m,i}\right]\dot{\bar{\bar{\boldsymbol{\xi}}}}$$
$$=\left\{\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\cdot\left(\dot{\boldsymbol{r}}^{si}-\dot{\boldsymbol{r}}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\right)+\dot{\boldsymbol{r}}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})g_N^i\bar{\bar{\boldsymbol{n}}}^{m,i}\right\} \tag{F.8}$$

Finally, note that in the case of exact fulfilment of the impenetrability condition (Eq. 2.5.4b), i.e. $g_N^i=0$, Eq. F.8 simplifies to

$$\left[\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\cdot\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\right]\dot{\bar{\bar{\boldsymbol{\xi}}}}=\left\{\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\cdot\left(\dot{\boldsymbol{r}}^{si}-\dot{\boldsymbol{r}}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\right)\right\} \tag{F.9}$$

Substitution of this equation in Eq. F.5, one obtains an expression for the relative sliding velocity in terms of the velocities of the slave node S^i and the minimal distance point, and the tangent vectors computed at the minimal distance point:

$$\dot{\boldsymbol{g}}_T^i=\frac{\partial\boldsymbol{r}^{m,i}(\boldsymbol{\xi})}{\partial\boldsymbol{\xi}}\left[\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\cdot\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\right]^{-\mathbf{1}}\left\{\boldsymbol{r}_{,\xi}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\cdot\left(\dot{\boldsymbol{r}}^{si}-\dot{\boldsymbol{r}}^{m,i}(\bar{\bar{\boldsymbol{\xi}}})\right)\right\} \tag{F.10}$$

An explicit analytical expression for the above equation in terms of the generalized coordinates of the master and slave bodies can be found in Appendix A.

When solving the equations of motion of a mechanical system involving frictional contact interactions, additional contact conditions should be taken into account. In the case of sticking contact, the constraints $g_N^i=0$ (Eq. 2.5.4b) and $\boldsymbol{g}_{T,stick}^i=\boldsymbol{0}$ (Eq. F.4) should be fulfilled. The latter constraint is removed in the case of sliding contact, and instead tangential forces computed using Eq. F.1 are applied at the sliding contact point. The constraint enforcement methods discussed in Section 2.5.3 can readily be applied to the frictional contact problem. In the Lagrange multiplier method, three Lagrange multipliers are added to the equations of motion for every sticking contact point ($\boldsymbol{\lambda}^i=\{\lambda_N^i\quad\boldsymbol{\lambda}_T^i\}$, where λ_N^i is the Lagrange multiplier corresponding to the impenetrability constraint and $\boldsymbol{\lambda}_T^i$ is the vector of Lagrange multipliers corresponding to the tangent vectors $\boldsymbol{r}_{,\xi}^{m,i}$ and $\boldsymbol{r}_{,\eta}^{m,i}$), whereas in the penalty method the contact forces are computed using:

$$\boldsymbol{f}_c^i=\boldsymbol{f}_{c,N}^i+\boldsymbol{f}_{c,T}^i=\epsilon_N\langle g_N^i\rangle\bar{\bar{\boldsymbol{n}}}^{m,i}+\epsilon_T\boldsymbol{g}_{T,stick}^i \tag{F.11}$$

where ϵ_T is the penalty factor corresponding to the tangential direction. In the case of sliding contact, the above equation is replaced by

$$\boldsymbol{f}_c^i = \epsilon_N \langle g_N^i \rangle \left(\bar{\bar{\boldsymbol{n}}}^{m,i} - \mu_D \frac{\dot{\boldsymbol{g}}_T}{\|\dot{\boldsymbol{g}}_T\|} \right) \tag{F.12}$$

where $\dot{\boldsymbol{g}}_T$ is computed using Eq. F.10. When using the Lagrange multiplier method, $\boldsymbol{\lambda}_T^i$ is removed from the EOMs and the Coulomb friction force (Eq. F.1, where $\dot{\boldsymbol{g}}_T$ is computed using Eq. F.10) is directly applied to the system. The convergence of the Lagrange multiplier method can be improved using Uzawa's algorithm, as was explained in Section 2.5.3.3.

Appendix G

Coefficient tables for numerical integration schemes

G.1 Explicit Runge-Kutta schemes

Classical Runge-Kutta methods The general s-stage ERK integration scheme is given by Eq. 2.6.8 in Section 2.6.1.1. The coefficients in this equation are usually written in a compact form using the so-called Butcher tableau:

$$\begin{array}{c|cccc} c_1 & & & & \\ c_2 & a_{21} & & & \\ c_3 & a_{31} & a_{32} & & \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & a_{s1} & a_{s2} & \cdots & a_{s,s-1} \\ \hline & b_1 & b_2 & \cdots & b_{s-1} \quad b_s \end{array} \tag{G.1}$$

where $c_1 = 0$ for explicit methods. The Butcher tableau's of the ERK1 (also known as the explicit Euler method), ERK2 (also known as the midpoint method), and ERK4 (also known as the *classical* Runge-Kutta method) are given below:

ERK1

$$\begin{array}{c|c} 0 & \\ \hline & 1 \end{array}$$

ERK2

$$\begin{array}{c|cc} 0 & & \\ ½ & ½ & \\ \hline & 0 & 1 \end{array}$$

ERK4

$$\begin{array}{c|ccc} 0 & & & \\ ½ & ½ & & \\ ½ & 0 & ½ & \\ 1 & 0 & 0 & 1 \\ \hline & 1/6 & 2/6 & 2/6 \quad 1/6 \end{array}$$

(G.2)

It can be proven that these methods have orders of convergence of 1, 2, and 4, respectively [122]. The RK4 is the highest-order explicit method for which the number of stages equals the convergence order; for $s > 4$, the order of convergence is always smaller than the number of stages.

Embedded Runge-Kutta methods An embedded ERK method is a pair of Runge-Kutta formulas of different convergence order that have multiple intermediate stages in common and that can be used to estimate the local truncation error of the least accurate formula (see Section 2.6.1.1). In Chapter 7, two embedded ERK methods are applied to the dynamic gear contact problem. These are the Runge-Kutte-Fehlberg 1(2) method [189] and the Bogacki-Shampine 2(3) method [190], which is implemented in Matlab's ODE23 solver [191]. The Butcher tableau's of these methods are

RKF12

$$\begin{array}{c|ccc} 0 & & & \\ \frac{1}{2} & \frac{1}{2} & & \\ 1 & \frac{1}{256} & \frac{255}{256} & \\ \hline & \frac{1}{256} & \frac{255}{256} & 0 \\ \hdashline & \frac{1}{512} & \frac{255}{256} & \frac{1}{512} \end{array}$$

BOSHA23

$$\begin{array}{c|cccc} 0 & & & & \\ \frac{1}{2} & \frac{1}{2} & & & \\ \frac{3}{4} & 0 & \frac{3}{4} & & \\ 1 & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & \\ \hline & \frac{2}{9} & \frac{1}{3} & \frac{4}{9} & 0 \\ \hdashline & \frac{7}{24} & \frac{1}{4} & \frac{1}{3} & \frac{1}{8} \end{array} \qquad \text{(G.3)}$$

where the first row of b coefficients corresponds to the lower-order method and the second row to the higher-order method. Both methods have the FSAL (First Same As Last) property: the last stage is evaluated at the same point as the first stage of the next step. The method of Fehlberg uses two function evaluations per time increment to compute first- and second-order accurate solutions, whereas the method of Bogacki and Shampine uses three function evaluations to compute two- and three-order accurate solutions. Both methods are optimized for local extrapolation: the coefficients of the methods are selected such that the error of the highest-order method is minimized.

G.2 Adams methods

The general form of a k-step Adams method is given by Eq. 2.6.13 in Section 2.6.1.2. The explicit Adams methods (also known as the Adams-Bashforth methods) have $\beta_0 = 0$, whereas implicit methods (also known as Adams-Moulton methods) have $\beta_0 \neq 0$. The β_i coefficients, orders and error constants of explicit and implicit Adams methods are given in Table G.1 and Table G.2, respectively.

p	k	$i \rightarrow$	1	2	3	4	5	6	C_{p+1}
1	1	β_i	1						$\frac{1}{2}$
2	2	$2\beta_i$	3	-1					$\frac{5}{12}$
3	3	$12\beta_i$	23	-16	5				$\frac{3}{8}$
4	4	$24\beta_i$	55	-59	37	-9			$\frac{251}{720}$
5	5	$720\beta_i$	1901	-2774	2616	-1274	251		$\frac{95}{288}$
6	6	$1440\beta_i$	4277	-7923	9982	-7298	2877	-475	$\frac{19087}{60480}$

Table G.1 Coefficients of Adams-Bashforth methods up to order 6

p	k	$i \rightarrow$	0	1	2	3	4	5	C_{p+1}
1	1	β_i	1						$-\frac{1}{2}$
2	1	$2\beta_i$	1	1					$-\frac{1}{12}$
3	2	$12\beta_i$	5	8	-1				$-\frac{1}{24}$
4	3	$24\beta_i$	9	19	-5	1			$-\frac{19}{720}$
5	4	$720\beta_i$	251	646	-264	106	-19		$-\frac{3}{160}$
6	5	$1440\beta_i$	475	1472	-798	482	-173	27	$-\frac{863}{60480}$

Table G.2 Coefficients of Adams-Moulton methods up to order 6

G.3 BDF methods

The general form of a *k*-step BDF scheme is given by Eq. 2.6.29 in Section 2.6.2.1. The coefficients α_i and β_0 of the BDF methods up to order 6 are given in Table G.3.

p	k	β_0	α_1	α_1	α_1	α_1	α_1	α_1	C_{p+1}
1	1	1	-1						$-\frac{1}{2}$
2	2	$\frac{2}{3}$	$-\frac{4}{3}$	$\frac{1}{3}$					$-\frac{2}{9}$
3	3	$\frac{6}{11}$	$-\frac{18}{11}$	$\frac{9}{11}$	$-\frac{2}{11}$				$-\frac{3}{22}$
4	4	$\frac{12}{25}$	$-\frac{48}{25}$	$\frac{36}{25}$	$-\frac{16}{25}$	$\frac{3}{25}$			$-\frac{12}{125}$
5	5	$\frac{60}{137}$	$-\frac{300}{137}$	$\frac{300}{137}$	$-\frac{200}{137}$	$\frac{75}{137}$	$-\frac{12}{137}$		$-\frac{10}{137}$
6	6	$\frac{60}{147}$	$-\frac{360}{147}$	$\frac{450}{147}$	$-\frac{400}{147}$	$\frac{225}{147}$	$-\frac{72}{147}$	$\frac{10}{147}$	$-\frac{20}{343}$

Table G.3 Coefficients of BDF methods up to order 6

Appendix H

Variable-coefficient central difference method

In Section 2.6.1.3, the central difference scheme with lagging velocities was introduced to obtain a fully-explicit close-to-second-order accurate update formula for the generalized displacements (Eq. 2.6.23). In this appendix a variable-coefficient update formula is derived so as to enable smooth time increment changes in the numerical implementation of the central difference method. This formula is derived based on Fig. H.1, where the discretization of the time interval of interest using variable increment sizes is shown. At the considered instant, the solution was propagated from time t_{n-1} to t_n using an increment size of h_{n-1}, and the integrator wants to propagate the current solution with a new increment size of h_n, where $h_n > h_{n-1}$. The acceleration at time t_n can be obtained from the central difference between the velocities at times $t_{n-1/2}$ and $\tilde{t}_{n+1/2}$, where the latter is defined as $\tilde{t}_{n+1/2} = t_n + h_{n-1}/2$, i.e.

$$\ddot{\boldsymbol{q}}_n = \frac{\tilde{\dot{\boldsymbol{q}}}_{n+1/2} - \dot{\boldsymbol{q}}_{n-1/2}}{h_{n-1}} \tag{H.1}$$

where $\tilde{\dot{\boldsymbol{q}}}_{n+1/2}$ is the numerical solution of $\dot{\boldsymbol{q}}$ at time $\tilde{t}_{n+1/2}$. The latter is obtained by interpolating between the numerical solutions of $\dot{\boldsymbol{q}}$ at times $t_{n-1/2}$ and $t_{n+1/2} = t_n + h_n/2$, i.e.

$$\tilde{\dot{\boldsymbol{q}}}_{n+1/2} = \dot{\boldsymbol{q}}_{n+1/2}\left(\frac{2h_{n-1}}{h_{n-1}+h_n}\right) + \dot{\boldsymbol{q}}_{n-1/2}\left(\frac{h_n - h_{n-1}}{h_{n-1}+h_n}\right) \tag{H.2}$$

Substitution of this equation into Eq. H.1 yields the following expression for the acceleration at time t_n:

$$\ddot{\boldsymbol{q}}_n = \frac{2}{h_{n-1}+h_n}\left(\dot{\boldsymbol{q}}_{n+1/2} - \dot{\boldsymbol{q}}_{n-1/2}\right) \tag{H.3}$$

The above half-time velocities can be written in terms of the generalized displacements as follows (see Eq. 2.6.22)

$$\dot{\boldsymbol{q}}_{n+1/2} = \frac{\boldsymbol{q}_{n+1} - \boldsymbol{q}_n}{h_n}, \qquad \dot{\boldsymbol{q}}_{n-1/2} = \frac{\boldsymbol{q}_n - \boldsymbol{q}_{n-1}}{h_{n-1}} \tag{H.4}$$

which upon substitution in Eq. H.3 yields:

$$\ddot{\boldsymbol{q}}_n = \boldsymbol{q}_{n+1}\frac{2}{h_n(h_{n-1}+h_n)} - \boldsymbol{q}_n\frac{2}{h_n h_{n-1}} + \boldsymbol{q}_{n-1}\frac{2}{h_{n-1}(h_{n-1}+h_n)} \tag{H.5}$$

Note that for $h_n = h_{n-1} = h$ this equation reduces to the familiar form of Eq. 2.6.20b. For the case when $h_n < h_{n-1}$, the same formula for $\ddot{\boldsymbol{q}}_n$ is obtained. Substitution of the above equation in the EOMs (Eq. 2.6.19) yields the following update formula

$$\begin{aligned}\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h_n\dot{\boldsymbol{q}}_{n-1/2} \\ + \frac{h_n(h_{n-1}+h_n)}{2}\boldsymbol{M}(\boldsymbol{q}_n)^{-1}\left[\boldsymbol{f}\left(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_{n-1/2}, t_n\right) - \boldsymbol{D}\dot{\boldsymbol{q}}_{n-1/2} - \boldsymbol{K}\boldsymbol{q}_n\right]\end{aligned} \tag{H.6}$$

where $\dot{\boldsymbol{q}}_{n-1/2}$ is defined as in Eq. H.4. Note that when $h_n = h_{n-1} = h$ this equation reduces to the update formula of Eq. 2.6.23.

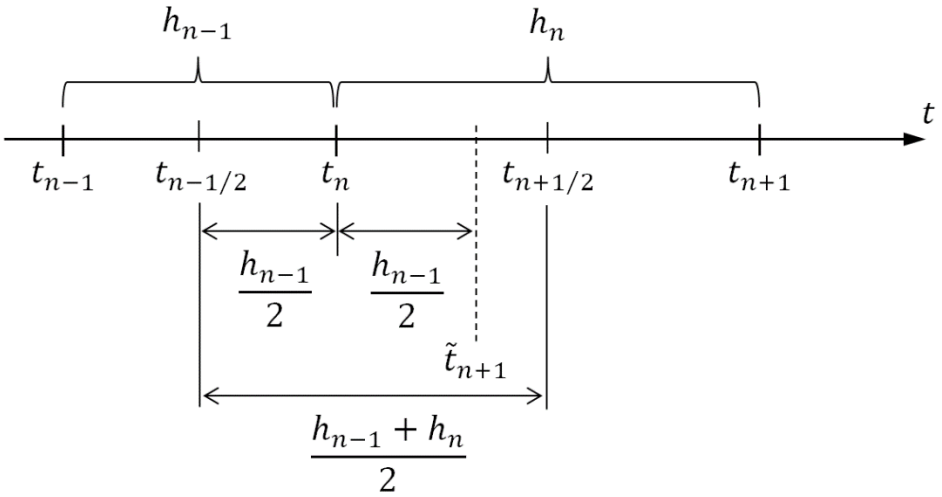


Fig. H.1 Variable increment sized in the central difference method

Appendix I

Direct integration using the BDF2 method

In this appendix, the direct formulation of the second-order BDF method (see Section 2.6.2.1) is derived. This formulation allows to solve the EOMs (Eq. 2.6.2) *directly*, i.e. without conversion to first-order form (Eq. 2.6.5). Applying BDF2 to the trivial equations $\ddot{\boldsymbol{q}} = \ddot{\boldsymbol{q}}$ and $\dot{\boldsymbol{q}} = \dot{\boldsymbol{q}}$, one obtains:

$$\dot{\boldsymbol{q}}_{n+1} = \frac{2}{3}h\ddot{\boldsymbol{q}}_{n+1} + \frac{4}{3}\dot{\boldsymbol{q}}_n - \frac{1}{3}\dot{\boldsymbol{q}}_{n-1} \tag{I.1a}$$

$$\boldsymbol{q}_{n+1} = \frac{2}{3}h\dot{\boldsymbol{q}}_{n+1} + \frac{4}{3}\boldsymbol{q}_n - \frac{1}{3}\boldsymbol{q}_{n-1} \tag{I.1b}$$

Eliminating the velocities from Eq. I.1a, the following set of equations is obtained:

$$\dot{\boldsymbol{q}}_{n+1} = \frac{4}{3}\dot{\boldsymbol{q}}_n - \frac{1}{3}\dot{\boldsymbol{q}}_{n-1} + \frac{2}{3}h\ddot{\boldsymbol{q}}_{n+1} \tag{I.2a}$$

$$\boldsymbol{q}_{n+1} = \frac{4}{3}\boldsymbol{q}_n - \frac{1}{3}\boldsymbol{q}_{n-1} + h\left(\frac{8}{9}\dot{\boldsymbol{q}}_n - \frac{2}{9}\dot{\boldsymbol{q}}_{n-1}\right) + \frac{4}{9}h^2\ddot{\boldsymbol{q}}_{n+1} \tag{I.2b}$$

Similar to the approach outlined in Section 2.6.1.3, the *predictor* velocities and displacements are obtained by setting $\ddot{\boldsymbol{q}}_{n+1} = \boldsymbol{0}$ in the above equations:

$$\dot{\boldsymbol{q}}^*_{n+1} \coloneqq \frac{4}{3}\dot{\boldsymbol{q}}_n - \frac{1}{3}\dot{\boldsymbol{q}}_{n-1} \tag{I.3a}$$

$$\boldsymbol{q}^*_{n+1} \coloneqq \frac{4}{3}\boldsymbol{q}_n - \frac{1}{3}\boldsymbol{q}_{n-1} + h\left(\frac{8}{9}\dot{\boldsymbol{q}}_n - \frac{2}{9}\dot{\boldsymbol{q}}_{n-1}\right) \tag{I.3b}$$

which allows the accelerations and velocities to be written in terms of the displacements corrections $\boldsymbol{q}_{n+1} - \boldsymbol{q}^*_{n+1}$:

$$\ddot{\boldsymbol{q}}_{n+1} = \frac{9}{4h^2}(\boldsymbol{q}_{n+1} - \boldsymbol{q}^*_{n+1}) \tag{I.4a}$$

$$\dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}^*_{n+1} + \frac{3}{2h}(\boldsymbol{q}_{n+1} - \boldsymbol{q}^*_{n+1}) \tag{I.4b}$$

The procedure to solve Eq. 2.6.2 now proceeds along the lines of Algorithm 6.4, with β' and γ' defined by $\beta' = 9/4h^2$ and $\gamma' = 3/2h$. Note that, as compared to the Newmark method (Eq 2.6.34), the direct formulation of the BDF2 method remains a multistep method, and changing the increment size thus involves the same difficulties as outlined in Section 2.6.2.1. The local truncation error $\boldsymbol{q}_{n+1}$ is given by Eq. 2.6.15:

$$\boldsymbol{e}_{n+1} = C_3 h^3 \dddot{\boldsymbol{q}}(t_{n+1}) + \mathcal{O}(h^3) \approx -\frac{2}{9} h^2 (\ddot{\boldsymbol{q}}_{n+1} - \ddot{\boldsymbol{q}}_n) \tag{I.5}$$

where C_3 was specified in Table G.3.

Appendix J

The generalized-α integrator

While the absence of numerical dissipation in the Newmark-β method is desirable in many applications, it often becomes necessary to have some form of algorithmic damping to filter out the response of high-frequency numerical artefacts [4]. Numerical dissipation can be introduced in Newmark's method by selecting $\beta = (1+\alpha^2)/4$ and $\gamma = 1/2 + \alpha$ for some $\alpha > 0$. However, the method then loses its second-order accuracy. Two methods have been proposed to introduce damping in the Newmark method without degrading its order of accuracy. The first approach consists of keeping the Newmark formulas (Eq. 2.6.34) but interpolating the generalized force vector $\boldsymbol{f}(\boldsymbol{q}, \boldsymbol{q}, t)$ (see Eq. 2.6.2) between the current and future times, i.e.

$$\boldsymbol{M}(\boldsymbol{q}_{n+1})\ddot{\boldsymbol{q}}_{n+1} = (1-\alpha)\boldsymbol{f}(\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1}, t_{n+1}) + \alpha \boldsymbol{f}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n) \tag{J.1}$$

This approach is referred to as the α-method or the HHT-method, after its inventors Hilber, Hughes, and Taylor [135]. Later on this approach was generalized by Chung and Hilbert [192] by introducing a similar interpolation of the inertia forces in Eq. J.1:

$$(1-\alpha_m)\boldsymbol{M}(\boldsymbol{q}_{n+1})\ddot{\boldsymbol{q}}_{n+1} + \alpha_m \boldsymbol{M}(\boldsymbol{q}_n)\ddot{\boldsymbol{q}}_n = \big(1-\alpha_f\big)\boldsymbol{f}(\boldsymbol{q}_{n+1}, \dot{\boldsymbol{q}}_{n+1}, t_{n+1}) + \alpha_f \boldsymbol{f}(\boldsymbol{q}_n, \dot{\boldsymbol{q}}_n, t_n) \tag{J.2}$$

where α_m and α_f are the weighting parameters affecting, respectively, the inertial forces and the generalized forces vector $\boldsymbol{f}$. The residual equation of this generalized-α formula can be expressed in terms of $\boldsymbol{q}_{n+1}$ using a similar approach as outlined in Eq. 2.6.35-2.6.37. However, a more appropriate approach for defining the generalized-α scheme was proposed by Arnold and Brüls [136] and is adopted in this dissertation. While conceptually their approach is the same as weighting the dynamic equilibrium equations (Eq. J.1) (at least in the linear case), it maintains consistency in the case of non-constant mass matrices and it computes accelerations with second-order (instead of first-order) accuracy. The key to their approach is the introduction of a vector of pseudo-accelerations $\boldsymbol{a}$ that replaces the real accelerations $\ddot{\boldsymbol{q}}$ in the Newmark formulas (Eq. 2.6.34), i.e.

$$\dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}_n + h[(1-\gamma)\boldsymbol{a}_n + \gamma \boldsymbol{a}_{n+1}] \tag{J.3a}$$

$$\boldsymbol{q}_{n+1} = \boldsymbol{q}_n + h\dot{\boldsymbol{q}}_n + \frac{h^2}{2}[(1-2\beta)\boldsymbol{a}_n + 2\beta \boldsymbol{a}_{n+1}] \tag{J.3b}$$

The pseudo-accelerations are defined by the following recurrence relation:

$$(1-\alpha_m)\boldsymbol{a}_{n+1} + \alpha_m \boldsymbol{a}_n = (1-\alpha_f)\ddot{\boldsymbol{q}}_{n+1} + \alpha_f \ddot{\boldsymbol{q}}_n \ , \quad \boldsymbol{a}_0 = \ddot{\boldsymbol{q}}_0 \tag{J.4}$$

where the accuracy and stability conditions of the method's coefficients are given by [73]:

$$\gamma = 1/2 - \alpha_m + \alpha_f \quad \text{(second order)} \qquad -1 \leq \alpha_m \leq 1/2 \quad \text{(0-stability)} \tag{J.5}$$

At time t_n, both $\ddot{\boldsymbol{q}}_n$ and $\boldsymbol{a}_n$ are known, hence $\boldsymbol{a}_{n+1}$ can be expressed in terms of $\ddot{\boldsymbol{q}}_{n+1}$ using Eq. J.3. The *predictor* velocities and displacements are then obtained by setting $\ddot{\boldsymbol{q}}_{n+1} = \boldsymbol{0}$, yielding:

$$\dot{\boldsymbol{q}}^*_{n+1} \coloneqq \dot{\boldsymbol{q}}_n + \frac{h}{1-\alpha_m}\left[\gamma\alpha_f \ddot{\boldsymbol{q}}_n + (1-\gamma-\alpha_m)\boldsymbol{a}_n\right] \tag{J.6a}$$

$$\boldsymbol{q}^*_{n+1} \coloneqq \boldsymbol{q}_n + h\dot{\boldsymbol{q}}_n + \frac{h^2}{2(1-\alpha_m)}\left[2\alpha_f \ddot{\boldsymbol{q}}_n + (1-\beta-\alpha_m)\boldsymbol{a}_n\right] \tag{J.6b}$$

Proceeding as in Eq. 2.6.36, the above equations are substituted in Eq. J.3 to obtain expressions for the accelerations and velocities in terms of the predictor values and $\boldsymbol{q}_{n+1}$:

$$\ddot{\boldsymbol{q}}_{n+1} = \frac{(1-\alpha_m)}{\beta h^2(1-\alpha_f)}(\boldsymbol{q}_{n+1} - \boldsymbol{q}^*_{n+1}) = \beta'(\boldsymbol{q}_{n+1} - \boldsymbol{q}^*_{n+1}) \tag{J.7a}$$

$$\dot{\boldsymbol{q}}_{n+1} = \dot{\boldsymbol{q}}^*_{n+1} + \frac{\gamma}{\beta h}(\boldsymbol{q}_{n+1} - \boldsymbol{q}^*_{n+1}) = \dot{\boldsymbol{q}}^*_{n+1} + \gamma'(\boldsymbol{q}_{n+1} - \boldsymbol{q}^*_{n+1}) \tag{J.7b}$$

The generalized-α scheme now proceeds along the lines of Algorithm 2.4, where the predictors are defined by Eq. J.6, and where the coefficients β' and γ' in Eq. 2.6.42 and Eq. 2.6.43 are defined by Eq. J.7.

Of particular interest is the behavior of the generalized-α scheme for time increments $h \gg T_{min}$, where as before T_{min} denotes the period of the highest frequency in the system. Through proper selection of the parameters α_m, α_f, and β, the numerical dissipation of high-frequency oscillations can be algorithmically controlled in the generalized-α scheme. Numerical dissipation is usually expressed in terms of the spectral radius at infinity, $\rho_\infty \in [0,1]$, where $\rho_\infty = 0$ represents asymptotic annihilation of the high frequency response (the method then becomes L-stable, meaning that the solution approaches zero in a single time step as the increment size goes to infinity), and $\rho_\infty = 1$ corresponds to no numerical dissipation. Unconditional stability and second-order accuracy of the generalized-α scheme (for the linear case) is achieved for the following parameters [136]:

$$\alpha_f = \frac{\rho_\infty}{1+\rho_\infty} \ , \alpha_m = \frac{2\rho_\infty - 1}{1+\rho_\infty} \ , \beta = \frac{1}{4}\left(1-\alpha_m+\alpha_f\right)^2 \ , \gamma = 1/2 - \alpha_m + \alpha_f \tag{J.8}$$

Note that these conditions guarantee unconditional stability in the linear case only. The stability of the generalized-α method in nonlinear dynamic analysis is investigated in [193].

Appendix K

Analytical geometry of external gears

The equations describing the geometry of external cylindrical gears are obtained through a kinematic description of the manufacturing process. Involute gears are typically manufactured using a rack cutter, as shown in Fig. K.1. During tooth cutting, the rack reciprocates in a direction parallel to the gear axis while feeding into the gear blank with a linear velocity v. At the same time, the gear blank performs a synchronized rotation about the gear axis with angular velocity ω, such that $v = \omega\rho = \omega d/2$. The rack cutter can be considered as a cylindrical gear with an infinite number of teeth (and hence an infinite pitch diameter) that meshes with the gear blank. In the nominal configuration, the pitch diameters of the rack cutter and gear blank coincide at the instantaneous center of rotation, *I*. Often the rack cutter is displaced from the pitch circle over an amount $e = xm_n$, where x is the so-called profile shift coefficient.

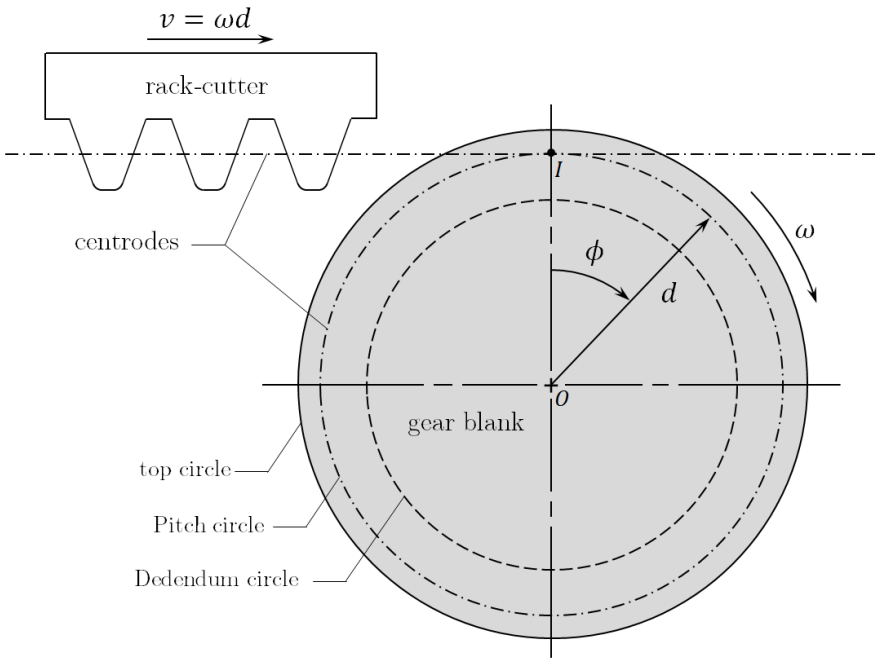


Fig. K.1 Manufacturing of an external involute gear using a rack cutter

A schematic representation of the manufacturing process of an external gear with profile shift is shown in Fig. K.2. The coordinate systems $\boldsymbol{x^1y^1}$ and $\boldsymbol{x^2y^2}$ are rigidly connected to, respectively,

the translating rack cutter and the rotating gear blank. An arbitrary point *P* is represented in $\boldsymbol{x^2y^2}$ by the homogeneous position vector

$$\boldsymbol{r}_P^2 = \overline{O_2P} = \{x_P^2 \quad y_P^2 \quad 1\}^T \tag{K.1}$$

where the superscript '*2*' refers to the coordinate system that is rigidly attached to the gear blank. The same point *P* can be expressed in the coordinate system $\boldsymbol{x^1y^1}$ using the following homogeneous transformation (see Fig. K.2):

$$\boldsymbol{r}_P^1 = \boldsymbol{M}_{12}(\phi)\boldsymbol{r}_P^2 = \begin{bmatrix} cos\,\phi & sin\,\phi & -\rho\phi \\ -sin\,\phi & cos\,\phi & -(\rho+e) \\ 0 & 0 & 1 \end{bmatrix}\boldsymbol{r}_P^2 \tag{K.2}$$

where $\boldsymbol{\phi}$ is the rotation angle of the gear blank, corresponding to a horizontal feed movement of the cutter by an amount of $\boldsymbol{s = \rho\phi}$. Inverting the above equation, one obtains the homogeneous transformation from a vector expressed in $\boldsymbol{x^1y^1}$ to the same vector expressed in $\boldsymbol{x^2y^2}$:

$$\boldsymbol{r}_P^2 = \boldsymbol{M}_{12}^{-1}\boldsymbol{r}_P^1 = \boldsymbol{M}_{21}\boldsymbol{r}_P^1 = \begin{bmatrix} cos\,\phi & -sin\,\phi & \rho(\phi\,cos\,\phi - sin\,\phi) - e\,sin\,\phi \\ sin\,\phi & cos\,\phi & \rho(\phi\,sin\,\phi + cos\,\phi) + e\,cos\,\phi \\ 0 & 0 & 1 \end{bmatrix}\boldsymbol{r}_P^1 \tag{K.3}$$

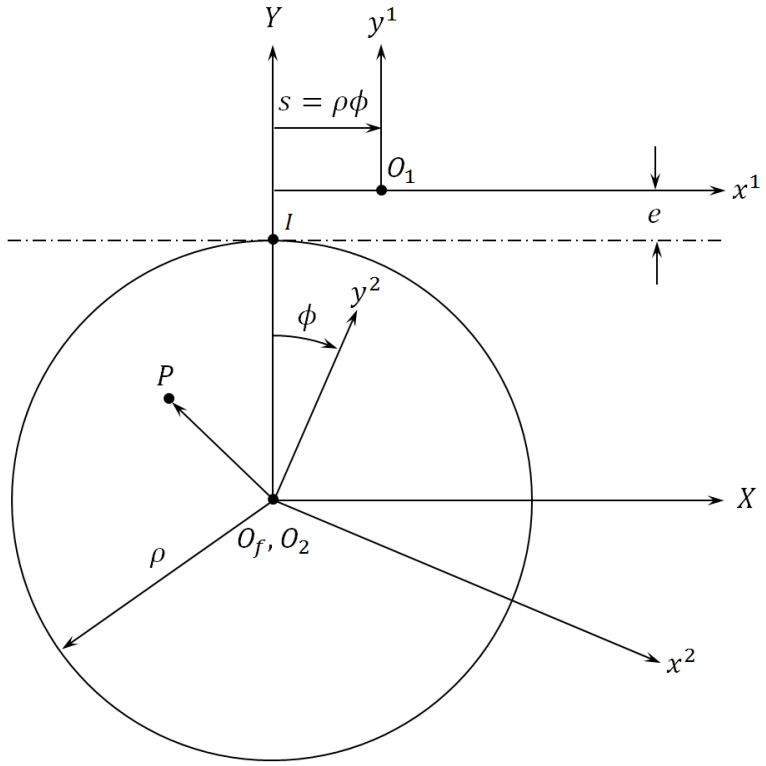


Fig. K.2 Schematic representation of the manufacturing process of an external gear

The equations describing the geometry of an external gear are now obtained by expressing the simple geometry of the rack cutter in $\boldsymbol{x^1y^1}$ and transforming the rack geometry to $\boldsymbol{x^2y^2}$, taking into account the relative motion between cutter and blank. The geometry of a rack cutter is defined in a normal section (with coordinate system $\boldsymbol{x^{1n}y^{1n}}$) and is standardized in the ISO 53 standard [194]. As shown in Fig. K.3, one edge of the rack cutter consists of four curves, each parameterized by a parameter θ: the top land of the rack (corresponding to the bottom land of the generated gear; ①); the rounded corner of the rack (which will generate the root fillet of the gear; ②); the straight flank (which will generate the involute flank of the gear; ③); and finally the bottom land of the rack (corresponding to the top land of the gear; ④).

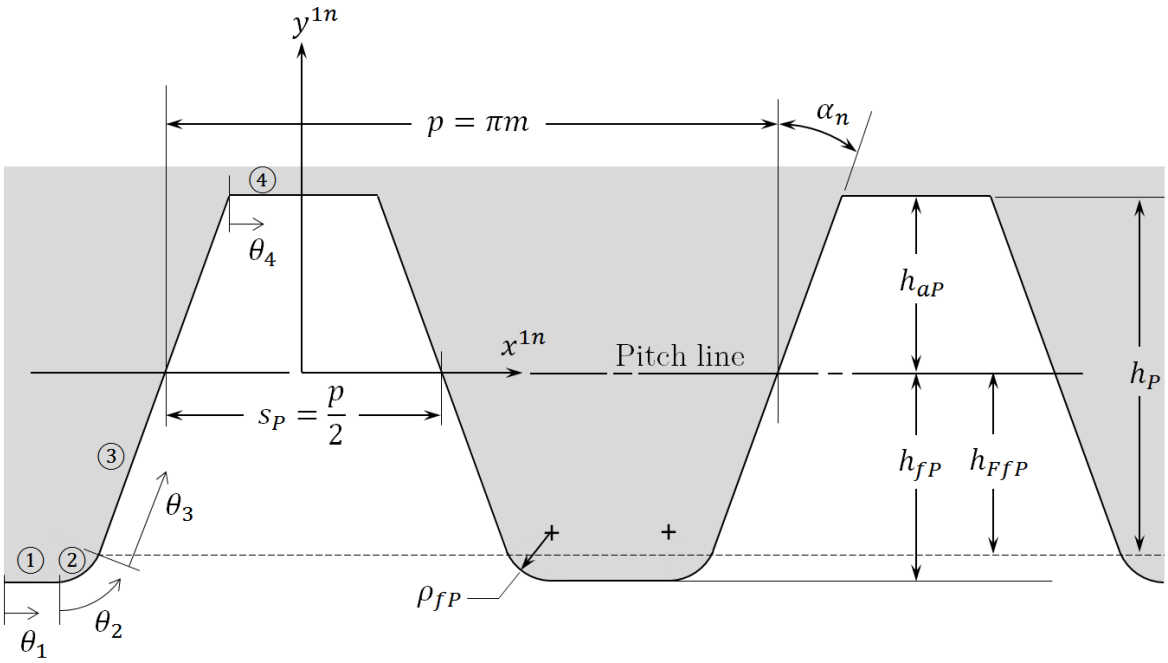


Fig. K.3 Geometry of the basic rack profile in a normal section

The three-dimensional geometry generation of cylindrical gears is reduced to a two-dimensional problem by generating the geometry of the gear in a transverse section and extruding the latter in a helicoid about the gear axis. To this end, the geometry of the rack cutter should be defined in a transverse section (with coordinate system $\boldsymbol{x^{1t}y^{1t}}$), which involves the following Cartesian transformation:

$$\boldsymbol{r^{1t}} = \boldsymbol{M_{tn}r^{1n}} = \begin{bmatrix} (\cos\beta)^{-1} & 0 \\ 0 & 1 \end{bmatrix} \boldsymbol{r^{1n}} \tag{K.4}$$

where β is the helix angle of the rack (and of the generated gear). The curve equations of the rack cutter in the normal and transverse directions are given in Table K.1 - Table K.4. Once a complete description of the rack geometry in $\boldsymbol{x^{1t}y^{1t}}$ is obtained, Eq. K.3 is used to transform the rack

geometry to the transverse gear blank axis system $\boldsymbol{x^{2t}y^{2t}}$. Fig. K.4 shows a family of rack cutter profiles obtained by transforming the geometry of Fig. K.3 to $\boldsymbol{x^{2t}y^{2t}}$ for $n = 40$ equidistant values of $\phi = \phi^i$ (see Fig. K.3), where the latter is obtained from:

$$\phi^i = \frac{-2\pi}{z} + (i-1)\frac{4\pi}{nz}, \quad i = 1 \dots n \tag{K.5}$$

Note that the rack cutter enters the gear blank from the left upper corner of Fig. K.4 and leaves via the right upper corner.

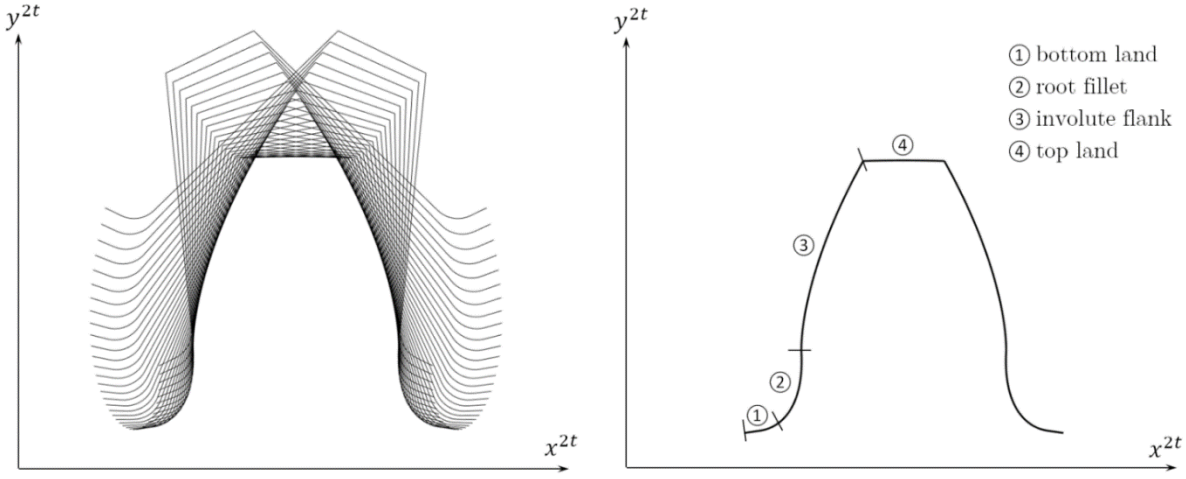


Fig. K.4 Family of rack cutter profiles represented in the gear blank axis system (left) and the corresponding envelope curves (right)

Eq. K.3 can be used to numerically simulate the cutting process and thereby obtain a family of rack cutter profiles that demarcate the part of the gear blank that is acquired after cutting. This family of rack cutter profiles is represented by $\boldsymbol{r^{2t}(\phi, \theta)}$, where ϕ denotes the gear blank rotation and θ is the parameter of the selected rack curve (see Table K.1 - Table K.4). To obtain a mathematical description of the envelope of these profiles (and hence of the geometry of the generated gear), a one-to-one relation must be established between ϕ and θ. This relation is provided by the equation of meshing, which states that the common normal to the surfaces of the rack cutter and the generated gear at their point of tangency must pass through the instantaneous axis of rotation [156]. In the case of planar gearing, this relation is expressed as:

$$g(\phi, \theta) = [x_I^1 - x^1(\theta)]n_y^1(\theta) - [y_I^1 - y^1(\theta)]n_x^1(\theta) = 0 \tag{K.6}$$

where $x_I^1 = -\rho\phi$ and $y_I^1 = -e$ are the coordinates of the instantaneous center of rotation (see Fig. K.2), and n_x^1 and n_y^1 denote the components of the normal vector $\boldsymbol{n^1} = \partial \boldsymbol{r^1}/\partial\theta \times \boldsymbol{e_z^1}$, where $\boldsymbol{e_z^1}$ is the unit vector along the z^1 axis. Eq. K.6 yields a one-to-one relation between ϕ and θ that upon

insertion in Eq. K.3 and Eq. K.4 yields the closed-form equations describing the geometry of involute external gears. The procedure outlined above is summarized for the four rack cutter curves in Table K.1 - Table K.4 (for a list of symbols, see Table K.5).

Begin points x_0^1 and y_0^1	$x_{1,0}^{1n} = -\frac{\pi m_n}{2}\,, \quad x_{1,0}^{1t} = -\frac{\pi m_t}{2}$	$y_{1,0}^{1n} = y_{1,0}^{1t} = -h_{fP}$
Curve equation in $\boldsymbol{x^{1n}y^{1n}}$	$\boldsymbol{r}_1^{1n} = \begin{Bmatrix} x_{1,0}^{1n} + \theta_1^n \\ y_{1,0}^{1n} \end{Bmatrix}$	$0 \le \theta_1^n \le \frac{\pi m_n}{4} - h_{FfP}\tan\alpha_n - \rho_{fP}\cos\alpha_n$
Curve equation in $\boldsymbol{x^{1t}y^{1t}}$	$\boldsymbol{r}_1^{1t} = \begin{Bmatrix} x_{1,0}^{1t} + \theta_1^t \\ y_{1,0}^{1t} \end{Bmatrix}$	$0 \le \theta_1^t \le \frac{\pi m_t}{4} - h_{FfP}\tan\alpha_t - \rho_{fP}\cos\alpha_t$
Curve equation in $\boldsymbol{x^{2t}y^{2t}}$	$\boldsymbol{r}_1^{2t} = \begin{Bmatrix} \left(x_{1,0}^{1t} + \theta_1^t + \rho\phi\right)\cos\phi - \left(y_{1,0}^{1t} + e + \rho\right)\sin\phi \\ \left(x_{1,0}^{1t} + \theta_1^t + \rho\phi\right)\sin\phi + \left(y_{1,0}^{1t} + e + \rho\right)\cos\phi \end{Bmatrix}$	
Equation of meshing	$\rho\phi + x_{1,0}^{1t} + \theta_1^t = 0$	
Equation of bottom land (left side of the tooth)	$\boldsymbol{r}_1^{2t} = \begin{Bmatrix} -\left(y_{1,0}^{1t} + e + \rho\right)\sin\phi \\ \left(y_{1,0}^{1t} + e + \rho\right)\cos\phi \end{Bmatrix} = \begin{Bmatrix} -\rho_f\sin\phi \\ \rho_f\cos\phi \end{Bmatrix}$	

Table K.1 Equations of curve 1, i.e. the rack's top land that generates the gear's bottom land

Begin points: x_0^{1n} and y_0^{1n}	$x_{2,0}^{1n} = -\frac{\pi m_n}{4} - h_{FfP}\tan\alpha_n - \rho_{fP}\cos\alpha_n$	$y_{2,0}^{1n} = -h_{fP}$
Begin points: x_0^{1t} and y_0^{1t}	$x_{2,0}^{1t} = -\frac{\pi m_t}{4} - h_{FfP}\tan\alpha_t - \rho_{fP}\cos\alpha_t$	$y_{2,0}^{1t} = y_{2,0}^{1n} = -h_{fP}$
Curve equation in $\boldsymbol{x^{1n}y^{1n}}$	$\boldsymbol{r}_2^{1n} = \begin{Bmatrix} x_{2,0}^{1n} + \rho_{fP}\sin\theta_2^n \\ y_{2,0}^{1n} + \rho_{fP}(1-\cos\theta_2^n) \end{Bmatrix}$	$0 \le \theta_2^n \le \frac{\pi}{2} - \alpha_n$
Curve equation in $\boldsymbol{x^{1t}y^{1t}}$	$\boldsymbol{r}_2^{1t} = \begin{Bmatrix} x_{2,0}^{1t} + \rho_{fP}\sin\theta_2^n/\cos\beta \\ y_{2,0}^{1t} + \rho_{fP}(1-\cos\theta_2^n) \end{Bmatrix}$	
Curve equation in $\boldsymbol{x^{2t}y^{2t}}$	$\boldsymbol{r}_2^{2t} = \begin{Bmatrix} \left(x_{2,0}^{1t} + \rho_{fP}\sin\theta_2^n/\cos\beta + \rho\phi\right)\cos\phi - \left(y_{2,0}^{1t} + \rho_{fP}(1-\cos\theta_2^n) + e + \rho\right)\sin\phi \\ \left(x_{2,0}^{1t} + \rho_{fP}\sin\theta_2^n/\cos\beta + \rho\phi\right)\sin\phi + \left(y_{2,0}^{1t} + \rho_{fP}(1-\cos\theta_2^n) + e + \rho\right)\cos\phi \end{Bmatrix}$	
Equation of meshing	$\left(\rho\phi + x_{2,0}^{1t} + \rho_{fP}\frac{\sin\theta_2^n}{\cos\beta}\right)\frac{\cos\theta_2^n}{\cos\beta} + \left(e + y0 + \rho_{fP}(1-\cos\theta_2^n)\right)\sin\theta_2^n = 0$	
Equation of root fillet	*Combine the above two equations*	

Table K.2 Equations of curve 2, i.e. the rack's rounded corner that generates the gear's root fillet

Begin points: x_0^{1n} and y_0^{1n}	$x_{3,0}^{1n} = -\frac{\pi m_n}{4} - h_{FfP}\tan\alpha_n$	$y_{3,0}^{1n} = -h_{FfP}$
Begin points: x_0^{1t} and y_0^{1t}	$x_{3,0}^{1t} = -\frac{\pi m_t}{4} - h_{FfP}\tan\alpha_t$	$y_{3,0}^{1t} = y_{3,0}^{1n} = -h_{FfP}$
Curve equation in $\boldsymbol{x^{1n}y^{1n}}$	$\boldsymbol{r}_3^{1n} = \begin{Bmatrix} x_{3,0}^{1n} + \theta_3^n \sin\alpha_n \\ y_{3,0}^{1n} + \theta_3^n \cos\alpha_n \end{Bmatrix}$	$0 \le \theta_3^n \le \frac{h_P}{\cos\alpha_n}$
Curve equation in $\boldsymbol{x^{1t}y^{1t}}$	$\boldsymbol{r}_3^{1t} = \begin{Bmatrix} x_{3,0}^{1t} + \theta_3^n \sin\alpha_n/\cos\beta \\ y_{3,0}^{1n} + \theta_3^n \cos\alpha_n \end{Bmatrix} = \begin{Bmatrix} x_{3,0}^{1t} + \theta_3^t \sin\alpha_t \\ y_{3,0}^{1t} + \theta_3^t \cos\alpha_t \end{Bmatrix}$	$0 \le \theta_3^t \le \frac{h_P}{\cos\alpha_t}$
Curve equation in $\boldsymbol{x^{2t}y^{2t}}$	$\boldsymbol{r}_3^{2t} = \begin{Bmatrix} \theta_3^t \sin(\alpha_t - \phi) + \left(x_{3,0}^{1t} + \rho\phi\right)\cos\phi - \left(y_{3,0}^{1t} + e + \rho\right)\sin\phi \\ \theta_3^t \cos(\alpha_t - \phi) + \left(x_{3,0}^{1t} + \rho\phi\right)\sin\phi + \left(y_{3,0}^{1t} + e + \rho\right)\cos\phi \end{Bmatrix}$	
Equation of meshing	$\left(\rho\phi + x_{3,0}^{1t}\right)\sin\alpha_t + \theta_3^t + \left(y_{3,0}^{1t} + e\right)\cos\alpha_t = 0$	
Equation of involute (left side of the tooth) $\boldsymbol{r}_3^{2t} = \begin{Bmatrix} -\left[\left(\rho\phi + x_{3,0}^{1t}\right)\sin\alpha_t + \left(y_{3,0}^{1t} + e\right)\cos\alpha_t\right]\sin(\alpha_t - \phi) + \left(x_{3,0}^{1t} + \rho\phi\right)\cos\phi - \left(y_{3,0}^{1t} + e + \rho\right)\sin\phi \\ -\left[\left(\rho\phi + x_{3,0}^{1t}\right)\sin\alpha_t + \left(y_{3,0}^{1t} + e\right)\cos\alpha_t\right]\cos(\alpha_t - \phi) + \left(x_{3,0}^{1t} + \rho\phi\right)\sin\phi + \left(y_{3,0}^{1t} + e + \rho\right)\cos\phi \end{Bmatrix}$		
Transverse normals to involute (left side of the tooth) $\boldsymbol{n}_3^{2t} = \begin{Bmatrix} \sin(\alpha_t - \phi) + (\rho\sin\alpha_t)^{-1}\left[-\theta_3^t\cos(\alpha_t - \phi) + \left(x_{3,0}^{1t} + \rho\phi\right)\sin\phi + \left(y_{3,0}^{1t} + e\right)\cos\phi\right] \\ \cos(\alpha_t - \phi) + (\rho\sin\alpha_t)^{-1}\left[\theta_3^t\sin(\alpha_t - \phi) + \left(x_{3,0}^{1t} + \rho\phi\right)\cos\phi + \left(y_{3,0}^{1t} + e\right)\sin\phi\right] \end{Bmatrix}$		

Table K.3 Equations of curve 3, i.e. the rack's straight flank that generates the gear's involute flank

Begin points: x_0^{1n} and y_0^{1n}	$x_{4,0}^{1n} = -\frac{\pi m_n}{4} + h_{aP}\tan\alpha_n$	$y_{4,0}^{1n} = h_{aP}$
Begin points: x_0^{1t} and y_0^{1t}	$x_{4,0}^{1t} = -\frac{\pi m_t}{4} + h_{aP}\tan\alpha_t$	$y_{4,0}^{1t} = y_{4,0}^{1n} = h_{aP}$
Curve equation in $\boldsymbol{x^{1n}y^{1n}}$	$\boldsymbol{r}_4^{1n} = \begin{Bmatrix} x_{4,0}^{1n} + \theta_4^n \\ y_{4,0}^{1n} \end{Bmatrix}$	$0 \le \theta_4^n \le \frac{\pi m_n}{4} - h_{aP}\tan\alpha_n$
Curve equation in $\boldsymbol{x^{1t}y^{1t}}$	$\boldsymbol{r}_1^{1t} = \begin{Bmatrix} x_{4,0}^{1t} + \theta_4^t \\ y_{4,0}^{1t} \end{Bmatrix}$	$0 \le \theta_4^t \le \frac{\pi m_t}{4} - h_{aP}\tan\alpha_t$
Curve equation in $\boldsymbol{x^{2t}y^{2t}}$	$\boldsymbol{r}_1^{2t} = \begin{Bmatrix} \left(x_{4,0}^{1t} + \theta_4^t + \rho\phi\right)\cos\phi - \left(y_{4,0}^{1t} + e + \rho\right)\sin\phi \\ \left(x_{4,0}^{1t} + \theta_4^t + \rho\phi\right)\sin\phi + \left(y_{4,0}^{1t} + e + \rho\right)\cos\phi \end{Bmatrix}$	
Equation of meshing	$\rho\phi + x_{4,0}^{1t} + \theta_4^t = 0$	
Equation of top land (left side of the tooth)	$\boldsymbol{r}_1^{2t} = \begin{Bmatrix} -\left(y_{4,0}^{1t} + e + \rho\right)\sin\phi \\ \left(y_{4,0}^{1t} + e + \rho\right)\cos\phi \end{Bmatrix} = \begin{Bmatrix} -\rho_a\sin\phi \\ \rho_a\cos\phi \end{Bmatrix}$	

Table K.4 Equations of curve 4, i.e. the rack's bottom land that generates the gear's top land

Symbol	Meaning	Equation
ρ	Pitch radius	$\rho = zm_t/2$
ρ_a	Top radius	$\rho_a = \rho + xm_n + h_{aP}$
ρ_f	Root radius	$\rho_f = \rho + xm_n - h_{fP}$
α_t	Pressure angle in the transverse section	$\alpha_t = \tan^{-1}\left(\frac{\tan\alpha_n}{\cos\beta}\right)$

Table K.5 Symbols used in Table K.1-Table K.4

Appendix L

Analytical geometry of internal gears

Unlike external gears, internal gears cannot be manufactured using rack cutters. Instead, internal gears are often cut using the method of *shaping* by a pinion cutter, as show in Fig. L.1. In this process, the pinion cutter reciprocates in the direction of the gear axis while the gear blank and cutter perform synchronous rotations that are related by the following equation:

$$\frac{\phi_1}{\phi_2} = \frac{z_2}{z_1} = const. \tag{L.1}$$

where the subscripts '*1*' and '*2*' denote the cutter and the gear blank, respectively. The center distance E_c is held constant and is given by

$$E_c = \rho_2 - \rho_1 - e = \frac{1}{2}(z_2 - z_1)m_t - e \tag{L.2}$$

where ρ_2 and ρ_1 are the pitch radii of the gear and cutter, respectively.

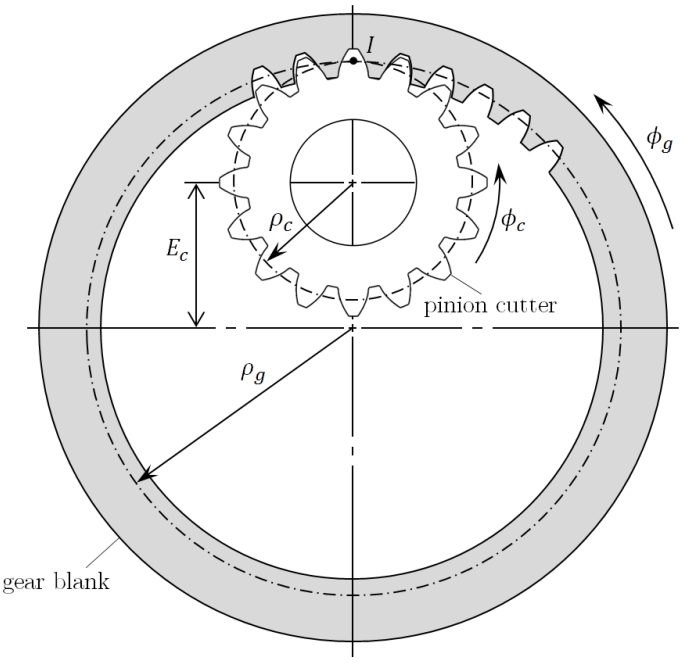


Fig. L.1 Manufacturing of an internal involute gear using a pinion cutter

A schematic representation of the manufacturing process of an internal gear using a shifted pinion cutter is shown in Fig. L.2. Analogous to Fig. K.2, the coordinate systems $\boldsymbol{x^1y^1}$ and $\boldsymbol{x^2y^2}$ are rigidly connected to, respectively, the pinion cutter and the gear blank. An arbitrary point *P* that is represented in $\boldsymbol{x^1y^1}$ by the position vector $\boldsymbol{r_P^1}$ can be expressed in $\boldsymbol{x^2y^2}$ using the following homogeneous transformation (see Eq. K.3):

$$\boldsymbol{r_P^2} = \boldsymbol{M_{21}r_P^1} = \begin{bmatrix} cos(\phi_1 - \phi_2) & -sin(\phi_1 - \phi_2) & E_c\, sin\,\phi_2 \\ sin(\phi_1 - \phi_2) & cos(\phi_1 - \phi_2) & E_c\, cos\,\phi_2 \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{r_P^1} \tag{L.3}$$

By expressing the shape of the pinion cutter in $\boldsymbol{x^1y^1}$, the above equation yields a family of pinion cutter profiles in $\boldsymbol{x^2y^2}$. The internal gear tooth geometry is then obtained by applying the equation of meshing (Eq. K.6), with $x_I^1 = \rho_1 \sin\phi_1$ and $y_I^1 = \rho_1 \cos\phi_1$. The pinion cutter profile is composed of three curves, as shown in Fig. L.3: the involute flank of the pinion (generating the involute flank of the internal gear; ①); the rounded corner of the cutter (which will generate the root fillet of the gear; ②); and the rounded top land of the cutter (which will generate the bottom land of the internal gear; ③). Using a similar procedure as outlined in the previous section, the closed-form equations of the internal gear are obtained and summarized in Table L.1 - Table L.4.

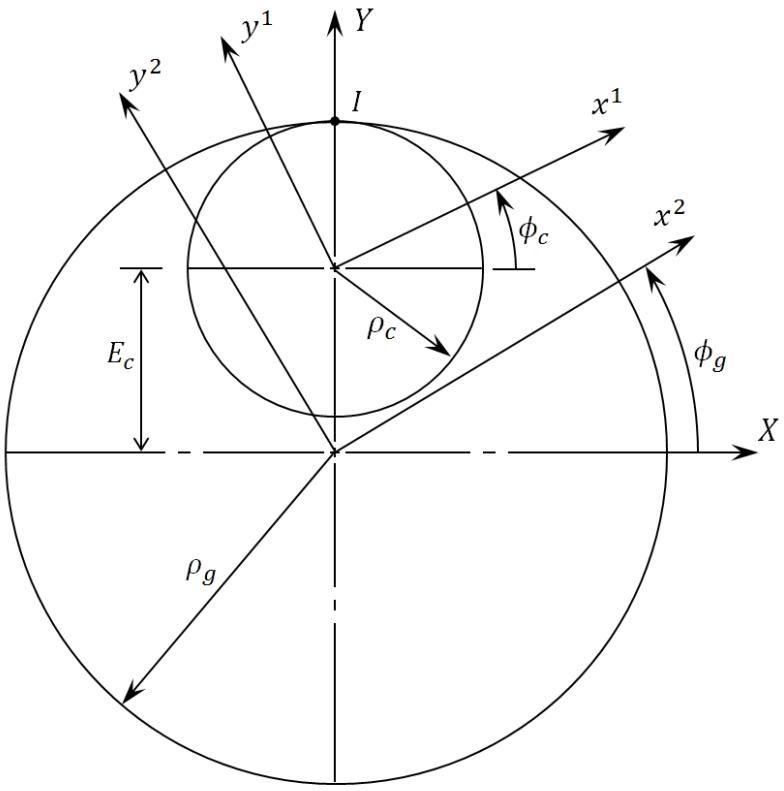


Fig. L.2 Schematic representation of the manufacturing process of an internal gear

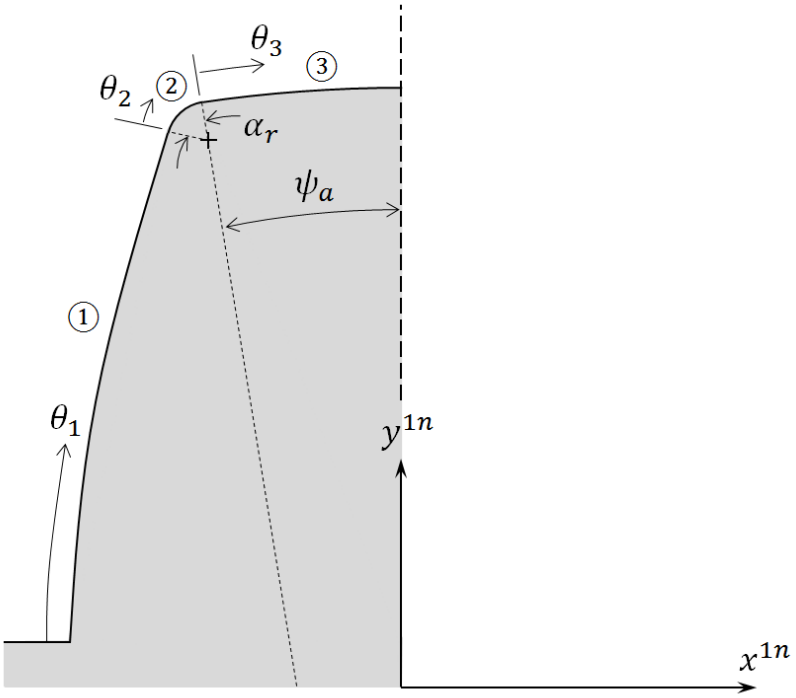


Fig. L.3 Geometry of the pinion cutter profile in a normal section

Begin points: x_0^{1t} and y_0^{1t}	$x_{1,0}^{1n} = -\rho_{b1}\sin\psi_{b1}$	$y_{1,0}^{1n} = \rho_{b1}\cos\psi_{b1}$
Curve equation in $\boldsymbol{x^{1t}y^{1t}}$	$\boldsymbol{r}_1^{1t} = \rho_{b1}\begin{Bmatrix}(\sin\theta_1 - \theta_1\cos\theta_1)\cos\psi_{b1} - (\cos\theta_1 + \theta_1\sin\theta_1)\sin\psi_{b1}\\(\sin\theta_1 - \theta_1\cos\theta_1)\sin\psi_{b1} + (\cos\theta_1 + \theta_1\sin\theta_1)\cos\psi_{b1}\end{Bmatrix}$	$0 \le \theta_1 \le \theta_{1,max}$
Curve equation in $\boldsymbol{x^{2t}y^{2t}}$	$\boldsymbol{r}_1^{2t} = \begin{Bmatrix}\rho_{b1}(\sin\theta_1 - \theta_1\cos\theta_1)\cos\phi_{12b} - \rho_{b1}(\cos\theta_1 + \theta_1\sin\theta_1)\sin\phi_{12b} + E_c\sin\phi_2\\\rho_{b1}(\sin\theta_1 - \theta_1\cos\theta_1)\sin\phi_{12b} + \rho_{b1}(\cos\theta_1 + \theta_1\sin\theta_1)\cos\phi_{12b} + E_c\cos\phi_2\end{Bmatrix}$	
Equation of meshing	$\rho_1\cos(\theta_1 - \phi_1 - \psi_{b1}) - \rho_{b1} = 0 \rightarrow \phi_1 = \theta_1 - \cos^{-1}\left(\frac{\rho_{b1}}{\rho_1}\right) - \psi_{b1}$	
Equation of involute	*Combine the above two equations*	
Transverse normals to involute $\boldsymbol{n}_1^{2t} = \begin{Bmatrix}\rho_{b1}\cos(\phi_{12b})\left[(s\theta_1 - \theta_1 c\theta_1)(1 - z_{1/2}) + \theta_1 c\theta_1\right] - \rho_{b1}\sin(\phi_{12b})\left[(c\theta_1 + \theta_1 s\theta_1)(1 - z_{1/2}) - \theta_1 s\theta_1\right] - E_c\\\rho_{b1}\sin(\phi_{12b})\left[(s\theta_1 - \theta_1 c\theta_1)(1 - z_{1/2}) + \theta_1 c\theta_1\right] + \rho_{b1}\cos(\phi_{12b})\left[(c\theta_1 + \theta_1 s\theta_1)(1 - z_{1/2}) - \theta_1 s\theta_1\right] - E_c\end{Bmatrix}$		

Table L.1 Equations of curve 1, i.e. the cutter's involute flank that generates the internal gear's involute flank

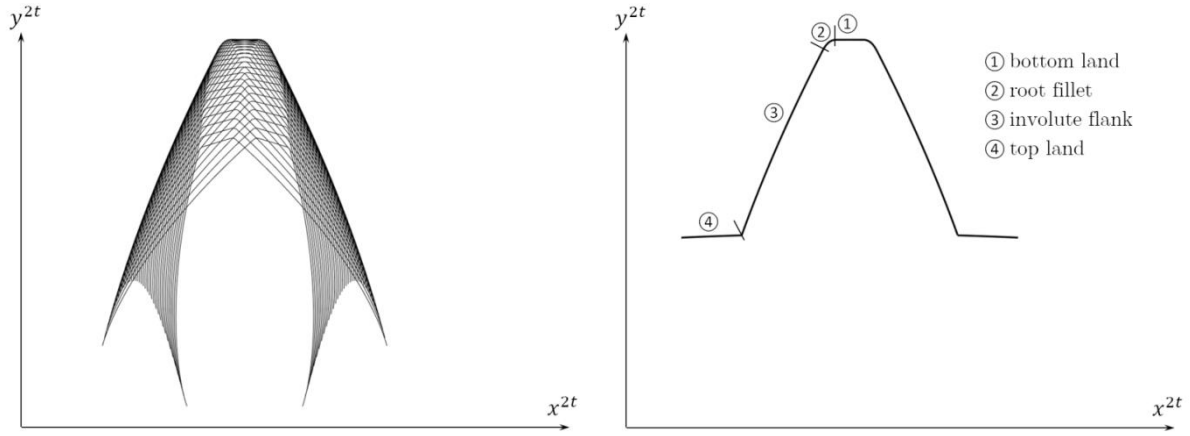


Fig. L.4 Family of pinion cutter profiles represented in the gear blank axis system (left) and the corresponding envelope curves (right)

Begin point: x_0^{1t}	$x_{2,0}^{1t} = -\rho_{a1}\sin\psi_{a1} + \rho_r[\sin\psi_{a1} - \sin(\psi_{a1} + \alpha_r)]$	
Begin point: y_0^{1t}	$y_{2,0}^{1t} = \rho_{a1}\cos\psi_{a1} - \rho_r[\cos\psi_{a1} - \cos(\psi_{a1} + \alpha_r)]$	
Curve equation in $\boldsymbol{x^{1t}y^{1t}}$	$\boldsymbol{r}_2^{1t} = \begin{Bmatrix} -(\rho_{a1} - \rho_r)\sin\psi_{a1} - \rho_r\sin(\psi_{a1} + \alpha_r - \theta_2) \\ (\rho_{a1} - \rho_r)\cos\psi_{a1} + \rho_r\cos(\psi_{a1} + \alpha_r - \theta_2) \end{Bmatrix}$	$0 \le \theta_2 \le \alpha_r$
Curve equation in $\boldsymbol{x^{2t}y^{2t}}$	$\boldsymbol{r}_2^{2t} = \begin{Bmatrix} -(\rho_{a1} - \rho_r)\sin[\phi_1 - \phi_2 + \psi_{a1}] - \rho_r\sin[\phi_1 - \phi_2 + \psi_{a1} + \alpha_r - \theta] + E_c\sin\phi_2 \\ (\rho_{a1} - \rho_r)\cos[\phi_1 - \phi_2 + \psi_{a1}] + \rho_r\cos[\phi_1 - \phi_2 + \psi_{a1} + \alpha_r - \theta] + E_c\cos\phi_2 \end{Bmatrix}$	
Equation of meshing	$\rho_1\sin(\phi_1 + \psi_{a1} + \alpha_r - \theta_2) - (\rho_{a1} - \rho_r)\sin(\theta_2 - \alpha_r) = 0$	
Equation of root fillet	*Combine the above two equations*	

Table L.2 Equations of curve 2, i.e. the cutter's rounded corner that generates the internal gear's root fillet

Begin points: x_0^{1t} and y_0^{1t}	$x_{3,0}^{1t} = -\rho_{a1} \sin\psi_{a1}$	$y_{3,0}^{1t} = \rho_{a1} \cos\psi_{a1}$
Curve equation in $\boldsymbol{x^{1t}y^{1t}}$	$\boldsymbol{r}_3^{1t} = \rho_{a1} \begin{Bmatrix} -\sin(\psi_{a1} - \theta_3) \\ \cos(\psi_{a1} - \theta_3) \end{Bmatrix}$	$0 \le \theta_3 \le \psi_{a1}$
Curve equation in $\boldsymbol{x^{2t}y^{2t}}$	$\boldsymbol{r}_3^{2t} = \rho_{a1} \begin{Bmatrix} -\sin[\phi_1 - \phi_2 + \psi_{a1} - \theta_3] + E_c \sin\phi_2 \\ \cos[\phi_1 - \phi_2 + \psi_{a1} - \theta_3] + E_c \cos\phi_2 \end{Bmatrix}$	
Equation of meshing	$\rho_1 \sin(\phi_1 + \psi_a - \theta_3) = 0$	
Equation of root fillet	*Combine the above two equations*	

Table L.3 Equations of curve 3, i.e. the cutter's top land that generates the internal gear's bottom land

Symbol	**Meaning**	**Equation**
ρ_b	Base radius	$\rho_b = \rho \cos\alpha_t$
ρ_r	Radius of rounded corner	*User-specified*
α_r	Angle or rounded corner	*Computed numerically based on Fig. L.3*
ψ	Tooth thickness half angle at pitch circle	$\psi = \dfrac{\pi + 4x \tan\alpha_n}{2z}$
ψ_b	Tooth thickness half angle at base circle	$\psi_b = \psi + \tan\alpha_t - \alpha_t$
ψ_a	Tooth thickness half angle at top circle	*Computed numerically based on Fig. L.3*
$\theta_{1,max}$	Maximal parameter value of involute curve	*Computed numerically based on Fig. L.3*
ϕ_{12b}	Summation of angles	$\phi_{12b} = \phi_1 - \phi_2 + \psi_b$
$z_{1/2}$	Tooth ratio	$z_{1/2} = z_1/z_2$

Table L.4 Symbols used in Table L.1 - Table L.3; all quantities refer to the pinion cutter, unless indicated otherwise

Appendix M

Fundamental kinematic relations of involute gears

In this appendix an overview is given of some fundamental kinematic relations that relate to the contact conditions of gear pairs. Of particular interest are the initial and final points of tooth contact, and the evolution of the contact point along the flank of the tooth as the rotation angle of the driving gear is changed.

M.1 Characteristic points of contact

In practice, gears are usually mounted at a center distance a_w that is different from the theoretical center distance $a = (d_2 \pm d_1)/2$, where d_1 and d_2 denote the pitch diameters of the pinion (i.e. smaller gear) and wheel (i.e. larger gear) of the gear pair, respectively, and where the minus sign is used when the second gear is an internal gear. As a consequence, the pitch point (i.e. the instantaneous center of rotation) will displace and the pressure angles of the gears will change. The equations for the *working* pitch diameters and the *working* pressure angle are summarized in Table M.1.

Symbol	Meaning	Equation
a_w	Working center distance	$a_w = (d_{w2} \pm d_{w1})/2$
d_w	Working pitch radius	$d_w = d_b / \cos \alpha_{wt}$
α_{wt}	Working transverse pressure angle	$\alpha_{wt} = \cos^{-1}\left[\lvert z_1 \pm z_2 \rvert \left(\frac{m_n \cos \alpha_t}{2a_w \cos \beta}\right)\right]$

Table M.1 Working pitch radii, pressure angle and working depth for a gear pair at center distance a_w

When two gears engage, the active flanks of the respective teeth are bounded by two diameters: the active root diameter d_{Nf} and the active tip diameter d_{Na}, see Fig. M.1. These diameters depend on the characteristics of both gears and on the working center distance a_w (see Table M.2). The path of contact of the gear pair coincides with the line of action, and extends from point A (the initial point of contact) to point E (the final point of contact). It passes through the pitch point C, which is characterized by a state of pure rolling contact between two mating tooth flanks. At point A, the active root diameter of the driving gear and the active top diameter of the driven gear cross the line of action. At point E, the active top diameter of the driving gear and the active root

diameter of the driven gear cross the line of action. The transverse gear contact angles corresponding to points A, C, and E are summarized in Table M.3.

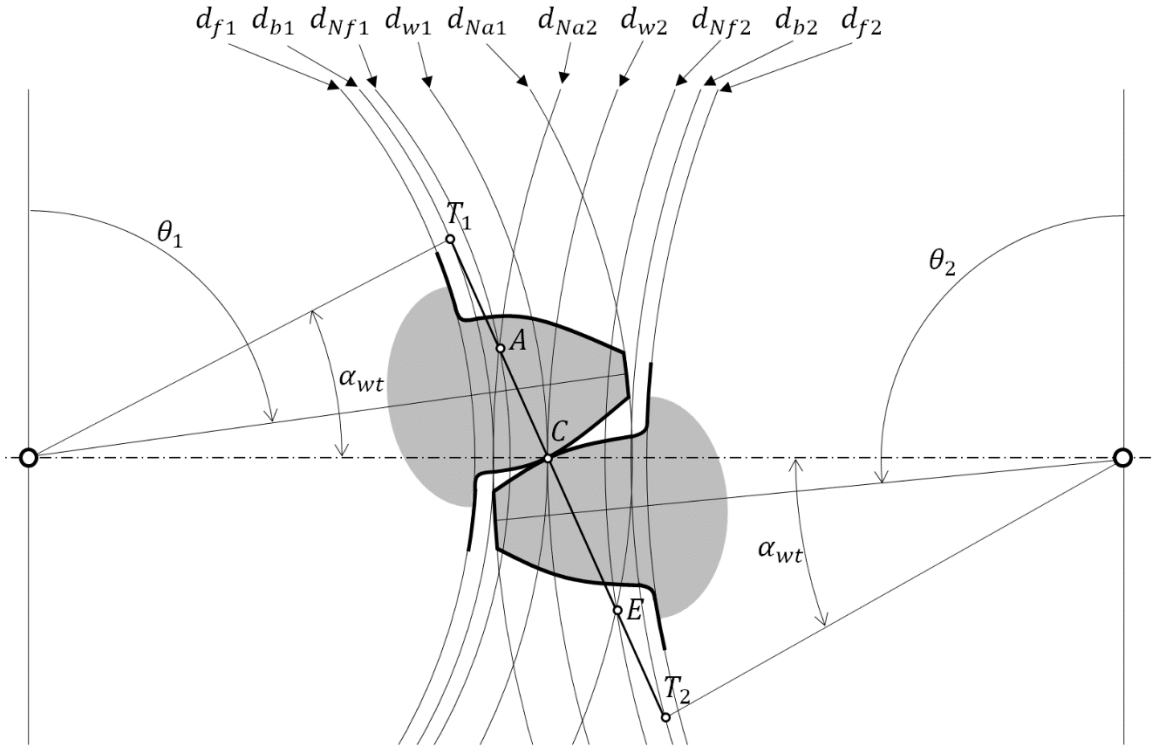


Fig. M.1 Active tooth flanks in a transverse section of an external gear pair

Symbol	Meaning	Equation
d_{Nf1}	Active root diameter pinion	$d_{Nf1} = \max\left(\sqrt{\left(2a_w \sin\alpha_{wt} \mp \sqrt{d_{Fa2}^2 - d_{b2}^2}\right)^2 + d_{b1}^2}, d_{Ff1}\right)$
d_{Nf2}	Active root diameter wheel	$d_{Nf2} = \max\left(\sqrt{\left(2a_w \sin\alpha_{wt} - \sqrt{d_{Fa1}^2 - d_{b1}^2}\right)^2 + d_{b2}^2}, d_{Ff2}\right)$
d_{Na1}	Active top diameter pinion	if $d_{Nf2} = d_{Ff2}$ then $d_{Na1} = \sqrt{\left(2a_w \sin\alpha_{wt} \mp \sqrt{d_{Ff2}^2 - d_{b2}^2}\right)^2 + d_{b1}^2}$; else $d_{Na1} = d_{Fa1}$
d_{Na2}	Active top diameter wheel	if $d_{Nf1} = d_{Ff1}$ then $d_{Na2} = \sqrt{\left(2a_w \sin\alpha_{wt} - \sqrt{d_{Ff1}^2 - d_{b1}^2}\right)^2 + d_{b2}^2}$; else $d_{Na2} = d_{Fa2}$
d_{Ff}	Root form diameter	The start of the involute portion of the gear profile, i.e. the point where the root fillet evolves into the involute flank.
d_{Fa}	Tip form diameter	The end of the involute portion of the gear profile, i.e. the point where the involute flank evolves into the top land.

Table M.2 Active root and top diameters of the driving gear (pinion) and the driven gear (wheel)

Symbol	Meaning	Pinion	Wheel
θ_T	Gear angle at point T	$\theta_{T1} = \pi/2 - \alpha_{wt} - \psi_{b1}$	$\theta_{T2} = \pi/2 + \alpha_{wt} + \psi_{b2}$
θ_A	Gear angle at point A	$\theta_{A1} = \pi/2 - \alpha_{wt} + \alpha_{Nf1} - \psi_{Nf1}$	$\theta_{A2} = \pi/2 + \alpha_{wt} - \alpha_{Na2} + \psi_{Na2}$
θ_C	Gear angle at point C	$\theta_{C1} = \pi/2 - \psi_1$	$\theta_{C2} = \pi/2 + \psi_2$
θ_E	Gear angle at point E	$\theta_{E1} = \pi/2 - \alpha_{wt} + \alpha_{Na1} - \psi_{Na1}$	$\theta_{E2} = \pi/2 + \alpha_{wt} - \alpha_{Nf2} + \psi_{Nf2}$
α_{Nf}, α_{Na}	Transverse pressure angle at d_{Nf} and d_{Na}	$\alpha_{Nf} = \cos^{-1}\left(\frac{d_b}{d_{Nf}}\right), \alpha_{Na} = \cos^{-1}\left(\frac{d_b}{d_{Na}}\right)$	
ψ_{Nf}, ψ_{Na}	Tooth thickness half angle at d_{Nf} and d_{Na}	$\psi_{Nf} = \psi \pm (\tan\alpha_t - \tan\alpha_{Nf} - \alpha_t + \alpha_{Nf})$ $\psi_{Na} = \psi \pm (\tan\alpha_t - \tan\alpha_{Na} - \alpha_t + \alpha_{Na})$	

Table M.3 Transverse gear contact angles at the points T, A, C, and E

The linear distance between points A and E is defined as the *length of the path of contact* of two mating cylindrical gears, and is given by:

$$g_\alpha = \overline{AE} = \frac{1}{2}\left[\sqrt{d_{Na1}^2 - d_{b1}^2} \pm \left(\sqrt{d_{Na2}^2 - d_{b2}^2} - 2a_w \sin\alpha_{wt}\right)\right] \tag{M.1}$$

The angle through which a gear rotates from start to finish of transverse tooth engagement is called the transverse angle of transmission φ_α, and is related to the length of the path of contact g_α as follows:

$$\varphi_{\alpha 1} = \theta_{E1} - \theta_{A1} = \frac{2g_\alpha}{d_{b1}} = \left(\frac{z_2}{z_1}\right)\varphi_{\alpha 2} = \left(\frac{z_2}{z_1}\right)\frac{2g_\alpha}{d_{b2}} \tag{M.2}$$

The ratio of this angle to the angular pitch $\tau = 2\pi/z$ is referred to as the transverse contact ratio ε_α:

$$\varepsilon_\alpha = \frac{\varphi_{\alpha 1}}{\tau_1} = \frac{\varphi_{\alpha 2}}{\tau_2} \tag{M.3}$$

It defines the average number of teeth pairs in contact in a transverse section as the gear rotates over one angular pitch. Note that while for spur gears the total contact ratio is equal to the transverse contact ratio, this is not the case for helical gears, where the contact ratio (as well as the actual contact angles) depends on the active face width of the helical gear pair.

M.2 Radius of curvature and sliding speed at the contact point

As a final important piece of information, the radius of curvature, sliding speed and diameter at the point of contact are expressed as a function of the angular displacement of the gear. Assume the gear is at an angle θ and contact occurs at the point Y, see Fig. M.2. The line of action is the tangent to the base circle that passes through the points T and Y, and is locally perpendicular to the tooth surface at Y. The current angular configuration of the gear can be expressed in terms of the *roll angle* ξ_y of the gear and of the angle θ_T, as defined in Table M.3. The former is defined as the angle over the center of the base circle from the origin U of the involute to the point T, and can be obtained (for the pinion and wheel) from the following equations:

$$\theta_1 = \theta_{T1} + \xi_{y1} = \pi/2 - \alpha_{wt} - \psi_{b1} + \xi_{y1} \quad \rightarrow \quad \xi_{y1}(\theta_1) = \theta_1 - \pi/2 + \alpha_{wt} + \psi_{b1} \tag{M.4a}$$

$$\theta_2 = \theta_{T2} - \xi_{y2} = \pi/2 + \alpha_{wt} + \psi_{b2} - \xi_{y2} \quad \rightarrow \quad \xi_{y2}(\theta_2) = -\theta_2 + \pi/2 + \alpha_{wt} + \psi_{b2} \tag{M.4b}$$

The roll angle ξ_y is related to the transverse working pressure angle α_{yt} and the radius of curvature ϱ_y at the contact point through the following equations:

$$\alpha_{yt}(\theta) = tan^{-1}\,\xi_y\,(\theta) \tag{M.5}$$

$$\varrho_y(\theta) = \pm\frac{d_b}{2}\xi_y(\theta) \tag{M.6}$$

Let ϱ_{y1} and ϱ_{y2} denote the radii of curvature at point Y of the pinion and wheel, respectively, then the sliding velocity at point Y is defined as the difference in speeds of the two transverse profiles in the direction of the common tangent at Y, i.e.:

$$v_{slide}(\theta_1, \theta_2) = \dot{\theta}_1\left[\frac{\varrho_{y2}(\theta_2)}{u} - \varrho_{y1}(\theta_1)\right] \tag{M.7}$$

where $\dot{\theta}_1$ is the angular velocity of the pinion. Note that at the pitch point $\varrho_{y2} = u\varrho_{y1}$; hence the sliding velocity is zero. Finally, the contact diameter d_y can be computed from:

$$d_y(\theta) = \frac{d_b}{cos\,\alpha_{yt}(\theta)} \tag{M.8}$$

Eq. M.4-M.8 thus allow to express the radius of curvature, the pressure angle, the sliding velocity and the diameter at the point of contact as a function of the rotation angle of the gear.

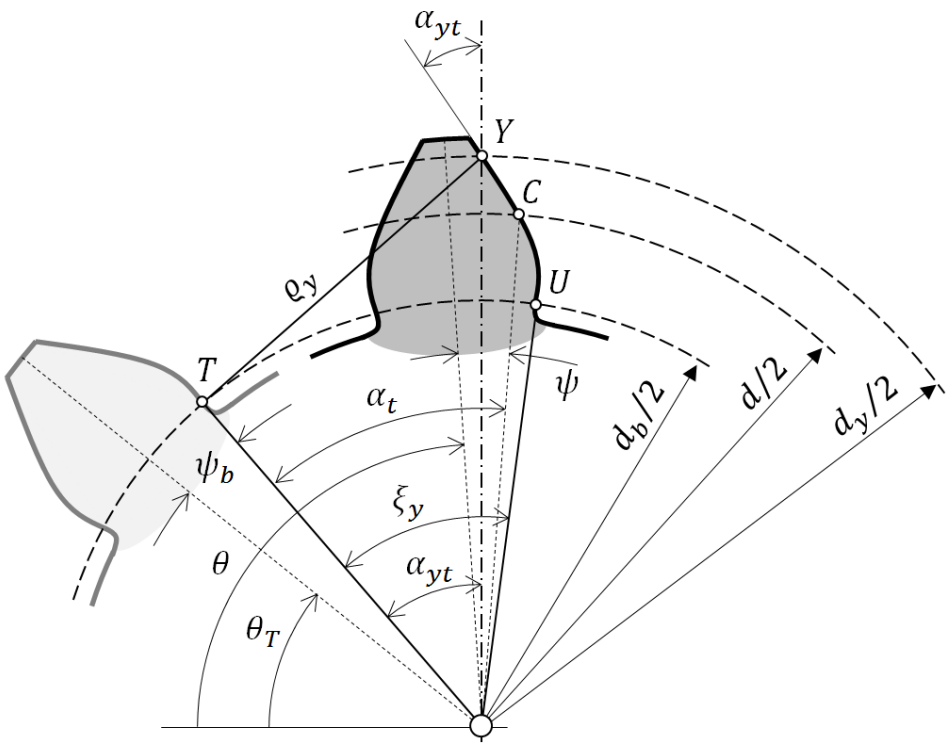


Fig. M.2 Definition of the roll angle, radius of curvature, and contact diameter at the point of contact Y

Appendix N

State-of-the-art gear contact models

N.1 A lumped-parameter gear contact model

The lumped parameter approach was introduced in Section 1.2.1 as an effective means for system-level gear contact analysis. In this approach, the flexibility of a pair of engaging teeth is *lumped* into a spring element that acts between the teeth (see Fig. N.1). The element is directed along the gear pair's line of action and models the contact forces between the teeth. Having lumped the flexibility of the gears into discrete spring elements, the gears themselves are modelled as rigid bodies and the location of the contact points are computed using the kinematic relations developed in Appendix M[30]. The equivalent contact force between two teeth is proportional to the penetration of one rigid tooth flank into the other, and is computed from (see Fig. N.1):

$$e_i = |T_1Y_1|_i + |T_2Y_2|_i - |T_1T_2| = \frac{d_{b1}}{2}\xi_{y1,i}(\theta_{1,i}) + \frac{d_{b2}}{2}\xi_{y2,i}(\theta_{2,i}) - a_w \sin\alpha_{wt} \qquad \text{(N.1)}$$

where use was made of Eq. M.6 and where the roll angles ξ_{y1} and ξ_{y2} are computed from Eq. M.4. The contact force $f_{c,i}$ between the *i*-th tooth pair is then obtained as

$$f_{c,i} = e_i k_i = e_i \frac{k_{t,i1} \times k_{t,i2}}{k_{t,i1} + k_{t,i2}} \qquad \text{(N.2)}$$

where k_i is the tooth pair stiffness, and $k_{t,1i}$ and $k_{t,2i}$ are the tooth contact stiffnesses of the respective gears.

Among the various approaches that exist in literature to compute these contact stiffnesses, the approach by Kuang & Yang [13] is considered in this dissertation due to its widespread use in the MBSD community. Kuang and his co-workers obtained the tooth stiffness of a large variety of gears by successively applying unit loads along the involute flanks of the gear and computing the resulting displacements using a quadratic plane-strain FE model. The results of these analyses were

[30] The assumption that is made here is that contact occurs along the theoretical line of action, and that the direction of the line of action is solely determined by the center distance and gear parameters, i.e. it is not influenced by the deformation of the gears.

then curve-fitted to obtain a function that yields the tooth contact stiffness in terms of the contact diameter d_y, i.e.:

$$k_t(d_y) = (a_0 + a_1 x) + (a_2 + a_3 x)\frac{d_y - d}{2(1+x)m} \tag{N.3}$$

where the coefficients a_i are cubic functions of the number of teeth, and x and m are the gear's profile shift and normal module. The contact stiffness k_t is expressed in units $[N/\mu m/mm]$ and is defined as the load that is necessary to deform a single gear tooth having $1\ mm$ face width by an amount of $1\ \mu m$ along the line of action.

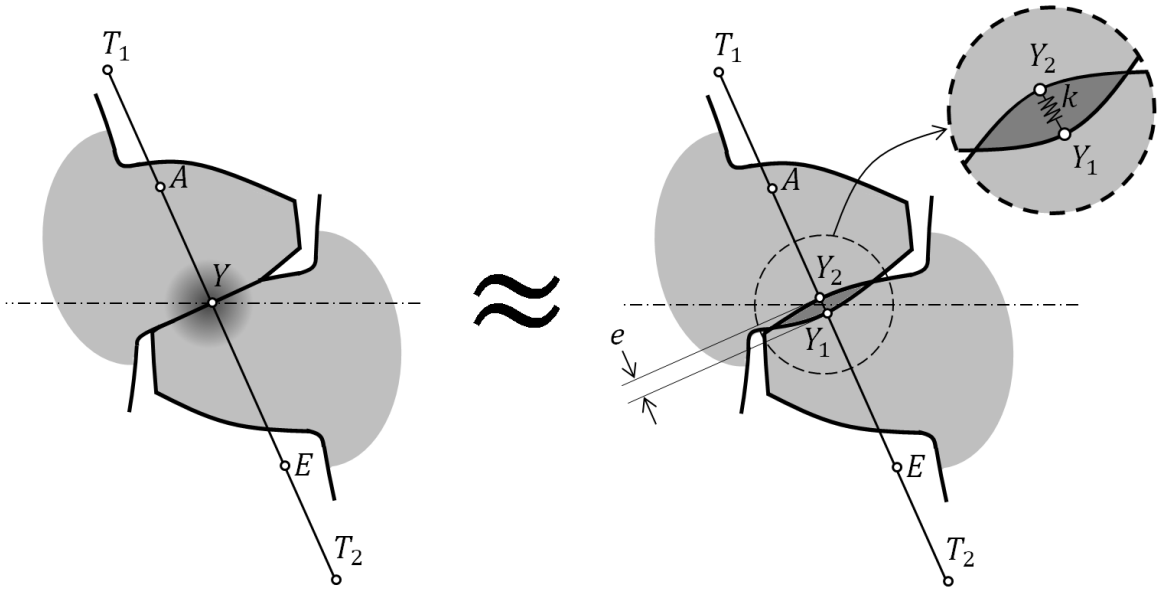


Fig. N.1 Lumped parameter approximation of the tooth contact flexibility

While Eq. N.3 was developed for (perfectly aligned) spur gears, the same formula can be applied to helical (and misaligned spur) gears by *slicing* the gear in a number of thin spur gear slices (see Fig. N.2) and computing the penetration e and corresponding contact force f_c independently for each slice. The contact moment $\boldsymbol{m}_{c,ij}$ about the gear center due to the slice contact force $\boldsymbol{f}_{c,ij}$ acting on the *j*-th slice of the *i*-th teeth pair in contact is obtained from

$$\begin{aligned}\boldsymbol{m}_{c,ij} &= \boldsymbol{r}_{c,ij} \times \boldsymbol{f}_{c,ij} \\ &= \begin{Bmatrix} (d_y/2)\cos\left(\frac{\pi}{2} - \alpha_{wt} + \alpha_{yt}\right) \\ (d_y/2)\sin\left(\frac{\pi}{2} - \alpha_{wt} + \alpha_{yt}\right) \\ \frac{b}{2}\left(1 - \frac{2j-1}{n_{slice}}\right) \end{Bmatrix} \times f_{c,ij} \begin{bmatrix} \cos\beta & 0 & -\sin\beta \\ 0 & 1 & 0 \\ \sin\beta & 0 & \cos\beta \end{bmatrix} \begin{Bmatrix} -\sin\alpha_{wt} \\ \cos\alpha_{wt} \\ 0 \end{Bmatrix}\end{aligned} \tag{N.4}$$

where it was assumed that the axial coordinate of the central slice is zero, and where n_{slice} denotes the number of axial slices.

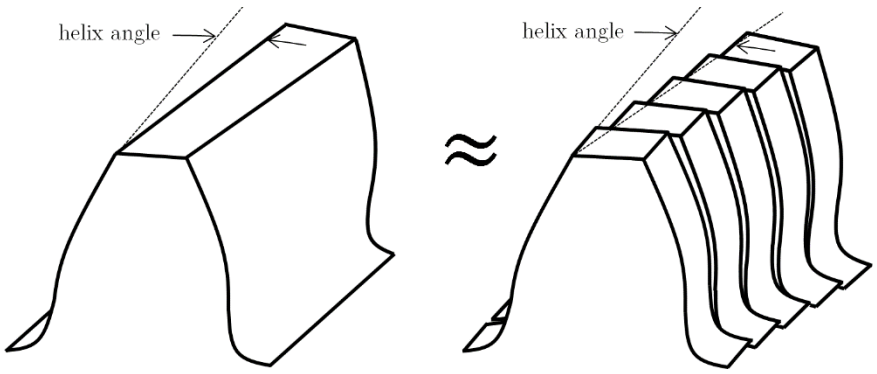


Fig. N.2 Thin slice representation of a helical gear

The lumped parameter model used in this dissertation is obtained as a simplified version of the multibody model of Eq. 3.2.6, where all elastic generalized coordinates are eliminated and the contact forces are computed using Eq. N.1-N.2. The lumped parameter model thus has two degrees of freedom, namely the rotation angle of the driving and driven gear. The number of slices is equal to the number of elements along the width of the gear's finite element model. The procedure for computing the corresponding generalized contact forces (i.e. the contact moments about the gears' axes) is given in Algorithm N.1. Note that in this approach, the slices are uncoupled, meaning that a contact force acting on one slice will not deform the adjacent slices. Furthermore, three important effects are not considered in the stiffness model of Kuang and Yang, namely the torsional stiffness of the gear blank (which can be included as a constant stiffness in series with the gear mesh stiffness), three-dimensional effects such as edge effects, and the non-linear dependency of the contact stiffness on the magnitude of the applied load.

N.2 A semi-analytical gear contact model

The finite element method offers an effective means for computing deformations of arbitrarily shaped flexible bodies, yet requires a huge amount of finite elements in regions where contact interactions can occur. This results in a prohibitively large number of nodal degrees of freedom and highly complex contact forces. One way to alleviate the associated computational burden is to develop dedicated model-order reduction and hyperreduction schemes, as is the objective of this dissertation. Another approach is to combine the finite element method with (semi-)analytical contact solutions, as was done by Vijayakar [30, 31] and, most notably, Andersson & Vedmar [19].

Algorithm N.1 Computation of the lumped parameter contact torques

Input: gear angles θ_1 and θ_2

Output: contact torques m_{c1} and m_{c2}

1. Determine teeth in contact by comparison of θ with θ_A and θ_E: $z_{tic,1}$ and $z_{tic,2}$
2. **for** $i = 1..n_{tic}$ **do**
3. Compute tooth angle: $\theta_i = \theta - (z_{tic,1}(i) - 1) \cdot 2\pi/z$
4. **for** $j = 1..n_{slice}$ **do**
5. Compute slice angle: $\theta_{ij} = \theta_i + 0.5\beta'[1 - (2j - 1)/n_{slice}]$
6. Compute the roll angles using Eq. M.4: ξ_y
7. Compute the flank penetration using Eq. N.1: e_{ij}
8. Compute the contact diameters using Eq. M.8: d_y
9. Compute the tooth stiffnesses at the contact point using Eq. N.3: $k_{t,ij}$
10. Compute the contact force using Eq. N.2: $f_{c,ij}$
11. Compute the contact moment about the gear axes using Eq. N.4: $m_{c,ij}$
12. Add all contact moments: $m_c = m_c + m_{c,ij}$
13. **end**
14. **end**

The key idea of the method of Andersson & Vedmar is to separate the deformation of a gear into 1) the bulk or *global* deformation of the teeth and gear body, and 2) the *local* deformation of the flank near the contact point (see Fig. N.3). In accordance with Saint-Venant's principle[31] [195], the global deformation of the gear is unaffected by the precise distribution of the contact load, and depends linearly on the magnitude of this load. The local deformation, on the other hand, is affected only by the local flank parameters; in accordance with Hertz' contact theory, it is nonlinearly dependent on the magnitude of the applied load.

Following the above reasoning, the effects of a distributed contact load can be captured by superimposing the following three load cases (see Fig. N.4):

1) A statically equivalent point load acting at the center of pressure of the distributed contact load;
2) The same point load with opposite direction acting on a fixed border of the loaded tooth;
3) The distributed contact load acting on the same fixed boarder of the loaded tooth.

[31] Saint-Venant's principle may be expressed as follows: the difference between the effects of two different but statically equivalent loads becomes very small at sufficiently large distances from the load.

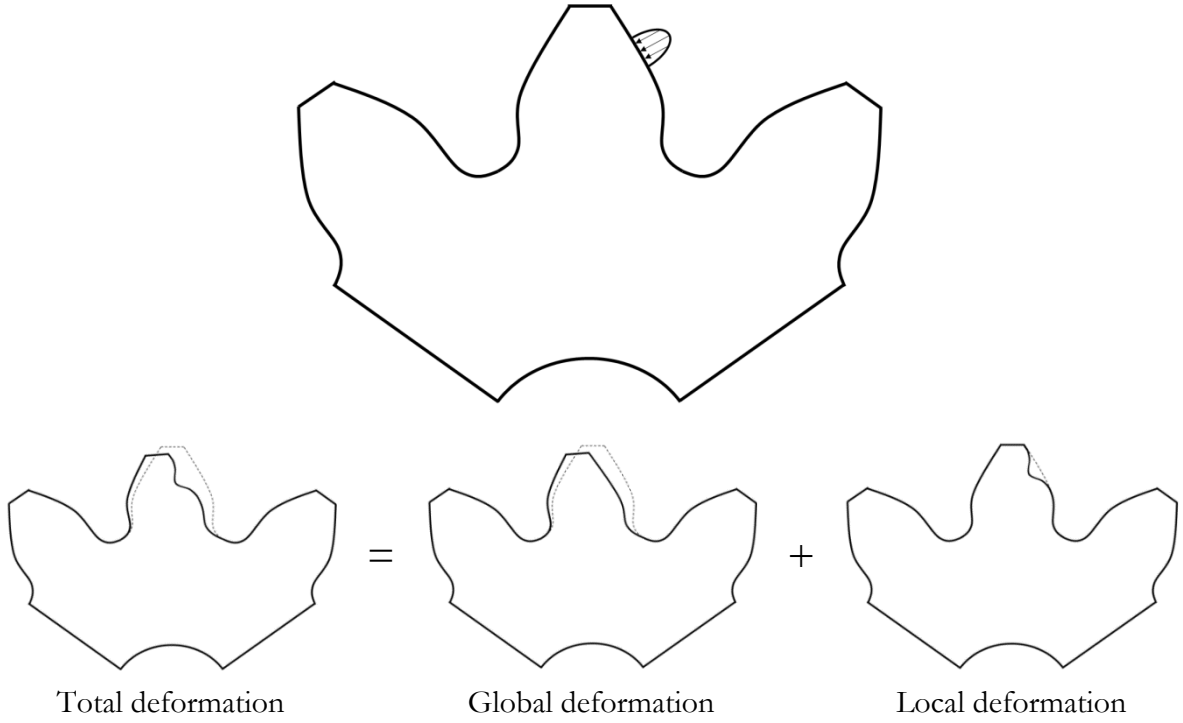


Fig. N.3 Separation of the total deformation of a gear into a global and local contribution

Significant savings in computational cost are achieved by evaluating the former two load cases (which add up to the global gear deformation) using a coarse (linear) FE model, and solving the latter using (nonlinear) analytical contact formulas. The formula used by Andersson & Vedmar is due to Weber & Banasheck [164], which yields the normal displacement of an involute tooth flank subjected to a Hertzian contact pressure with load intensity q at the depth h:

$$u_{local} = \frac{2(1-\nu^2)}{\pi E} q \left[ln\left(\frac{h}{L} + \sqrt{1+\left(\frac{h}{L}\right)^2} \right) - \frac{\nu}{1-\nu}\left(\frac{h}{L}\right)^2 \left(\sqrt{1+\left(\frac{h}{L}\right)^2} - 1 \right) \right] \qquad \text{(N.5)}$$

where $2L$ is the extension of the load along the flank, computed as (compare with Eq. 3.2.3):

$$L = \sqrt{\frac{4}{\pi} \frac{(1-\nu_1^2)/E_1 + (1-\nu_2^2)/E_2}{1/\varrho_1 + 1/\varrho_2}} \qquad \text{(N.6)}$$

where the subscripts '1' and '2' refer to the driving and driven gear, respectively. Computing the global deformation involves an *offline* phase based on linear finite element analysis, and an *online* phase based on the interpolation of FE results. During the offline phase, the global deformation of a gear flank is mapped by successively applying unit normal loads to the respective nodes on the gear flank and storing the resulting normal displacements of all the nodes on the flank (after elimination of the unphysical local deformation, as in load case 2). This results in a four-

dimensional $(N_{fef}N_{few}) \times (N_{fef}N_{few})$ compliance matrix (with N_{fef} and N_{few} the number of nodes along the flank and along the width of the gear, respectively) that can either be interpolated directly, or approximated by closed-form interpolation formulas.

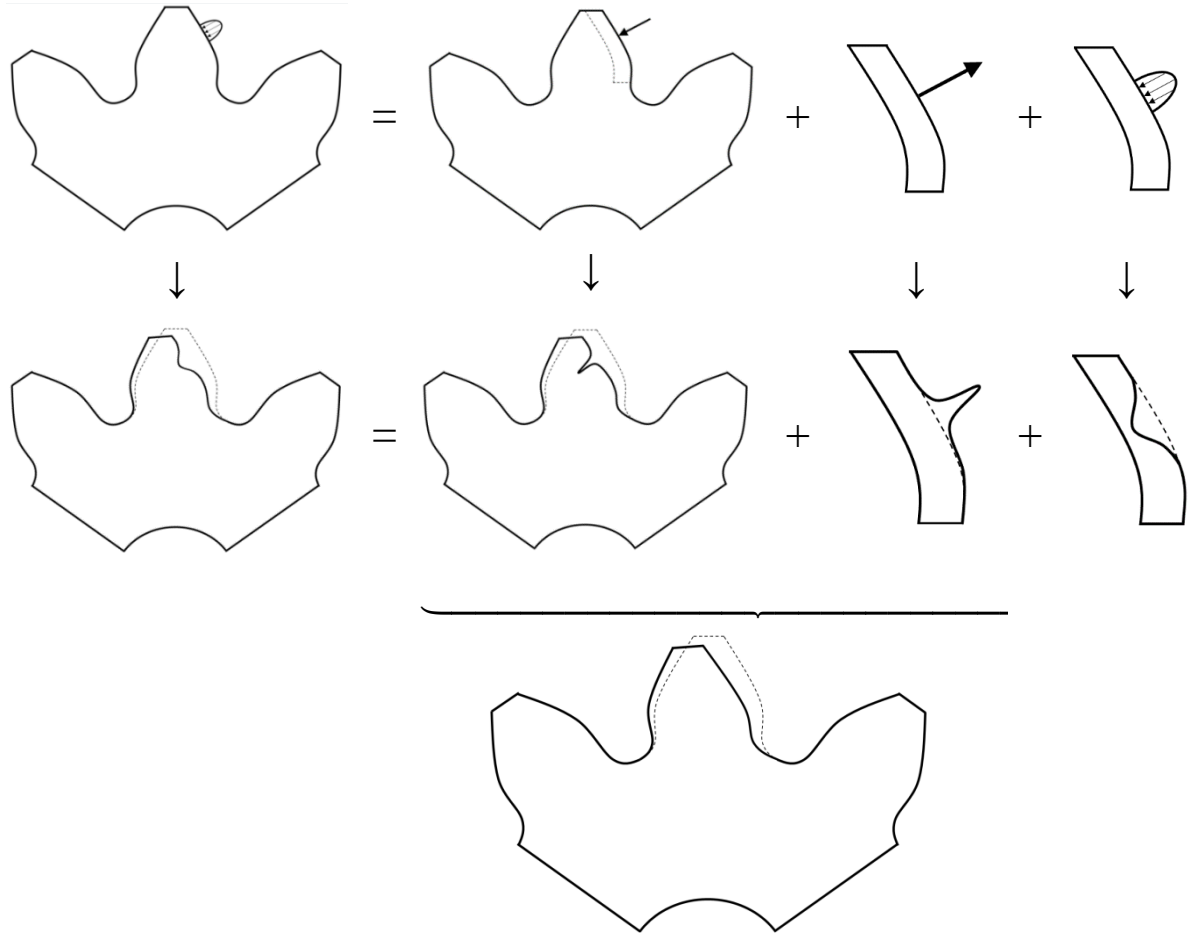


Fig. N.4 A distributed contact load as a superposition of three load cases

The semi-analytical gear contact model used in this dissertation is again obtained by eliminating the generalized elastic coordinates from the multibody model of Eq. 3.2.6, and keeping only the two rotational degrees of freedom of the gears. Similar to the lumped parameter model of Section N.1, the contact location along the flank is computed using the kinematic relations developed in Appendix M (i.e. the line of action is assumed to be contact). The gear is divided in a number of slices (see Fig. N.2), and the flank penetration e (Eq. N.1) is computed for each slice. Then, in order to compute the slice contact forces $f_{c,ij}, j = 1..n_{slice}$, the following system of nonlinear equations is solved:

$$e_{ij} - \left[u_{global,1}(f_{c,ij}) + u_{local,1}(f_{c,ij})\right] - \left[u_{global,2}(f_{c,ij}) + u_{local,2}(f_{c,ij})\right] = 0 \qquad \text{(N.7)}$$

where u_{global} is the global deformation of the gear due to the slice contact loads on all n_{slice} slices, and u_{local} represents the local deformation due to the load intensity $q = f_{c,ij}/b_j$ at slice *j*, computed using Eq. N.5. Note that the slices are coupled through the global deformation, whereas the local deformation is uncoupled. Having obtained the slice contact forces, these are translated into contact moments about the gear axis using Eq. N.4. In summary, Algorithm N.1 can be applied also for the semi-analytical gear contact model by eliminating Steps 8 and 9, and replacing Eq. N.1 by Eq. N.7 in Step 10.

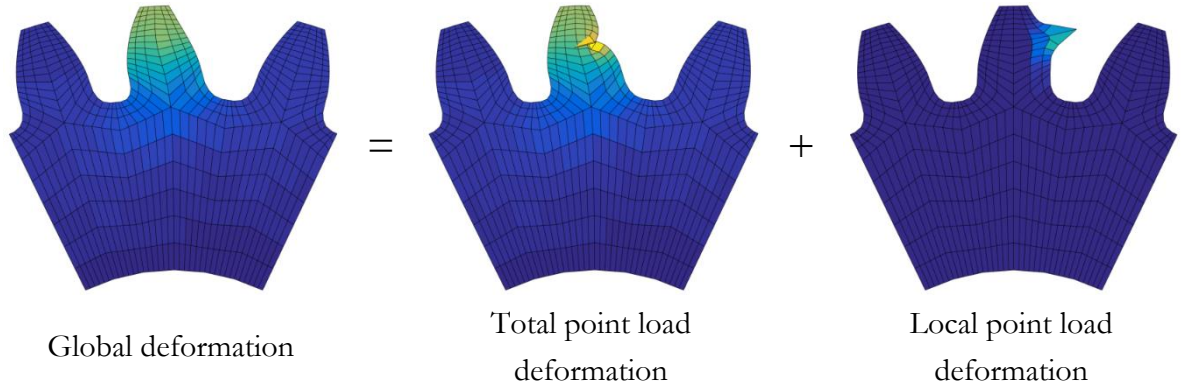


Fig. N.5 Separation of the total deformation of a gear into a global and local contribution

N.3 A fully-nonlinear FE-based gear contact model

As a final source of numerical validation, the commercial non-linear finite element package ABAQUS [196] is used to conduct fully nonlinear FE-based contact simulations. The FE models decribed in Section 3.2.1 are imported into ABAQUS using the Abaqus Input File. Static contact analyses are performed using ABAQUS/STANDARD, which uses Newton's method to incrementally solve the nonlinear equilibrium equations. The contact detection algorithm is based on a double-pass surface-to-surface algorithm (see Section 2.5.2.1), and the contact constraints are enforced using the penalty method (see Section 2.5.3.2) with a user-defined penalty factor. Dynamic contact analyses, on the other hand, are conducted using ABAQUS/EXPLICIT, which uses the explicit conditionally stable central-difference method (see Section 2.6.1.3) with velocities lagging half a time increment behind (see Eq. 2.6.23) and with step-size control based on the stability limit of Eq. 2.6.25. Expensive mass matrix inversions are avoided by using diagonally lumped mass matrices. The contact detection algorithm is based on a double-pass node-to-surface algorithm, with contact constraints enforced using a penalty approach in which the penalty factor is automatically selected by the algorithm. In order to minimize the CPU cost of the contact search algorithm, so-called *contact pairs* are defined that select the tooth flanks that will engage with each other. While the CMS-based ROM (Section 3.2.3) is the primary reference model for evaluating the *computational efficiency* of the methods developed in this dissertation, the ABAQUS-based gear contact model is the primary reference for evaluating the *accuracy* of the developed gear models.

References

[1] M. Rüssmann, M. Lorenz, P. Gerbert, M. Waldner, J. Justus, P. Engel and M. Harnisch, „Industry 4.0: The future of productivity and growth in manufacturing industries," *Boston Consulting Group,* vol. 9, 2015.

[2] E. J. Haug, Computer aided kinematics and dynamics of mechanical systems, vol. 1, Allyn and Bacon Boston, 1989.

[3] E. T. Whittaker, A treatise on the analytical dynamics of particles and rigid bodies, Cambridge University Press, 1988.

[4] R. R. Craig and A. J. Kurdila, Fundamentals of structural dynamics, John Wiley & Sons, 2006.

[5] J. N. Reddy, An introduction to continuum mechanics, Cambridge university press, 2013.

[6] A. A. Shabana, Dynamics of multibody systems, Cambridge university press, 2013.

[7] K. L. Johnson, „One hundred years of Hertz contact," *Proceedings of the Institution of Mechanical Engineers,* vol. 196, pp. 363-378, 1982.

[8] A. Popp and others, „Mortar methods for computational contact mechanics and general interface problems," 2012.

[9] I. Temizer, P. Wriggers and T. J. R. Hughes, „Contact treatment in isogeometric analysis with NURBS," *Computer Methods in Applied Mechanics and Engineering,* vol. 200, pp. 1100-1112, 2011.

[10] P. Wriggers, Computational contact mechanics, Springer Science & Business Media, 2006.

[11] O. C. Zienkiewicz and R. L. Taylor, The finite element method, vol. 3, McGraw-hill London, 1977.

[12] T. J. R. Hughes, J. A. Cottrell and Y. Bazilevs, „Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement," *Computer methods in applied mechanics and engineering,* vol. 194, pp. 4135-4195, 2005.

[13] J. H. Kuang and Y. T. Yang, „An estimate of mesh stiffness and load sharing ratio of a spur gear pair," *Advancing power transmission into the 21 st century,* pp. 1-9, 1992.

[14] Y. Cai and T. Hayashi, „The linear approximated equation of vibration of a pair of spur gears (theory and experiment)," *Journal of Mechanical Design,* vol. 116, pp. 558-564, 1994.

[15] I. S. O. Standard, *ISO 6336-20:2017, Calculation of load capacity of spur and helical gears.*

[16] H. Hertz, Schriften vermischten Inhalts, herausgegeben von Ph. Lenard, vol. 1, JA Barth, 1895.

[17] K. L. Johnson and K. L. Johnson, Contact mechanics, Cambridge university press, 1987.

[18] C. Weber and K. Banaschek, „Schriftenreihe Antriebstechnik," Friedr. Vieweg & sohn, 1953.

[19] A. Andersson and L. Vedmar, „A dynamic model to determine vibrations in involute helical gears," *Journal of Sound and Vibration,* vol. 260, pp. 195-212, 2003.

[20] A. F. Rincon, F. Viadero, R. Sancibrian, P. G. Fernandez and A. De Juan, „A dynamic model for the study of gear transmissions," *WIT transactions on modelling and simulation,* vol. 48, pp. 523-534, 2009.

[21] T. Eritenel and R. G. Parker, „Three-dimensional nonlinear vibration of gear pairs," *Journal of Sound and Vibration,* vol. 331, pp. 3628-3648, 2012.

[22] J. M. De Mul, J. M. Vree and D. A. Maas, „Equilibrium and associated load distribution in ball and roller bearings loaded in five degrees of freedom while neglecting friction_Part I: General theory and application to ball bearings," *Journal of tribology,* vol. 111, pp. 142-148, 1989.

[23] R. J. LeVeque, Finite volume methods for hyperbolic problems, vol. 31, Cambridge university press, 2002.

[24] J. S. Hesthaven, S. Gottlieb and D. Gottlieb, Spectral methods for time-dependent problems, vol. 21, Cambridge University Press, 2007.

[25] W. E. Schiesser, The numerical method of lines: integration of partial differential equations, Elsevier, 2012.

[26] M. A. Crisfield, J. J. C. Remmers, C. V. Verhoosel and others, Nonlinear finite element analysis of solids and structures, John Wiley & Sons, 2012.

[27] M. Géradin and A. Cardona, Flexible multibody dynamics: a finite element approach, Wiley, 2001.

[28] A. A. Shabana, „Computer implementation of the absolute nodal coordinate formulation for flexible multibody dynamics," *Nonlinear Dynamics,* vol. 16, pp. 293-306, 1998.

[29] A. A. Shabana and R. Y. Yakoub, „Three dimensional absolute nodal coordinate formulation for beam elements: theory," *Journal of Mechanical Design,* vol. 123, pp. 606-613, 2001.

[30] S. Vijayakar, „A combined surface integral and finite element solution for a three-dimensional contact problem," *International Journal for Numerical Methods in Engineering,* vol. 31, pp. 525-545, 1991.

[31] R. G. Parker, S. M. Vijayakar and T. Imajo, „Non-linear dynamic response of a spur gear pair: modelling and experimental comparisons," *Journal of Sound and vibration,* vol. 237, pp. 435-455, 2000.

[32] A. C. Antoulas, Approximation of large-scale dynamical systems, vol. 6, Siam, 2005.

[33] R. R. Craig, „A review of time-domain and frequency-domain component mode synthesis methods," *Journal of Modal Analysis,* vol. 2, pp. 59-72, 1987.

[34] J. Fehr, „Automated and Error Controlled Model Reduction in Elastic Multibody Systems. Dissertation," 2011.

[35] B. Besselink, U. Tabak, A. Lutowska, N. Van De Wouw, H. Nijmeijer, D. J. Rixen, M. E. Hochstenbach and W. H. A. Schilders, „A comparison of model reduction techniques from structural dynamics, numerical mathematics and systems and control," *Journal of Sound and Vibration,* vol. 332, pp. 4403-4422, 2013.

[36] P. Ziegler and P. Eberhard, „Simulative and experimental investigation of impacts on gear wheels," *Computer Methods in Applied Mechanics and Engineering,* vol. 197, pp. 4653-4662, 2008.

[37] M. Géradin and D. J. Rixen, „A nodeless dual superelement formulation for structural and multibody dynamics application to reduction of contact problems," *International Journal for Numerical Methods in Engineering,* vol. 106, pp. 773-798, 2016.

[38] G. Virlez, O. Brüls, V. Sonneville, E. Tromme, P. Duysinx and M. Géradin, „Contact model between superelements in dynamic multibody systems," in *Proceedings of the ASME IDETC/CIE*, Portland, 2013.

[39] J. A. Wensing, „On the dynamics of ball bearings," Enschede, 1998.

[40] I. Nakhimovski, „Contributions to the modeling and simulation of mechanical systems with detailed contact analyses," Sweden, 2006.

[41] G. H. K. Heirman, T. Tamarozzi and W. Desmet, „Static modes switching for more efficient flexible multibody simulation," *International Journal for Numerical Methods in Engineering,* vol. 87, pp. 1025-1045, 2011.

[42] T. Tamarozzi, P. Ziegler, P. Eberhard and W. Desmet, „Static modes switching in gear contact simulation," *Mechanism and Machine Theory,* vol. 63, pp. 98-106, 2013.

[43] T. Tamarozzi, „Efficient numerical simulation strategies for flexible multibody systems with variable topology," Belgium, 2014.

[44] S. Gugercin, „Projection methods for model reduction of large-scale dynamical systems," 2003.

[45] E. J. Grimme, „Krylov projection methods for model reduction," 1997.

[46] P. Benner and A. Schneider, „Model order and terminal reduction approaches via matrix decomposition and low rank approximation," J. Roos and L. Costa, Red., Berlin, : Springer-Verlag, 2009, pp. 523-530.

[47] A. Quarteroni, A. Manzoni and F. Negri, Reduced basis methods for partial differential equations: an introduction, vol. 92, Springer, 2015.

[48] P. Benner, A. Cohen, M. Ohlberger and K. Willcox, Model Reduction and Approximation: Theory and Algorithms, vol. 15, SIAM, 2017.

[49] G. Kerschen, J.-c. Golinval, A. F. Vakakis and L. A. Bergman, „The method of proper orthogonal decomposition for dynamical characterization and order reduction of mechanical systems: an overview," *Nonlinear dynamics,* vol. 41, pp. 147-169, 2005.

[50] J. W. Demmel, Applied numerical linear algebra, vol. 56, Siam, 1997.

[51] M. F. A. Azeez and A. F. Vakakis, „Proper orthogonal decomposition (POD) of a class of vibroimpact oscillations," *Journal of Sound and vibration,* vol. 240, pp. 859-889, 2001.

[52] P. Benner, S. Gugercin and K. Willcox, „A survey of projection-based model reduction methods for parametric dynamical systems," *SIAM review,* vol. 57, pp. 483-531, 2015.

[53] B. Haasdonk and M. Ohlberger, „Efficient reduced models and a-posteriori error estimation for parameterized dynamical systems by offline/online decomposition," *Mathematical and Computer Modelling of Dynamical Systems,* vol. 17, pp. 145-161, 2011.

[54] D. Amsallem, J. Cortial, K. Carlberg and C. Farhat, „A method for interpolating on manifolds structural dynamics reduced-order models," *International journal for numerical methods in engineering,* vol. 80, p. 1241, 2009.

[55] D. Amsallem and C. Farhat, „An online method for interpolating linear parametric reduced-order models," *SIAM Journal on Scientific Computing,* vol. 33, pp. 2169-2198, 2011.

[56] D. Amsallem, J. Cortial, K. Carlberg and C. Farhat, „A method for interpolating on manifolds structural dynamics reduced-order models," *International journal for numerical methods in engineering,* vol. 80, p. 1241, 2009.

[57] H. Panzer, J. Mohring, R. Eid and B. Lohmann, „Parametric model order reduction by matrix interpolation," *at-Automatisierungstechnik Methoden und Anwendungen der Steuerungs-, Regelungs-und Informationstechnik,* vol. 58, pp. 475-484, 2010.

[58] T. Tamarozzi, G. H. K. Heirman and W. Desmet, „An on-line time dependent parametric model order reduction scheme with focus on dynamic stress recovery," *Computer Methods in Applied Mechanics and Engineering,* vol. 268, pp. 336-358, 2014.

[59] J. Fiszer, T. Tamarozzi, B. Blockmans and W. Desmet, „A time-dependent parametric model order reduction technique for modelling indirect bearing force measurements," *Mechanism and Machine Theory,* vol. 83, pp. 152-174, 2015.

[60] J. Fiszer, T. Tamarozzi and W. Desmet, „A semi-analytic strategy for the system-level modelling of flexibly supported ball bearings," *Meccanica,* vol. 51, pp. 1503-1532, 2016.

[61] M. Rewienski and J. White, „Model order reduction for nonlinear dynamical systems based on trajectory piewewise-linear approximations," *Linear Algebra and its Applications,* vol. 415, pp. 426-454, 2006.

[62] O. Brüls, P. Duysinx and J.-C. Golinval, „The global modal parameterization for non-linear model-order reduction in flexible multibody dynamics," *International journal for numerical methods in engineering,* vol. 69, pp. 948-977, 2007.

[63] G. Heirman, F. Naets and W. Desmet, „A system-level model reduction technique for the efficient simulation of flexible multibody systems," *International Journal for Numerical Methods in Engineering,* vol. 85, pp. 330-354, 2011.

[64] F. Naets, T. Tamarozzi, G. H. K. Heirman and W. Desmet, „Real-time flexible multibody simulation with Global Modal Parameterization," *Multibody System Dynamics,* vol. 27, pp. 267-284, 2012.

[65] S. Chaturantabut and D. C. Sorensen, „Discrete empirical interpolation for nonlinear model reduction," in *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, 2009.

[66] S. Chaturantabut and D. C. Sorensen, „Nonlinear model reduction via discrete empirical interpolation," *SIAM Journal on Scientific Computing,* vol. 32, pp. 2737-2764, 2010.

[67] C. Farhat, P. Avery, T. Chapman and J. Cortial, „Dimensional reduction of nonlinear finite element dynamic models with finite rotations and energy-based mesh sampling and weighting for computational efficiency," *International Journal for Numerical Methods in Engineering,* vol. 98, pp. 625-662, 2014.

[68] C. Farhat, T. Chapman and P. Avery, „Structure-preserving, stability, and accuracy properties of the energy-conserving sampling and weighting method for the hyper reduction of nonlinear finite element dynamic models," *International Journal for Numerical Methods in Engineering,* vol. 102, pp. 1077-1110, 2015.

[69] P. Tiso and D. J. Rixen, „Discrete empirical interpolation method for finite element structural dynamics," in *Topics in Nonlinear Dynamics, Volume 1*, Springer, 2013, pp. 203-212.

[70] F. Fritzen, B. Haasdonk, D. Ryckelynck and S. Schöps, „An algorithmic comparison of the Hyper-Reduction and the Discrete Empirical Interpolation Method for a nonlinear thermal problem," *Mathematical and Computational Applications,* vol. 23, p. 8, 2018.

[71] J. B. Rutzmoser and D. J. Rixen, „A lean and efficient snapshot generation technique for the Hyper-Reduction of nonlinear structural dynamics," *Computer Methods in Applied Mechanics and Engineering,* vol. 325, pp. 330-349, 2017.

[72] K. C. Park and P. G. Underwood, „A variable-step central difference method for structural dynamics analysis - part 1. Theoretical aspects," *Computer methods in applied mechanics and engineering,* vol. 22, pp. 241-258, 1980.

[73] M. Géradin and D. J. Rixen, Mechanical vibrations: theory and application to structural dynamics, John Wiley & Sons, 2014.

[74] A. J. M. Spencer, Continuum mechanics, Courier Corporation, 2004.

[75] T. J. R. Hughes, The finite element method: linear static and dynamic finite element analysis, Courier Corporation, 2012.

[76] K.-J. Bathe, Finite element procedures, Klaus-Jurgen Bathe, 2006.

[77] P. Benner, S. Gugercin and K. Willcox, „A survey of model reduction methods for parametric systems," 2013.

[78] M. Geuss, H. Panzer and B. Lohmann, „On parametric model order reduction by matrix interpolation," in *Proceedings of the European Control Conference*, 2013.

[79] S. Boyd and L. Vandenberghe, Convex optimization, Cambridge university press, 2004.

[80] D. Amsallem, R. Tezaur and C. Farhat, „Real-time solution of computational problems using databases of parametric linear reduced-order models with arbitrary underlying meshes," *arXiv preprint arXiv:1506.07153,* 2015.

[81] D. Amsallem, „Interpolation on manifolds of CFD-based fluid and finite element-based structural reduced-order models for on-line aeroelastic predictions," 2010.

[82] J. Hammersley, Monte carlo methods, Springer Science & Business Media, 2013.

[83] M. D. McKay, R. J. Beckman and W. J. Conover, „A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics,* vol. 42, pp. 55-61, 2000.

[84] T. Bui-Thanh, K. Willcox and O. Ghattas, „Model reduction for large-scale systems with high-dimensional parametric input space," *SIAM Journal on Scientific Computing,* vol. 30, pp. 3270-3288, 2008.

[85] Z. Bai and Y. Su, „Dimension reduction of large-scale second-order dynamical systems via a second-order Arnoldi method," *SIAM Journal on Scientific Computing,* vol. 26, pp. 1692-1709, 2005.

[86] S. Gugercin and A. C. Antoulas, „A survey of model reduction by balanced truncation and some new results," *International Journal of Control,* vol. 77, pp. 748-766, 2004.

[87] T. Reis and T. Stykel, „Balanced truncation model reduction of second-order systems," *Mathematical and Computer Modelling of Dynamical Systems,* vol. 14, pp. 391-406, 2008.

[88] S. B. Salimbahrami, „Structure preserving order reduction of large scale second order models," 2005.

[89] M. J. Rewienski, „A trajectory piecewise-linear approach to model order reduction of nonlinear dynamical systems," 2003.

[90] M. Rewienski and J. White, „Model order reduction for nonlinear dynamical systems based on trajectory piecewise-linear approximations," *Linear algebra and its applications,* vol. 415, pp. 426-454, 2006.

[91] L. Wu and P. Tiso, „Nonlinear model order reduction for flexible multibody dynamics: a modal derivatives approach," *Multibody System Dynamics,* vol. 36, pp. 405-425, 2016.

[92] C. Nowakowski, J. Fehr, M. Fischer and P. Eberhard, „Model order reduction in elastic multibody systems using the floating frame of reference formulation," *IFAC Proceedings Volumes,* vol. 45, pp. 40-48, 2012.

[93] P. Koutsovasilis and M. Beitelschmidt, „Comparison of model reduction techniques for large mechanical systems," *Multibody System Dynamics,* vol. 20, pp. 111-128, 2008.

[94] W. C. Hurty, Dynamic analysis of structural systems by component mode synthesis, JPL, Jet Propulsion Laboratory, California Institute of Technology, 1964.

[95] M. C. C. Bampton and R. R. CRAIG, „Coupling of substructures for dynamic analyses.," *Aiaa Journal,* vol. 6, pp. 1313-1319, 1968.

[96] R. H. MacNeal, „A hybrid method of component mode synthesis," *Computers & Structures,* vol. 1, pp. 581-601, 1971.

[97] S. Rubin, „Improved component-mode representation for structural dynamic analysis," *AIAA journal,* vol. 13, pp. 995-1006, 1975.

[98] R. R. a. C. C.-j. Craig Jr, „On the use of attachment modes in substructure coupling for dynamic analysis," *transformation,* vol. 10, nr. 4, p. 3, 1977.

[99] H. De Gersem, „Interval component mode synthesis procedure for the dynamic analysis of large FE models," 2008.

[100] S. N. Voormeeren, Dynamic substructuring methodologies for integrated dynamic analysis of wind turbines, TU Delft, Delft University of Technology, 2012.

[101] R. M. Hintz, „Analytical methods in component modal synthesis," *AIAA Journal,* vol. 13, pp. 1007-1016, 1975.

[102] G. Berkooz, P. Holmes and J. L. Lumley, „The proper orthogonal decomposition in the analysis of turbulent flows," *Annual review of fluid mechanics,* vol. 25, pp. 539-575, 1993.

[103] R. A. Johnson, D. W. Wichern and others, Applied multivariate statistical analysis, vol. 5, Prentice hall Upper Saddle River, NJ, 2002.

[104] K. Carlberg, C. Bou-Mosleh and C. Farhat, „Efficient non-linear model reduction via a least-squares Petrov--Galerkin projection and compressive tensor approximations," *International Journal for Numerical Methods in Engineering,* vol. 86, pp. 155-181, 2011.

[105] K. Carlberg, C. Farhat, J. Cortial and D. Amsallem, „The GNAT method for nonlinear model reduction: effective implementation and application to computational fluid dynamics and turbulent flows," *Journal of Computational Physics,* vol. 242, pp. 623-647, 2013.

[106] K. Willcox, „Unsteady flow sensing and estimation via the gappy proper orthogonal decomposition," *Computers & fluids,* vol. 35, pp. 208-226, 2006.

[107] M. Barrault, Y. Maday, N. C. Nguyen and A. T. Patera, „An 'empirical interpolation' method: application to efficient reduced-basis discretization of partial differential equations," *Comptes Rendus Mathematique,* vol. 339, pp. 667-672, 2004.

[108] S. Jain, „Model Order Reduction for Non-Linear Structural Dynamics," 2015.

[109] C. L. Lawson and R. J. Hanson, Solving least squares problems, vol. 15, Siam, 1995.

[110] Z.-H. Zhong, Finite element procedures for contact-impact problems, Oxford university press, 1993.

[111] T. A. Laursen, *Computational Contact and Impact Mechanics: Fundamentals of Modelling Interfacial Phenomena in Nonlinear Finite Element Analysis,* Springer, Berlin, 2002.

[112] G. Zavarise and L. De Lorenzis, „A modified node-to-segment algorithm passing the contact patch test," *International journal for numerical methods in engineering,* vol. 79, p. 379, 2009.

[113] M. A. Puso and T. A. Laursen, „A mortar segment-to-segment frictional contact method for large deformations," *Computer methods in applied mechanics and engineering,* vol. 193, pp. 4891-4913, 2004.

[114] P. Papadopoulos, R. E. Jones and J. M. Solberg, „A novel finite element formulation for frictionless contact problems," *International Journal for Numerical Methods in Engineering,* vol. 38, pp. 2603-2617, 1995.

[115] R. A. Sauer and L. De Lorenzis, „An unbiased computational contact formulation for 3D friction," *International Journal for Numerical Methods in Engineering,* vol. 101, pp. 251-280, 2015.

[116] J. C. Simo, P. Wriggers and R. L. Taylor, „A perturbed Lagrangian formulation for the finite element solution of contact problems," *Computer methods in applied mechanics and engineering,* vol. 50, pp. 163-180, 1985.

[117] P. Papadopoulos and R. L. Taylor, „A mixed formulation for the finite element solution of contact problems," *Computer Methods in Applied Mechanics and Engineering,* vol. 94, pp. 373-389, 1992.

[118] G. Zavarise and P. Wriggers, „A segment-to-segment contact strategy," *Mathematical and Computer Modelling,* vol. 28, pp. 497-515, 1998.

[119] M. A. Puso and T. A. Laursen, „A mortar segment-to-segment contact method for large deformation solid mechanics," *Computer methods in applied mechanics and engineering,* vol. 193, pp. 601-629, 2004.

[120] F. B. Belgacem, P. Hild and P. Laborde, „The mortar finite element method for contact problems," *Mathematical and Computer Modelling,* vol. 28, pp. 263-271, 1998.

[121] M. Reymond, MSC/NASTRAN quick reference guide: version 68, MacNeal-Schwendler, 1994.

[122] U. M. Ascher and L. R. Petzold, Computer methods for ordinary differential equations and differential-algebraic equations, vol. 61, Siam, 1998.

[123] E. Hairer, S. P. Norsett and G. Wanner, Solving ordinary differential equations I. Nonstiff problems, Springer, 1993.

[124] E. Eich-Soellner and C. Führer, Numerical methods in multibody dynamics, vol. 45, Springer, 1998.

[125] K.-J. Bathe, „Conserving energy and momentum in nonlinear dynamics: a simple implicit time integration scheme," *Computers & structures,* vol. 85, pp. 437-445, 2007.

[126] W. Kutta, „Beitrag zur näherungweisen Integration totaler Differentialgleichungen," *Z. Math. Phys.,* vol. 46, pp. 435-453, 1901.

[127] L. F. Shampine, „Linear conservation laws for ODEs," *Computers & Mathematics with Applications,* vol. 35, pp. 45-53, 1998.

[128] L. F. Shampine, „Conservation laws and the numerical solution of ODEs, II," *Computers & Mathematics with Applications,* vol. 38, pp. 61-72, 1999.

[129] R. D. Skeel, „Construction of variable-stepsize multistep formulas," *MATHEMATICS of computation,* vol. 47, pp. 503-510, 1986.

[130] J. Hallquist, „LS-DYNA Theoretical Manual, Livermore Software Technology Corporation," *Livermore, Ca,* 1998.

[131] R. D. Cook and others, Concepts and applications of finite element analysis, John Wiley & Sons, 2007.

[132] L. M. Milne-Thomson, The calculus of finite differences, American Mathematical Soc., 2000.

[133] A. C. Hindmarsh, P. N. Brown, K. E. Grant, S. L. Lee, R. Serban, D. E. Shumaker and C. S. Woodward, „SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers," *ACM Transactions on Mathematical Software (TOMS),* vol. 31, pp. 363-396, 2005.

[134] C. Kane, „Variational integrators and the Newmark algorithm for conservative and dissipative mechanical systems," 1999.

[135] H. M. Hilber, T. J. R. Hughes and R. L. Taylor, „Improved numerical dissipation for time integration algorithms in structural dynamics," *Earthquake Engineering & Structural Dynamics,* vol. 5, pp. 283-292, 1977.

[136] M. Arnold and O. Brüls, „Convergence of the generalized-α scheme for constrained mechanical systems," *Multibody System Dynamics,* vol. 18, pp. 185-202, 2007.

[137] K. E. Brenan, S. L. Campbell and L. R. Petzold, Numerical solution of initial-value problems in differential-algebraic equations, vol. 14, Siam, 1996.

[138] L. Petzold, „Differential/algebraic equations are not ODE's," *SIAM Journal on Scientific and Statistical Computing,* vol. 3, pp. 367-384, 1982.

[139] M. Arnold, „A perturbation analysis for the dynamical simulation of mechanical multibody systems," *Applied numerical mathematics,* vol. 18, pp. 37-56, 1995.

[140] G. Wanner and E. Hairer, Solving ordinary differential equations II, vol. 1, Springer-Verlag, Berlin, 1991.

[141] M. Arnold, „Numerical methods for simulation in applied dynamics," in *Simulation Techniques for Applied Dynamics*, Springer, 2008, pp. 191-246.

[142] J. Baumgarte, „Stabilization of constraints and integrals of motion in dynamical systems," *Computer methods in applied mechanics and engineering,* vol. 1, pp. 1-16, 1972.

[143] R. Von Schwerin, Multibody system simulation: numerical methods, algorithms, and software, vol. 7, Springer Science & Business Media, 2012.

[144] C. W. Gear, B. Leimkuhler and G. K. Gupta, „Automatic integration of Euler-Lagrange equations with constraints," *Journal of Computational and Applied Mathematics,* vol. 12, pp. 77-90, 1985.

[145] N. J. Carpenter, R. L. Taylor and M. G. Katona, „Lagrange constraints for transient finite element surface contact," *International journal for numerical methods in engineering,* vol. 32, pp. 103-128, 1991.

[146] V. Brasey and E. Hairer, „Half-explicit Runge-Kutta methods for differential-algebraic systems of index 2," *SIAM Journal on Numerical Analysis,* vol. 30, pp. 538-552, 1993.

[147] M. Arnold, K. Strehmel and R. Weiner, „Half-explicit Runge-Kutta methods for semi-explicit differential-algebraic equations of index 1," *Numerische Mathematik,* vol. 64, pp. 409-431, 1993.

[148] A. Ostermann, „A class of half-explicit Runge-Kutta methods for differential-algebraic systems of index 3," *Applied numerical mathematics,* vol. 13, pp. 165-179, 1993.

[149] O. Brüls, V. Acary and A. Cardona, „Simultaneous enforcement of constraints at position and velocity levels in the nonsmooth generalized-α scheme," *Computer Methods in Applied Mechanics and Engineering,* vol. 281, pp. 131-161, 2014.

[150] C. Lunk and B. Simeon, „Solving constrained mechanical systems by the family of Newmark and α-methods," *ZAMM-Journal of Applied Mathematics and Mechanics/Zeitschrift für Angewandte Mathematik und Mechanik,* vol. 86, pp. 772-784, 2006.

[151] L. Jay and D. Negrut, „Extensions of the HHT-method to differential-algebraic equations in mechanics," *Electron. Trans. Numer. Anal,* vol. 26, pp. 190-208, 2007.

[152] A. Cardona and M. Geradin, „Time integration of the equations of motion in mechanism analysis," *Computers & structures,* vol. 33, pp. 801-820, 1989.

[153] D. Negrut, R. Rampalli, G. Ottarsson and A. Sajdak, „On the use of the HHT method in the context of index 3 differential algebraic equations of multibody dynamics," in *ASME 2005 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2005.

[154] C. L. Bottasso, D. Dopico and L. Trainelli, „On the optimal scaling of index three DAEs in multibody dynamics," *Multibody System Dynamics,* vol. 19, pp. 3-20, 2008.

[155] M. Haddouni, V. Acary and J.-D. Beley, „Comparison of index-3, index-2 and index-1 DAE solvers for nonsmooth multibody systems with unilateral and bilateral constraints," in *Multibody Dynamics 2013*, 2013.

[156] F. L. Litvin and A. Fuentes, Gear geometry and applied theory, Cambridge University Press, 2004.

[157] J. R. Colbourne, The geometry of involute gears, Springer Science & Business Media, 2012.

[158] S. P. Radzevich, Theory of gearing: kinematics, geometry, and synthesis, CRC Press, 2012.

[159] H. Link, W. LaCava, J. Dam, B. McNiff, S. Sheng, R. Wallen, M. McDade, S. Lambert, S. Butterfield and F. Oyague, „Gearbox reliability collaborative project report: findings from phase 1 and phase 2 testing," 2011.

[160] S. Kurokawa, Y. Ariura and A. Ogata, „High precision measurement and analysis of transmission errors of gears under load," in *World congress on gearing and power transmission*, 1999.

[161] N. Cappellini, T. Tamarozzi, B. Blockmans, J. Fiszer, F. Cosco and W. Desmet, „Semi-analytic contact technique in a non-linear parametric model order reduction method for gear simulations," *Meccanica,* vol. 53, pp. 49-75, 2018.

[162] Hibbitt, Karlsson and Sorensen, ABAQUS/Explicit: User's Manual, vol. 1, Hibbitt, Karlsson and Sorenson Incorporated, 2001.

[163] D. B. Welbourn, „Fundamental knowledge of gear noise: a survey," 1979.

[164] C. Weber and K. Banaschek, „Formänderung und Profilrücknahme bei Gerad-und Schrägverzahnten Antriebstechnik," *Vieweg, Braunschweig,* vol. 11, 1953.

[165] M. J. Puttock and E. G. Thwaite, Elastic compression of spheres and cylinders at point and line contact, Commonwealth Scientific and Industrial Research Organization Melbourne, Australia, 1969.

[166] T. A. Harris and M. N. Kotzalas, Advanced concepts of bearing technology: rolling bearing analysis, CRC Press, 2006.

[167] C. G. Broyden, „A class of methods for solving nonlinear simultaneous equations," *Mathematics of computation,* vol. 19, pp. 577-593, 1965.

[168] J. Sherman and W. J. Morrison, „Adjustment of an inverse matrix corresponding to a change in one element of a given matrix," *The Annals of Mathematical Statistics,* vol. 21, pp. 124-127, 1950.

[169] F. W. Williams, „An algorithm for exact eigenvalue calculations for rotationally periodic structures," *International journal for numerical methods in engineering,* vol. 23, pp. 609-622, 1986.

[170] I. Amidror, „Scattered data interpolation methods for electronic imaging systems: a survey," *Journal of electronic imaging,* vol. 11, pp. 157-177, 2002.

[171] J. E. Dennis Jr and R. B. Schnabel, Numerical methods for unconstrained optimization and nonlinear equations, vol. 16, Siam, 1996.

[172] R. Hunger, Floating point operations in matrix-vector calculus, Munich University of Technology, Inst. for Circuit Theory and Signal Processing Munich, 2005.

[173] B. Blockmans, T. Tamarozzi, F. Naets and W. Desmet, „A nonlinear parametric model reduction method for efficient gear contact simulations," *International Journal for Numerical Methods in Engineering,* vol. 102, pp. 1162-1191, 2015.

[174] G. H. Golub and C. F. Van Loan, Matrix computations, vol. 3, JHU Press, 2012.

[175] B. Blockmans, „Equations of motion of flexible bodies with configuration-dependent reduction basis in planar rigid-body motion," 2013.

[176] B. Blockmans, J. Helsen, F. Vanhollebeke and W. Desmet, „Dynamic response of a multi-megawatt wind turbine drivetrain under voltage dips using a coupled flexible multibody approach," in *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, 2013.

[177] M. Geradin, „A classification and discussion of integration operators for transient structural response," in *American Institute of Aeronautics and Astronautics, Aerospace Sciences Meeting, 12 th, Washington, D. C*, 1974.

[178] O. A. Bauchau and C. L. Bottasso, „On the design of energy preserving and decaying schemes for flexible, nonlinear multi-body systems," *Computer Methods in Applied Mechanics and Engineering,* vol. 169, pp. 61-80, 1999.

[179] T. Belytschko and T. J. R. Hughes, „Computational methods for transient analysis," *Amsterdam, North-Holland(Computational Methods in Mechanics.,* vol. 1, 1983.

[180] K. Sherif, W. Witteveen and K. Maryhofer, „Quasi-static consideration of high-frequency modes for more efficient flexible multibody simulations," *Acta Mechanica,* vol. 223, pp. 1285-1305, 2012.

[181] M. L. Ghrist, B. Fornberg and J. A. Reeger, „Stability ordinates of Adams predictor-corrector methods," *BIT Numerical Mathematics,* vol. 55, pp. 733-750, 2015.

[182] A. B. Chaudhary and K.-J. Bathe, „A solution method for static and dynamic analysis of three-dimensional contact problems with friction," *Computers & Structures,* vol. 24, pp. 855-873, 1986.

[183] R. A. Wehage and E. Haug, „Generalized coordinate partitioning for dimension reduction in analysis of constrained dynamic systems," *Journal of mechanical design,* vol. 104, pp. 247-255, 1982.

[184] P. E. Nikravesh, Computer-aided analysis of mechanical systems, Prentice-Hall, Inc., 1988.

[185] F. P. Bowden and D. Tabor, The friction and lubrication of solids, vol. 1, Oxford university press, 2001.

[186] V. Popov, Contact mechanics and friction: physical principles and applications, Springer Science & Business Media, 2010.

[187] T. A. Laursen and J. C. Simo, „A continuum-based finite element formulation for the implicit solution of multibody, large deformation-frictional contact problems," *International Journal for numerical methods in engineering,* vol. 36, pp. 3451-3485, 1993.

[188] I. M. Hutchings and P. Shipway, Tribology: friction and wear of engineering materials, CRC press Boca Raton, FL, 1992.

[189] E. Fehlberg, „Low-order classical Runge-Kutta formulas with stepsize control and their application to some heat transfer problems," 1969.

[190] P. Bogacki and L. F. Shampine, „A 3 (2) pair of Runge-Kutta formulas," *Applied Mathematics Letters,* vol. 2, pp. 321-325, 1989.

[191] L. F. Shampine and M. W. Reichelt, „The matlab ode suite," *SIAM journal on scientific computing,* vol. 18, pp. 1-22, 1997.

[192] J. Chung and G. M. Hulbert, „A time integration algorithm for structural dynamics with improved numerical dissipation: the generalized-α method," *Journal of applied mechanics,* vol. 60, pp. 371-375, 1993.

[193] S. Erlicher, L. Bonaventura and O. S. Bursi, „The analysis of the generalized-α method for non-linear dynamic problems," *Computational Mechanics,* vol. 28, pp. 83-104, 2002.

[194] I. S. O. Standard, *ISO 53:1998, Cylindrical gears for general and heavy engineering _ Standard basic rack tooth profile.*

[195] A. E. H. Love, A treatise on the mathematical theory of elasticity, vol. 1, Cambridge University Press, 2013.

[196] Hibbitt, Karlsson and Sorensen, ABAQUS: Theory Manual, Hibbitt, Karlsson & Sorensen, 1997.

[197] V. A. Yastrebov, Numerical methods in contact mechanics, John Wiley & Sons, 2013.

[198] P. Wriggers, Nonlinear finite element methods, Springer Science & Business Media, 2008.

[199] T. M. Wasfy and A. K. Noor, „Computational strategies for flexible multibody systems," *Applied Mechanics Reviews,* vol. 56, pp. 553-613, 2003.

[200] S. P. Wang and E. Nakamachi, „The inside-outside contact search algorithm for finite element analysis," *International journal for numerical methods in engineering,* vol. 40, pp. 3665-3685, 1997.

[201] I. S. O. Standard, „53: 1998," *Cylindrical gears for general and heavy engineering, Standard basic rack tooth profile,* 1998.

[202] B. Simeon, „MBSPACK-Numerical integration software for constrained mechanical motion," *Surveys on Mathematics for Industry,* vol. 5, pp. 169-201, 1995.

[203] B. Simeon, „Computational flexible multibody dynamics," in *A Differential-Algebraic Approach. Differential-Algebraic Equations Forum. Springer, Heidelberg*, 2013.

[204] A. Shabana, Dynamics of multibody systems, New, York: John Wiley & Sons, Inc., 1989.

[205] S. S. Rao, The finite element method in engineering, Elsevier, 2010.

[206] A. Popp, M. Gitterle, M. W. Gee and W. A. Wall, „A dual mortar approach for 3D finite deformation contact with consistent linearization," *International Journal for Numerical Methods in Engineering,* vol. 83, pp. 1428-1465, 2010.

[207] K. C. Park and P. G. Underwood, „A variable-step central difference method for structural dynamics analysis_part 1. Theoretical aspects," *Computer methods in applied mechanics and engineering,* vol. 22, pp. 241-258, 1980.

[208] P. C. Kohnke, ANSYS Theory Reference: Release 5.5, ANSYS, Incorporated, 1998.

[209] C. T. Kelley, Solving nonlinear equations with Newton's method, vol. 1, Siam, 2003.

[210] I. Jolliffe, Principal component analysis, Wiley Online Library, 2002.

[211] P. Feldmann, „Model order reduction techniques for linear systems with large number of terminals," in *Proceedings of the Design, Automation and Test in Europe Conference and Exhibition*, 2004.

[212] J. Fehr and P. Eberhard, „Simulation process of flexible multibody systems with non-modal model order reduction techniques," *Multibody System Dynamics,* vol. 25, pp. 313-334, 2011.

[213] B. F. Feeny and R. Kappagantu, „On the physical interpretation of proper orthogonal modes in vibrations," *Journal of sound and vibration,* vol. 211, pp. 607-616, 1998.

[214] J. R. Dormand and P. J. Prince, „A family of embedded Runge-Kutta formulae," *Journal of computational and applied mathematics,* vol. 6, pp. 19-26, 1980.

[215] J. M. De Mul, J. M. Vree and D. A. Maas, „Equilibrium and associated load distribution in ball and roller bearings loaded in five degrees of freedom while neglecting friction_Part II: application to roller bearings and experimental verification," *Journal of tribology,* vol. 111, pp. 149-155, 1989.

[216] J. A. Cottrell, T. J. R. Hughes and Y. Bazilevs, Isogeometric analysis: toward integration of CAD and FEA, John Wiley & Sons, 2009.

[217] M. Arnold, „Half-explicit Runge-Kutta methods with explicit stages for differential-algebraic systems of index 2," *BIT Numerical Mathematics,* vol. 38, pp. 415-438, 1998.

[218] P. Bergan and E. Mollestad, „An automatic time-stepping algorithm for dynamic problems," *Computer Methods in Applied Mechanics and Engineering,* vol. 49, pp. 299-318, 1985.

[219] I. S. O. STANDARD, „Calculation of load capacity of spur and helical gears_," *ISO,* vol. 6336, p. 1996, 2006.

[220] F. Dewalque, P. Rochus and O. Brüls, „Importance of structural damping in the dynamic analysis of compliant deployable structures," *Acta Astronautica,* vol. 111, pp. 323-333, 2015.

List of publications

Articles in international journals

B. Blockmans, T. Tamarozzi, F. Naets and W. Desmet, „A nonlinear parametric model reduction method for efficient gear contact simulations," *International Journal for Numerical Methods in Engineering,* vol. 102, pp. 1162-1191, 2015.

J. Fiszer, T. Tamarozzi, B. Blockmans and W. Desmet, „A time-dependent parametric model order reduction technique for modelling indirect bearing force measurements," *Mechanism and Machine Theory,* vol. 83, pp. 152-174, 2015.

N. Cappellini, T. Tamarozzi, B. Blockmans, J. Fiszer, F. Cosco and W. Desmet, „Semi-analytic contact technique in a non-linear parametric model order reduction method for gear simulations," *Meccanica,* vol. 53, pp. 49-75, 2018.

Articles in international conference proceedings

B. Blockmans, J. Helsen, F. Vanhollebeke and W. Desmet, „Dynamic response of a multi-megawatt wind turbine drivetrain under voltage dips using a coupled flexible multibody approach," in *ASME 2013 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*. Portland, Oregon, USA, August 4-7, 2013.

B. Blockmans, R. Adduci, J. Fiszer, N. Cappelini, F. Cosco, W. Desmet, „Finite element based simulation of planetary gearboxes using model-order reduction," in *Conference proceedings of the International Gear Conference 2018*, pp. 1024-1033, Lyon, France, August 27-29, 2018.

T. Tamarozzi, B. Blockmans, and W. Desmet, „ Dynamic stress analysis of the high-speed stage of a wind turbine gearbox using a coupled flexible multibody approach," in *CWD Conference for wind power drives*, Aachen, Germany, March 3-4, 2015.

T. Tamarozzi, B. Blockmans, and W. Desmet, „Efficient transient analysis of the high-speed stage of a wind turbine gearbox by advanced model reduction techniques and memory-effective discretization," in *ASME 2015 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference:* pp. V010T11A022-V010T11A022). American Society of Mechanical Engineers. Boston, Massachusetss, USA, August 2-5, 2015.

F. Cosco, B. Blockmans, J. Fiszer, T. Tamarozzi, and W. Desmet, „Efficient and Accurate Formulation of FE-based Contact Mechanics Problems," in *proceedings of isma2016 international conference on noise and vibration engineering and usd2016 international conference on uncertainty in structural dynamics 2016*, pp. 1347-356, Leuven, Belgium, September 2016.

N. Cappellini, B. Blockmans, J. Fiszer, T. Tamarozzi, F. Cosco, and W. Desmet, „Reduced-Order Modelling of Multibody Contact Problems: A Novel Semi-Analytic Method," In *ASME 2017 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, pp. V006T10A028-V006T10A028. American Society of Mechanical Engineers. Cleveland, Ohio, USA, August 6-9, 2017.

Abstracts in international conference proceedings

B. Blockmans, T. Tamarozzi, F. Naets, W. Desmet, „Interpolation strategies for non-linear parametric model-order reduction in gear contact simulation," at *the 5th European conference on computational contact mechanics*. Barcelona, Spain, July 20-25, 2014.

B. Blockmans, T. Tamarozzi, W. Desmet, „Efficient solution of dynamic gear contact problems using a novel paramteric model-order reduction technique," at *The 4th international conference on computational contact mechanics*. Hannover, Germany, May 27-29, 2015.

B. Blockmans, T. Tamarozzi, N. Cappellini, F. Cosco, W. Desmet, „Efficient construction and integration of a parametric reduced-order model for solving misaligned gear contact problems," at *The 4th joint international conference on multibody system dynamics*. Montreal, Canada, May 29-June 2, 2016.

J. Fiszer, T. Tamarozzi, B. Blockmans, B. Verrelst, W. Desmet, „An efficient parametric model-order reduction technique applied to bearing force estimation," at *the Joint international conference on multibody system dynamics*. Korea, June 30-July 3, 2014.

Book chapters

R. Cerdá, B. Blockmans, J. Fiszer, T. Tamarozzi, B. Pluymers, and W. Desmet, „Dynamic Behavior of Bearings on Offshore Wind Turbine Gearboxes," In *MARE-WINT: New Materials and Reliability in Offshore Wind Turbine Technology*, pp. 123-46. Springer, 2016.

N. Cappellini, F. Cosco, B. Blockmans, J. Croes, T. Tamarozzi, and W. Desmet, „Towards Efficient System-level Simulation and Design of Modern Mechanical Transmissions," In *EMVeM Energy Efficiency Management for Vehicles and Machines*, pp. 23-36. KU Leuven, 2016.