Extending M3D-C^1 to stellarator geometry: preliminary results

Yao Zhou

Princeton Plasma Physics Laboratory

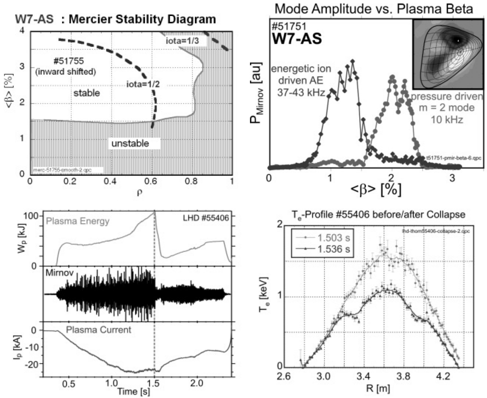
Acknowledgements: N. M. Ferraro, S. C. Jardin, H. R. Strauss.

PPPL Theory Seminar, Nov 19, 2020

Outline

Stellarators exhibit exotic nonlinear MHD physics

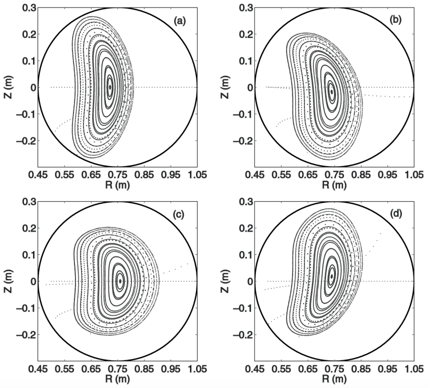
- Designs base on linear MHD stability:
  - mostly for pressure-driven modes;
  - impose limits on achievable plasma β.
- Experiments show ‘self-stabilization’[1]:
  - modes often saturate on harmless levels;
  - not always – sometimes large collapses.
- Understanding nonlinear stability helps:
  - expand operation windows (present);
  - improve designs and lower costs (future).
- Need nonlinear, dynamical simulations.

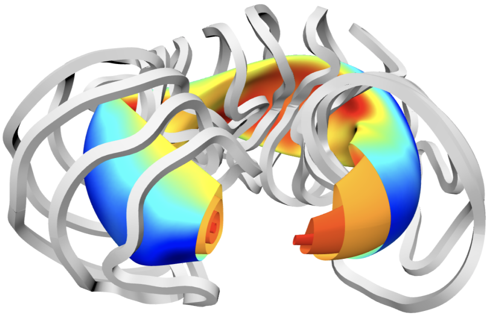


[1]A. Weller et al, Fusion Sci. Technol. 50, 158 (2006).

Most initial-value MHD codes assume axisymmetric geometry

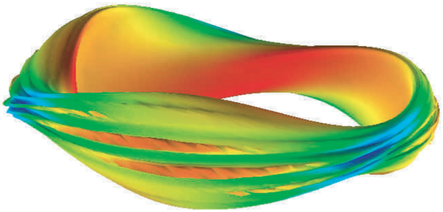
- Same mesh at every toroidal angle φ; can simulate simple stellarators[2] (left: CTH).
- Non-axisymmetric domain/mesh required for more complex stellarators (right: NCSX).
- NIMROD, JOREK both currently being extended to stellarator geometry.

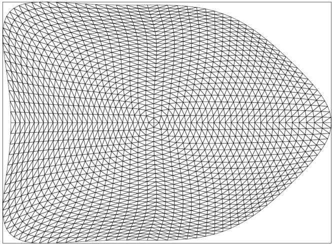




[2] M. G. Schlutt et al, Nucl. Fusion 52, 103023 (2012).

M3D has previously been adapted to stellarator geometry

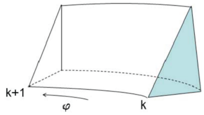


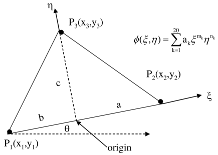


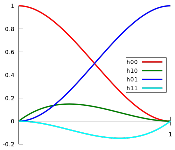
- VMEC outputs $R(s, \theta, \zeta)$, $Z(s, \theta, \zeta)$ used for meshing[3].
  - Different but logically connected meshes on different poloidal planes.
  - Involved minimal changes to the code.
- Our approach to M3D-C^1 will be largely similar.
  - More complications due to more advanced numerical methods.
  - Different philosophy from building a new code from scratch.

[3] H. R. Strauss et al, Nucl. Fusion 44, 1008 (2004).

M3D-C^1 is a different code from M3D

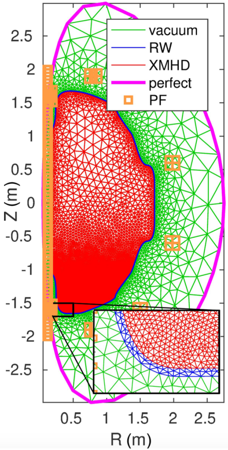






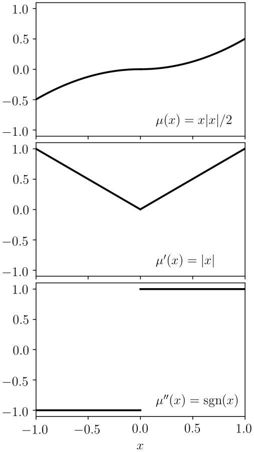
- 'Extruded' high-order elements with C^1 continuity[4]:
  - basis functions $\mu_{ij}(R,Z,\varphi)=\nu_i(R,Z)h_j(\varphi)$;
  - reduced quintics $\nu_i(R,Z)$, unstructured mesh;
  - Hermite cubics $h_j(\varphi)$, packing available.
- Fully implicit and semi-implicit time advance:
  - suitable for long-time simulations (up to $10^6\tau_A$).
- Extended MHD and more physics capabilities:
  - hot particles, radiation, pellets, resistive wall, etc.

Mesh for NSTX VDE[5]



[4] S. C. Jardin et al, Comput. Sci. Disc. 5, 014002 (2012).
[5] D. Pfefferlé et al, Phys. Plasmas 25, 056106 (2018).

C^1-continuous elements allow for higher-order derivatives

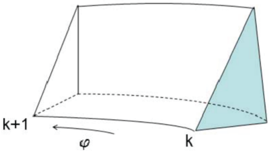


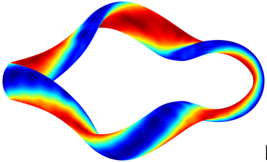
- Galerkin method: $\hat{\mathcal{L}}u = v$, $u = \sum_j u_j \mu_j$,

$$\sum_j (\mu_i, \hat{\mathcal{L}}\mu_j) u_j = (\mu_i, v). \tag{1}$$

  u_j: degrees of freedom (DoFs); $(u, v) \equiv \int uv \,\mathrm{d}V$.
- For C^{m-1} elements, $\hat{\mathcal{L}}$ can be m^{th} order at most;
  - with integration by parts, up to $2m^{\text{th}}$ order.
- M3D-C^1 elements allow for μ_{RR}, μ_{RZ}, $\mu_{\varphi\varphi}$, etc.,
  - and even $\mu_{RR\varphi\varphi}$ for being 'extruded' (bonus);
  - but not μ_{RRZ} or μ_{RRZZ} (not needed anyway).
- Vector fields represented with potentials $(U, \omega, \chi, \psi, f)$:
  - $\mathbf{V} = R^2 \nabla U \times \nabla\varphi + \omega R^2 \nabla\varphi + R^{-2}\nabla_\perp \chi$;
  - $\mathbf{A} = R^2 \nabla\varphi \times \nabla f + \psi \nabla\varphi - F_0 \ln R\, \hat{Z}$.

Stellarator extension should induce minimal changes to M3D-C^1





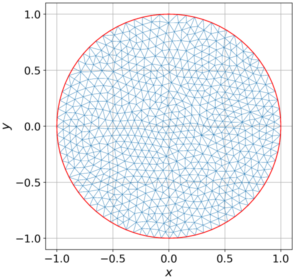
- One hand: extruded C^1 elements.
  - Assumes axisymmetric mesh.
- Use genuinely 3D C^1 elements?
  - No good candidates known.
  - Changes foundation of the code.

- The other hand: physics equations.
  - In (R, Z, φ): non-axisymmetric domain.
- Recast equations in different coordinates?
  - Need to rewrite every term (a lot).
  - Much more complicated (metric factors).

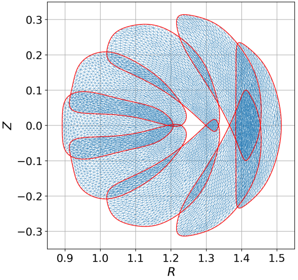
- Solution: an interface between axisymmetry and non-axisymmetry.
  - On the level of basis functions, not equations.
  - Need to respect C^1 property for the namesake.

Coordinate mapping connects axisymmetry and non-axisymmetry

- Logical coordinates (x, y, ζ):
  - axisymmetric domain/mesh;
  - use existing elements in $\mu(x, y, \zeta)$.



- Physical coordinates (R, Z, φ):
  - non-axisymmetric domain/mesh;
  - use existing equations in μ_R, μ_Z, etc.



- Coordinate mapping: $R = R(x, y, \zeta)$, $Z = Z(x, y, \zeta)$, $\varphi = \zeta$,
  - e.g., from VMEC output: $x = \sqrt{s}\cos\theta$, $y = \sqrt{s}\sin\theta$;
  - other options possible, physical meaning unnecessary.
- Common approach for (structured) finite elements.

Physical derivatives are calculated using chain rule

- For example, first derivatives (of basis function μ) are

$$\begin{pmatrix} \mu_R \\ \mu_Z \\ \mu_\varphi \end{pmatrix} = J \begin{pmatrix} \mu_x \\ \mu_y \\ \mu_\zeta \end{pmatrix}, \; J = \begin{pmatrix} R_x & Z_x & 0 \\ R_y & Z_y & 0 \\ R_\zeta & Z_\zeta & 1 \end{pmatrix}^{-1} = \frac{1}{D} \begin{pmatrix} Z_y & -Z_x & 0 \\ -R_y & R_x & 0 \\ A & B & D \end{pmatrix}, \qquad (2)$$

  where $A = R_y Z_\zeta - R_\zeta Z_y$, $B = R_\zeta Z_x - R_x Z_\zeta$, $D = R_x Z_y - R_y Z_x$.
- $(\mu_R, \mu_Z, \mu_\varphi)$ are continuous if R and Z have C^1 continuity;
  - can be guaranteed by having $R = \sum_j R_j \mu_j(x, y, \zeta)$, $Z = \sum_j Z_j \mu_j(x, y, \zeta)$.
- Volume integral: $\int u \, \mathrm{d}V = \int u \, R \mathrm{d}R \mathrm{d}Z \mathrm{d}\varphi = \int u \, R D \mathrm{d}x \mathrm{d}y \mathrm{d}\zeta$.
- DoFs u_j involve derivatives here and need to be transformed as well:
  - boundary conditions are imposed on DoFs.
- (Metric tensors, etc. can be calculated but not needed.)

High-order mixed derivatives are no longer allowed

- Second derivatives still exist in a weak sense, for example,

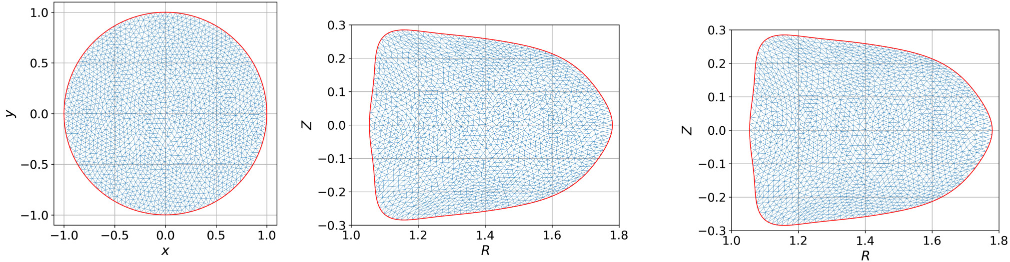
$$\mu_{R\varphi} = [(Z_yB - Z_xA)/D^2]\mu_{xy} + (Z_yA/D^2)\mu_{xx} - (Z_xB/D^2)\mu_{yy}$$
$$+[(Z_yA_x - Z_xA_y)/D^2 + GA/D^3]\mu_x + (Z_y/D)\mu_{x\zeta}$$
$$+[(Z_yB_x - Z_xB_y)/D^2 + GB/D^3]\mu_y - (Z_x/D)\mu_{y\zeta}, \quad (3)$$

  where $G = Z_xD_y - Z_yD_x$.
- However, $\mu_{R\varphi}$ is no longer continuous as in axisymmetric case ('extruded bonus' lost).
- Therefore, operations like $\mu_{RR\varphi}$ or $\mu_{RR\varphi\varphi}$ are now prohibited.
- In 3D, (discrete) physics equations can be modified to avoid these operations:
  - perform more integration by parts to move derivatives around;
  - introduce auxillary fields (e.g., current density).
  - In particular, $f \to f_\varphi$: $\mathbf{B} = \nabla\psi \times \nabla\varphi - \nabla_\perp f_\varphi + F\nabla\varphi$, $R^2\nabla_\perp^2 f_\varphi = F_\varphi$.

2D implementation can be compared with original code

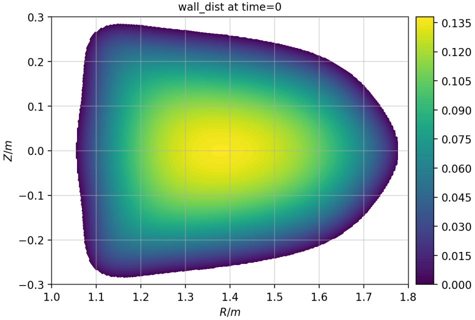
- Implemented: logical–physical transformation;
  - elements constructed on logical mesh.

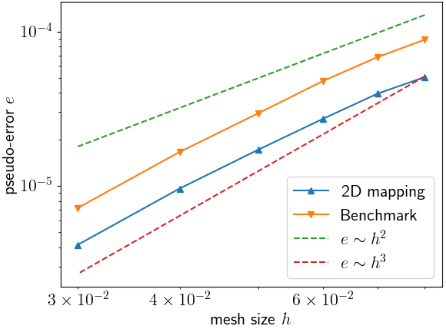
- Benchmark: original 2D code;
  - elements on physical mesh;
  - direct computation.



- Useful practice before 3D implementation:
  - make sure the method works;
  - identify where changes are needed.

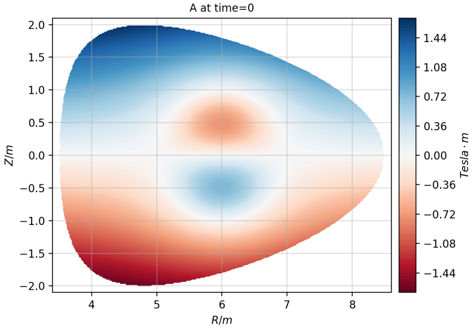
Convergence tests show acceptable results



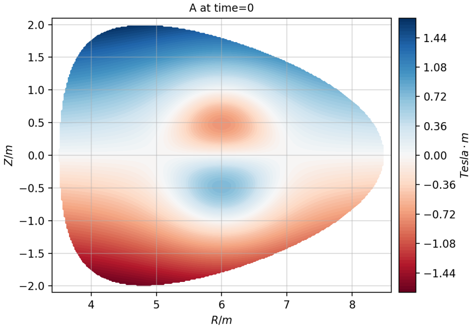


- 'Wall distance': $\nabla^4 \lambda = 0$; $\lambda = 0$ and $\partial_n \lambda = -1$ at boundary.
- Convergence rate is comparable to benchmark.

Nonlinear dynamical simulation seems to work properly



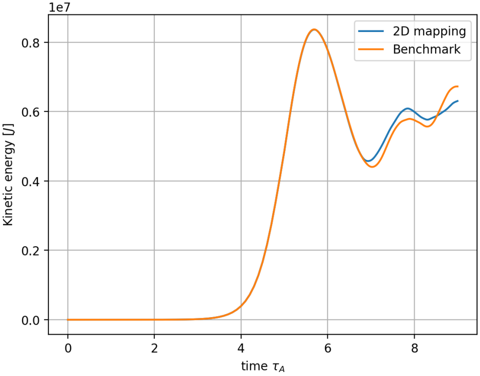
Solution using 2D coordinate mapping

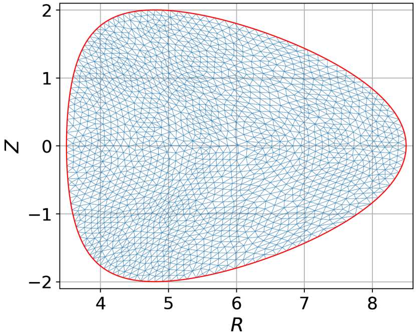


Benchmark using original 2D code

- Ideal tilt instability in a tokamak-shaped domain.
  - boundaries don't quite matter (at least early on).

Time traces also show good agreement

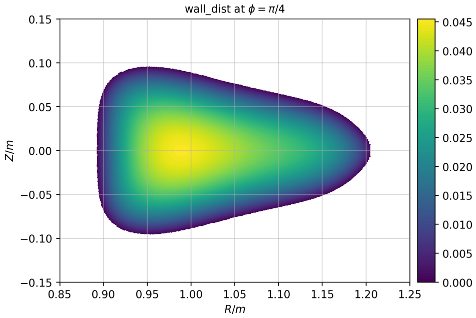


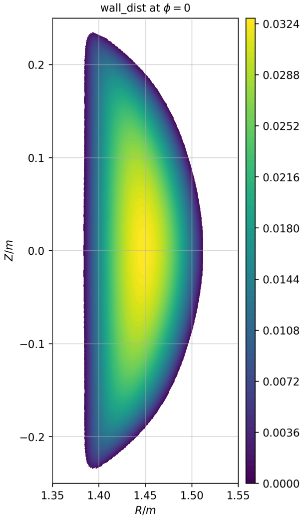


- More than satisfactory, given relatively low resolution (~ 3000 elements).

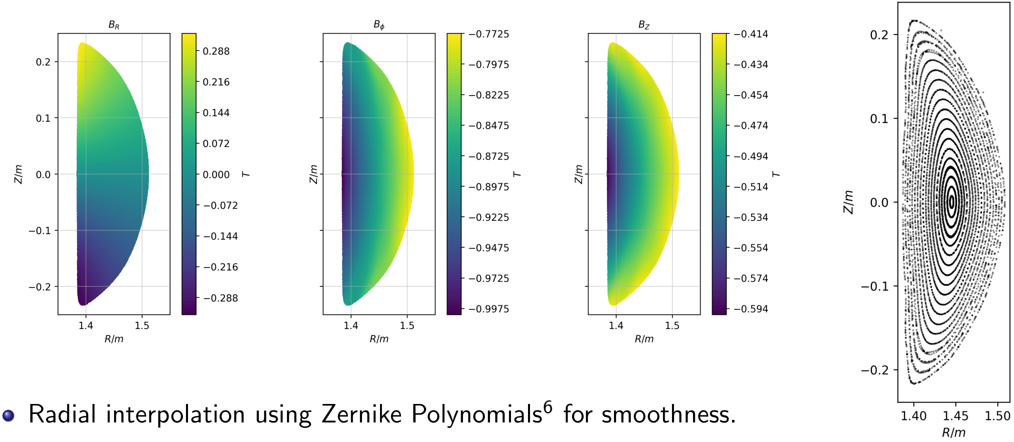
3D boundary value problem can be solved properly

- 'Wall distance' in HSX-like geometry:
  - $\nabla_\perp^4 \lambda = 0$; $\lambda = 0$ and $\partial_{n_\perp} \lambda = -1$ at boundary.





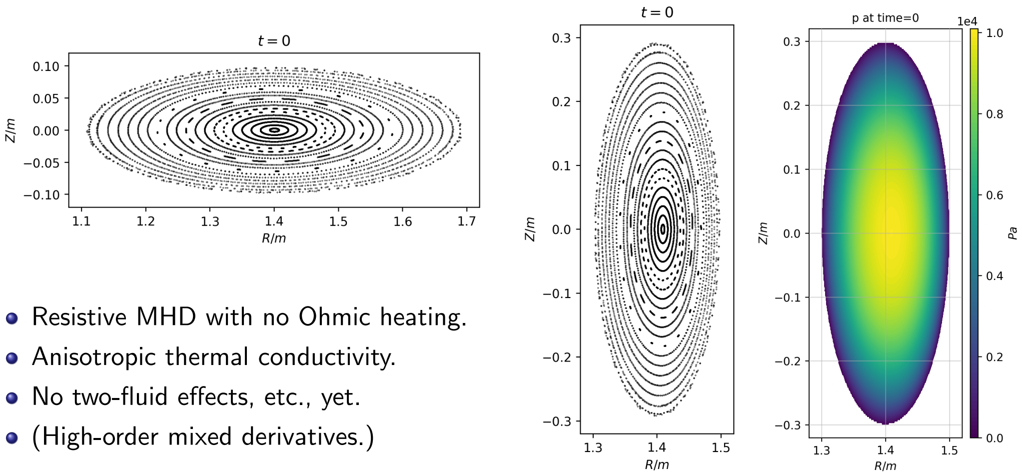
3D VMEC equilibrium set up by solving boundary value problem



- Radial interpolation using Zernike Polynomials[6] for smoothness.

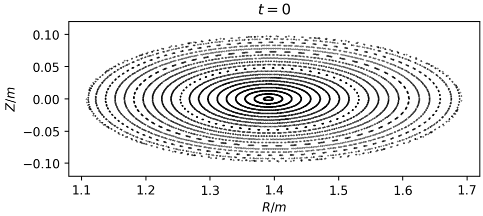
[6] Z. Qu et al, PPCF 62 124004 (2020). D. Dudt and E. Kolemen, PoP 27, 102513 (2020).

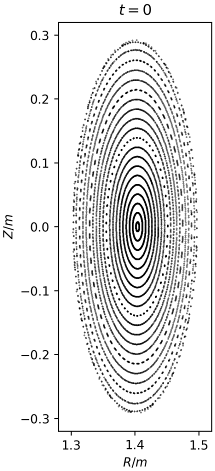
Rotating ellipse: VMEC equilibrium stays (almost) static

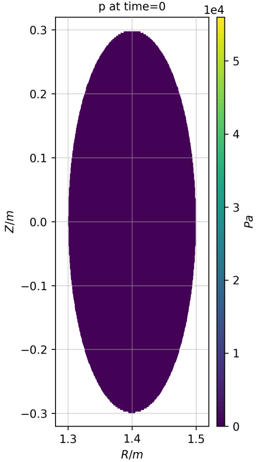


- Resistive MHD with no Ohmic heating.
- Anisotropic thermal conductivity.
- No two-fluid effects, etc., yet.
- (High-order mixed derivatives.)

Rotating ellipse: heat source and Shafranov shift







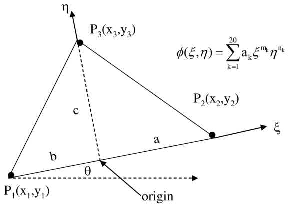
- Cannot handle realistic geometries yet.
- Improper magnetic boundary conditions.
- $\mathbf{B} = \nabla\psi \times \nabla\varphi - \nabla_{\perp} f_{\varphi} + F\nabla\varphi$.
- Need $f \to f_{\varphi}$ change (ongoing).

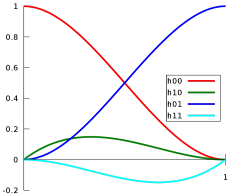
Summary and outlook

- Motivation: nonlinear, initial-value MHD simulations of stellarator plasmas.
- Project: extend the state-of-the-art M3D-C^1 code to stellarator geometry.
- Approach: logical–physical coordinate transformation.
  - Use existing finite elements in logical coordinates (x, y, ζ).
  - Use existing physics equations in physical coordinates (R, Z, φ).
  - Derivatives connected by chain rule; C^1 property respected.
- 2D: implementation verified against original code.
- 3D: preliminarily working, more to be done ($f \to f_\varphi$ in particular).
  - Properly treat magnetic boundary conditions.
  - Avoid high-order mixed derivatives like $\mu_{RR\varphi\varphi}$.
- Hope to simulate realistic stellarator plasmas in a few months.

Reduced quintics and Hermite cubics are both C^1 elements

- Reduced quintics[7] (local coordinates ξ, η):
  $u(\xi,\eta) = \sum_{k=1}^{20} a_k \xi^{m_k} \eta^{n_k} = \sum_{j=1}^{18} u_j \nu_j(\xi,\eta)$.
  - Coefficients: $a_k = \sum_{j=1}^{18} g_{kj} u_j$.
  - Basis functions: $\nu_j(\xi,\eta) = \sum_{k=1}^{20} g_{kj} \xi^{m_k} \eta^{n_k}$.
  - DoFs: $u, u_\xi, u_\eta, u_{\xi\eta}, u_{\xi\xi}, u_{\eta\eta}$ on three nodes.
- Hermite cubics ($x \in [0,1]$):
  $v(x) = v_0 h_{00}(x) + v_0' h_{10}(x) + v_1 h_{01}(x) + v_1' h_{11}(x)$.
  - Basis functions:
    $h_{00} = 2x^3 - 3x^2 + 1,\ h_{10} = x^3 - 2x^2 + x,$
    $h_{01} = -2x^3 + 3x^2,\ h_{11} = x^3 - x^2.$
  - DoFs: v_0, v_0', v_1, v_1'.





[7] S. C. Jardin, J. Comput. Phys. 200, 133 (2004).