PRINCETON PLASMA PHYSICS LABORATORY

A survey of fusion reactor systems codes, and how you can help

Charles Swanson

PPPL Theory Seminar, 2022/06/09

Outline

- What is a systems code?
- Common architectures
- Notable systems codes
- Recent and ongoing developments
- What is holding back widespread adoption of the front runner systems codes?
- What can *you* do?

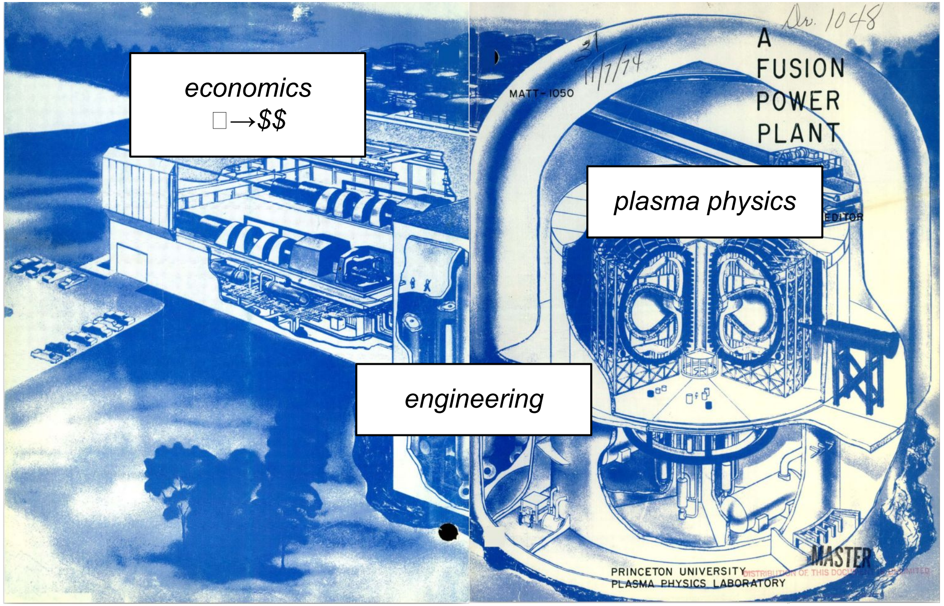
PRINCETON PLASMA PHYSICS LABORATORY

What is a systems code?

A systems code models the whole facility

Rather than modeling one plasma physics phenomenon or engineering system, a systems code models (or designs) the **entire fusion reactor facility.**

A systems code integrates **plasma physics**, **engineering**, and **economics** models.

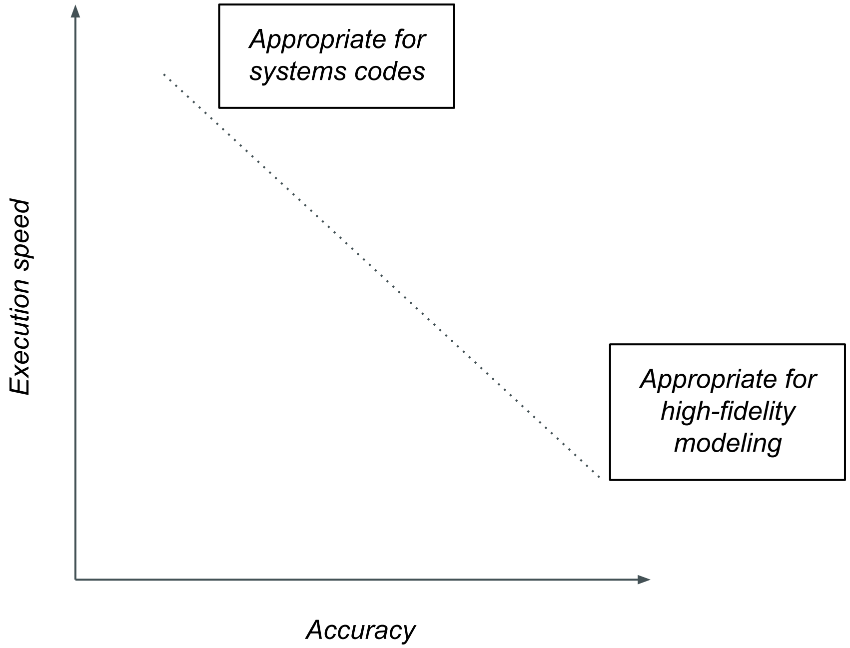


A scan of the cover of the 1974 PPPL technical report "A Fusion Power Plant,"[1] with plasma physics, engineering, and economic features visible

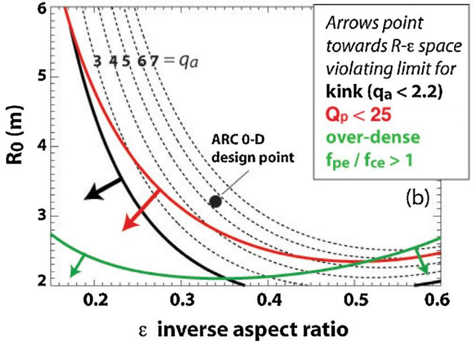
Systems code models are often low-fidelity

Systems codes prioritize simplicity and execution speed over accuracy.

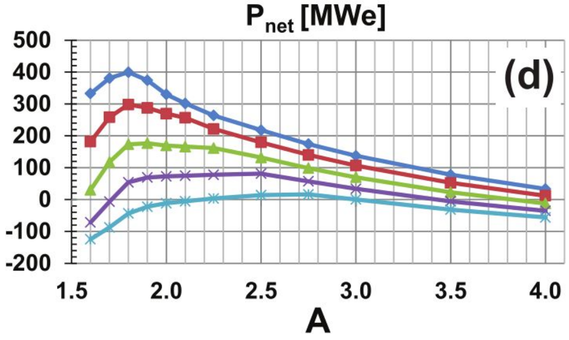
Fast, low-fidelity models are better for systems codes.



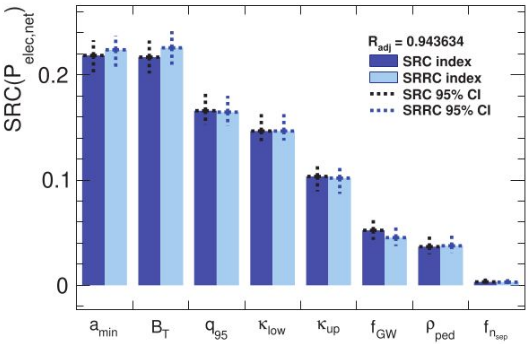
Systems codes are used for scoping, optimization, and more recently, sensitivity and uncertainty quantification



A scoping study of the ARC reactor, by Sorbom et al.[2]



A scoping study of a spherical tokamak pilot plant, by Menard et al.[3], showing an optimal aspect ratio



A sensitivity analysis, showing regression coefficients of several parameters on the net electric power, by Kahn et al.[4]

The integration of many low-fidelity models allows broad surveys of parameter space.

Approximate optima may be found, to target designs for higher-fidelity analysis.

"Design it for me"

More recently, analyses have attempted to quantify the sensitivity of high-level parameters (net electric power) to low-level inputs.

Examples of low-fidelity models

Plasma physics:

- Transport: IBP98(y,2)
- MHD stability: Prescribed β_N
- Profiles: $(1-\varrho^2)^\alpha$

Engineering:

- Neutron shielding: Exponential model
- TF coil stress: Generalized plane strain, vertical separating + centering
- Heat cycle efficiency: (Carnot efficiency) x (derating factor)

Economic:

- Availability: Prescribed factor
- Cost: Levelized Cost Of Electricity

PRINCETON PLASMA PHYSICS LABORATORY

Common Architectures

Common Architectures: Summary

Common systems code architectures are:

- Explicit
- Optimizer-based
- Database-based

Explicit

Explicit systems codes are the simplest: Give an output (P_{net}, Q, LCOE) for each set of inputs.

Constraints are typically used to reduce the dimensionality of the input domain to 1D or 2D.

These codes are best suited to scoping studies.

Examples: Spreadsheets[5,6,more], Segal and Freidberg's FusionSystems[7,8,9], Tokamak Energy Systems Code (TESC)[10-13]

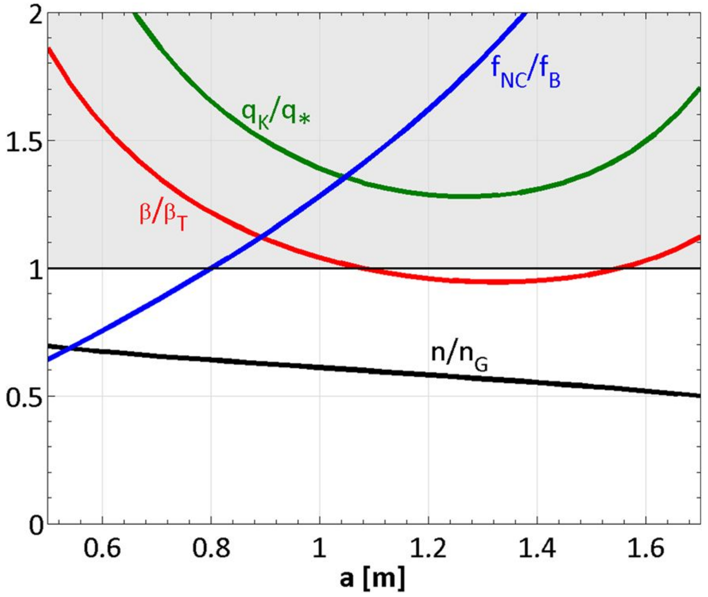


Figure from Friedberg et al.[7] which later became FusionSystems[9]. The entire space of fusion reactors is boiled down to a 1D dependence on minor radius, ***a****. All other parameters are formulated as functions of* ***a*** *using the chosen set of constraints: Neutron wall load, max toroidal field, net electric power, ignition criterion, etc.*

Aside: Spreadsheets

Reactor design spreadsheets are a kind of explicit systems code!

Because *everyone* has written one, does this mean there are thousands of systems codes?

Food for thought.

A1 | *fx* | Plasma physicists who have, at some point, written a reactor design spreadsheet

	A	B
1	Plasma physicists who have, at some point, written a reactor design spreadsheet	100%
2	Plasma physicists who have never written a reactor design spreadsheet	0%

Constrained numerical optimizer

Optimizer-based systems codes are built around constrained numerical optimizers.

They numerically maximize or minimize some figure of merit such as major radius or capital cost, subject to numerous constraints.

These codes attempt to find a near-optimal design point from which to start a high-fidelity design ("*design it for me*").

Examples: PROCESS[14-16], SYCOMORE[4,17]

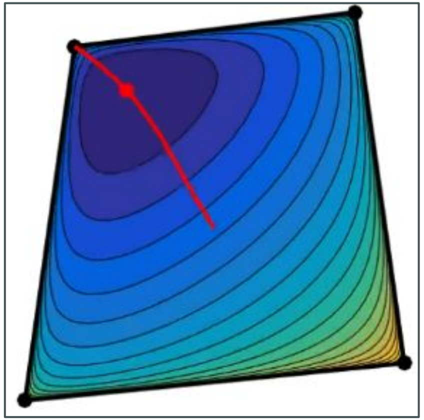


Illustration of constrained optimization via the interior point method, Gpeyre/Wikimedia commons, CC BY-SA 4.0

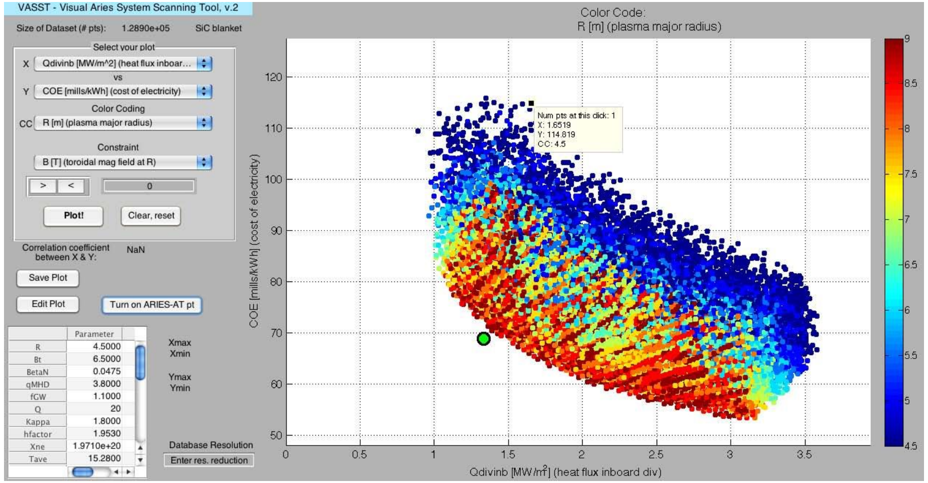
Database

Systems codes based on the **database** model run explicit analyses on a large number of inputs (10^4-10^8), spanning a high-dimensional input domain.

Constraints and figures of merit are applied after the fact, and can be easily mixed-and-matched.

These codes are well-suited to uncertainty and sensitivity analyses.

Examples: ARIES Systems Code (ASC)[18,19], Unnamed FESS systems code.[20,21]



Screenshot from a post-processor for ASC[18], in which constraints are applied and figures of merit are evaluated after the fact

Notable systems codes

Notable systems codes: Summary

There are 20+ systems codes in the literature that meet either criterion: 2+ peer-reviewed papers, or open source. I have chosen 6 to discuss here:

- PROCESS
- TESC
- ARIES/FESS
- SYCOMORE
- BLUEMIRA
- FAROES

PROCESS

Name: PROCESS[14-16,22]

Institution: CCFE / UKAEA

Model: Optimizer

Availability: Proprietary, can collaborate with license agreement. Hosted on UKAEA gitlab repository. PPPL has a license agreement.

Language: FORTRAN90 (+Python)

Arguably the **front runner**, with 60+ peer-reviewed publications.[22] The most benchmarked systems code. Based around the VMCON optimizer. Extensive library of specific plasma physics and engineering models.

Unofficial PROCESS logo from CCFE gitlab server

TESC

Name: Tokamak Energy Systems Code (TESC)[10-13]

Institution: Tokamak Energy Ltd (TE)

Model: Explicit

Availability: Proprietary. "TE is very open to the idea of collaborations and/or licensing arrangements." - AE Costley

Language: MATLAB

TE logo

TE and other **private companies have opted to write their own** systems codes. TESC is an explicit code, and so implements a prescribed workflow:

Geometry→Wall load sets P_{fusion}→Required Q sets P_{aux}→Prescribed T_0 sets n_0→*etc.*

ARIES / FESS Systems Code

Name: ARIES Systems Code (ASC)[18,19], later the unnamed FESS Systems Code[20,21]

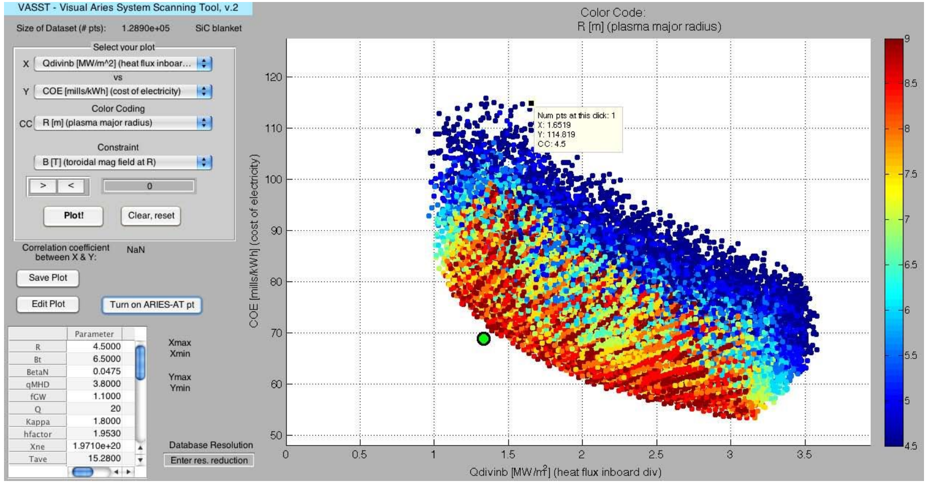
Institution: ARIES team, later FESS or CE Kessel

Model: Database

Availability: Proprietary (ASC ran on PPPL's cluster in the mid 2000s)

Language: Fortran, then C++, then Fortran again

Fortran version used by the ARIES team to design and cost the various ARIES studies. Then ported to C++. Now a successor to the Fortran ASC is used by CE Kessel and MS Tillack to design FESS FNSF. The **only widely used database-based systems code**.



Screenshot from a post-processor for ASC[18], in which constraints are applied and figures of merit are evaluated after the fact

SYCOMORE

Name: SYCOMORE[4,17]

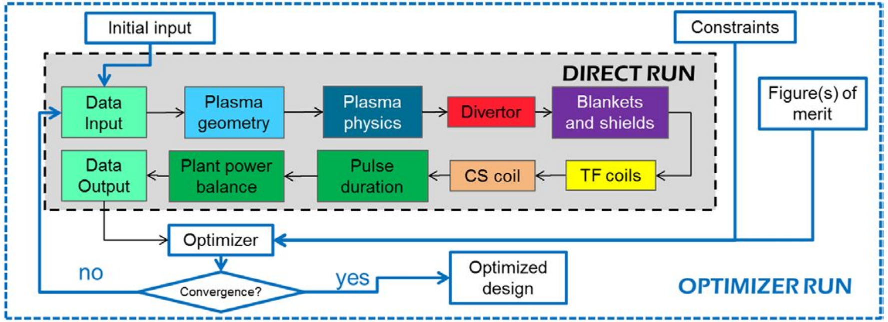
Institution: CEA (French atomic energy commission)

Model: Optimizer

Availability: Proprietary, can collaborate with a license agreement. PPPL does not have a license agreement.

Language: EU-ITM (predecessor to IMAS)

The only **truly modular code** on this list. Modular by virtue of existing within the EU-ITM framework, the predecessor to IMAS. Kepler workflow manager. Uranie optimizer framework.



Screenshot of the modular design of SYCOMORE enabling a specific workflow to be implemented.[17]

BLUEMIRA

Name: BLUEMIRA[25,26]

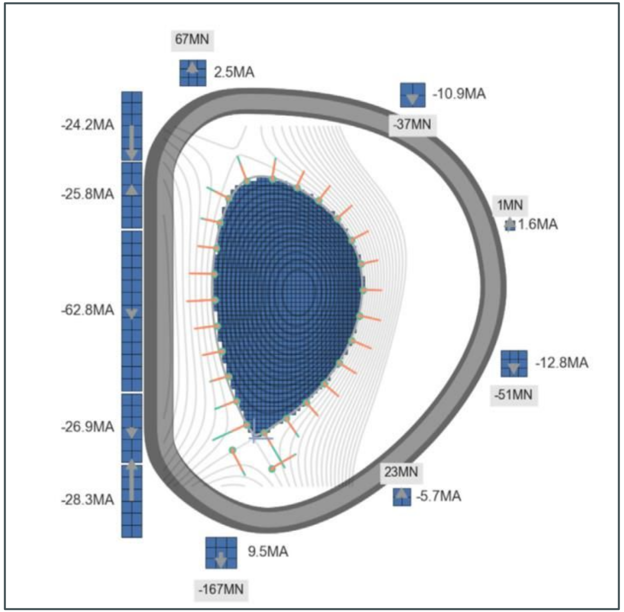
Institution: CCFE and KIT

Model: Multi-fidelity

Availability: Open-source, github (not ready yet)

Language: Python, MATLAB

Formed by merger of CCFE BLUEPRINT[23] and KIT MIRA[24] codes. Aimed at **extending 0D PROCESS runs to 2D and 3D, higher-fidelity**. From a PROCESS solution, applies several analyses: Free-boundary MHD, neutronics, TF coil design, finite element stress analysis, *etc.*



Screenshot of a BLUEMIRA free-boundary MHD analysis[27]

FAROES

Name: FAROES

Institution: Princeton University (JA Schwartz)

Model: Optimizer

Availability: Open-source, github (not ready yet)

Language: Python

Planned to be **open source** when complete. Emphasizes cost modeling and grid integration. Uses open-source optimizer framework OpenMDAO.

Faroe islands. Vincent van Zejist / Wikimedia commons, CC-BY-SA 3.0

Aside: For Stellarators

These have been *Tokamak* systems codes. There are also a couple examples of stellarator systems codes:

- Warmer / IPP stellarator models for PROCESS
  - Originally applicable only to W-7X-like Helias configurations,[28] apparently there is now a general "stellarator preprocessor" for PROCESS which takes in arbitrary shapes.
  - We do not have access to this.
- HELIOSCOPE[29]
  - Specific to the LHD-like Heliotron configuration
  - Used by the Japanese National Institute for Fusion Science to produce their Force Free Helical Reactor series of power plant designs.
  - Explicit systems code.
- ARIES / Lyon *et al.*[30,31]
  - Scales a prescribed stellarator configuration up and down; does not optimize the stellarator plasma itself.
  - Originally for a Torsatron, then eventually a NCSX-style QA design.
- **This is obviously a hole in our capabilities.**

Recent and ongoing developments

Recent and ongoing developments: Summary

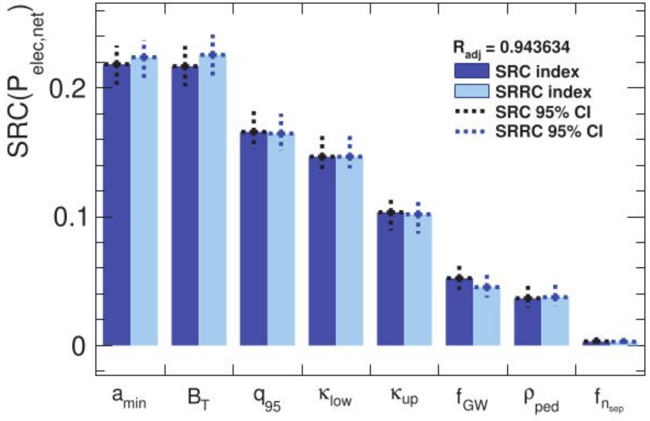
Recent systems code efforts include:

- Sensitivity and uncertainty
- Multi-fidelity analyses
- Cost model development

Sensitivity and uncertainty

Database systems codes have always been useful for sensitivity and uncertainty, because after applying constraints *a posteriori*, one is left with a *region* of valid solutions rather than one single design point.

More recently, statistical analyses have been applied to SYCOMORE[4] and PROCESS[26]. Examples include a linear regression of points distributed around the design point, and the method of elementary effects.



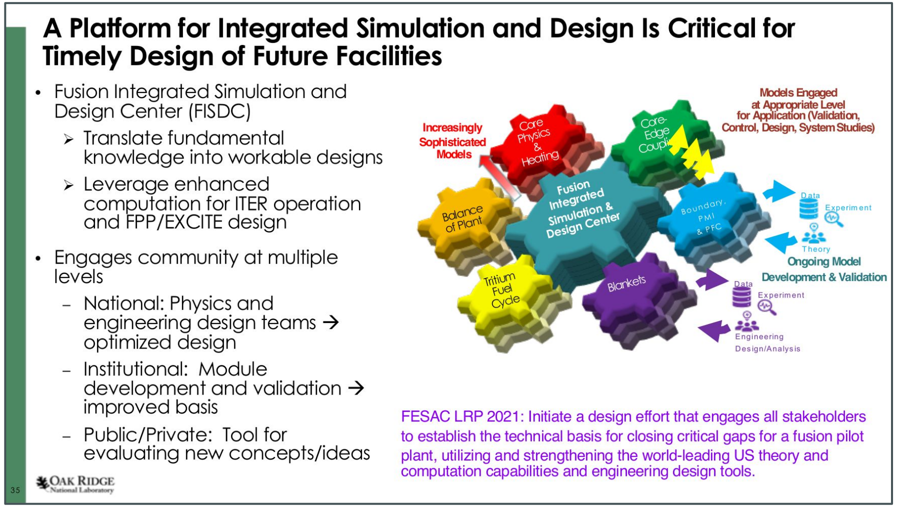
A sensitivity analysis, showing regression coefficients of several parameters on the net electric power, by Kahn et al.[4]

Multi-fidelity analysis

There is a capability gap between the fast, low-fidelity models of systems codes and slow, high-fidelity models. New efforts aim to address this.

BLUEMIRA[25,26] starts from a PROCESS solution and applies higher-fidelity analyses to this design point.

M. Wade gave a PPPL colloquium on ORNL's proposed "Fusion Integrated Simulation and Design Center" (FISDC).[32] ORNL would like to lead a multi-institution effort for integrated design and modeling. "Princeton on the physics side has things like TRANSP" -M. Wade. No recent public activity.



A slide from M. Wade's PPPL colloquium[32]

Cost model

Capital cost and LCOE are arguably the ultimate figures of merit for any fusion reactor. However they are also the most uncertain.

The majority of systems codes with cost models cite that of Starfire[33] and Generomak.[34,35] These assume cost scales with mass or power, with some exponent.

STEP, under Hanni Lux, is writing a new cost model for PROCESS. However it will be proprietary.

Woodruff Scientific has an ARPA-E award to update their costing model.

Better costing models are required.

Milburn Pennybags, alias "Mr. Monopoly"

PRINCETON PLASMA PHYSICS LABORATORY

What is holding back widespread adoption of the front runner systems codes?

ALERT! **Opinion** starts here. We have now left behind the realm of facts

Why would widespread adoption of an accessible, modular, flexible systems code be a good thing?

Common framework for comparisons

Results are reproducible, can be shared

Small teams get better tools with fewer resources

Obvious what assumptions a team is using when making a performance claim (based on what modules they're using)

- See the spreadsheet included in the proposal packet for ARPA-E BETHE concept teams

Summary

What is holding back widespread adoption of the front runner systems codes?

- Lack of accessibility
- Lack of flexibility
- Lack of modularity

Lack of accessibility

The best systems codes are proprietary. They require license agreements between institutions.

Some systems codes have chosen open-source software licenses: FAROES, BLUEMIRA[25,26], FusionSystems[7,8,9]. This should be encouraged.

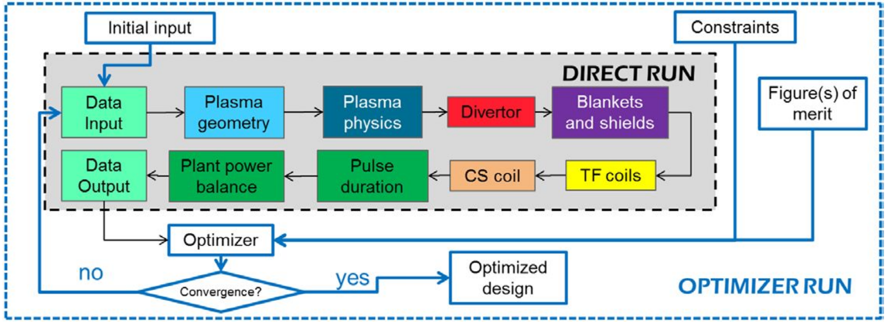
Various open-source software licenses

Lack of flexibility

Every explicit systems code implements a different workflow. Different parameters are inputs, other parameters are set by different constraints. Ex: “Holding fusion power constant,” or “Holding plasma volume constant,” or “Holding major radius constant.”

Optimizer-based systems codes partially address this, with user-selectable iteration variables and constraints.

Systems code architectures should be workflow-agnostic.



Screenshot of the modular design of SYCOMORE enabling general workflows to be implemented.[17]

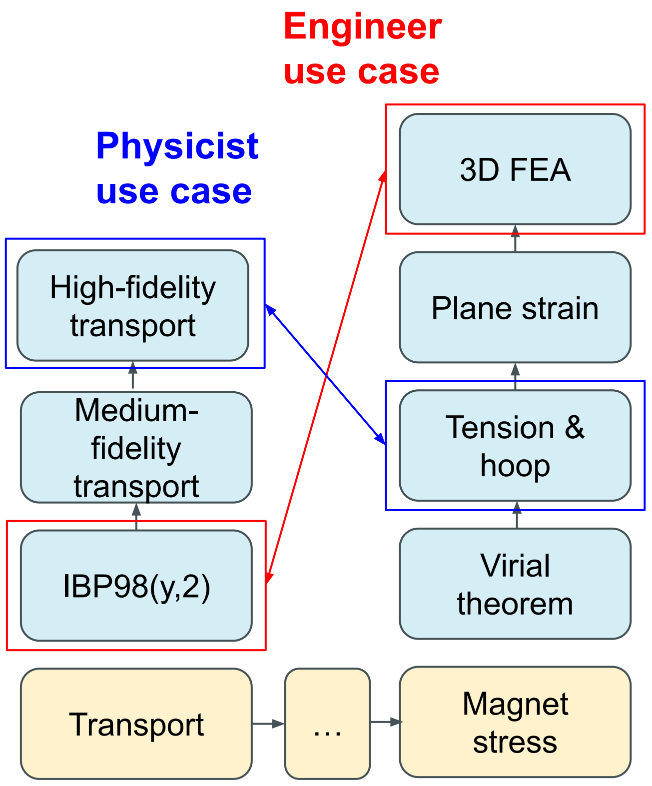
Lack of modularity

(this is related to flexibility)

If models of phenomena and systems are *modules*, they can be mixed-and-matched and considered in isolation.

T. Brown: [*paraphrased*] "Every systems code is based on a specific machine design, with specific assumptions baked deeply into the implementation"

Modularity can address this shortcoming. Modules for different divertor types, different current drive systems, different operating scenarios, different heat conversion cycles, *etc.*



What can *you* do?

What can *you* do? Summary

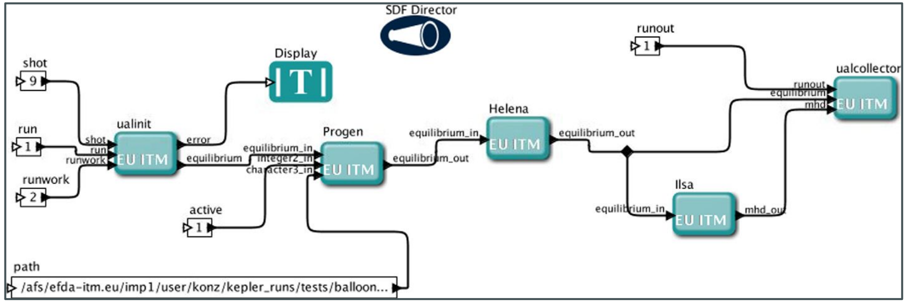
The following are avenues toward increased flexibility, fidelity, and utility:

- New architectures for interfacing modules
  - Think IMAS IDS
- New architectures for multi-fidelity capability
  - Feed-forward up-fidelity: How to keep track of the additional information?
  - Feed-back down-fidelity: How to incorporate the result of higher-fidelity analyes to *a posteriori* update the low-fidelity analysis?
- Low-fidelity models of Stellarator phenomena and systems

New architectures for interfacing modules

IMAS Interface Data Structure (IDS)[37] stores useful info: What module produced it, what version, what inputs. We'll want something similar.

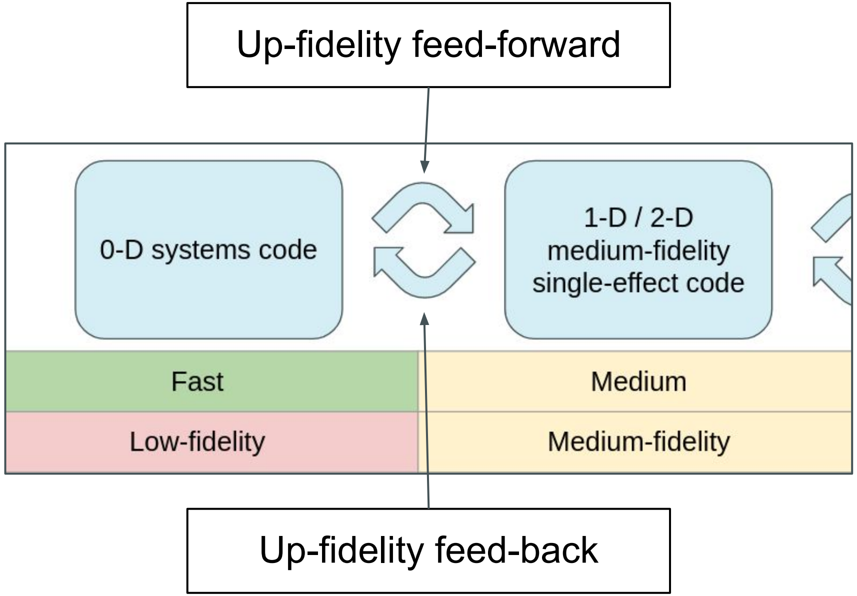
Full modularity would require modules to be called in any direction, by any other module, at any level of fidelity. This requires a dedicated architecture.



Screenshot of a modular workflow implemented in EU-ITM, the predecessor to IMAS[37]

The up-fidelity process: feed-forward and feed-back

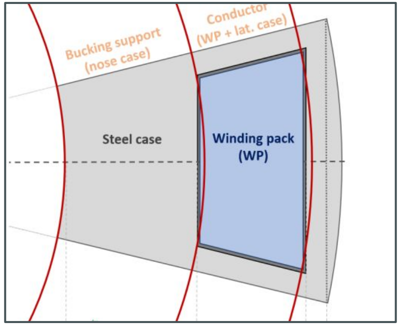
- Feed-forward: When a low fidelity solution is used as the input to a higher fidelity analysis
  - This process is *underconstrained*; see next slide
- Feed-back: When the result of a higher fidelity solution is used to *a posteriori* update the low fidelity model
  - This process has *a few known implementations*; see two slides from now



The underconstrained feed-forward process

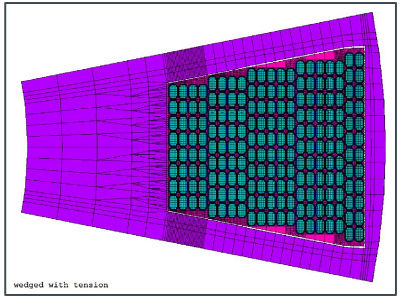
Going from low to high fidelity requires the implicit choice of many more free parameters

- Ex: Lumped-element TF coil winding pack → Multi-layer winding pack.
- Proposed implementation: User selects from a library of specific pre-made up-fidelity functions implementing different engineering strategies, with user-selectable customization parameters.
- Fanciful pseudocode:
  - `dLumped = Converge(dLumped);`
  - `dWinding = YZhaiCORCWinding();`
  - `dWinding.CORCAround = [5 4 4];`
  - `dMultiLayer = YZhaiCORCWinding(dLumped,dWinding);`
- Output data structures (`dLumped`, `dMultiLayer`) store provenance and input parameters as with IMAS IDS



PROCESS TF coil documentation

layers
CORC/Viper/CroCo/CICC
Pitch twist
Grading info
Poisson's Ratio
Anisotropy
Winding pattern

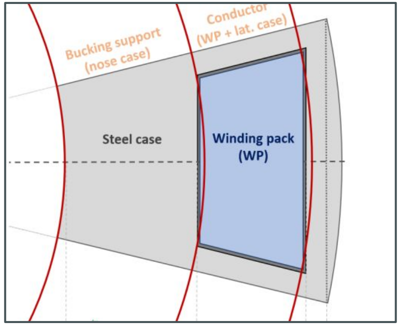


Zhai *et al.* 2018[36]

The *a posteriori* update feed-back process

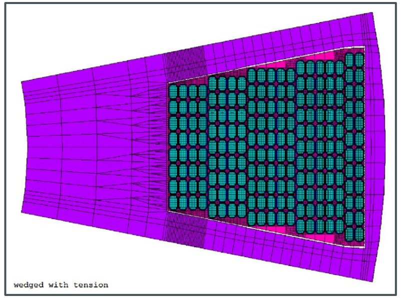
Every high fidelity run is information that could be used to improve the low-fidelity model

- Getting a different answer with a higher fidelity run is expected. It shouldn't *invalidate* the low fidelity result, it should *inform* it for next time. Some methods for doing this are known:
    - Simplest possible example: Store a **fudge factor** that corrects the low fidelity answer
    - The default / classic method is **power law regression**
    - The exciting new method is **Neural Network** back-propagation
    - Sometimes structure of the problem **implies a form** of the *a posteriori* update. Ex: Update e-folding length from MNCP runs
- Conversation with JA Schwartz: The updates to the low fidelity model should be a live service such as Amazon cloud. Everyone gets the immediate benefit from every analysis.



PROCESS TF coil documentation

Fudge factor
Multivariable regression
Neural network
Implied form (use physics judgement)



Zhai *et al.* 2018[36]

Low-fidelity models of Stellarator phenomena and systems

Tokamaks have a β limit; Stellarators have soft limit, $P_{transport}(\beta)$

Tokamaks have IPB98(y,2); Stellarators have ISS04

Tokamaks have ripple α-particle loss; Stellarators have 3D α-particle loss

Tokamaks have known B_{max} vs B_0; Stellarators have to solve Laplace's equation

Tokamaks have X-pt divertor, 2-pt model; Stellarators have island divertor, ergodic cross-field diffusion

Tokamak components can be represented axisymmetrically; Stellarators are strongly 3D

etc.

PRINCETON
PLASMA PHYSICS
LABORATORY

Backup slides

References

[1] Mills, R. G. “Fusion Power Plant.” PPPL (Princeton Plasma Physics Laboratory (PPPL), Princeton, NJ (United States)), August 1, 1974. https://doi.org/10.2172/4266652.

[2] Sorbom, B. N., J. Ball, T. R. Palmer, F. J. Mangiarotti, J. M. Sierchio, P. Bonoli, C. Kasten, et al. “ARC: A Compact, High-Field, Fusion Nuclear Science Facility and Demonstration Power Plant with Demountable Magnets.” *Fusion Engineering and Design* 100 (November 1, 2015): 378–405. https://doi.org/10.1016/j.fusengdes.2015.07.008.

[3] Menard, J E, T Brown, L El-Guebaly, M Boyer, J Canik, B Colling, R Raman, et al. “Fusion Nuclear Science Facilities and Pilot Plants Based on the Spherical Tokamak.” *Nucl. Fusion*, 2016, 44.

[4] Kahn, S., C. Reux, J.-F. Artaud, G. Aiello, J.-B. Blanchard, J. Bucalossi, S. Dardour, et al. “Sensitivity Analysis of Fusion Power Plant Designs Using the SYCOMORE System Code.” *Nuclear Fusion* 60, no. 1 (November 2019): 016015. https://doi.org/10.1088/1741-4326/ab4879.

[5] Menard, J E, T Brown, L El-Guebaly, M Boyer, J Canik, B Colling, R Raman, et al. “Fusion Nuclear Science Facilities and Pilot Plants Based on the Spherical Tokamak.” *Nucl. Fusion*, 2016, 44.

[6] Stambaugh, R. D., V. S. Chan, A. M. Garofalo, M. Sawan, D. A. Humphreys, L. L. Lao, J. A. Leuer, et al. “Fusion Nuclear Science Facility Candidates.” *Fusion Science and Technology* 59, no. 2 (February 1, 2011): 279–307. https://doi.org/10.13182/FST59-279.

[7] Freidberg, J. P., F. J. Mangiarotti, and J. Minervini. “Designing a Tokamak Fusion Reactor—How Does Plasma Physics Fit In?” *Physics of Plasmas* 22, no. 7 (July 1, 2015): 070901. https://doi.org/10.1063/1.4923266.

[8] Segal, Daniel J., Antoine J. Cerfon, and Jeffrey P. Freidberg. “Steady State versus Pulsed Tokamak Reactors.” *Nuclear Fusion*, January 2021. https://doi.org/10.1088/1741-4326/abe0d2.

[9] djsegal. *Djsegal/FusionSystems.Jl*. Jupyter Notebook, 2020. https://github.com/djsegal/FusionSystems.jl.

References

[10] Chan, V. S., A. E. Costley, B. N. Wan, A. M. Garofalo, and J. A. Leuer. “Evaluation of CFETR as a Fusion Nuclear Science Facility Using Multiple System Codes.” *Nuclear Fusion* 55, no. 2 (January 2015): 023017. https://doi.org/10.1088/0029-5515/55/2/023017.

[11] Costley, A. E. “Towards a Compact Spherical Tokamak Fusion Pilot Plant.” *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 377, no. 2141 (March 25, 2019): 20170439. https://doi.org/10.1098/rsta.2017.0439.

[12] Costley, A. E., J. Hugill, and P. F. Buxton. “On the Power and Size of Tokamak Fusion Pilot Plants and Reactors.” *Nuclear Fusion* 55, no. 3 (January 28, 2015): 033001. https://doi.org/10.1088/0029-5515/55/3/033001.

[13] Costley, AE. “A System Code for the Determination of Key Parameters of Tokamaks Pilot Plants and Reactors.” Tokamak Energy Ltd. Note. Culham Science Centre, Abingdon, OX14 3DB, UK: Tokamak Energy Ltd., October 13, 2014. https://iopscience.iop.org/article/10.1088/0029-5515/55/3/033001/data.

[14] Knight, PJ, and MD Kovari. “A User Guide to the PROCESS Fusion Reactor Systems Code.” Culham Centre for Fusion Energy, January 12, 2016. https://ccfe.ukaea.uk/wp-content/uploads/2019/11/A-User-Guide-to-the-PROCESS-systems-code.pdf.

[15] Kovari, M., F. Fox, C. Harrington, R. Kembleton, P. Knight, H. Lux, and J. Morris. “‘PROCESS’: A Systems Code for Fusion Power Plants – Part 2: Engineering.” *Fusion Engineering and Design* 104 (March 1, 2016): 9–20. https://doi.org/10.1016/j.fusengdes.2016.01.007.

[16] Kovari, M., R. Kemp, H. Lux, P. Knight, J. Morris, and D. J. Ward. “‘PROCESS’: A Systems Code for Fusion Power Plants—Part 1: Physics.” *Fusion Engineering and Design* 89, no. 12 (December 1, 2014): 3054–69. https://doi.org/10.1016/j.fusengdes.2014.09.018.

[17] Reux, C., L. Di Gallo, F. Imbeaux, J.-F. Artaud, P. Bernardi, J. Bucalossi, G. Ciraolo, et al. “DEMO Reactor Design Using the New Modular System Code SYCOMORE.” *Nuclear Fusion* 55, no. 7 (July 1, 2015): 073011. https://doi.org/10.1088/0029-5515/55/7/073011.

[18] Carlson, Lane, Mark Tillack, Farrokh Najmabadi, and Charles Kessel. “Development of a Visualization Tool for the ARIES Systems Code.” *Fusion Science and Technology* 60, no. 2 (August 1, 2011): 459–63. https://doi.org/10.13182/FST60-459.

References

[19] Dragojlovic, Zoran, A. Rene Raffray, Farrokh Najmabadi, Charles Kessel, Lester Waganer, Laila El-Guebaly, and Leslie Bromberg. "An Advanced Computational Algorithm for Systems Analysis of Tokamak Power Plants." *Fusion Engineering and Design* 85, no. 2 (April 1, 2010): 243–65. https://doi.org/10.1016/j.fusengdes.2010.02.015.

[20] Kessel, C. E., D. B. Batchelor, P. T. Bonoli, M. E. Rensink, T. D. Rognlien, P. Snyder, G. M. Wallace, and S. J. Wukitch. "Core Plasma Physics Basis and Its Impacts on the FNSF." *Fusion Engineering and Design*, Special Issue: FESS-FNSF Study, 135 (October 1, 2018): 356–69. https://doi.org/10.1016/j.fusengdes.2017.06.003.

[21] Kessel, C.E., and Mark Tillack. "FESS Conference Call." Slideshow presented at the FESS Conference Call, April 14, 2021.

[22] PROCESS Publications Webpage. "PROCESS Publications Webpage," n.d. http://process.gitpages.ccfe.ac.uk/process/publications/.

[23] Coleman, M., and S. McIntosh. "BLUEPRINT: A Novel Approach to Fusion Reactor Design." *Fusion Engineering and Design* 139 (February 1, 2019): 26–38. https://doi.org/10.1016/j.fusengdes.2018.12.036.

[24] Franza, F, L V Boccaccini, E Fable, I Landman, I A Maione, S Petschanyi, R Stieglitz, and H Zohm. "MIRA: A Multiphysics Approach to Designing a Fusion Power Plant." In *Proceedings of the 28th IAEA Fusion Energy Conference*, 16. Virtual: IAEA, 2021. https://conferences.iaea.org/event/214/contributions/17054/.

[25] Holloway, Nick. "New Code Will Bring Better Power Plant Designs in a Fraction of the Time." *Culham Centre for Fusion Energy* (blog), March 18, 2021. https://ccfe.ukaea.uk/new-code-will-bring-better-power-plant-designs-in-a-fraction-of-the-time/.

[26] Morris, J., M. Coleman, S. Kahn, S. I. Muldrew, A. J. Pearce, D. Short, J. E. Cook, et al. "Preparing Systems Codes for Power Plant Conceptual Design." *Nuclear Fusion* 61, no. 11 (September 2021): 116020. https://doi.org/10.1088/1741-4326/ac18dd.

[27] Morris, J, M Coleman, S Kahn, S I Muldrew, A J Pearce, D Short, J E Cook, et al. "Preparing Systems Codes for Power Plant Conceptual Design." In *Proceedings of the 28th IAEA Fusion Energy Conference (FEC 2020)*, 11. IAEA, 2021. https://conferences.iaea.org/event/214/contributions/17055/.

References

[28] Warmer, F., C. D. Beidler, A. Dinklage, K. Egorov, Y. Feng, J. Geiger, F. Schauer, Y. Turkin, R. Wolf, and P. Xanthopoulos. “HELIAS Module Development for Systems Codes.” *Fusion Engineering and Design* 91 (February 1, 2015): 60–66. https://doi.org/10.1016/j.fusengdes.2014.12.028.

[29] Goto, T., Y. Suzuki, N. Yanagi, K. Y. Watanabe, S. Imagawa, and A. Sagara. “Importance of Helical Pitch Parameter in LHD-Type Heliotron Reactor Designs.” *Nuclear Fusion* 51, no. 8 (July 2011): 083045. https://doi.org/10.1088/0029-5515/51/8/083045.

[30] Lyon, J. F., K. Gulec, R. L. Miller, and L. El-Guebaly. “Status of the US Stellarator Reactor Study.” *Fusion Engineering and Design* 25, no. 1 (August 1, 1994): 85–103. https://doi.org/10.1016/0920-3796(94)90056-6.

[31] Lyon, J. F., L. P. Ku, L. El-Guebaly, L. Bromberg, L. M. Waganer, and M. C. Zarnstorff. “Systems Studies and Optimization of the ARIES-CS Power Plant.” *Fusion Science and Technology* 54, no. 3 (October 2008): 694–724. https://doi.org/10.13182/FST54-694.

[32] Wade, Mickey. “ORNL’s Role in Enabling an Aggressive Path to Fusion Energy.” PPPL Colloquium presented at the PPPL Colloquium, Princeton Plasma Physics Laboratory, Princeton, NJ, USA, September 15, 2021. https://pppl-intranet.princeton.edu/event/pppl-colloquium-mickey-wade-oak-ridge-national-lab-ornl-s-role-enabling-aggressive-path-fusion.

[33] “STARFIRE: A Commercial Tokamak Fusion Power Plant Study.” Argonne National Lab., IL (USA), September 1, 1980. https://doi.org/10.2172/6633213.

[34] Sheffield, J., R. A. Dory, S. M. Cohn, J. G. Delene, L. Parsly, D. E. T. F. Ashby, and W. T. Reiersen. “Cost Assessment of a Generic Magnetic Fusion Reactor.” *Fusion Technology* 9, no. 2 (March 1, 1986): 199–249. https://doi.org/10.13182/FST9-2-199.

[35] Sheffield, Joh, and Stanley L. Milora. “Generic Magnetic Fusion Reactor Revisited.” *Fusion Science and Technology* 70, no. 1 (July 1, 2016): 14–35. https://doi.org/10.13182/FST15-157.

[36] Zhai, Yuhu, Peter Titus, Charles Kessel, and Laila El-Guebaly. “Conceptual Magnet Design Study for Fusion Nuclear Science Facility.” *Fusion Engineering and Design*, Special Issue: FESS-FNSF Study, 135 (October 1, 2018): 324–36. https://doi.org/10.1016/j.fusengdes.2017.06.028.

[37] Imbeaux, F., J. B. Lister, G. T. A. Huysmans, W. Zwingmann, M. Airaj, L. Appel, V. Basiuk, et al. “A Generic Data Structure for Integrated Modelling of Tokamak Physics and Subsystems.” *Computer Physics Communications* 181, no. 6 (June 1, 2010): 987–98. https://doi.org/10.1016/j.cpc.2010.02.001.