



Old Solver, New Tricks

Modernizing and developing the UEDGE codebase in the age of AI

UKAEA Exhaust Code Workshop 2026

A. Holm on behalf of the UEDGE Team

October 9th, 2026

Prepared by LLNL under Contract DE-AC52-07NA27344.

UEDGE is a 2D multi-fluid, edge-plasma code with a fully time-implicit solver, written in Fortran and ported to Python

- Branched off Braams' B2 in 1992 [1], developed over 35 years

1992-2000

Solver and preconditioner improvements [2]
Nonorthogonal grids (1996) [3]
Multi-charge state impurity models (1996) [2,4]
Cross-field drift implementation (1999) [5]
Double null capabilities (1999) [6]

2005-2015

Initial Python interface (2007) [10]
Snowflake capabilities introduced (2007) [11]
Attempts at core-edge coupling for whole-device modeling FACETS project (2008) [10]

2020-2025

Fluid molecular model (2021) [14, 15]
Neutral temperature equations (2021) [16]
Ion anisotropy model (2021) [16]
Increased use training surrogate models [17, 18]
Python interfacing using UETOOLS (2023) [19]

2000-2005

Explicit coupling to DEGAS2 (2000) [7]
First UEDGE user manual revision published (2000) [8]
MPI parallelization using Fortran-C interface and KINSOL package (2002) [9]

2015-2020

Implicit coupling attempts to DEGAS2 (2015-2017) [12]
UEDGE made publicly available on GitHub with Python interface [13]

2025-

Teletransport model [20]
OpenMP parallelization of preconditioner and RHS (2025) [21]
Advanced fluid neutral boundary condition implementation
Grid extensions to device walls
Code modernization and agentic workflows

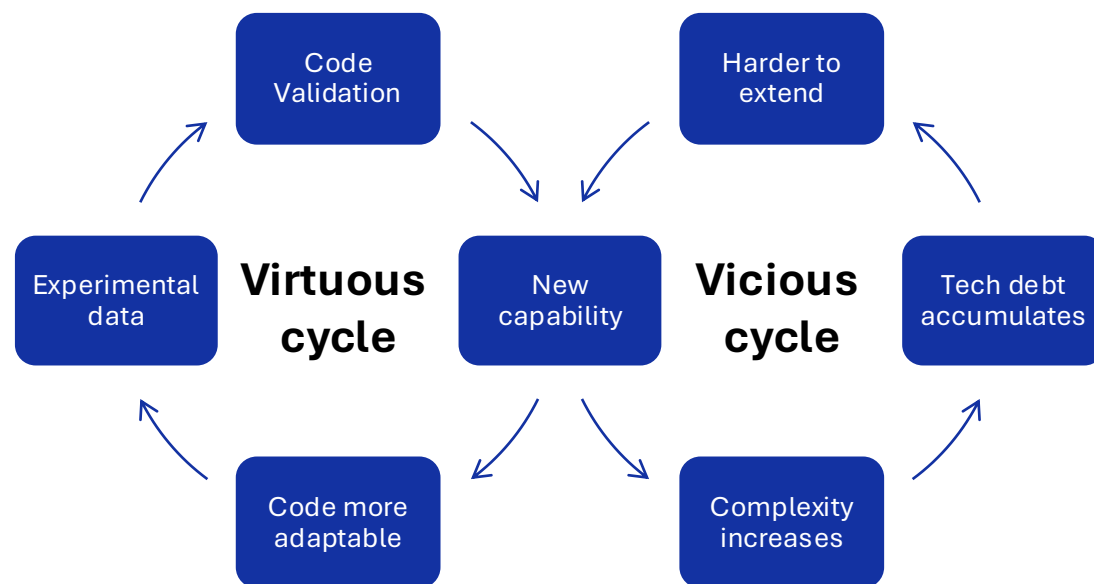
[1] T.D. Rognlien, et al., *J. Nucl. Mater.* **196** (1992) 347
[2] G.R. Smith, et al., *J. Nucl. Mater.* **220** (1995) 1024
[3] M.E. Rensink, T.D. Rognlien, LLNL report doi.org/10.2172/371415
[4] T.D. Rognlien, et al., *Contrib. Plasma Phys.* **34** (1994) 362
[5] T.D. Rognlien, et al., *Phys. Plasmas* **6** (1999) 1851
[6] M.E. Rensink, et al., *Contrib. Plasma Phys.* **40** (2000) 302
[7] D.P. Stotler et al., *Contrib. Plasma Phys.* **40** (2000) 221

[8] T.D. Rognlien, et al., LLNL Report, OSTI ID 15007243
[9] T.D. Rognlien, et al., *J. Comp. Phys.*, 175 (2002) 249
[10] J.R. Cary, et al., *J. Phys.: Conf. Ser.* **125** (2008) 012040
[11] D.D. Ryutov, *Phys. Plasmas*, **14** (2007) 064502
[12] I. Joseph et al., *Nucl. Mater. Energy*, **12** (2017) 813
[13] T.D. Rognlien, et al., LLNL Software (LLNL-CODE-845914), github.com/LLNL/UEDGE
[14] A. Holm A, et al., *Contrib. Plasma Phys.*, **60** (2020) e201900150

[15] A. Holm, et al., *Nucl. Mater. Energy*, **27** (2021) 100982
[16] M. Zhao, et al., *Nucl. Mater. Energy*, **26** (2021) 100881
[17] B. Zhu, et al., *J. Plasma Phys.* **88** (2022) 905880504
[18] B. Zhu, et al., *Phys. Plasmas*, **32** (2024) 062508
[19] A. Holm, et al., LLNL Software (LLNL-CODE-846751), github.com/LLNL/UETOOLS
[20] M. Zhao, et al., *Contrib. Plasma Phys.*, preprint (2026) e70093
[21] A. Holm, et al., *Contrib. Plasma Phys.*, preprint (2026) e70135

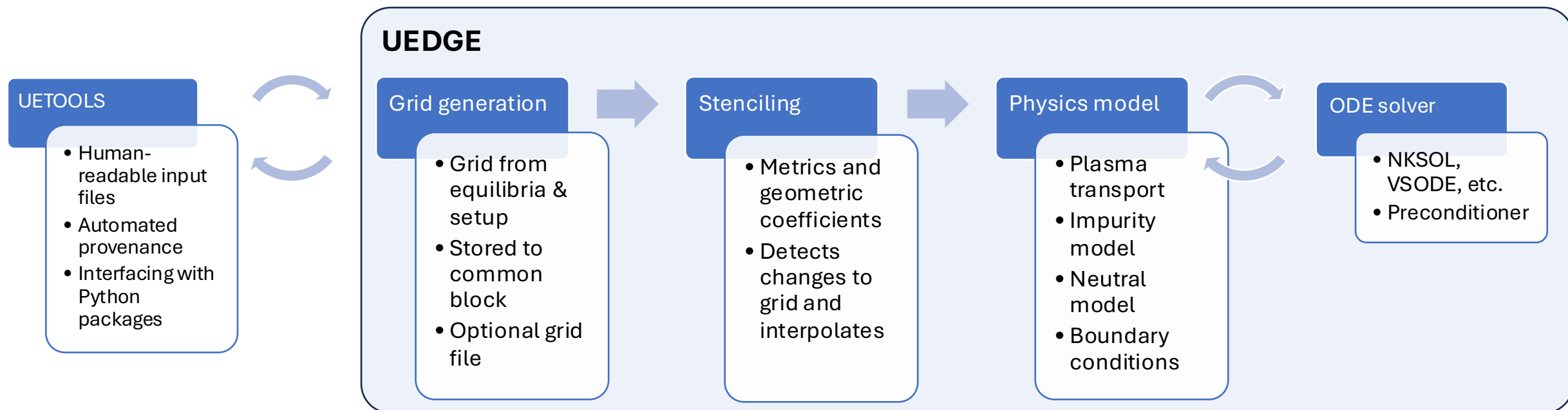
Development driven by tight code-experiment validation loops and applications to various geometries and devices

- Agile and adaptable code, responsive to experimental results
 - Routine validation with DIII-D, C-mod, and NSTX
 - Also used to model JET, JT60, KSTAR, ITER, K-DEMO, ARC, SPARC, FSNF, ...



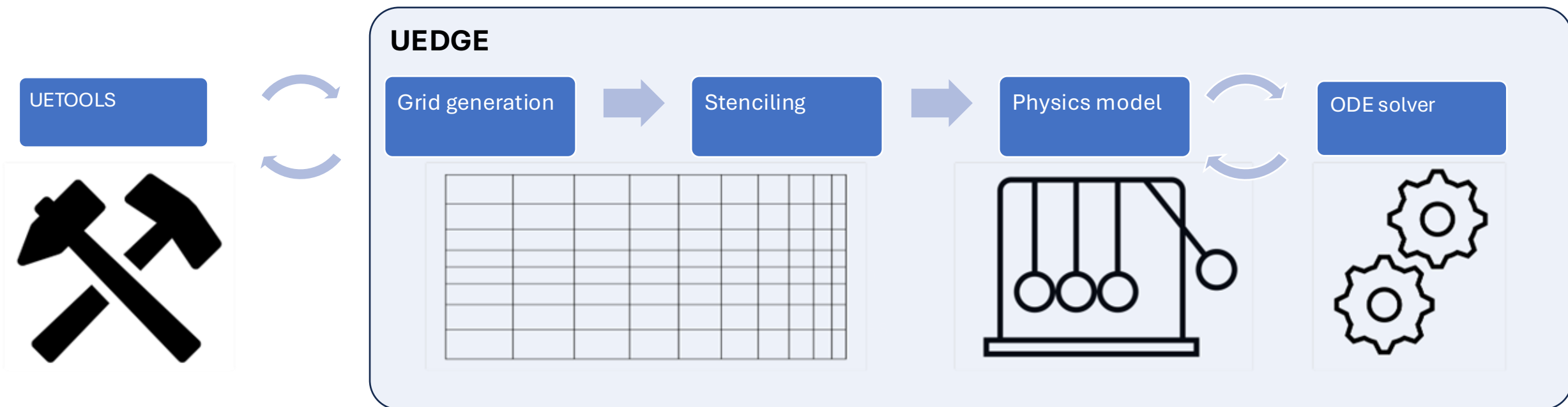
UEDGE has evolved into a tightly coupled end-to-end workflow tied together by common coupling

- Organic growth concentrated in a few subroutines
 - Convoluted coding with order-dependent execution
 - No entry points for unit testing



UEDGE has evolved into a tightly coupled end-to-end workflow tied together by common coupling

- Organic growth concentrated in a few subroutines
 - Convoluted coding with order-dependent execution
 - No entry points for unit testing



UETOOLS [19] provides a Python user interface that abstracts UEDGE-specific idiosyncrasies, lowering barriers to entry

- A natural API-like layer between UEDGE and downstream workflows
 - Bayesian optimization, multiprocessing, geometric packages, etc.

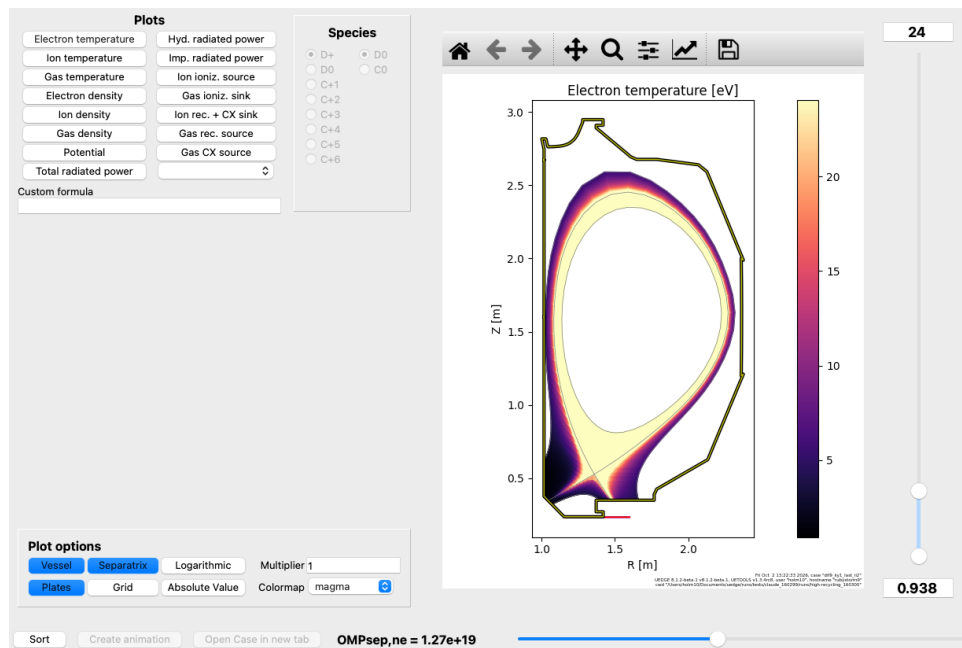
```
casename: '160300'
savefile: '160300.hdf5'
restart: 1

species:
  ngsp: 2 # Variable depends on options
  nhsp: 2 # Variable depends on options
  nhgsp: 1 # Variable depends on options
  isimpon: 6
  nzsp:
    0: 6 # Index depends on options

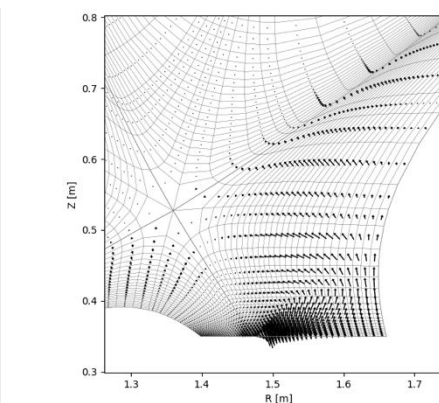
grid:
  mhdgeo: 1
  gengrid: 0
  geometry: "snul"
  isnonog: 1
  isgriduehdf5: 1
  GridFileName: "gridue.hdf5"

physics:
  minu:
    0: 2
  ziin:
    0: 1
```

Human-readable YAML input files



GUI dashboard

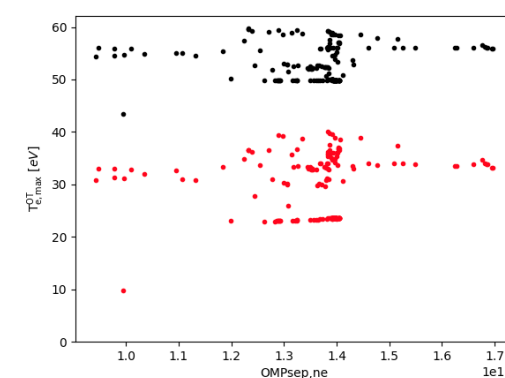


Plotting routines

```
In [4]: c.about.uedge_setup()
The UEDGE set-up contains:
- 2 hydrogenic species:
  - D ions
  - Inertial D atoms
- 1 impurity species:
  - Charge-state resolved C12
  - Fully force-balance

Ion array : [ D+, D0, C+1, C+2, C+3, C+4, C+5, C+6 ]
Gas array : [ D0, C0 ]
```

Setup diagnostics



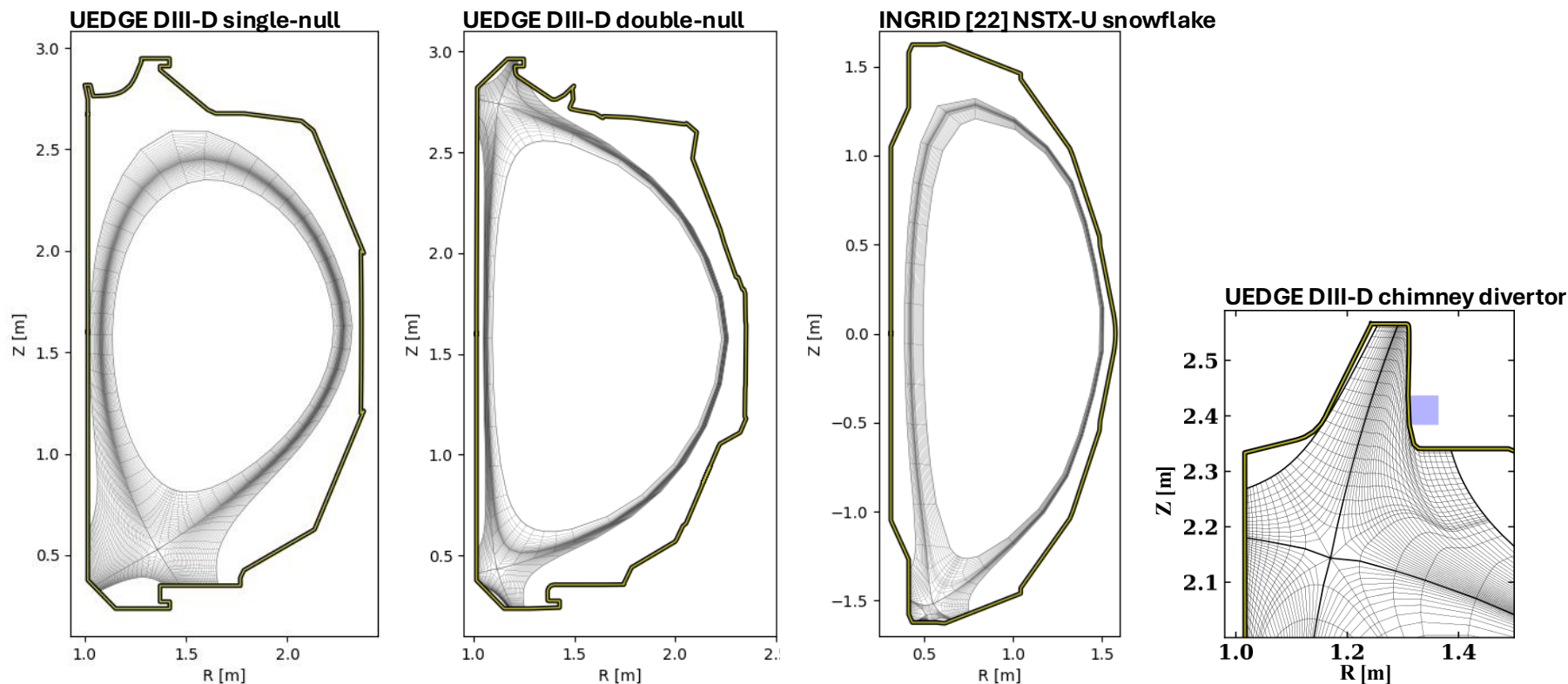
Parameter scan/database support

```
In [8]: c.tools.hdf5tree('iter11_case01_last_ii2.hdf5',
depth=1, group='setup/detected')
<HDF5 file "iter11_case01_last_ii2.hdf5" (mode r)>
setup/detected
├─ albedo_by_user (int)
├─ albedo1 (94, 2)
├─ albedo2 (94, 2)
├─ alblb (52, 6, 2)
├─ albrb (52, 6, 2)
```

Provenance-embedded and self-describing save files

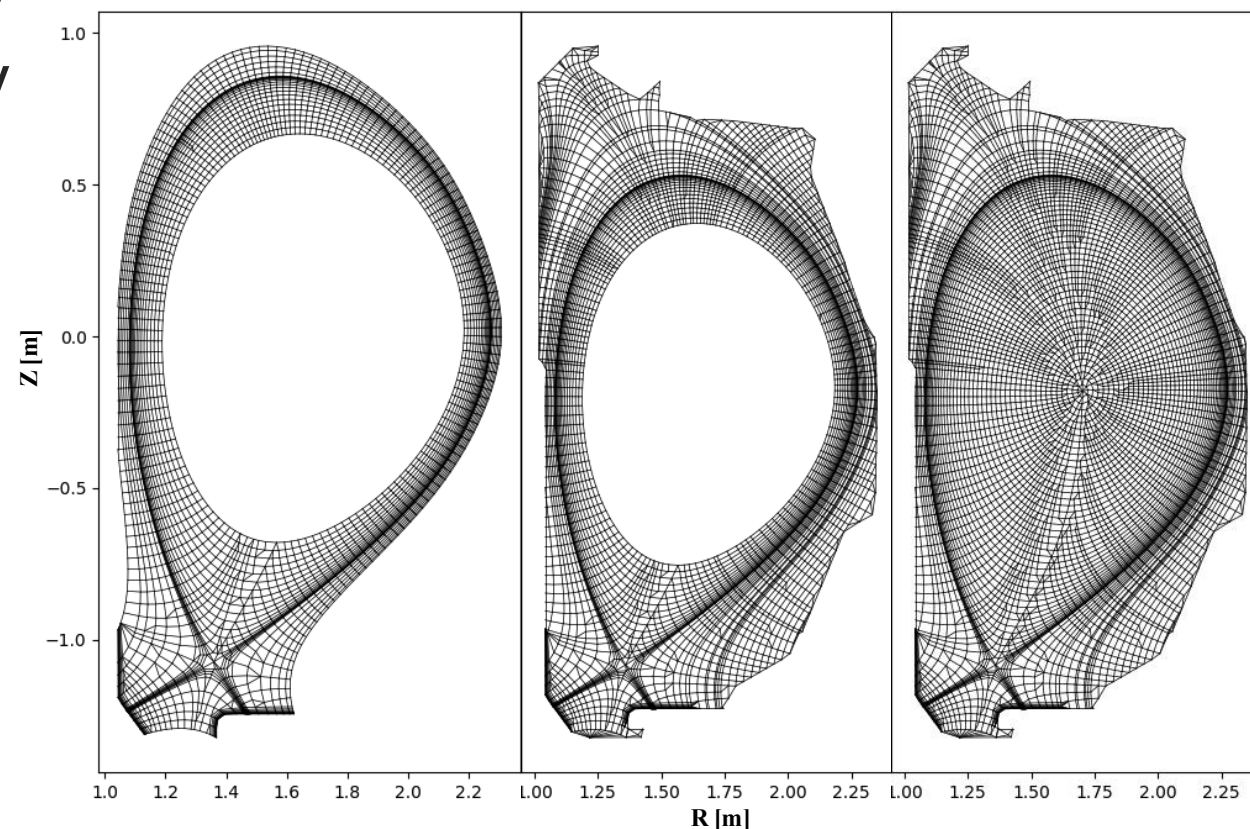
Envisioned refactoring applies separation of concerns, building on the natural interface grid files already provide

- Separating the physics modules requires deconvolution of variables
- Physics solver & stencil assume structured, flux-surface aligned grids



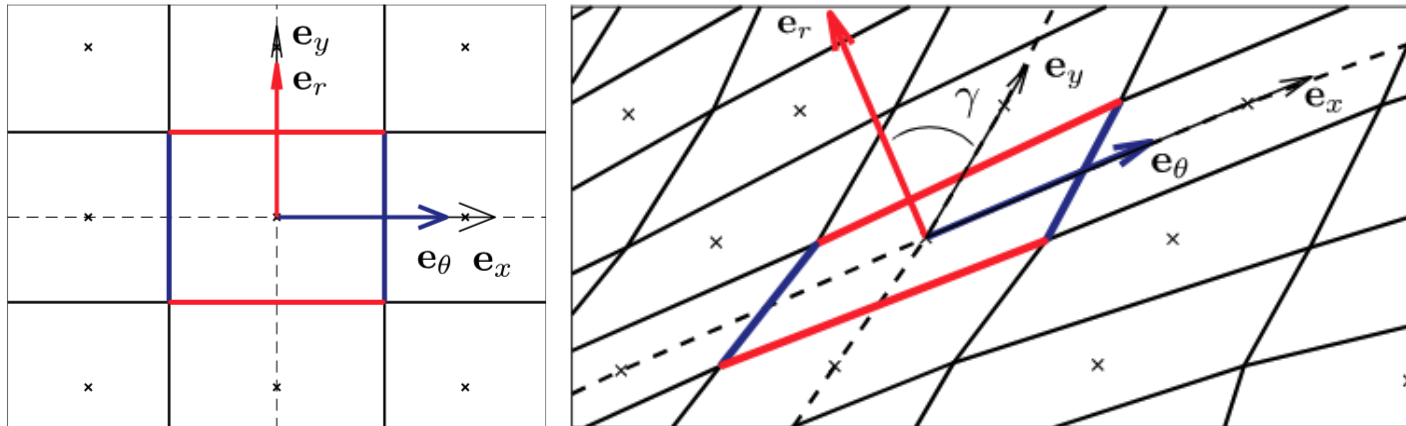
Potential for extended grids using GOAT [23] investigated as part of LDRD project: constrained by UEDGE architecture

- Significant refactoring necessary for unstructured mesh capability
 - B2-style distributed loops
 - Staggered mesh
 - Boundary physics model need to be implemented
- Structured grids with cut-cell approaches?
- Limited by monolithic subroutine structure

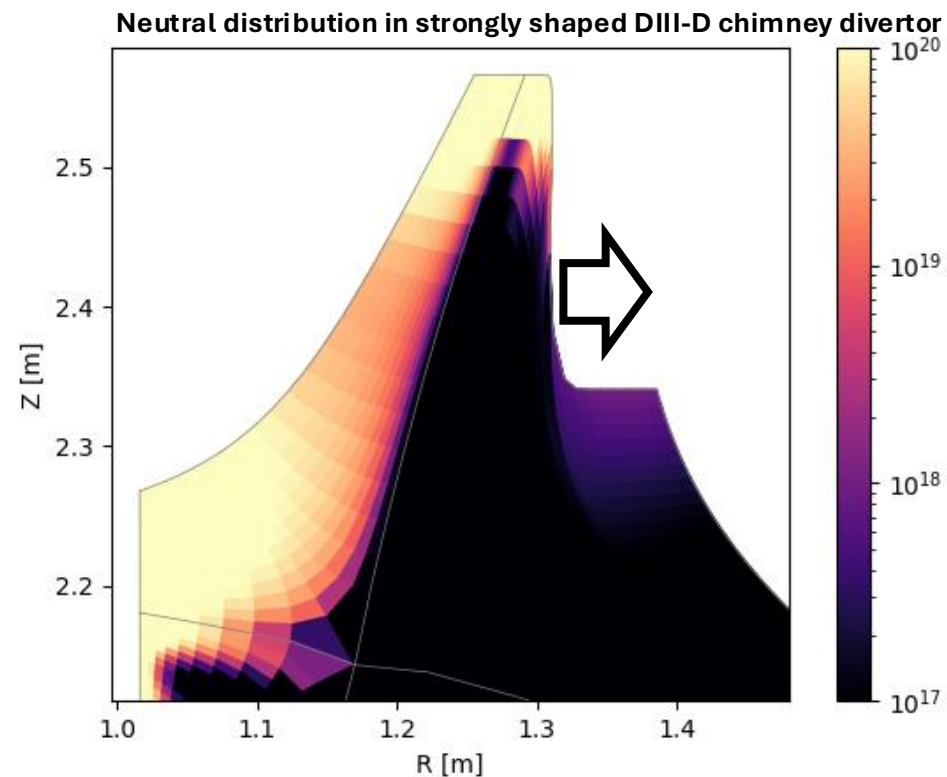


The fluid model of atoms solves a continuity, parallel momentum, and energy equation using a 9-point stencil

- 9-point stencil applied since its inception and alleviates discrepancies observed in early SOLPS simulations [24]



Orthogonal 5-point vs. nonorthogonal 9-point stenciling [26]

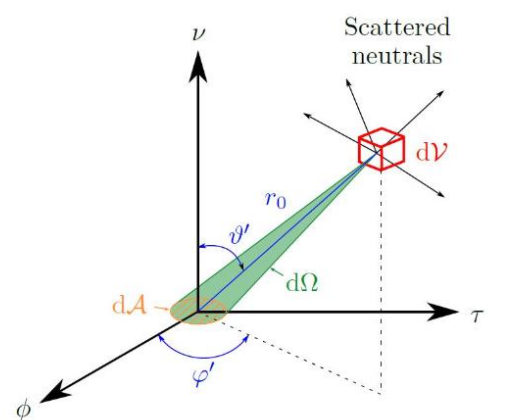


The fluid model of atoms solves a continuity, parallel momentum, and energy equation using a 9-point stencil

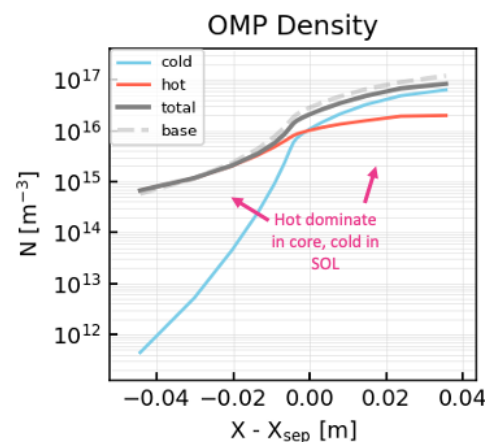
- 9-point stencil applied since its inception and alleviates discrepancies observed in early SOLPS simulations [24]
- Explicit coupling to MC neutrals exist but not routinely run
 - Implicit couplings through preconditioner displayed Δt -dependence
→ Implicit coupling through propagator-based approaches?
 - Containerized workflows implemented for routine post-processing

Internal LLNL development and research project for increasing neutral model fidelity funded 2025-2027

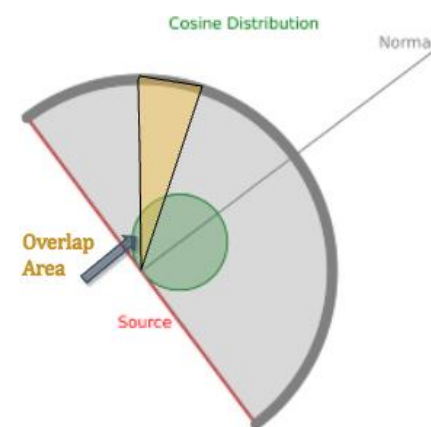
- Implement Advanced Fluid Neutral BCs developed by KU Leuven [25,26]
- Geometric, grid-free vacuum neutral model leveraging transfer matrix methods
- Potential two-temperature distribution models



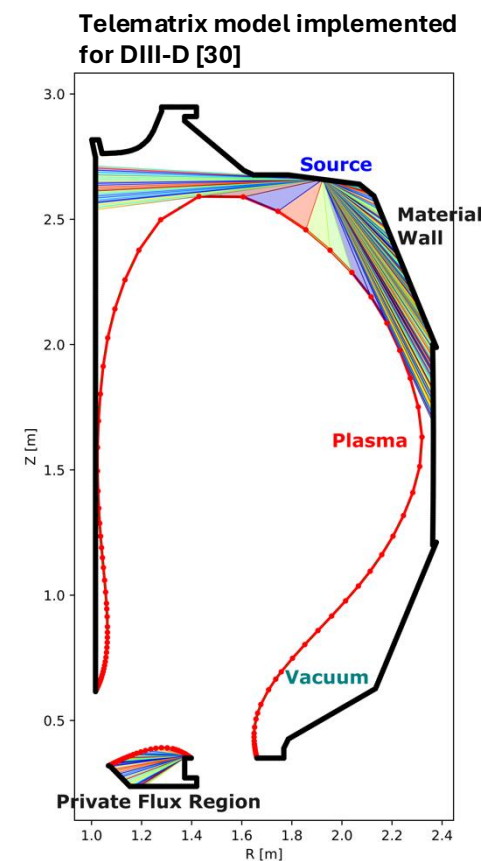
Scattering of neutrals in control volume for AFN boundary model [25,26]



OMP effect of Hot-Neutral Model (HNM) with two-temperature atoms [27]



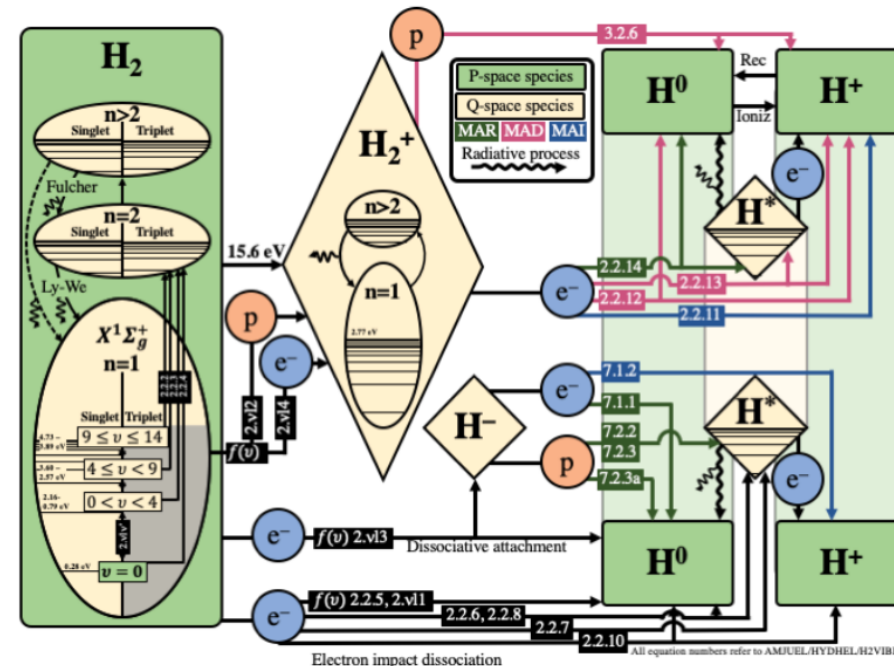
Schematic of angular coefficient method [28]



- [25] N. Horsten, et al., *Nucl. Fusion*, **57** (2017), 116043
 [26] W. Van Uytven et al., *Nucl. Fusion*, **62** (2022) 086023
 [27] M. Kryjak, et al., *EMU(S) Meeting* (2024)
 [28] J. Laura, et al., *67th Annual Meeting of the APS Division of Plasma Physics* (2025)

Effective Greenland collisional-radiative rates [29] are used in the UEDGE fluid molecular model [30,31]

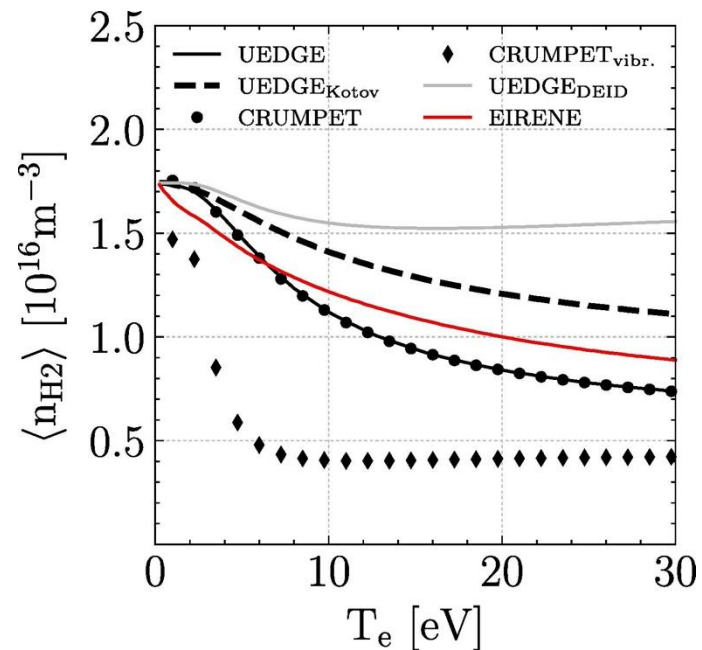
- Diffusive transport of H^+ , H^0 , H_2 modeled using CRUMPET rates [31,32]
 - Transient H_2^+ , H^- , H^* , H_2^{*+}
 - Effective $\langle\sigma v\rangle$ and $\langle\sigma v E\rangle$ rates calculated from fundamental data



- [29] P.T. Greenland, *Proc. Royal Soc. London A* **457** (2001) 1821
 [30] A. Holm, et.al., *Contrib. Plasma Phys.* **60** (2020) e201900150
 [31] A. Holm, et.al., *Nucl. Mater. Energy* **27**, (2021) 100982
 [32] A. Holm, *CRUMPET*, Computer Software, <https://github.com/holm10/CRUMPET>

Effective Greenland collisional-radiative rates [29] are used in the UEDGE fluid molecular model [30,31]

- Diffusive transport of H^+ , H^0 , H_2 modeled using CRUMPET rates [31,32]
- Differences in rates dominant over transport in comparisons to EIRENE



[29] P.T. Greenland, *Proc. Royal Soc. London A* **457** (2001) 1821

[30] A. Holm, et.al., *Contrib. Plasma Phys.*, **60** (2020) e201900150

[31] A. Holm, et al., *Nucl. Mater. Energy* **27**, (2021) 100982

[32] A. Holm, *CRUMPET*, Computer Software, <https://github.com/holm10/CRUMPET>

Effective Greenland collisional-radiative rates [29] are used in the UEDGE fluid molecular model [30,31]

- Diffusive transport of H^+ , H^0 , H_2 modeled using CRUMPET rates [31,32]
- Differences in rates dominant over transport in comparisons to EIRENE
- Kinetic transport effects and boundary conditions need to be evaluated (see W. van Uytven, J. Bryant this workshop)

[29] P.T. Greenland, *Proc. Royal Soc. London A* **457** (2001) 1821

[30] A. Holm, et.al., *Contrib. Plasma Phys.*, **60** (2020) e201900150

[31] A. Holm, et al., *Nucl. Mater. Energy* **27**, (2021) 100982

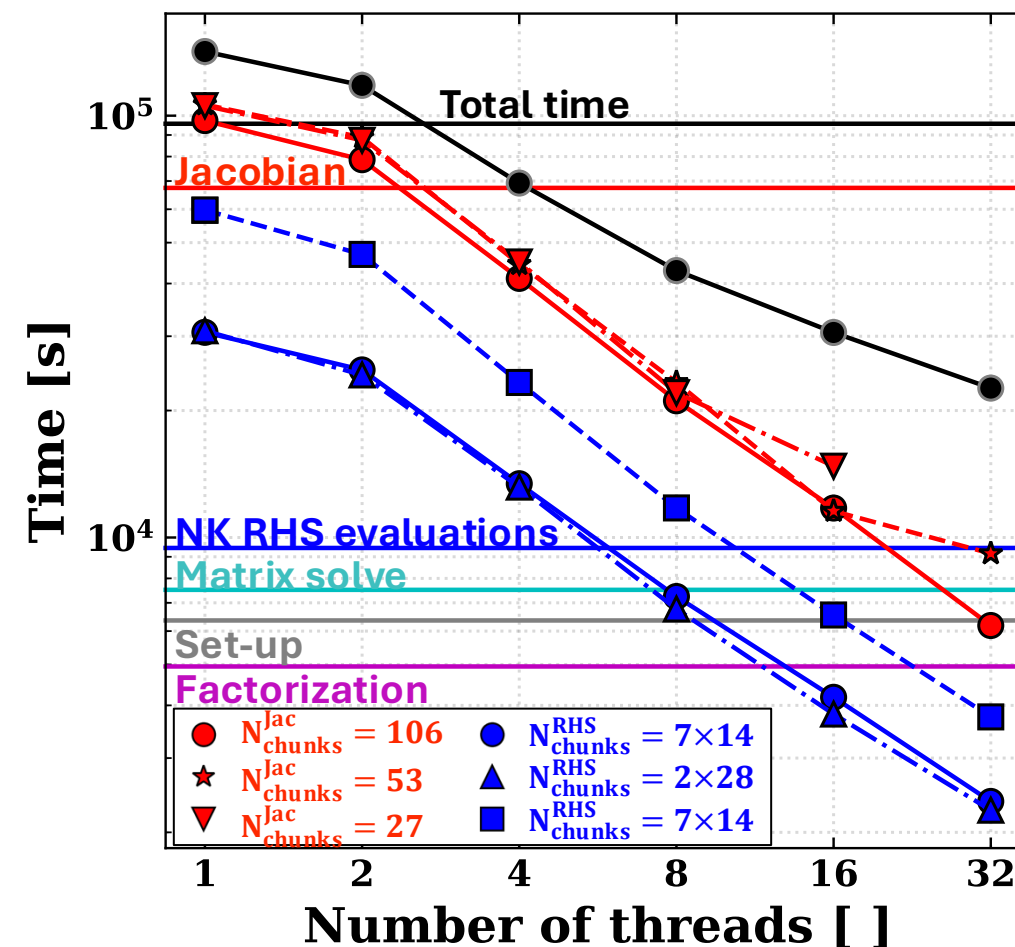
[32] A. Holm, *CRUMPET*, Computer Software, <https://github.com/holm10/CRUMPET>

UEDGE uses the bespoke, internal JFNK solver, NKSOL: later developed into the KINSOL solver of SUNDIALS

- Time-implicit NKSOL unconditionally stable wrt Δt owing to the noise-free fluid neutral models
 - Routine steady-state solutions including plasma drifts
- Performance dependent on ILUT preconditioner quality
 - RHS evaluator called w/ perturbed basis vectors to construct FD Jacobian
 - Computationally expensive Jacobian can be re-used in several iterations
- NKSOL interfaced to time-dependent and continuation solvers [21]
 - Time-dependent solver: first-order time-evolution to steady-state
 - Continuation solver: steady-state parameter scans

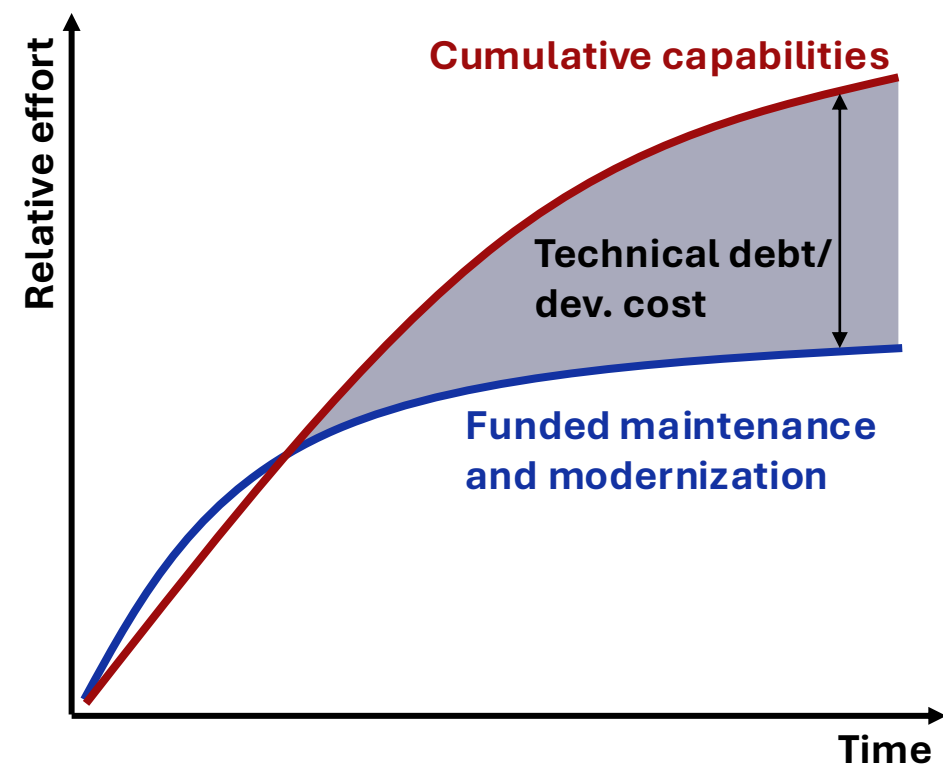
OpenMP parallelization of the preconditioner and RHS evaluation provides 80% speedup vs. serial version [21]

- Preconditioner construction embarrassingly parallelizable
- Shared memory parallelization complicated by common coupling and structured meshes
 - MPI-like implementation for RHS physics routines
- Higher-order terms and staggered velocity mesh increases stencil size



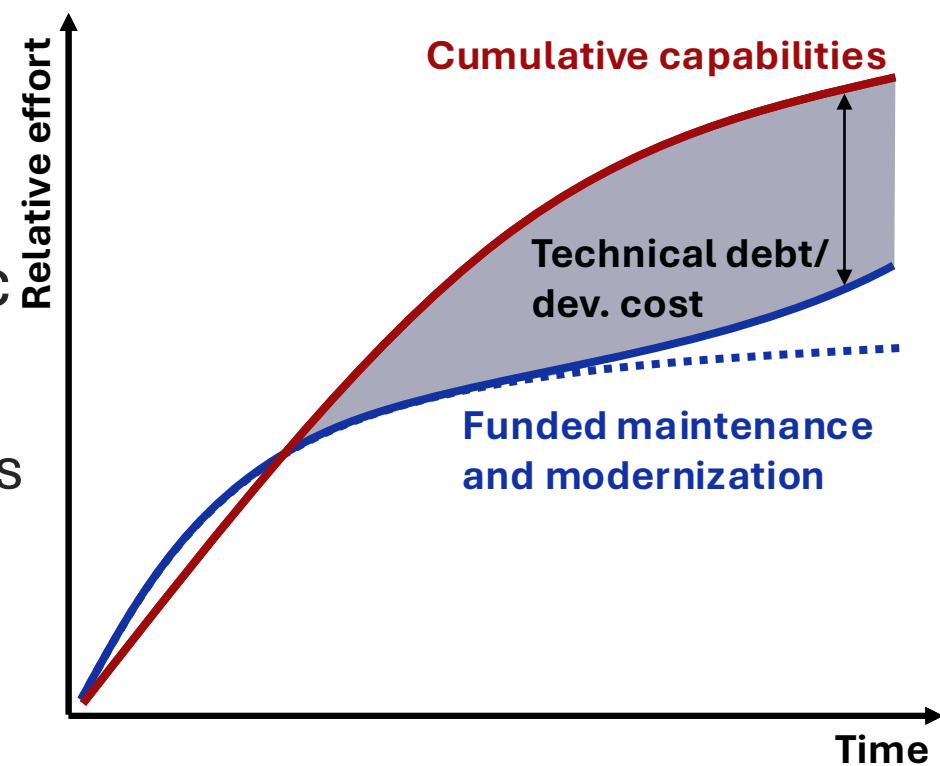
Continuous development has long outpaced funded maintenance: refactoring needed to meet modern standards

- Maintenance and modernization opportunities rarely funded:
 - Perpetual competition with science output



Continuous development has long outpaced funded maintenance: refactoring needed to meet modern standards

- Maintenance and modernization opportunities rarely funded:
 - Perpetual competition with science output
 - Google.org code modernization and agentic workflow award (2-year at ~50% coverage)
 - Create API-like layers for JAXification of routines
 - Automated test coverage and refactoring using AI/LLM-assisted tooling
- SOLPS-ITER refactoring by KU Leuven group amounted to multi-year full-team effort



UEDGE is scientifically mature but technically strained following decades of validated development

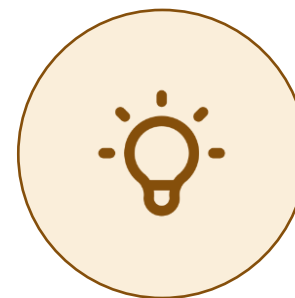
- Development remains active, continually extending physics models
- UEDGE's track record and a modernized codebase are not in tension
- With the advent of AI/LLM tooling a practical path to addressing technological debt is available



Refactor



Test



Comprehend

Auspices and acknowledgements

This work was supported by US DOE under contract no. DE-AC52-07NA27344.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

JFNK solver equations

Newton step

$$J(x_k) s_k = -F(x_k), \quad x_{k+1} = x_k + s_k$$

Jacobian-free directional derivative

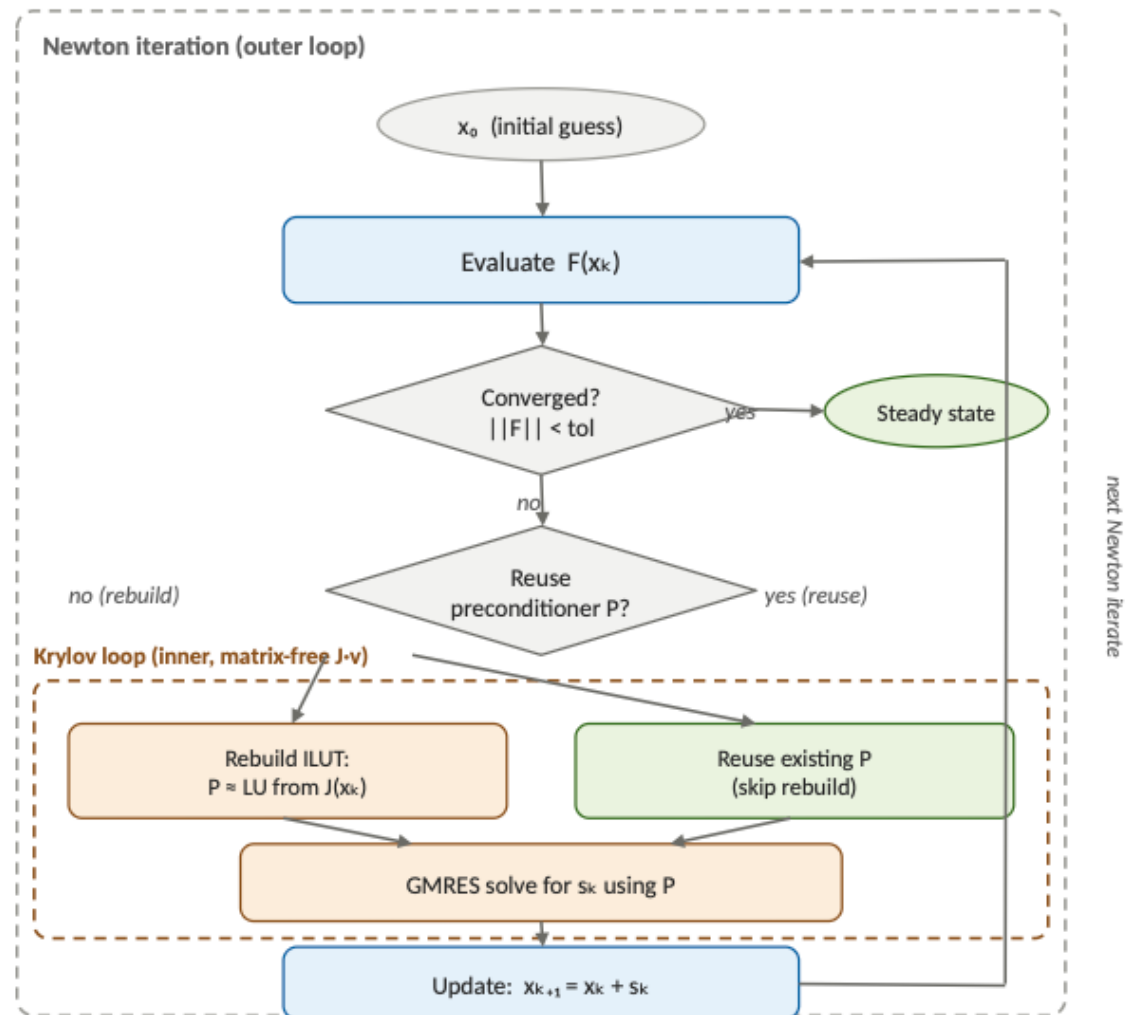
$$J(x_k) v \approx \frac{F(x_k + \epsilon v) - F(x_k)}{\epsilon}$$

Preconditioned Krylov (GMRES) solve

$$(J(x_k) P^{-1}) y = -F(x_k), \quad s_k = P^{-1} y$$

ILUT preconditioner

$$P \approx LU \approx J(x_k) \quad (\text{incomplete LU, threshold } \tau, \text{ fill } p)$$



next Newton iterate

Atom model equations

•

$$\frac{\partial}{\partial t} n_a + \nabla \cdot (n_a \bar{v}_a) = \langle \sigma_i v_{te} \rangle n_e n_a - \langle \sigma_r v_{te} \rangle n_e n_i$$

$$\begin{aligned} \frac{\partial}{\partial t} m_a n_a v_{a,\parallel} + \nabla \cdot (m_a n_a v_{a,\parallel} \bar{v}_a - \boldsymbol{\eta}_a \nabla v_{a,\parallel}) &= \frac{B_\theta}{B} \left(-\frac{\partial P_a}{\partial \theta} \right) + m_i n_i v_{cx} (v_{i,\parallel} - v_{a,\parallel}) - m_a n_a v_i v_{a,\parallel} + m_i n_i v_r v_{i,\parallel} \\ \frac{\partial}{\partial t} \left(\frac{3}{2} n_a T_a \right) + \nabla \cdot \left(\frac{5}{2} n_a T_a \bar{v}_a - \kappa_a \nabla T_a \right) + \boldsymbol{\Pi}_a : \nabla \bar{v}_a &= \bar{v}_a \cdot \nabla P_a + Q_a + S_a^E + S_a^m \cdot \bar{v}_a + \frac{1}{2} m_a v_a^2 S_a \end{aligned}$$

Molecular model equations

$$\frac{\partial}{\partial t} n_m + \nabla \cdot (n_m \bar{v}_m) = -n_m v_{\text{diss}},$$

$$\frac{\partial}{\partial t} (m_m n_m \bar{v}_m) + \nabla \cdot (m_m n_m \bar{v}_m \bar{v}_m - \eta_m \nabla \bar{v}_m) = -\nabla P_m - m_m n_m v_{\text{diss}} \bar{v}_m + m_m n_m [v_{el,mi}(\bar{v}_i - \bar{v}_m) + v_{el,ma}(\bar{v}_a - \bar{v}_m)],$$

$$\begin{aligned} \frac{\partial}{\partial t} \left(\frac{3}{2} n_m T_m \right) + \nabla \cdot \left(\frac{5}{2} m_m n_m \bar{v}_m T_m - \kappa_m \nabla T_m \right) = & \bar{v}_m \cdot \nabla P_m + \eta_m (\nabla \cdot \bar{v}_m)^2 \\ & - \frac{3}{2} n_m v_{eqp} (T_m - T_i) - n_m v_{\text{diss}} \left(\frac{1}{2} m_m \bar{v}_m^2 + \frac{3}{2} T_m \right). \end{aligned}$$