

Usability-Driven Progress in Multiphysics Tools

FY26 WPMs: Alexander Lindsay (INL), Robert Lefebvre (ORNL), Emily Shemon (ANL),
Cody Permann (INL)



U.S. DEPARTMENT
of **ENERGY**

Office of
Nuclear Energy

Session Overview

- MOOSE Framework Enhancements – Cody Permann
- GPU acceleration efforts – Alex Lindsay
- CSG efforts in MOOSE – Kalin Kiesling
- User Oriented Enhancements in Workbench – Rob Lefebvre

- Related Topics covered in other sessions:
 - AI developments (MOOSEngier) – Mengnan Li, yesterday during the working lunch
 - Stochastic Tools, Surrogates, and Optimization – Zach Prince, today @11:30AM MT



Convergence System

- A one-size fits all convergence criteria has issues:
 - Absolute residual tolerance difficult to link to solution accuracy
 - Relative residual convergence is dependent on the initial guess
 - L-2 norm of residual is biased on mesh refinement
 - Larger residuals for one variable may dominate the others
- The **Convergence System** allows users to set customized criteria:
 - Nonlinear systems
 - Fixed point systems
 - Steady-state detection in transients
- Benefits:
 - Avoid pitfalls of default criteria
 - Consider other sources of error in the system to avoid over-solving
 - Allow more detailed convergence troubleshooting
- **PostprocessorConvergence** compares Postprocessor value to tolerance.
- **ParsedConvergence** allows user-defined convergence/divergence expressions containing Convergence objects, Postprocessors, and Functions.
- **FunctorChangeFunctorMaterial** computes the change of any functor over iterations or time steps, useful for step criteria.
- **DiscreteVariableResidualNorm** computes a residual norm for a variable, across any subdomain, without mesh bias.



MOOSE Framework Enhancements – cont.

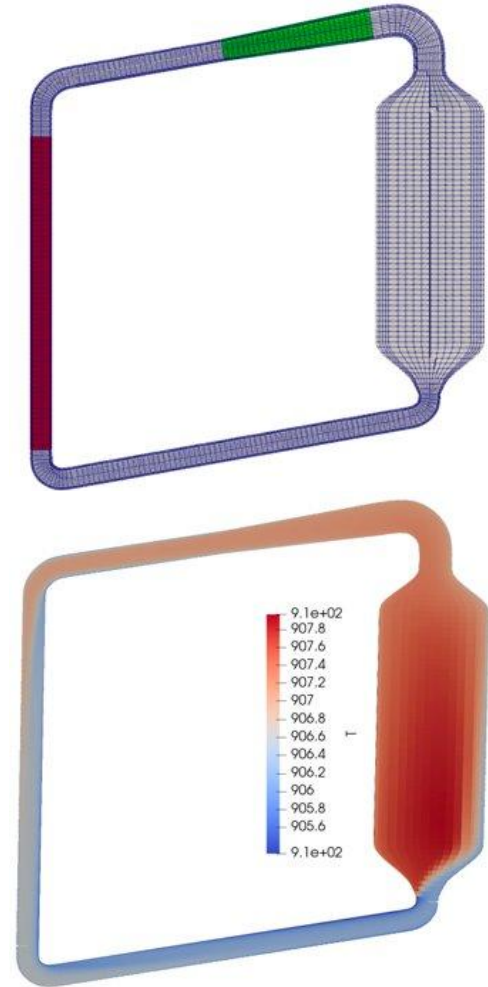
- Physics and Components workflow
- Traditional MOOSE syntax involves:
 - kernels, boundary conditions & helper materials to represent the equation
 - numerical scheme setup through the preconditioner and executioner
- The abstraction can be simplified with:
 - Physics to describe the equations and set default numerical settings
 - (Action)Components to describe the geometry
- The ideal Physics input should be minimal while remaining flexible
- Simplifying the abstraction should help us with:
 - Reducing user support needs & facilitating input verification
 - Modeling complex systems with minimal input
 - Reducing maintenance costs of setup with all Physics following the same code paths

```
[ActionComponents]
[bottom_pipe]
  type = CylinderComponent
  dimension = 3
  direction = '-.17365 .98481 0'
  position = '-.45581 0.122896 0'
  length = 1.60
  n_axial = ${fparse int(length/dx)}

  radius = .0525
  boundary_layer_width = ${fparse radius * boundary_layer_fraction}
[]
[hx_cylinder]
  type = CylinderComponent
  dimension = 3
  direction = '1 0 0'
  position = '-.604 1.877 0'
  length = 1.6
  n_axial = ${fparse int(length/dx)}

  radius = .0525
  boundary_layer_width = ${fparse radius * boundary_layer_fraction}
[]
[junction_top_left]
  type = JunctionComponent
  first_component = bottom_pipe
  second_component = hx_cylinder
  first_boundary = 'bottom_pipe_top_boundary'
  second_boundary = 'hx_cylinder_bottom_boundary'
  n_elem_normal = ${n_junction_axial}
[]
[pipe_next_to_pump]
  type = CylinderComponent
  dimension = 3
  length = 0.705
  radius = 0.0525
  n_axial = ${fparse int(length/dx)}

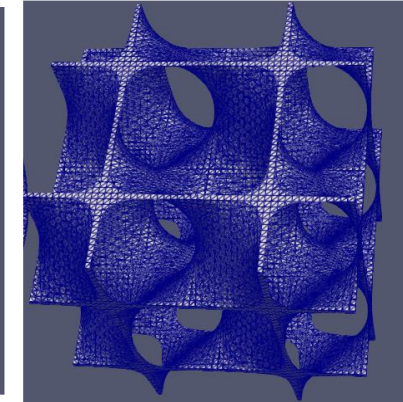
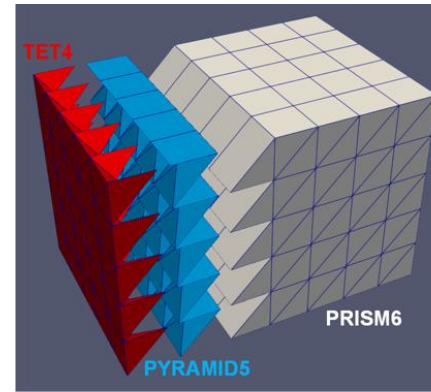
  direction = '-.17365 .98481 0'
  position = '1.31051 1 0'
  boundary_layer_width = ${fparse radius * boundary_layer_fraction}
[]
...
```



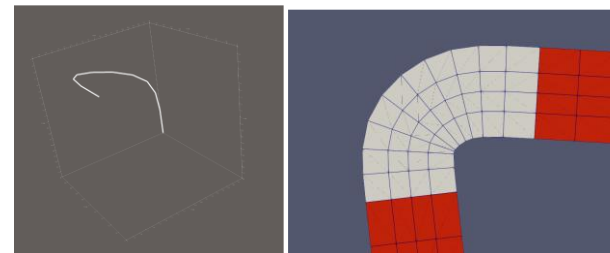
Meshing Usability: Improvements to the Reactor Module

MOOSE's Reactor Module empowers users to build increasingly complex reactor geometries, with new features added in FY25:

- 3D tetrahedral mesher (XYZDelaunayGenerator) including surface meshing, transition layer, and cutting utilities
- Generalized 2D triangulation
- Multi-D extruded mesh junctions
- Batch mesh generation: succinct input creates array of mesh objects with fixed & variable parameters
- MOOSE-to-Constructive Solid Geometry work to support Monte Carlo workflows – (e.g. Griffin/Shift calculations) – *Kalin will discuss further*
- User workshop hosted in May 2025 on INL HPC (w/ updated [online tutorial](#))

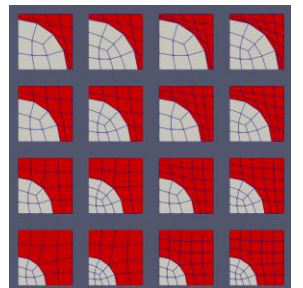
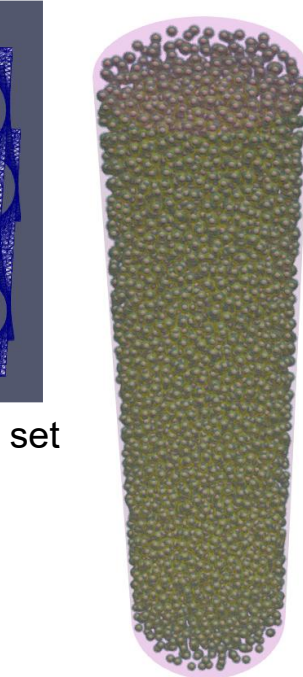


3D tet meshing (a) transition layer utility, (b) level set cutting utility to generate a triply periodic minimal surface, (c) full-size TRISO compact mesh (<100 GB, <1 day)



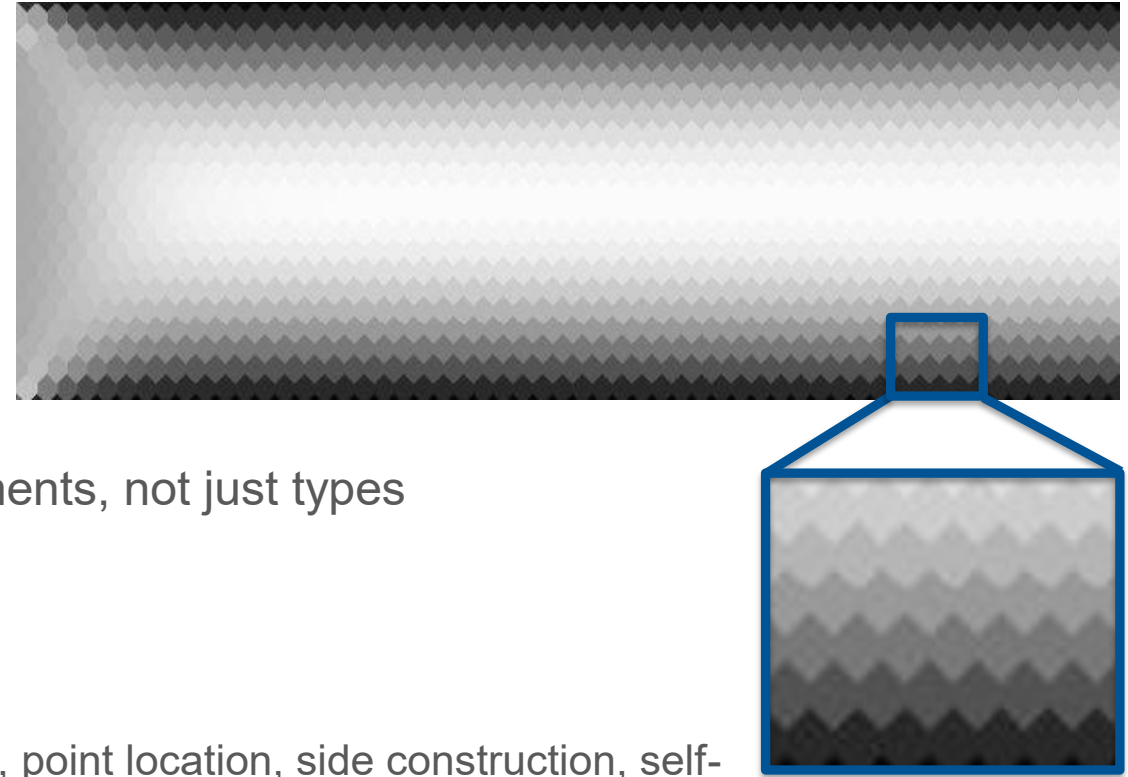
Pipe junction using extrusion along spline curve

Batch mesh generation with one mesh block



MOOSE Framework Enhancements – cont.

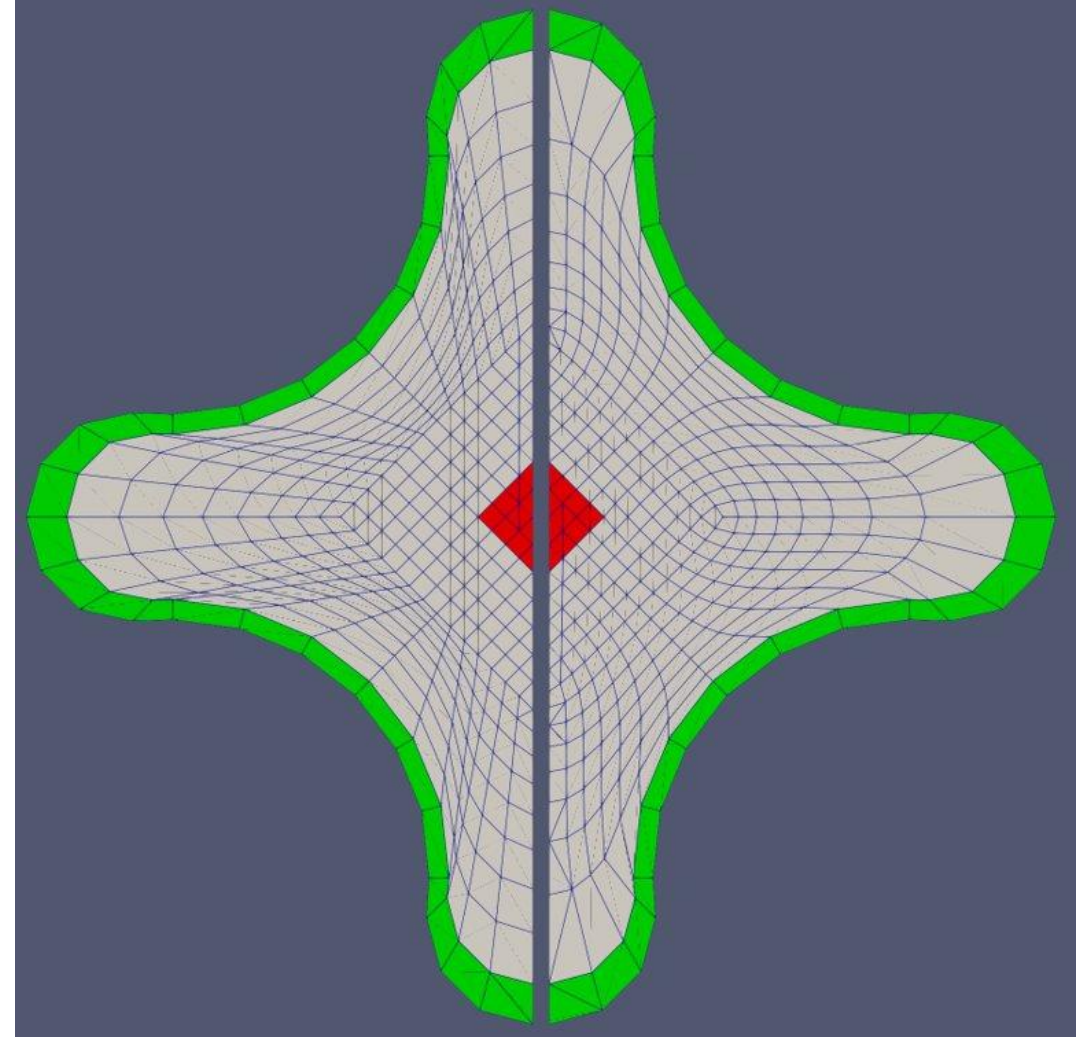
- Polyhedral mesh support
 - Popular among Finite Volume users
 - Requested at International MOOSE Workshop
 - Major libMesh refactoring required:
 - FE and Quadrature initialization requires specific elements, not just types
 - Added C0Polygon, C0Polyhedron element classes
 - Each passing dozens of standard Elem unit tests
 - Geometry: bounding boxes, quality metrics, permutations, point location, side construction, self-consistency
 - FE: values, derivatives, partition-of-unity, projection accuracy in Monomial+XYZ+Lagrange spaces



Variational Mesh Smoother

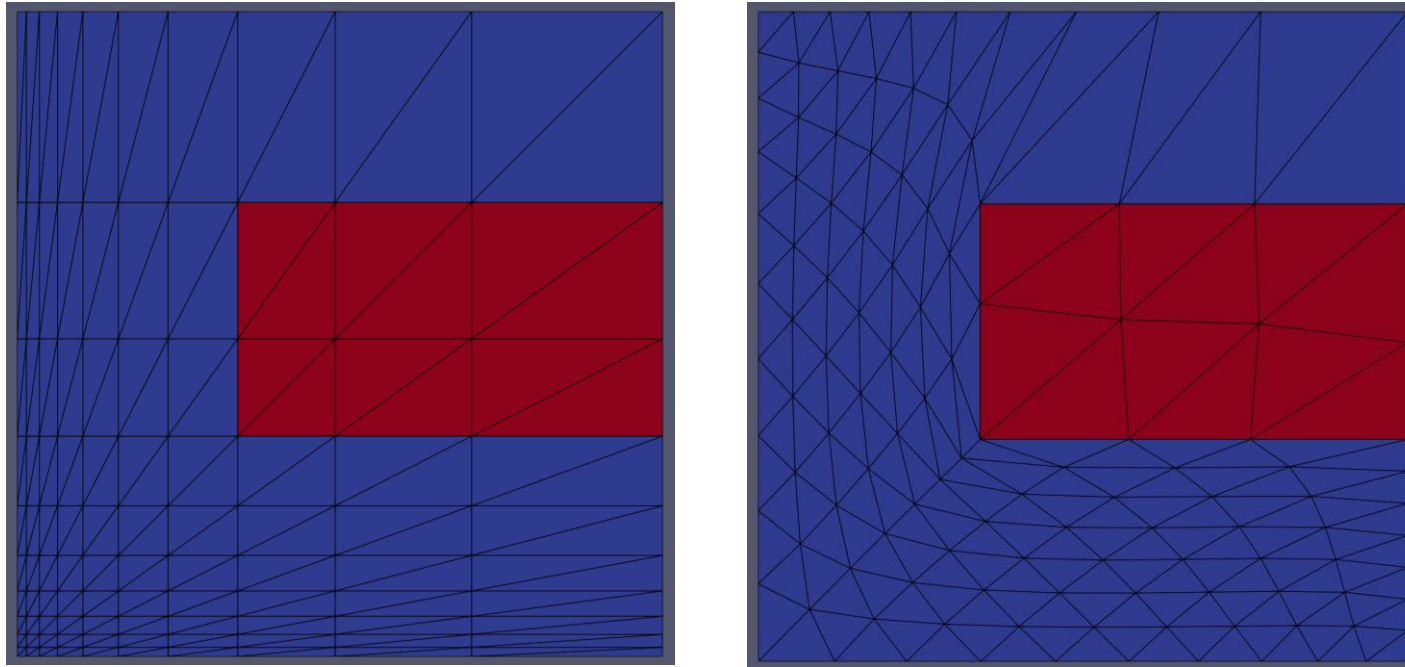
- Motivation
 - Mesh smoothing improves mesh quality
 - Mesh quality affects convergence rates of numerical solves
- Approach
 - Solve variational problem that minimizes element-based distortion-dilation metric [1]
 - Implemented in libMesh, available through MOOSE
- Current Capabilities
 - Smooths 1/2/3D meshes
 - Node sliding allowed on linear external and block boundaries
 - Preserves block and total volumes
 - Can untangle some tangled meshes

[1] Branets, Larisa Vladimirovna. A variational grid optimization method based on a local cell quality metric. The University of Texas at Austin, 2005.



SmoothMeshGenerator

Test Mesh before and after smoothing



```
[Mesh]
[gm]
  type = GeneratedMeshGenerator
  dim = 2
  nx = 10
  ny = 10
  bias_x = 1.4
  bias_y = 1.4
  elem_type = TRI3
[]

[add_block]
  type = ParsedSubdomainMeshGenerator
  input = gm
  combinatorial_geometry = 'x > 0.35 & y
                           > 0.35 & y < 0.7'
  block_id = 1
[]

[smooth]
  type = SmoothMeshGenerator
  input = add_block
  algorithm = variational
[]
```

New "algorithm" parameter to select between variational (new) and Laplace (old) smoothing.

Documentation: <https://mooseframework.inl.gov/source/meshgenerators/SmoothMeshGenerator.html>

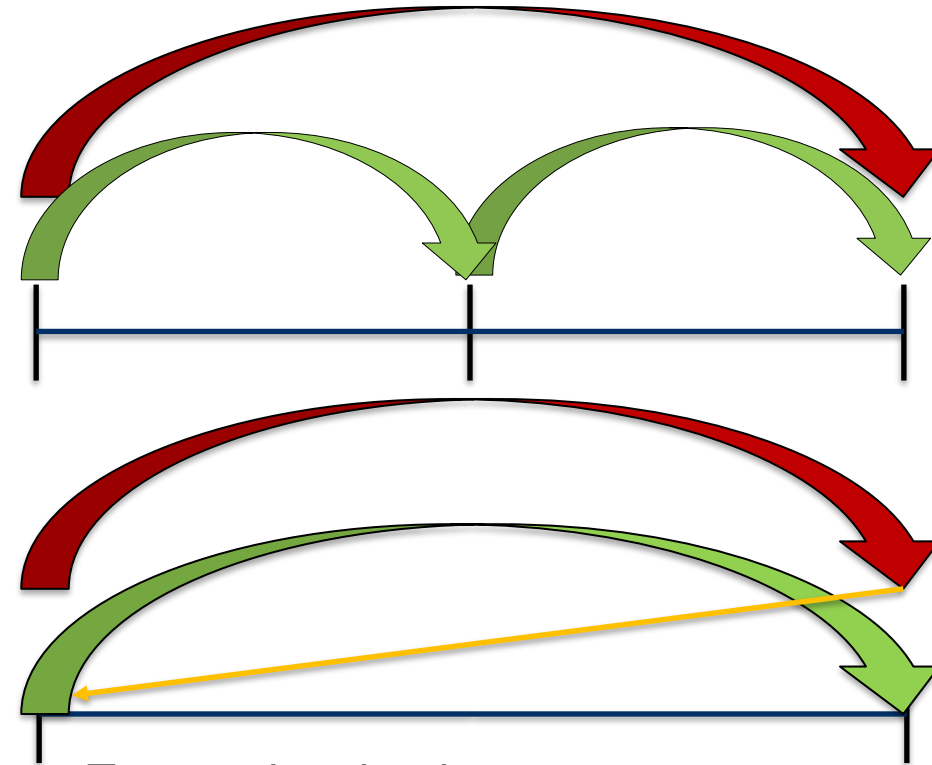
MOOSE Framework Enhancements - Restep

- Failed solves during Transient simulations are typically resolved by repeating timesteps with smaller size (top-right)
- Ensuring restoration of stateful data is vital to ensure expected behavior
- “Restep” testing added to MOOSE test infrastructure and CI/CD

```
$ ./run_tests --restep <other test options>
```

```
$ <application>-(opt|devel|dbg) -i input.i --test-restep
```

- Global enhancements developed from testing deployment:
 - Restoration of Postprocessor, VectorPP, and Reporter data
 - Proper updating of displaced meshes with adaptivity
 - Correct use of old dt in BDF2 time integrator
- Restep testing inherently available to downstream applications

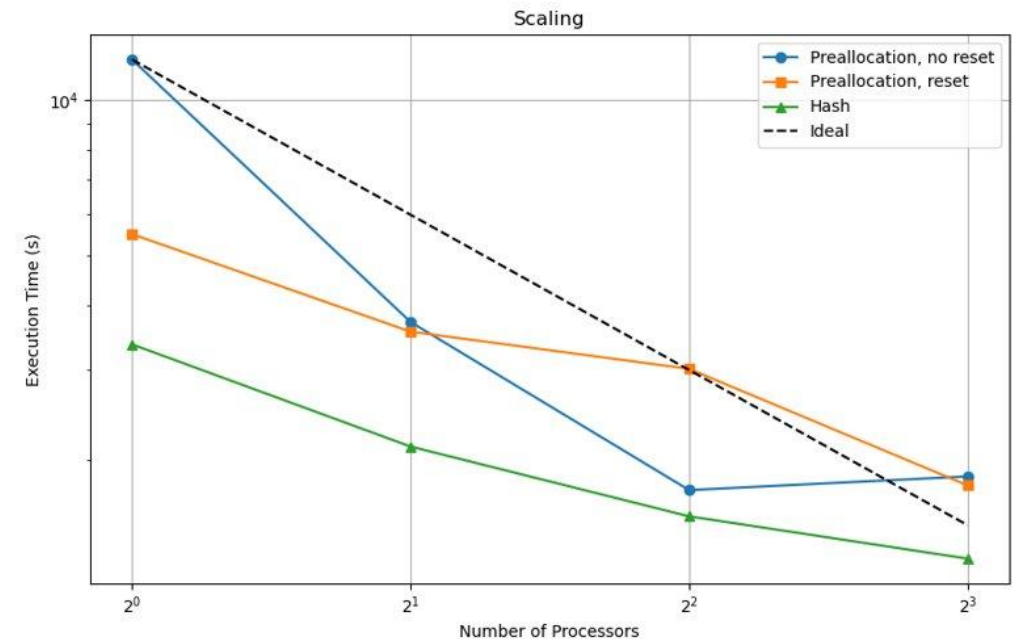


Top: cutting the time step to get around a failed step

Bottom: testing the reset procedure by re-taking the exact same step after a reset

MOOSE Framework Enhancements – cont.

- Hash table matrix assembly
- It can be difficult to predict sparsity pattern for things like fuel performance with deforming mesh constantly changing contact couplings
- Users are now able to specify `Problem/prefer_hash_table_matrix_assembly=true` in order to use a hash table-based approach which does not require guessing an initial sparsity pattern
- For one BISON assessment case, this reduced the initial Jacobian assembly time from 2 hours to 50 seconds



Technical Area Support

- Easy use of Controls for turning an object on for a single time step for BISON
- Silencing thousands of linking warnings due to compact unwinding on macOS
- Block restricted Exodus output for BWXT
- Mesh displacement & contact miscellaneous improvements for BISON/AFC
- IAPWS97 water fluid properties improvement for SAM
- Warning table summary for BWXT (not merged yet)

Warnings History:

Object	Step	Interval Count	Total Count
WarningUserObject : initial: warning	0-1	4	4
WarningUserObject : tbegin: warning	0-1	2	2
WarningUserObject : tbegin: warning	1-2	2	4
WarningUserObject : tbegin: warning	2-3	2	6
WarningUserObject : jacobian: warning	0-1	2	2
WarningUserObject : jacobian: warning	1-2	2	4
WarningUserObject : residual: warning	0-1	2	2
WarningUserObject : residual: warning	1-2	6	8
WarningUserObject : residual: warning	2-3	2	10
WarningUserObject : tend: warning	0-1	2	2
WarningUserObject : tend: warning	1-2	2	4
WarningUserObject : tend: warning	2-3	2	6
WarningUserObject : final: warning	0-1	2	2

Left: example warning summary table
 Right: IAPWS97 fluid regions (from IAPWS97 paper)

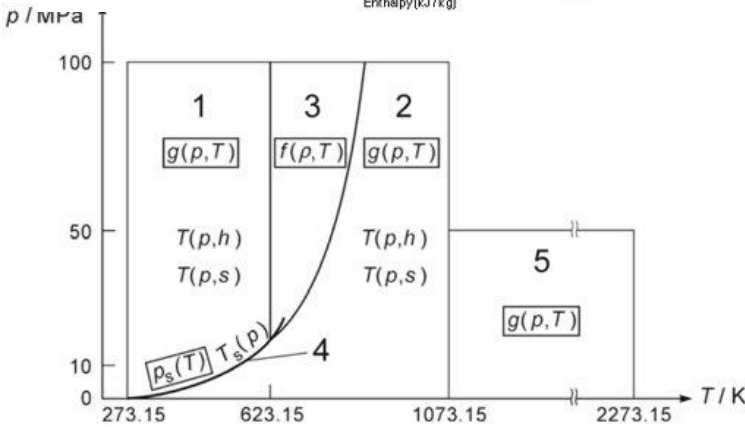
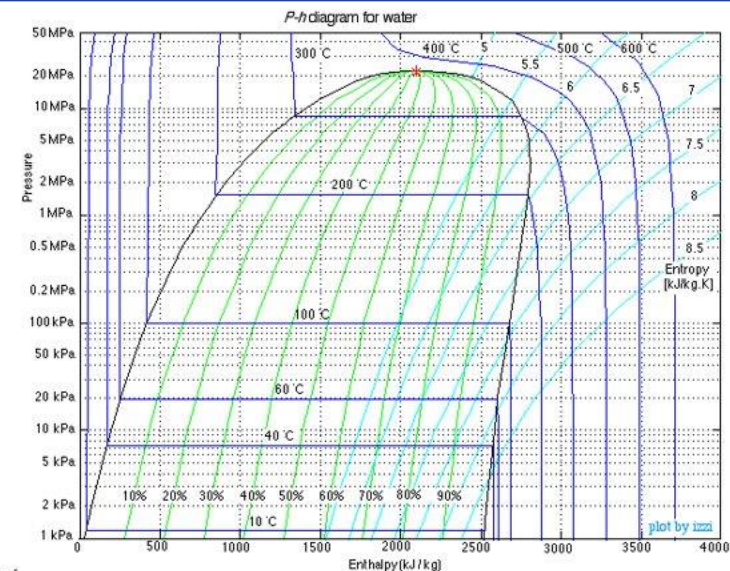
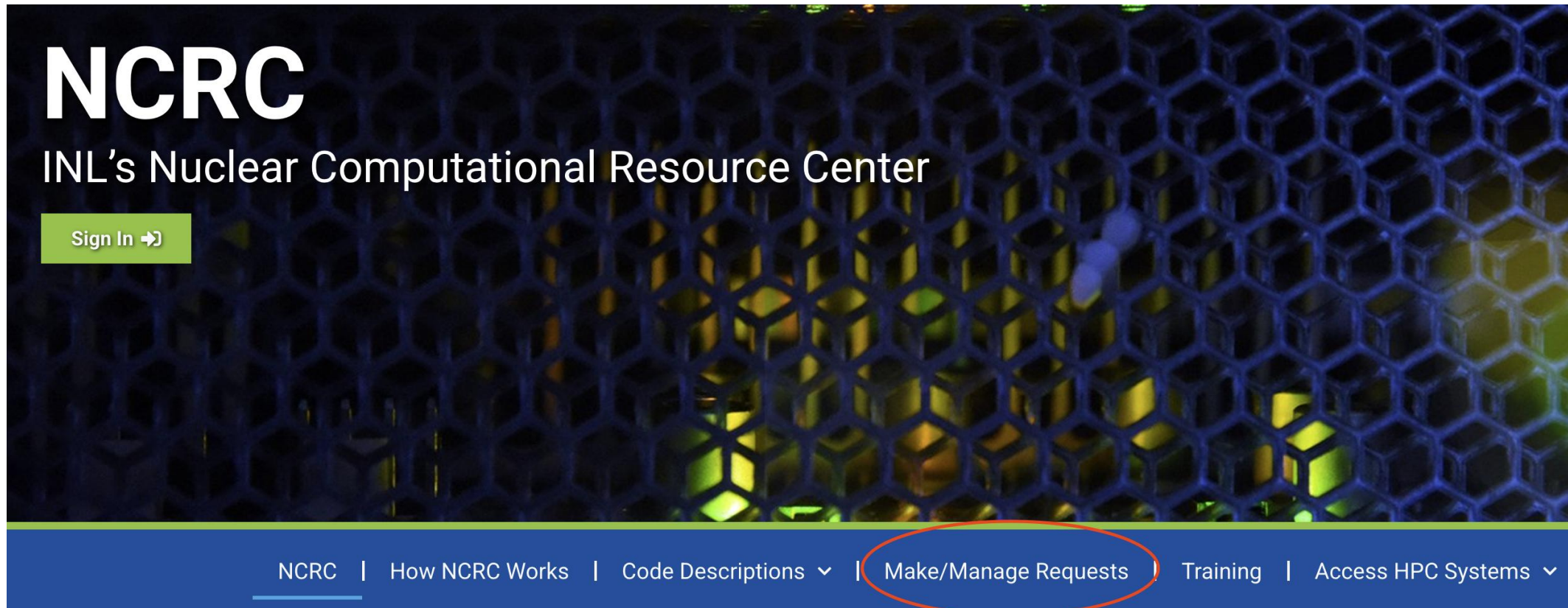


Fig. 1. Regions and equations of IAPWS-IF97.

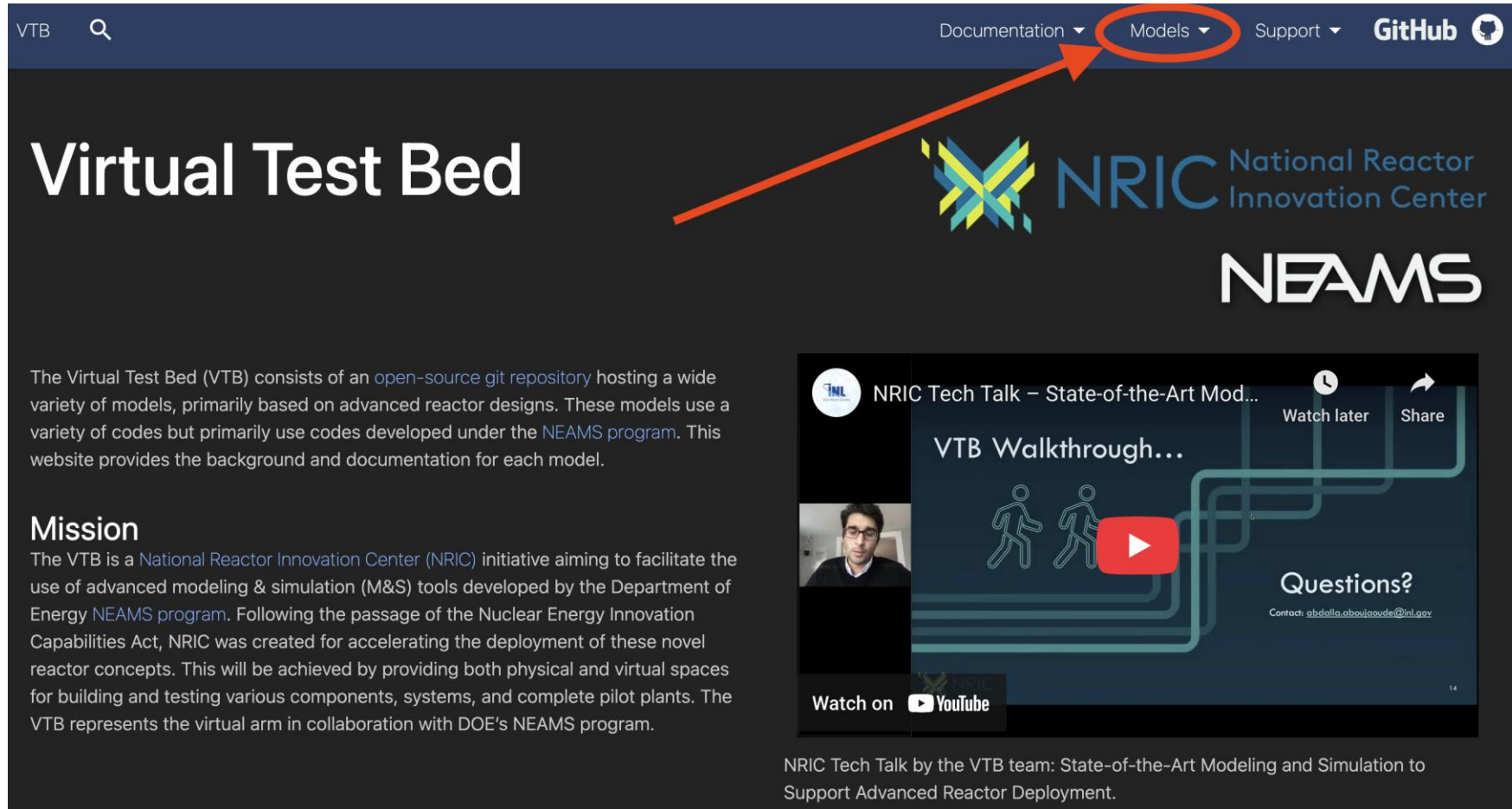
Accessing Codes, Compute Resources, and Models

- Apply for access to NEAMS codes and computing resources at inl.gov/ncrc:



Accessing Codes, Compute Resources, and Models (cont.)

mooseframework.inl.gov/virtual_test_bed



The screenshot shows the Virtual Test Bed website. The top navigation bar includes 'VTB', a search icon, 'Documentation', 'Models' (highlighted with a red circle and an arrow), 'Support', and a 'GitHub' link. The main header features the 'NRIC National Reactor Innovation Center' and 'NEAMS' logos. The main content area has a title 'Virtual Test Bed' and a paragraph describing the VTB as an open-source git repository. Below this is a 'Mission' section. On the right, there is a video player for 'NRIC Tech Talk – State-of-the-Art Mod...' with a 'Watch later' button and a 'Share' button. The video player also shows a 'VTB Walkthrough...' title and a 'Questions?' section with contact information.

VTB

Documentation

Models

Support

GitHub

Virtual Test Bed

The Virtual Test Bed (VTB) consists of an [open-source git repository](#) hosting a wide variety of models, primarily based on advanced reactor designs. These models use a variety of codes but primarily use codes developed under the [NEAMS program](#). This website provides the background and documentation for each model.

Mission

The VTB is a [National Reactor Innovation Center \(NRIC\)](#) initiative aiming to facilitate the use of advanced modeling & simulation (M&S) tools developed by the Department of Energy [NEAMS program](#). Following the passage of the Nuclear Energy Innovation Capabilities Act, NRIC was created for accelerating the deployment of these novel reactor concepts. This will be achieved by providing both physical and virtual spaces for building and testing various components, systems, and complete pilot plants. The VTB represents the virtual arm in collaboration with DOE's NEAMS program.

NRIC Tech Talk – State-of-the-Art Mod...

Watch later

Share

VTB Walkthrough...

Questions?

Contact: abdalla.aboujaoude@inl.gov

Watch on YouTube

NRIC Tech Talk by the VTB team: State-of-the-Art Modeling and Simulation to Support Advanced Reactor Deployment.



U.S. DEPARTMENT
of **ENERGY**

Office of
Nuclear Energy