OAK RIDGE National Laboratory

March 19, 2026

Leveraging Trilinos scientific computing library for computational fluid dynamics applications

Marco Delchini

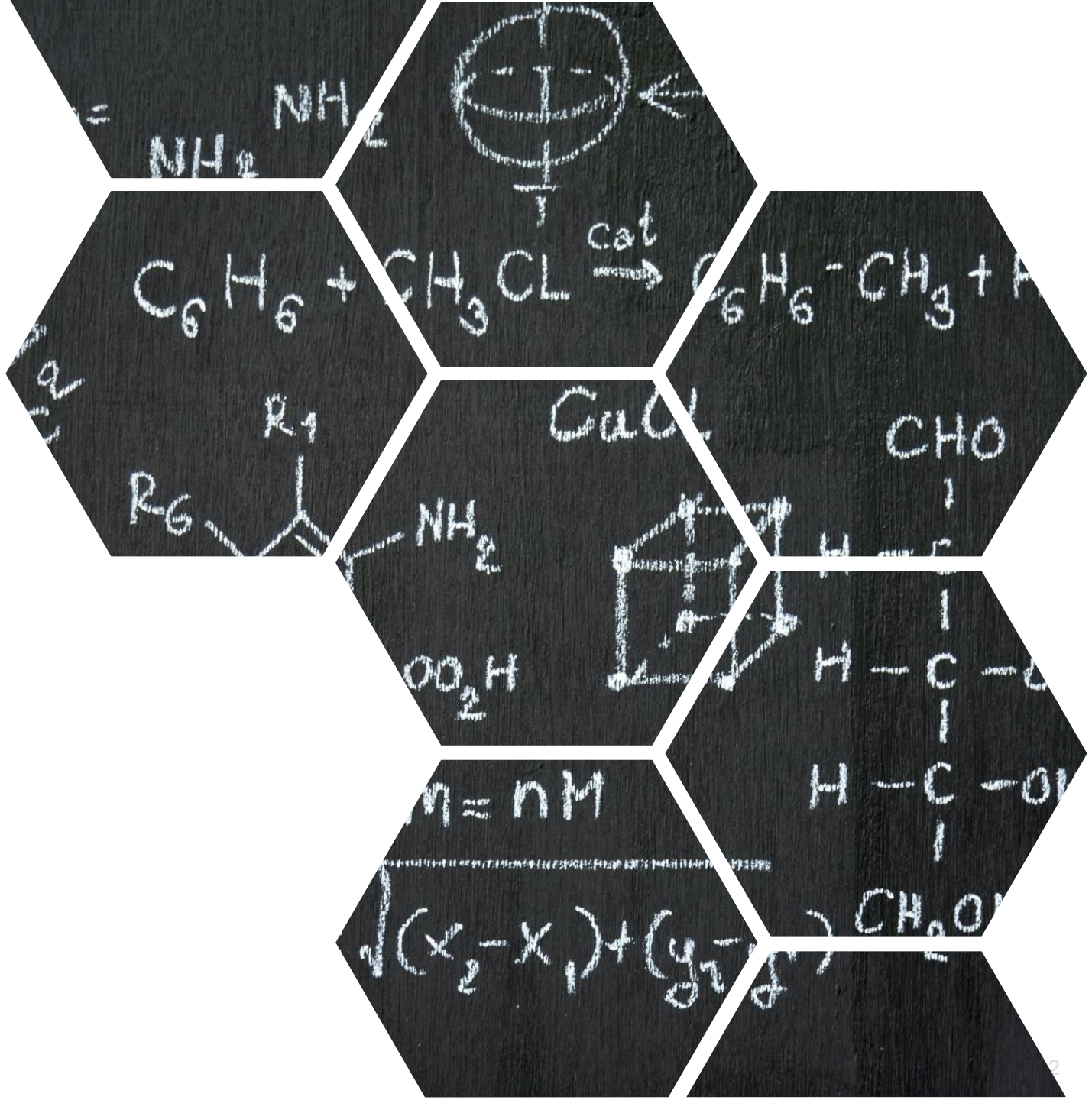
Oak Ridge National Laboratory

U.S. DEPARTMENT of ENERGY

ORNL IS MANAGED BY UT-BATTELLE LLC
FOR THE US DEPARTMENT OF ENERGY

Outline

1. Overview of projects using Trilinos library: ASINA and VERTEX
2. What is our current approach to solving PDEs?
3. What are we currently working on?
4. Wishlist and conclusions

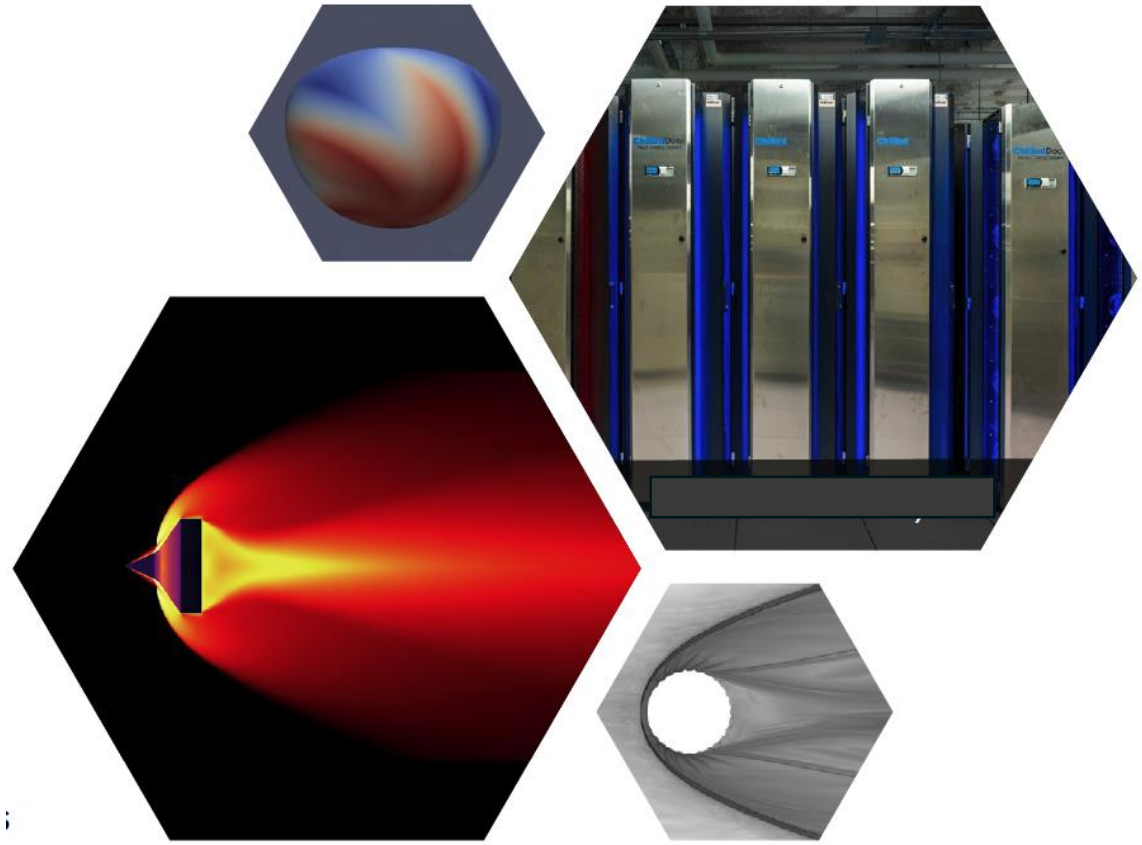


Overview of projects using Trilinos library: ASINA and VERTEX

Advance Simulation Initiatives for Nonproliferation Applications Project (ASINA)

Application problems and objectives:

- Develop modeling capability for high-speed flows.
- Generate high-fidelity physics-based modeling capabilities to further the understanding of high-speed flows.
- Establish a science and technology infrastructure to enable high-performance modeling and simulation.
- Develop the next generation of interdisciplinary scientists and engineers.



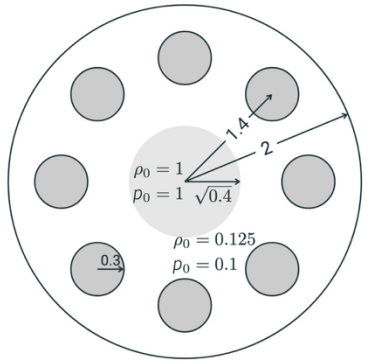
Modeling of inviscid flows with ASINA

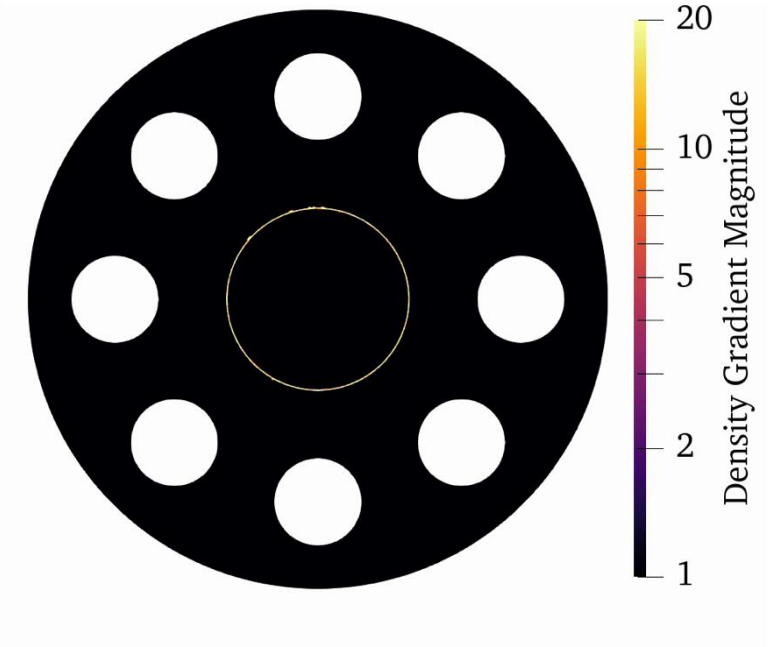
2D Pressure Waves in a Domain with Cylinders

- **Domain**: Outer cylinder with radius of 2 with eight smaller cylinders with a radius of 0.3
- **Boundary conditions**: Free-slip
- **Mesh edge lengths**: 0.01 and 0.005

v_0	ρ_0	p_0	Time
0	0.125	0.1	5

v_{shade}	ρ_{shade}	p_{shade}
0	1	1

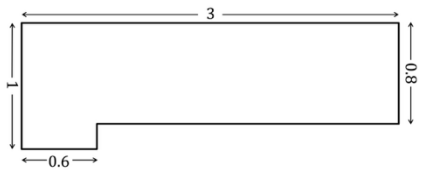


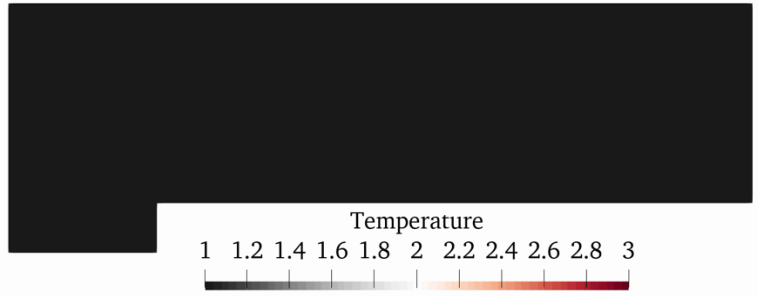


Supersonic Inviscid Flow in 2D Forward Step

- **Domain**: 3×1 with a 20% vertical step located 20% from the inlet
  - Step corner angle is $90°$
- **Boundary conditions**: Dirichlet (left inlet), supersonic outflow (right outlet), free-slip (elsewhere)
- **Mesh edge lengths**: 0.01, 0.005, 0.0025, and 0.00125

ρ_∞	M_∞	T_∞	Time
1.4	3	1	4

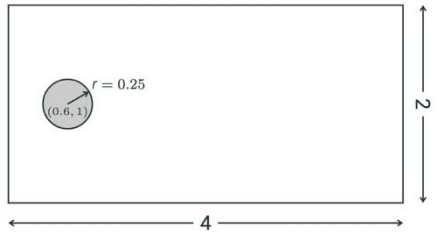




Inviscid Supersonic flow over 2D cylinder

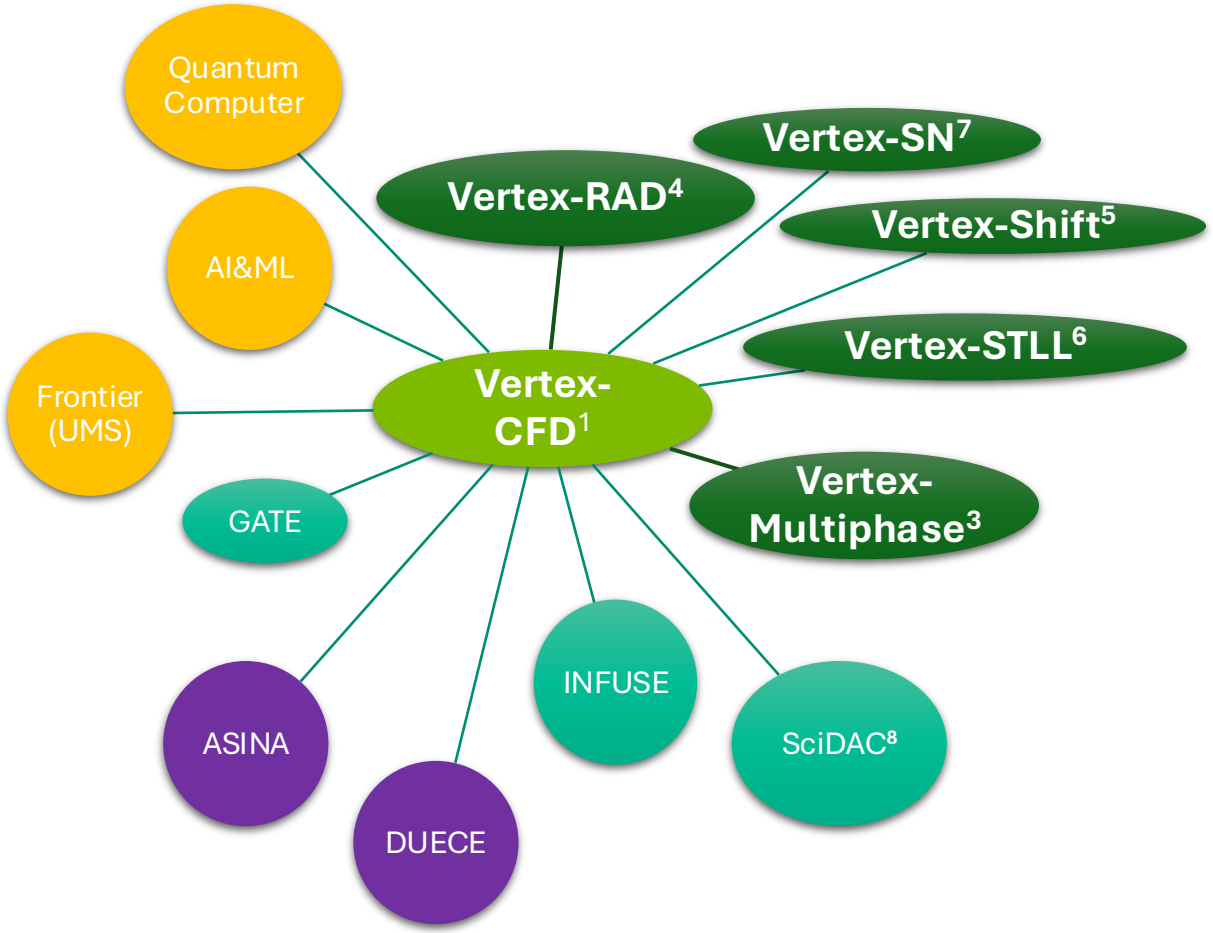
- **Domain**: 4×2 with a cylinder of radius 0.25 centered at $(0.6, 1)$
- **Boundary conditions**: Dirichlet (left inlet), supersonic outflow (right outlet), free-slip (elsewhere)
- **Mesh edge lengths**: 0.01, 0.005, and 0.0025

ρ_∞	M_∞	T_∞	Time
1.4	3	1	5



K. Doetsch et al. High-Fidelity, Low-Dissipation/Symmetry-Preserving Numerical Scheme for Solving the Euler Equations with Unstructured, Metric-Based Mesh Adaptation, AIAA 2026-0765.

VERTEX: a multiphysics platform



[3]**PI:** Arpan Sircar
[4]**PI:** Will Gurecky
[5]**PI:** Steven Hamilton
[6]**PI:** Joy Fan
[7]**PI:** Lance Bullerwell
[8]**PI:** Sergey Smolentsev

Vertex Initiative (PI Tim Younkin[2] - ESED):

- ORNL developed M&S package for HPC platforms.
- Started in FY 2021 and will end in FY 2027.
- https://code.ornl.gov/vertex/vertex-cfd

Application problems:

- *Fusion blanket problems:* Vertex-Multiphase and SciDac.
- *Molten salt reactors:* Vertex-SN, -RAD, and -Shift.
- *Centriguge modeling:* DUECE and ASINA.
- *Reduced-order modeling of liquid metal blanket:* Vertex-STLL.

Collaborations:

- **ORNL divisions:** FED, CSED, CSMD, NCCS and ESED.
- **DOE labs:** LANL and LLNL.
- **Universities:** UTK (GATE and LDRD) and UGA.

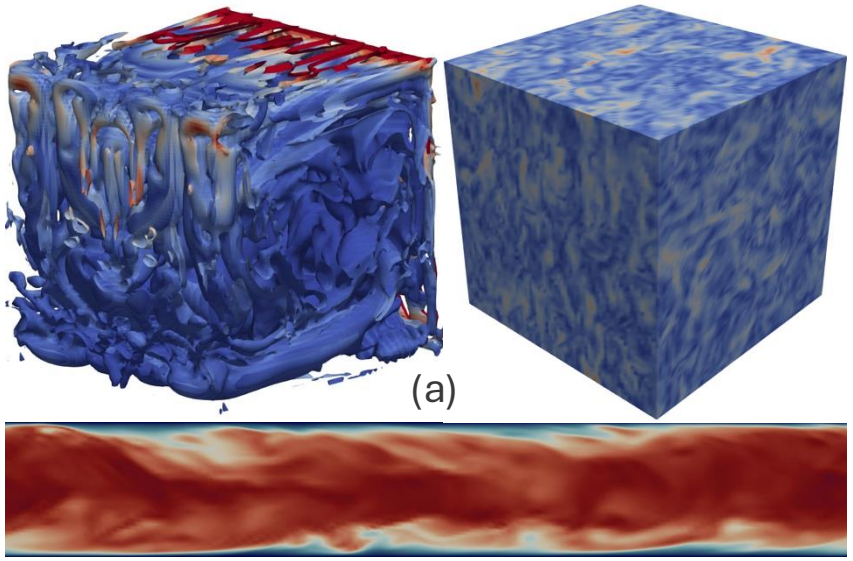
Funding programs for Vertex-CFD:

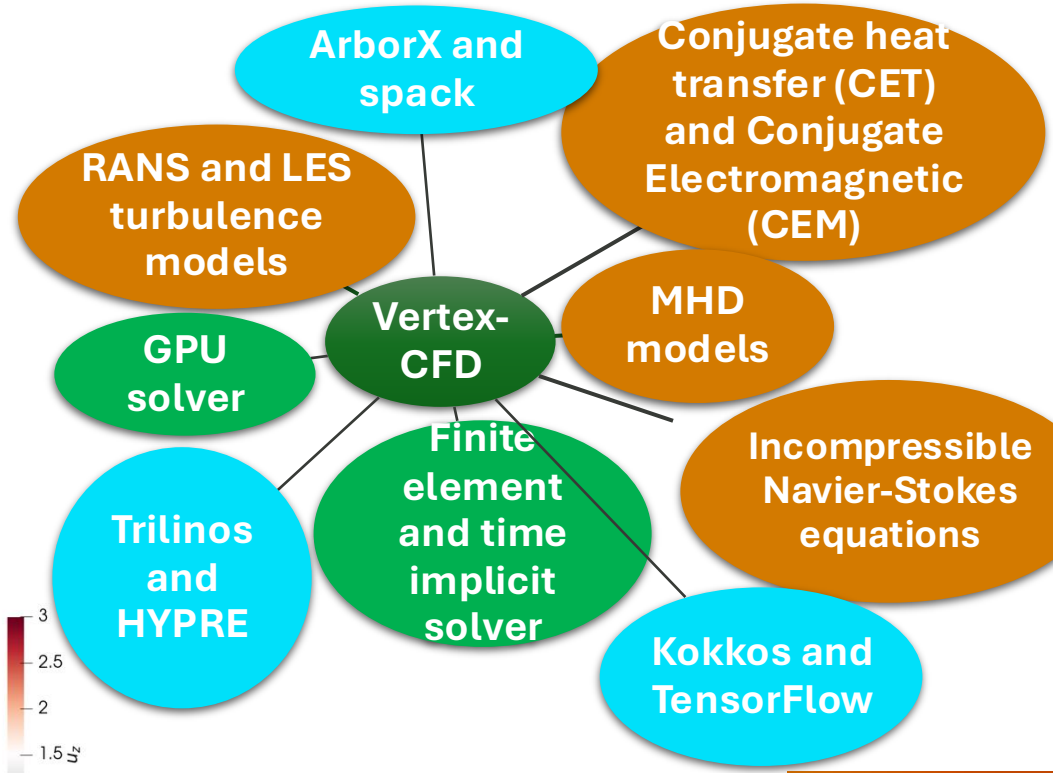
- LDRD.
- SciDAC (ASCR, FES, and NE).
- NNSA.
- UT-ORII.
- INFUSE.

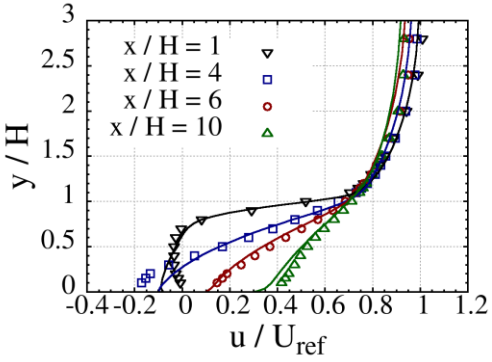
[1] **Lead developer:** Marco Delchini, delchinimg@ornl.gov.

[2] **Principal Investigator**: Tim Younkin, younkintr@ornl.gov

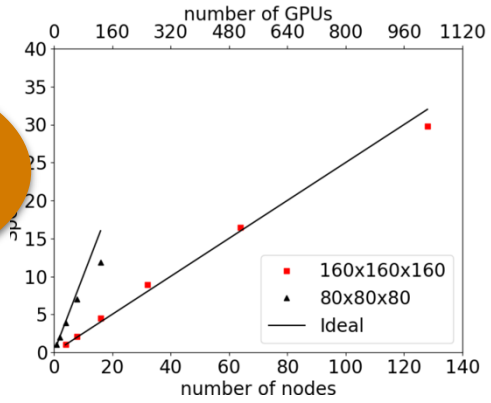
Vertex-CFD







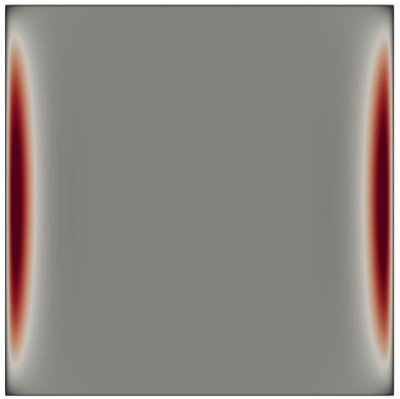
(b)



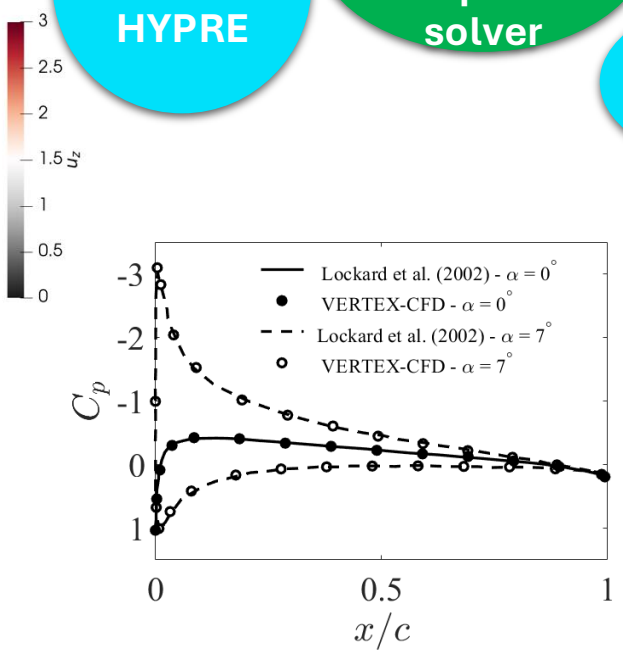
(c)

Legend (clockwise):

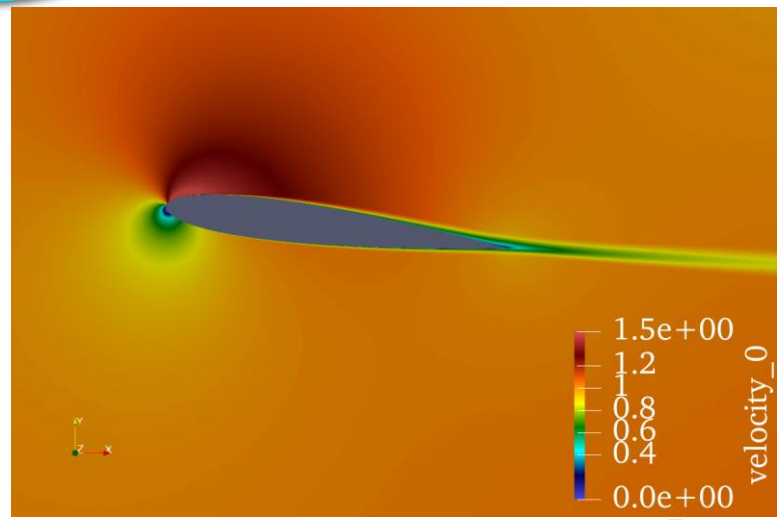
a. *Modeling turbulent flow of lid-driven cavity and turbulent channel with LES.*
b. *RANS solution of flow over a backward step.*
c. *Strong scaling on Frontier.*
d. *Flow over a NACA0012 airfoil (Re = 500,000).*
e. *Pressure coefficient for different angle of attacks.*
f. *Hunt problem (Ha = 300, Re = 10 and c = 0.05)*



(f)



(e)



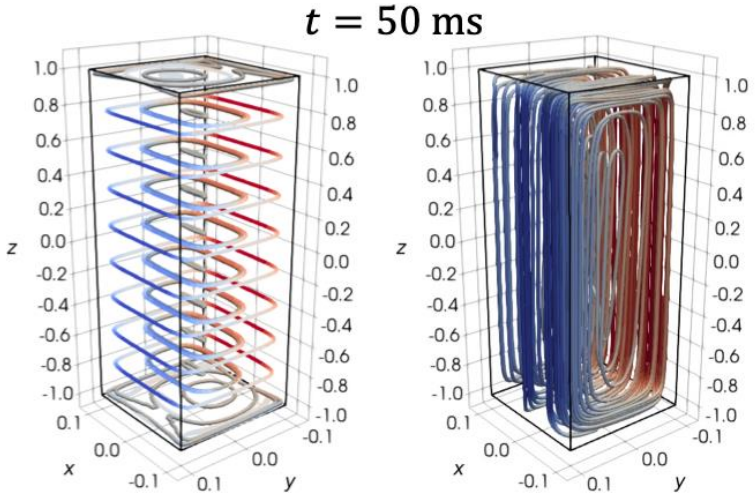
(d)

OAK RIDGE National Laboratory

Results with Vertex-CFD:

SciDAC: Full-induced MHD equations

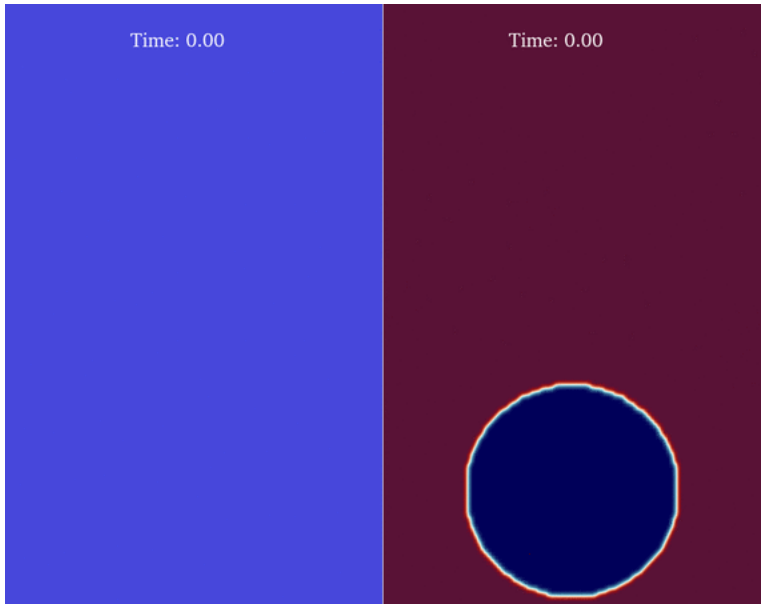
Project: Center for Simulation of Plasma-Liquid Metal Interactions in Plasma Facing Components and Breeding Blankets of a Fusion Power Reactor.



*Vertex-CFD simulation of 3D MHD duct flow driven by a transient external magnetic field. Streamlines of induced **eddy currents** (left) and **velocity** (right) illustrate the development of rapid $\mathcal{O}(10\ \mathrm{m/s})$ opposing liquid metal flows driven by a decaying plasma field. The resulting magnetic Reynolds numbers exceed unity, demonstrating the need for full-induction MHD during transient events.*

Vertex-Multiphase: volume of fluid

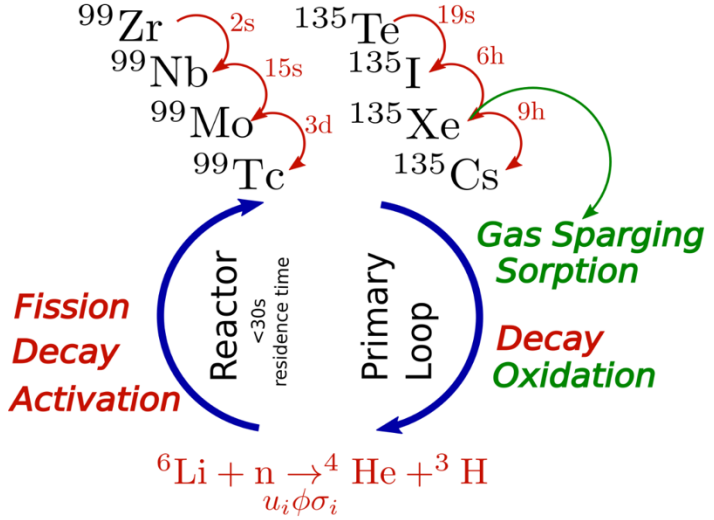
Implement volume of fluid method and level set method to track gas fission products in molten salt reactors and fusion blanket.

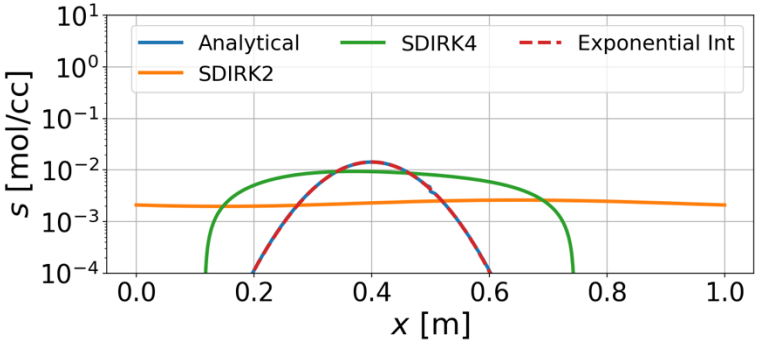


Vertex-CFD simulation of a rising bubble with VOF model coupled to the incompressible Navier-Stokes equations.

Vertex-RAD: exponential integrators

Characterization of off gas systems, dynamic salt composition and properties, tritium production, transient response to **off-normal events in MSRs**.

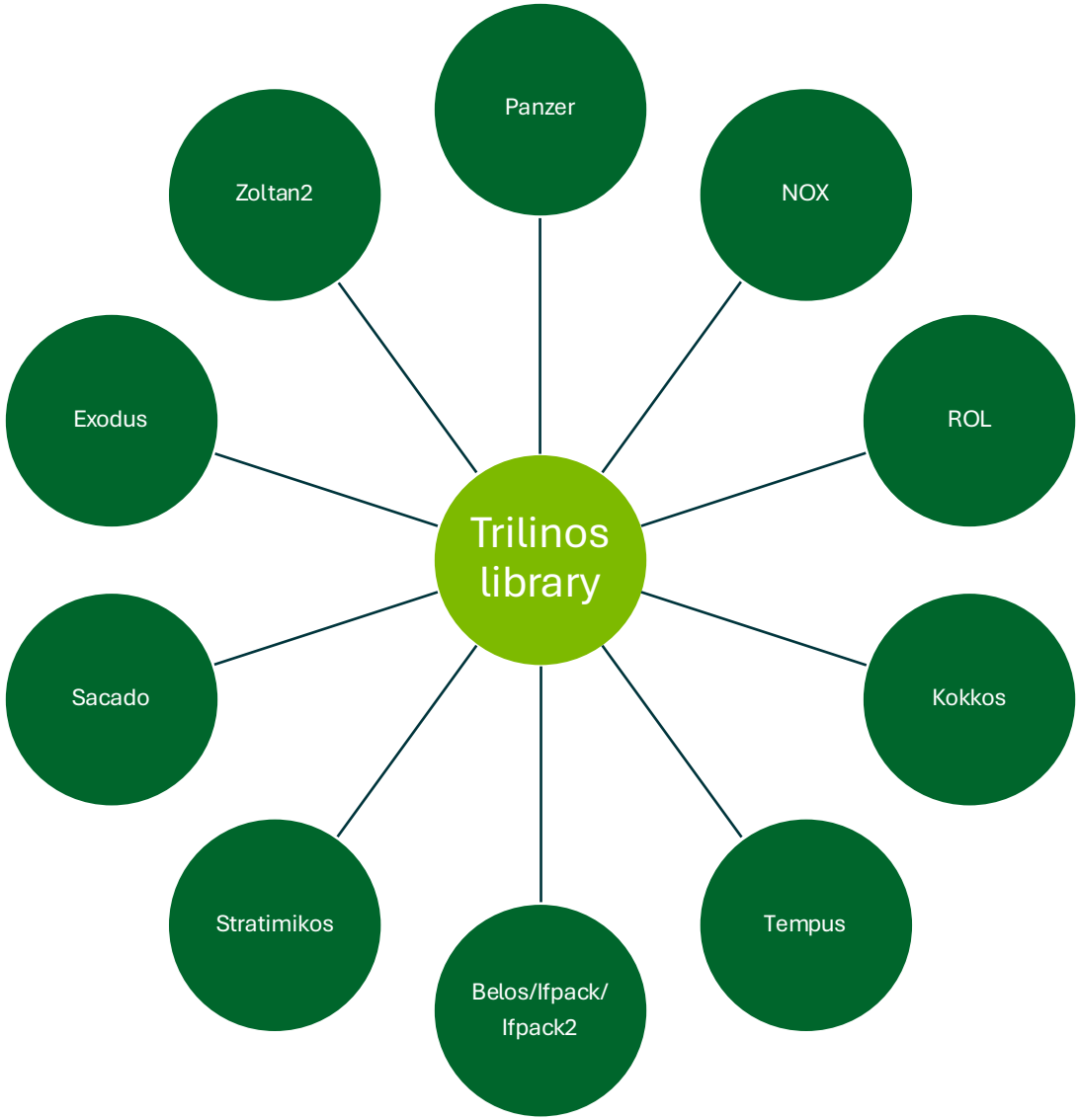




Species concentration predicted with Vertex-CFD. A reaction-advection-diffusion equation is solved with different temporal integrators, and the numerical solution is compared to the analytical solution.

What is our current approach to solving PDEs?

Main Packages of Use from the Trilinos Library



High-level approach used by ASINA and VERTEX modeling tools:

- Finite element method.
- Fully-implicit/explicit temporal solver.
- Automated differentiation to evaluate the preconditioner.
- Exodus mesh files.
- Weakly imposed boundary conditions.
- Symmetric interior penalty method for diffusion terms.
- Support quadrilateral, triangle, tetrahedral and hexagonal mesh elements.
- Multi-domain capability.

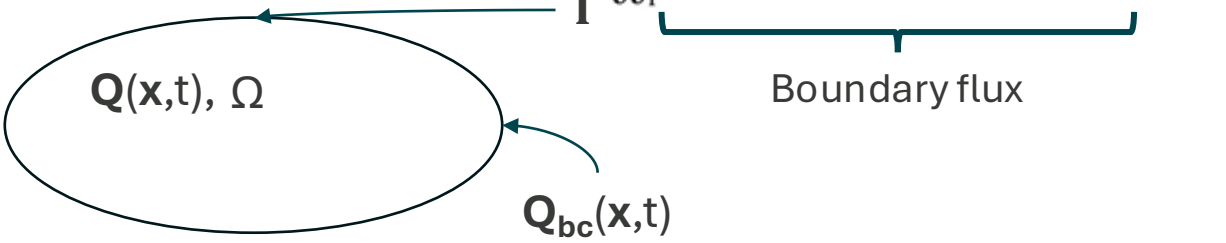
Approach to code development and analysis

Approach to code development and analysis: Partial differential equations (PDEs)

Weak form

$$\iiint_{\Omega_k} \phi_i \left(\frac{\partial \mathbf{Q}(\mathbf{x},t)}{\partial t}\right)_k \partial\Omega_k - \iiint_{\Omega_k} \nabla \cdot \phi_i \left(F_c(\mathbf{Q}(\mathbf{x})) - F_{ad}(\mathbf{Q}(\mathbf{x}), \nabla\mathbf{Q}(\mathbf{x}))\right) \partial\Omega_k +$$

$$\iint_{\Gamma} \phi_i \left(F_c(\mathbf{Q}(\mathbf{x})) \cdot \mathbf{n} - F_{ad}(\mathbf{Q}(\mathbf{x}), \nabla\mathbf{Q}(\mathbf{x})) \cdot \mathbf{n}\right) \partial\Gamma = 0$$



- **Boundary flux is weakly imposed:**
  - Computed from boundary values and interior values.
  - Symmetry interior penalty method for diffusion-like fluxes: $F_{ad}(\mathbf{Q}(\mathbf{x}), \nabla\mathbf{Q}(\mathbf{x}))$
- **Solution is discontinuous at boundaries:**
  - Boundary values $\mathbf{Q_{bc}}(\mathbf{x},t)$ and interior values $\mathbf{Q}(\mathbf{x},t)$ at each integration point on the outer boundary Γ.
  - Allows for use of Riemann-like flux for added numerical stability.

Numerical method

- No numerical method: pure Galerkin method.
- Artificial dissipation method:
  - Add an artificial dissipative flux with artificial viscosity coefficient:. $\nabla(\mu_{ad}(\mathbf{x},t)\nabla\mathbf{Q}(\mathbf{x},t))$.
  - SUPG, Taylor-Hood, Grad-Div, and entropy viscosity method (EVM).
  - Artificial viscosity coefficient is sensitive to shocks, discontinuities or strong gradients (pressure-based shock switch, entropy residual, etc…)

$$\mu_{ad}(\mathbf{x},t) = \min\left(\mu_{fo}(\mathbf{x},t), \mu_e(\mathbf{x},t)\right) = \min\left(C_{max}h(|\vec{u}| + c), C_e h^2 \frac{|R_e(\mathbf{x},t)|}{\Omega_e \mathbf{x},t)}\right)$$

$$R_e(\mathbf{x},t) = \partial_t s(\mathbf{x},t) + \vec{v} \cdot \nabla s(\mathbf{x},t)$$

$$\Omega_e = s - \bar{s}$$

Approach to code development and analysis: Implementation

Testing and continuous integration (CI)

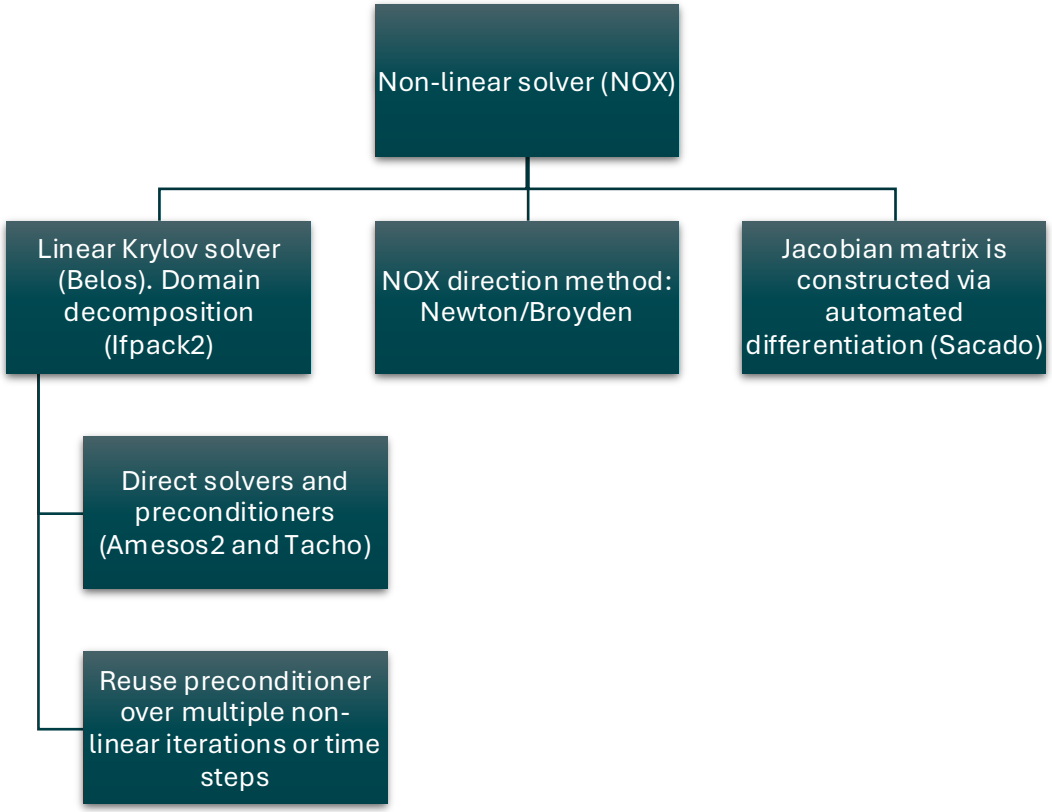
- Review process with check list.
- Unit test for closure models.
- Regression tests: verification and validation problems.
- Keep code coverage above 90%.
- Continuous integration:
    - Unit test on CPUs and GPUs.
    - Regression tests on CPUs and GPUs.
    - Code formatting.

Documentation

- Limited user and developer documentation available.
- Currently under development.

Approach to code development and analysis: HPC performance and production runs

Current strategy: brut force



Take away:

- Current strategy provides a robust but slow solver.
- Jacobian matrix is built with automated differentiation (Secado package)
  - Memory intensive (limit GPU usage).
  - Time-consuming task (up to 40% of clock time)
- The solution of the linear system, including the application of the preconditioner, constitutes the main performance bottleneck in production simulations.
- Current approach does not leverage block preconditioner.

What are we working on?

Blocking DOFs unlocks new preconditioning options.

- Motivations and approach:
  - Speed-up solver by using physics-based preconditioners.
  - Reduce memory footprint of preconditioner on GPUs.
  - Use of Teko for block preconditioning.

- Preconditioning for systems of equations commonly targets structure of system, e.g. for incompressible Navier-Stokes:

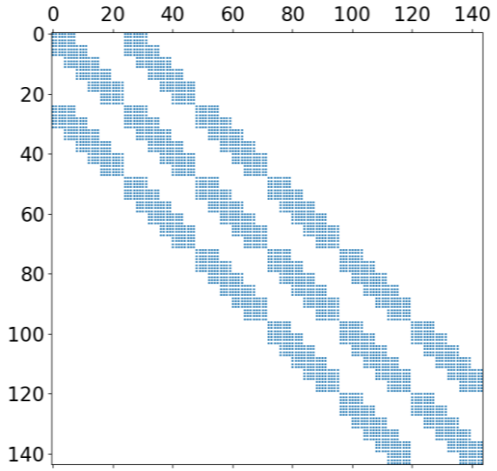
$$\begin{bmatrix} A & B^T \\ B & C \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix}$$

$$S \equiv C - BA^{-1}B^T$$

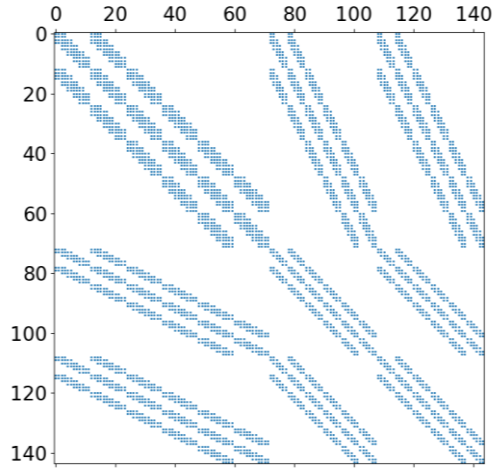
- Many common approaches use Schur complement of system:

$$\mathcal{P} = \begin{bmatrix} \tilde{A} & 0 \\ C & \tilde{S} \end{bmatrix}$$

- Different approximations to block inverses lead to different schemes
  - Semi-implicit method for pressure linked equations (SIMPLE).
  - Least squares commutator (LSC).
  - Pressure convection diffusion (PCD).



All DOFs grouped together



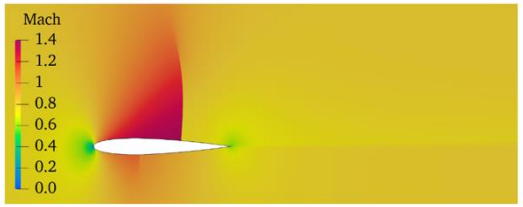
Velocity and pressure separated

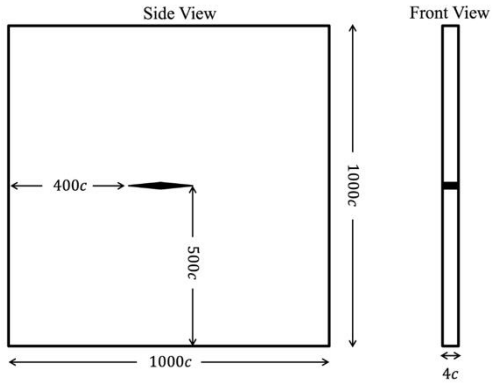
Adaptive Mesh Refinement

Inviscid transonic flow over diamond and NACA airfoil

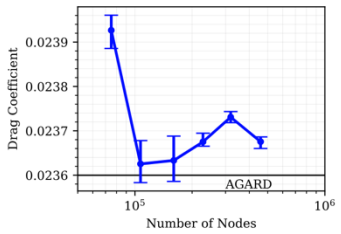
- **Geom**: NACA 0012 airfoil with chord of 1 and span of 4
- **Boundary conditions**: free-slip (airfoil, side walls) and farfield (elsewhere)
- **Initial complexity**: 25,000
- **Increase factor** 1.5
- **Number of complexities**: 6
- **Adaptations per complexity**: 8

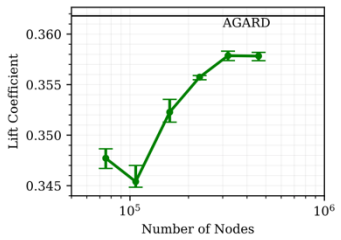
Airfoil	M_∞	AOA (°)
NACA 0012	0.8	1.25





OAK RIDGE National Laboratory



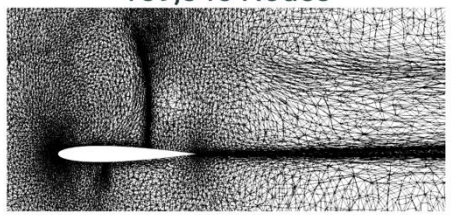


Centerline Cut

Top

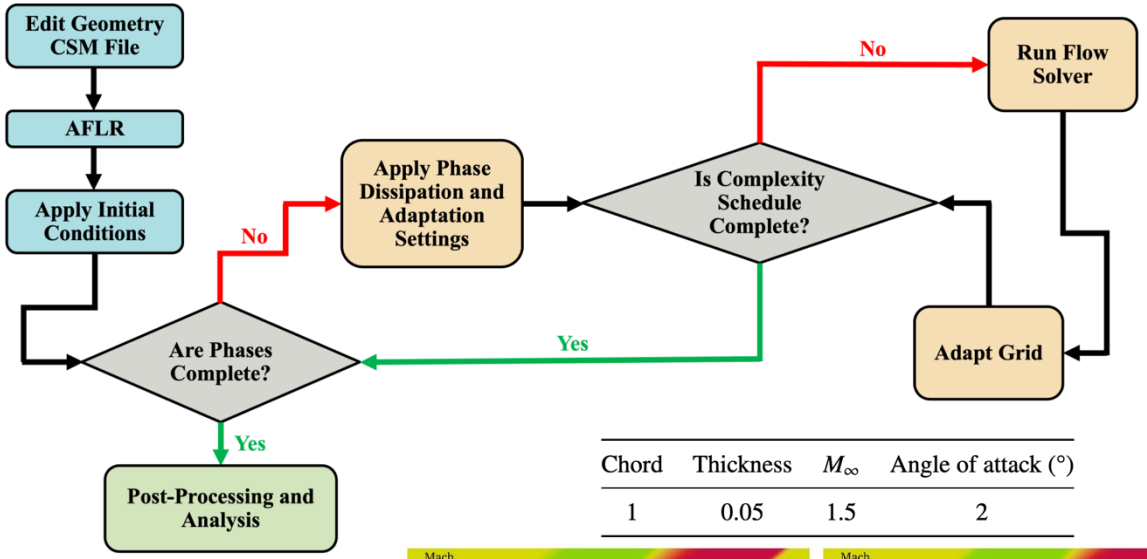
Bottom

189,845 Nodes

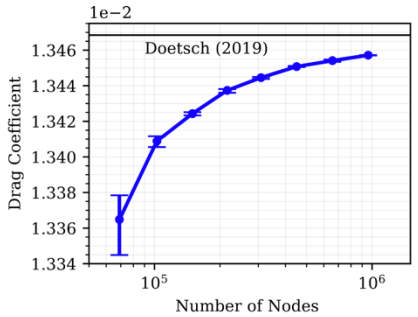


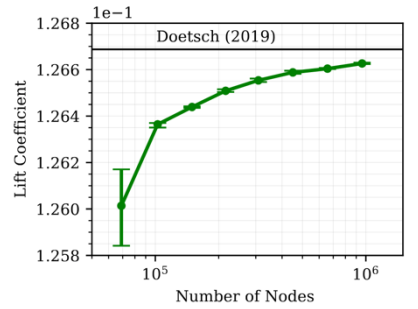
OAK RIDGE National Laboratory

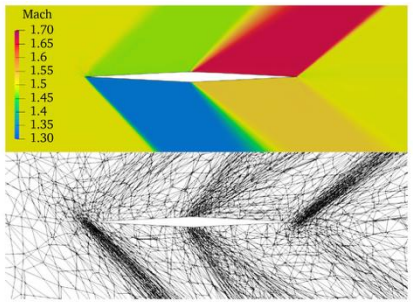
AMR approach



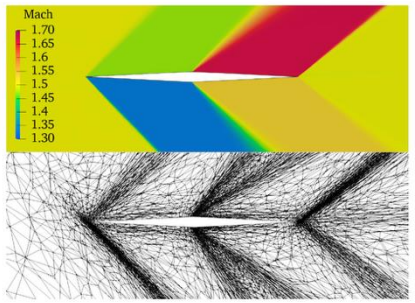
Chord	Thickness	M_∞	Angle of attack (°)
1	0.05	1.5	2



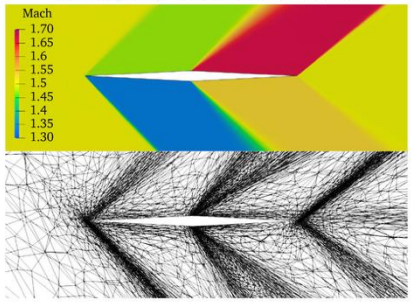




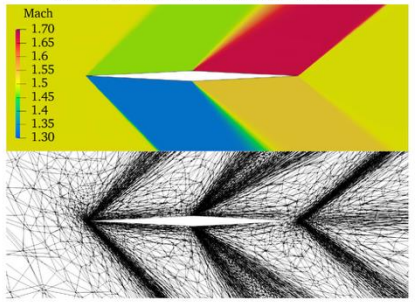
(a) Complexity = 25,000 (Nodes = 69,628)



(b) Complexity = 84,375 (Nodes = 215,636)



(c) Complexity = 126,563 (Nodes = 307,322)



(d) Complexity = 427,149 (Nodes = 959,985)

K. Doetsch et al. High-Fidelity, Low-Dissipation/Symmetry-Preserving Numerical Scheme for Solving the Euler Equations with Unstructured, Metric-Based Mesh Adaptation, AIAA 2026-0765.

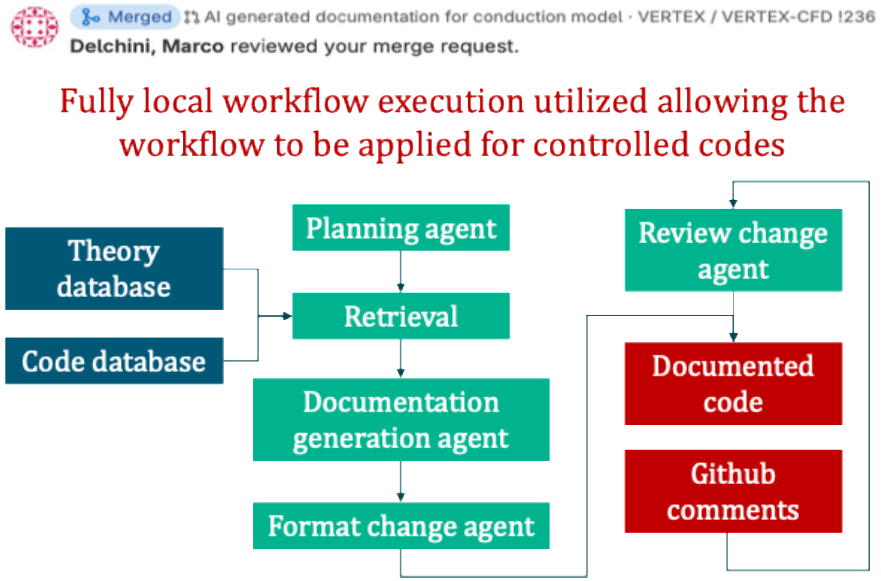
Artificial intelligence, machine learning and code development

Code development

- Update source code to take advantage of composite class for multiphysics implementation.
- Multi-domain capability for high-speed flows.
- Physics-based temporal integrators.
- Output fields on boundaries.

AI for documentation

- Use AI agent to generate Doxygen documentation.
- Investigating use of AI agent to generate user input documentation.



Wishlist and conclusions

Wishlist and conclusions

Wishlist

- Discontinuous Gakerkin method:
  - Mixed finite element method
  - Hybridizable Discontinuous Galerkin (hDG)
- Better documentation of Trilinos packages:
  - Solver settings.

Conclusions

- Trilinos library is heavily leveraged at ORNL and will continue for the next five years at least.
- Overall, developers are very well satisfied with the level of support from the Trilinos team.
- Steep learning curve for new developers.
- Current modeling approach has allowed analysis of complex problems.
- Time to solution will be improved with better preconditioning strategies.

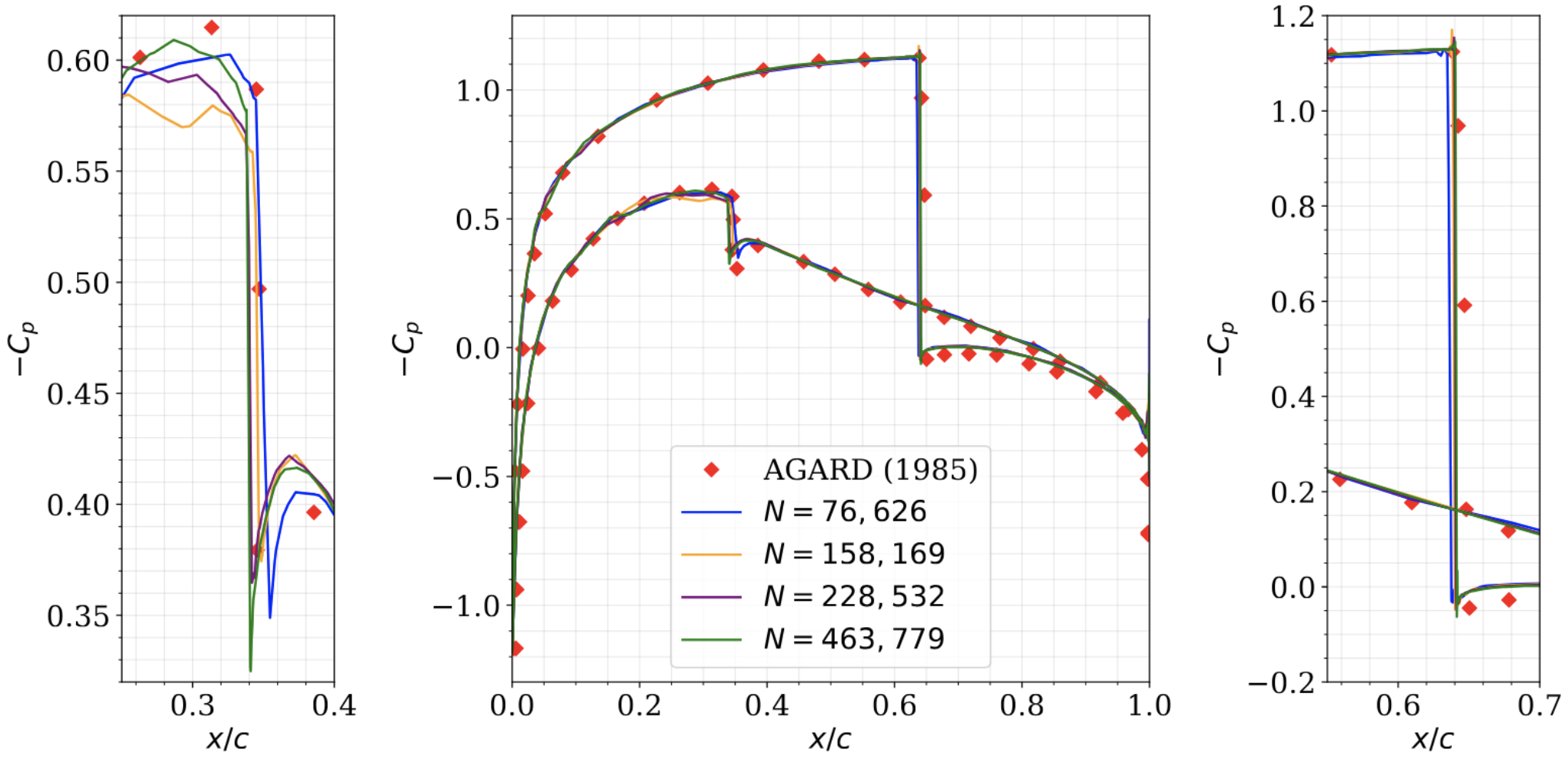
Q&A

Thanks for your attention

Names of contributors (alphabetical order):

- Lance Bullerwell
- Kwitae Chong
- Marco Delchini
- Kevin Doetsch
- Eirik Endeve
- Taylor Erwin
- Ryan Glasby
- Will Gurecky
- Steven Hamilton
- Austin Isner
- Daniel Reasor
- Lawton Shoemake
- Stuart Slattery
- Doug Stefanski

NACA 0012 Airfoil in Transonic, Inviscid Flow



Ref: *Pulliam, T., and Barton, J., "Euler computations of AGARD Working Group 07 airfoil test cases," 23rd Aerospace Sciences Meeting, 1985*

Metric Tensor Element Length

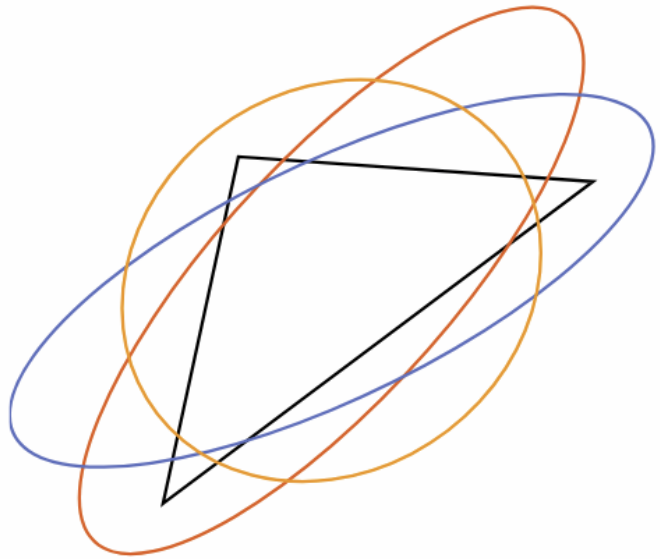
- Component of local mesh size, $h_{\vec{x}}$, is

$$h_{\vec{x},i} = \sqrt{\mathcal{M}_{ii}}$$

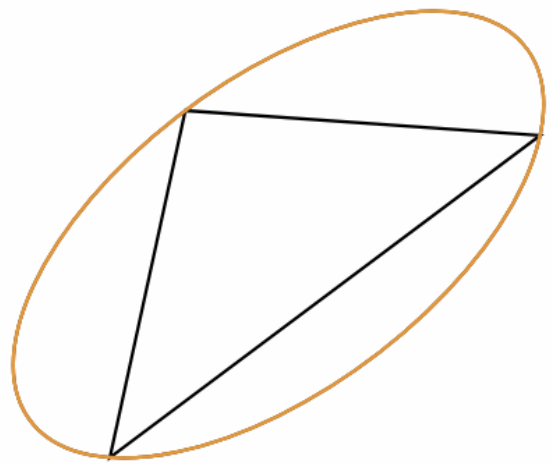
- $\mathcal{M}$ is a metric tensor derived from the element-local mapping from the ideal (unit equilateral) to the physical element
- Using a right triangle as the reference element leads to three different metrics for each of the three node orderings
- The metric tensor, $\mathcal{M}$, is calculated from the Jacobian of the coordinate transformation as

$$\mathcal{M} = \mathcal{J}\mathcal{J}^{\mathsf{T}}$$

Reference Triangle



Ideal Triangle



Shock Switch Parameter

- Define ϵ_S as a scaling term that spans from $[0, 1]$:

$$\epsilon_S = \tilde{\epsilon}_S \tanh\left(\chi \tilde{\epsilon}_S\right)$$

- χ is a user-defined parameter that designates how quickly ϵ_S approaches zero
- The term $\tilde{\epsilon}_S$ is defined as

$$\tilde{\epsilon}_S = \frac{\sqrt{\nabla p \mathcal{M} \nabla p^{\mathsf{T}}}}{\sqrt{\nabla p \mathcal{M} \nabla p^{\mathsf{T}}} + \kappa_S p}$$

- κ_S is a non-dimensional tunable multiplier to modify the influence ϵ_S has on the artificial diffusion terms
- The elemental shock switch is defined as $\epsilon_{\text{elm}} = \max(\epsilon_S, \epsilon_{\text{global}})$, where ϵ_{global} is a global field of artificial diffusion

Modified Weak Form with Artificial Diffusion Fluxes

$$\sum_k \iiint_{\Omega_k} \left(\boldsymbol{\phi}^\mathsf{T} \frac{\partial \mathbf{U}_h}{\partial t} - \nabla \boldsymbol{\phi}^\mathsf{T} \cdot \left[\mathbf{F}(\mathbf{U}_h) - \tilde{\mathbf{G}}(\mathbf{U}_h, \nabla \mathbf{U}_h) \right] \right) d\Omega_k$$
$$+ \sum_k \oiint_{\partial\Omega_k \bigcap \partial\Omega} \mathbf{n} \cdot \left(\boldsymbol{\phi}^\mathsf{T} \left[\mathbf{F}(\mathbf{U}_b) - \tilde{\mathbf{G}}(\mathbf{U}_h, \mathbf{U}_b, \nabla \mathbf{U}_h) \right] \right) d\Gamma_b = 0$$

- $\tilde{\mathbf{G}}$ represents the artificial diffusion viscous fluxes:

$$\tilde{\mathbf{G}}_x = h_x \epsilon_{\text{elm}} \lambda_{\max} \begin{bmatrix} \frac{\partial \rho}{\partial x} \\ \frac{\partial \rho u}{\partial x} \\ \frac{\partial \rho v}{\partial x} \\ \frac{\partial \rho w}{\partial x} \\ \frac{\partial \rho h_t}{\partial x} \end{bmatrix}, \quad \tilde{\mathbf{G}}_y = h_y \epsilon_{\text{elm}} \lambda_{\max} \begin{bmatrix} \frac{\partial \rho}{\partial y} \\ \frac{\partial \rho u}{\partial y} \\ \frac{\partial \rho v}{\partial y} \\ \frac{\partial \rho w}{\partial y} \\ \frac{\partial \rho h_t}{\partial y} \end{bmatrix}, \quad \tilde{\mathbf{G}}_z = h_z \epsilon_{\text{elm}} \lambda_{\max} \begin{bmatrix} \frac{\partial \rho}{\partial z} \\ \frac{\partial \rho u}{\partial z} \\ \frac{\partial \rho v}{\partial z} \\ \frac{\partial \rho w}{\partial z} \\ \frac{\partial \rho h_t}{\partial z} \end{bmatrix}$$

- Inclusion of artificial viscous boundary fluxes is an area of our research

The EDAC-FI-MHD model with divergence cleaning method

$$\partial_t(\rho_0\,\boldsymbol{v}) + \nabla\cdot\left(\rho_0\,\boldsymbol{v}\otimes\boldsymbol{v} + \phi\,\boldsymbol{I} - \boldsymbol{\tau}\right) = \boldsymbol{J}\times\boldsymbol{B}$$

$$\partial_t\phi + \rho_0\,c_0^2\,\nabla\cdot\boldsymbol{v} = \zeta\,\nabla^2\phi$$

$$\boldsymbol{\tau} = \rho_0\,\nu\left((\nabla\boldsymbol{v}) + (\nabla\boldsymbol{v})^{\mathsf{T}}\right) \qquad \zeta = \gamma\,\nu/\mathrm{Pr} \quad ①$$

$$\partial_t\boldsymbol{B} + \nabla\times\boldsymbol{E} = -\boldsymbol{v}\,(\nabla\cdot\boldsymbol{B}) - \nabla\psi$$

$$\partial_t\psi + c_h^2\,\nabla\cdot\boldsymbol{B} = -\alpha\,\psi$$

$$\boldsymbol{J} = (\nabla\times\boldsymbol{B})/\mu_0 \qquad \boldsymbol{E} = -(\boldsymbol{v}\times\boldsymbol{B}) + \eta\,\boldsymbol{J} \quad ②$$

[1]Clausen (2013), *Phys Rev E*, **87**, 013309

[2]Dedner et al. (2002), *JCP*, **175**, 645

EDAC-FI-MHD model:

- Fully implicit scheme: SDIRK22 and SDIRK54.
- Finite element method.
- Solver package from Trilinos (see next slide).
- Divergence cleaning method (DCM) with cleaning wave speed c_h.

Liquid Metal MHD for Fusion Blanket Design

OAK RIDGE National Laboratory

SciDAC Center for Simulation of Plasma-Liquid Metal Interactions in Plasma Facing Components and Breeding Blankets of a Fusion Power Reactor

Scientific Achievement

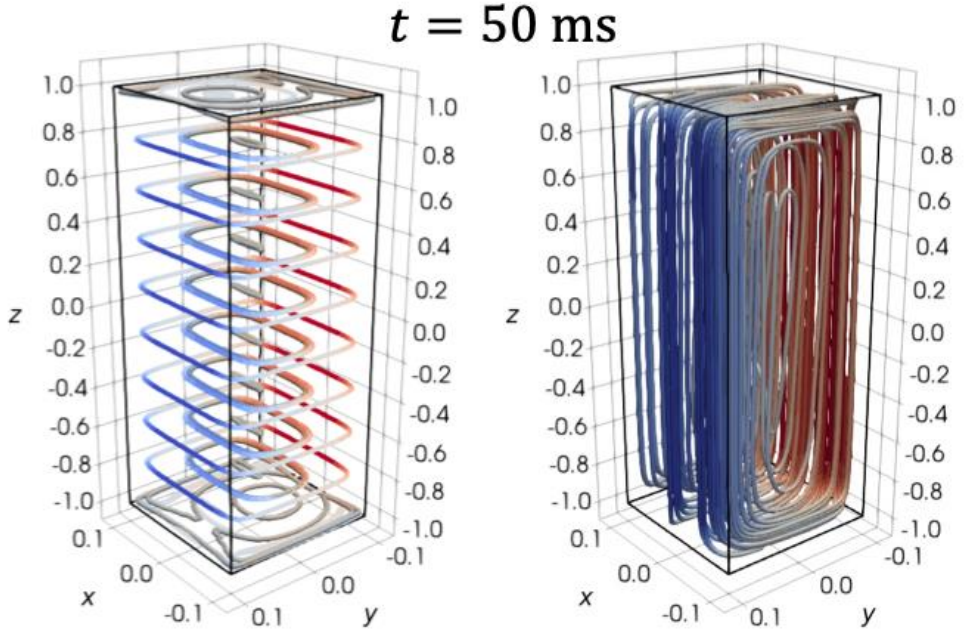
- **Motivation**—Liquid metal blankets around fusion devices may experience strong electromagnetic forces during plasma disruptions.
- **Science Driver**—SciDAC partnership aims to improve our understanding of plasma-liquid metal coupling during transient scenarios through multiphysics modeling.
- **Accomplishment**—Implemented and verified a full-induction MHD solver in the open-source Vertex-CDF multiphysics modeling framework, enabling simulation of transient liquid metal flows with dynamically evolving magnetic fields.

Significance and Impact

- New full-induction MHD capability in Vertex-CFD provides a foundation for advancing modeling of liquid metal flows in fusion-relevant transient scenarios and represents a key step towards enabling studies of coupled behavior of liquid metals, structural materials, and plasma on modern HPC systems.

Technical Approach

- Implemented incompressible, viscous and resistive MHD using artificial compressibility and generalized Lagrange multiplier methods for divergence control.
- Implicit finite element discretization with Newton–Krylov solvers for nonlinear system.
- Leverages Trilinos for linear/nonlinear solvers and preconditioning.
- Utilizes Kokkos for performance portability across CPU and GPU architectures.



*Vertex-CFD simulation of 3D MHD duct flow driven by a transient external magnetic field. Streamlines of induced **eddy currents** (left) and **velocity** (right) illustrate the development of rapid $\mathcal{O}(10\ \mathrm{m/s})$ opposing liquid metal flows driven by a decaying plasma field. The resulting magnetic Reynolds numbers exceed unity, demonstrating the need for full-induction MHD during transient events.*

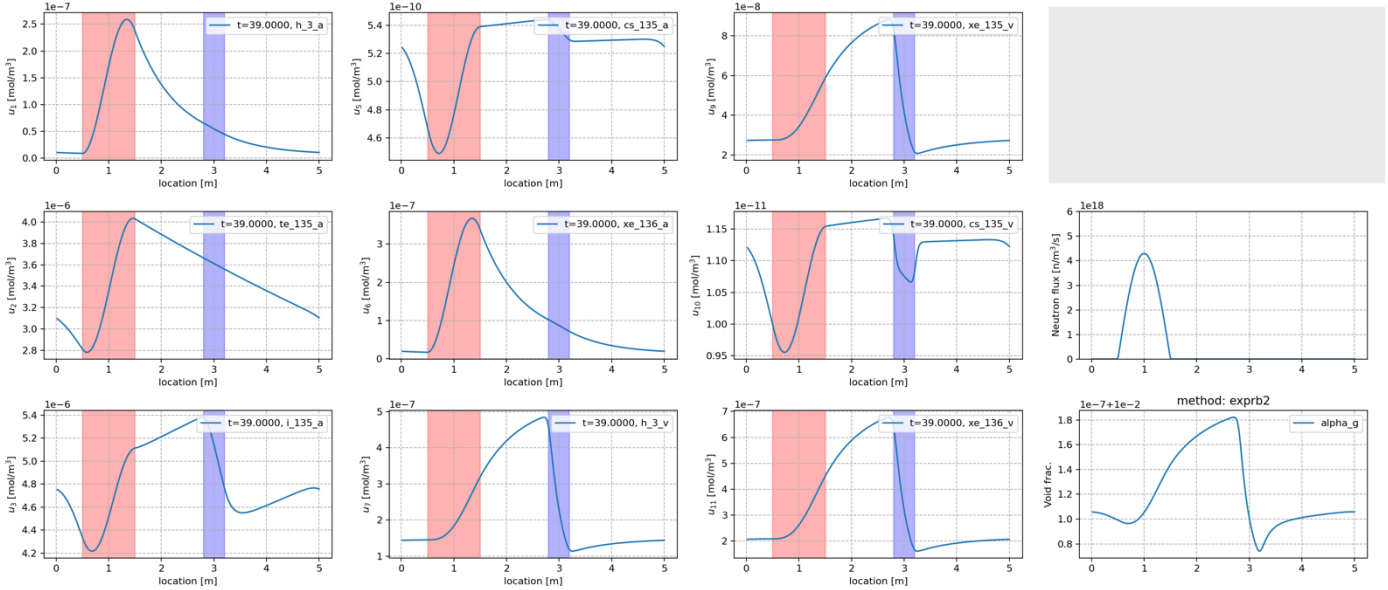
PI(s)/Facility Lead(s): Cory Hauck
Collaborating Institutions: Oak Ridge National Laboratory
ASCR Program: SciDAC
ASCR PM: Michael Halfmoon
Publication(s) for this work: Endeve, Stefanski, Delchini, et al., "A Full-Induction Magnetohydrodynamics Solver for Liquid Metal Fusion Blankets in Vertex-CFD," *Fusion Engineering and Design* (2025): Submitted (arXiv:2511.15549)

VERTEX-RAD Project Overview

Bring value to users of VERTEX and Trilinos by implementing general, **GPU-accelerated exponential integrators** (EI) which provide leading performance for certain classes of stiff PDEs.

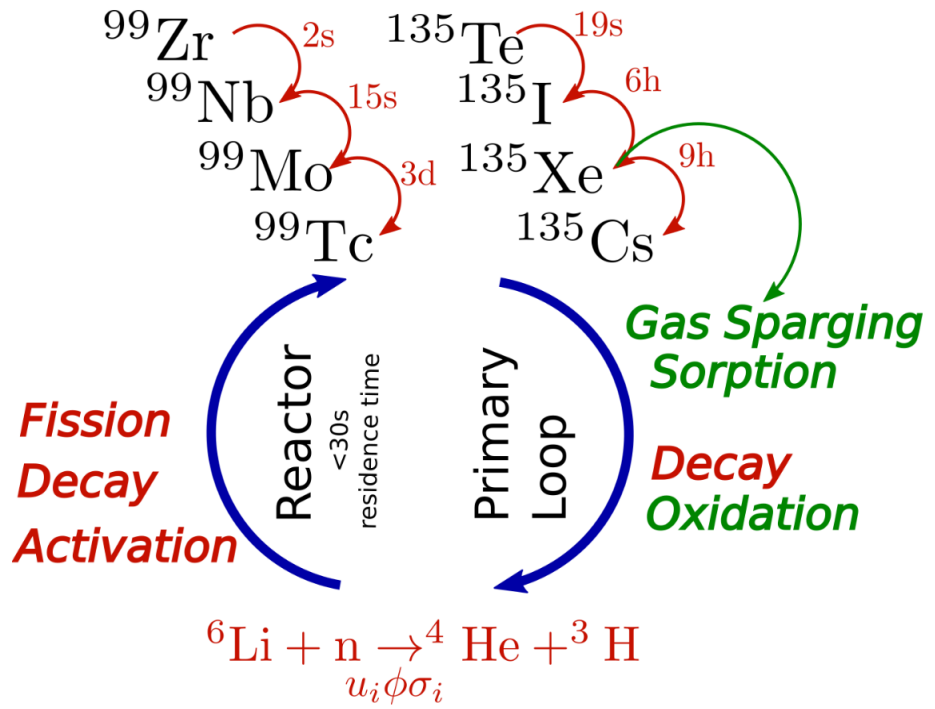
- Characterization of off gas systems, dynamic salt composition and properties, tritium production, transient response to **off-normal events in MSRs**.
- Opportunity to provide backstop solution and confirmatory analysis of lower-order codes.

Provide scalable, high performance solvers for stiff, many-species, **Reaction-Advection-Diffusion** (RAD) systems.



Exponential Rosenbrock Result: Transient, Periodic 1D, MSR Case. @t=39s. CFL~50. dt=0.5(s). V=0.5(m/s). Bateman+liquid/vapor mass transfer. Reactor region in red. Gas extraction zone in blue.

MSR System



Application: Example MSR System

$$\frac{\partial \mathbf{u}}{\partial t} = \underbrace{\mathbf{L}\mathbf{u}}_{Decay,Trans.} - \underbrace{v\nabla\mathbf{u} + D\nabla^2\mathbf{u}}_{Adv.,Diffusion} \pm \underbrace{S(\mathbf{u},t)}_{Chem.,Offgas.}$$

Bateman equations coupled to reaction, advection, diffusion (RAD) produces **stiff, linear dominate systems.**

Implementing Exponential Integrators in Tempus

Work underway to implement new class of Exponential time integrators in Tempus.

Targets stiff, semilinear PDEs. Application of interest: many-species Reaction-Advection-Diffusion systems.

Exponential integrators (EI) may be profitable when linear term is primary contributor to system stiffness

Consider $$\frac{d\mathbf{u}}{dt} = L\mathbf{u} + N(\mathbf{u}, t)$$

$$\mathbf{u}_{t+1} = e^{L\Delta t}\mathbf{u}_{t_0} + \int_0^{\Delta t} e^{L(\Delta t-\tau)} N(t_0+\tau, \mathbf{u}_{t_0+\tau}) d\tau$$

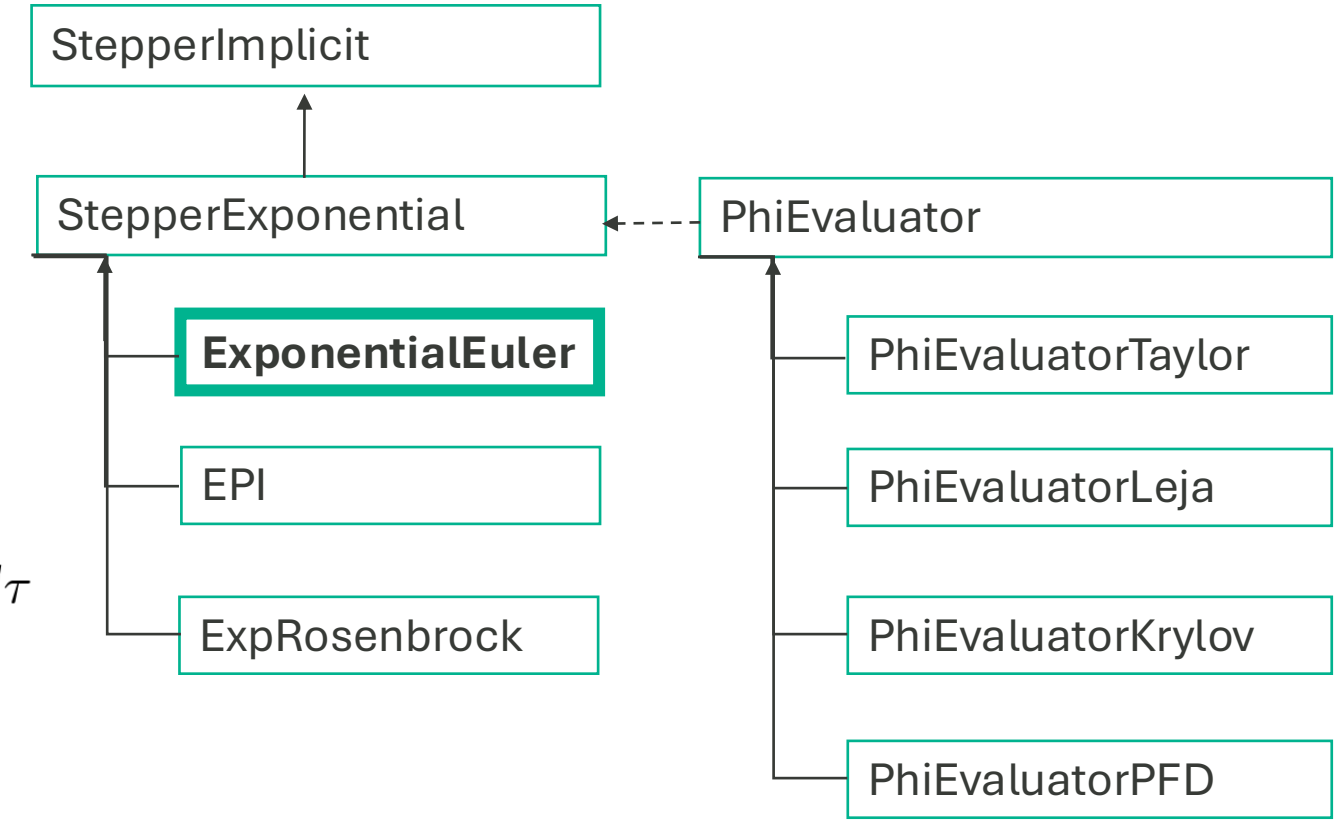
Approximating the integral via box rule on the left yields the simplest exponential integrator:

$$\mathbf{u}_{t+1} = e^{L\Delta t}\mathbf{u}_{t_0} + \Delta t \varphi_1(L\Delta t) N(t_0, \mathbf{u}_{t_0})$$

ExponentialEuler (2nd Order)

$$\varphi_0(Z) = e^Z,\ \varphi_1(Z) = Z^{-1}(e^Z - 1)$$

High level design in Tempus



Implementing Exponential Integrators in Tempus

The **PhiEvaluator** interface unifies backend implementations that compute φ-function-vector products, $\varphi_s(\boldsymbol{L})f$, which are building-blocks of exponential integration methods.

Using the **PhiEvaluatorTalyor** backend:

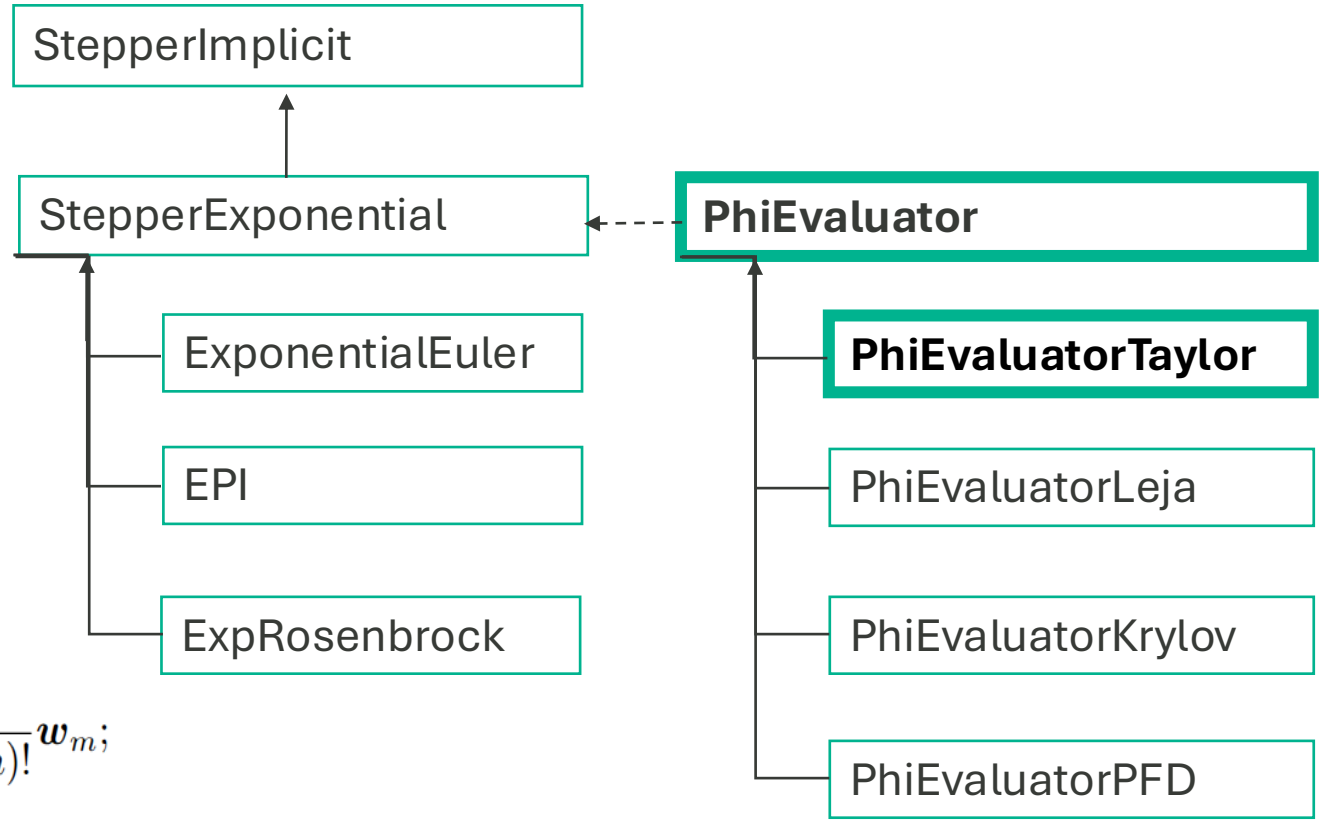
The action of the s order φ-function on a vector f can be expressed as a Taylor series

$$\varphi_s(\mathcal{L})\boldsymbol{f} = \sum_{m=s}^{\infty} \frac{\mathcal{L}^{m-s}}{m!}\boldsymbol{f} = \sum_{m=0}^{\infty} \frac{\mathcal{L}^{m}}{(m+s)!}\boldsymbol{f}$$

requiring only action of the linear operator on a vector, the sum can be rewritten as the iteration,

$$\boldsymbol{w}_0 = \boldsymbol{f}, \quad \boldsymbol{x}_0 = \frac{1}{s!}\boldsymbol{w}_0; \qquad \boldsymbol{w}_m = \frac{1}{\gamma}\mathcal{L}\boldsymbol{w}_{m-1}, \quad \boldsymbol{x}_m = \boldsymbol{x}_{m-1} + \frac{\gamma^m}{(s+m)!}\boldsymbol{w}_m;$$

High level design in Tempus



- **Similarly, the PhiEvaluatorLeja** backend computes $\varphi_s(\boldsymbol{L})f$ using a Leja polynomial interpolation method, only requiring the action of a LinOp on vector. -> Matrix free compatible.

Implementing Exponential Integrators in Tempus

Exponential integrators (EI) do not employ newton iterations, nor require solution to large linear systems, instead depend primarily on Jacobian-vector products.

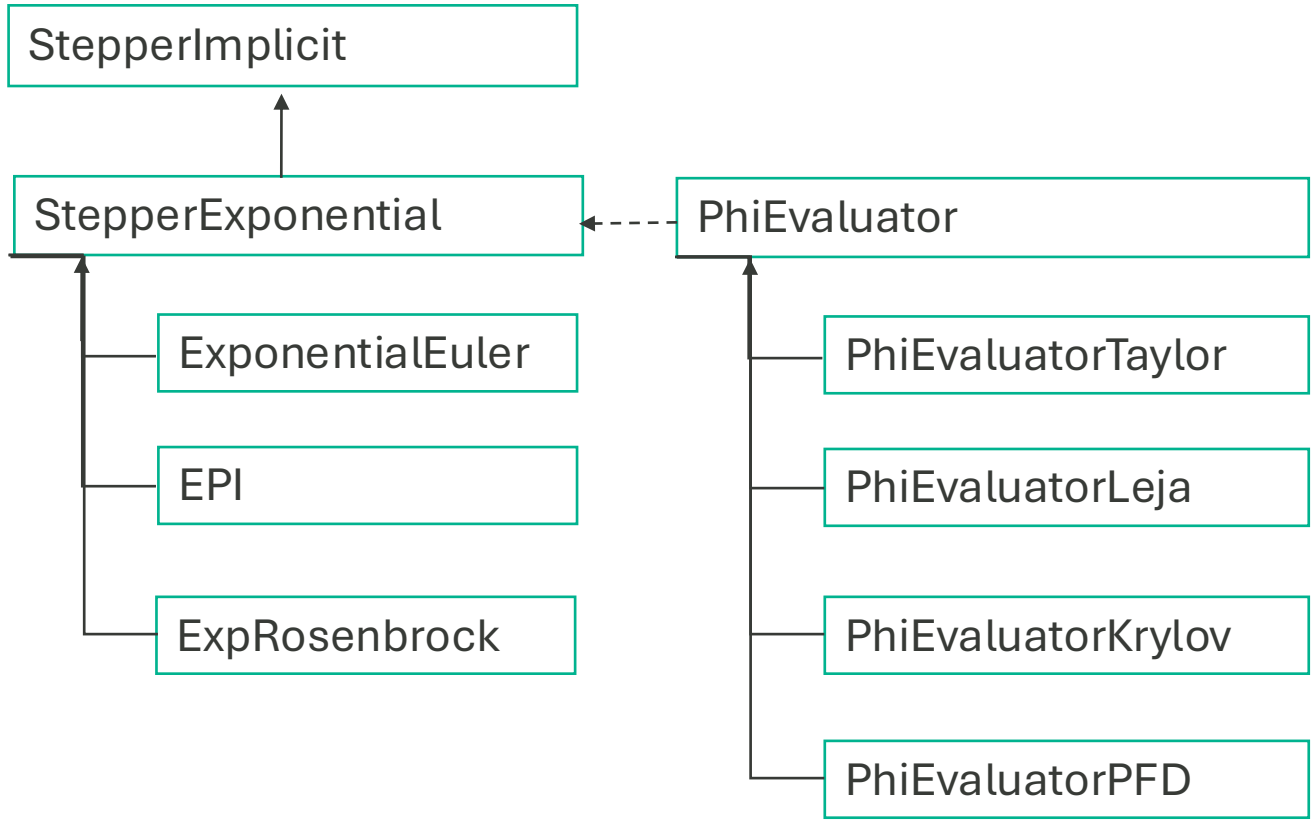
EI benefit from optimizing Jacobian-vector product logic

- Multi-species problems have block-diagonal Jacobian structure which can be exploited.

Requires total Jacobian of RHS, thus requires action of the inverse mass matrix on a vector.

- Consequently, EI have strong preference for diagonal mass matrix
  - Mass lumping
  - GLL points

High level design in Tempus

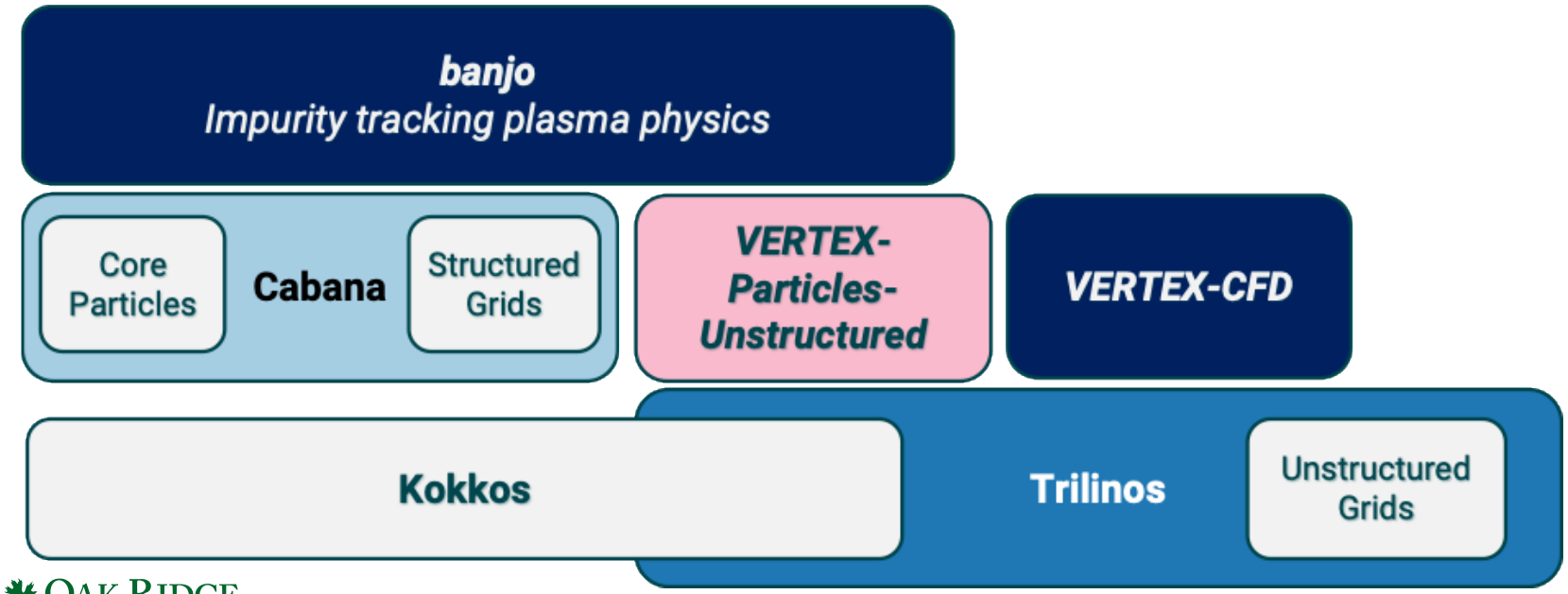


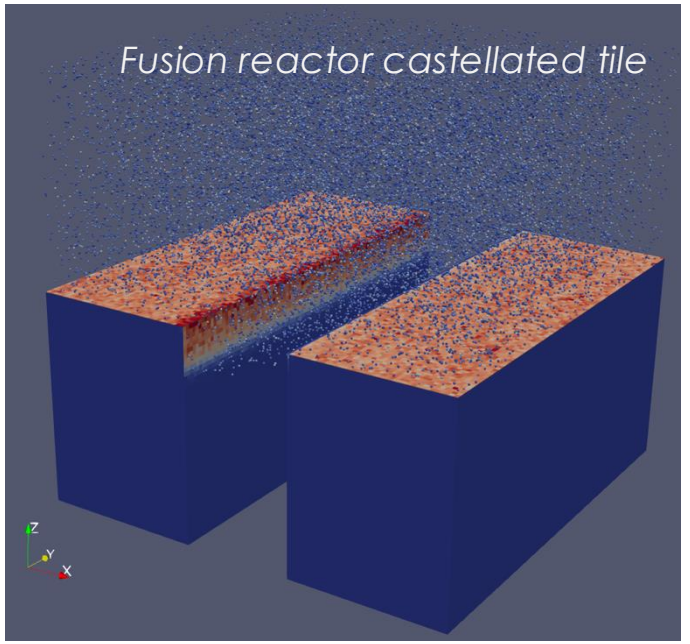
banjo: the next generation GITR[1] plasma physics tool

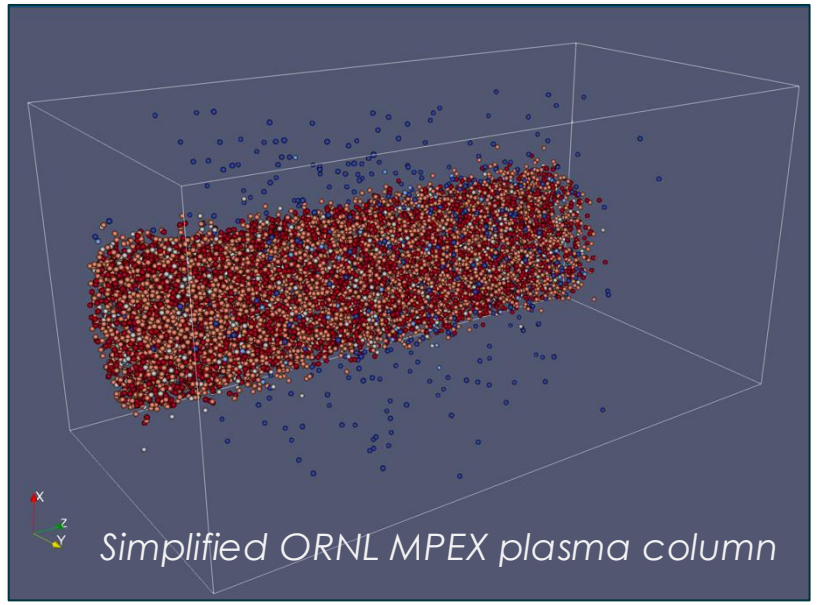
Developed through **VERTEX (PI: Franklin Curtis)**
Lance Bullerwell, Kwitae Chong, Wenjun Ge, Sam Reeve, Stuart Slattery, Tim Younkin

Objective:
- Develop particle simulation infrastructure for plasma physics predictions, including plasma impurity transport and plasma induced erosion
- Broadly support particles within the VERTEX ecosystem

Continuing work:
- Unstructured mesh integration for particles with the Trilinos library
- Particle sputtering and local re-erosion physics







1. Younkin et al. *Comp. Phys. Comm.*, *264*, 107885. 2021; https://github.com/ORNL-Fusion/GITR

Volume of fluid (VOF) method

A novel VOF method is implemented stabilized by entropy viscosity (EV, ν) and benchmarked on translating bubble and vortex stretching problems

$$\frac{\partial(\phi_\alpha)}{\partial t} + \frac{\partial(\phi_\alpha u_i)}{\partial x_i} - \nabla \cdot (\nu \nabla \phi) = 0$$

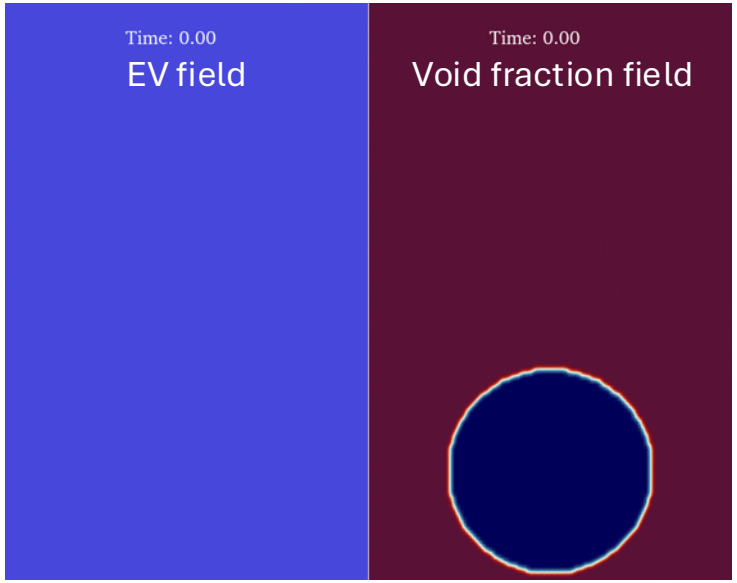
$$\nu_{max} = \frac{c_{max}}{p} h_k |\mathbf{u}|_k,$$

$$\nu_{E,k} = c_E h_k^2 \frac{D_{h,k}}{E(\phi)_k - \bar{E}(\phi)},$$

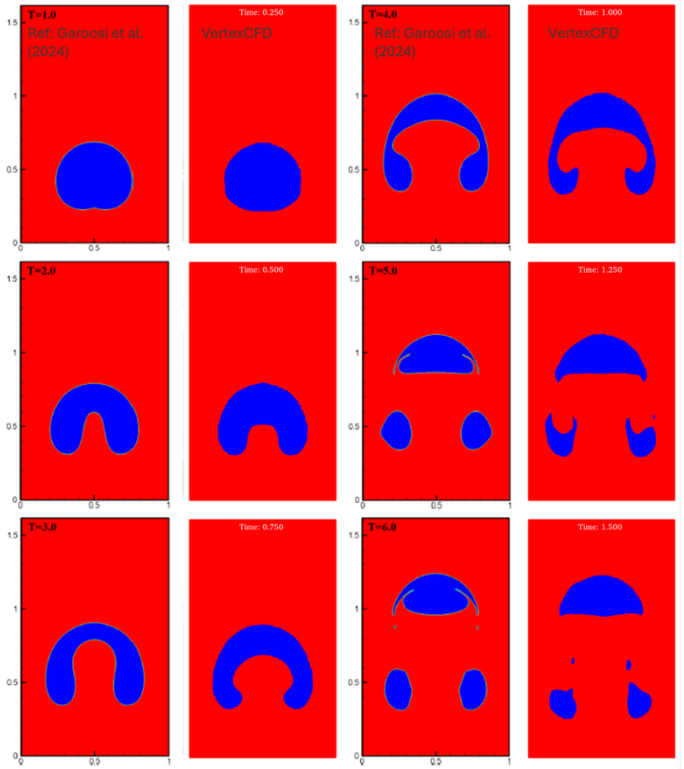
$$\nu = min(\nu_{max}, \nu_{E,k}).$$

$$E(\phi) = \frac{1}{2}\phi^2$$

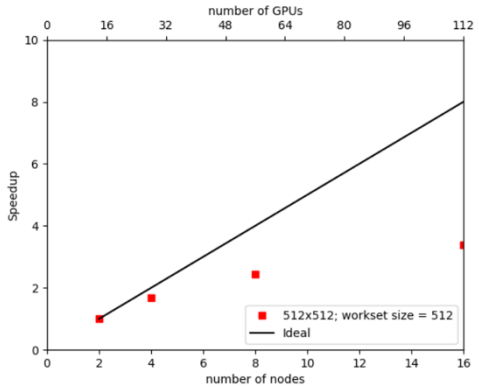
VOF has been coupled with Navier Stokes solvers in VertexCFD and tested on the bubble rising problem



$c_{max} = 0.5$ $c_E = 1.0$



Comparison at different time steps show some instabilities which are being investigated currently.



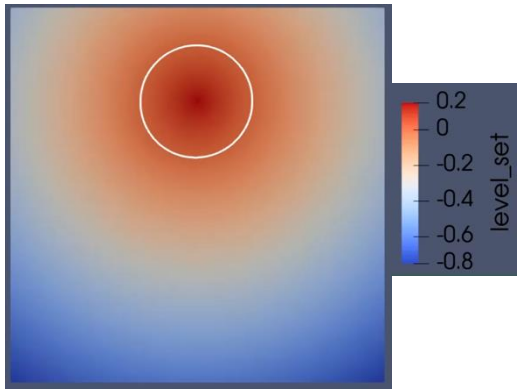
Solver also works on Frontier GPUs with investigation on scaling and speed-up underway

Conservative Level Set (CLS) method

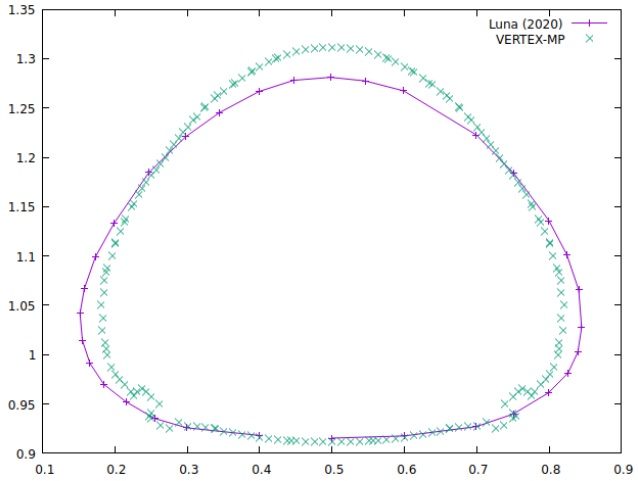
The CLS method implemented in VertexCFD uses a regularized sign function for redistancing to penalize deviations of the level set function as shown in the equations below:

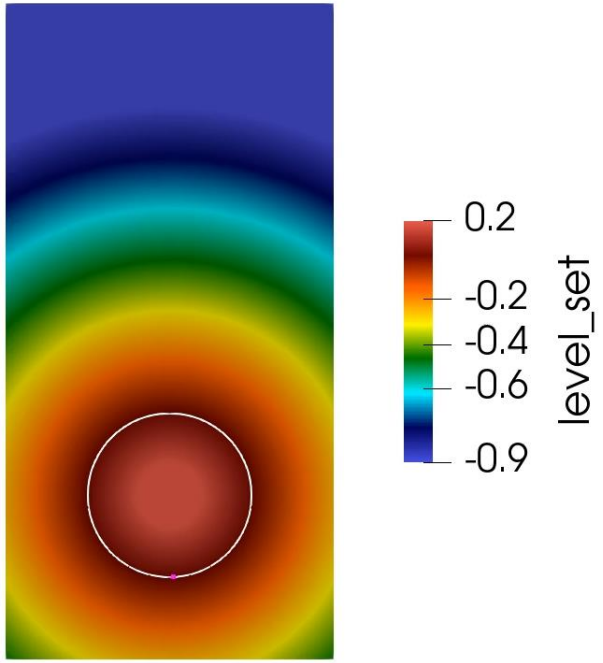
$$\partial_t S_\epsilon(\phi) + \nabla \cdot [\mathbf{v} S_\epsilon(\phi) - \lambda(\nabla\phi - \mathbf{q})] = 0, \quad \forall \mathbf{x} \in \Omega$$
$$\sqrt{|\nabla\phi|^2 + \delta^2}\mathbf{q} = \nabla\phi, \quad \forall \mathbf{x} \in \Omega$$
$$S_\epsilon(\phi) = S_\epsilon(\phi_{BC}), \quad \forall \mathbf{x} \in \partial\Omega^-$$
$$(\nabla\phi - \mathbf{q}) \cdot \mathbf{n} = 0, \quad \forall \mathbf{x} \in \partial\Omega$$

CLS shows some instabilities for the vortex stretching problem which are expected to be fixed with higher order temporal schemes



Mis-match in the bubble rising problem is being investigated currently





Dimension bridging via PDE-informed Gaussian Process

- Dimension bridging calibrates QoIs from inexpensive 2D simulations to the QoIs from expensive 3D simulations.

- Calibrations are based on 2D models and QoI data from few 3D simulations:

2D model calibration

$$\left(\mathcal{P}_L\left(u^\dagger\right), v\right) = \left(f_L, v\right) + \left(g(\theta), v\right) \quad \forall v \in V_L$$

$$\left(\mathcal{B}_L\left(u^\dagger\right), w\right) = \left(h_L, w\right) \quad \forall w \in W_L$$

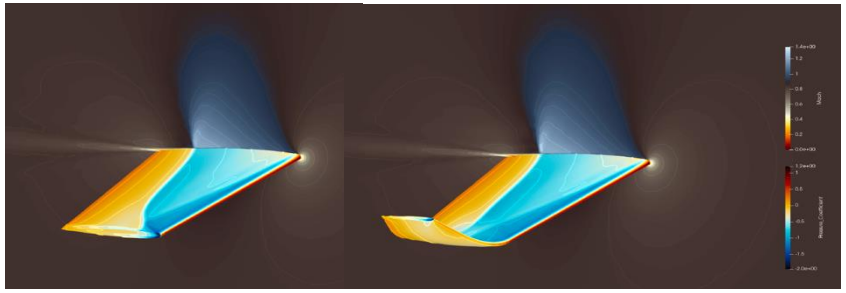
$$q_H(\theta) = q^\dagger(\theta) \quad \forall \theta \in T$$

QoI matching

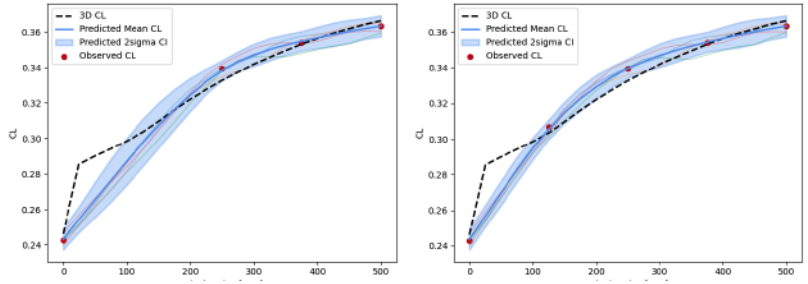
- The calibration g is modeled as a functional Gaussian process (FGP) with a PDE-informed kernel from the 2D model.

- g is trained to calibrate the 2D QoI from data.

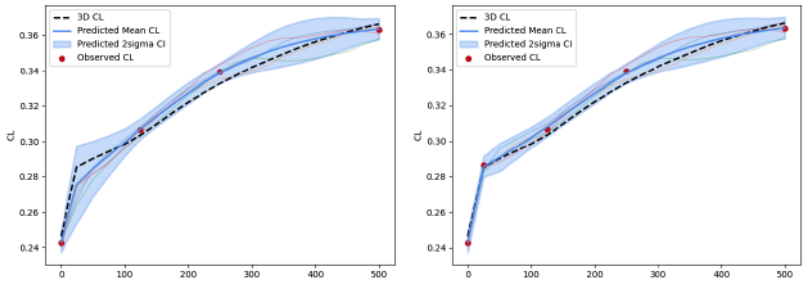
- Numerical results: 3D airfoil winglet design



- Regular GP: misses QoI feature when data is sparse



- PDE-kernel FGP: captures QoI feature early by guiding data acquisition



MHD turbulence model details

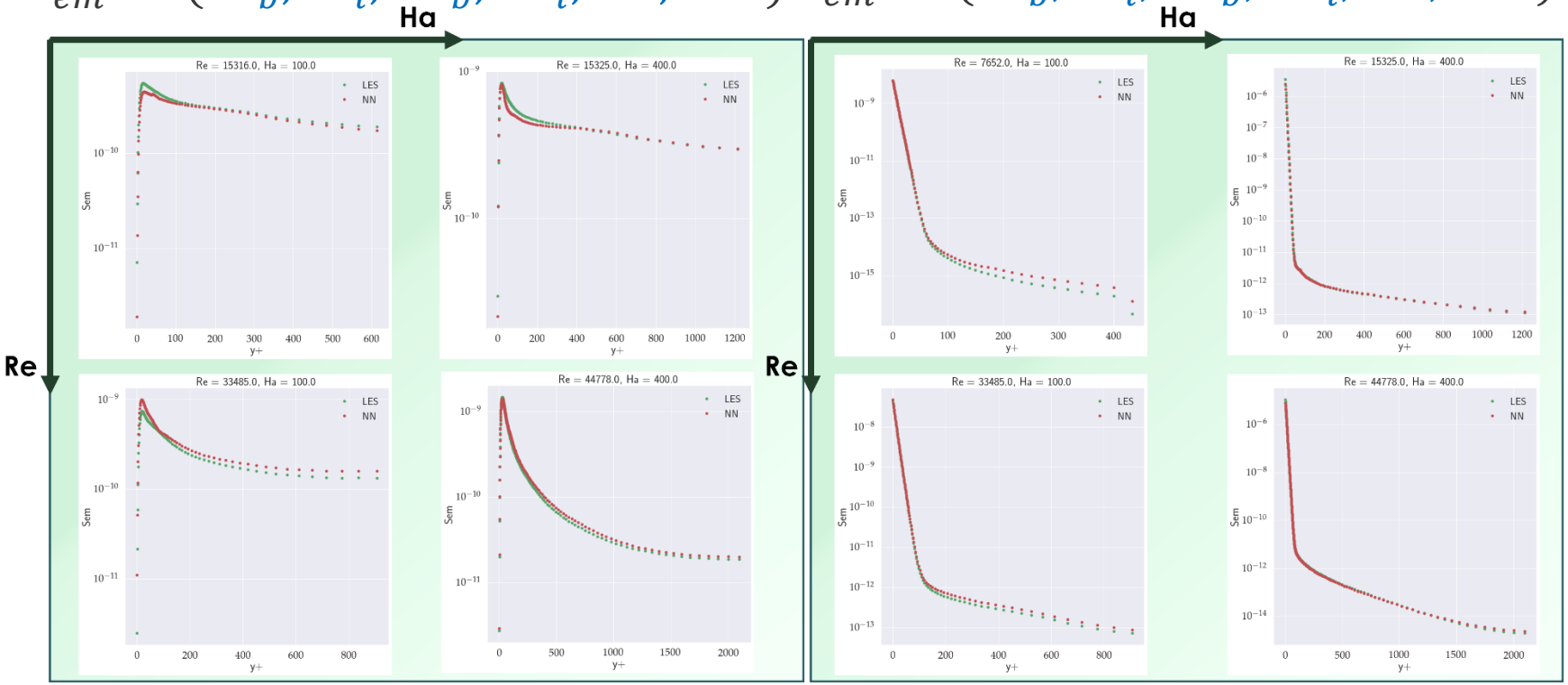
$$Re = \rho u h/\mu \qquad Ha = Bh\sqrt{\sigma/\mu} \qquad N = Ha^2/Re$$

$$\frac{\partial k}{\partial t} + (\boldsymbol{u}\cdot\nabla)k = \nu_t(\nabla\boldsymbol{u}:\nabla\boldsymbol{u}) + \left(\nu + \frac{\nu_t}{\sigma_k}\right)\nabla^2 k - \varepsilon - \varepsilon_{em}^{k}$$

$$\frac{\partial \varepsilon}{\partial t} + (\boldsymbol{u}\cdot\nabla)\varepsilon = C_1\frac{\varepsilon}{k}\nu_t(\nabla\boldsymbol{u}:\nabla\boldsymbol{u}) + \left(\nu + \frac{\nu_t}{\sigma_\varepsilon}\right)\nabla^2 k - C_2\frac{\varepsilon}{k}\varepsilon - \varepsilon_{em}^{\varepsilon}$$

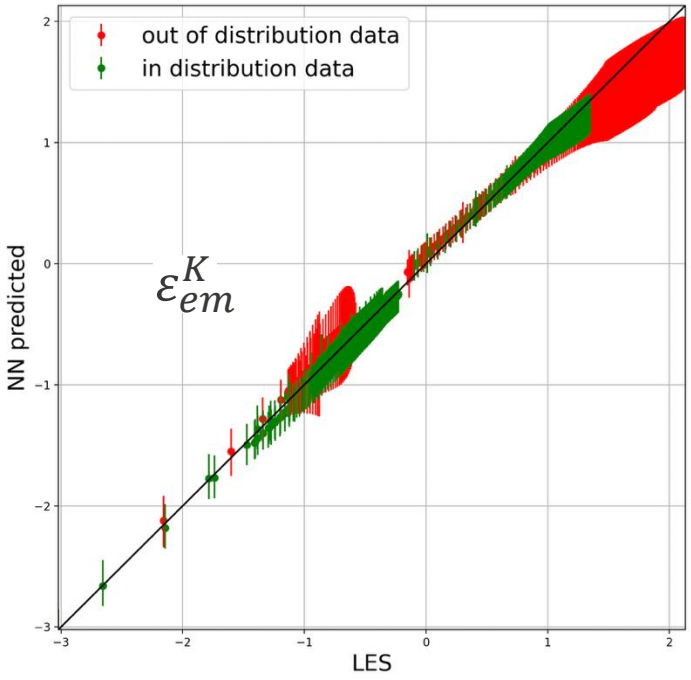
Magnetic field direction	ε_{em}^{K}	$\varepsilon_{em}^{\varepsilon}$	C_3	C_4
Streamwise	$C_3\frac{\sigma}{\rho}B_0^2K$	$C_4\frac{\sigma}{\rho}B_0^2\varepsilon$	0.02	0.015
Wall-normal	$C_3\frac{\sigma}{\rho}B_0^2K$	$C_4\frac{\sigma}{\rho}B_0^2\varepsilon$	$1.9\exp\{-1.0N\}$	$1.9\exp\{-2.0N\}$
Spanwise	$C_3\frac{\sigma}{\rho}B_0^2K$	$C_4\frac{\sigma}{\rho}B_0^2\varepsilon$	$1.9\exp\{-1.0N\}$	$1.9\exp\{-2.0N\}$

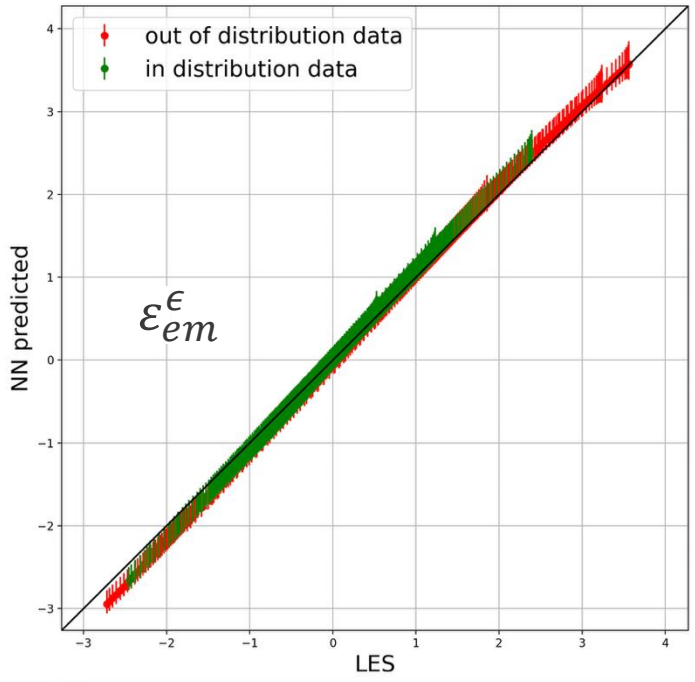
$$\varepsilon_{em}^{K} = \mathrm{f}(Re_b, Re_l, Ha_b, Ha_l, tke, diss) \quad \varepsilon_{em}^{\epsilon} = \mathrm{f}(Re_b, Re_l, Ha_b, Ha_l, tke, diss)$$



Uncertainty quantification

- Modification of prediction interval method by Liu, 2022
- Train 3 neural networks which predict - lower bound, upper bound and mean
- Clever initialization keeps high uncertainty where there is no training data.





- Two separate testing data were generated:
  - in-distribution data
    Ha = 10-40.
  - out-of-distribution data Ha = 100
- The results for the ε_{em}^{K} and $\varepsilon_{em}^{\epsilon}$ term show higher uncertainty for the out of distribution data.

[1] Siyan Liu, Pei Zhang, Dan Lu, and Guannan Zhang. "PI3NN: Out-of-distribution-aware Prediction Intervals from Three Neural Networks." International Conference on Learning Representations, 2022. https://openreview.net/forum?id=NoB8YgRuoFU