Pushing OLCF Frontier

>100T Grid Point CFD Sims via MFC and IGR

Spencer Bryngelson

Georgia Institute of Technology

OLCF User Meeting 2025

Computational Physics Group

https://comp-physics.group

7-rocket booster configuration

This work: *arXiv:2505.07392*

Pushing OLCF Frontier

>100T Grid Point CFD Sims via MFC and IGR

Spencer Bryngelson

Georgia Institute of Technology

OLCF User Meeting 2025

F. Schäfer

A. Radhakrishnan

B. Wilfong

R. Budiardja

S. Abbott

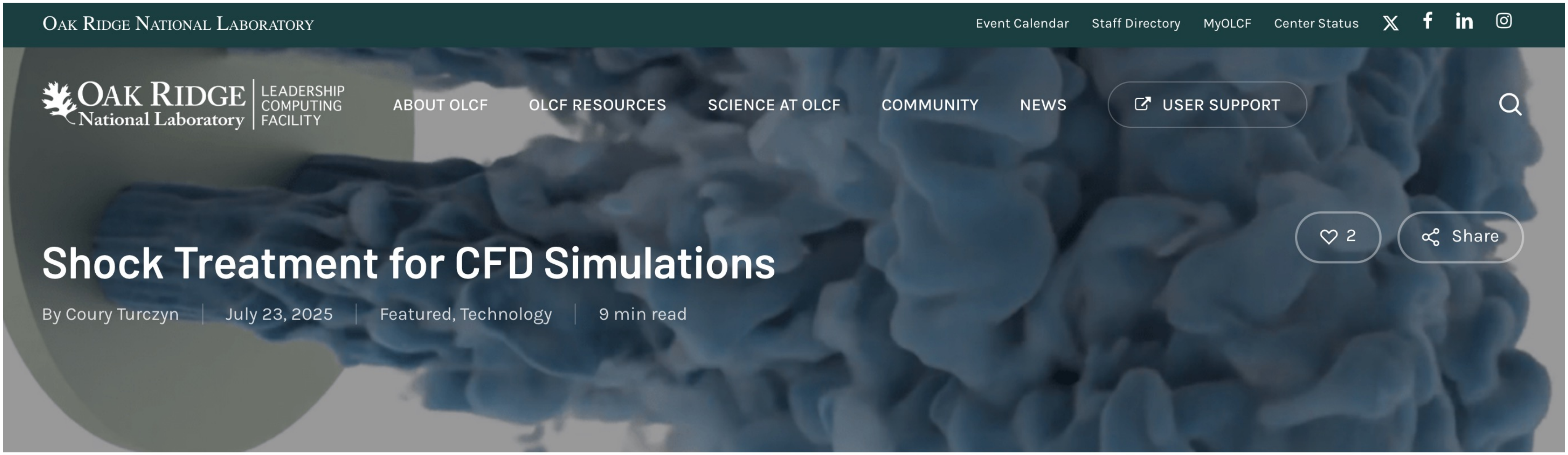
B. Cornille

Who made this possible

CPC

Computational Physics Group

https://comp-physics.group


Oak Ridge National Laboratory
Event Calendar Staff Directory MyOLCF Center Status

Oak Ridge National Laboratory | Leadership Computing Facility
ABOUT OLCF OLCF RESOURCES SCIENCE AT OLCF COMMUNITY NEWS USER SUPPORT

2 Share

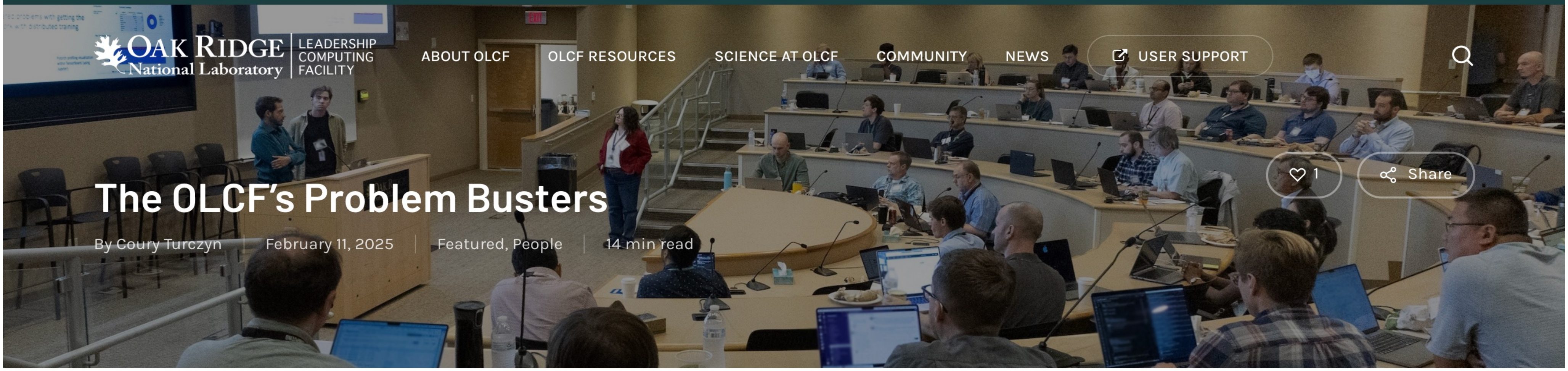
Shock Treatment for CFD Simulations

By Coury Turczyn | July 23, 2025 | Featured, Technology | 9 min read

Georgia Tech researchers use IGR and a unified memory approach to achieve the world's largest fluid dynamics simulation on Frontier

CPG
Computational Physics Group
https://comp-physics.group

I'm talking about this OLCF News post



If there's something weird in your code, who are you going to call? The OLCF's User Assistance Center provides an array of expert support services, from office hours to hackathons

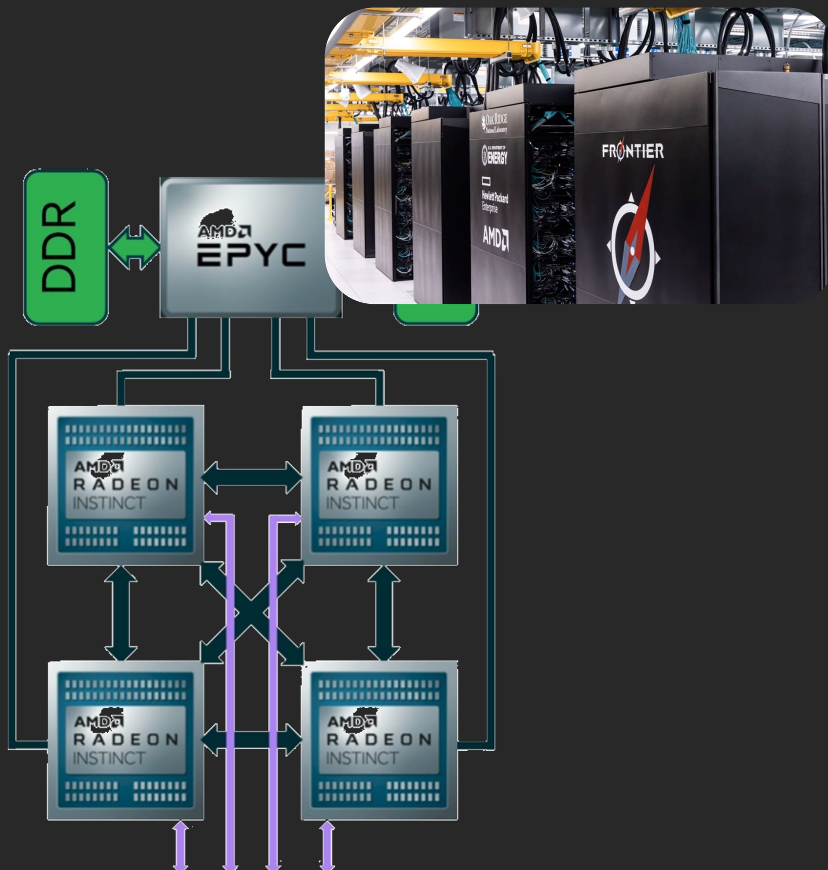
Professor Spencer Bryngelson and his team of researchers at Georgia Tech's School of Computational Science and Engineering were excited to bring their code into the new world of exascale-class supercomputing. The Multicomponent Flow Code, or MFC, originated in the early 2000s to model the simultaneous movements of gases and liquids as they transform through different thermodynamic phases, such as water boiling into steam. When exascale computing arrived some 20 years later, the freshly updated code was ready to leverage the latest advances in high-performance computing, or HPC.

These folks helped!

Today

Solving multiphase eng. problems

- A modified equation
- Exascale solves
- Example problem



A usual suspect

mflowcode.github.io

Compressible, multi-component

Shock/interface-capturing

More! [in MFC]

Sub-grid models, phase change,

surface tension, complex geom.,

chem. reactions, ...

$$\frac{\partial \alpha_i \rho_i}{\partial t} + \nabla \cdot (\alpha_i \rho_i \boldsymbol{u}) = 0,$$

$$\frac{\partial \rho \boldsymbol{u}}{\partial t} + \nabla \cdot (\rho \boldsymbol{u} \otimes \boldsymbol{u} + p\boldsymbol{I}) = 0,$$

$$\frac{\partial \rho E}{\partial t} + \nabla \cdot [(\rho E + p)\boldsymbol{u}] = 0,$$

$$\frac{\partial \alpha_i}{\partial t} + \boldsymbol{u} \cdot \nabla \alpha_i = -k \nabla \cdot \boldsymbol{u}$$

Bryngelson et. al *CPC* (2021), Wilfong et al. *arXiv* (2025)

Simplifying/removing “problems”

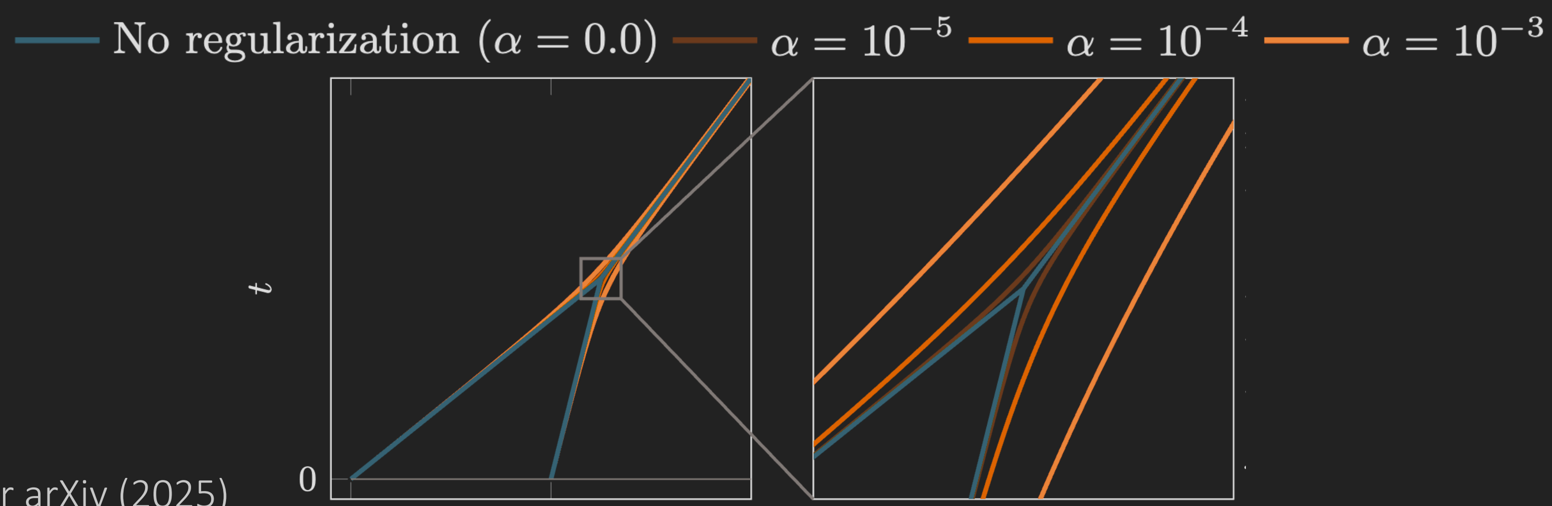
Information Geometric Regularization

Shock *capturing* entails artificial dissipation or limiters 👎

Avoid the problem

IGR:

$$\begin{cases} \partial_t \begin{pmatrix} \rho \boldsymbol{u} \\ \rho \end{pmatrix} + \vec{\operatorname{div}} \begin{pmatrix} \rho \boldsymbol{u} \otimes \boldsymbol{u} + (P(\rho) + \Sigma)\mathbf{I} \\ \rho \boldsymbol{u} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix} \\ \rho^{-1}\Sigma - \alpha \vec{\operatorname{div}}(\rho^{-1}\nabla\Sigma) = \alpha\left(\operatorname{tr}^2([\mathrm{D}\boldsymbol{u}]) + \operatorname{tr}([\mathrm{D}\boldsymbol{u}]^2)\right) \end{cases}$$



Cao & Schafer arXiv (2025)

Information Geometric Regularization

Shock *capturing* entails artificial dissipation or limiters 👎

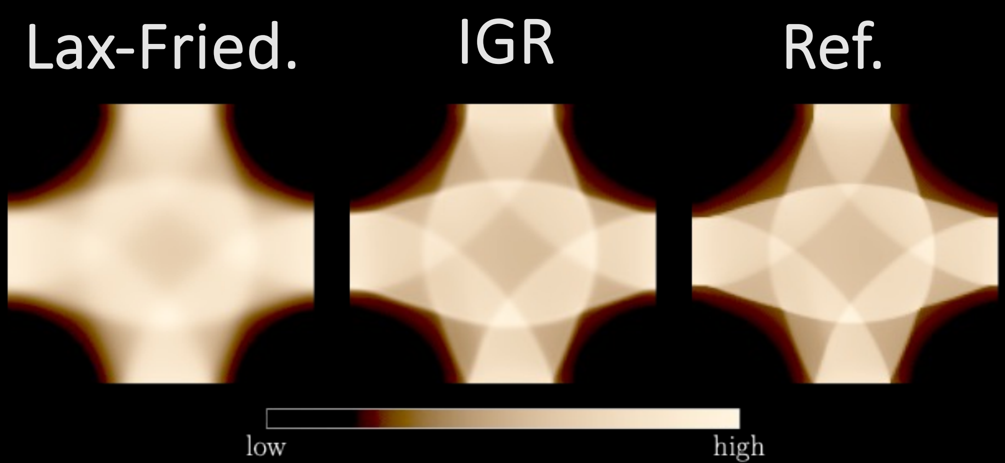
Avoid the problem

IGR:

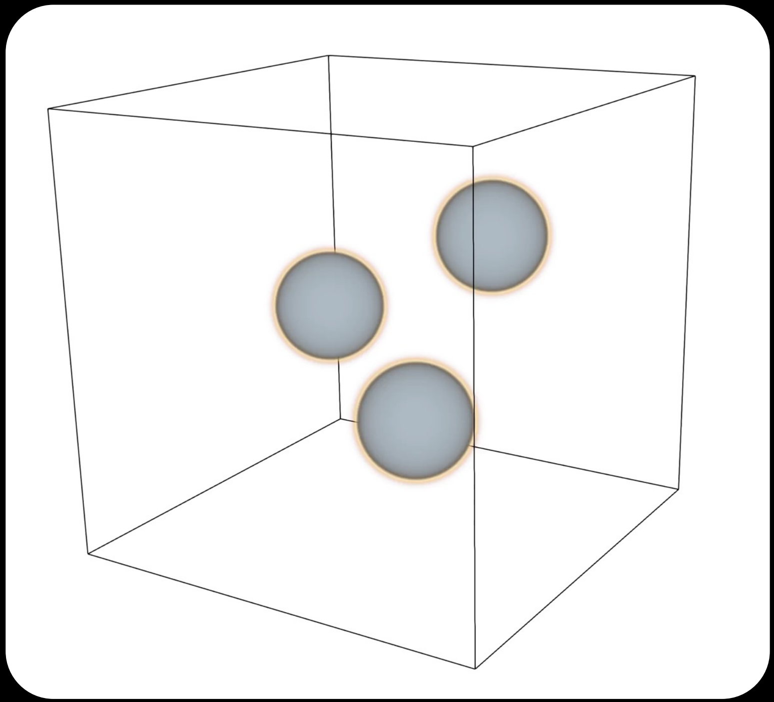
$$\begin{cases} \partial_t \begin{pmatrix} \rho \boldsymbol{u} \\ \rho \end{pmatrix} + \vec{\operatorname{div}} \begin{pmatrix} \rho \boldsymbol{u} \otimes \boldsymbol{u} + (P(\rho) + \Sigma)\, \mathbf{I} \\ \rho \boldsymbol{u} \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix} \\ \rho^{-1} \Sigma - \alpha \vec{\operatorname{div}}(\rho^{-1} \nabla \Sigma) = \alpha \left(\operatorname{tr}^2 ([\mathrm{D} \boldsymbol{u}]) + \operatorname{tr}\left([\mathrm{D} \boldsymbol{u}]^2\right)\right) \end{cases}$$

Now: Solve with linear finite differences, store no intermediaries

Regularization: Sedov blast problem

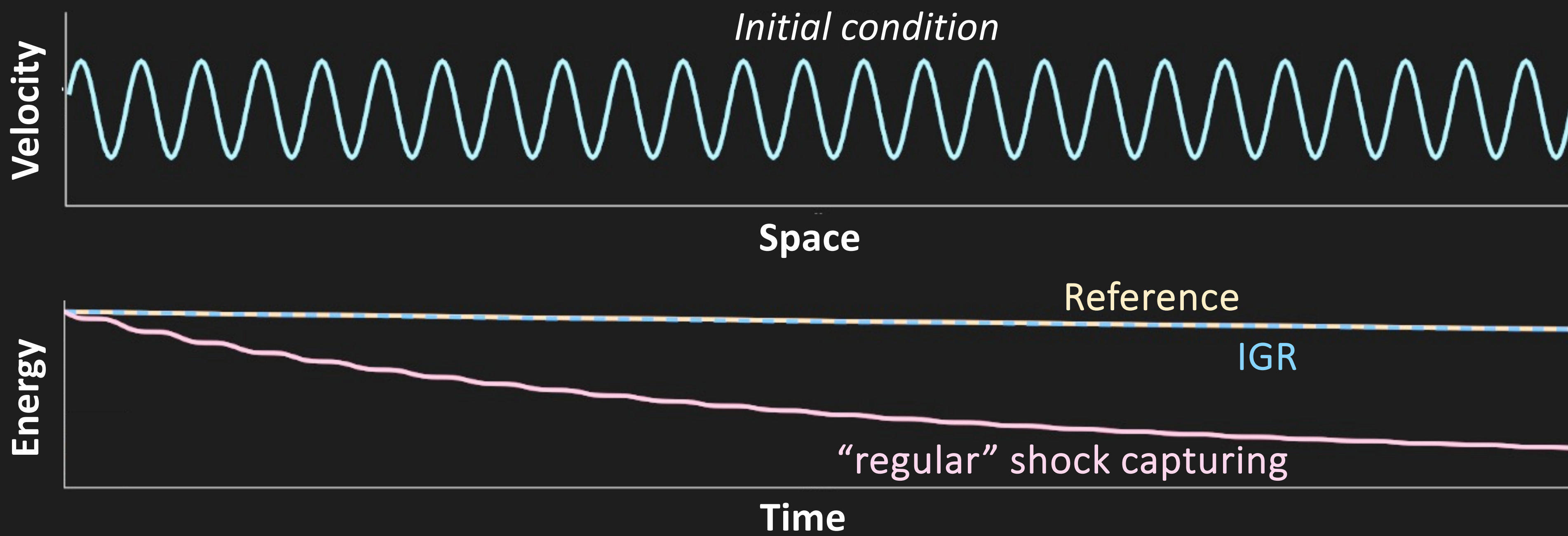


Better match to exact (ref.) sol'n



Maintain shocks and mixture quantities

Regularization: No systematic dissipation



No energy dissipation with IGR

Why? Never introduced viscous mechanism

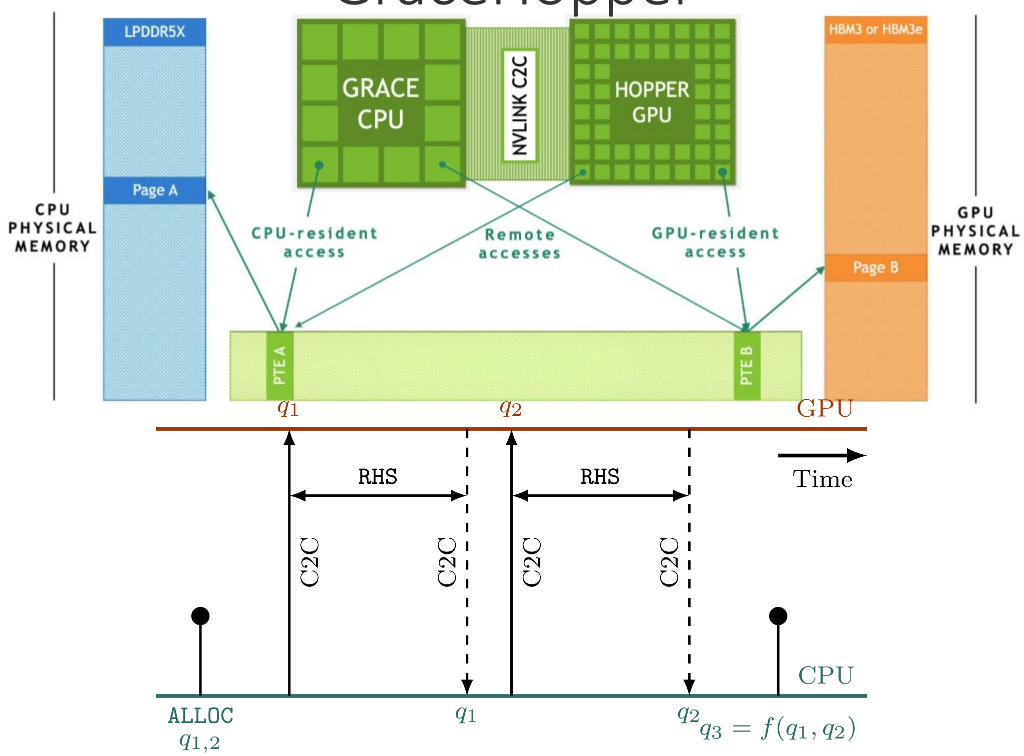
Regularization: So what?

Enables

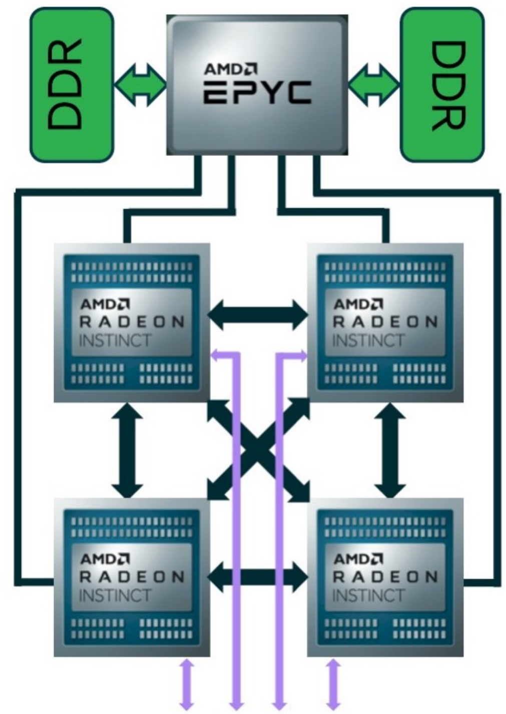
- Design ➡️ Sensitivities/adjoints across shocks
- Speed ➡️ No [W]ENO/Riemann solve, limiters, detectors
- Accuracy ➡️ High-order accurate and *no dissipation*
- Less memory ➡️ Larger sims. by two orders mag.

Going big (problems)

GraceHopper



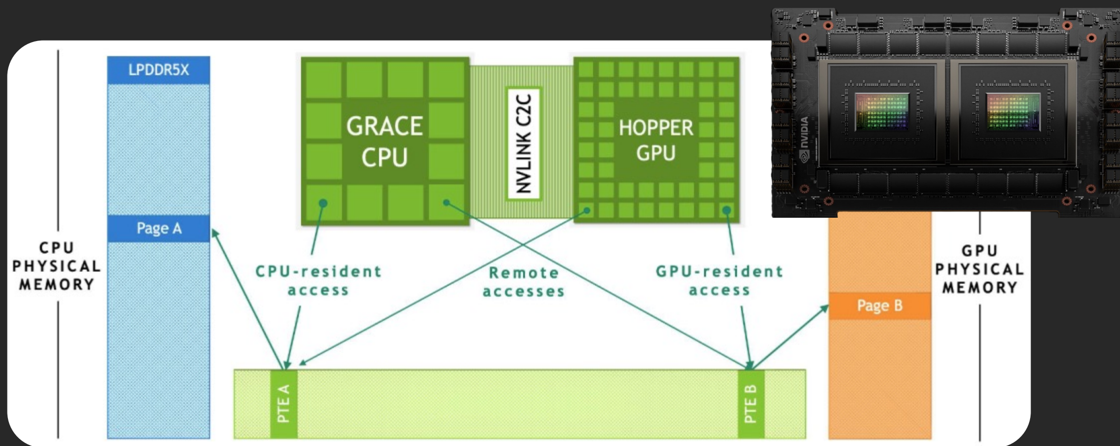
Trento–MI250X

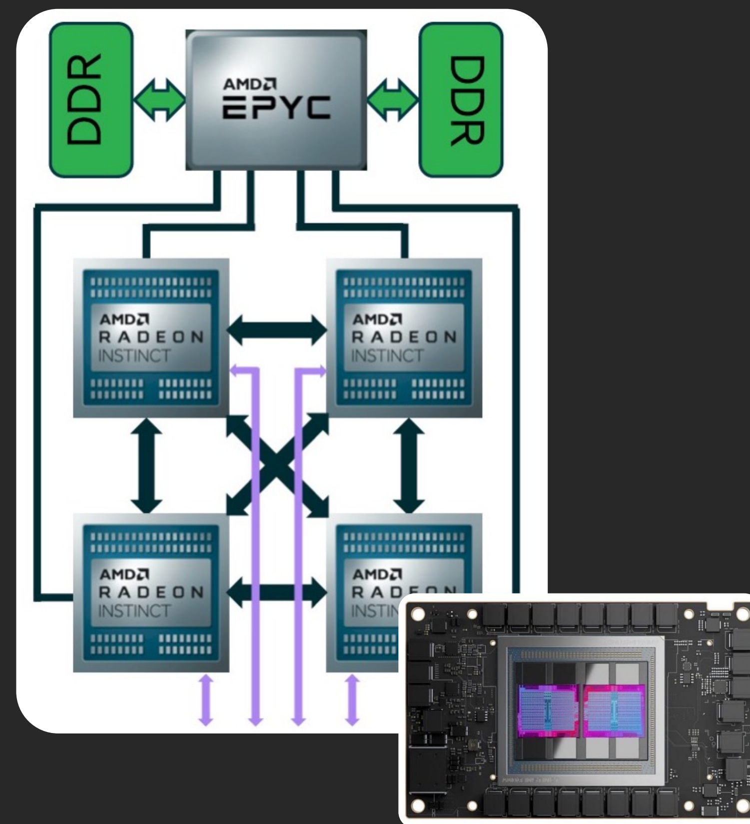


1 timestep stage in CPU DRAM, other in GPU HBM.

Zero-copy accesses via InfinityFabric or C2C

Going big (problems)



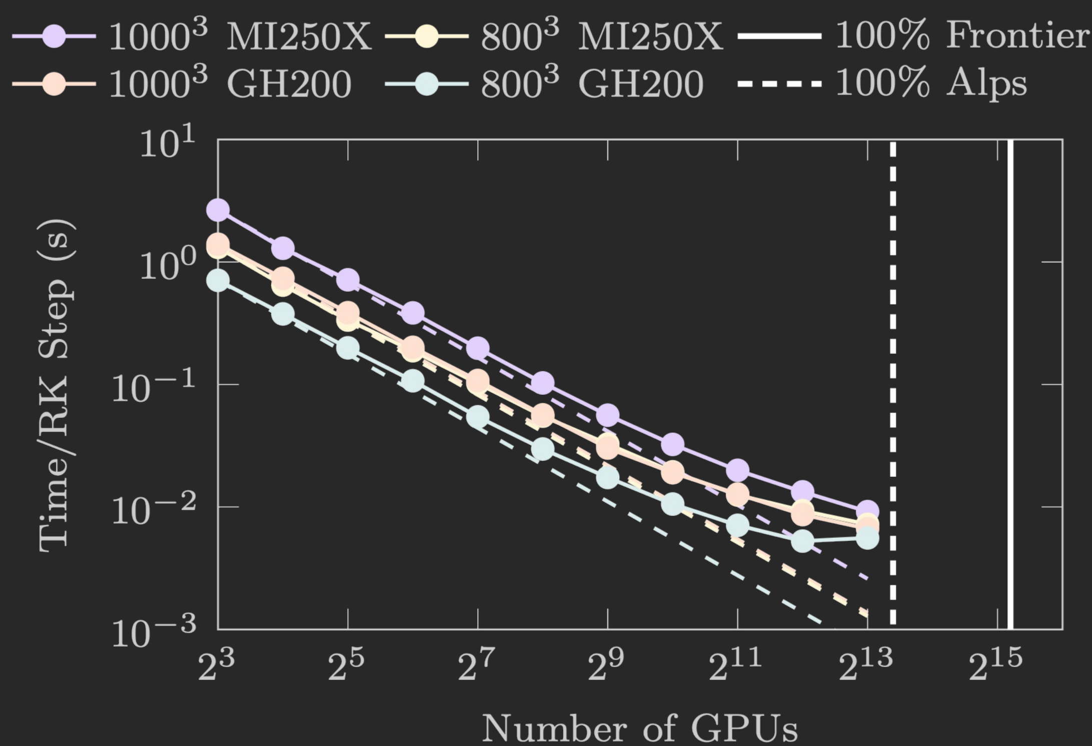


2557^3 grid points per Alps node
[4-way GH200] FP32/16

2780^3 grid points per Frontier node
[4-way MI250X] FP32/16

Performance: Strong scaling

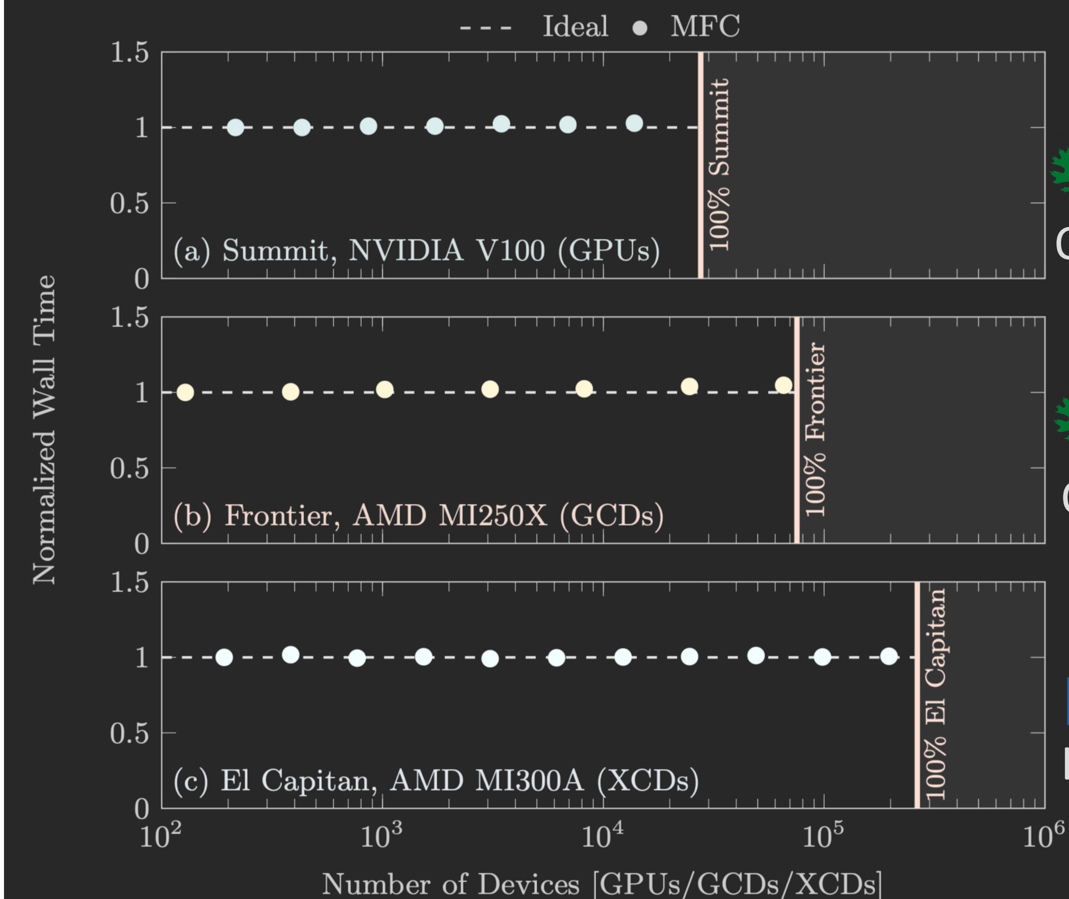
mflowcode.github.io



From 2 nodes to
~full system!

20B grid point sim.
~1 Frontier wall minute
[this is near real-time]

Performance: Weak scaling



OLCF Summit

OLCF Frontier

LLNL El Capitan

mflowcode.github.io

Ideal scaling

~1 EFLOPS

***200T** points* [Frontier]

1 quadrillion DOF

20x SoA

Performance: Speed and power

Wall-time performance

Device	Baseline	IGR (in-core)	IGR (unified)	
GH200	21.73	5.41	5.65	FP64
MI250X GCD	76.25	18.23	25.90	FP64
GH200	*N/A	4.27	4.46	FP32
MI250X GCD	*N/A	14.60	17.61	FP32

4-5x faster

mflowcode.github.io

Energy use

Energy (μJ)	Frontier	Alps
Baseline (WENO + HLLC)	165.76	49.88
IGR	48.08	8.80

4x less energy

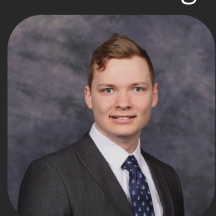
...all compared to optimized baseline in same code

A real example

mflowcode.github.io

A. Radhakrishnan

B. Wilfong

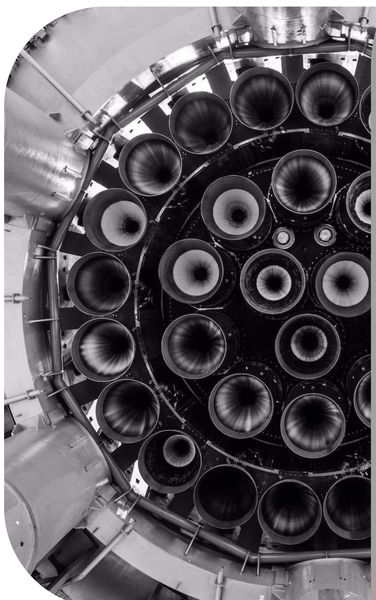


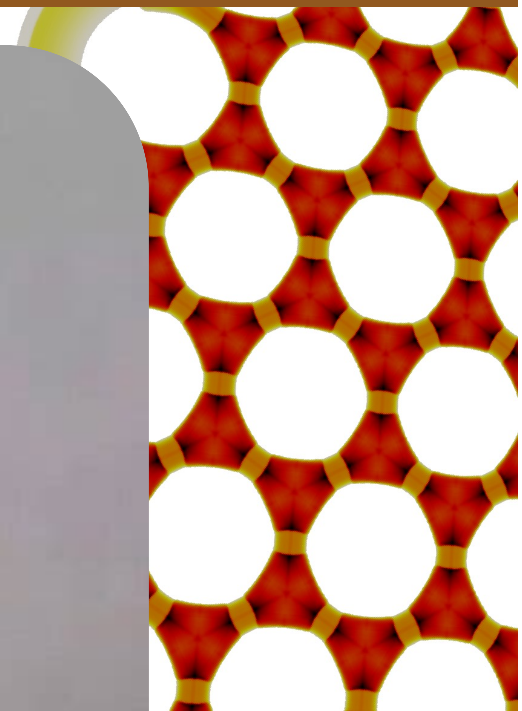
D. Adam

Example: Rocket engines

2-species (gas/gas) Mach 10 mock exhaust

Example: Rocket engines

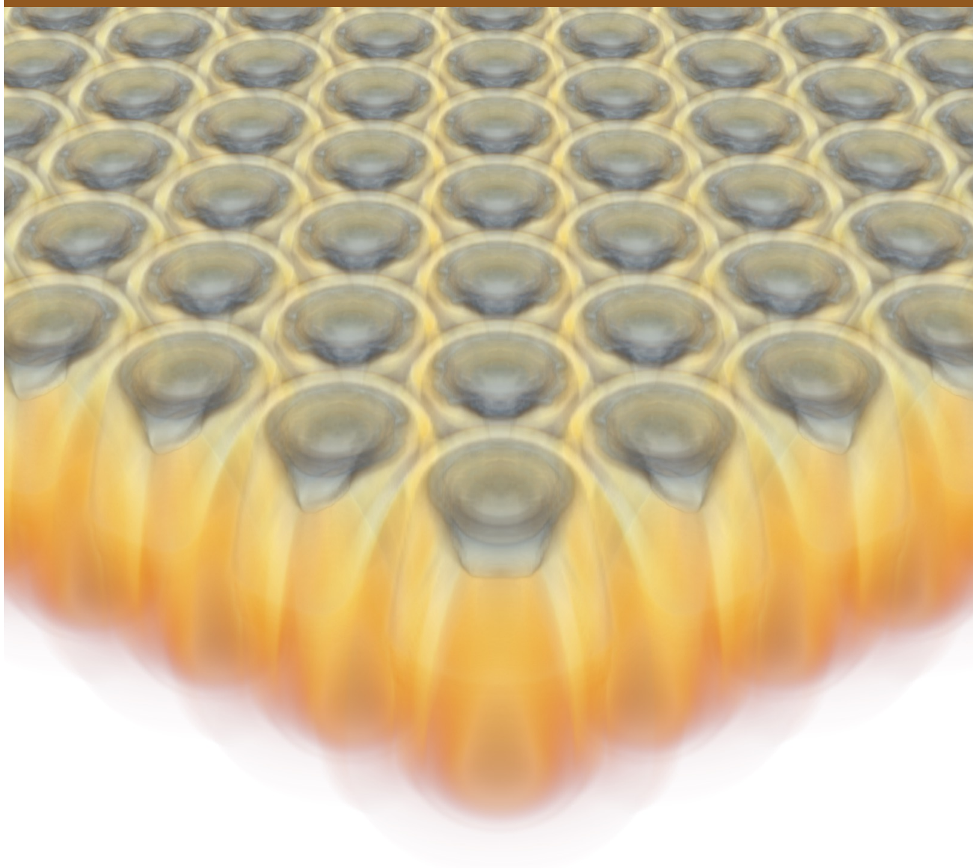




SpaceX Super

ter

Example: Rocket engines



~ 1000 rocket engine exhausts

200T grid points [ORNL Frontier]

5-state vars. – 1 quadrillion DoF

> 20-times larger than prior SoA

Take away

Remove "problems" in the equations, solve larger problems more efficiently

AMD NNSA

DEPARTMENT OF ENERGY UNITED STATES OF AMERICA

AIR FORCE OFFICE OF SCIENTIFIC RESEARCH UNITED STATES AIR FORCE

CRAY

Take away

Remove "problems" in the equations, solve larger problems more efficiently

Simulating many-engine spacecraft: Exceeding 100 trillion grid points via information geometric regularization and the MFC flow solver

Benjamin Wilfong[1], Anand Radhakrishnan[1], Henry Le Berre[1], Nikolaos Tselepidis[2], Benedikt Dorschner[2], Reuben Budiardja[3], Brian Cornille[4], Stephen Abbott[5], Florian Schäfer[1,§], Spencer H. Bryngelson[1,§]

This work: *arXiv:2505.07392*