Developing Digital Twins for Energy Applications Using Modulus

Tarak Nandi, NETL Support Contractor

Oliver Hennighn NVIDIA

NVIDIA GTC

March 2022

Disclaimer

NATIONAL ENERGY TECHNOLOGY LABORATORY

This project was funded by the United States Department of Energy, National Energy Technology Laboratory, in part, through a site support contract. Neither the United States Government nor any agency thereof, nor any of their employees, nor the support contractor, 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.

Authors and Contact Information

Tarak Nandi[1,3] Oliver Hennigh[2], Mohammad Nabian[2], Yong Liu[1,3], Mino Woo[1], Terry Jordan[1], Mehrdad Shahnam[1], Christopher Guenther[1], and Dirk Van Essendelft[1]

[1]National Energy Technology Laboratory, 3610 Collins Ferry Road, Morgantown, WV 26507, USA

[2]NVIDIA, Santa Clara, CA 95051

[3]NETL Support Contractor, 3610 Collins Ferry Road, Morgantown, WV 26507, USA

Introduction/Overview

NATIONAL ENERGY TECHNOLOGY LABORATORY

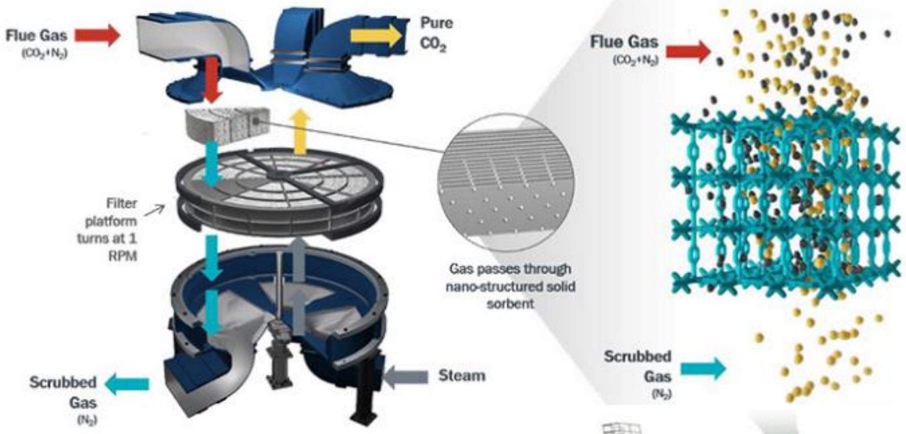
DOE Earthshots: Innovations using AI

★ **Towards net-zero energy economy**

- Reduce emission of greenhouse gases
- Design more energy-efficient systems (industrial/domestic)
- Capture CO_2 from point sources and air
- Enhanced oil recovery (EOR)
- Geological storage of CO_2
- All the above require fast development cycles

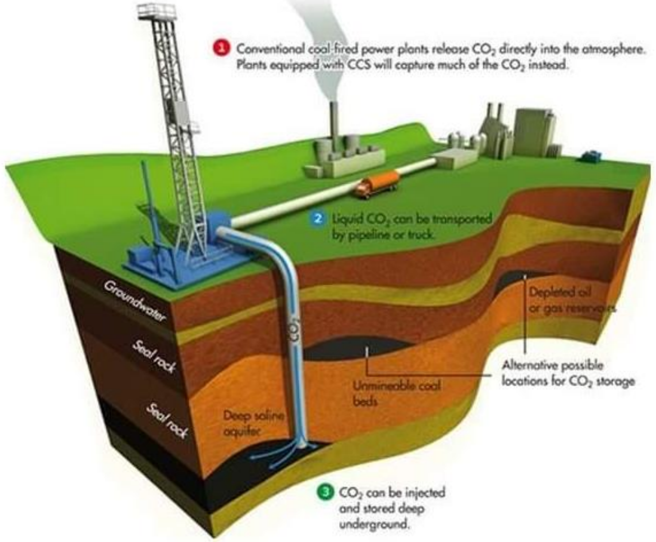
★ **Developing AI-based digital twins of engineering devices**

- Digital replica of engineering process/device
- Greatly accelerate the transition to net-zero
- Efficiently handle a large design space
- Develop and scale CCS devices
- Develop fuel efficient systems



Flue gas capture

Direct air capture



Underground carbon storage

Image sources: NETL Communications/Multimedia Services

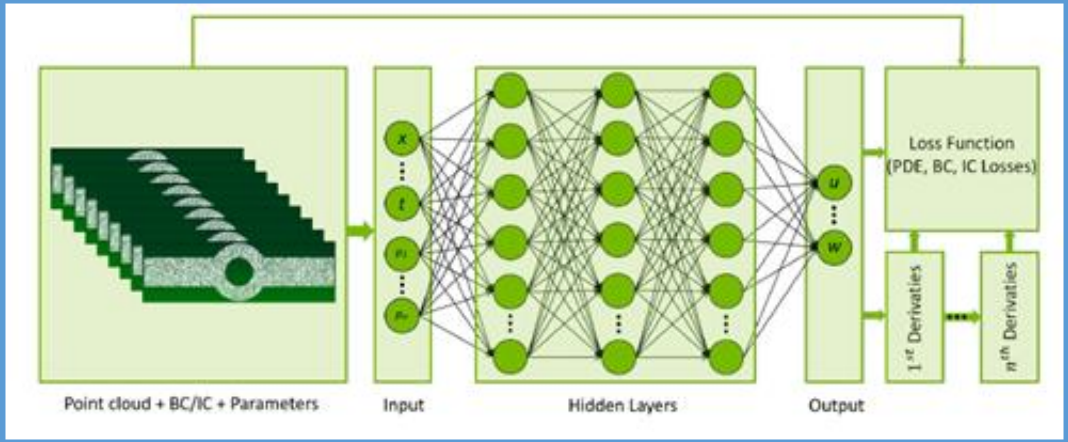
Physics Informed Neural Networks

Physics guided deep learning approach using Modulus from NVIDIA

NATIONAL ENERGY TECHNOLOGY LABORATORY

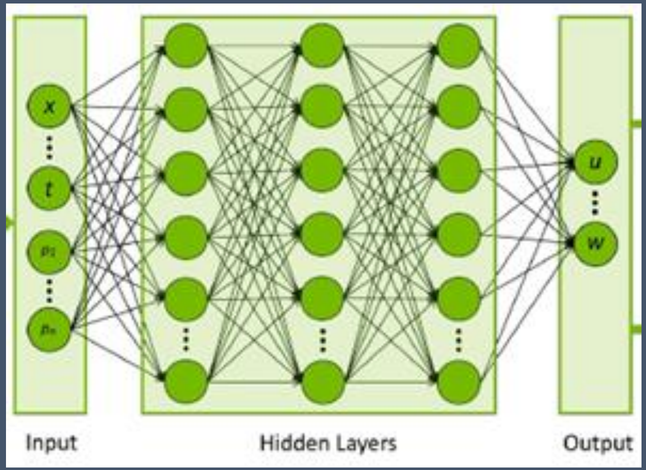
- Relates spatial & temporal coordinates (*x, y, z, t*) to physical fields (e.g., velocity, pressure, temperature)
- NNs trained to minimize the residual form of the same physical equations as in CFD
- Parameterized system representation can solve for multiple configurations simultaneously
- Near-instantaneous inference
- Orders of magnitude faster than CFD for UQ/design optimization problems

OFFLINE TRAINING over a parameter space



TRAINABLE NETWORK

ONLINE prediction for arbitrary conditions



FROZEN NETWORK

Figure source: Hennigh et al., "NVIDIA SimNet: An AI-accelerated multiphysics simulation framework" (2020)

Modulus: Salient Features

Modulus Model Training

- Uses TensorFlow 1.x but is being ported to PyTorch
- Exact differentiation
- Soft constraints on governing equations and BCs
- No requirement for structured grid format or grid connectivity information
- Can utilize advanced NN architectures like Fourier Network/SiReN
- Options for dynamic loss weighting (being tested now)

$$Loss(\boldsymbol{\theta}) = \Sigma w_i Loss_{interior} + \Sigma w_b Loss_{boundary}$$

Hyperparameters

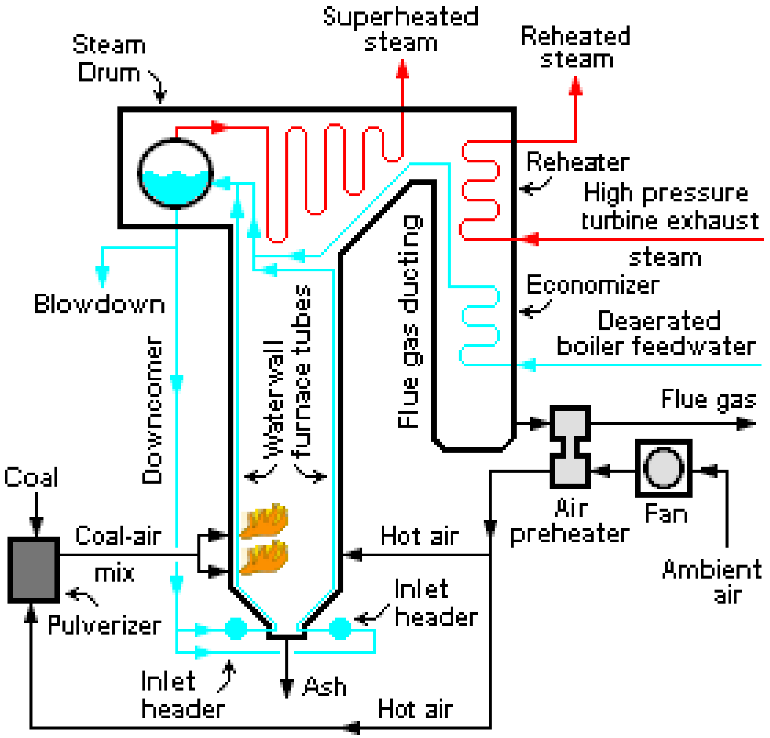
$Loss_{interior} = |LHS_{gov} - RHS_{gov}|^{order}$ $Loss_{boundary} = |LHS_{BC} - RHS_{BC}|^{order}$

PINN for Reacting Flows

Formulation and PINN vs CFD

- **Aim:** Create a digital twin of an industrial scale boiler
- Simplified methane oxidation
- Implemented reacting flow transport equations for kinetics-controlled combustion
- No requirement for training data

★ Single PINN model for a range of input conditions
★ Fidelity and accuracy comparable to CFD
★ Trained PINN can provide near-instantaneous inference for any input condition



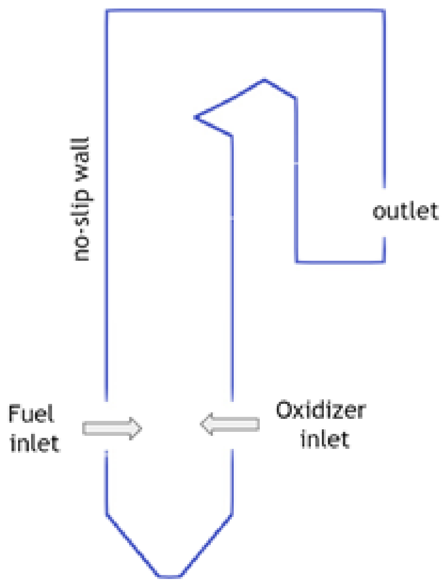


Figure source: https://commons.wikimedia.org/wiki/File:Steam_Generator.png

Towards a Reacting Flow Solver

Governing Equations: Strongly Coupled PDEs

- Single species
- Turbulent
- Steady state
- 2D

→

- **Multiple species**
- Turbulent
- Steady state
- 2D

→

- Multiple species
- **Reacting**
- Turbulent
- Steady state
- 2D

→

- Multiple species
- Reacting
- Turbulent
- **Transient**
- 2D

→

- Multiple species
- Reacting
- Turbulent
- Transient
- **3D**

- Continuity: $\frac{\partial \rho}{\partial t} + \frac{\partial (\rho u_i)}{\partial x_i} = 0$
- Species mass fraction: $\rho \frac{\partial Y_k}{\partial t} + \rho u_i \frac{\partial Y_k}{\partial x_i} = \dot{\omega}_k + \frac{\partial}{\partial x_i}\left(\rho D_k \frac{\partial Y_k}{\partial x_i}\right)$
- Momentum: $\frac{\partial (\rho u_i)}{\partial t} + \frac{\partial (\rho u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{ij}}{\partial x_j}$
- Temperature: $\frac{\partial T}{\partial t} + \frac{\partial}{\partial x_i}(u_j T) = \frac{\dot{\omega}_T}{\rho c_p} + \frac{\partial}{\partial x_i}\left(\alpha \frac{\partial T}{\partial x_i}\right)$
- Kinetics-controlled single step irreversible reaction: $CH_4 + 2(O_2 + 3.76N_2) \rightarrow CO_2 + 2H_2O + 7.52N_2$
- Species source/sink terms: $\dot{\omega}_{CH4} = -MW_{CH4} k_f \left(\frac{\rho Y_{CH4}}{MW_{CH4}}\right) \left(\frac{\rho Y_{O2}}{MW_{O2}}\right)$ etc
- Temperature source term: $\dot{\omega}_T = -\sum_{k=1}^{N} h_k \dot{\omega}_k = -\sum_{k=1}^{N} h_{sk} \dot{\omega}_k - \sum_{k=1}^{N} \Delta h_{f,k}^0 \dot{\omega}_k$

Parametric Boundary Conditions

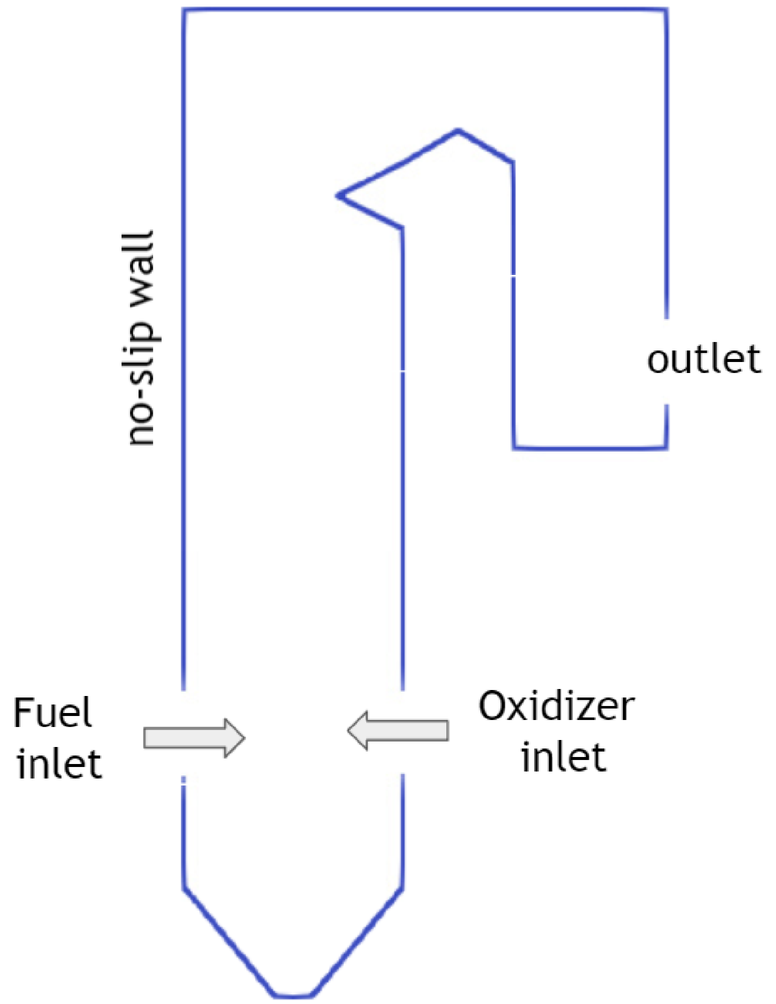
Species Mass Fractions, Velocity and Temperature

Inlet Conditions

	Fuel inlet	Oxidizer inlet
Y_ch4	0.5	0.0
Y_o2	0.0	0.23
Y_co2	0.01	0.01
Y_h2o	0.01	0.01
Y_n2	0.48	0.75
Velocity, m/s	1.0	**1.0 – 5.0**
Temperature, K	650	650

Wall Conditions

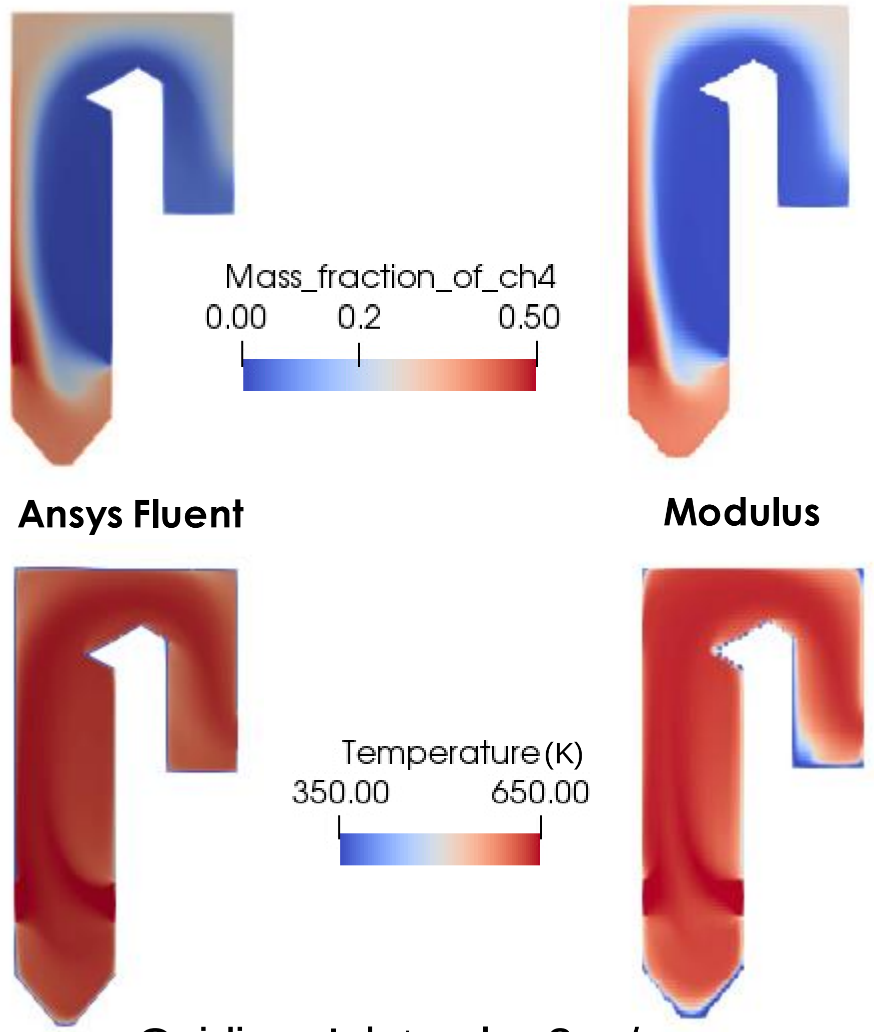
Temperature, K	350.0
Species	Zero flux
Velocity	No-slip



Multi-Species Nonreacting Flow

CFD vs. PINN

- 3 non-reacting species (CH_4, O_2, N_2)
- Species (Y) and Temperature (T) distributions compare well with ANSYS Fluent
- Improved predictions after normalization of the governing equations and BCs

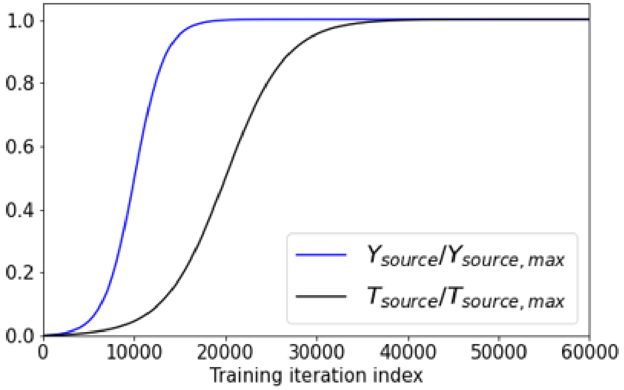


Oxidizer Inlet vel = 2 m/s

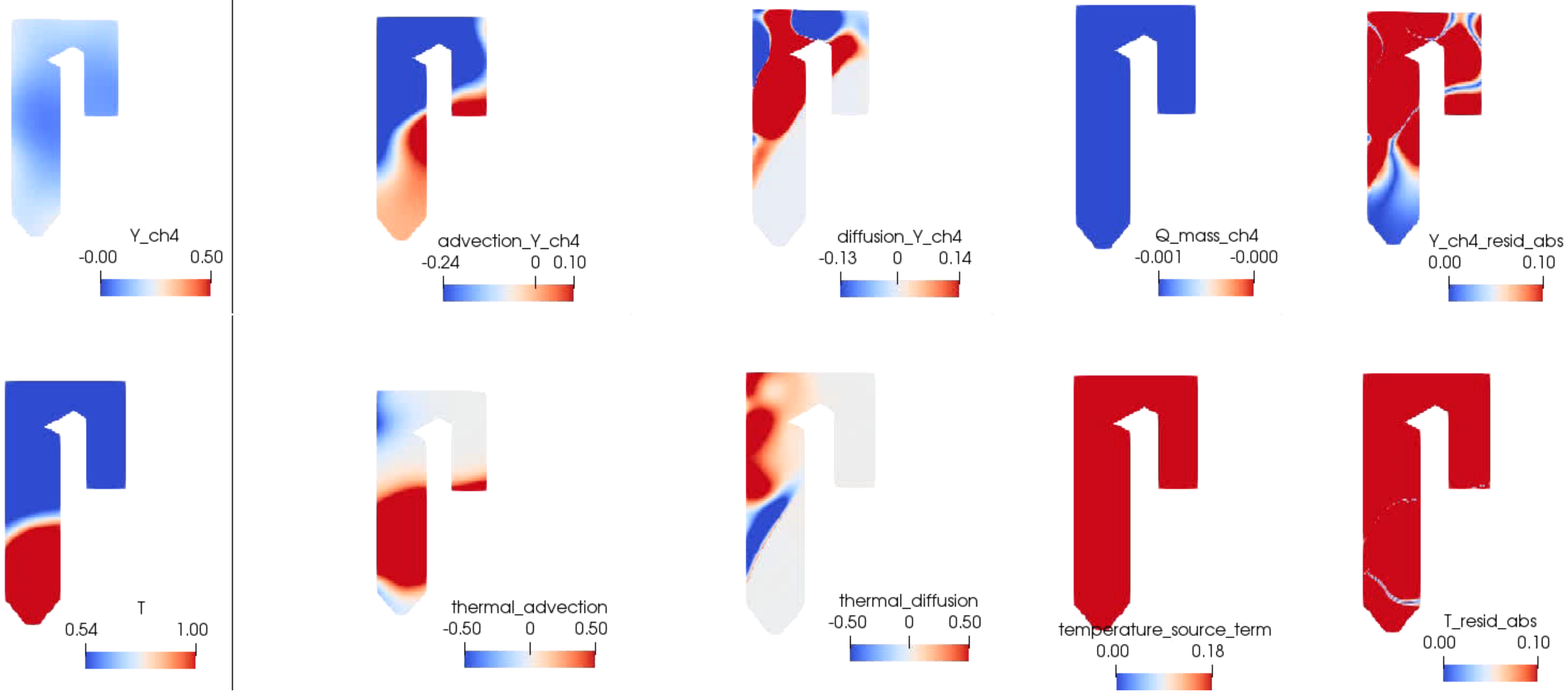
First Attempt at Reacting Flow Solver

Handling Y-T coupling

- Strongly coupled PDEs
- Y and T source terms allowed to increase gradually



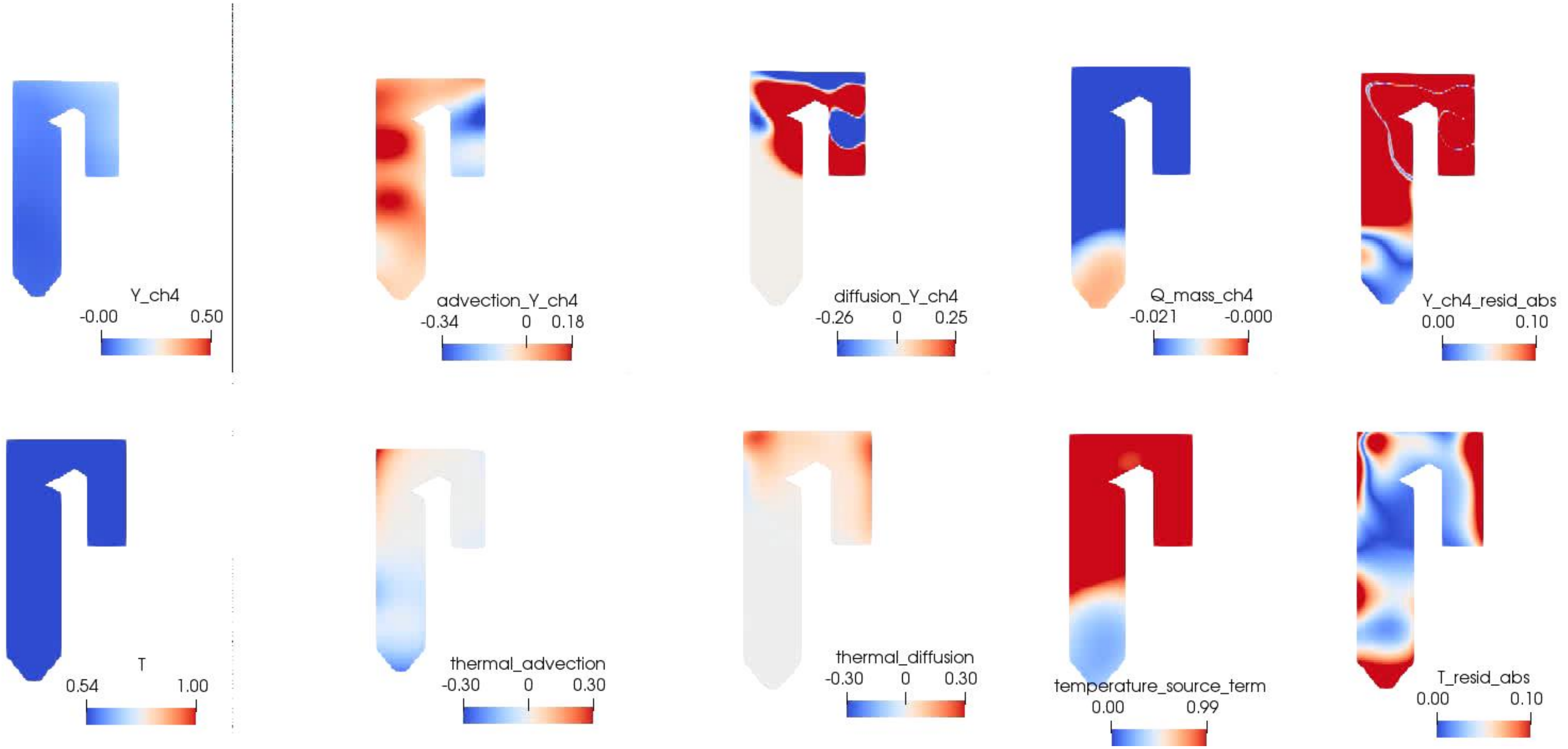
- Y distribution goes unphysical after initial increase
- Unphysical Y distribution affects the T field



Without T-Source

Simplified system

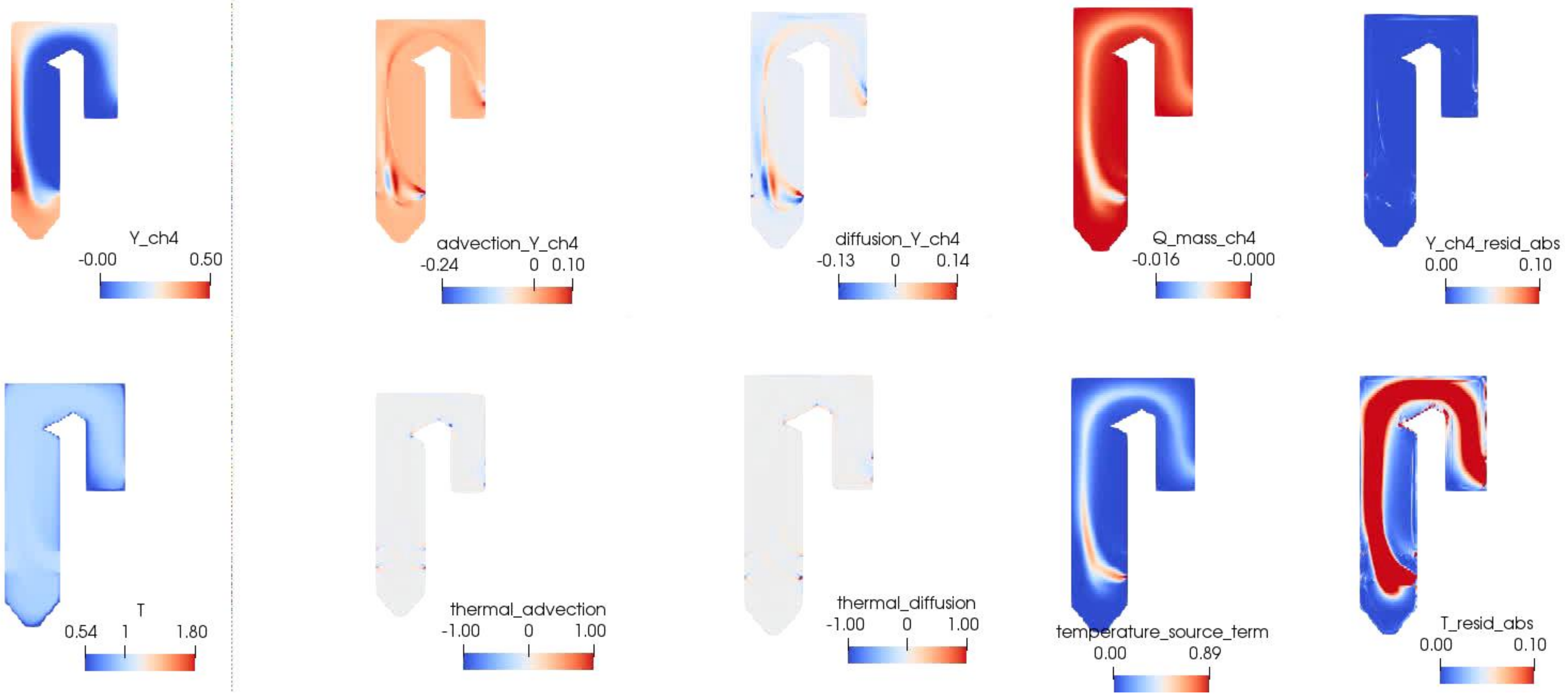
- T source removed from the system for the time being
- The reactant and product distributions are reasonable
- This field is used as IC for subsequent cases



One Way Y-T Coupling

Frozen Species Field

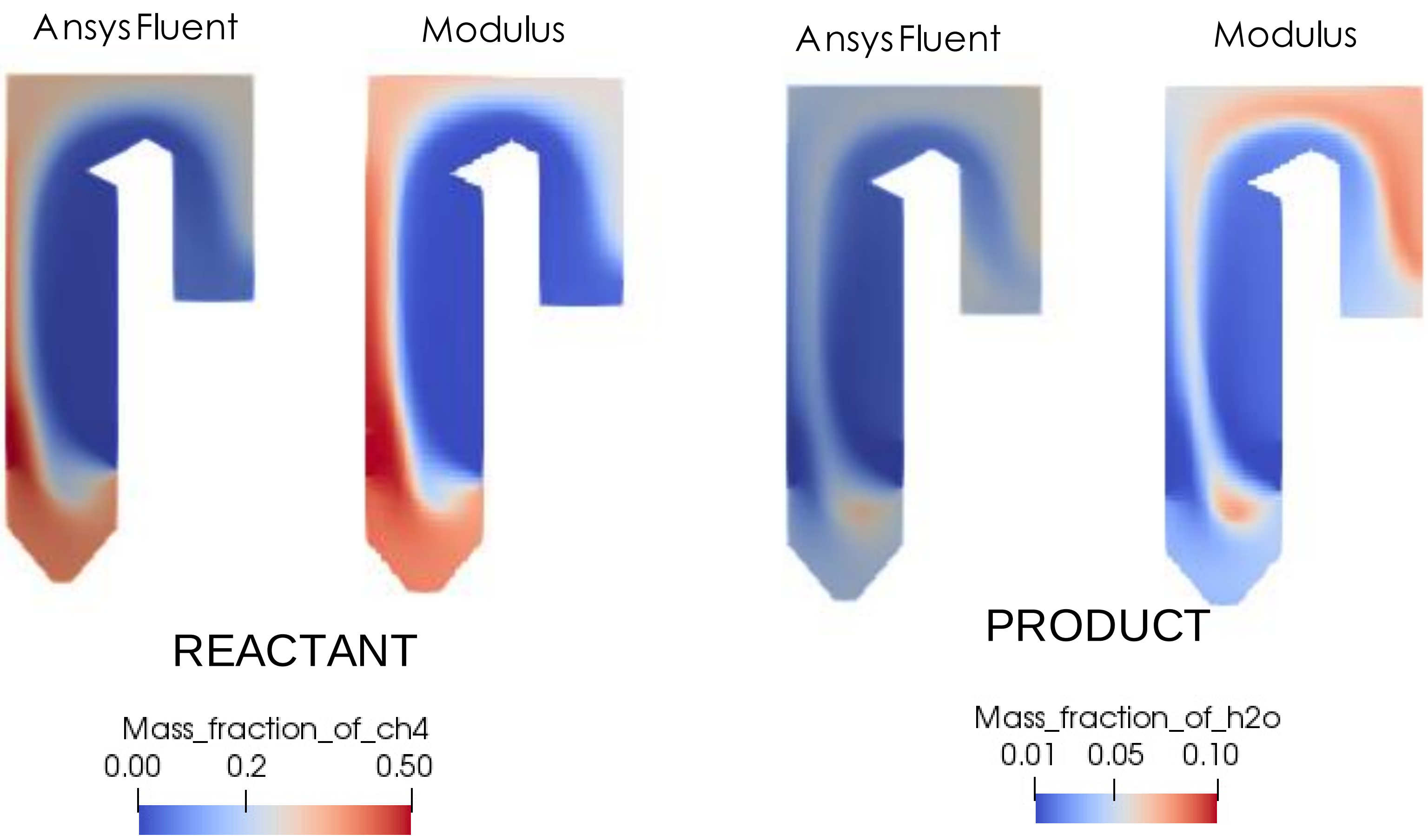
- Species not allowed to evolve from the IC
- T-source reasonably developed at the reaction zone
- Decoupled T and Y fields can lead to severe errors



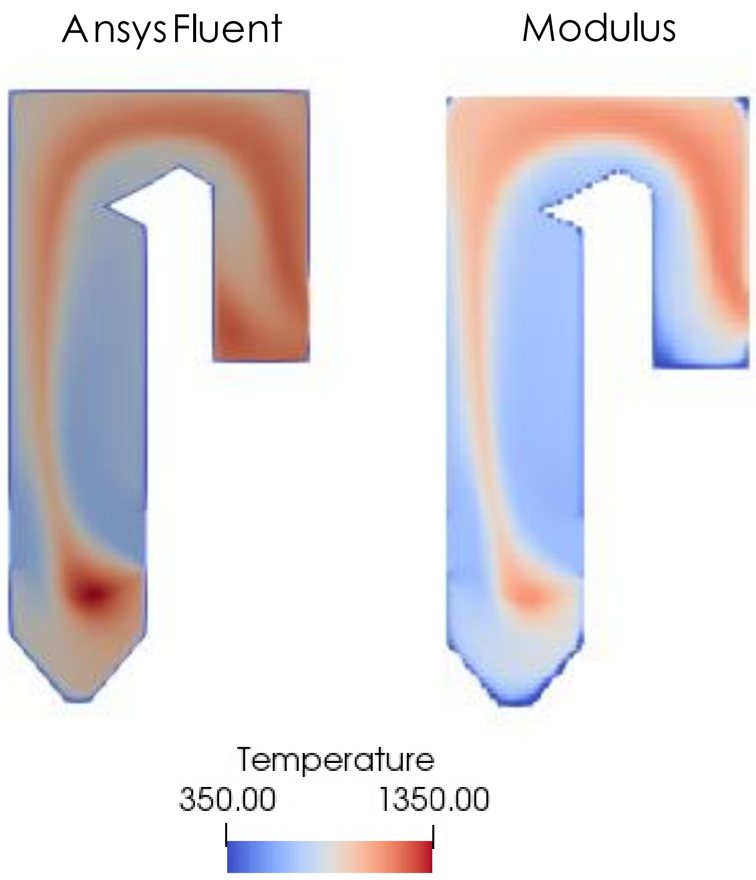
Simplified Systems

PINN vs. CFD

Species distributions for case without T source term

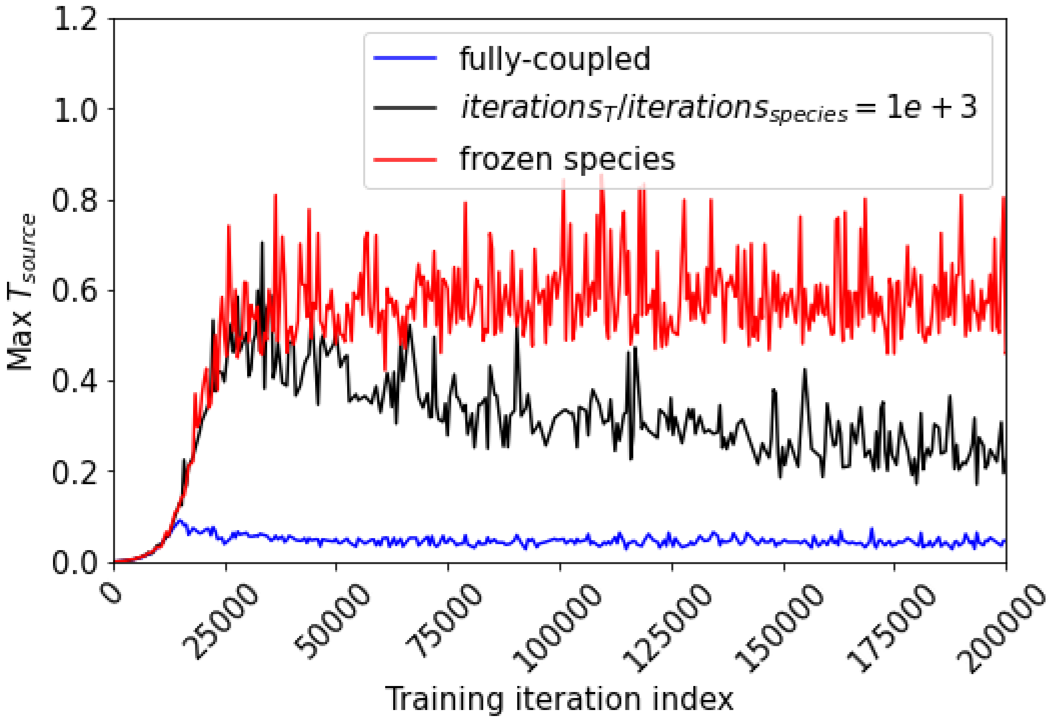


Temperature distribution with frozen species



Increasing Y-T Coupling

- To increase the Y-T coupling, Y solved for every 100-1000 T iterations
- Issue persists: T-source drops off



Resolving the issue with large T-source

Handling large T-source

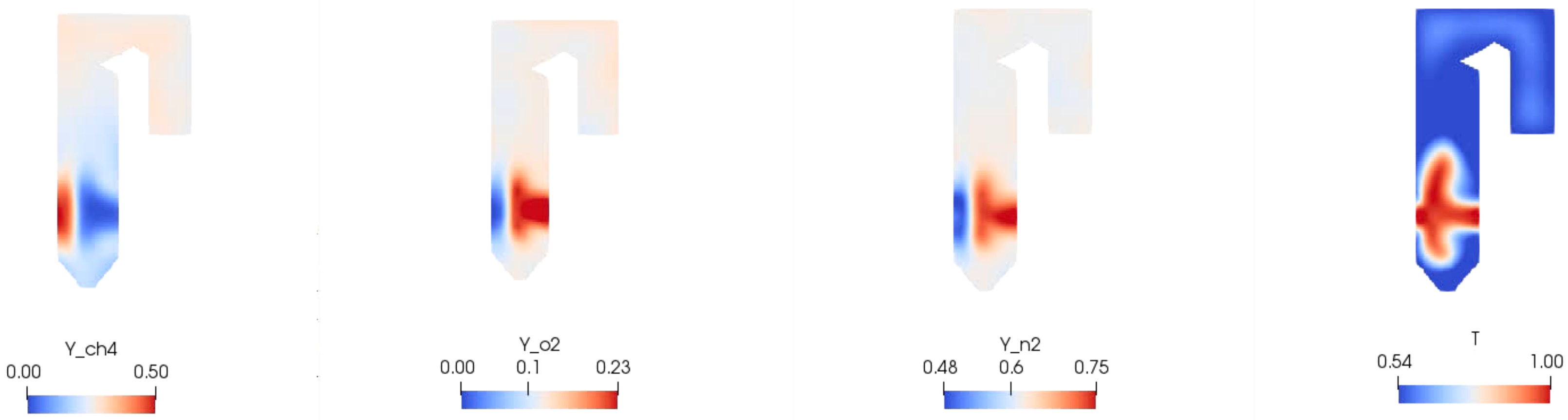
- T-source dominates the Y-source
- This can lead to imbalances between the backpropagated gradients

A) Gradient normalization approach

- Attempts to remove the dominance of any component of the global loss function
- Dynamically assigns weights to different constraints

B) Transient approach

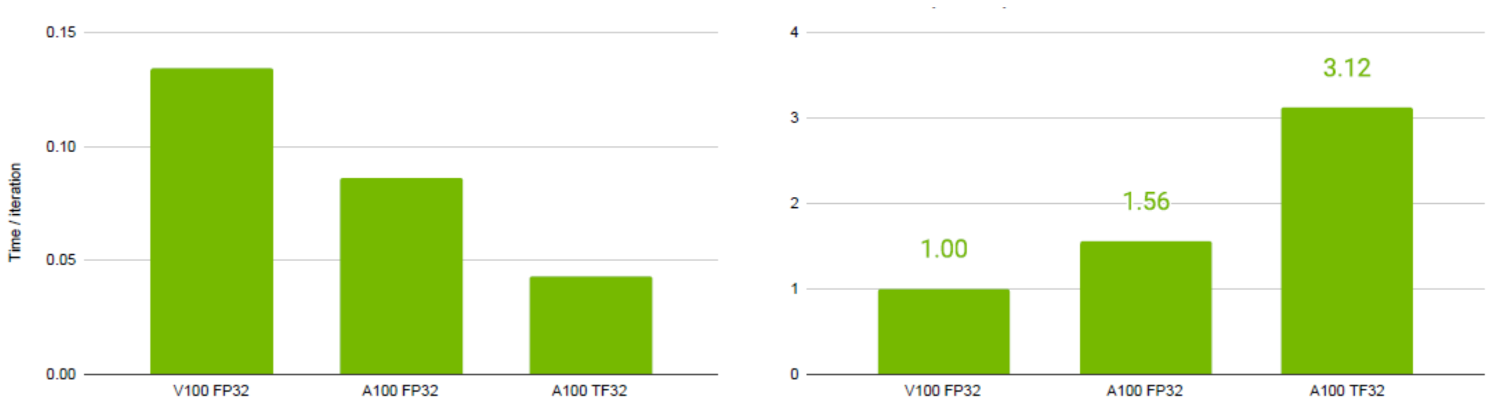
- Handles large source terms by learning the change between states instead of learning everything at once
- Uses a moving time window approach



TF32 vs. FP32 and Scaling Study

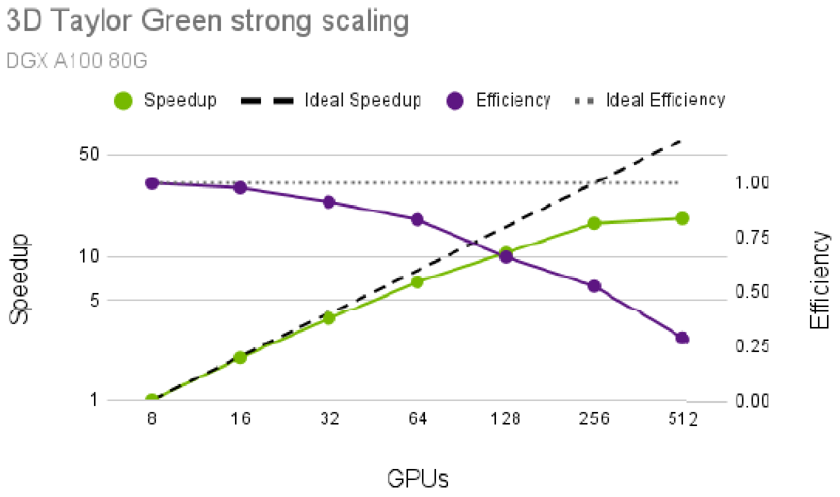
Math Mode & Strong Scaling Performance

- Boiler cases ran on NVIDIA P100 GPUs
- Performance metrics available for sample problems using TF32 on A100 GPUs
- Good multi-node scaling obtained using Horovod for a 3D decaying turbulence case



(a) Time per iteration (b) Speed-up

Accelerated training using TF32 on A100 GPUs



Strong Scaling- Speedup and Scaling Efficiency

Next Steps

Short and Long-Term Goals

- Transition to PyTorch framework from TensorFlow
- Develop a PINN methodology for stiff chemistry
- Develop an RNN-based PINN for transient problems
- Integration with NVIDIA tools for deployment as digital twin

Thank You!

NETL
RESOURCES

VISIT US AT: **www.NETL.DOE.gov**

@NETL_DOE

@NETL_DOE

@NationalEnergyTechnologyLaboratory