Efficient Physics-Informed Machine Learning Algorithms for Flow Problems

Compressible/Incompressible Flows

Tulin Kaman and Xuan Gu

https://kaman.uark.edu
Computational and Applied Mathematics Group
Department of Mathematical Sciences
University of Arkansas, Fayetteville, AR, USA

February 27, 2024

Turbulent Mixing

Rayleigh–Taylor Instability (RTI):

arises at the interface between two fluids of different densities whenever the pressure gradient opposes the density gradient.

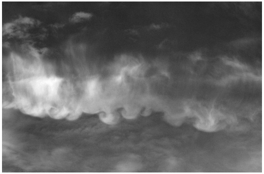
Richtmyer-Meshkov instability (RMI):

occurs when a shock wave passes through a perturbed interface.

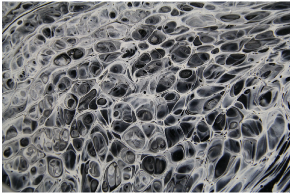
- An idealized subproblem of important scientific and engineering problems:
- crucial in all forms of fusion whether the confinement be magnetic, inertial or gravitational
- Supernovae explosions, inertial confinement fusion, gravitational induced mixing in oceanography

Turbulent Mixing

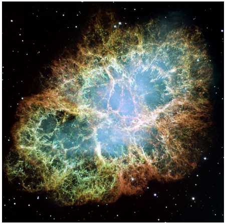
Characterize the structure and the evolution of the flows



Rayleigh–Taylor cirrus clouds. Source: Photograph courtesy of Prof. David Jewitt, UCLA.



Pattern created using then ''accidental painting'' technique developed by Siqueiros in the 1930s. It is the result of a RTI of a viscous layer. Source: From Fig. 1 of de la Calleja et al., Phys. Fluids.



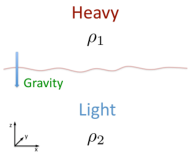
A composite Hubble Space Telescope image of the Crab Nebula. Source: NASA Space Telescope Science Institute, Baltimore, Maryland, USA.

Problem statement

$$h_i^{RTI} = \alpha_i A g t^2$$
$$h_i^{RMI} = \beta_i t^{\theta_i}$$

$$\begin{cases} i = b & \text{for bubbles (light fluid)} \\ i = s & \text{for spikes (heavy fluid)} \end{cases}$$

(α_b, α_s) and (θ_b, θ_s) growth rates of RTI and RMI mixing zone.
h_b: penetration distance of the light fluid into the heavy fluid
h_s: penetration distance of the heavy fluid into the light fluid
A: Atwood ratio $= (\rho_1 - \rho_2)/(\rho_1 + \rho_2)$, g: acceleration



Simple planar RT instability under gravitational acceleration.

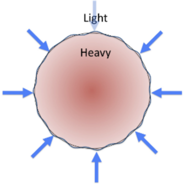


Diagram depicting the RT unstable interface created in an ICF fuel capsule.

Numerical Approaches to Model Turbulent Flows

Three levels of numerical simulation for turbulent flows:

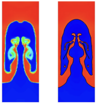
- Direct Numerical Simulation (DNS)
  - The full NSE is solved without any model for turbulence
  - The most demanding method among the three, very accurate, but limited to moderate Reynolds numbers and simplified geometries
- Large Eddy Simulation (LES)
  - Flow field is resolved down to a certain length scale, and scales smaller than that are modeled rather than resolved
  - Computational cost higher than RANS, but much lower than DNS
- Reynolds Averaged Navier Stokes (RANS)
  - Time-averaged equations solving for the mean values of all quantities
  - The least demanding in terms of resources

UNIVERSITY OF ARKANSAS

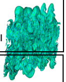
Turbulence Mixing and Combustion Simulations

Front Tracking Method (FronTier) to achieve resolution of steep and sharp density gradients

- RTI single mode simulation at t=10s. Comparison of a FLASH run without (left) and with (right) front tracking



Use of front tracking API - FTI

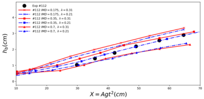


- Initial Conditions

Uncertainty Quantification(UQTk@SNL)

Effect of input parameters on the growth rate

T. Kaman, Model calibration for Turbulent Mixing Simulations, Proceedings of 16th International Workshop on the Physics of Compressible Turbulent Mixing, pp.129-134, Marseille, France.



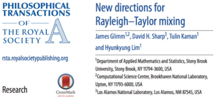
- Initial Perturbations

Long-wavelength perturbations:

Could be present in the initial data ?

Could contribute to the self similar growth constant a by a factor of two or more?

Modal Analysis of experimental data of Smeeton Youngs 87


PHILOSOPHICAL TRANSACTIONS OF THE ROYAL SOCIETY A

rsta.royalsocietypublishing.org

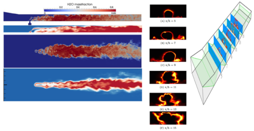
Research

New directions for Rayleigh–Taylor mixing

James Glimm[1,2], David H. Sharp[3], Tulin Kaman[1] and Hyunkyung Lim[1]

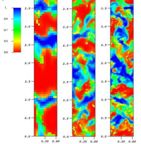
[1]Department of Applied Mathematics and Statistics, Stony Brook University, Stony Brook, NY 11794-3600, USA
[2]Computational Science Center, Brookhaven National Laboratory, Upton, NY 11793-6000, USA
[3]Los Alamos National Laboratory, Los Alamos, NM 87545, USA

- Verification and Validation study of Large Eddy Simulations of turbulent mixing and combustion within the engine of a scramjet", in collaboration with Stanford University's Predictive Science Academic Alliance Program Center, *Continuous Dynamical Systems, 36(8), pp. 4383 - 4402, 2016.*

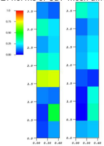


- w* convergence to a Young measure valued solution
  - Extract Probability Distribution Functions (PDF) by binning results. 8x2 supercell grids. For each supercell, bin the concentration values
  - Capture the local fluctuations of the solutions
  - Study of Convergence of PDF and associated Cumulative Distribution Functions (CDF)

Heavy fluid concentration at the midplane t = 50



L1 norms of CDF mesh differences



- Performance of **FronTier** Simulation Package - Argonne's IBM Blue Gene/P supercomputer and scale it to entire system - 163,840 cores.
- 2011 DoE INCITE Award Project “Uncertainty Quantification for Turbulent Mixing”
- 2012 DoE INCITE Award Project “Stochastic w* convergence for Turbulent Combustion”
- 2024-2026 NSF CC** Campus Compute: A High-Performance Computing System for Research and Education in AR

Objective: Use deep learning for solving time-dependent PDEs

- Euler equations for compressible, inviscid flows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho u) = 0 \ , \tag{1}$$

$$\frac{\partial (\rho u)}{\partial t} + \nabla \cdot (\rho u u^T) + \nabla p - f = 0 \ , \tag{2}$$

$$\frac{\partial E}{\partial t} + \nabla \cdot (u(E + p)) \ = 0 \ , \tag{3}$$

- Navier–Stokes equations for incompressible, viscous flows

$$\frac{\partial u}{\partial t} + (u \cdot \nabla) u - \nu \Delta u + \nabla p \quad = 0 \tag{4}$$

$$\nabla \cdot u \quad = 0. \tag{5}$$

Physics-Informed Neural Networks (PINNs)

- Replace numerical methods with a neural network that approximates the solution
- Embed a PDE into the loss of the neural network using automatic differentiation technique Rumelhart, Hinton, and Williams, Learning representations by back- propagating errors, Nature, 323 (1986)
- Improve the training efficiency of PINNs using the residual-based adaptive refinement (RAR) method
- Implementation in the Python library DeepXDE

Raissi, Perdikaris, Karniadakis, JCP 378, 2019.

Lu, Meng, Mao, Karniadakis, SIAM Review 2021.

Deep Neural Networks

- Feed-Forward NN/ Multi-Layer Perceptrons, Convolutional NN, Recurrent NN, Radial Basis Functional NN, ...
- Classified depending on their: structure, data flow, neurons used and their density, layers and their depth activation filters etc.
- L-layer neural network $\mathcal{N}^{\mathrm{L}}(\mathrm{x}) : \mathbb{R}^{\mathrm{d_{in}}} \to \mathbb{R}^{\mathrm{d_{out}}}$ with N_ℓ neurons in the ℓth layer.
- The weight matrix $\mathrm{W}^\ell \in \mathbb{R}^{\mathrm{N}_\ell \times \mathrm{N}_{\ell-1}}$ and bias vector $\mathrm{b}^\ell \in \mathbb{R}^{\mathrm{N}_\ell}$ in the ℓth layer
- σ activation function the logistic sigmoid, the hyperbolic tangent, the rectified linear unit.

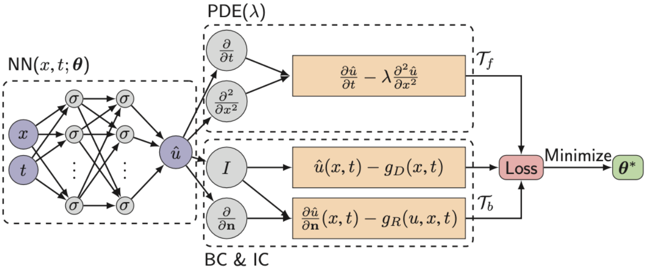
$$\begin{aligned}
\text{intput layer} \quad & \mathcal{N}^0(\mathrm{x}) = \mathrm{x} \in \mathbb{R}^{\mathrm{d_{in}}} \\
\text{hidden layers} \quad & \mathcal{N}^\ell(\mathrm{x}) = \sigma(\mathrm{W}^\ell \mathcal{N}^{\ell-1}(\mathrm{x}) + \mathrm{b}^\ell) \in \mathbb{R}^{\mathrm{d}_\ell} \text{ for } 1 \le \ell \le \mathrm{L}-1 \\
\text{output layer} \quad & \mathcal{N}^{\mathrm{L}}(\mathrm{x}) = \mathrm{W}^{\mathrm{L}} \mathcal{N}^{\mathrm{L}-1}(\mathrm{x}) + \mathrm{b}^{\mathrm{L}} \in \mathbb{R}^{\mathrm{d_{out}}}
\end{aligned}$$

Automatic Differentiation (AD)

- Compute the derivatives of the network outputs $\hat{u}$ w.r.t the network inputs x
- Techniques: i) Hand-coded analytical derivative; ii)finite difference; iii) symbolic differentiation; iv) automatic differentiation
- The derivatives are evaluated using backpropagation
  Rumelhart, Hinton, Williams. Nature 1986
- Take the derivatives of $\hat{u}$ with respect to its input x by applying the chain rule for differentiating compositions of functions using AD.

PINN for solving heat equation

Source: Lu, Meng, Mao, Karniadakis, SIAM Review 2021.



1. Construct a neural network $\hat{u}(x; \theta)$ with parameters θ .
2. Specify two training sets for the equation $\mathcal{T}_f$ and BC/IC $\mathcal{T}_b$.
3. Specify a loss function by summing the weighted L_2 norm of both the PDE and BC residuals.
4. Train the neural network to find the best parameters θ^* by minimizing the loss function $\mathcal{L}(\theta; \mathcal{T})$.

Minimizing the loss function

$$\mathcal{L}(\theta;\mathcal{T}) = \lambda_{\mathrm{f}}\mathcal{L}_{\mathrm{f}}(\theta;\mathcal{T}_{\mathrm{f}}) + \lambda_{\mathrm{b}}\mathcal{L}_{\mathrm{b}}(\theta;\mathcal{T}_{\mathrm{b}})$$

- Loss is highly nonlinear and nonconvex with respect to θ
- Minimize the loss function by gradient-based optimizers
  - Adam: Adaptive Moment Estimation
  - L-BFGS: Limited Broyden–Fletcher–Goldfarb–Shanno algorithm
  - The required number of iterations highly depends on the problem (the smoothness of the solution)
- Loss balancing scheme: ensure that each term in the loss function makes the same amount of progress over time.

Incompressible Navier-Stokes Equation

Governing Equation

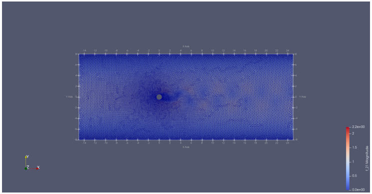
$$\rho(\frac{\partial u}{\partial t} + u \cdot \nabla u) = \nabla \cdot \sigma(u, p) + f$$
$$\nabla \cdot u = 0$$

- ρ, u, p, μ denote density, velocity, pressure and viscosity
- $\sigma(u, p)$ denotes the stress tensor: $\sigma(u, p) = 2\mu\epsilon(u) - pI$
- $\epsilon(u)$ denotes the strain-rate tensor: $\epsilon(u) = \frac{1}{2}(\nabla u + (\nabla u)^T)$
- Re denotes Reynolds Number: $Re = \frac{\rho u H}{\mu}$

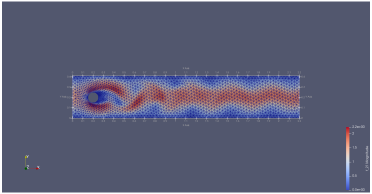
Test1/2: Flow past a cylinder

Geometry and parameters are taken from problem DFG 2D-2 benchmark test

- Free Stream with $u_\infty = 1, Re = 100$



- Free Stream with $u_\infty = 2, Re = 1000$



PINNs Setup

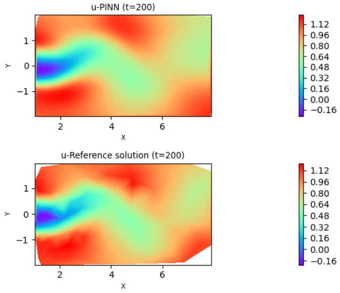
- Fully Connected Feedforward neural network:
  Layers $= 50 \times 6$
- Activation Function: "tanh"
- Optimization Algorithm: "adam"
- Learning Rate:
  - First half iterations: 10^{-3}
  - Second half iterations: 10^{-4}

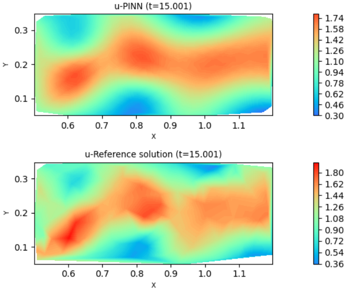
Case 1: $u_\infty = 1, Re = 100$

- Spacial Domain $[-2, 2] \times [1, 8]$
- Temporal Domain: $[200, 207]$
- Sample Size: 7000
- Iterations to Convergence: 20K

Case 2: $u_\infty = 2, Re = 1000$

- Spacial Domain $[0.05, 0.35] \times [0.5, 1.2]$
- Temporal Domain: $[15, 16]$
- Sample Size: 20000
- Iterations to Convergence: 600K





Compressible Euler Equations

Euler Equations

$$\mathrm{U_t + F(U)_x = 0}$$

$$\mathrm{U} = \begin{pmatrix} \rho \\ \rho \mathrm{u} \\ \mathrm{E} \end{pmatrix}, \mathrm{F(U)} = \begin{pmatrix} \rho \mathrm{u} \\ \rho \mathrm{u}^2 + \mathrm{p} \\ (\mathrm{E} + \mathrm{p})\mathrm{u} \end{pmatrix}$$

- where $\rho, \mathrm{u}, \mathrm{p}$ denote density, velocity, and pressure, γ is the constant specific heat ratio.
- Total Energy: $\mathrm{E} = \rho \mathrm{e} + \frac{1}{2}\rho \mathrm{u}^2$;
- Internal energy: $\mathrm{e} = \frac{\mathrm{p}}{(\gamma-1)\rho}$;
- Conservation Laws of mass, momentum, energy of compressible flow

Test1: Sod's Shock Tube

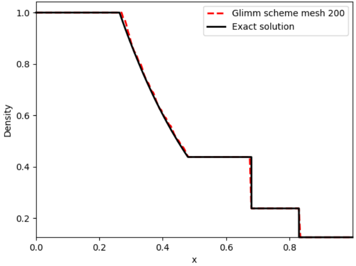
For the contact discontinuity tracking

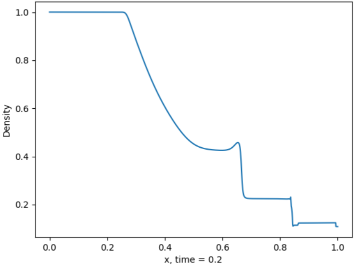
$$(\rho, \mathrm{u}, \mathrm{p}) = \begin{cases} (1, 0, 1) & \text{if } 0 \leq \mathrm{x} \leq 0.5 \\ (0.125, 0, 0.1) & \text{if } 0.5 < \mathrm{x} \leq 1 \end{cases}$$

- Fully Connected Feedforward neural network: Layers $= 20 \times 5$
- Activation Function: "tanh"
- Optimization Algorithm: "adam"
- Learning Rate:
  - First half iterations: 10^{-3}
  - Second half iterations: 10^{-4}

PINNs Setup

- Spacial Domain $[0, 1]$
- Temporal Domain: $[0, 0.2]$
- Sample Size: 20000
- Iterations to Convergence: 30M





PINNs Result

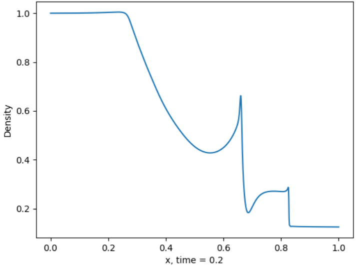


Figure: 300K iterations

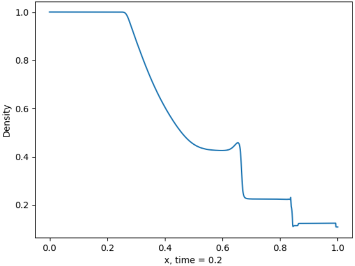


Figure: 30M iterations

Test2: Shock-Entropy Wave Interaction

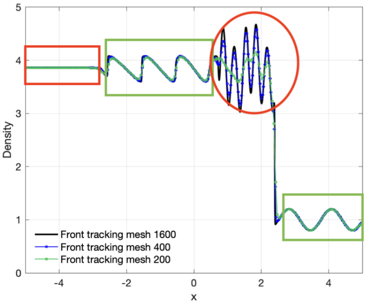
Study the stability and accuracy of the scheme for strong shocks.

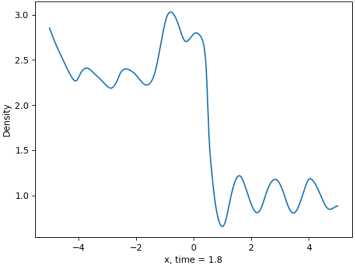
$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.33333) & \text{if } x \leq 0 \\ (1 + 0.2 \sin 5x, 0, 1) & \text{if } x \geq 0 \end{cases}$$

- Fully Connected Feedforward neural network
  - Train PINNs from random initialization 10 independent runs
  - Layers $= 10 + 20 \times 5$
- Activation Function: “tanh”
- Optimization Algorithm: “adam”
- Learning Rate:
  - First half iterations: 10^{-3}
  - Second half iterations: 10^{-4}

PINNs Setup

- Spacial Domain $[-5, 5]$
- Temporal Domain: $[0, 2.0]$
- Sample Size: 20K
- Iterations to Convergence: 450K





Observation:

- For accuracy, tune all the hyperparameters,
  - network size, learning rate, and the number of residual points.
  - Residual-based adaptive refinement by increasing residual points at "critical intervals".
- The required network size depends on the smoothness of the PDE solution.
  - Smooth PDE solution: a small network is sufficient
  - Stiff PDE solution: a deeper and wider network is required for the PDEs to achieve a similar level of accuracy

UNIVERSITY OF ARKANSAS

Thank you!






- Dr. James Glimm (SBU) and Dr. Snezhana Abarzhi (UWA) for many helpful comments and discussions
- Dr. David Youngs (AWE) and Dr. Jeffrey W. Jacob (UofA) for providing their RTI and RMI experimental data
- Dr. Oleg Schilling (LLNL) for sharing RMI simulation data
- Dr. Ann Almgren (LBNL) for supporting new AMReX-FronTier
- Lawrence Jesser Toll Jr. Endowed Chair at the Department of Mathematical Sciences in the Fulbright College of Arts & Sciences, 2017-present.
- University of Arkansas Chancellor's Innovation Fund, 2022-2025.
- Arkansas High Performance Computing Center (AHPCC)
- NSF #2204490, NSF #2346752