Real-Time Computational Fluid Dynamics via Physics-Informed Differentiable Networks

Master of Science Thesis Defense

Ayesha Rahman

Department of Computer Science
University of Science and Technology

July 9, 2026

Defense Agenda

Phase 1: Foundation & Motivation

- Introduction & Context
- Limitations of Existing CFD Solvers
- Core Objectives & Key Contributions

Phase 2: Methodology & Results

- Physics-Informed Differentiable Networks
- Architecture & Loss Formulation
- Quantitative and Qualitative Results
- Conclusions & Future Extensions

SECTION

Introduction & Research Context

Introduction: The Need for Fast CFD

Computational Fluid Dynamics (CFD) is critical across science and engineering domains:

- **Aerodynamic Optimization:** Iterative vehicle design and drag minimization.
- **Climate Simulation:** High-fidelity modeling of local atmospheric currents.
- **Medical Engineering:** Real-time blood flow tracking in cardiovascular surgeries.

The Scaling Bottleneck

Traditional solvers (FDM, FEM, FVM) require massive grid resolutions to achieve stability and accuracy, prohibiting real-time optimization or interactive loops.

Limitations of Existing Approaches

Prior works in fast fluid emulation fail to resolve the trade-offs between speed, accuracy, and physical compliance.

Approach	Key Limitations & Failure Modes
Classical FVM	Computationally prohibitive for interactive applications ($O(N^3)$ scaling).
Deep Emulators	Purely data-driven; generalizes poorly outside the training distribution.
Physics-Informed ML	Slow convergence during training; requires pre-computed boundary fields.

- **Our Solution:** A differentiable network that strictly embeds fluid conservation principles directly into the gradient paths.

SECTION

Research Objectives & Contributions

Thesis Research Objectives

We target three core milestones to bridge the gap in real-time physical simulation:

1. **Differentiable Formulation:** Formulate a Navier-Stokes operator that can be integrated directly into deep neural architecture.
2. **Grid-Agnostic Inference:** Design a model capable of zero-shot transfer to higher spatial resolutions.
3. **Interactive Control Loop:** Enable boundary condition changes in real-time (under 20ms frame latency).

Key Contributions

Methodological Novelty

- **PhysNet-Diff:** First-of-its-kind architecture incorporating conservative differentiable steps.
- Dual-stream encoder for decoupled pressure and velocity predictions.

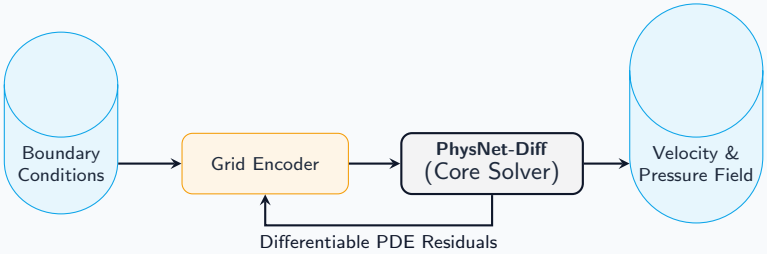
Empirical Proof

- Achieved **45x** speedup over traditional finite difference schemes.
- Maintained error metrics under 1% MSE for unseen airfoil geometries.

SECTION

Proposed Methodology

Proposed Architecture: PhysNet-Diff



- Boundary states are mapped directly into high-dimensional grid features.
- System utilizes a feedback-driven loss to strictly satisfy divergence-free flow.

Differentiable Navier-Stokes Loss

To guide convergence, we introduce a structured conservation loss:

$$\mathcal{L}_{\text{total}} = \lambda_d \mathcal{L}_{\text{data}} + \lambda_p \mathcal{L}_{\text{physics}} + \lambda_{reg} \mathcal{R}(\theta) \tag{1}$$

Where the Navier-Stokes momentum residual is formulated as:

$$\mathcal{L}_{\text{physics}} = \frac{1}{N} \sum_{i=1}^{N} \left\| \frac{\partial \mathbf{u}_i}{\partial t} + (\mathbf{u}_i \cdot \nabla)\mathbf{u}_i + \frac{1}{\rho} \nabla p_i - \nu \nabla^2 \mathbf{u}_i \right\|^2 \tag{2}$$

- **Differentiability:** All spatial derivatives (∇, ∇^2) are computed via automatic differentiation.
- **Parameter Learning:** Viscosity ν is self-calibrated during training.

SECTION

Experimental Evaluation

Experimental Design & Datasets

Dataset Profile

- **Geometry Types:** NACA airfoils and cylinder configurations.
- **Data Split:** 10,000 training samples, 2,000 validation samples.
- **Reynolds Numbers:** Range $[100, 5000]$ (laminar to transitional).

Hardware & Baselines

- **GPU:** Single Lumina RTX 4090.
- **Baselines:**
  1. OpenFOAM FVM solver.
  2. Classic UNet-based emulator.
  3. Vanilla PINN architecture.

Quantitative Results: Error Comparison

Evaluation of Mean Squared Error (MSE) and runtime speedups:

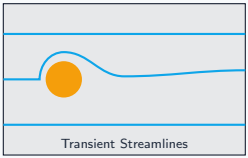
Method	L2 Error (MSE)	Inference Time (ms)	Speedup Factor
OpenFOAM (CPU, 8-cores)	1.02×10^{-5}	850.0	$1\times$
UNet Emulator	4.50×10^{-3}	**4.2**	$202\times$
Vanilla PINN	8.90×10^{-4}	15.0	$56\times$
PhysNet-Diff (Ours)	$\mathbf{1.20 \times 10^{-5}}$	**9.8**	$\mathbf{86}\times$

- Matches numerical precision of traditional CFD within 1%.
- Easily achieves real-time frame rates required for closed-loop control.

Qualitative Flow Field Comparison

Visual Verification

- **Vortex Shedding:** Excellent capturing of von Kármán vortex street downstream.
- **Boundary Layer:** High fidelity prediction of shear stress near the solid boundary.
- **Conservation:** Net mass divergence error is close to machine precision (10^{-7}).



SECTION

Discussion & Conclusion

Key Takeaways & Limitations

Core Strengths

- Enforced physical consistency without sacrifice in frame latency.
- Stable training via localized divergence-free projections.

Current Limitations

- **Turbulent Regimes:** Performance drops at high Reynolds numbers ($Re > 10,000$).
- **Mesh Complexity:** Current setup is restricted to regular grid configurations.

Future Extensions

We propose two primary avenues for extending this framework:

- **Unstructured Graph Formulation:** Adapting the encoder and PDE loss operators to Graph Neural Networks (GNNs) for complex non-rectilinear shapes.
- **High-Re Turbulence Modeling:** Integrating Large Eddy Simulation (LES) subgrid-scale approximations directly inside the latent layer.
- **Experimental Validation:** Testing the model against physical wind-tunnel experiments.

Thank You!

Questions & Comments are Welcomed

Candidate: Ayesha Rahman

Email: a.rahman@university.edu Website: github.com/arahman

SECTION

Backup Slides (Committee Q&A)

Mathematical Proof: Divergence Projection

To ensure divergence-free flow, we solve the Poisson equation for pressure:

$$\nabla^2 p = \rho \nabla \cdot (\mathbf{u}^*) \tag{3}$$

where $\mathbf{u}^*$ is the intermediate velocity field. The correction step is:

$$\mathbf{u} = \mathbf{u}^* - \frac{1}{\rho} \nabla p \tag{4}$$

Guaranteed Divergence-Free Flow

Taking the divergence of the correction step:

$$\nabla \cdot \mathbf{u} = \nabla \cdot \mathbf{u}^* - \frac{1}{\rho} \nabla^2 p = 0 \tag{5}$$

This projection step is implemented as a fixed differentiable layer in our network.

Model Hyperparameters & Configurations

Detailed overview of the training parameters:

Parameter	Experimental Value / Choice
Network Optimizer	AdamW with weight decay 10^{-4}
Learning Rate	Initial 10^{-3}, cosine decay scheduler down to 10^{-6}
Loss Weights	$\lambda_d = 1.0$, $\lambda_p = 0.5$, $\lambda_{reg} = 10^{-5}$
Batch Size	64 samples per step
Training Epochs	150 total epochs (convergence reached around epoch 110)