Oral presentation | Data science and AI

Data science and AI-I

Wed. Jul 17, 2024 4:30 PM - 6:30 PM Room C

[9-C-02] Fast and Efficient hp-Variational PINNs framework for solving the Incompressible Navier-Stokes equations

*Thivin Anandh[1], Divij Ghose[1], Sashikumaar Ganesan[1] （1. Indian Institute of Science, India）

Keywords: hp-VPINNs, Incompressible Navier-Stokes, Scientific ML, VPINNs for flow problems

Fast and Efficient hp-Variational PINNs framework for solving the Incompressible Navier-Stokes equations

T. Anandh*, D. Ghose* and S. Ganesan*
Corresponding author: sashi@iisc.ac.in * Department of Computational and Data Sciences,
Indian Institute of Science, Bangalore, India.

1 Introduction

Physics-informed Neural Networks (PINNs) solve partial differential equations (PDE) by incorporating the strong-form residual of the PDE into the neural network's loss function [1]. Variational physics-informed neural networks (VPINNs), which use the variational form of the PDE in the loss function, have shown promise in being more accurate than PINNs [2]. Moreover, concepts like h-refinement and p-refinement can be applied to VPINNs to further increase accuracy, resulting in the hp-VPINNs framework[3]. However, despite their benefits, hp-VPINNs face two significant challenges. First, training hp-VPINNs is computationally expensive, especially as the number of elements in the domain increases. Second, current frameworks are limited to uniform meshes and cannot handle geometries with irregular or skewed quadrilateral cells commonly found in CFD applications. In this work, we present a novel hp-VPINN framework called FastVPINNs [4] for solving 2D incompressible Navier-Stokes equations. Our framework efficiently computes the variational residual using tensor-based operations, resulting in speedups of up to 100x over existing implementation of the hp-VPINNs and facilitates computation on complex geometries using bilinear transformation. In this work, we propose to solve the 2D incompressible Navier-Stokes equation using hp-VPINNs, which are absent in the literature. We demonstrate this by solving the Kovasznay flow and lid-driven cavity flows at lower Reynolds numbers.

1.1 Incompressible Navier-Stokes Equation

Let $\Omega \subset \mathbb{R}^2$, be a bounded domain with boundary $\partial\Omega$. The simple version steady-state incompressible Navier-Stokes equation in 2D can be written as

$$\begin{aligned} -\nu\Delta\mathbf{u} + (\mathbf{u}.\nabla)\,\mathbf{u} + \nabla p &= \mathbf{0}, \quad \text{in } \Omega, \\ \nabla.\mathbf{u} &= 0, \quad \text{in } \Omega, \\ \mathbf{u} &= \mathbf{g}, \quad \text{in } \partial\Omega. \end{aligned} \tag{1}$$

The unknowns, velocity and pressure within the domain Ω are represented by $\mathbf{u}$ and p, respectively. The Dirichlet boundary value at $\partial\Omega$ is denoted by $\mathbf{g}$.

2 Methodology

In our framework, we improve the existing implementation of hp-VPINNs [3] by assembling the test function and its derivatives into a tensor, which enables us to compute the variational loss of all elements within the domain in a single BLAS operation, thus significantly reducing the training time of the network. Refer to [4] for additional details regarding the implementation. Figure 1 shows the speedup of our framework compared to the existing implementation of hp-VPINNs for a Poisson-2D problem. In this work, we have used a special form of Legendre Polynomial, where n^{th} order polynomial is represented as $v_n = P_{n+1} - P_{n-1}$ as mentioned in [2]

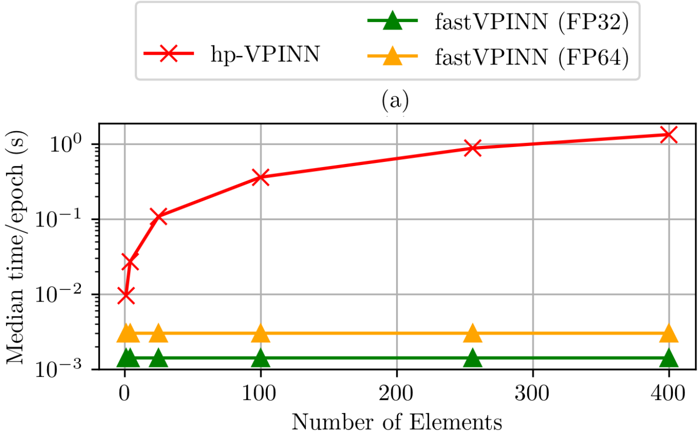

Figure 1: Training time taken per epoch for various frameworks for a Poisson 2D problem

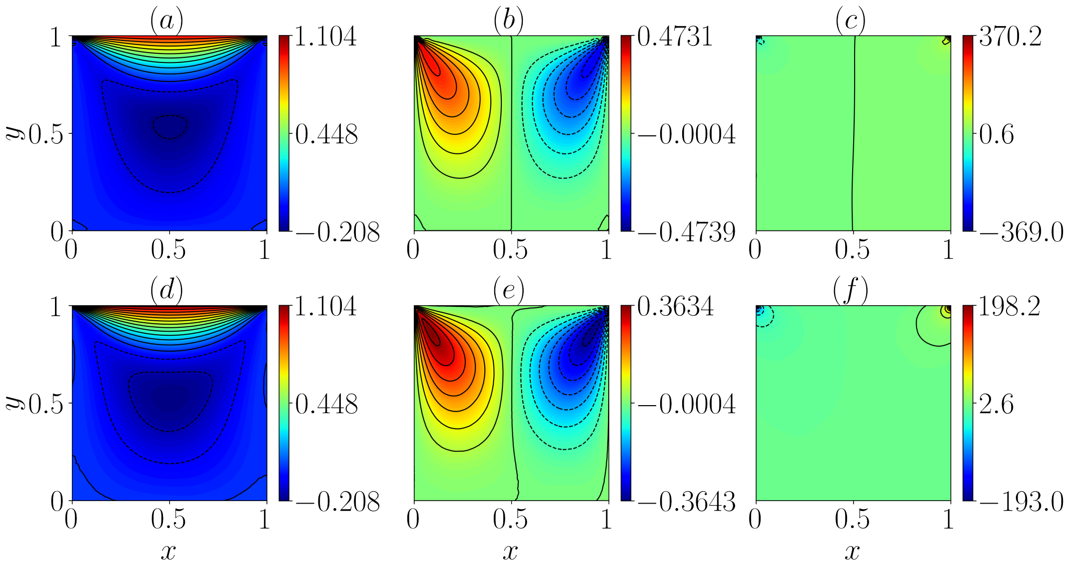

Figure 2: (a),(b),(c) - FEM Solution: velocity in x-direction, velocity in y-direction , pressure. (d),(e),(f) - Predicted Solution: velocity in x-direction, velocity in y-direction , pressure

3 Numerical Results

3.1 Lid Driven Cavity

We have used FastVPINNs to solve the lid-driven cavity problem. The domain is a unit square, $[0, 1]$ in both directions. A lid velocity of $u = 1$ is applied, and a Reynolds number of 1 is used. The mesh consists of 8 cells in each direction, with 64 test functions and 100 quadrature points per cell. A total of 400 Dirichlet boundary points are used. The neural network architecture consists of 5 layers with 30 neurons per layer. We compared our solution with the solution obtained using the Finite Element Method (FEM) [5] implemented in the FEM package ParMooN [6]. Our results show a good agreement with the FEM solution.

3.1.1 Kovasznay Flow

Kovasznay flow [7] in two dimensions for the incompressible Navier-Stokes equation has an analytical solution in the form:

$$u(x, y) = 1 - e^{\zeta x} \cos(2\pi y),\ v(x, y) = \frac{\zeta}{2\pi} e^{\zeta x} \sin(2\pi y),\ p(x, y) = \frac{1}{2} \left(1 - e^{2\zeta x}\right), \tag{2}$$

where

$$\zeta = \frac{1}{2\mu} - \sqrt{\frac{1}{4\mu^2} + 4\pi^2}, \quad \mu = \frac{1}{Re} = \frac{1}{40}.$$

The calculation was performed in a domain with $[-0.5, 1]$ in the x direction and $[-0.5, 1.5]$ in the y direction. We used 6 cells in the x-direction and 10 cells in the y-direction. we employed a neural network architecture with 4 hidden layers and 50 neurons per layer, 36 quadrature points per cell, resulting in 2160 quadrature points and 400 boundary points used 16 test functions per cell. We have solved for a Reynolds number of 40. For accuracy, we trained our model 10 times and reported the mean and standard deviation of the relative L_2 errors of u, v, and p.

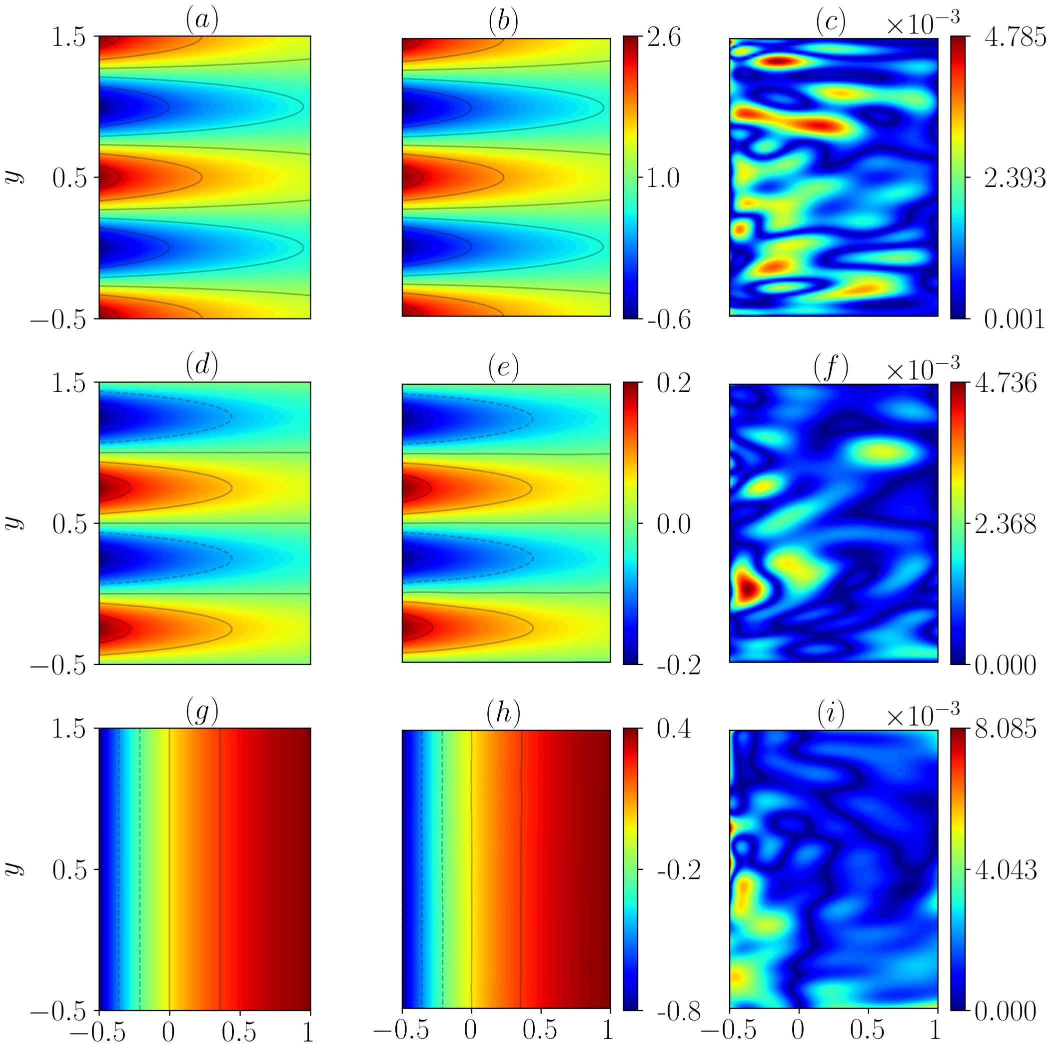

Figure 3: (a),(b),(c): actual, predicted and pointwise error of u respectively. (d),(e),(f): actual, predicted and pointwise error of v respectively. (g),(h),(i): actual, predicted and pointwise error of p respectively.

$\|\|u_{pred} - u_{exact}\|\|_2$	$\|\|v_{pred} - v_{exact}\|\|_2$	$\|\|p_{pred} - p_{exact}\|\|_2$
0.0049 ± 0.0036	0.0238 ± 0.0163	0.0063 ± 0.0017

Table 1: L_2 norms between predicted and exact solutions.

References

[1] Maziar Raissi, Paris Perdikaris, and George E Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *Journal of Computational physics*, 378:686–707, 2019.

[2] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. Variational physics-informed neural networks for solving partial differential equations. *arXiv preprint arXiv:1912.00873*, 2019.

[3] Ehsan Kharazmi, Zhongqiang Zhang, and George Em Karniadakis. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. *Computer Methods in Applied Mechanics and Engineering*, 374:113547, 2021.

[4] Thivin Anandh, Divij Ghose, Himanshu Jain, and Sashikumaar Ganesan. FastVPINNs: Tensor-Driven Acceleration of VPINNs for Complex Geometries, 2024.

[5] Sashikumaar Ganesan and Lutz Tobiska. *Finite elements: Theory and algorithms*. Cambridge University Press, 2017.

[6] Ulrich Wilbrandt, Clemens Bartsch, Naveed Ahmed, Najib Alia, Felix Anker, Laura Blank, Alfonso Caiazzo, Sashikumaar Ganesan, Swetlana Giere, Gunar Matthies, et al. ParMooN—A modernized program package based on mapped finite elements. *Computers & Mathematics with Applications*, 74(1):74–88, 2017.

[7] Leslie I George Kovasznay. Laminar flow behind a two-dimensional grid. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 44, pages 58–62. Cambridge University Press, 1948.