MAP55640 - Comparative Analysis of Hybrid-Parallel Finite Difference Methods on Multicore Systems and Physics-Informed Neural Networks on CUDA Systems

LI YIHAI, Mathematics Institute High Performance Computing, Ireland

MIKE PEARDON*, Mathematics Institute High Performance Computing, Ireland

This is the abstract of this project.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Additional Key Words and Phrases: Keywords

1 INTRODUCTION

Numerical methods of solving partial differential equations (PDE) have demonstrate far better performance than many other methods such as finite difference methods (FDM) [**<empty citation>**], finite element methods (FEM) [**<empty citation>**], Lattice Boltzmann Method (LBM) [**<empty citation>**] and Monte Carlo Method (MC) [**<empty citation>**]. In recent years, researchers in the field of deep learning have mainly focused on how to develop more powerful system architectures and learning methods such as convolution neural networks (CNNs) [**<empty citation>**], Transformers [**<empty citation>**] and Perceivers [**<empty citation>**] . In addition, more researchers have tried to develop more powerful models specifically for numerical simulations. Despite of the relentless progress, modeling and predicting the evolution of nonlinear multiscale systems which has inhomogeneous cascades-scales by using classical analytical or computational tools inevitably encounts severe challanges and comes with prohibitive cost and multiple sources of uncertainty.

This project focuses on the promotions on performance gained from the parallel compute systems, in general, the FDMs and Neural Networks (NNs) are evaluated. Moreover, it prompted series Message Passing and Shared

*Supervisor of this Final Project

Authors' addresses: Li Yihai, liy35@tcd.ie, Mathematics Institute and High Performance Computing, Dublin, Ireland; Mike Peardon, mjp@maths.tcd.ie, Mathematics Institute and High Performance Computing, Dublin, Ireland.

2024. Manuscript submitted to ACM

Memory hybrid parallel strategies using Message Passing Interface (MPI) [**MPI**] and Open Multi-processing (OpenMP) [**OpenMP**].

2 RELATED WORK

To gain well quality solution of various types of PDEs is prohibitive and notoriously challanging. The number of methods avaliable to determine canonical PDEs is limited as well, includes separation of variables, superposition, product solution methods, Fourier transforms, Laplace transforms and perturbation methods, among a few others. Even though there methods are exclusively well-performed on constrained conditions, such as regular shaped geometry domain, constant coefficients, well-symmetric conditions and many others. These limits strongly constrained the range of applicability of numerical techniques for solving PDEs, rendering them nearly irrelevant for solving problems pratically.

General, the methods of determining numerical solutions of PDEs can be broadly classified into two types: deterministic and stochastic. The mostly widely used stochastic method for solving PDEs is Monte Carlo Method [**Monte Carlo Method**] which is a popular method in solving PDEs in higher dimension space with notable complexity.

2.1 Finite Difference Method

The Finithe Difference Method(FDM) is based on the numerical approximation method in calculus of finite differences. The motivation is quiet straightforward which is approximating solutions by finding values satisfied PDEs on a set of prestibed interconnected points within the domain of it. Those points are which referd as nodes, and the set of nodes are so called as a grid of mesh. A notable way to approximate derivatives are using Taylor Series expansions. Taking 2 dimension Possion Equation as instance, assuming the investigated value as, φ,

$$\frac{\partial^2 \varphi}{\partial x^2} + \frac{\partial^2 \varphi}{\partial y^2} = f(x, y) \tag{1}$$

The total amount of nodes is denoted with $N = 15$, which gives the numerical equation which governing equation 1 shown in equation 2 and nodes layout as shown in the figure 1

$$\frac{\partial^2 \varphi_i}{\partial x_i^2} + \frac{\partial^2 \varphi_i}{\partial y_i^2} = f(x_i, y_i) = f_i, \quad i = 1, 2, \ldots, 15 \tag{2}$$

In this case, we only need to find the value of internal nodes which i is ranging from 1 to 10. Next is aimming to solve

Fig. 1. The Schematic Representa of of a 2D Computiatioal Domain and Grid. The nodes are used for the FDM by solid circles. Nodes $11 - 15$ denote boundary nodes, while nodes $1 - 10$ denote internal nodes.

this linder system 2.

2.2 Physics Informed Neural Networks

With the explosive growth of avaliable data and computing resouces, recent advances in machine learning and data analytics have yieled good results across science discipline, including Convolutional Neural Networks (CNNs) [**CNN**] for image recoginition, Generative Pre-trained Transformer (GPT) [**GPT**] for natual language processing and Physics Informed Neural Networks (PINNs) [**PINN**] for handling science problems with high complexity. PINNs is a type of machine learning model makes full use of the benefits from Auto-differentiation (AD) [**AD**] which led to the emergence of a subject called Matrix Calculus [**Matrix_Calculus**]. Considering the parametrized and nonlinear PDEs of the general form [E.q. 3] of function $u(t, x)$

$$u_t + \mathcal{N}\left[u; \lambda\right] = 0 \tag{3}$$

The $\mathcal{N}[\cdot;\lambda]$ is a nonlinear operator which parametrized by λ. This setup includes common PDEs problems like heat equation, and black-stokz equation and others. In this case, we setup a neural network $NN[t, x; \theta]$ which has trainable weights θ and takes t and x as inputs, outputs with the predicting value $\hat{u}(t, x)$. In the training process, the next step is calculating the necessary derivatives of u with the respect to t and x. The value of loss function is a combination of the metrics of how well does these predictions fit the given conditions and fit the natural law [Fig. 2].

Fig. 2. The Schematic Representa of a structure of PINN.

3 PROBLEM SETUPS

In this project, I chose to use various numerical approaches to approximate the solutions. It begins with Finite Difference Methods(FDM) and an other method employed by deep neural networks which leverage by their capability as universal function approximators [2].

The Kormoglov PDEs are series of equations which describe the motions of Brownian Motions [1]. In general, let $T \in (0, +\infty)$, $d \in \mathbb{N}$, let $\mu : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma : \mathbb{R}^d \to \mathbb{R}^{d\times d}$ be the Lipschitz continuous fucntions. Let $\varphi : \mathbb{R}^d \to \mathbb{R}$ be a function, and u be a function from Hilbert Space $[0, T] \times \mathbb{R}^d$ to $\mathbb{R}$

$$u : [0, T] \times \mathbb{R}^d \longrightarrow \mathbb{R}$$
$$(t, x) \longmapsto u$$

with at most polynomially growing partial derivatives. The problem is that u satisfied a below system on $D = [0, T] \times [a, b]^d$ and $a, b \in \mathbb{R}^d$ with $a < b$,

$$\frac{\partial u}{\partial t} = \frac{1}{2}\mathrm{Trace}_{\mathbb{R}^d}\left[\sigma(x)\sigma^*(x)\left(\mathrm{Hess}_x u\right)\right] + \langle \mu, \nabla_x u\rangle_{\mathbb{R}^d} \tag{4}$$

$$u(0, x) = \varphi(x), \quad x \in [a, b]^d \tag{5}$$

$$\left(\frac{\partial u}{\partial \vec{n}} + \lambda u\right)\bigg|_{x\in\Gamma} = g(t, x) \quad \forall t \in [0, T],\ x \in \mathbb{R}^d \tag{6}$$

The goal is to numerically appriximate the stablized state $u(T, x)$ of the system in the fureture time T, and in the hypercude space $[a, b] \in \mathbb{R}^d$ with various boundary conditions g.

Reprompt problem, this work aimed at solving this problem in a bigger picture which requires a more general form. In this project, I consider the parametrized and nonliner PDEs of the general form

$$\frac{\partial u}{\partial t} + \mathcal{N}\left[u; \lambda\right] = 0 \quad t \in [0.T], x \in D \tag{7}$$

where $\mathcal{N}[\cdot; \lambda]$ stands for a nonlinear operator parametrized by λ.

3.1 Navior-Stokz Equation

4 METHEDOLOGY

For solving the PDEs, we need to begin with descritizing the objects or the region we are going to evaluate via matrices. Without consdering more detail, the navie way for these transforming processes is to simply using the coordinates in d dimension spaces and the values of function at the points to simplify the objects. One of the famous numerical methods based on such methedology is called Finite Difference Method (FDM [**FDM**]), which is a fine way to investigate objects with regular shapes such as Cube.

Fig. 3. Figure Shows the FDM idea, If NEED

The other type of methods are aiming to solve systems on irregular shapes, such as simulating the aerodynamics effects of a jet. The methods, Finate Elements Methods (FEM [**FEM**]), are start with deviding the objects into numbers of elements, typically quadrilateral in 2D spaces and tetrahedron in 3D spaces.

Fig. 4. Figure Shows the FEM idea, a jet, If NEED

4.1 Finite Difference Methods

4.1.1 Pure Message Passing Parallel.

4.1.2 Hybrid Parallel.

4.2 Physics Informed Neural Networks

4.2.1 CUDA parallel.

4.2.2 Hybrid Parallel.

5 IMPLEMENTATION

5.1 Finite Difference Methods

5.1.1 Pure Message Passing Parallel.

5.1.2 Hybrid Parallel.

5.2 Physics Informed Neural Networks

5.2.1 CUDA parallel.

5.2.2 Hybrid Parallel.

5.3

6 EXPERIMENTS

7 CONCLUSION

8 ACKNOWLEDGEMENT

REFERENCES

[1] Christian Beck, Sebastian Becker, Philipp Grohs, Nor Jaafari, and Arnulf Jentzen. 2021. Solving the kolmogorov pde by means of deep learning. *Journal of Scientific Computing*, 88, 3, 73. DOI: 10.1007/s10915-021-01590-0.

[2] Kurt Hornik, Maxwell Stinchcombe, and Halbert White. 1989. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2, 5, 359–366. DOI: https://doi.org/10.1016/0893-6080(89)90020-8.