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

Li Yihai
liy35@tcd.ie
Mathematics Institute
High Performance Computing
Dublin, Ireland

Mike Peardon[*]
mjp@maths.tcd.ie
Mathematics Institute
High Performance Computing
Dublin, Ireland

ABSTRACT

This is the abstract of this project

KEYWORDS

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 the benefits of the parallalization on appriximating nonliner systems. In general, the FDM and FEM methods are evaluated on pure message passing interface (MPI) [**MPI**] parallelizations. Moreover, it also prompted Hybrid parallel strategies on FDM which combined MPI and share memory parallalization [**SMP**] (SMP) to make more use of the speed promotions come from faster caches of each compute unit. Besides the Hybrid strategies on FDM, the FEM is built on the external library Deal.II [**DEAL.II**], which also built on MPI parallalizations. (BUT CURRENTLY NOT SURE for Hybrid).

[*]Supervisor of this Final Project

MAP55640, May 25, 2024, Dublin, Ireland
2024.

2 RELATED WORK

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 $\mathbb{R}$

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

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 $a < b$,

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

$$u(0, x) = \varphi(x), \quad x \in [a, b]^d \tag{2}$$

$$\left.\left(\frac{\partial u}{\partial \vec{n}} + \lambda u\right)\right|_{x\in\Gamma} = g(t, x) \quad \forall t \in [0, T],\ x \in \mathbb{R}^d \tag{3}$$

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

$$\frac{\partial u}{\partial t} + \mathcal{N}[u; \lambda] = 0 \quad t \in [0.T], x \in D \tag{4}$$

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.

Figure 1: 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.

Figure 2: 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.