

Meta-Programming and Hybrid Parallel Strategies for Solving PDEs: An FDM and PINN Comparison ^{1 2}

Seminar Presentation III

LI YIHAI

Student ID: 23345919

Supervision: Michael Peardon

September 17, 2024



Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath

¹full docs: <https://liyihai.com/html/index.html>

²repository: <https://github.com/liyihai-official/Final-Project>

1 Introduction

- Related Work
- Challenges & Objectives

2 Problem Setups

- General Form

- Thermal Conduction Systems

3 Methodology

- N-Dimension Matrix

4 Experiments

5 Discussion

1 Introduction

- Related Work
- Challenges & Objectives

2 Problem Setups

- General Form

- Thermal Conduction Systems

3 Methodology

- N-Dimension Matrix

4 Experiments

5 Discussion

Introduction

Recap - Deep Neural Network

Kolmogorov PDEs

Solving $u(x, T)$, for $\mathbb{R}^1 \ni T > 0, x \in \mathbb{R}^d, t \in [0, T], u(t, x) = u \in \mathbb{R}^1, \mu(x) \in \mathbb{R}^d, \sigma(x) \in \mathbb{R}^{d \times d}$,

$$u_t = \frac{1}{2} \text{Trace}_{\mathbb{R}^d} [\sigma(x) [\sigma(x)]^* \text{Hess}_x u] + \langle \mu(x), \nabla_x u \rangle_{\mathbb{R}^d} \quad (1)$$

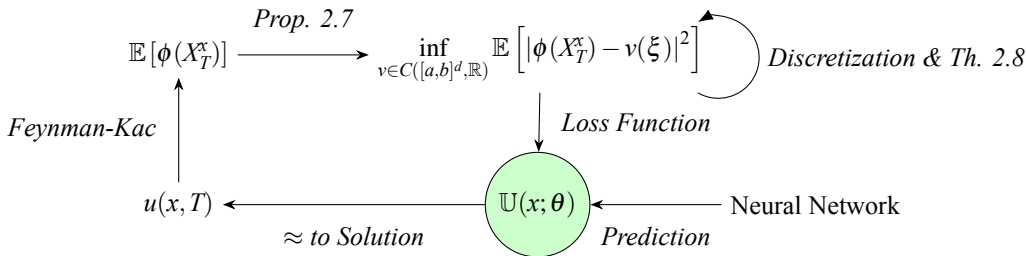


Figure: Deep Neural Network (DNN) Methodology of Solving Kolmogorov PDEs **[FIRST]**

Introduction

Recap - Physics Informed Neural Network

General Form of PDEs

- $u(t, x)$ denotes with the target function, $x \in \mathbb{R}^d$.
- $\Gamma[\cdot; \lambda]$ is a non-linear operator parameterized by λ .

$$u_t(t, x) + \Gamma[u; \lambda] = 0 \quad (2)$$

Define $f(t, x)$ to be given by $f(t, x) = u_t(t, x) + \Gamma[u; \lambda]$ (3)

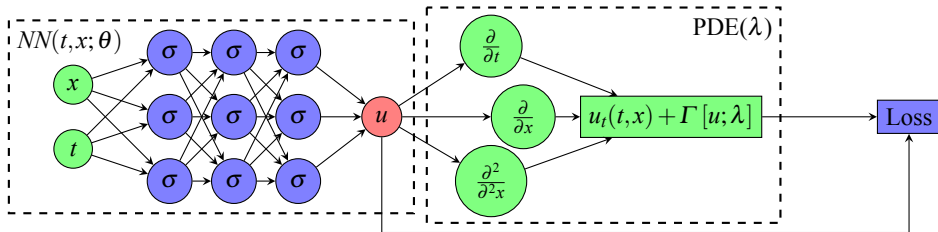


Figure: PINN, with 3 fully connected hidden layers

Comparing With Finite Difference Time Domain Method (FDTD)

① Deep Neural Network [**FIRST**]

- Gives lower quality approximations.
- Takes longer time to train.
- Possible to solve high dimension PDEs

② Physics Informed Neural Network

- Gives higher quality approximations.
- Takes longer time to train.
- Has more flexible way to get results.
- Possible to solve high dimension PDEs

Objectives

This project focused on following objectives:

- ① Implement FDTD and PINN in C++/C.
- ② Implement FDTD hybrid parallel version using MPI/OpenMP.
- ③ Implement PINN GPU parallel version using Libtorch/CUDA.
- ④ Evaluate the efficiency and accuracy of FDTDs and PINNs.

However, there were many obstacles including:

- ① Portability.
- ② Overlapping Communication and Computation.
- ③ Unnecessary intra-node communication
- ④ Communication Overhead.
- ⑤ Scalability Issues.
- ⑥ Memory Management.

1 Introduction

- Related Work
- Challenges & Objectives

2 Problem Setups

- General Form

- Thermal Conduction Systems

3 Methodology

- N-Dimension Matrix

4 Experiments

5 Discussion

General Form of problem

The PDE parametrized by number λ and an operator $\mathcal{N}[\cdot; \lambda]$, and assume the variable x is a 2D or 3D spatio-vector which is written in

$$\begin{cases} \frac{\partial u}{\partial t}(t, \vec{x}) + \mathcal{N}[u; \lambda] = 0 \\ u(0, \vec{x}) = \varphi(\vec{x}) \end{cases} \quad (4)$$

where φ is the initial condition, and $\vec{x} \in \Omega, t \in [0, +\infty)$.

Boundary Conditions

The Dirichlet and Von Neumann boundary conditions are formed as

$$\begin{cases} u(t, \vec{x}) = g(t, \vec{x}) \\ \frac{\partial u}{\partial \vec{n}} = g(t, \vec{x}) \end{cases} \quad (5)$$

where \vec{n} is the normal vector on $\overline{\Omega}$ the boundary of domain Ω .

Thermal Conduction Systems

Heat Equation 2D

The function

$$u(t, x, y) = x + y - xy, \forall \alpha \in \mathbb{R}^1 \quad (6)$$

is the solution of 2D Heat Equation 7 below

$$\begin{aligned} \frac{\partial u}{\partial t} &= \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) & (x, y) \in \Omega, t \in [0, +\infty) \\ u(0, x, y) &= \varphi(x, y) = 0 & (x, y) \in \Omega \\ u(t, x, y) &= g(x, y) = \begin{cases} y, & x = 0, y \in (0, 1) \\ 1, & x = 1, y \in (0, 1) \\ x, & y = 0, x \in (0, 1) \\ 1, & y = 1, x \in (0, 1) \end{cases} & t \in [0, +\infty) \end{aligned} \quad (7)$$

Thermal Conduction Systems

Heat Equation 3D

The function

$$u(t, x, y, z) = x + y + z - 2xy - 2xz - 2yz + 4xyz, \forall \alpha \in \mathbb{R}^1 \quad (8)$$

is the solution of 3D Heat Equation 9 below

$$\begin{aligned} \frac{\partial u}{\partial t} &= \alpha \left(\frac{\partial^2 u}{\partial^2 x} + \frac{\partial^2 u}{\partial^2 y} + \frac{\partial^2 u}{\partial^2 z} \right) & (x, y, z) \in \Omega, t \in [0, +\infty) \\ u(0, x, y, z) &= \varphi(x, y, z) = 0 & (x, y, z) \in \Omega \quad (9) \\ u(t, x, y, z) &= g(x, y, z) = \begin{cases} y + z - 2yz, & x = 0, \\ 1 - y - z + 2yz, & x = 1, \\ x + z - 2xz, & y = 0, \\ 1 - x - z + 2xz, & y = 1, \\ x + y - 2xy, & z = 0, \\ 1 - x - y + 2xy, & z = 1 \end{cases} & t \in [0, +\infty) \end{aligned}$$

1 Introduction

- Related Work
- Challenges & Objectives

2 Problem Setups

- General Form

- Thermal Conduction Systems

3 Methodology

- N-Dimension Matrix

4 Experiments

5 Discussion

N-dimension Matrix

STL containers

STL provides containers `std::array` and `std::vector` for creating one-dimension array. There are two way for dealing with high-dimension data:

- 1 Nesting the one dimension arrays or vectors.
- 2 Hierarchy approach, designing derived classes of one-dimension array base class.

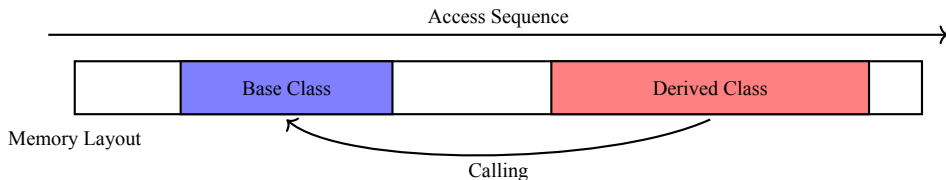


Figure: Derived Class calling members in Base class, timing is not predictable.

- ① Nesting multi-dimension array has non-contiguous memory layout.
- ② Derived class needs more time to access members in base class.
- ③ Poor cache utilization leads to poor performance.
- ④ MPI type create requires contiguous memory layout.

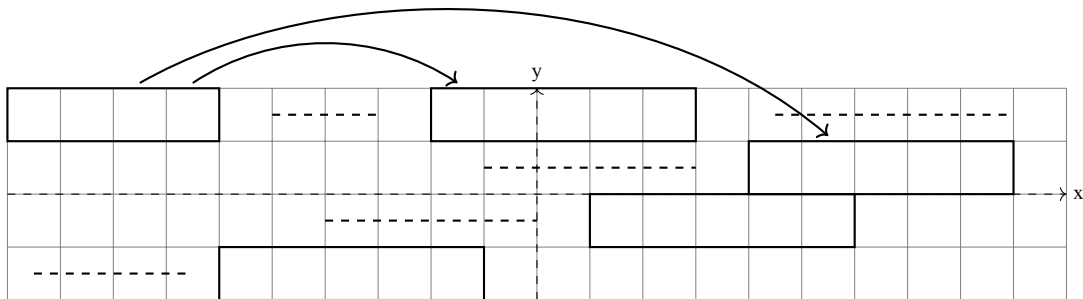


Figure: <caption>

N-dimension Matrix

Solution

- An external small `__multi_array_shape` object defines the routines for accessing the elements.
- Smart pointer, ensure memory's contiguous layout and safety.
- Separate into two detail and user interface objects adhering RAII rules.

1 Introduction

- Related Work
- Challenges & Objectives

2 Problem Setups

- General Form

- Thermal Conduction Systems

3 Methodology

- N-Dimension Matrix

4 Experiments

5 Discussion

1 Introduction

- Related Work
- Challenges & Objectives

2 Problem Setups

- General Form

- Thermal Conduction Systems

3 Methodology

- N-Dimension Matrix

4 Experiments

5 Discussion

Thank you for your attention!

Any questions?