Parallel Computing for Partial Differential Equations

Student: Wu Tong Supervisor: Dr. Kwai Wong

[1]The Chinese University of Hong Kong [2]The University of Tennessee, Knoxville

Abstract

Solving Partial Differential Equations(**PDE**) accurately and efficiently is of crucial importance in Computational Mathematics and many other fields. Our group implement **Parallel Computing** in solving common PDEs by Finite Element Method(FEM) using PETSc, FEniCSx and MFEM library. Our group study the structure of parallel matrix assembly process, and solving techniques such as FEM and iteration methods. Various Krylov Subspace Methods(KSP) is applied to solve linear systems.

Introduction

- The main task of our group is solving Partial Differential Equations using parallel computing techniques. Our group apply Finite Element Methods to PDEs by following the **reference** → **local** → **global** framework. After transforming the PDEs to ODEs, and ODEs to linear systems, KSP methods such as Conjugate Gradient Method(CG), Preconditioned Conjugate Gradient Method(PCG) and Generalized Minimal Residual Method (GMRES) are applied to solve large linear systems. Parallel matrix and vector assembly process and solving skills are studied.

- For time-dependent problem, both Euler methods and Crank-Nicolson Method is applied, and errors with respect to different time step are computed and analyzed.

- For nonlinear PDEs such as 2D/3D Burger's Equation, Newton's method and Picard iteration are applied.

- For more complicated problems such as Incompressible Navier Stokes Equations, different solving techniques such as projection method and velocity-vorticity formulation are also studied. Further implementation are still need.

Methodology: Finite Element Method for Linear PDE

Step 1: Construct basis function ϕ and finite element space.

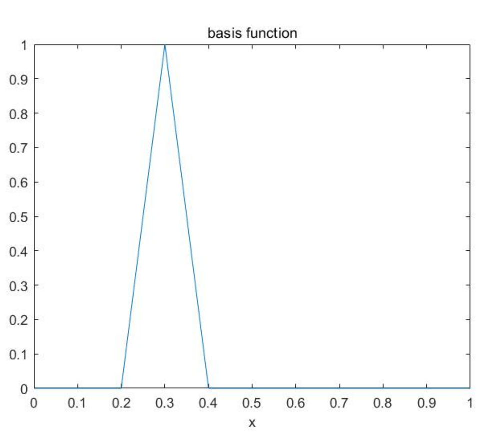

Figure 1. basis function in 1D

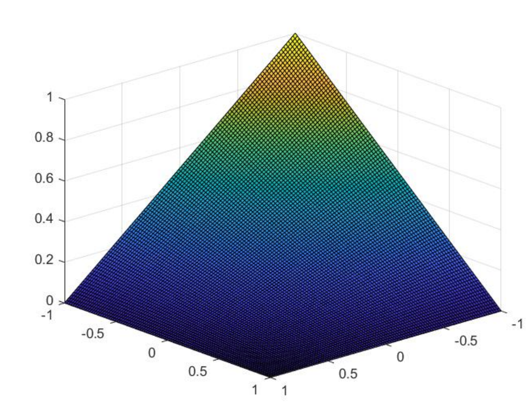

Figure 2. 2D reference basis function

Step 2: Construct Weak Formulation and semi-discretization of PDEs.

$$\begin{cases} -\nabla\cdot(b(x,y)\nabla u) = f & in \quad \Omega \\ \nabla u\cdot n + pu = q(x,y) & on \quad \partial\Omega \end{cases} \tag{1}$$

Let bilinear form a and functional l be defined as

$$a(u,v) = \int_{\partial\Omega} bpuv + \int_{\Omega} b\nabla u\cdot\nabla v, \quad l(v) = \int_{\Omega} fv - \int_{\partial\Omega} bqv$$

Let the approximate solution $\hat{u}$ be expressed by

$$\hat{u}(x) = \sum_{i=1}^{N} u_i\phi_i(x)$$

We shall get from Equation (1)

$$\sum_{i=1}^{N} a(\phi_i,\phi_j)u_i = f(\phi_j)$$

Put it into linear systems, let A be a $N\times N$ matrix such that

$$A_{ij} = a(\phi_i,\phi_j)$$

Let $F\in\mathbf{R}^N$ such that

$$F_i = f(\phi_i)$$

for each i, we get:

$$AU = F$$

Step 3: Solve the linear system by using iterative solvers. Recover $\hat{u}$ from nodal values vector U, and calculate the error norm.

Methodology: Parallel Computing

PETSc, the Portable, Extensible Toolkit for Scientific Computation, is for the scalable (parallel) solution of scientific applications modeled by partial differential equations.

- Assembly matrix and vectors in parallel

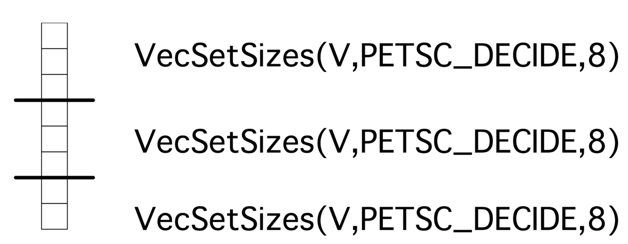

Figure 3. Set Vector in parallel

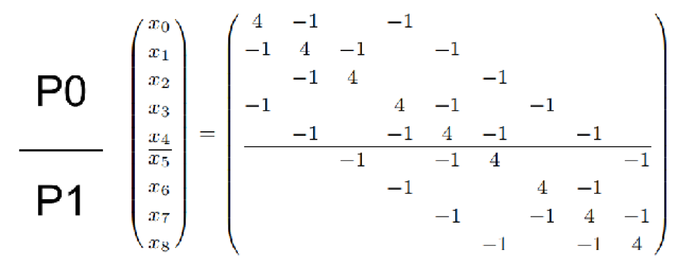

Figure 4. Set Matrix in parallel

- Storage Scheme: Coordinate Storage and Compressible Sparse Row Storage

$$A = \begin{pmatrix} 10 & 20 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 40 & 0 & 0 \\ 0 & 0 & 50 & 60 & 70 & 0 \\ 0 & 0 & 0 & 0 & 0 & 80 \end{pmatrix}$$

$$\begin{pmatrix} 10 & 20 & 30 & 40 & 50 & 60 & 70 & 80 \\ 0 & 0 & 1 & 1 & 2 & 2 & 2 & 3 \\ 0 & 1 & 1 & 3 & 2 & 3 & 4 & 5 \end{pmatrix}$$

Figure 5. Coordinate Storage

```
V         = [ 10 20 30 40 50 60 70 80 ]
COL_INDEX = [  0  1  1  3  2  3  4  5 ]
ROW_INDEX = [  0  2  4  7  8 ]
```

Figure 6. Compressible Sparse Row Storage

- Implementing Parallel Iterative Solvers

Solving a linear system $Ax = b$ with Gaussian elimination can take lots of time/memory.

Alternative: iterative solvers such as KSP solvers use successive approximations of the solution.

- Convergence not always guaranteed.
- Possibly much faster / less memory.
- Also need a preconditioner.

Implementation Using MFEM and PETSc

Using Conjugate Gradient Method to solve

$$\begin{cases} -\Delta u + 2u = f & in \quad \Omega \\ \dfrac{\partial u}{\partial n} + \alpha u = g & on \quad \partial\Omega \end{cases} \tag{2}$$

Desired solution: $u(x,y) = \cos(x+y)$. Number of finite element unknowns: 132225

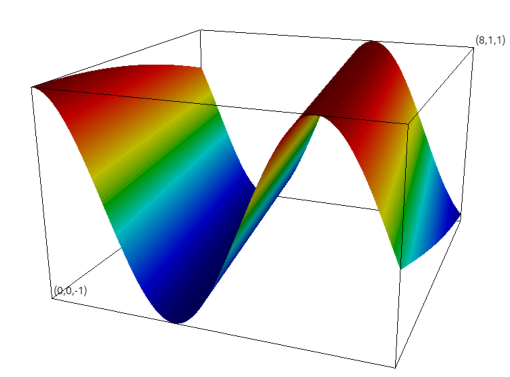

Figure 7. L^2 norm of error: 8.18589×10^{-6}

Using Conjugate Gradient Method to solve

$$\begin{cases} -\Delta u + 2u + \lambda\cdot\nabla u = f & in \quad \Omega \\ u = g & on \quad \partial\Omega \end{cases} \tag{3}$$

Desired solution: $u(x,y) = \exp(x+0.2y)$. Number of finite element unknowns: 132225

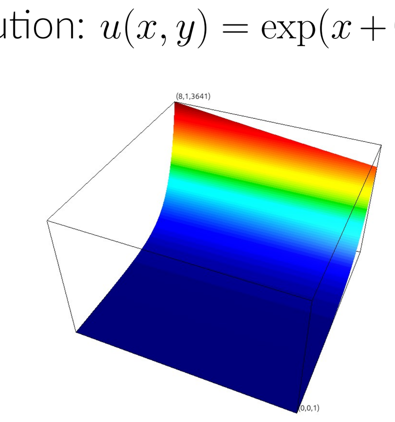

Figure 8. CG does not converge
L_2 error norm: 1.33×10^{-2}

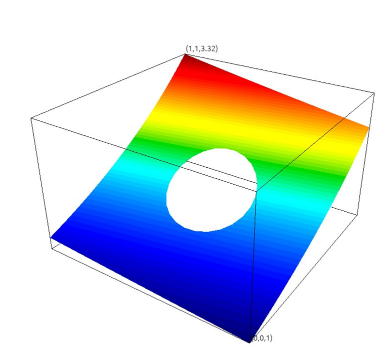

Figure 9. CG converges
L_2 error norm: 7.7752×10^{-6}

Implementation Using MFEM and PETSc

Using Conjugate Gradient Method to solve the 2D nonlinear PDE

$$\begin{cases} \dfrac{\partial u}{\partial t} - \Delta u + u^2 = f & in \quad \Omega\times[0,1] \\ u = g & on \quad \partial\Omega\times[0,1] \\ u = u_0 & at \quad t = 0 \end{cases} \tag{4}$$

Desired solution: $u(x,y) = x^2t + 10y^3$. Time step $\Delta t = 0.01$. Number of finite element unknowns: 2601 at each time step. Apply Picard Method and Newton's Iteration at each time step.

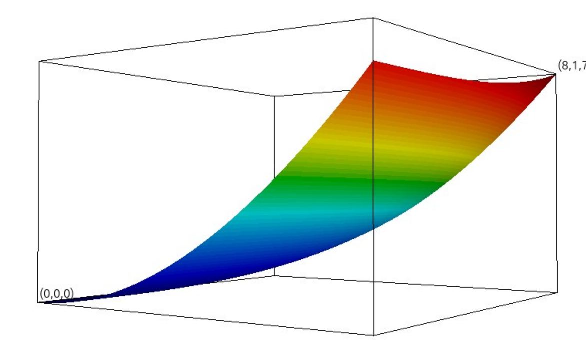

Figure 10. solution at $t = 1$, L^2 norm of error: 1.14×10^{-2}

Error Comparison

For Equation (3) with the desired solution $u(x,y) = \cos(x+y)$ and rectangular domain, mesh refinements are applied.

Number of Refinement	Number of Unknowns	Error norm
0	132225	1.72495×10^{-5}
1	526593	4.31147×10^{-6}
2	2101761	1.077872×10^{-6}
3	8397825	2.69495×10^{-7}

Table 1. L_2 Error norm Comparison

For Equation (4), different time step size are applied.

Δt	Number of Unknowns	Error norm
0.02	2601	1.85×10^{-2}
0.01	2601	1.14×10^{-2}
0.005	2601	7.84×10^{-3}
0.0025	2601	6.09×10^{-3}

Table 2. L_2 Error norm Comparison

Incompressible Navier-Stokes Equations

The following two equations are the Momentum equation and Continuity Equation.

$$\frac{\partial \mathsf{u}}{\partial t} + \sum_{j=1}^{N} u_j\frac{\partial \mathsf{u}}{\partial x_j} - \nu\Delta\mathsf{u} + \nabla p = \mathsf{f} \tag{5}$$

$$\nabla\cdot\mathsf{u} = 0 \tag{6}$$

- Projection Method: One nonlinear evaluation, one poisson solve with Neumann Boundary Condition, and one Helmoltz solve with Dirichlet Boundary Condition for each time step.

- Velocity-Vorticity Formulation.

Future work

Benchmark problems from INS, such as Lid-driven Cavity need to be solved, and error analysis also needs to be studied and examined.

Acknoledgements

I would like to express my gratitude to Dr. Kwai Wong for his guidance and support throughout this research. Additionally, I extend my sincere thanks to William Lau and Tony Cheung for their assistance and collaboration during the project. Their expertise and experience have been instrumental in my progress.