Parallel Adaptive Discontinuous Galerkin Method for Chemical Transport Equations

Ng, Yin-ki, Chinese University of Hong Kong

Mentors: Dr. Ohannes Karakashian, Dr. Kwai Wong

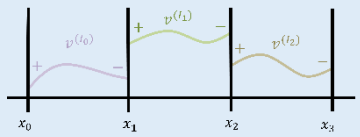
Overview

Discontinuous Galerkin Method (DG-FEM) is a class of Finite Element Method (FEM) for finding approximation solutions to systems of differential equations that can be used to simulate chemical transport phenomena.

The goal of my project is to implement parallelization on DG-FEM codes that can be scaled on existing supercomputers.

DG-FEM

In standard Finite Element Method, test functions are chosen for the weak formulation of the differential equation. DG-FEM chooses test functions that are discontinuous across adjacent elements, resulting jump conditions on the shared boundaries.



v+, v- : test function values on element boundaries

Adaptive DG-FEM over FEM

Generally using a very fine grid for the whole computational domain would be ineffective. Discontinuity between element boundaries provides local support and thus promotes :

✓ Local refinement ✓ Complex geometries
✓ Parallelization ✓ Higher-order accuracy

Constructing 1-Dimension DG-FEM parallel code

1D Poisson's Equation on domain $I = [a, b]$

$$\begin{cases} -u'' = f \\ u(a) = u(b) = 0 \end{cases}$$

Weak formulation using test function v

$$-\int u''v = -\sum_{j=0}^{j=N}\int_{x_j}^{x_{j+1}} u''\phi$$

$$= -\sum_{j=0}^{j=N}\left(\int_{x_j}^{x_{j+1}} u'v' + u'(x_j^+)\,v(x_j^+) - u'(x_{j+1}^-)\,v(x_{j+1}^-)\right)$$

$$= -\sum_{j=0}^{j=N}\int_{x_j}^{x_{j+1}} u'v' + u'(x_0^+)\,v(x_0^+) + \left(\left(-u'(x_1^-)\,v(x_1^-) + u'(x_1^+)\,v(x_1^+)\right) + \dots \right.$$

$$\left. -\left(-u'(x_N^-)\,v(x_N^-) + u'(x_N^+)\,v(x_N^+)\right)\right) - u'(x_{N+1}^-)\,v(x_{N+1}^-)$$

$$= -\sum_{j=0}^{j=N}\int_{x_j}^{x_{j+1}} u'v' + u'(x_1^+)\,v(x_1^+) + \sum_{j=1}^{j=N}[\boldsymbol{u'v}]_{x_j} \quad u'(x_{N+1}^-)\,v(x_{N+1}^-)$$

Jumps Conditions

$$[u'v]_{x_j} = -u'(x_j^-)\,v(x_j^-) + u'(x_j^+)\,v(x_j^+)$$

$$= \{\tfrac{1}{2}\left(u'(x_j^+) - u'(x_j^-)\right)\}[v(x_j^+) - v(x_j^-)] + \{u'(x_j^+) - u'(x_j^-)\}\{\tfrac{1}{2}\left(v(x_j^+) + v(x_j^-)\right)\}$$

$$= \{u'\}_j[v]_j + [u']_j\{v\}_j$$

Bilinear Function:

$$a(u,v) = \sum_{j=0}^{j=N}\int_{x_j}^{x_{j+1}} u'v' + \sum_{j=0}^{j=N+1}\left(\{u'\}_j[v]_j + \{v'\}_j[u]_j\right) + \underbrace{\gamma\sum_{j=0}^{N+1}\frac{1}{|I_j|}[u]_j[v]_j}_{\text{penalty term}}$$

Parallel Computing over Serial Computing

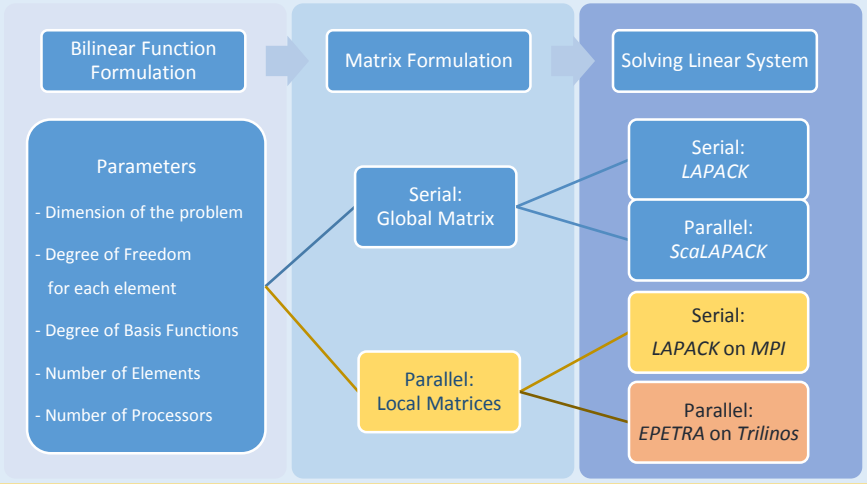
LAPACK / ScaLAPACK
- Dense matrix solver only

DG-FEM Stiffness Matrix
- Positive Definite - Sparse

Less time
Less storage
Larger problem

EPETRA on Trilinos
- Compressed row storage
- Also sparse matrix solver

Parallelization



Parallel Code Performance Analysis

- Solving time decrease with number of processors:

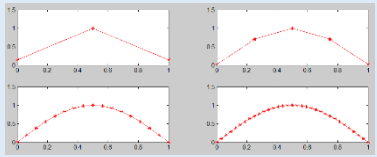
Example: number of cells equal to 100000

- Different type of solvers and preconditions may be used in *EPETRA*
  - *typically: Conjugate Gradient method with precondition (positive definite sparse matrix)*

Current Work

✓ Construct the 1-Dimension DG-FEM partial parallel code (*LAPACK* on *MPI*)

✓ Construct the 1-Dimension DG-FEM fully parallel code (*EPETRA* on *Trilinos*)



1-Dimension DG-FEM example on increase of elements

Future Work

? Extend to 2-Dimension and 3-Dimension DG-FEM fully parallel code

? Expand the code to be adaptive for local refinement, and cover chemical transport equations

Contact Information

Ng Yin-ki : kiking0501@gmail.com
Kwai Wong : kwong@utk.edu