Multi-dimensional Parallel Discontinuous Galerkin Method

Zhe Zhu
7/14/2016
Content

1. Abstract
2. Introduction of DG-FEM
3. How DG works
4. Math behind DG
5. Sample result of 1D cases
Abstract

• 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 scientific transport phenomena.

• The goal of my project is to implement DG-FEM in 3D to solve a set of partial differential equations in parallel on HPC platform.
Discontinuous Galerkin Method (DG-FEM)

For a Poisson’s equation:

\[
\begin{align*}
-\Delta u &= f \quad \text{in} \quad \Omega \\
u &= g_d \quad \text{on} \quad \Gamma_D \\
\frac{\partial u}{\partial n} &= g_d \quad \text{on} \quad \Gamma_D
\end{align*}
\]

Choose test functions \( v \) can be chosen to transform the equation into the weak form of the differential equation:

\[
- \int_\Omega \Delta u \, v \, dx = \int_\Omega \nabla u \cdot \nabla v \, dx - \int_{\partial \Omega} (\nabla u \cdot n) v \, ds = \int_\Omega \nabla u \cdot \nabla v \, dx - \int_{\partial \Omega} \frac{\partial u}{\partial n} v \, ds = \int_\Omega f \, v \, dx
\]

DG-FEM chooses test functions that are discontinuous across adjacent elements, resulting in jump conditions on the shared boundaries.
Why DG-FEM

Discontinuity between element boundaries provides local support and leads to:

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

An example of test function on 1D

(\(v^+, v^-\): test function values on element boundaries)
How DG works

- Given equation
- Mathematical deduction
  - Weak form
    - LHS
    - RHS
- Mathematical transform
  - $a_h(u,v)$
  - $f(v)$
- $a_h(u,v)$
  - Element term
  - Interior edge term
  - Boundary edge term
    - Penalty term
    - Jump term
  - Diagonal local block
  - Local block
  - Local block
- Global matrix $A$
- Solve $Ax=b$

- Divided into three parts
- Computed in parallel
- Combine all local blocks
- Use Trilinos to finish parallel solving
Get the weak form of the equation

Weak formulation using test function $v$:

$$-\int_{\Omega} \Delta u v \, dx = - \sum_{K \in T_h} \int_{K} \Delta u v \, dx$$

$$= \sum_{K \in T_h} \int_{K} \nabla u \cdot \nabla v \, dx - \sum_{K \in T_h} \int_{\partial K} \frac{\partial u}{\partial n} v \, ds$$

$$= \sum_{K \in T_h} \int_{K} \nabla u \cdot \nabla v \, dx - \sum_{e_h \in \partial T_h} \int_{e_h} \frac{\partial u}{\partial n} v \, ds - \sum_{e_h \in \partial T_h^N} \int_{e_h} \frac{\partial u}{\partial n} v \, ds$$

$$- \sum_{e_h \in \partial T_h^I} \int_{e_h} \left( \frac{\partial u^+}{\partial n^+} v^+ + \frac{\partial u^-}{\partial n^-} v^- \right) ds$$

$$= \int_{\Omega} f v \, dx$$
Bilinear Function for Stiffness Matrix:

\[ a_h(u, v) \equiv \sum_{K \in T_h} (\nabla u, \nabla v)_K - \sum_{e_h \in E_h^i} \left( <\partial_n u, [v]>_{e_h} + <\partial_n v, [u]>_{e_h} - \frac{\gamma}{|e_h|} <[u], [v]>_{e_h} \right) \]

\[ - \sum_{e_h \in E_h^D} \left( <\partial_n u, v>_{e_h} + <\partial_n v, u>_{e_h} - \frac{\gamma}{|e_h|} <u, v>_{e_h} \right) \]

: element term \hspace{1cm} : jump term \hspace{1cm} : penalty term

Solving Linear System:

\[
\sum_{j=1}^{j=M} \alpha(\phi_j, \phi_i) \alpha_j = \int_{\Omega} f \phi_i + \text{symmetric term} + \text{penalty term}
\]
Multi-dimensional jump term

2D:

3D:
Sample result

1D Element term:

Local matrices

\[
\begin{pmatrix}
  4.00 & -4.00 \\
  -4.00 & 4.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
  0.00 & 0.00 \\
\end{pmatrix}
\]
Sample result

1D Jump term:

Local matrices

\[
\begin{pmatrix}
-8.00 & 4.00 \\
4.00 & 0.00
\end{pmatrix}
\] (Boundary node)

\[
\begin{pmatrix}
0.00 & 2.00 & -2.00 & 0.00 \\
2.00 & -4.00 & 4.00 & -2.00 \\
-2.00 & 4.00 & -4.00 & 2.00 \\
0.00 & -2.00 & 2.00 & 0.00
\end{pmatrix}
\]

\[
\begin{pmatrix}
0.00 & 4.00 \\
4.00 & -8.00
\end{pmatrix}
\] (Boundary node)
Sample result

1D Penalty term:
Future works

- Extend the partial differential equation to some time-dependent equations
- Expand the equation to parallel code, which can be scaled on existing supercomputers.
Q & A