Multi-dimensional Parallel Discontinuous Galerkin Method

Zhe Zhu

7/14/2016

Content

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:

$$\begin{cases} -\Delta u = f & in \quad \Omega \\ u = g_d & on \quad \Gamma_D \\ \dfrac{\partial u}{\partial \vec{n}} = g_d & on \quad \Gamma_D \end{cases}$$

a test functions v can be choose to transform the equation into the weak form of the differential equation:

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

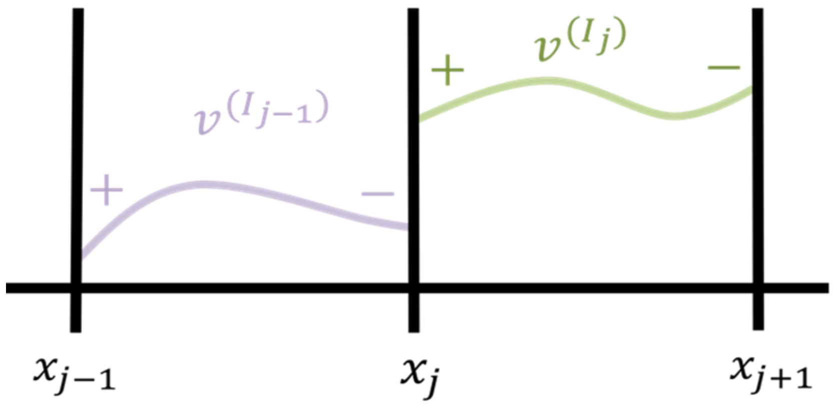
DG-FEM chooses test functions that are discontinuous across adjacent elements, resulting jump conditions on the shared boundaries.

Why DG-FEM

Discontinuity between element boundaries

provides local support and leads to :

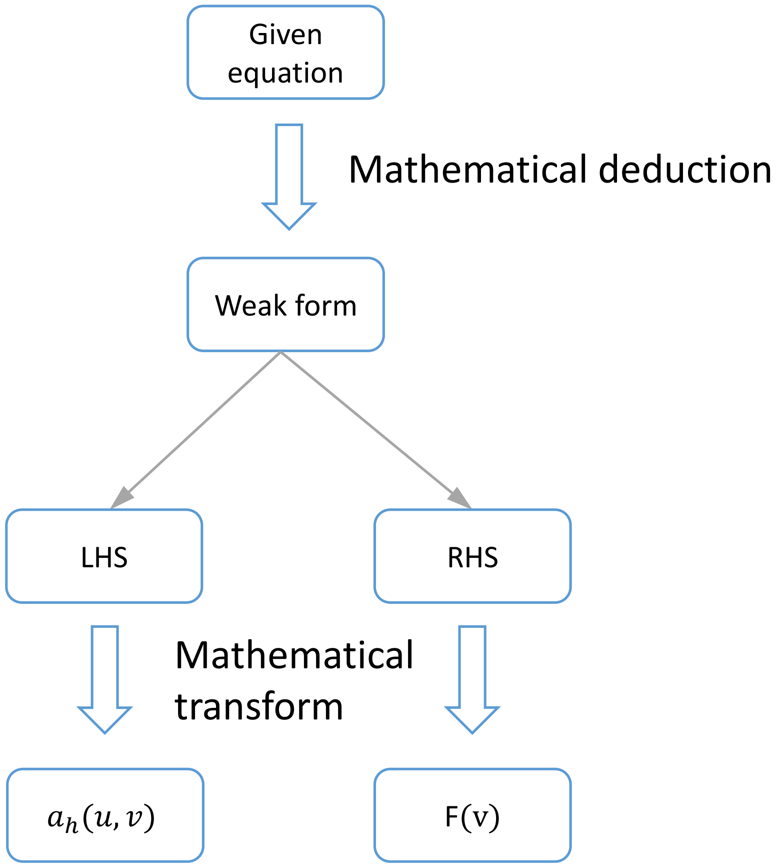
- Local refinement
- Complex geometries
- Parallelization
- Higher-order accuracy

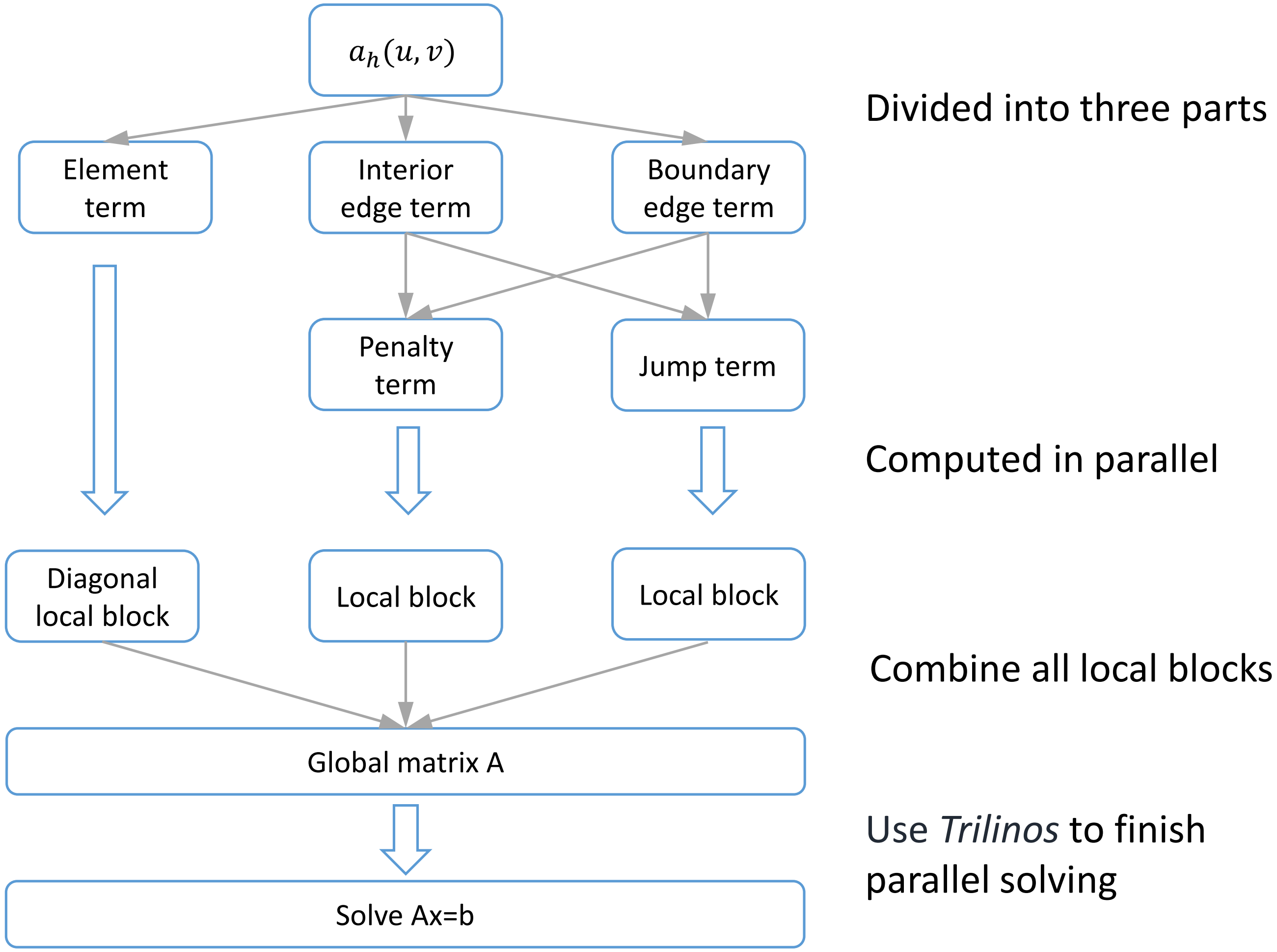


An example of test function on 1D

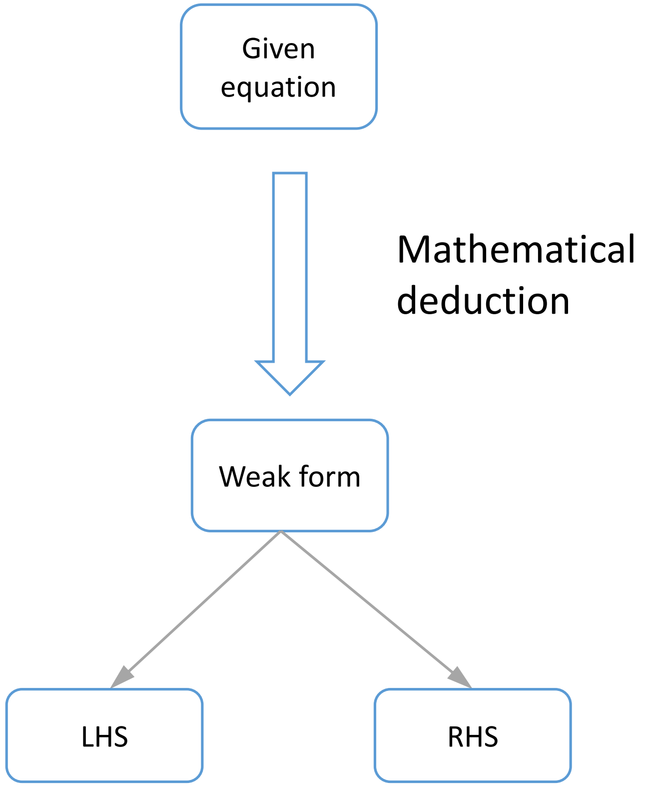
(v+, v- : test function values on element boundaries)

How DG works


Given equation
Mathematical deduction
Weak form
LHS
RHS
Mathematical transform
$a_h(u, v)$
F(v)


$a_h(u, v)$
Divided into three parts
Element term
Interior edge term
Boundary edge term
Penalty term
Jump term
Computed in parallel
Diagonal local block
Local block
Local block
Combine all local blocks
Global matrix A
Use *Trilinos* to finish parallel solving
Solve Ax=b

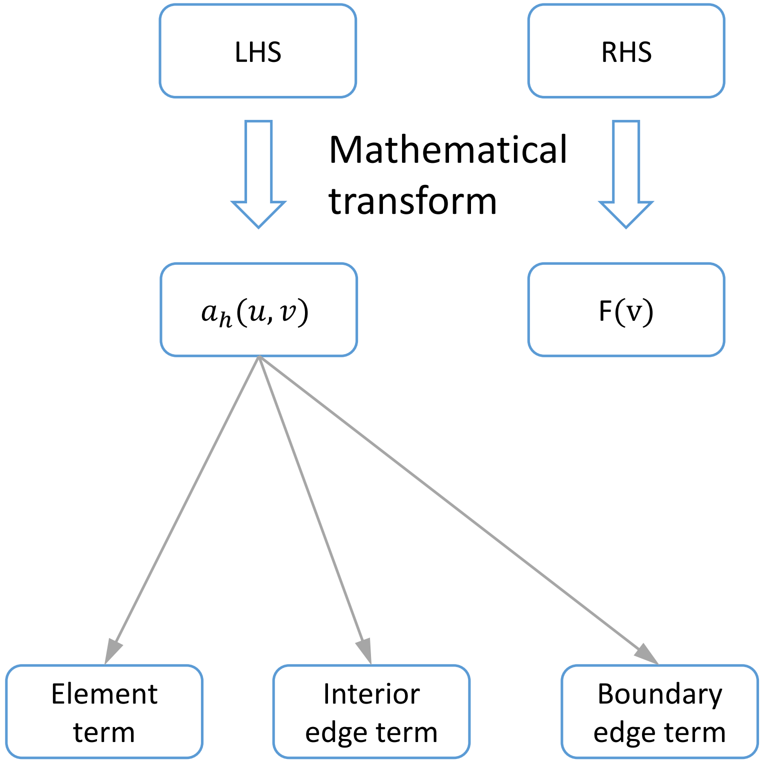
Get the weak form of the equation



Weak formulation using test function *v* :

$$
\begin{aligned}
-\int_{\Omega} \Delta u \, v \, dx &= -\sum_{K \in \mathcal{T}_h} \int_K \Delta u \, v \, dx \\
&= \sum_{K \in \mathcal{T}_h} \int_K \nabla u \cdot \nabla v \, dx - \sum_{K \in \mathcal{T}_h} \int_{\partial K} \frac{\partial u}{\partial \mathbf{n}} v \, ds \\
&= \sum_{K \in \mathcal{T}_h} \int_K \nabla u \cdot \nabla v \, dx - \sum_{e_h \in \mathcal{E}_h^D} \int_{e_h} \frac{\partial u}{\partial \mathbf{n}} v \, ds - \sum_{e_h \in \mathcal{E}_h^N} \int_{e_h} \frac{\partial u}{\partial \mathbf{n}} v \, ds \\
&\quad - \sum_{e_h \in \mathcal{E}_h^I} \int_{e_h} \left(\frac{\partial u^+}{\partial \mathbf{n} n^+} v^+ + \frac{\partial u^-}{\partial \mathbf{n} n^-} v^- \right) ds \\
&= \int_{\Omega} f \, v \, dx
\end{aligned}
$$

Bilinear Function for Stiffness Matrix



Bilinear Function for Stiffness Matrix:

$$a_h(u,v) \equiv \sum_{K \in \mathcal{T}_h} (\nabla u, \nabla v)_K - \sum_{e_h \in \mathcal{E}_h^I} \left(< \{\partial_n u\}, [v] >_{e_h} + < \{\partial_n v\}, [u] >_{e_h} - \frac{\gamma}{|e_h|} < [u], [v] >_{e_h} \right)$$

$$- \sum_{e_h \in \mathcal{E}_h^D} \left(< \partial_n u, v >_{e_h} + < \partial_n v, u >_{e_h} - \frac{\gamma}{|e_h|} < u, v >_{e_h} \right)$$

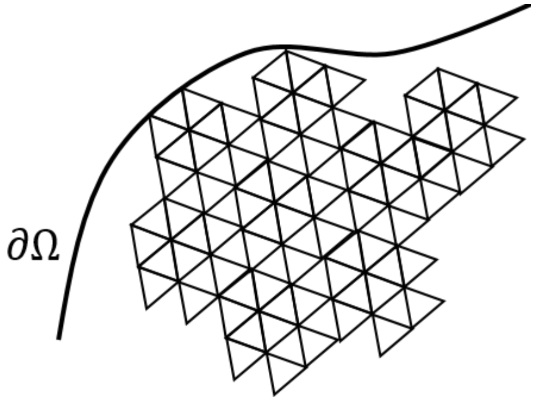
:element term : jump term :penalty term

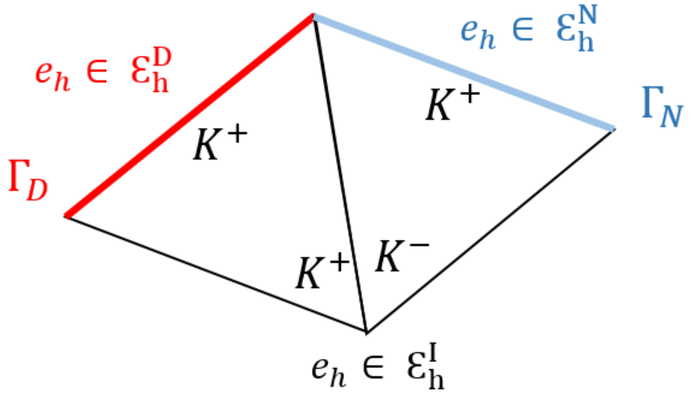
Solving Linear System:

$$\sum_{j=1}^{j=M} \underbrace{a(\phi_j, \phi_i)}_{S_{ij}} \alpha_j = \underbrace{\int f \, \phi_i}_{r_i} + \text{symmetric term} + \text{penalty term}$$

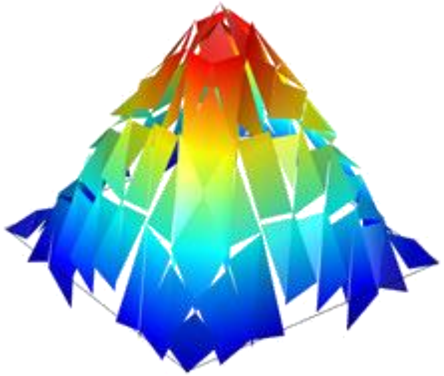
Multi-dimensional jump term

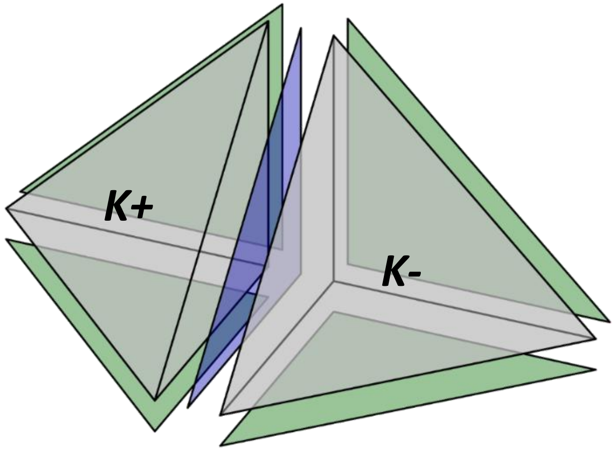
2D:





3D:





Sample result

1D Element term:

Local matrices

$$\begin{pmatrix} 4.00 & -4.00 \\ -4.00 & 4.00 \end{pmatrix}$$

$$\begin{bmatrix} 4.00 & -4.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -4.00 & 4.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 4.00 & -4.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -4.00 & 4.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 4.00 & -4.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -4.00 & 4.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 4.00 & -4.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -4.00 & 4.00 \end{bmatrix}$$

Sample result

1D Jump term:

Local matrices

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

$$\begin{bmatrix} 0.00 & 2.00 & -2.00 & 0.00 \\ 2.00 & -4.00 & 4.00 & -2.00 \\ -2.00 & 4.00 & -4.00 & 2.00 \\ 0.00 & -2.00 & 2.00 & 0.00 \end{bmatrix}$$

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

$$\begin{bmatrix} -8.00 & 6.00 & -2.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 6.00 & -4.00 & 4.00 & -2.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ -2.00 & 4.00 & -4.00 & 4.00 & -2.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & -2.00 & 4.00 & -4.00 & 4.00 & -2.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & -2.00 & 4.00 & -4.00 & 4.00 & -2.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -2.00 & 4.00 & -4.00 & 4.00 & -2.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & -2.00 & 4.00 & -4.00 & 6.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -2.00 & 6.00 & -8.00 \end{bmatrix}$$

Sample result

1D Penalty term:

Local matrices

$$\begin{pmatrix} 20.00 & 0.00 \\ 0.00 & 0.00 \end{pmatrix} \text{(Boundary node)}$$

$$\begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 20.00 & -20.00 & 0.00 \\ 0.00 & -20.00 & 20.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 \end{bmatrix}$$

$$\begin{pmatrix} 0.00 & 0.00 \\ 0.00 & 20.00 \end{pmatrix} \text{(Boundary node)}$$

$$\begin{bmatrix} 20.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 20.00 & -20.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & -20.00 & 20.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 20.00 & -20.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & -20.00 & 20.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 20.00 & -20.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & -20.00 & 20.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 20.00 \end{bmatrix}$$

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