

## 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

For a Poisson's equation:

$$\begin{cases} -\Delta u = f & \text{in } \Omega \\ u = g_d & \text{on } \Gamma_D \\ \frac{\partial u}{\partial \mathbf{n}} = g_n & \text{on } \Gamma_N \end{cases}$$

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.

## Linear System Construction

Weak formulation using test function  $v$ :

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

Bilinear Function for Stiffness Matrix:

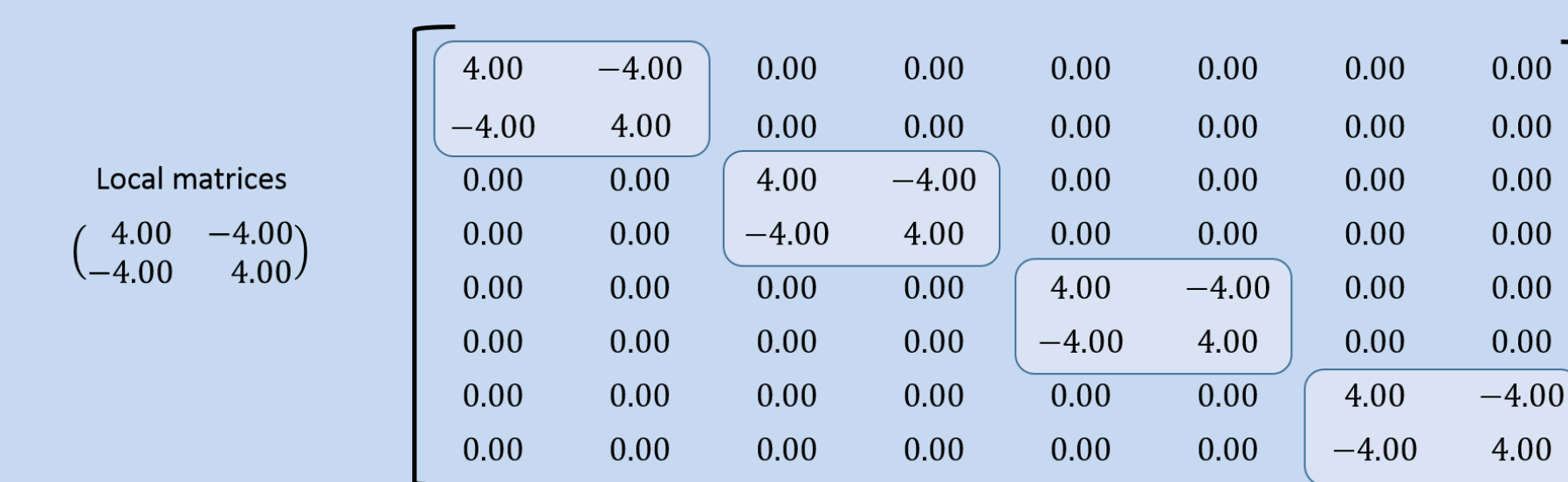
$$\begin{aligned} a_h(u, v) \equiv & \underbrace{\sum_{K \in \mathcal{T}_h} (\nabla u, \nabla v)_K}_{\text{:element term}} - \underbrace{\sum_{e_h \in \mathcal{E}_h^I} \left( \langle \partial_n u \rangle, [v] \rangle_{e_h} + \langle \partial_n v \rangle, [u] \rangle_{e_h} \right)}_{\text{: jump term}} - \underbrace{\sum_{e_h \in \mathcal{E}_h^D} \left( \langle \partial_n u, v \rangle_{e_h} + \langle \partial_n v, u \rangle_{e_h} - \frac{\gamma}{|e_h|} \langle u, v \rangle_{e_h} \right)}_{\text{:penalty term}} \end{aligned}$$

Solving Linear System:

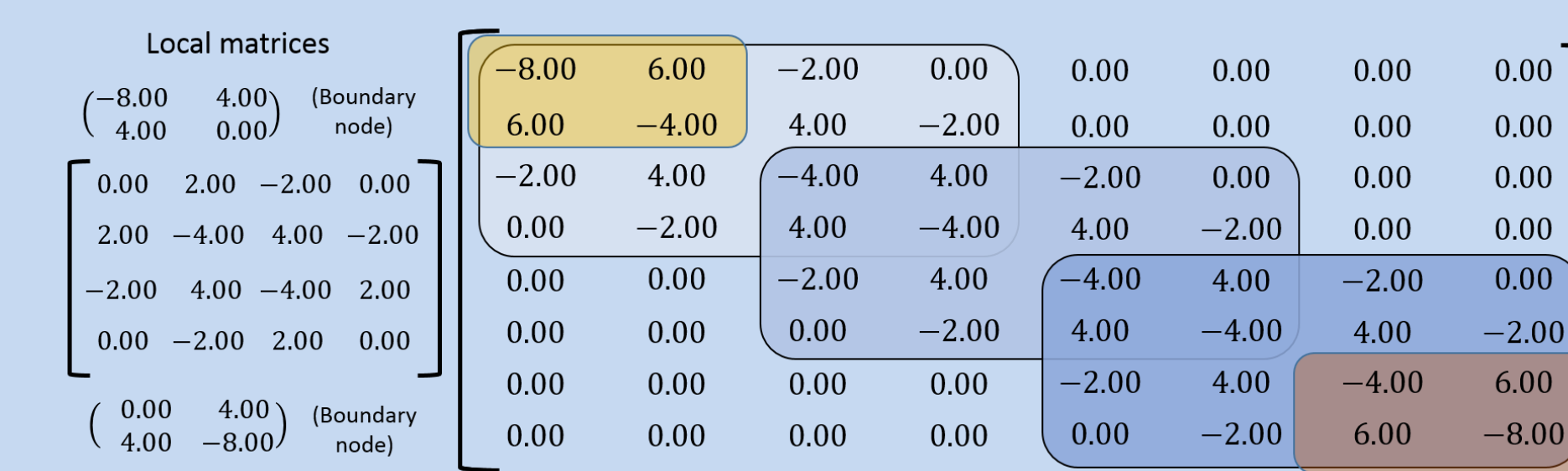
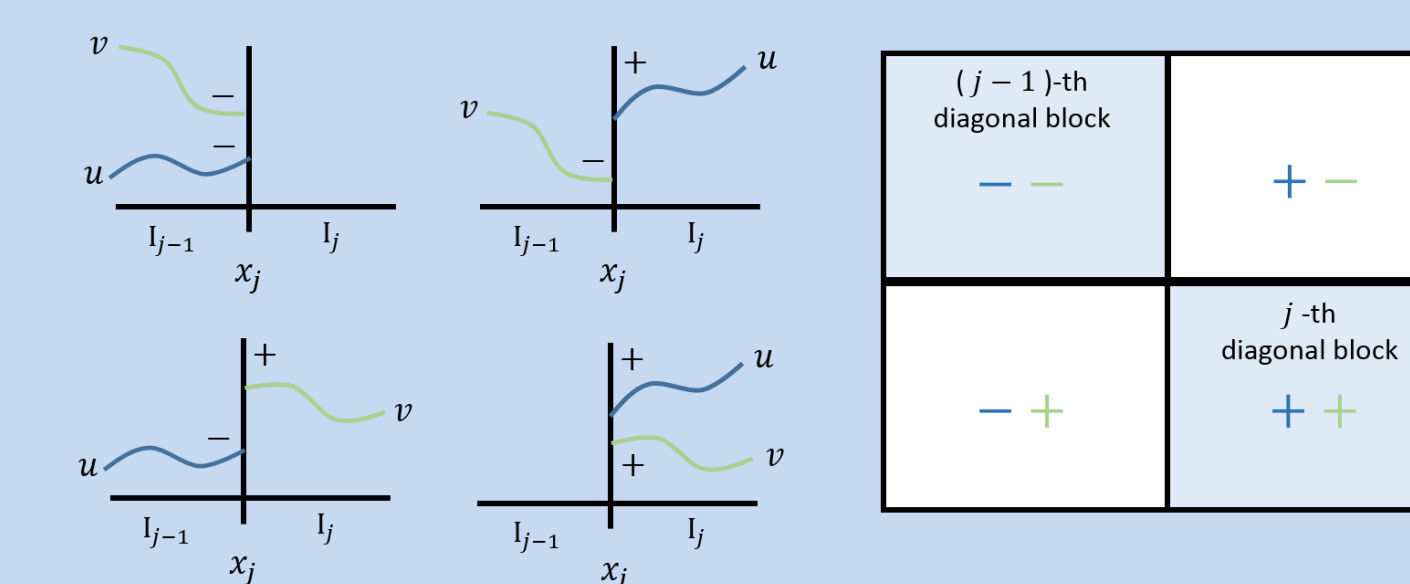
$$\sum_{j=1}^M \underbrace{a(\phi_j, \phi_i)}_{S_{ij}} \alpha_j = \underbrace{\int f \phi_i}_{r_i} + \text{symmetric term} + \text{penalty term}$$

## Sample Result

Element term:



Jump term:

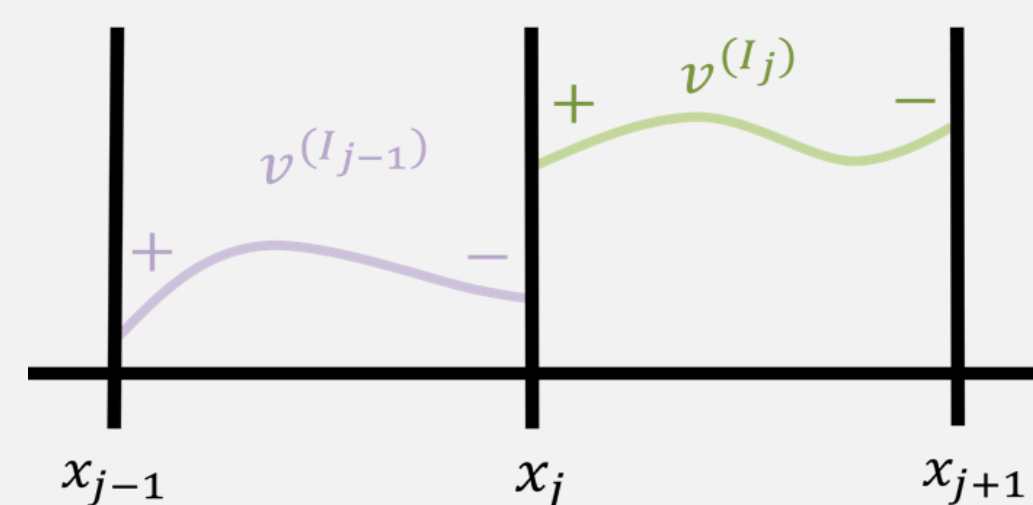


Penalty term:

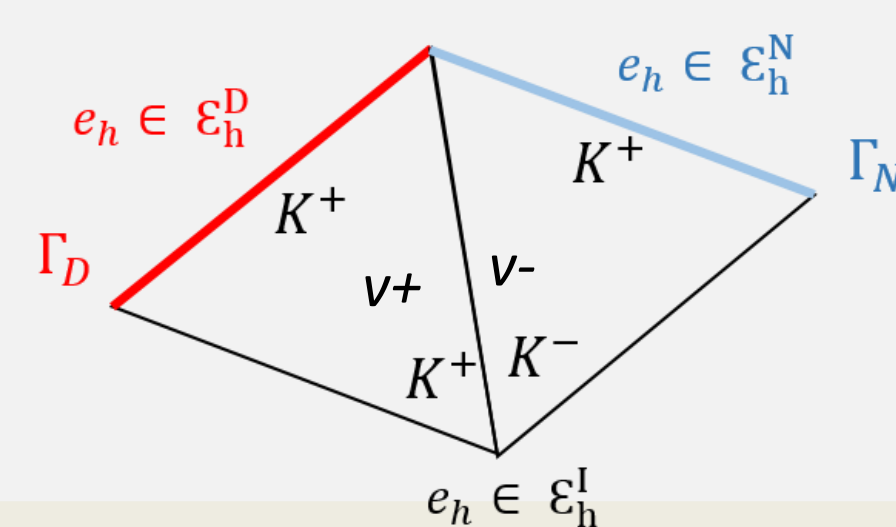


## Multi-dimensional Jump Cases

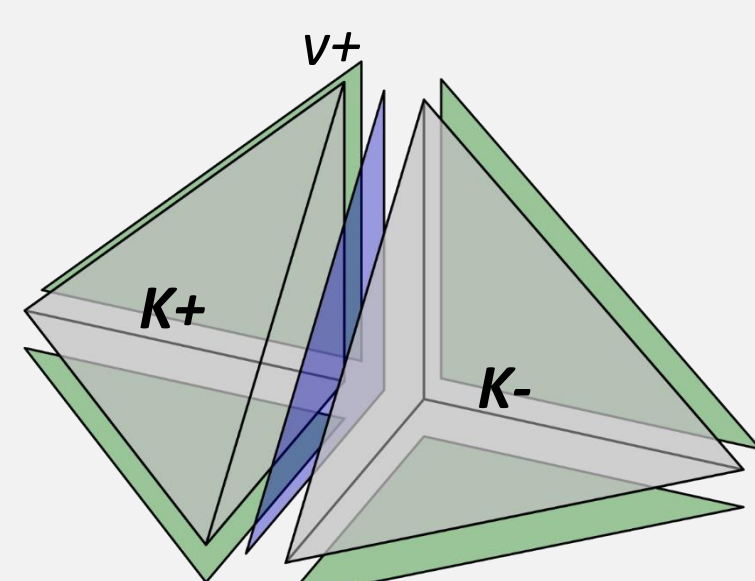
1D:



2D:

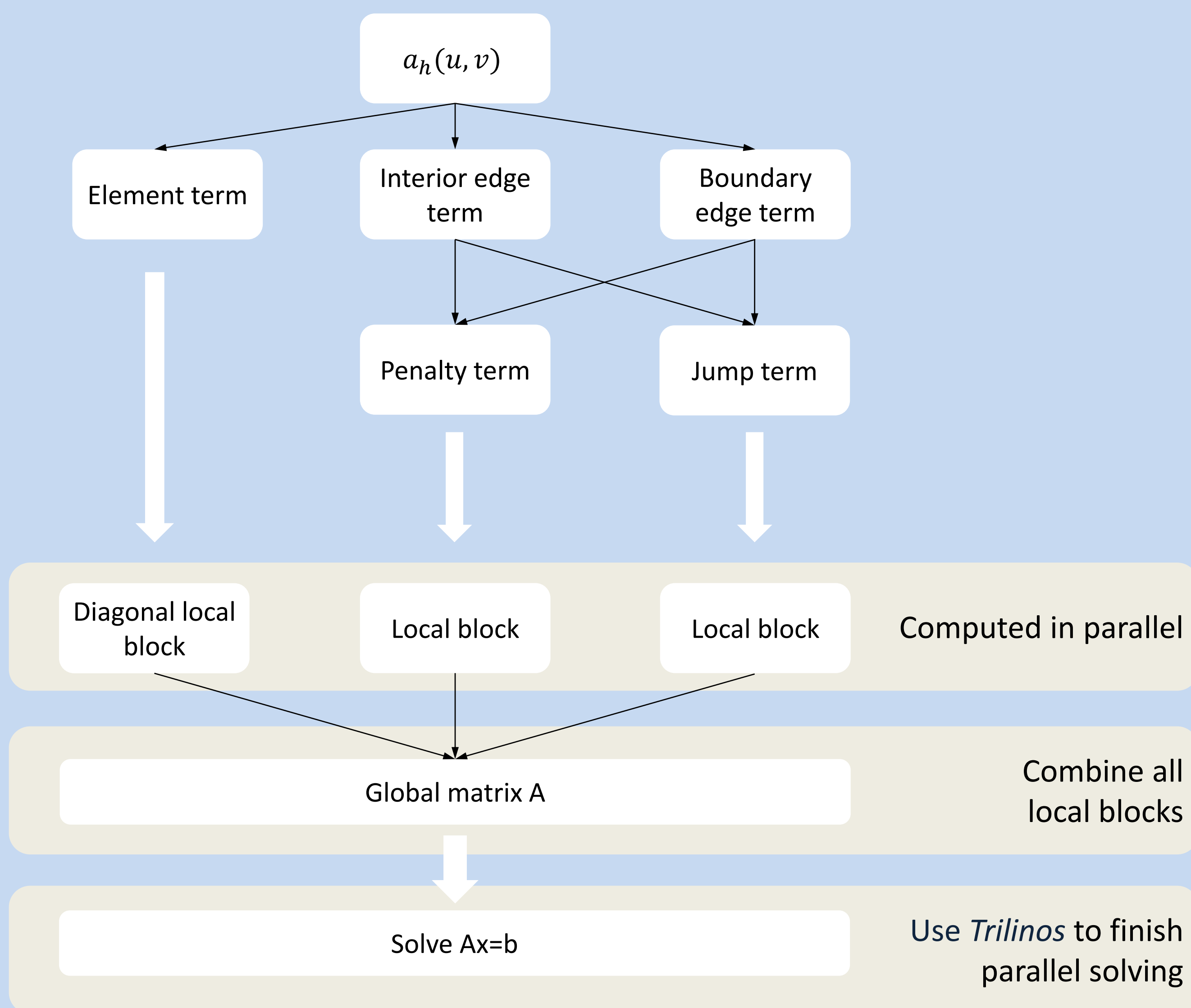


3D:



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

## Parallel Computing



## Why DG-FEM matters

Discontinuity between element boundaries provides local support and leads to :

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

## Future

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

## Acknowledgements

This project was sponsored by Oak Ridge National Laboratory, Joint Institute for Computational Sciences, University of Tennessee, Knoxville and the Chinese University of Hong Kong.