Multilevel Solvers for Discontinuous Galerkin Methods

Maurice Fabien
Matthew G. Knepley
Béatrice M. Rivierè

Department of Computational and Applied Mathematics
Rice University

June 10, 2016

Main Objective:

Design and analyze efficient solution techniques for $hp-$discontinous Galerkin discretizations of

$$-\nabla \cdot (\mathbf{K} \nabla u) = f \quad \text{in } \Omega \subset \mathbb{R}^d$$
$$u = 0 \quad \text{on } \partial\Omega.$$

Discontinuous Galerkin Formulation

Let $\mathcal{E}_h$ be a subdivision of Ω.

Discontinuous Galerkin Formulation

Let $\mathcal{E}_h$ be a subdivision of Ω.

Discontinuous finite element space:

$$\mathcal{D}_k(\mathcal{E}_h) = \{v \in L^2(\Omega) : \ \forall E \in \mathcal{E}_h, \ v|_E \in \mathbb{Q}_k(E)\}.$$

Discontinuous Galerkin Formulation

Let $\mathcal{E}_h$ be a subdivision of Ω.

Discontinuous finite element space:

$$\mathcal{D}_k(\mathcal{E}_h) = \{v \in L^2(\Omega) : \ \forall E \in \mathcal{E}_h, \ v|_E \in \mathbb{Q}_k(E)\}.$$

Weak formulation of PDE: find $u_h \in \mathcal{D}_k(\mathcal{E}_h)$ such that

$$\forall v \in \mathcal{D}_k(\mathcal{E}_h), \quad a_\epsilon(u_h, v) = \int_\Omega f v,$$

Discontinuous Galerkin Formulation

Let $\mathcal{E}_h$ be a subdivision of Ω.

Discontinuous finite element space:

$$\mathcal{D}_k(\mathcal{E}_h) = \{v \in L^2(\Omega) : \ \forall E \in \mathcal{E}_h, \ v|_E \in \mathbb{Q}_k(E)\}.$$

Weak formulation of PDE: find $u_h \in \mathcal{D}_k(\mathcal{E}_h)$ such that

$$\forall v \in \mathcal{D}_k(\mathcal{E}_h), \quad a_\epsilon(u_h, v) = \int_\Omega f v,$$

where the bilinear form a_ϵ is given by

$$a_\epsilon(u_h, v) = \sum_{E \in \mathcal{E}_h} \int_E \mathbf{K} \nabla u_h \cdot \nabla v - \sum_{e \in \Gamma_h \cup \Gamma_D} \int_e \{\{\mathbf{K} \nabla u_h \cdot \vec{n}_e\}\} [\![v]\!]$$
$$+ \epsilon \sum_{e \in \Gamma_h \cup \Gamma_D} \int_e \{\{\mathbf{K} \nabla v \cdot \vec{n}_e\}\} [\![u_h]\!] + \sum_{e \in \Gamma_h \cup \Gamma_D} \frac{\sigma_e}{|e|^{\beta_0}} \int_e [\![u_h]\!] [\![v]\!].$$

This gives rise to a large sparse linear system:

$$\mathbf{A}\vec{u} = \vec{f}.$$

This gives rise to a large sparse linear system:

$$\mathbf{A}\vec{u} = \vec{f}.$$

To solve this system we employ a Schwarz domain decomposition method:

$$B = \mathbf{R}_0 \mathbf{A}_0^{-1} \mathbf{R}_0^T + \sum_{j=1}^{N_S} \mathbf{R}_j \mathbf{A}_j^{-1} \mathbf{R}_j^T$$

This gives rise to a large sparse linear system:

$$\mathbf{A}\vec{u} = \vec{f}.$$

To solve this system we employ a Schwarz domain decomposition method:

$$B = \mathbf{R}_0 \mathbf{A}_0^{-1} \mathbf{R}_0^T + \sum_{j=1}^{N_S} \mathbf{R}_j \mathbf{A}_j^{-1} \mathbf{R}_j^T$$

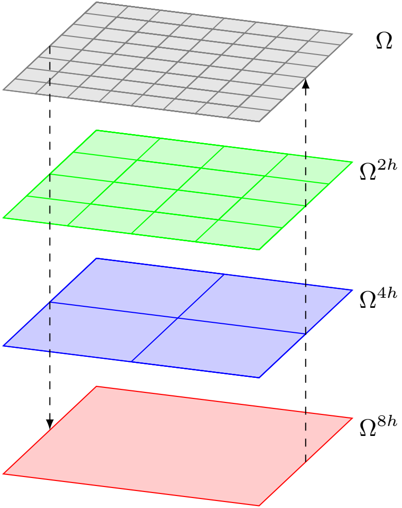
R_j are elliptic projection operators, and $\mathbf{A}_j$ are the restriction of $a_\epsilon(\cdot,\cdot)$ to a subdomain Ω_j.

Schwarz domain decomposition

Parition the domain $\Omega = \cup_{j=1}^{N_S} \Omega_j$, pick a coarse grid ($\Omega^{2h}$, Ω^{4h}, or Ω^{8h})

Schwarz domain decomposition

Parition the domain $\Omega = \cup_{j=1}^{N_S} \Omega_j$, pick a coarse grid ($\Omega^{2h}$, Ω^{4h}, or Ω^{8h})



How does can we implement this in parallel?

$$B = \mathbf{R}_0 \mathbf{A}_0^{-1} \mathbf{R}_0^T + \sum_{j=1}^{N_S} \mathbf{R}_j \mathbf{A}_j^{-1} \mathbf{R}_j^T$$

How does can we implement this in parallel?

$$B = \mathbf{R}_0 \mathbf{A}_0^{-1} \mathbf{R}_0^T + \sum_{j=1}^{N_S} \mathbf{R}_j \mathbf{A}_j^{-1} \mathbf{R}_j^T$$

- Each term of B can be computed independently.

How does can we implement this in parallel?

$$B = \mathbf{R}_0 \mathbf{A}_0^{-1} \mathbf{R}_0^T + \sum_{j=1}^{N_S} \mathbf{R}_j \mathbf{A}_j^{-1} \mathbf{R}_j^T$$

- Each term of B can be computed independently.
- The **coarse** grid operator $(\mathbf{R}_0 \mathbf{A}_0^{-1} \mathbf{R}_0^T)$ is amenable to **coarse** grain parallelism

How does can we implement this in parallel?

$$B = \mathbf{R}_0 \mathbf{A}_0^{-1} \mathbf{R}_0^T + \sum_{j=1}^{N_S} \mathbf{R}_j \mathbf{A}_j^{-1} \mathbf{R}_j^T$$

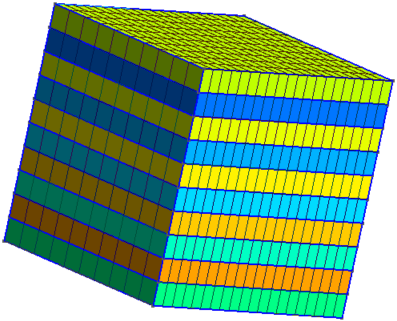
- Each term of B can be computed independently.
- The **coarse** grid operator $(\mathbf{R}_0 \mathbf{A}_0^{-1} \mathbf{R}_0^T)$ is amenable to **coarse** grain parallelism
- The **fine** grid operators $(\mathbf{R}_j \mathbf{A}_j^{-1} \mathbf{R}_j^T)$ are amenable to **fine** grain parallelism

Hybrid computing strategy

- Global partition of mesh

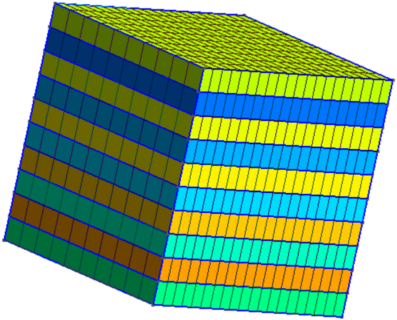
Hybrid computing strategy

- Global partition of mesh



Hybrid computing strategy

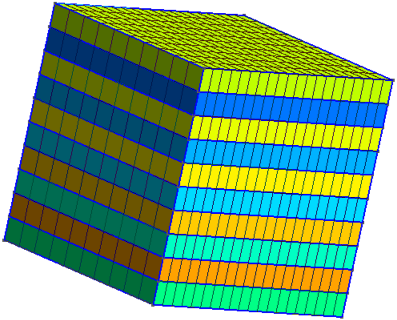
- Global partition of mesh



- Each layer of hexahedra is processed by MPI

Hybrid computing strategy

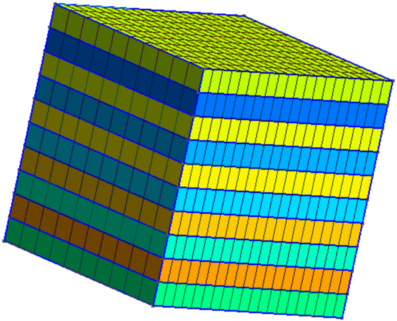
- Global partition of mesh



- Each layer of hexahedra is processed by MPI
- Layers are sent to coprocessors for fine grain parallelism

Hybrid computing strategy

- Global partition of mesh



- Each layer of hexahedra is processed by MPI
- Layers are sent to coprocessors for fine grain parallelism
- Coarse grid is handled by a small number of MPI ranks

Closer look at mapping to coprocessor

Closer look at mapping to coprocessor