Adaptive algebraic multigrid methods and Helmholtz decompositions on graphs

Ludmil T Zikatanov

Department of Mathematics, Penn State, USA

Interplay of discretization and algebraic solvers:
a posteriori error estimates and adaptivity

June 8-10, 2022 @ INRIA, Paris, France

Joint with: Xiaozhe Hu(Tufts), James H Adler (Tufts),
A. Budiša, M. Kuchta, and K.-A. Mardal (UiO and Simula, Oslo)
Yuwen Li (Penn State), Junyuan Lin (Loyola Marymount U), and
Kaiyi Wu(Tufts)

Talking points

- Interplay between....
  - How **preconditioning** provides efficient and reliable a posteriori error indicators for discretized PDEs.
  - How **a posteriori error indicators on graphs** provide multilevel hierarchies for AMG.

Operator preconditioning

Setup:

- Hilbert space $\mathcal{H}$ equipped with inner product $(\cdot,\cdot)_{\mathcal{H}}$ and norm $\|\cdot\|_{\mathcal{H}}$
- Operator $\mathbf{A} : \mathcal{H} \mapsto \mathcal{H}'$

Linear problem: given $\mathbf{f} \in \mathcal{H}'$, find $\mathbf{u} \in \mathcal{H}$ such that $\mathbf{Au} = \mathbf{f}$

Well-posedness (is $\mathbf{A}$ an isomorphism?):

Continuity of $\mathbf{A}$: $$\sup_{0\neq \mathbf{x}\in\mathcal{H}} \sup_{0\neq \mathbf{y}\in\mathcal{H}} \frac{\langle \mathbf{Ax}, \mathbf{y}\rangle}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \leq \beta$$

Continuity of $\mathbf{A}^{-1}$ (inf-sup condition): $$\inf_{0\neq \mathbf{x}\in\mathcal{H}} \sup_{0\neq \mathbf{y}\in\mathcal{H}} \frac{\langle \mathbf{Ax}, \mathbf{y}\rangle}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \geq \gamma > 0$$

Example: Stokes equation
$$\mathbf{Ax} = \mathbf{f} \Longrightarrow \begin{pmatrix} -\Delta & \operatorname{div}^* \\ \operatorname{div} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix}$$

where $\mathcal{H} = [H_0^1]^3 \times L^2$, and $\|\mathbf{x}\|^2_{\mathcal{H}} := \|\nabla \boldsymbol{u}\|^2 + \|p\|^2$

Operator preconditioning

Setup:

- Hilbert space $\mathcal{H}$ equipped with inner product $(\cdot,\cdot)_{\mathcal{H}}$ and norm $\|\cdot\|_{\mathcal{H}}$
- Operator $\mathbf{A} : \mathcal{H} \mapsto \mathcal{H}'$

Linear problem: given $\mathbf{f} \in \mathcal{H}'$, find $\mathbf{u} \in \mathcal{H}$ such that $\mathbf{Au} = \mathbf{f}$

Well-posedness (is $\mathbf{A}$ an isomorphism?):

$$\text{Continuity of } \mathbf{A}: \quad \sup_{\mathbf{0}\neq\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{0}\neq\mathbf{y}\in\mathcal{H}} \frac{\langle \mathbf{Ax}, \mathbf{y}\rangle}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \leq \beta$$

$$\text{Continuity of } \mathbf{A}^{-1} \text{ (inf-sup condition)}: \quad \inf_{\mathbf{0}\neq\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{0}\neq\mathbf{y}\in\mathcal{H}} \frac{\langle \mathbf{Ax}, \mathbf{y}\rangle}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \geq \gamma > 0$$

Example: Stokes equation

$$\mathbf{Ax} = \mathbf{f} \Longrightarrow \begin{pmatrix} -\Delta & \mathrm{div}^* \\ \mathrm{div} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix}$$

where $\mathcal{H} = [H_0^1]^3 \times L^2$, and $\|\mathbf{x}\|_{\mathcal{H}}^2 := \|\nabla \boldsymbol{u}\|^2 + \|p\|^2$

Operator preconditioning

Setup:

- Hilbert space $\mathcal{H}$ equipped with inner product $(\cdot,\cdot)_{\mathcal{H}}$ and norm $\|\cdot\|_{\mathcal{H}}$
- Operator $\mathbf{A} : \mathcal{H} \mapsto \mathcal{H}'$

Linear problem: given $\mathbf{f} \in \mathcal{H}'$, find $\mathbf{u} \in \mathcal{H}$ such that $\mathbf{A}\mathbf{u} = \mathbf{f}$

Well-posedness (is $\mathbf{A}$ an isomorphism?):

$$\text{Continuity of } \mathbf{A}: \quad \sup_{\mathbf{0}\neq\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{0}\neq\mathbf{y}\in\mathcal{H}} \frac{\langle \mathbf{A}\mathbf{x}, \mathbf{y}\rangle}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \leq \beta$$

$$\text{Continuity of } \mathbf{A}^{-1} \text{ (inf-sup condition)}: \quad \inf_{\mathbf{0}\neq\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{0}\neq\mathbf{y}\in\mathcal{H}} \frac{\langle \mathbf{A}\mathbf{x}, \mathbf{y}\rangle}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \geq \gamma > 0$$

Example: Stokes equation

$$\mathbf{A}\mathbf{x} = \mathbf{f} \Longrightarrow \begin{pmatrix} -\Delta & \mathrm{div}^* \\ \mathrm{div} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{u} \\ p \end{pmatrix} = \begin{pmatrix} \boldsymbol{f} \\ 0 \end{pmatrix}$$

where $\mathcal{H} = [H_0^1]^3 \times L^2$, and $\|\mathbf{x}\|_{\mathcal{H}}^2 := \|\nabla \boldsymbol{u}\|^2 + \|p\|^2$

Operator preconditioning

Preconditioner B

$$\mathbf{Au} = \mathbf{f} \Longrightarrow \mathbf{BAu} = \mathbf{Bf}$$

Requirements on $\mathbf{B}$: $\kappa(\mathbf{BA}) = \|BA\|\|(BA)^{-1}\| = \mathcal{O}(1) \ll \kappa(\mathbf{A})$

$\mathbf{B} \approx \mathbf{A}^{-1}$ and the action of $\mathbf{B}$ is easy to compute

- Apply Krylov iterative methods to the preconditioned system

Operator preconditioning

Preconditioner B

$$\mathbf{Au} = \mathbf{f} \Longrightarrow \mathbf{BAu} = \mathbf{Bf}$$

Requirements on $\mathbf{B}$: $\kappa(\mathbf{BA}) = \|BA\|\|(BA)^{-1}\| = \mathcal{O}(1) \ll \kappa(\mathbf{A})$

$\mathbf{B} \approx \mathbf{A}^{-1}$ and the action of $\mathbf{B}$ is easy to compute

- Apply Krylov iterative methods to the preconditioned system

Operator preconditioning

Preconditioner B

$$\mathbf{A}\mathbf{u} = \mathbf{f} \Longrightarrow \mathbf{B}\mathbf{A}\mathbf{u} = \mathbf{B}\mathbf{f}$$

Requirements on **B**: $\kappa(\mathbf{BA}) = \|BA\|\|(BA)^{-1}\| = \mathcal{O}(1) \ll \kappa(\mathbf{A})$

$\mathbf{B} \approx \mathbf{A}^{-1}$ and the action of $\mathbf{B}$ is easy to compute

- Apply Krylov iterative methods to the preconditioned system

Operator preconditioning

Preconditioner B

$$\mathbf{Au} = \mathbf{f} \Longrightarrow \mathbf{BAu} = \mathbf{Bf}$$

Requirements on $\mathbf{B}$: $\kappa(\mathbf{BA}) = \|BA\|\|(BA)^{-1}\| = \mathcal{O}(1) \ll \kappa(\mathbf{A})$

$\mathbf{B} \approx \mathbf{A}^{-1}$ and the action of $\mathbf{B}$ is easy to compute

- Apply Krylov iterative methods to the preconditioned system

Operator preconditioning (Mardal and Winther 2011; Loghin & Wathen 2004, Hiptmair 2006)

Problem: Given $\mathbf{f} \in \mathcal{H}'$, find $\mathbf{u} \in \mathcal{H}$ such that $\mathbf{Au} = \mathbf{f}$. Here, $\mathbf{A} : \mathcal{H} \mapsto \mathcal{H}'$ is an isomorphism.

Riesz operator: $\mathbf{B} : \mathcal{H}' \mapsto \mathcal{H}$, such that for every $\mathbf{f} \in \mathcal{H}'$,

$$\langle \mathbf{f}, \mathbf{x} \rangle = (\mathbf{Bf}, \mathbf{x})_{\mathcal{H}}, \qquad \forall\, \mathbf{x} \in \mathcal{H}$$

Estimate $\kappa(\mathbf{BA})$:

$$\|\mathbf{BA}\| = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y}\rangle|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} \le \beta$$

$$\|[\mathbf{BA}]^{-1}\|^{-1} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y}\rangle|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} \ge \gamma$$

$$\Longrightarrow \kappa(\mathbf{BA}) = \|\mathbf{BA}\| \|[\mathbf{BA}]^{-1}\| \le \frac{\beta}{\gamma}$$

- Riesz operator is a robust preconditioner!

Operator preconditioning (Mardal and Winther 2011; Loghin & Wathen 2004, Hiptmair 2006)

Problem: Given $\mathbf{f} \in \mathcal{H}'$, find $\mathbf{u} \in \mathcal{H}$ such that $\mathbf{Au} = \mathbf{f}$. Here, $\mathbf{A} : \mathcal{H} \mapsto \mathcal{H}'$ is an isomorphism.

Riesz operator: $\mathbf{B} : \mathcal{H}' \mapsto \mathcal{H}$, such that for every $\mathbf{f} \in \mathcal{H}'$,

$$\langle \mathbf{f}, \mathbf{x} \rangle = (\mathbf{Bf}, \mathbf{x})_{\mathcal{H}}, \qquad \forall\, \mathbf{x} \in \mathcal{H}$$

Estimate $\kappa(\mathbf{BA})$:

$$\|\mathbf{BA}\| = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y}\rangle|}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \leq \beta$$

$$\|[\mathbf{BA}]^{-1}\|^{-1} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y}\rangle|}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \geq \gamma$$

$$\Longrightarrow \kappa(\mathbf{BA}) = \|\mathbf{BA}\|\|[\mathbf{BA}]^{-1}\| \leq \frac{\beta}{\gamma}$$

- Riesz operator is a robust preconditioner!

Operator preconditioning (Mardal and Winther 2011; Loghin & Wathen 2004, Hiptmair 2006)

Problem: Given $\mathbf{f} \in \mathcal{H}'$, find $\mathbf{u} \in \mathcal{H}$ such that $\mathbf{Au} = \mathbf{f}$. Here, $\mathbf{A} : \mathcal{H} \mapsto \mathcal{H}'$ is an isomorphism.
Riesz operator: $\mathbf{B} : \mathcal{H}' \mapsto \mathcal{H}$, such that for every $\mathbf{f} \in \mathcal{H}'$,

$$\langle \mathbf{f}, \mathbf{x} \rangle = (\mathbf{Bf}, \mathbf{x})_{\mathcal{H}}, \qquad \forall\, \mathbf{x} \in \mathcal{H}$$

Estimate $\kappa(\mathbf{BA})$:

$$\|\mathbf{BA}\| = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y} \rangle|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} \le \beta$$

$$\|[\mathbf{BA}]^{-1}\|^{-1} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y} \rangle|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} \ge \gamma$$

$$\Longrightarrow \kappa(\mathbf{BA}) = \|\mathbf{BA}\| \|[\mathbf{BA}]^{-1}\| \le \frac{\beta}{\gamma}$$

- Riesz operator is a robust preconditioner!

Operator preconditioning (Mardal and Winther 2011; Loghin & Wathen 2004, Hiptmair 2006)

Problem: Given $\mathbf{f} \in \mathcal{H}'$, find $\mathbf{u} \in \mathcal{H}$ such that $\mathbf{Au} = \mathbf{f}$. Here, $\mathbf{A} : \mathcal{H} \mapsto \mathcal{H}'$ is an isomorphism.

Riesz operator: $\mathbf{B} : \mathcal{H}' \mapsto \mathcal{H}$, such that for every $\mathbf{f} \in \mathcal{H}'$,

$$\langle \mathbf{f}, \mathbf{x} \rangle = (\mathbf{Bf}, \mathbf{x})_{\mathcal{H}}, \qquad \forall\ \mathbf{x} \in \mathcal{H}$$

Estimate $\kappa(\mathbf{BA})$:

$$\|\mathbf{BA}\| = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y}\rangle|}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \le \beta$$

$$\|[\mathbf{BA}]^{-1}\|^{-1} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y}\rangle|}{\|\mathbf{x}\|_{\mathcal{H}}\|\mathbf{y}\|_{\mathcal{H}}} \ge \gamma$$

$$\Longrightarrow \kappa(\mathbf{BA}) = \|\mathbf{BA}\|\|[\mathbf{BA}]^{-1}\| \le \frac{\beta}{\gamma}$$

- Riesz operator is a robust preconditioner!

Operator preconditioning (Mardal and Winther 2011; Loghin & Wathen 2004, Hiptmair 2006)

Problem: Given $\mathbf{f} \in \mathcal{H}'$, find $\mathbf{u} \in \mathcal{H}$ such that $\mathbf{Au} = \mathbf{f}$. Here, $\mathbf{A} : \mathcal{H} \mapsto \mathcal{H}'$ is an isomorphism.

Riesz operator: $\mathbf{B} : \mathcal{H}' \mapsto \mathcal{H}$, such that for every $\mathbf{f} \in \mathcal{H}'$,

$$\langle \mathbf{f}, \mathbf{x} \rangle = (\mathbf{Bf}, \mathbf{x})_{\mathcal{H}}, \qquad \forall\, \mathbf{x} \in \mathcal{H}$$

Estimate $\kappa(\mathbf{BA})$:

$$\|\mathbf{BA}\| = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} = \sup_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y} \rangle|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} \leq \beta$$

$$\|[\mathbf{BA}]^{-1}\|^{-1} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|(\mathbf{BAx}, \mathbf{y})_{\mathcal{H}}|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} = \inf_{\mathbf{x}\in\mathcal{H}} \sup_{\mathbf{y}\in\mathcal{H}} \frac{|\langle \mathbf{Ax}, \mathbf{y} \rangle|}{\|\mathbf{x}\|_{\mathcal{H}} \|\mathbf{y}\|_{\mathcal{H}}} \geq \gamma$$

$$\Longrightarrow \kappa(\mathbf{BA}) = \|\mathbf{BA}\| \|[\mathbf{BA}]^{-1}\| \leq \frac{\beta}{\gamma}$$

- Riesz operator is a robust preconditioner!

Error indicators: Estimating the residual

- Existence of a preconditioner $\mathbf{B} \Longrightarrow$ two-sided estimate on $\|\mathbf{e}\|_{\mathbf{B}^{-1}} = \|\mathbf{e}\|_{\mathcal{H}}$. Let $r \in \mathcal{H}'$ be the residual $\mathbf{r} = \mathbf{f} - \mathbf{A}\mathbf{u}_h$.

Lemma

We have the following two sided bound

$$\|\mathbf{BA}\|_{\mathbf{B}^{-1}}^{-1}\|\mathbf{r}\|_{\mathbf{B}} \leq \|e\|_{\mathbf{B}^{-1}} \leq \|(\mathbf{BA})^{-1}\|_{\mathbf{B}^{-1}}\|\mathbf{r}\|_{\mathbf{B}}.$$

$$\|\mathbf{BA}\|_{\mathcal{H}}\|\mathbf{r}\|_{\mathcal{H}'} \leq \|e\|_{\mathcal{H}} \leq \|(\mathbf{BA})^{-1}\|_{\mathcal{H}}\|\mathbf{r}\|_{\mathcal{H}'}.$$

Proof.

Using the relation $\mathbf{e} = \mathbf{A}^{-1}\mathbf{r}$, we have

$$\|\mathbf{e}\|_{\mathbf{B}^{-1}} = \|\mathbf{e}\|_{\mathcal{H}} = \|\mathbf{A}^{-1}\mathbf{B}^{-1}\mathbf{B}\mathbf{r}\|_{\mathcal{H}} \leq \|(\mathbf{BA})^{-1}\|_{\mathcal{H}}\|\mathbf{B}\mathbf{r}\|_{\mathcal{H}}.$$

On the other hand: $\|\mathbf{r}\|_{B} = \|\mathbf{r}\|_{\mathcal{H}'} = \|\mathbf{BA}e\|_{\mathcal{H}} \leq \|\mathbf{BA}\|_{\mathcal{H}}\|e\|_{\mathcal{H}}$. □

Error indicators: Estimating the residual

- Existence of a preconditioner $\mathbf{B} \Longrightarrow$ two-sided estimate on $\|\mathbf{e}\|_{\mathbf{B}^{-1}} = \|\mathbf{e}\|_{\mathcal{H}}$. Let $r \in \mathcal{H}'$ be the residual $\mathbf{r} = \mathbf{f} - \mathbf{A}\mathbf{u}_h$.

Lemma

We have the following two sided bound

$$\|\mathbf{BA}\|_{\mathbf{B}^{-1}}^{-1}\|\mathbf{r}\|_{\mathbf{B}} \le \|e\|_{\mathbf{B}^{-1}} \le \|(\mathbf{BA})^{-1}\|_{\mathbf{B}^{-1}}\|\mathbf{r}\|_{\mathbf{B}}.$$

$$\|\mathbf{BA}\|_{\mathcal{H}}\|\mathbf{r}\|_{\mathcal{H}'} \le \|e\|_{\mathcal{H}} \le \|(\mathbf{BA})^{-1}\|_{\mathcal{H}}\|\mathbf{r}\|_{\mathcal{H}'}.$$

• Result: Efficient and reliable error indicator, provided that the norm of the residual $\|\mathbf{r}\|_{\mathcal{H}'}$ can be efficiently approximated by local operations.

A Posteriori Error Estimation I

- A point of view: try to rewrite the (a posteriori error estimator) as a (two-level) Schwarz preconditioner.
- Take the infinite dimensional V as the fine grid: $V_h \subset V$ instead of $V_H \subset V_h$ in the two-level method.
- The error $e = u - u_h \in V$ and residual $r = f - Au_h \in V'$ are related by the error equation

$$Ae = r.$$

- Let $\{\phi_i\}_{i=1}^n$ be the nodal basis of V_h. Take $\Omega_i := \text{supp}\phi_i$, and $V_i = H_0^1(\Omega_i)$, and $I_i : V_i \to V$ be inclusion, $Q_i = I_i'$, $A_i = Q_i A I_i$.

A Posteriori Error Estimation I

- A point of view: try to rewrite the (a posteriori error estimator) as a (two-level) Schwarz preconditioner.
- Take the infinite dimensional V as the fine grid: $V_h \subset V$ instead of $V_H \subset V_h$ in the two-level method.
- The error $e = u - u_h \in V$ and residual $r = f - Au_h \in V'$ are related by the error equation

$$Ae = r.$$

- Let $\{\phi_i\}_{i=1}^n$ be the nodal basis of V_h. Take $\Omega_i := \text{supp}\phi_i$, and $V_i = H_0^1(\Omega_i)$, and $I_i : V_i \to V$ be inclusion, $Q_i = I_i'$, $A_i = Q_i A I_i$.

A Posteriori Error Estimation I

- A point of view: try to rewrite the (a posteriori error estimator) as a (two-level) Schwarz preconditioner.
- Take the infinite dimensional V as the fine grid: $V_h \subset V$ instead of $V_H \subset V_h$ in the two-level method.
- The error $e = u - u_h \in V$ and residual $r = f - Au_h \in V'$ are related by the error equation
$$Ae = r.$$
- Let $\{\phi_i\}_{i=1}^n$ be the nodal basis of V_h. Take $\Omega_i := \text{supp}\phi_i$, and $V_i = H_0^1(\Omega_i)$, and $I_i : V_i \to V$ be inclusion, $Q_i = I_i'$, $A_i = Q_i A I_i$.

A Posteriori Error Estimation I

- A point of view: try to rewrite the (a posteriori error estimator) as a (two-level) Schwarz preconditioner.
- Take the infinite dimensional V as the fine grid: $V_h \subset V$ instead of $V_H \subset V_h$ in the two-level method.
- The error $e = u - u_h \in V$ and residual $r = f - Au_h \in V'$ are related by the error equation
$$Ae = r.$$
- Let $\{\phi_i\}_{i=1}^n$ be the nodal basis of V_h. Take $\Omega_i := \text{supp}\phi_i$, and $V_i = H_0^1(\Omega_i)$, and $I_i : V_i \to V$ be inclusion, $Q_i = I_i'$, $A_i = Q_i A I_i$.

A posteriori error estimation II

- $V = V_h + \sum_{i=1}^n V_i$ corresponds to

$$B = A_h^{-1} Q_h + \sum_{i=1}^{n} A_i^{-1} Q_i : V' \to V,$$

which is a preconditioner for A,

A^{-1} is spectrally equivalent to B
(follows from S. Nepomnyaschikh's fictitious space Lemma)

- B yields an error estimator

$$\|e\|_A^2 = \|A^{-1} r\|_A^2 = \langle r, A^{-1} r\rangle \approx \langle r, Br\rangle$$
$$= \langle r, A_h^{-1} Q_h r\rangle + \sum_{i=1}^{n} \langle r, A_i^{-1} Q_i r\rangle.$$

- In case of FEM by definition $Q_h r = 0$!

A posteriori error estimation II

- $V = V_h + \sum_{i=1}^n V_i$ corresponds to

$$B = A_h^{-1} Q_h + \sum_{i=1}^n A_i^{-1} Q_i : V' \to V,$$

which is a preconditioner for A,

A^{-1} is spectrally equivalent to B
(follows from S. Nepomnyaschikh's fictitious space Lemma)

- B yields an error estimator

$$\|e\|_A^2 = \|A^{-1} r\|_A^2 = \langle r, A^{-1} r\rangle \approx \langle r, Br\rangle$$
$$= \langle r, A_h^{-1} Q_h r\rangle + \sum_{i=1}^n \langle r, A_i^{-1} Q_i r\rangle.$$

- In case of FEM by definition $Q_h r = 0$!

A posteriori error estimation II

- $V = V_h + \sum_{i=1}^n V_i$ corresponds to

$$B = A_h^{-1} Q_h + \sum_{i=1}^{n} A_i^{-1} Q_i : V' \to V,$$

which is a preconditioner for A,

A^{-1} is spectrally equivalent to B

(follows from S. Nepomnyaschikh's fictitious space Lemma)

- B yields an error estimator

$$\|e\|_A^2 = \|A^{-1} r\|_A^2 = \langle r, A^{-1} r \rangle \approx \langle r, Br \rangle$$
$$= \langle r, A_h^{-1} Q_h r \rangle + \sum_{i=1}^{n} \langle r, A_i^{-1} Q_i r \rangle.$$

- In case of FEM by definition $Q_h r = 0$!

Residual Error Estimator

- The error estimator is

$$\|e\|_A^2 \simeq \sum_{i=1}^{n} \langle r, A_i^{-1} Q_i r\rangle = \sum_{i=1}^{n} \|\eta_i\|_{A_i}^2,$$

  where $\eta_i \in V_i = H_0^1(\Omega_i)$ solves

$$(\nabla \eta_i, \nabla v_i) = (f, v_i) - (\nabla u_h, \nabla v_i), \quad \forall v_i \in V_i.$$

- It was first proposed in [Babuška&Rheinbolt(1978)SINUM].
- Go to computable quantities by standard arguments (so called Verfürth's bubble function approach): [book: Verfürth(2013)].

Residual Error Estimator

- The error estimator is

$$\|e\|_A^2 \simeq \sum_{i=1}^{n} \langle r, A_i^{-1} Q_i r \rangle = \sum_{i=1}^{n} \|\eta_i\|_{A_i}^2,$$

where $\eta_i \in V_i = H_0^1(\Omega_i)$ solves

$$(\nabla \eta_i, \nabla v_i) = (f, v_i) - (\nabla u_h, \nabla v_i), \quad \forall v_i \in V_i.$$

- Similarly: efficient and reliable error indicators (using Nodal Auxiliary Space Preconditioning) to discretizations of δd, Hodge Laplacian problems, and linear elasticity with weak symmetry.
- The only ingredients needed are: well-posedness of the problem and the existence of regular decomposition on continuous level (for singularly perturbed $H(d)$ problems). .

Li&Z.(2020)CAMWA, Li&Z. arXiv:2010.06774v1

A posteriori estimates in AMG for Graph Laplacians

- We consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $\mathcal{V} = \{1, \dots, n\}$, $n = |\mathcal{V}|$ and $n \gg 1$.
- Let $A \in \mathbb{R}^{n\times n}$ be defined via the bilinear form:

$$(Au, v) = \sum_{(i,j)\in\mathcal{E}} (-a_{ij})(u_i - u_j)(v_i - v_j).$$

  The sum runs over all edges $e = (i, j) \in \mathcal{E}$. The resulting matrix is known as the Graph Laplacian of $\mathcal{G}$.
- We are interested in good approximations of the above bilinear form on a smaller subspace (constructing multilevel hierarchies).
- Applications: Fast solution of $Au = f$ for a huge number of problems.

Gradients and divergence

- Define $G : V = \mathbb{R}^{|\mathcal{V}|} \mapsto \mathbb{R}^{|\mathcal{E}|} = \boldsymbol{W}$ and $D : \mathbb{R}^{|\mathcal{E}|} \mapsto \mathbb{R}^{|\mathcal{E}|}$ in the following way

$$(Gv)_e = v_{head} - v_{tail}, \quad D_{e,e} = a_e, \qquad a_e = -a_{ij}, \qquad e = (i,j).$$

- Thus we get another form of the bilinear form A: $(Au, v) = (DGu, Gv)$ (weighted graph Laplacian).
- Taking $D = I$ one obtains the standard graph Laplacian ($a_{ij} = -1$).

Applications

- Discretizations of PDEs (P1, DG, whatever discretizations of elliptic equations)
- Diffusion State distance
- Modeling small world networks (protein-protein interraction; social networks).
- Many other problems lead to systems spectrally equivalent to the graph Laplacians.

Adaptivity in solvers (AMG)

- Typical numerical models: $Au = f$, $A = -(\nabla \cdot \alpha \nabla)$ or $A \in \mathbb{R}^{n\times n}$.
- Such models do not have to use FEM or even to correspond to discretizations of PDEs.
- A look at the "adaptive" linear solvers (adaptive AMG, bootstrap AMG, adaptive SA, etc) reveals:
  - What is available in the literature is adaptive but with respect to A;
  - These methods do not involve any estimates of the error during iterations.
- Q: Are there ways to extend, at least partially, what is done in FE, FV, FD for a posteriori error analysis and adaptivity to areas such as approximation of data sets?
- Q: Can we use such estimates to create multilevel hierarchical representation of complicated data-sets (graphs).

Tools for solution: two level and multilevel methodology

- Algorithms for construction of multilevel hierarchical approximations of functions defined on graphs.
- By multilevel hierarchies here, we mean splitting of both edges and vertices in a way that gives: coarser graphs; corresponding Laplacians; operators that transfer data between the graphs.
- Goal: The solution on a coarser graph has to be close to the solution on a finer graph.

αAMG for graph Laplacians

- Adaptive AMG methods: aim at optimizing (wrt convergence) the choice of coarse spaces and multilevel hierarchies in an AMG algorithm.
- The majority of known to date adaptive AMG methods approximate the optimal coarse space and do not use all the information available such as right hand side.
- The basic ideas on adaptive AMG are outlined in the early works on classical AMG from the 80s (Brandt, McCormick and Ruge'82).
- Some adaptive multilevel methods:
  - Adaptive filtering (Wittum'92; Wittum&Wagner 1997); Adaptive ML-ILU (Bank& Smith'02);

αAMG for graph Laplacians

- Adaptive AMG methods: aim at optimizing (wrt convergence) the choice of coarse spaces and multilevel hierarchies in an AMG algorithm.
- The majority of known to date adaptive AMG methods approximate the optimal coarse space and do not use all the information available such as right hand side.
- The basic ideas on adaptive AMG are outlined in the early works on classical AMG from the 80s (Brandt, McCormick and Ruge'82).
- Some adaptive multilevel methods:
    - αAMG and αSA (Brezina,Falgout,MacLachlan,Manteuffel,McCormick,Ruge, 2004, 2006);

αAMG for graph Laplacians

- Adaptive AMG methods: aim at optimizing (wrt convergence) the choice of coarse spaces and multilevel hierarchies in an AMG algorithm.
- The majority of known to date adaptive AMG methods approximate the optimal coarse space and do not use all the information available such as right hand side.
- The basic ideas on adaptive AMG are outlined in the early works on classical AMG from the 80s (Brandt, McCormick and Ruge'82).
- Some adaptive multilevel methods:
  - Bootstrap AMG (Brandt'02, Brandt, Brannick, Livshits, Kahl, 2011,2015)

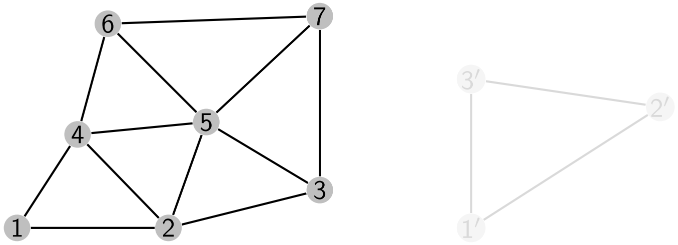
αAMG for graph Laplacians

- Adaptive AMG methods: aim at optimizing (wrt convergence) the choice of coarse spaces and multilevel hierarchies in an AMG algorithm.
- The majority of known to date adaptive AMG methods approximate the optimal coarse space and do not use all the information available such as right hand side.
- The basic ideas on adaptive AMG are outlined in the early works on classical AMG from the 80s (Brandt, McCormick and Ruge'82).
- Some adaptive multilevel methods:
  - Adaptive matching (Vassilevski & D'Ambra '2016)

αAMG for graph Laplacians

- Adaptive AMG methods: aim at optimizing (wrt convergence) the choice of coarse spaces and multilevel hierarchies in an AMG algorithm.
- The majority of known to date adaptive AMG methods approximate the optimal coarse space and do not use all the information available such as right hand side.
- The basic ideas on adaptive AMG are outlined in the early works on classical AMG from the 80s (Brandt, McCormick and Ruge'82).
- **This talk:**
    - Adaptive path covering (Hu, Lin, Z. 2019).
    - αAMG with Helmholtz decomposition (Hu, Wu, Z. 2022).

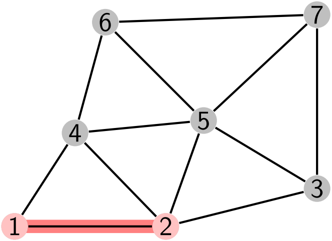
Coarse spaces: Recursive matching algorithm



The matching algorithm works as follows.

1. Choose the a vertex of smaller degree and group it with one of its unmatched neighbors (if such neighbor exists).
2. Repeat this until there are no unmatched neighbors.
3. Then group each isolated vertex with a neighbor with which it has the most connections: coarse graph

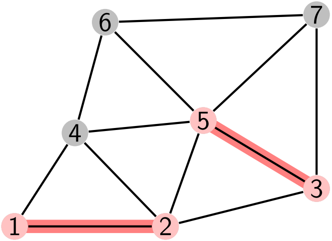
Coarse spaces: Recursive matching algorithm



The matching algorithm works as follows.

1. Choose the a vertex of smaller degree and group it with one of its unmatched neighbors (if such neighbor exists).
2. Repeat this until there are no unmatched neighbors.
3. Then group each isolated vertex with a neighbor with which it has the most connections: coarse graph

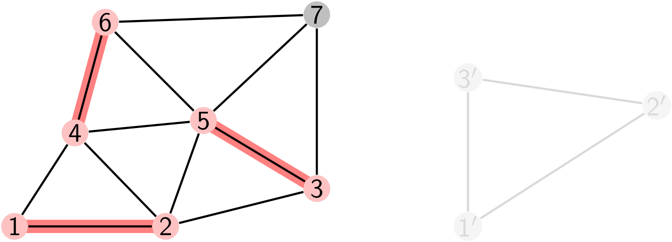
Coarse spaces: Recursive matching algorithm



The matching algorithm works as follows.

1. Choose the a vertex of smaller degree and group it with one of its unmatched neighbors (if such neighbor exists).
2. Repeat this until there are no unmatched neighbors.
3. Then group each isolated vertex with a neighbor with which it has the most connections: coarse graph

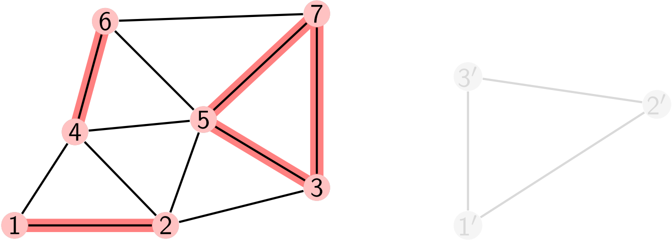
Coarse spaces: Recursive matching algorithm



The matching algorithm works as follows.

1. Choose the a vertex of smaller degree and group it with one of its unmatched neighbors (if such neighbor exists).
2. Repeat this until there are no unmatched neighbors.
3. Then group each isolated vertex with a neighbor with which it has the most connections: coarse graph

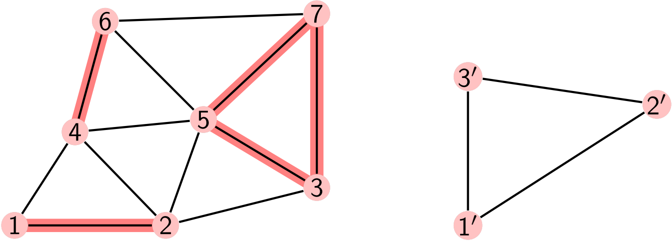
Coarse spaces: Recursive matching algorithm



The matching algorithm works as follows.

1. Choose the a vertex of smaller degree and group it with one of its unmatched neighbors (if such neighbor exists).
2. Repeat this until there are no unmatched neighbors.
3. Then group each isolated vertex with a neighbor with which it has the most connections: coarse graph

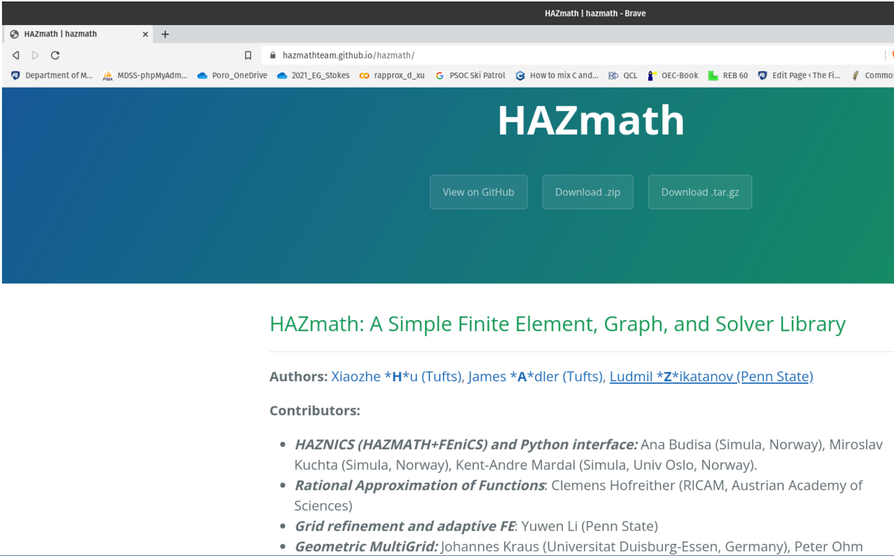
Coarse spaces: Recursive matching algorithm



The matching algorithm works as follows.

1. Choose the a vertex of smaller degree and group it with one of its unmatched neighbors (if such neighbor exists).
2. Repeat this until there are no unmatched neighbors.
3. Then group each isolated vertex with a neighbor with which it has the most connections: coarse graph

UA-AMG in action (HAZmath)



HAZmath | hazmath - Brave

HAZmath | hazmath

hazmathteam.github.io/hazmath/

Department of M... MDSS-phpMyAdm... Poro_OneDrive 2021_EG_Stokes rapprox_d_xu PSOC Ski Patrol How to mix C and... QCL OEC-Book REB 60 Edit Page ‹ The Fi... Commo

HAZmath

View on GitHub Download .zip Download .tar.gz

HAZmath: A Simple Finite Element, Graph, and Solver Library

Authors: Xiaozhe *H*u (Tufts), James *A*dler (Tufts), Ludmil *Z*ikatanov (Penn State)

Contributors:

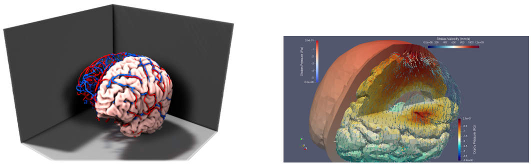
- ***HAZNICS (HAZMATH+FEniCS) and Python interface:*** Ana Budisa (Simula, Norway), Miroslav Kuchta (Simula, Norway), Kent-Andre Mardal (Simula, Univ Oslo, Norway).
- ***Rational Approximation of Functions***: Clemens Hofreither (RICAM, Austrian Academy of Sciences)
- ***Grid refinement and adaptive FE***: Yuwen Li (Penn State)
- ***Geometric MultiGrid:*** Johannes Kraus (Universitat Duisburg-Essen, Germany), Peter Ohm

UA-AMG in action

```
ltz1@ziggy:~/sync_psu_gdrive/Talks/2022_GATIPOR

%%%CPUtime(assembly) = 3.294 sec

 Calling UA AMG ...
-----------------------------------------------------------
  Level   Num of rows   Num of nonzeros   Avg. NNZ / row
-----------------------------------------------------------
    0         46549          1625725          34.93
    1          2547           147119          57.76
    2            45             1263          28.07
-----------------------------------------------------------
  Grid complexity = 1.056  |  Operator complexity = 1.091
-----------------------------------------------------------
Unsmoothed aggregation setup costs 0.0224 seconds.
**********************************************************
 --> using Conjugate Gradient Method:
-----------------------------------------------------------
It Num |   ||r||/||b||   |     ||r||      |  Conv. Factor
-----------------------------------------------------------
     0 |  1.000000e+00   |  9.561499e-03  |     -.--
     1 |  4.190978e-02   |  4.007203e-04  |     0.0419
     2 |  1.340687e-02   |  1.281898e-04  |     0.3199
     3 |  1.635897e-03   |  1.564163e-05  |     0.1220
     4 |  9.954893e-04   |  9.518370e-06  |     0.6085
     5 |  3.216388e-04   |  3.075349e-06  |     0.3231
     6 |  1.333691e-04   |  1.275208e-06  |     0.4147
     7 |  5.501142e-05   |  5.259917e-07  |     0.4125
     8 |  1.660677e-05   |  1.587856e-07  |     0.3019
     9 |  5.885296e-06   |  5.627226e-08  |     0.3544
    10 |  2.421312e-06   |  2.315138e-08  |     0.4114
    11 |  9.302008e-07   |  8.894114e-09  |     0.3842
    12 |  3.107932e-07   |  2.971649e-09  |     0.3341
    13 |  1.349787e-07   |  1.290599e-09  |     0.4343
    14 |  5.361918e-08   |  5.126797e-10  |     0.3972
    15 |  1.740890e-08   |  1.664552e-10  |     0.3247
    16 |  5.417964e-09   |  5.180386e-11  |     0.3112
Num_iter(krylov.c) = 16 with relative residual 5.417965e-09.
Iterative method costs 0.1155 seconds.
**********************************************************
AMG_Krylov method totally costs 0.1409 seconds.
**********************************************************

%% **** NO PLOT: Dimension=5 is too large for plotting
```

UA-AMG for a Laplacian on the unit cube in 5D after 14 bisection refinements

Preconditioning Darcy-Stokes (Rational approximations+UA-AMG)



(a) (b)

Figure: Left: shows the network (graph) of vessels and the porous tissue. Right: Computational result on coupled flow characteristics in the brain done with `HAZniCS` (dated Sep 10, 2021).

- Darcy-Stokes equations with these boundary conditions are used to model CSF-brain interaction.
- The action of the Riesz operator: requires computing the action of fractional Laplacian ($s > 0, t > 0$, $\mathcal{D} := (-\Delta)$):

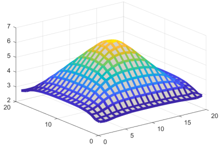
Making UA-AMG adaptive:approximation of level sets

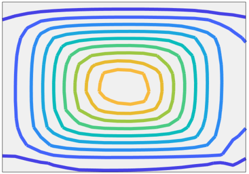
- Given an (approximation to the) error $\boldsymbol{e}$;
- Define auxiliary graph with same set of edges and with weights based on the computed approximation of the error $\boldsymbol{e}$.

$$w_{ij} = \frac{1}{|e_i - e_j|}, \quad (i,j) \in \mathcal{E}; \quad \mathcal{G} = (\mathcal{V}, \mathcal{E}).$$

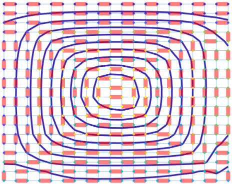
- Form a max weight path cover for this graph.
- By construction the paths follow the level sets of $\boldsymbol{e}$, i.e. $\boldsymbol{e}\Big|_p \approx \text{const}$ for any path p from the covering.

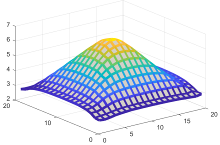
Path cover of level sets: illustration





• Left: smooth error; Right: Path cover following the level sets of the error;



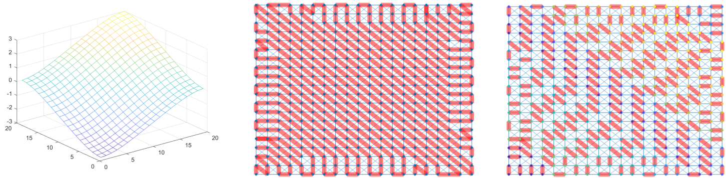


• Left: matchings following the level sets; Right: coarse space approximation. Error of approximation $\approx 10^{-10}$!

What if...

Level sets are not aligned with the grid

- We augment the set of adds of $\mathcal{G}(A)$ by adding edges from $\mathcal{G}(\boldsymbol{A}^2)$ (correponding to paths in $\mathcal{G}(A)$ of length 2).



• Left: smooth error; Middle: path cover; Right: matching/aggregates on this path cover.

How do we know the level sets of the error?

- Indeed, $\boldsymbol{e}$ is readily known only when $\boldsymbol{b} = 0$: not practical.

Practical adaptive algorithm

- With the current approximation $\boldsymbol{x}_k$, run several (couple of) W cycles on $A\boldsymbol{e} = \boldsymbol{f} - A\boldsymbol{x}_k$ to obtain an approximation of the error.
- Build hierarchy following the level sets of $\boldsymbol{e}$.
- Perform AMG iterations until the convergence slows down and go to the first step; or go to the first step every iteration.

• This algorithm looks expensive but it is also aimed to solve hard problems (not just at Laplace equation on uniform grid)

Numerical experiments (Real World Graphs)

Table: Largest connected components of the networks from the University of Florida sparse matrix collection (UF)

	$n \times 10^{-6}$	$\text{nnz} \times 10^{-6}$	Description
333SP	3.7	22.0	2-dimensional FE triangular meshes
belgium_osm	1.4	3.0	Belgium street network
M6	3.5	2.1	2-dimensional FE triangular meshes
NACA0015	1.0	6.2	2-dimensional FE triangular meshes
netherlands_osm	2.2	4.9	Netherlands street network
packing 500x100x100-b050	2.1	35.0	DIMACS Implementation Challenge
roadNet-CA	1.9	5.5	California road network
roadNet-PA	1.1	3.1	Philadelphia road network
roadNet-TX	1.3	3.7	Texas road network
fl2010	0.5	2.8	Florida census 2010
as-Skitter	1.6	22.0	Autonomous systems by Skitter
hollywood-2009	1.0	113.0	Hollywood movie actor network

Numerical experiments(continued)

Table: Largest connected components of the networks from Stanford large network datasets collection

	n	nnz	Description
com-DBLP	3.17080e5	2.41681e6	DBLP collaboration network
web-NotreDame	3.25729e5	1.09011e6	Web graph of Notre Dame
amazon0601	4.03364e5	5.28999e6	Amazon product co-purchasing network

Numerical experiments(continued)

Table: UF Collection (Low-Frequency ***b***)

	UA-AMG w/MWM			Algorithm A		Algorithm B		
	Iter	ConvR	OC	Iter	OC	Iter	Re	OC
UF large network datasets collection								
333SP	–	0.997	1.89	9	2.01	14	7	2.08
belgium_osm	1629	0.996	1.99	11	2.02	15	9	2.02
M6	–	0.997	1.86	10	2.11	15	8	2.11
NACA0015	–	0.995	1.86	9	2.10	14	7	2.10
netherlands_osm	–	0.997	1.98	10	2.02	16	9	2.02
packing	–	0.999	1.06	11	2.46	19	10	2.46
roadNet-CA	878	0.991	2.05	8	2.08	14	7	2.08
roadNet-PA	1382	0.991	2.05	8	2.10	14	7	2.09
roadNet-TX	1424	0.994	2.04	9	2.08	14	7	2.08
fl2010	–	0.998	1.83	9	2.19	15	7	2.19
as-Skitter	–	0.998	1.21	10	3.13	19	8	3.14
hollywood-2009	–	0.999	1.01	5	3.17	11	3	3.18

Numerical experiments (continued)

Table: Stanford Collection (Low-Frequency $\boldsymbol{b}$)

	UA-AMG w/MWM			Algorithm A		Algorithm B		
	Iter	ConvR	OC	Iter	OC	Iter	Re	OC
Stanford large network datasets collection								
com-DBLP	297	0.986	2.01	4	3.22	11	2	3.22
web-NotreDame	–	0.999	1.26	7	2.43	13	6	2.40
amazon0601	–	0.998	1.58	5	3.49	12	4	3.52

Numerical experiments(continued)

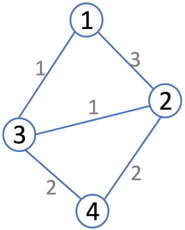
Table: UF collection: Zero-Sum Random $\boldsymbol{b}$, tol=1e-6

	UA-AMG w/MWM			Algorithm A		Algorithm B		
	Iter	ConvR	OC	Iter	OC	Iter	Re	OC
UF large network datasets collection								
333SP	–	0.997	1.89	9	2.09	6	1	2.08
belgium_osm	–	0.996	1.99	11	2.02	15	9	2.02
M6	–	0.997	1.86	8	2.11	5	1	2.11
NACA0015	1565	0.995	1.86	8	2.10	5	1	2.10
netherlands_osm	–	0.997	1.98	12	2.02	17	11	2.02
packing	–	0.999	1.06	11	2.46	17	10	2.47
roadNet-CA	1308	0.994	2.08	8	2.08	15	7	2.08
roadNet-PA	970	0.991	2.05	8	2.09	14	6	2.08
roadNet-TX	1168	0.992	2.04	9	2.08	14	7	2.08
fl2010	–	0.998	1.83	8	2.19	16	7	2.19
as-Skitter	–	0.998	1.21	10	3.04	17	7	3.06
hollywood-2009	–	0.999	1.01	7	3.17	13	5	3.18

Numerical experiments (continued)

	UA-AMG w/MWM			Algorithm A		Algorithm B		
	Iter	ConvR	OC	Iter	OC	Iter	Re	OC
Stanford collection Zero sum $\boldsymbol{b}$								
com-DBLP	573	0.987	2.01	4	3.23	11	3	3.22
web-NotreDame	–	0.999	1.26	7	2.47	15	6	2.56
amazon0601	–	0.998	1.58	6	3.49	10	4	3.50

Graph Operators



- Discrete gradient operator $G : \mathbb{R}^n \to \mathbb{R}^m$:

$$(G\boldsymbol{v})_e = \boldsymbol{v}_i - \boldsymbol{v}_j, \quad \forall\ \boldsymbol{v} \in \mathbb{R}^n.$$

- Edge weight matrix $D : \mathbb{R}^m \to \mathbb{R}^m$,

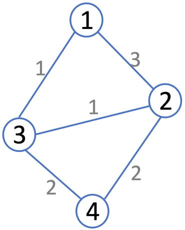
$$(D\boldsymbol{\tau})_e = w_e \boldsymbol{\tau}_e, \quad \forall\ \boldsymbol{\tau} \in \mathbb{R}^m.$$

- Weighted graph Laplacian $L := G^T DG$.

$$G = \begin{pmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \quad L = \begin{pmatrix} 4 & -3 & -1 & 0 \\ -3 & 6 & -1 & -2 \\ -1 & -1 & 4 & -2 \\ 0 & -2 & -2 & 4 \end{pmatrix}$$

$$D = \mathrm{diag}(3, 1, 1, 2, 2)$$

Graph Operators



- Discrete gradient operator $G : \mathbb{R}^n \to \mathbb{R}^m$:

$$(G\boldsymbol{v})_e = \boldsymbol{v}_i - \boldsymbol{v}_j, \quad \forall \ \boldsymbol{v} \in \mathbb{R}^n.$$

- Edge weight matrix $D : \mathbb{R}^m \to \mathbb{R}^m$,

$$(D\boldsymbol{\tau})_e = w_e \boldsymbol{\tau}_e, \quad \forall \ \boldsymbol{\tau} \in \mathbb{R}^m.$$

- Weighted graph Laplacian $L := G^T D G$.

$$G = \begin{pmatrix} -1 & 1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \quad L = \begin{pmatrix} 4 & -3 & -1 & 0 \\ -3 & 6 & -1 & -2 \\ -1 & -1 & 4 & -2 \\ 0 & -2 & -2 & 4 \end{pmatrix}$$

$$D = \operatorname{diag}(3, 1, 1, 2, 2)$$

Error Estimator (quick intro)

For the graph Laplacian problem , $L\boldsymbol{u} = \boldsymbol{f}$.

Lemma (Prager-Synge and S. Repin)

Fix $\boldsymbol{v} \in \mathbb{R}^n$, *for any* $\boldsymbol{\tau} \in \mathbb{R}^m$,

$$\|\boldsymbol{u} - \boldsymbol{v}\|_L \leq \|DG\boldsymbol{v} - \boldsymbol{\tau}\|_{D^{-1}} + C_p^{-1}\|G^T\boldsymbol{\tau} - \boldsymbol{f}\|. \quad (1)$$

C_p *is the Poincaré's constant of* L*.*

Remarks:

- RHS of this inequality provides a reliable upper bound of the error.
- This error estimate is expensive to compute.

W. Xu, Z., J. Comput. Appl. Math. (2018)

Error Estimator (quick intro)

For the graph Laplacian problem , $L\boldsymbol{u} = \boldsymbol{f}$.

Lemma (Prager-Synge and S. Repin)

Fix $\boldsymbol{v} \in \mathbb{R}^n$, *for any* $\boldsymbol{\tau} \in \mathbb{R}^m$,

$$\|\boldsymbol{u} - \boldsymbol{v}\|_L \leq \|DG\boldsymbol{v} - \boldsymbol{\tau}\|_{D^{-1}} + C_p^{-1}\|G^T\boldsymbol{\tau} - \boldsymbol{f}\|. \tag{1}$$

C_p *is the Poincaré's constant of* L.

Remarks:

- RHS of this inequality provides a reliable upper bound of the error.
- This error estimate is expensive to compute.

W. Xu, Z., J. Comput. Appl. Math. (2018)

A Posteriori Error Estimates

Denote $\mathscr{W}(\boldsymbol{f}) = \{\boldsymbol{\tau} \in \mathbb{R}^m | G^T\boldsymbol{\tau} = \boldsymbol{f}\}$.

Theorem (Exact error)

Let $\boldsymbol{u}$ be the solution to $L\boldsymbol{x} = \boldsymbol{f}$. Then for any $\boldsymbol{v} \in \mathbb{R}^n$,

$$\|\boldsymbol{u} - \boldsymbol{v}\|_L = \min_{\boldsymbol{\tau} \in \mathscr{W}(\boldsymbol{f})} \|DG\boldsymbol{v} - \boldsymbol{\tau}\|_{D^{-1}}.$$

Remark: If $\boldsymbol{v}$ is the approximate solution to $L\boldsymbol{x} = \boldsymbol{f}$, $\|DG\boldsymbol{v} - \boldsymbol{\tau}\|_{D^{-1}}$ is always an upper bound of the error $\boldsymbol{u} - \boldsymbol{v}$ for any $\boldsymbol{\tau} \in \mathscr{W}(\boldsymbol{f})$.

K. Wu, X. Hu, & Z., arXiv (2021)

Minimize $\psi(\boldsymbol{\tau})$

Goal: solve for $\boldsymbol{\tau} \in \mathscr{W}(\boldsymbol{f})$ by minimizing $\psi(\boldsymbol{\tau}) := \|DG\boldsymbol{v} - \boldsymbol{\tau}\|_{D^{-1}}$, with reasonable computational cost.

Helmholtz decomposition:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_0,$$

$\boldsymbol{\tau}_f \in \mathscr{W}(\boldsymbol{f})$: curl free.
$\boldsymbol{\tau}_0 \in \mathscr{W}(\mathbf{0})$: divergence free ($G^T\boldsymbol{\tau}_0 = \mathbf{0}$).

Minimize $\psi(\boldsymbol{\tau})$

Goal: solve for $\boldsymbol{\tau} \in \mathscr{W}(\boldsymbol{f})$ by minimizing $\psi(\boldsymbol{\tau}) := \|DG\boldsymbol{v} - \boldsymbol{\tau}\|_{D^{-1}}$, with reasonable computational cost.

Helmholtz decomposition:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_0,$$

$\boldsymbol{\tau}_f \in \mathscr{W}(\boldsymbol{f})$: curl free.
$\boldsymbol{\tau}_0 \in \mathscr{W}(\boldsymbol{0})$: divergence free ($G^T\boldsymbol{\tau}_0 = \boldsymbol{0}$).

Minimize $\psi(\boldsymbol{\tau})$

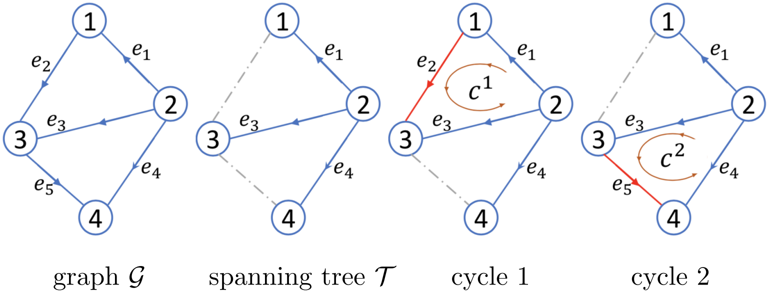
Goal: solve for $\boldsymbol{\tau}$ to minimize $\psi(\boldsymbol{\tau}) = \|DG\boldsymbol{v} - \boldsymbol{\tau}\|_{D^{-1}}$, with reasonable computational cost.

Helmholtz decomposition:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_0,$$

$\boldsymbol{\tau}_f \in \mathscr{W}(\boldsymbol{f})$: curl free. A gradient corresponding to a spanning tree of $\mathcal{G}$.
$\boldsymbol{\tau}_0 \in \mathscr{W}(\mathbf{0})$: divergence free. An element of the cycle space.

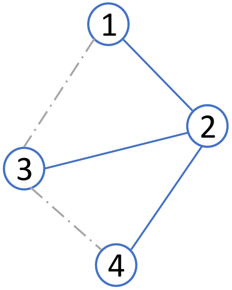
Spanning Tree and Cycle Space



Fundamental cycle basis:

$$\boldsymbol{c}^1 = [1, 1, -1, 0, 0]^T, \quad \boldsymbol{c}^2 = [0, 0, 1, -1, 1]^T.$$

Step 1: Compute $\boldsymbol{\tau}_f$ on the Spanning Tree



$\boldsymbol{\tau_f}$ is nonzero on the spanning tree

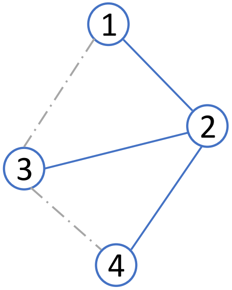
Goal: Solve $G^T \boldsymbol{\tau}_f = \boldsymbol{f}$ such that $(\boldsymbol{\tau_f})_e = 0$ for $e \in \mathcal{E} \setminus \mathcal{E}_T$.

- $\boldsymbol{f} = G^T \boldsymbol{\tau}_f = \begin{pmatrix} G_{\mathcal{T}}^T & G_{\mathcal{G}\setminus\mathcal{T}}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\tau}_{f\mathcal{T}} \\ \mathbf{0} \end{pmatrix} = G_{\mathcal{T}}^T \boldsymbol{\tau}_{f\mathcal{T}}$.
- to solve $G_{\mathcal{T}}^T \boldsymbol{\tau}_{f\mathcal{T}} = \boldsymbol{f}$,

  Key Idea: make use of $L_{\mathcal{T}} = G_{\mathcal{T}}^T D_{\mathcal{T}} G_{\mathcal{T}}$ and solve a linear system on $\mathcal{T}$ instead.

  - solve $L_{\mathcal{T}} \boldsymbol{x} = \boldsymbol{f}$
  - equivalent to solving: $G_{\mathcal{T}}^T D_{\mathcal{T}} G_{\mathcal{T}} x = \boldsymbol{f}$.

Step 1: Compute $\boldsymbol{\tau_f}$ on the Spanning Tree



$\boldsymbol{\tau_f}$ is nonzero on the spanning tree

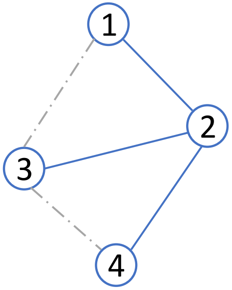
Goal: Solve $G^T\boldsymbol{\tau}_f = \boldsymbol{f}$ such that $(\boldsymbol{\tau_f})_e = 0$ for $e \in \mathcal{E} \setminus \mathcal{E}_T$.

- $\boldsymbol{f} = G^T\boldsymbol{\tau}_f = \begin{pmatrix} G_{\mathcal{T}}^T & G_{\mathcal{G}\setminus\mathcal{T}}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\tau}_{f\mathcal{T}} \\ \boldsymbol{0} \end{pmatrix} = G_{\mathcal{T}}^T \boldsymbol{\tau}_{f\mathcal{T}}$.
- to solve $G_{\mathcal{T}}^T\boldsymbol{\tau}_{f\mathcal{T}} = \boldsymbol{f}$,

  Key Idea: make use of $L_{\mathcal{T}} = G_{\mathcal{T}}^T D_{\mathcal{T}} G_{\mathcal{T}}$ and solve a linear system on $\mathcal{T}$ instead.

  - solve $L_{\mathcal{T}}\boldsymbol{x} = \boldsymbol{f}$
  - equivalent to solving: $G_{\mathcal{T}}^T D_{\mathcal{T}} G_{\mathcal{T}} x = \boldsymbol{f}$.

Step 1: Compute $\boldsymbol{\tau_f}$ on the Spanning Tree



$\boldsymbol{\tau_f}$ is nonzero on the spanning tree

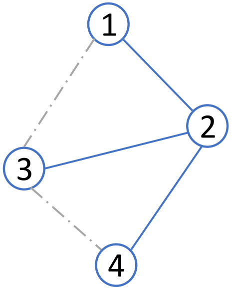
Goal: Solve $G^T \boldsymbol{\tau}_f = \boldsymbol{f}$ such that $(\boldsymbol{\tau_f})_e = 0$ for $e \in \mathcal{E} \setminus \mathcal{E}_{\mathcal{T}}$.

- $\boldsymbol{f} = G^T \boldsymbol{\tau}_f = \begin{pmatrix} G_{\mathcal{T}}^T & G_{\mathcal{G}\setminus\mathcal{T}}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\tau}_{f\mathcal{T}} \\ \mathbf{0} \end{pmatrix} = G_{\mathcal{T}}^T \boldsymbol{\tau}_{f\mathcal{T}}$.
- to solve $G_{\mathcal{T}}^T \boldsymbol{\tau}_{f\mathcal{T}} = \boldsymbol{f}$,

  Key Idea: make use of $L_{\mathcal{T}} = G_{\mathcal{T}}^T D_{\mathcal{T}} G_{\mathcal{T}}$ and solve a linear system on $\mathcal{T}$ instead.

  - solve $L_{\mathcal{T}} \boldsymbol{x} = \boldsymbol{f}$
  - equivalent to solving: $G_{\mathcal{T}}^T D_{\mathcal{T}} G_{\mathcal{T}} x = \boldsymbol{f}$.

Step 1: Compute $\boldsymbol{\tau}_f$ on the Spanning Tree



$\boldsymbol{\tau_f}$ is nonzero on the spanning tree

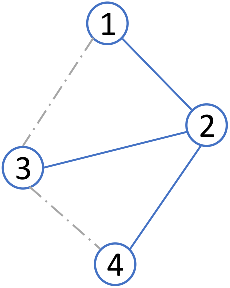
Goal: Solve $G^T\boldsymbol{\tau}_f = \boldsymbol{f}$ such that $(\boldsymbol{\tau_f})_e = 0$ for $e \in \mathcal{E} \setminus \mathcal{E}_\mathcal{T}$.

- $\boldsymbol{f} = G^T\boldsymbol{\tau}_f = \begin{pmatrix} G_\mathcal{T}^T & G_{\mathcal{G}\setminus\mathcal{T}}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\tau}_{f\mathcal{T}} \\ \mathbf{0} \end{pmatrix} = G_\mathcal{T}^T \boldsymbol{\tau}_{f\mathcal{T}}$.
- to solve $G_\mathcal{T}^T \boldsymbol{\tau}_{f\mathcal{T}} = \boldsymbol{f}$,

  Key Idea: make use of $L_\mathcal{T} = G_\mathcal{T}^T D_\mathcal{T} G_\mathcal{T}$ and solve a linear system on $\mathcal{T}$ instead.

  - solve $L_\mathcal{T}\boldsymbol{x} = \boldsymbol{f}$
  - equivalent to solving: $G_\mathcal{T}^T D_\mathcal{T} G_\mathcal{T} x = \boldsymbol{f}$.

Step 1: Compute $\boldsymbol{\tau}_f$ on the Spanning Tree



$\boldsymbol{\tau_f}$ is nonzero on the spanning tree

Goal: Solve $G^T \boldsymbol{\tau}_f = \boldsymbol{f}$ such that $(\boldsymbol{\tau_f})_e = 0$ for $e \in \mathcal{E} \setminus \mathcal{E}_T$.

- $\boldsymbol{f} = G^T \boldsymbol{\tau}_f = \begin{pmatrix} G_{\mathcal{T}}^T & G_{\mathcal{G}\setminus\mathcal{T}}^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\tau}_{f\mathcal{T}} \\ \mathbf{0} \end{pmatrix} = G_{\mathcal{T}}^T \boldsymbol{\tau}_{f\mathcal{T}}$.
- to solve $G_{\mathcal{T}}^T \boldsymbol{\tau}_{f\mathcal{T}} = \boldsymbol{f}$,

  Key Idea: make use of $L_{\mathcal{T}} = G_{\mathcal{T}}^T D_{\mathcal{T}} G_{\mathcal{T}}$ and solve a linear system on $\mathcal{T}$ instead.

  - solve $L_{\mathcal{T}} \boldsymbol{x} = \boldsymbol{f}$
  - equivalent to solving: $G_{\mathcal{T}}^T \underbrace{D_{\mathcal{T}} G_{\mathcal{T}} x}_{\boldsymbol{\tau}_{f\mathcal{T}}} = \boldsymbol{f}$.

D. Rose et al. SIAM J. Comput.(1976)

Step 2: Compute $\boldsymbol{\tau}_0$ in Cycle Space $\mathscr{C}$

Problem recap: solve $\min_{\boldsymbol{\tau}\in\mathscr{W}(\boldsymbol{f})} \|DG\boldsymbol{v} - \boldsymbol{\tau}\|_{D^{-1}}$, where $\boldsymbol{\tau} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_0$.

Constrained Minimization

For a given $\boldsymbol{\tau}_f$, we need to solve (approximately):

$$\min_{\boldsymbol{\tau}_0\in\mathscr{C}} \|DG\boldsymbol{v} - \boldsymbol{\tau}_f - \boldsymbol{\tau}_0\|_{D^{-1}}. \tag{2}$$

Schwarz Methods:

Decompose the cycle space $\mathscr{C}$ into subspaces:

$$\mathscr{C} = \mathscr{C}_1 + \mathscr{C}_2 + \cdots + \mathscr{C}_J.$$

Solve in each subspace $\mathscr{C}_i$, $i = 1, 2, \cdots, J$:

$$\min_{\Delta\boldsymbol{\tau}\in\mathscr{C}_{i+1}} \|DG\boldsymbol{v} - \boldsymbol{\tau}_f - (\boldsymbol{\tau}_0^i + \Delta\boldsymbol{\tau})\|_{D^{-1}}. \tag{3}$$

Step 2: Compute $\boldsymbol{\tau}_0$ in Cycle Space $\mathscr{C}$

Problem recap: solve $\min_{\boldsymbol{\tau}\in\mathscr{W}(\boldsymbol{f})} \|DG\boldsymbol{v} - \boldsymbol{\tau}\|_{D^{-1}}$, where $\boldsymbol{\tau} = \boldsymbol{\tau}_f + \boldsymbol{\tau}_0$.

Constrained Minimization

For a given $\boldsymbol{\tau}_f$, we need to solve (approximately):

$$\min_{\boldsymbol{\tau}_0\in\mathscr{C}} \|DG\boldsymbol{v} - \boldsymbol{\tau}_f - \boldsymbol{\tau}_0\|_{D^{-1}}. \tag{2}$$

Schwarz Methods:

Decompose the cycle space $\mathscr{C}$ into subspaces:

$$\mathscr{C} = \mathscr{C}_1 + \mathscr{C}_2 + \cdots + \mathscr{C}_J.$$

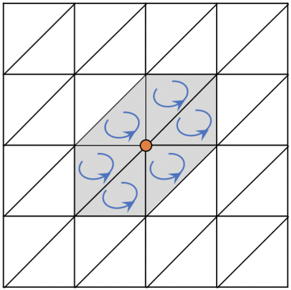
Solve in each subspace $\mathscr{C}_i$, $i = 1, 2, \cdots, J$:

$$\min_{\Delta\boldsymbol{\tau}\in\mathscr{C}_{i+1}} \|DG\boldsymbol{v} - \boldsymbol{\tau}_f - (\boldsymbol{\tau}_0^i + \Delta\boldsymbol{\tau})\|_{D^{-1}}. \tag{3}$$

Step 2: Schwarz Methods to Compute τ_0 in $\mathscr{C}$

Domain decomposition:

$\mathscr{C}_i = \operatorname{span}\{\boldsymbol{c}^j |$ cycle j contains vertex $i\}$,
$i = 1, \ldots, J$.



Cost of Schwarz method depends on:

- number of subspaces J: $\mathcal{O}(n)$.
- cost of solving (3) in each subspace: $\mathcal{O}(1)$.

Total cost of one iteration of Schwarz method: $\mathcal{O}(n)$.

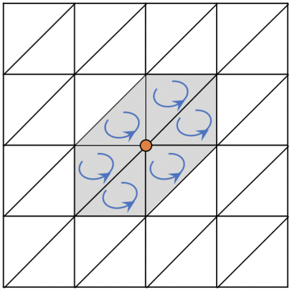
Remark: Worst case runtime: $\mathcal{O}(n \log n)$.

Kelner et al, STOC(2013)

Step 2: Schwarz Methods to Compute τ_0 in $\mathscr{C}$

Domain decomposition:

$\mathscr{C}_i = \mathrm{span}\{\boldsymbol{c}^j |$ cycle j contains vertex $i\}$,
$i = 1, \ldots, J$.



Cost of Schwarz method depends on:

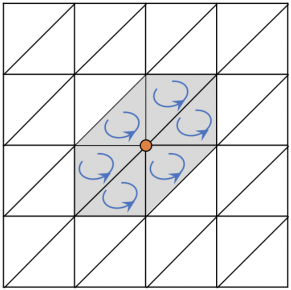
- number of subspaces J: $\mathcal{O}(n)$.
- cost of solving (3) in each subspace: $\mathcal{O}(1)$.

Total cost of one iteration of Schwarz method: $\mathcal{O}(n)$.

Remark: Worst case runtime: $\mathcal{O}(n \ \log n)$.

Kelner et al, STOC(2013)

Step 2: Schwarz Methods to Compute τ_0 in $\mathscr{C}$



Domain decomposition:

$\mathscr{C}_i = \text{span}\{\boldsymbol{c}^j \,|\, \text{cycle } j \text{ contains vertex } i\}$,
$i = 1, \ldots, J$.

Cost of Schwarz method depends on:

- number of subspaces J: $\mathcal{O}(n)$.
- cost of solving (3) in each subspace: $\mathcal{O}(1)$.

Total cost of one iteration of Schwarz method: $\mathcal{O}(n)$.

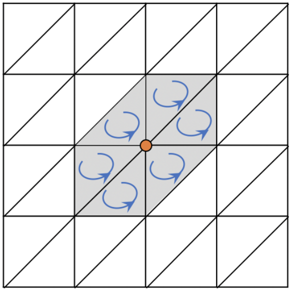
Remark: Worst case runtime: $\mathcal{O}(n \log n)$.

Kelner et al, STOC(2013)

Step 2: Schwarz Methods to Compute τ_0 in $\mathscr{C}$

Domain decomposition:

$\mathscr{C}_i = \text{span}\{\boldsymbol{c}^j |$ cycle j contains vertex $i\}$,
$i = 1, \ldots, J$.



Cost of Schwarz method depends on:

- number of subspaces J: $\mathcal{O}(n)$.
- cost of solving (3) in each subspace: $\mathcal{O}(1)$.

Total cost of one iteration of Schwarz method: $\mathcal{O}(n)$.

Remark: Worst case runtime: $\mathcal{O}(n \log n)$.

Kelner et al, STOC(2013)

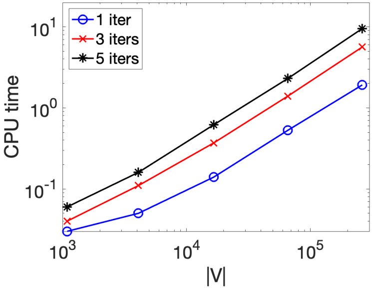
Results: Scalability

Parameters and Notation:

- Graph: 2D uniform triangular grids (corresponding to 2D Poisson equation on square domain with Neumann B.C.)
- Grid size: $h = 2^{-\ell}$, $\ell = 5, 6, 7, 8, 9$.
- Cycle type: face cycle.
- Efficiency coefficient: $e_{ff} := \frac{\psi(\boldsymbol{\tau})}{\|\boldsymbol{u}-\boldsymbol{v}\|_L}$.
- CPU time: in seconds.

Results: Scalability

		1 iter		3 iters		5 iters	
$\lvert\mathcal{V}\rvert$	$\lVert \boldsymbol{u} - \boldsymbol{v} \rVert_L$	$\psi(\boldsymbol{\tau})$	e_{ff}	$\psi(\boldsymbol{\tau})$	e_{ff}	$\psi(\boldsymbol{\tau})$	e_{ff}
1089	1.73	2.25	1.30	1.99	1.15	1.91	1.10
4097	1.73	2.67	1.55	2.28	1.32	2.16	1.25
16641	1.73	3.36	1.95	2.76	1.60	2.56	1.48
66049	1.72	4.43	2.57	3.51	2.03	3.20	1.86
263169	1.72	6.01	3.49	4.66	2.71	4.19	2.43



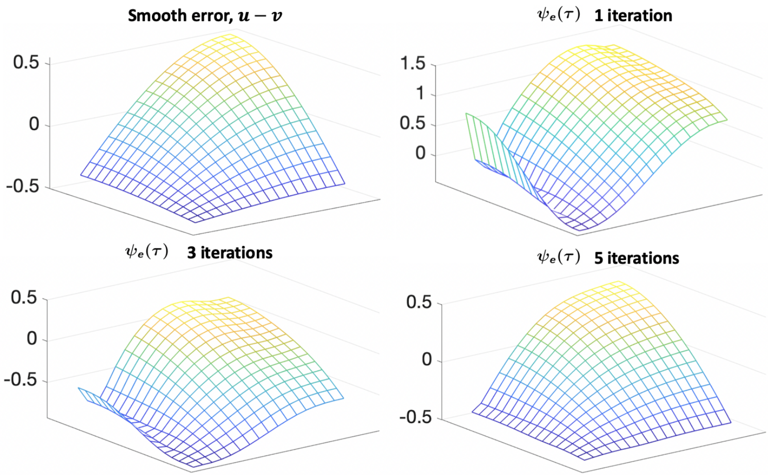
Results: Real World Graphs

$\|\mathcal{V}\|$	$\|\mathcal{E}\|$	Problem Type	$\|\boldsymbol{u}-\boldsymbol{v}\|_L$	$\psi(\boldsymbol{\tau})$	e_{ff}
292	958	Least Squares Problem	1.74	1.75	1.00
1879	5525	Circuit Simulation	2.71	2.71	1.00
5300	8271	Power Network	5.82	5.82	1.00
2048	4034	Electronagnetics Problem	0.47	0.50	1.07
1423	16342	Structural Problem	14.5	19.7	1.36
8205	58681	Accoustic Problem	23.8	37.7	1.58
1857	13762	Social Network	52.9	76.3	1.44
2361	13828	Protein Network	4.61	4.70	1.01

T. Davis and Y. Hu,The Univ. of Florida Sparse Matrix Collection

Results: Local Error Estimates

Localized error estimates: $\psi_e(\boldsymbol{\tau}) = \boldsymbol{\omega}_e^{-\frac{1}{2}} |(DG\boldsymbol{v} - \boldsymbol{\tau})_e|$.



Application: αAMG Coarsening

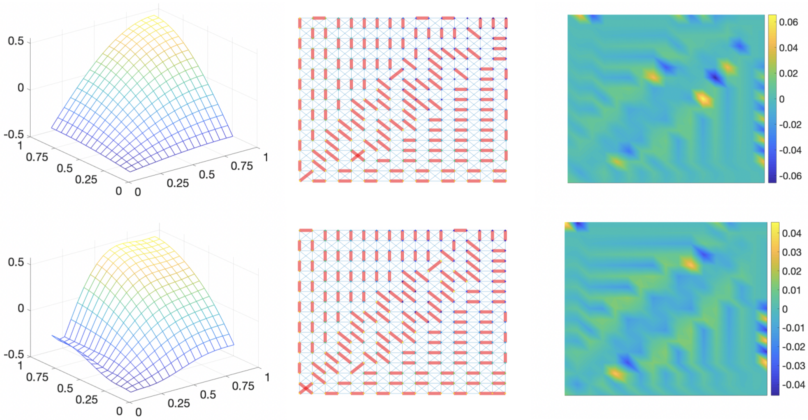
Idea: use approximate (smooth) error to build adaptive AMG.

Path Cover adaptive AMG (PC-αAMG):

- Approximate the smooth error with a posteriori error estimates.
- Find level sets of the smooth error by path cover.
- Aggregate along the level sets.
- Define AMG hierarchy using the aggregates and smooth error.

J. Lin, X. Hu, and L. Z. SISC(2019); Hu, Wu, Z. 2022 (SISC)

Application: αAMG coarsening



upper row: aggregation with smooth error.
lower row: aggregation with error estimator.

Summary

- Operator preconditioning: provides a path for constructing error indicators, right?
- A posteriori techniques can aid Adaptive AMG coarsening.
  - Approximate the smooth error using a posteriori estimator.
  - Adaptive path cover algorithm (coarsening following the level sets of an approximation of the error)
- Such techniques currently finding their way into the `HAZniCS` library `https://hazmathteam.github.io/hazmath/`

Thank you

Thank You!