Graph partitioning with matrix coefficients for symmetric positive definite linear systems

Eugene Vecharynski[1]
Yousef Saad[1]
Masha Sosonkina[2]

[1]University of Minnesota
Department of Computer Science and Engineering

[2]U.S. Department of Energy Ames Laboratory
Iowa State University

Twelfth Copper Mountain Conference on Iterative Methods

March 28, 2012

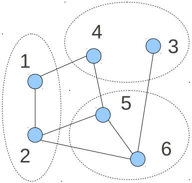
Acknowledgments

- U.S. Department of Energy
  - Grant DE-FG-08ER25841
  - Contract DE-AC02-05CH11231
- Iowa State University
  - Contract DE-AC02-07CH11358
- Minnesota Supercomputing Institute

Graph partitioning

Partition a graph $G = (V, E)$ into s subgraphs $G_k = (V_k, E_k)$, $V_k \subseteq V$, $E_k \subseteq E$, where

$$\bigcup_{k=1,s} V_k = V, \qquad \bigcap_{k=1,s} V_k = \emptyset.$$



- Balanced partitioning: $|V_k| \approx n/s$.
- Aims at satisfying an objective (e.g., minimize the edge cut).
- NP-complete $\Rightarrow$ heuristics/approximation algorithms.
- A lot of applications: VLSI circuit design, data clustering, image segmentation, **parallel scientific computations**.

Parallel linear solvers

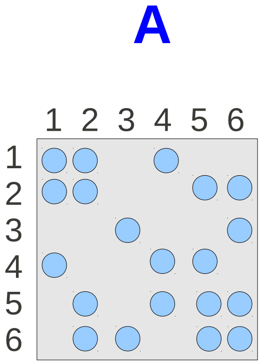
Consider solving a linear system

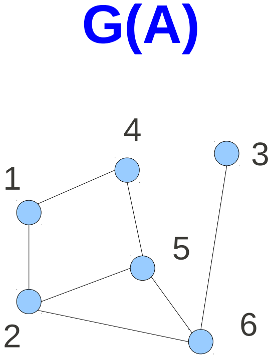
$$Ax = b$$

in parallel (distributed memory architecture).

- A is very large and sparse.
- Use a Krylov subspace linear solver (CG, MINRES, GMRES, ...).
- The “core” computation is the matrix-vector multiplication (matvec) $y = A * v$.
- Extract as much parallelism from matvec as possible (i.e., minimize communications).
- Partitioning of the linear system is the first step of the solution.

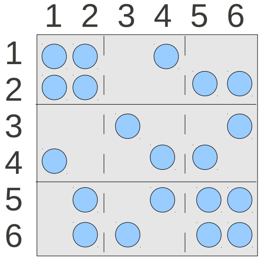
Partitioning of a linear system as a graph problem



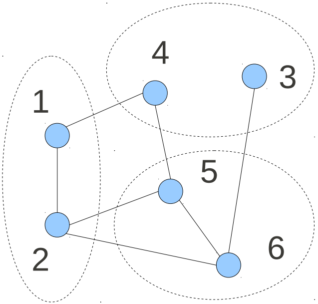


Partitioning of a linear system as a graph problem

A

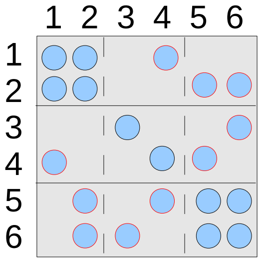


G(A)

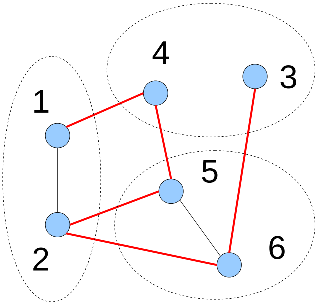


Partitioning of a linear system as a graph problem

A

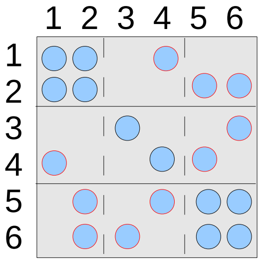


G(A)

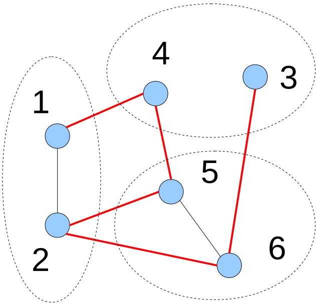


Partitioning of a linear system as a graph problem

A



G(A)



Problem: Partition $G(A)$ into subgraphs $G_k = (V_k, E_k)$, such that the edge cut is minimized and $|V_i| \approx |V_j|$.

Motivation

- The traditional goal of partitioning is to reduce communications for $y = Av$.
- Available graph partitioning software packages (Chaco, Metis, Scotch, ...).

Motivation

- The traditional goal of partitioning is to reduce communications for $y = Av$.
- Available graph partitioning software packages (Chaco, Metis, Scotch, ...).

BUT

- Performance of iterative solvers heavily depends on the **preconditioner**.
- Parallel preconditioners use partitions which minimize communications in matvecs. Partitions which maximize the quality of preconditioning may be different!
- Possible trade-off: communications vs iteration count.

Motivation

- The traditional goal of partitioning is to reduce communications for $y = Av$.
- Available graph partitioning software packages (Chaco, Metis, Scotch, ...).

BUT

- Performance of iterative solvers heavily depends on the **preconditioner**.
- Parallel preconditioners use partitions which minimize communications in matvecs. Partitions which maximize the quality of preconditioning may be different!
- Possible trade-off: communications vs iteration count.

Our goal: remove the requirement on minimizing communications and consider partitionings which favor the quality of a preconditioner.

- Restrict to SPD systems and nonoverlapping additive Schwarz (block Jacobi) preconditioners.

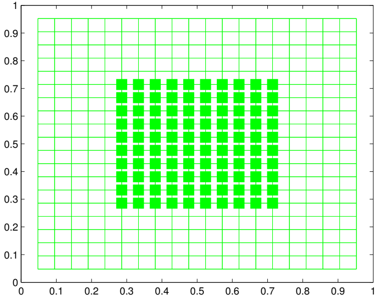
Quick example

The 2D diffusion equation

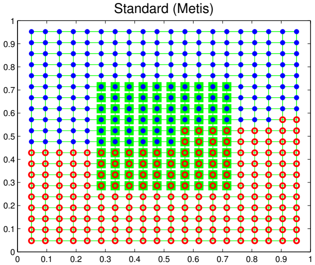
$$-\frac{\partial}{\partial x}\left(a(x,y)\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(b(x,y)\frac{\partial u}{\partial y}\right)=f(x,y)\ ,\ (x,y)\in[0,1]\times[0,1]$$

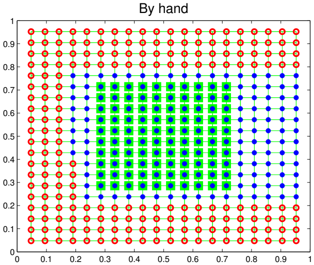
with zero Dirichlet boundary conditions, where

$$a(x,y)=b(x,y)=\begin{cases}100, & 0.25<x,y<0.75\\ 1, & \text{otherwise}\ .\end{cases}$$



Quick example. Additive Schwarz (AS) preconditioning



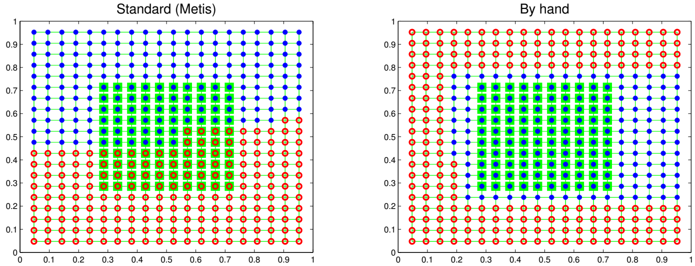


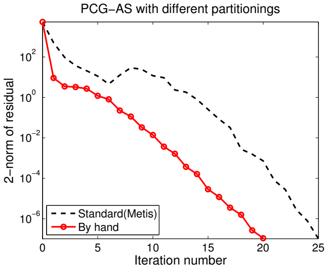
AS preconditioner

Input: A, r, $\{V_k\}_{k=1}^{s}$. Output: $w = T^{-1}r$.

1. For $k = 1, \ldots, s$, Do
2. Set $A_k := A(V_k, V_k)$, $r_k := r(V_k)$, and $w_k = \mathbf{0} \in \mathbb{R}^n$.
3. Solve $A_k\delta = r_k$.
4. Set $w_k(V_k) := \delta$.
5. EndDo
6. $w = w_1 + \ldots + w_s$.

Quick example. Additive Schwarz (AS) preconditioning





Optimal bipartitioning

Consider a **bipartition** $\{I, J\}$ of $V = \{1, 2, \ldots, n\}$,

$$V = I \cup J, \qquad I \cap J = \emptyset, \qquad |I| = |J| = n/2.$$

Let T be the AS preconditioner with respect to $\{I, J\}$.

Question: What $\{I, J\}$ leads to the best T (the fastest convergence)?

- For PCG an indicator of the preconditioner quality is $\kappa(T^{-1}A)$.
- $\Rightarrow$ Find $\{I, J\}$ which (approximately) minimizes $\kappa(T^{-1}A)$ over all bipartitions.

How?

Cauchy-Bunyakowski-Schwarz (CBS) constants

Reorder A and T with respect to $\{I, J\}$,

$$C = PAP^T = \begin{pmatrix} A_I & A_{IJ} \\ A_{IJ}^* & A_J \end{pmatrix}, \qquad B = PTP^T = \begin{pmatrix} A_I & 0 \\ 0 & A_J \end{pmatrix} .$$

- Note that $\kappa(T^{-1}A) = \kappa(B^{-1}C)$.
  $\Rightarrow$ Use available theory for block-diagonal preconditioning.
- Condition number bound based on the **CBS constant**:

$$\kappa(T^{-1}A)\ (= \kappa(B^{-1}C)) \ \leq \frac{1+\gamma_{IJ}}{1-\gamma_{IJ}}\ ,\ 0 \leq \gamma_{IJ} < 1\ ,$$

where

$$\gamma_{IJ} = \max_{u \in W_I, v \in W_J} \frac{|(u, Av)|}{(u, Au)^{1/2}(v, Av)^{1/2}}\ ;$$

$$W_I = \{u \in \mathbb{R}^n \ :\ u(J) = \mathbf{0}\}\ ,\ W_J = \{v \in \mathbb{R}^n \ :\ v(I) = \mathbf{0}\}\ .$$

Optimal bipartitioning

An **optimal bipartition** $\{I_{opt}, J_{opt}\}$ is such that

$$\gamma_{opt} = \min_{\substack{I,J \subset V=\{1,\dots,n\}, \\ |I|=|J|=\frac{n}{2}, J=V\setminus I}} \underbrace{\max_{u\in W_I, v\in W_J} \frac{|(u,Av)|}{(u,Au)^{1/2}(v,Av)^{1/2}}}_{\gamma_{IJ}},$$

Optimal bipartitioning

An **optimal bipartition** $\{I_{opt}, J_{opt}\}$ is such that

$$\gamma_{opt} = \min_{\substack{I,J \subset V = \{1,\ldots,n\}, \\ |I|=|J|=\frac{n}{2}, J=V \setminus I}} \underbrace{\max_{u \in W_I, v \in W_J} \frac{|(u, Av)|}{(u, Au)^{1/2}(v, Av)^{1/2}}}_{\gamma_{IJ}},$$

Our approach: Replace γ_{IJ} by a "simpler" $\tilde{\gamma}_{IJ}$.

Optimal bipartitioning

An **optimal bipartition** $\{I_{opt}, J_{opt}\}$ is such that

$$\gamma_{opt} = \min_{\substack{I,J\subset V=\{1,\ldots,n\}\ ,\\ |I|=|J|=\frac{n}{2},J=V\setminus I}} \underbrace{\max_{u\in W_I, v\in W_J} \frac{|(u, Av)|}{(u, Au)^{1/2}(v, Av)^{1/2}}}_{\gamma_{IJ}} ,$$

Our approach: Replace γ_{IJ} by a "simpler" $\tilde{\gamma}_{IJ}$.

- Consider a set

$$S = \{(e_i, e_j)\ :\ i \in I, j \in J, a_{ij} \neq 0\} \subset W_I \times W_J\ .$$

Optimal bipartitioning

An **optimal bipartition** $\{I_{opt}, J_{opt}\}$ is such that

$$\gamma_{opt} = \min_{\substack{I,J \subset V=\{1,\ldots,n\}, \\ |I|=|J|=\frac{n}{2}, J=V\setminus I}} \underbrace{\max_{u \in W_I, v \in W_J} \frac{|(u, Av)|}{(u, Au)^{1/2}(v, Av)^{1/2}}}_{\gamma_{IJ}},$$

Our approach: Replace γ_{IJ} by a "simpler" $\tilde{\gamma}_{IJ}$.

- Consider a set

$$S = \{(e_i, e_j) \ : \ i \in I, j \in J, a_{ij} \neq 0\} \subset W_I \times W_J \ .$$

- Evaluate $\frac{|(u,Av)|}{(u,Au)^{1/2}(v,Av)^{1/2}}$ at all pairs in S and find the mean,

$$\tilde{\gamma}_{IJ} = \frac{1}{|S|} \sum_{i \in I, j \in J} \frac{|a_{ij}|}{\sqrt{a_{ii} a_{jj}}} \ .$$

Optimal bipartitioning

An **optimal bipartition** $\{I_{opt}, J_{opt}\}$ is such that

$$\gamma_{opt} = \min_{\substack{I,J \subset V=\{1,\ldots,n\}, \\ |I|=|J|=\frac{n}{2}, J=V\setminus I}} \underbrace{\max_{u\in W_I, v\in W_J} \frac{|(u,Av)|}{(u,Au)^{1/2}(v,Av)^{1/2}}}_{\gamma_{IJ}},$$

Our approach: Replace γ_{IJ} by a "simpler" $\tilde{\gamma}_{IJ}$.

- Consider a set

$$S = \{(e_i, e_j) \; : \; i \in I, j \in J, a_{ij} \neq 0\} \subset W_I \times W_J \, .$$

- Evaluate $\frac{|(u,Av)|}{(u,Au)^{1/2}(v,Av)^{1/2}}$ at all pairs in S and find the mean,

$$\tilde{\gamma}_{IJ} = \frac{1}{|S|} \sum_{i\in I, j\in J} \underbrace{\frac{|a_{ij}|}{\sqrt{a_{ii}a_{jj}}}}_{w_{ij}} \, .$$

Optimal bipartitioning

An **optimal bipartition** $\{I_{opt}, J_{opt}\}$ is such that

$$\gamma_{opt} = \min_{\substack{I,J \subset V=\{1,\ldots,n\}, \\ |I|=|J|=\frac{n}{2}, J=V\setminus I}} \underbrace{\max_{u\in W_I, v\in W_J} \frac{|(u,Av)|}{(u,Au)^{1/2}(v,Av)^{1/2}}}_{\gamma_{IJ}},$$

Our approach: Replace γ_{IJ} by a "simpler" $\tilde{\gamma}_{IJ}$.

- Consider a set

$$S = \{(e_i, e_j) \ : \ i \in I, j \in J, a_{ij} \neq 0\} \subset W_I \times W_J \ .$$

- Evaluate $\frac{|(u,Av)|}{(u,Au)^{1/2}(v,Av)^{1/2}}$ at all pairs in S and find the mean,

$$\tilde{\gamma}_{IJ} = \underbrace{\frac{1}{|S|}}_{1/\mathrm{cut}(I,J)} \underbrace{\sum_{i\in I, j\in J} \underbrace{\frac{|a_{ij}|}{\sqrt{a_{ii}a_{jj}}}}_{w_{ij}}}_{w(I,J)}$$

Optimal bipartitioning

An **optimal bipartition** $\{I_{opt}, J_{opt}\}$ is such that

$$\gamma_{opt} = \min_{\substack{I,J \subset V=\{1,\ldots,n\}, \\ |I|=|J|=\frac{n}{2}, J=V\setminus I}} \underbrace{\max_{u\in W_I, v\in W_J} \frac{|(u,Av)|}{(u,Au)^{1/2}(v,Av)^{1/2}}}_{\gamma_{IJ}},$$

Our approach: Replace γ_{IJ} by a “simpler” $\tilde{\gamma}_{IJ}$.

- Consider a set

$$S = \{(e_i, e_j) \ : \ i \in I, j \in J, a_{ij} \neq 0\} \subset W_I \times W_J \ .$$

- Evaluate $\frac{|(u,Av)|}{(u,Au)^{1/2}(v,Av)^{1/2}}$ at all pairs in S and find the mean,

$$\tilde{\gamma}_{IJ} = \underbrace{\frac{1}{|S|}}_{1/\mathrm{cut}(I,J)} \underbrace{\sum_{i\in I, j\in J} \underbrace{\frac{|a_{ij}|}{\sqrt{a_{ii}a_{jj}}}}_{w_{ij}}}_{w(I,J)} = \frac{w(I,J)}{\mathrm{cut}(I,J)} \ .$$

Approximating optimal bipartitions

An **optimal bipartition** $\{I_{opt}, J_{opt}\}$ is such that

$$\gamma_{opt} = \min_{\substack{I,J \subset V = \{1,\dots,n\}, \\ |I| = |J| = \frac{n}{2}, J = V \setminus I}} \underbrace{\max_{u \in W_I, v \in W_J} \frac{|(u, Av)|}{(u, Au)^{1/2}(v, Av)^{1/2}}}_{\gamma_{IJ}},$$

Our approach: Replace γ_{IJ} by a "simpler" $\tilde{\gamma}_{IJ}$.

$$\tilde{\gamma}_{opt} = \min_{\substack{I,J \subset V = \{1,\dots,n\}, \\ |I| = |J| = \frac{n}{2}, J = V \setminus I}} \underbrace{\frac{w(I, J)}{\mathrm{cut}(I, J)}}_{\tilde{\gamma}_{IJ}}.$$

Approximating optimal bipartitions

An **optimal bipartition** $\{I_{opt}, J_{opt}\}$ is such that

$$\gamma_{opt} = \min_{\substack{I,J \subset V = \{1,\ldots,n\}, \\ |I|=|J|=\frac{n}{2}, J=V\setminus I}} \underbrace{\max_{u\in W_I, v\in W_J} \frac{|(u,Av)|}{(u,Au)^{1/2}(v,Av)^{1/2}}}_{\gamma_{IJ}},$$

Our approach: Replace γ_{IJ} by a "simpler" $\tilde{\gamma}_{IJ}$.

$$\tilde{\gamma}_{opt} = \min_{\substack{I,J \subset V = \{1,\ldots,n\}, \\ |I|=|J|=\frac{n}{2}, J=V\setminus I}} \underbrace{\frac{w(I,J)}{\mathrm{cut}(I,J)}}_{\tilde{\gamma}_{IJ}}.$$

$$\Updownarrow$$

$$\min_p \frac{p^T L_w p}{p^T L p}, \qquad p^T \mathbf{1} = 0, \qquad p(k) = \begin{cases} 1, & k \in I, \\ -1, & k \in J; \end{cases}$$

$\boldsymbol{L_w = D_w - W}$ and $\boldsymbol{L = D - Q}$ are the weighted and unweighted **graph Laplacians**, $\mathbf{1} = (1, 1, \ldots, 1)^T$.

Spectral graph bipartitioning for SPD systems

The combinatorial problem:

$$\min_p \frac{p^T L_w p}{p^T L p}, \qquad p^T \mathbf{1} = 0, \qquad p(k) = \pm 1 \ .$$

The "relaxed" problem:

$$\min_v \frac{v^T L_w v}{v^T L v}, \qquad v^T \mathbf{1} = 0, \qquad v \in \mathbb{R}^n \ .$$

Spectral graph bipartitioning for SPD systems

The combinatorial problem:

$$\min_p \frac{p^T L_w p}{p^T L p}, \qquad p^T \mathbf{1} = 0, \qquad p(k) = \pm 1 \ .$$

The “relaxed” problem:

$$\min_v \frac{v^T L_w v}{v^T L v}, \qquad v^T \mathbf{1} = 0, \qquad v \in \mathbb{R}^n \ .$$

$$\Updownarrow$$

Find the eigenvector associated with the smallest eigenvalue of

$$\boldsymbol{L_w v = \lambda L v}, \qquad \boldsymbol{v \in 1^{\perp}}.$$

Spectral graph bipartitioning for SPD systems

The combinatorial problem:

$$\min_p \frac{p^T L_w p}{p^T L p}, \qquad p^T \mathbf{1} = 0, \qquad p(k) = \pm 1\ .$$

The "relaxed" problem:

$$\min_v \frac{v^T L_w v}{v^T L v}, \qquad v^T \mathbf{1} = 0, \qquad v \in \mathbb{R}^n\ .$$

$$\Updownarrow$$

Find the eigenvector associated with the smallest eigenvalue of

$$\boldsymbol{L_w v = \lambda L v}, \qquad \boldsymbol{v \in 1^{\perp}}.$$

- $G(A)$ is connected $\Rightarrow$ null(L_w) = null(L) = span$\{\mathbf{1}\}$.
- SPD eigenproblem on $\mathbf{1}^{\perp}$.
- Sets I and J defined, e.g., as $I = \{i \ : \ v_i \geq 0\}$, $J = \{i \ : \ v_i < 0\}$.
- **Caution**: Sets I and J may correspond to **disconnected** subgrapghs.

Recursive bipartitioning

- $G(A)$ is connected

$$L_w v = \lambda L v, \qquad v \in \text{null}(L_w)^\perp = \text{null}(L)^\perp = 1^\perp.$$

Recursive bipartitioning

- $G(A)$ is connected

$$L_w v = \lambda L v, \qquad v \in \text{null}(L_w)^\perp = \text{null}(L)^\perp = 1^\perp.$$

- $G(A)$ is disconnected, i.e., has s connected components $G_i = (V_i, E_i)$

$$L_w v = \lambda L v, \qquad v \in \text{null}(L_w)^\perp = \text{null}(L)^\perp,$$

where null(L_w) = null(L) = span$\{z_1, \ldots, z_s\}$,

$$z_i(k) = \begin{cases} 1, & k \in V_i \ , \\ 0, & k \notin V_i \ . \end{cases}$$

Recursive bipartitioning

- $G(A)$ is connected

$$L_w v = \lambda L v, \qquad v \in \text{null}(L_w)^\perp = \text{null}(L)^\perp = 1^\perp.$$

- $G(A)$ is disconnected, i.e., has s connected components $G_i = (V_i, E_i)$

$$L_w v = \lambda L v, \qquad v \in \text{null}(L_w)^\perp = \text{null}(L)^\perp,$$

where null(L_w) = null(L) = span$\{z_1, \ldots, z_s\}$,

$$z_i(k) = \begin{cases} 1, & k \in V_i \ , \\ 0, & k \notin V_i \ . \end{cases}$$

Now ready state the recursive algorithm...

Recursive bipartitioning. The algorithm

CBSPartition($G(A)$, A)

Input: $G(A)$, A SPD. Output: Partition $\{V_i\}$ of $\{1, 2, \dots, n\}$.

1. Assign weights $w_{ij} = \frac{|a_{ij}|}{\sqrt{a_{ii} a_{jj}}}$.
2. Construct graph Laplacians $L_w = D_w - W$ and $L = D - Q$.
3. Find connected components $G_i \Leftrightarrow$ find null(L_w) = null(L).
4. Find the eigenpair corresponding to the smallest eigenvalue of
$$L_w v = \lambda L v, \qquad v \in \text{null}(L_w)^\perp = \text{null}(L)^\perp.$$
5. Define $\{I, J\}$ based on the computed eigenvector.
6. Form $G_I = (I, E_I)$ and $G_J = (J, E_J)$.
   If $|I| > maxSize$, apply CBSPartition(G_I,$A(I, I)$), **else** return I.
   If $|J| > maxSize$, apply CBSPartition(G_J,$A(J, J)$), **else** return J.

- If all w_{ij} are (nearly) the same in step 1, then construct $\{I, J\}$ by a “standard” method and go to step 6.
- We use LOBPCG (Knyazev ’01) with IC preconditioner (for L_w) for the eigensolve.

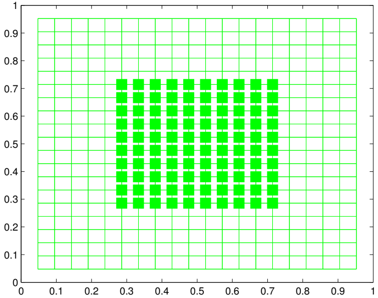
Our example

The 2D diffusion equation

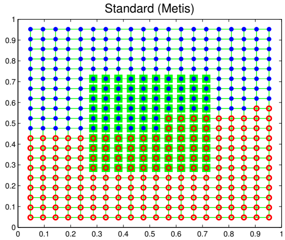
$$-\frac{\partial}{\partial x}\left(a(x,y)\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(b(x,y)\frac{\partial u}{\partial y}\right)=f(x,y)\ ,\ (x,y)\in[0,1]\times[0,1]$$

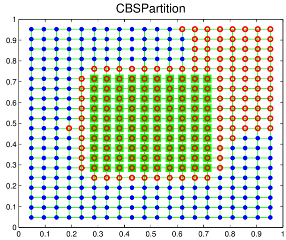
with zero Dirichlet boundary conditions, where

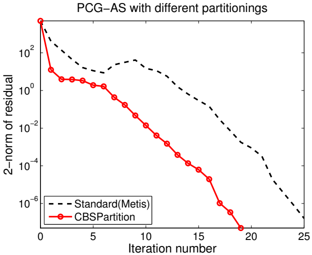
$$a(x,y)=b(x,y)=\begin{cases}100, & 0.25<x,y<0.75\\ 1, & \text{otherwise}\ .\end{cases}$$



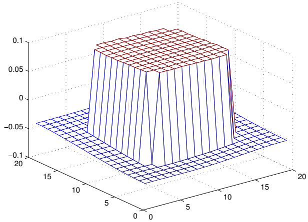
Example. Bipartitioning







The eigenvector:



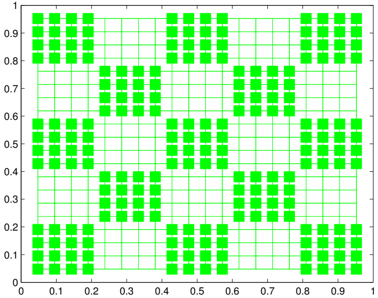
The “checkerboard” example

The 2D diffusion equation

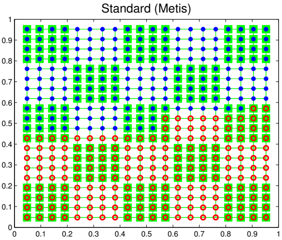
$$-\frac{\partial}{\partial x}\left(a(x,y)\frac{\partial u}{\partial x}\right)-\frac{\partial}{\partial y}\left(b(x,y)\frac{\partial u}{\partial y}\right)=f(x,y)\;,\;(x,y)\in[0,1]\times[0,1]$$

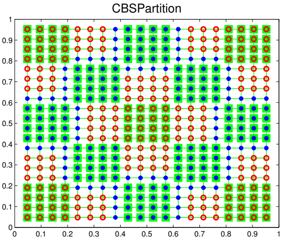
with zero Dirichlet boundary conditions, where

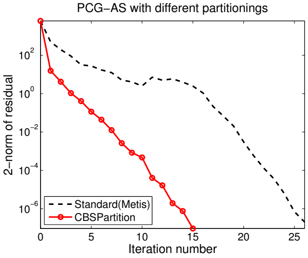
$$a(x,y)=b(x,y)=\begin{cases}100, & x,y \text{ in “black”}\\ 1, & \text{otherwise}\ .\end{cases}$$



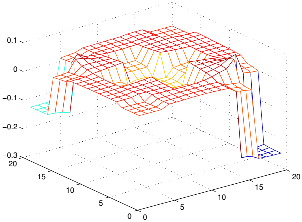
The “checkerboard” example. Bipartitioning



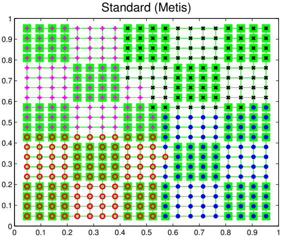


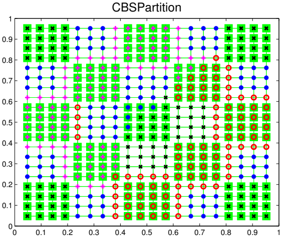


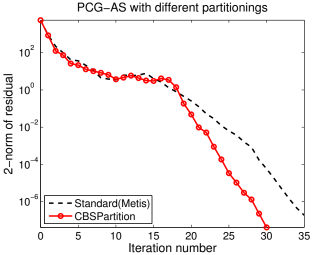
The eigenvector:



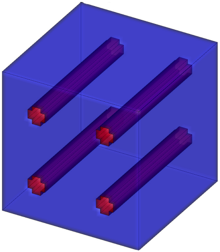
The “checkerboard” example. Four subdomains







Linear elasticity in 3D. Eight subdomains



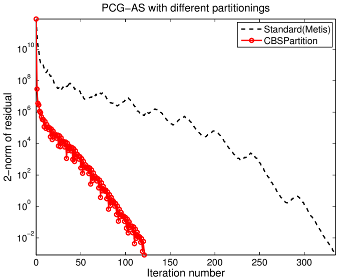


Figure: Example from *“Adaptive BDDC in Three Dimensions”* by Mandel, Sousedík, and Šístek (Math. Comp. Simul., in print, 2011). The figure and matrices are courtesy of the authors. Ndof = 107, 811.
The cube has material parameters $E = 10^6$ Pa and $\nu = 0.45$. The bars have parameters $E = 2.1\ 10^{11}$ Pa and $\nu = 0.3$.
CutMetis = 6.77% CutCBS = 14.78%

Concluding remarks

Remarks

- Partitioning for SPD systems and nonoverlapping AS.
- Partitions can be overlapped by “growing” for overlapping AS.
- The partitioning algorithm is purely algebraic, i.e., assumes no prior knowledge of discretization, problem origin, etc.

Current and future work

- More examples?
- Timings of parallel solves? Compromise with minimizing communications.
- Other applications for the new cut? (e.g., clustering)
- Partitioning schemes for nonsymmetric systems?
- Partitioning for other types of preconditioners?
- Alternative ways to use matrix coefficients? (e.g., automated weighting + “standard” partitioning objectives)

Question?

Thank you!