Preconditioning symmetric and highly indefinite problems

Matthias Bollhöfer

TU Braunschweig, Inst. *Computational Mathematics*

University of Minnesota, April 14, 2011

Technische Universität Braunschweig

Outline

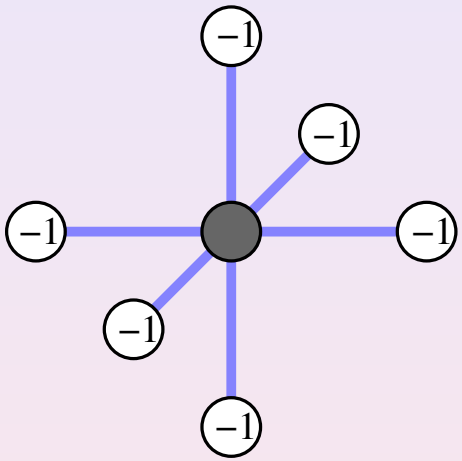
Technische Universität Braunschweig

Outline

The Anderson Model of Localization

Model describes electronic transport properties of disordered quantum systems

Discretization approach: periodic uniform grid in a 3D cube



- symmetric eigenvalue problem $Ax = \lambda x$ in 3D
- λ quantum mechanical energy levels
- x quantum mechanical wave functions

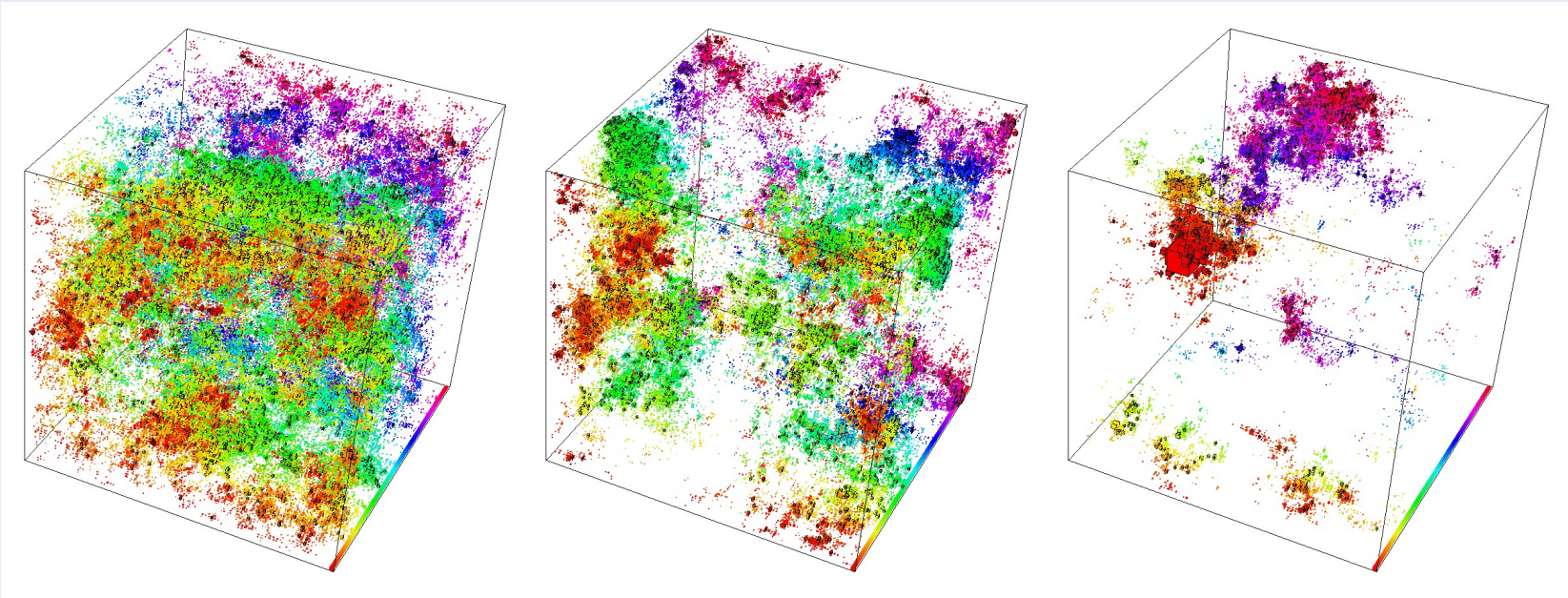
$$\varepsilon_{ijk} x_{ijk} - \sum_{|p-i|=1,|q-j|=1,|r-j|=1} x_{pqr} = \lambda x_{ijk}$$

$$\varepsilon_{ijk} \in [-\omega/2, \omega/2] \text{ i.i.d.}$$

- ω refers to the disorder strength
- $\omega \approx \omega_c = 16.5$ critical behaviour
- $\omega \ll \omega_c \rightarrow$ wave function is fluctuating from site to site
- $\omega \gg \omega_c \rightarrow$ wave function is localized

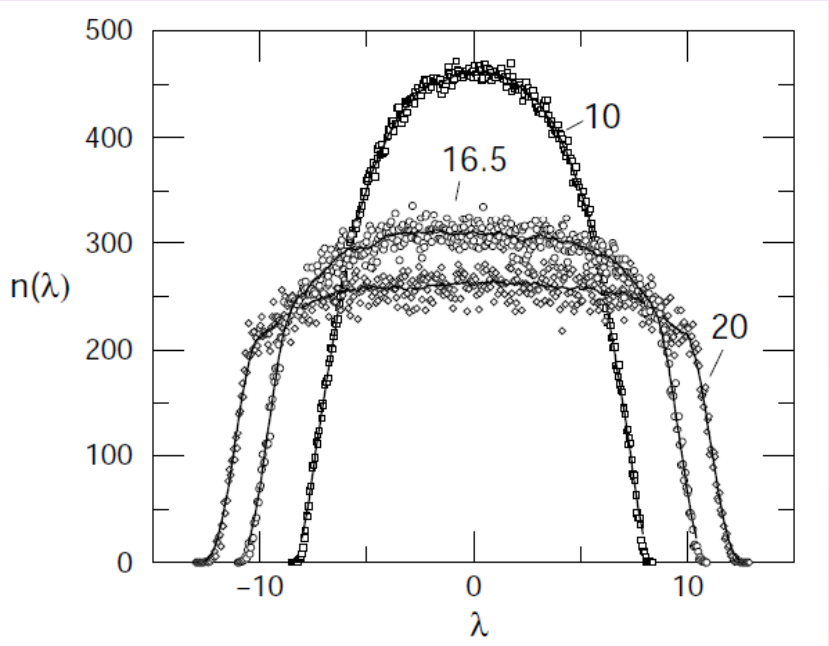
wave function distribution

$\omega = 14.5$ $\omega = 16.5$ $\omega = 18.5$



Challenges

- physically sensible resultats require large scale simulation, $n = m^3$ (e.g. $m = 100, 200, \ldots$)
- physically interesting: eigenvectors at $\lambda = 0$ where $\omega = \omega_c$
- solution approach based on preconditioned eigenvalue solver (Jacobi-Davidson)
- A is symmetric and highly indefinite



- a symmetry-preserving preconditioner for A is desirable

Helmholtz equations

$$-\Delta u - k^2 u = b \text{ in } \Omega$$

$$k = \frac{2\pi f}{c}, \quad f \text{ frequency}, c \text{ speed of sound}$$

Boundary conditions:

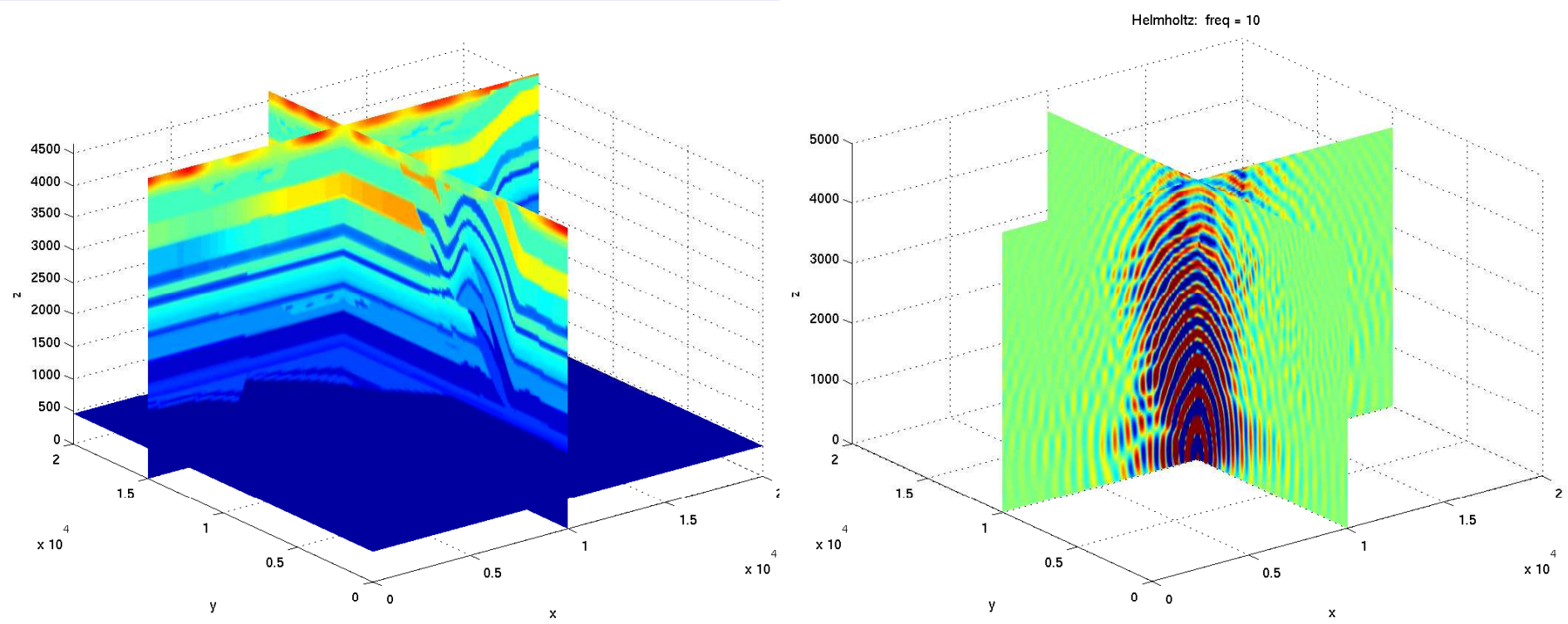
$$\underbrace{u = g_1 \text{ on } \Gamma_1}_{\text{Dirichlet}}, \quad \underbrace{\frac{\partial u}{\partial \nu} = g_2 \text{ on } \Gamma_2}_{\text{Neumann}}, \quad \underbrace{\frac{\partial u}{\partial \nu} + iku = 0 \text{ on } \Gamma_3}_{\text{Sommerfeld}}.$$

- Complex symmetric (Sommerfeld radiation b.c.!)
- Highly indefinite, discretization with mesh size h requires

$$kh \leqslant \frac{2\pi}{L}$$

to have L grid points per wave length.

3D Seismic Imaging Problem



- Helmholtz equation in a cube
- point source in the middle of the upper boundary
- Sommerfeld b.c.
- irregular velocity distribution

Time-harmonic Maxwell equations

First order formulation

$$\begin{cases} +\mathbf{i}\omega\varepsilon\mathbf{E} - \mathrm{curl}(\mathbf{H}) &= -\sigma\mathbf{E} - \boldsymbol{J}_{\boldsymbol{E}}, \\ -\mathbf{i}\omega\mu\mathbf{H} - \mathrm{curl}(\mathbf{E}) &= 0, \end{cases}$$

- boundary conditions on $\partial\Omega = \Gamma_a \cup \Gamma_m$
- perfect electric conductor condition $\boldsymbol{n} \times \boldsymbol{E} = 0$ on Γ_m
- Silver-Müller absorbing condition $\mathcal{L}(\boldsymbol{E}, \boldsymbol{H}) = \mathcal{L}(\boldsymbol{E}^{\mathrm{inc}}, \boldsymbol{H}^{\mathrm{inc}})$ on Γ_a

- discretization leads to complex-symmetric system $Ax = b$, where

$$A = \mathbf{i}\omega \begin{pmatrix} M_\varepsilon & 0 \\ 0 & -M_\mu \end{pmatrix} + \begin{pmatrix} C_{11} + M_\sigma & -C_{12} \\ -C_{12}^T & -C_{22} \end{pmatrix}$$

Outline

What are Matchings

Example

$$\begin{pmatrix} * & & * & & * \\ & & * & * & \\ * & * & & * & \\ & * & * & & \\ * & & & & \end{pmatrix}$$

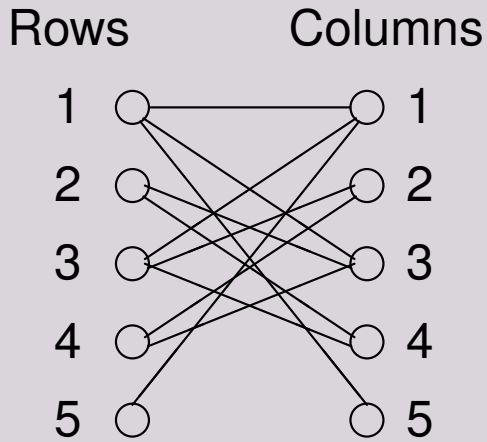
original A

What are Matchings

Example

$$\begin{pmatrix} * & & * & & * \\ & & * & * & \\ * & * & & * & \\ & * & * & & \\ * & & & & \end{pmatrix}$$

original A



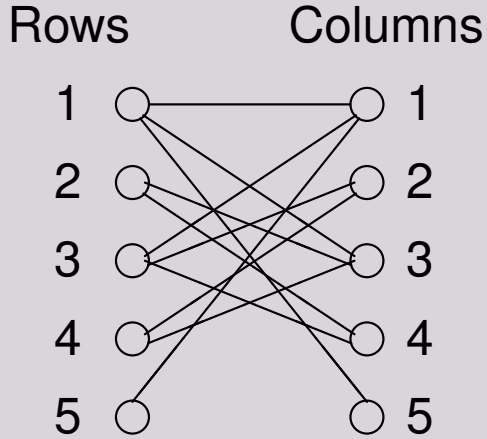
- Associated graph

What are Matchings

Example

$$\begin{pmatrix} * & & * & & * \\ & & * & * & \\ * & * & & * & \\ & * & * & & \\ * & & & & \end{pmatrix}$$

original A



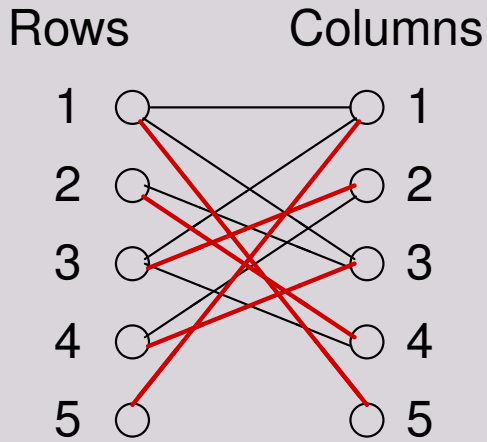
- Associated graph
- (Perfect) matching. (At most) one edge for each row/column

What are Matchings

Example

$$\begin{pmatrix} * & & * & & * \\ & & * & * & \\ * & * & & * & \\ & * & * & & \\ * & & & & \end{pmatrix}$$

original A



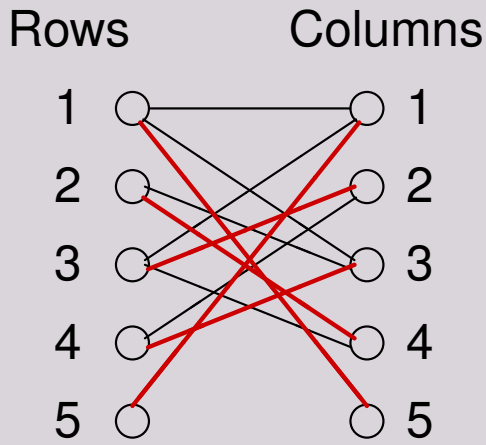
- (Perfect) matching. (At most) one edge for each row/column

What are Matchings

Example

$$\begin{pmatrix} * & & * & & * \\ & & * & * & \\ * & * & & * & \\ & * & * & & \\ * & & & & \end{pmatrix}$$

original A

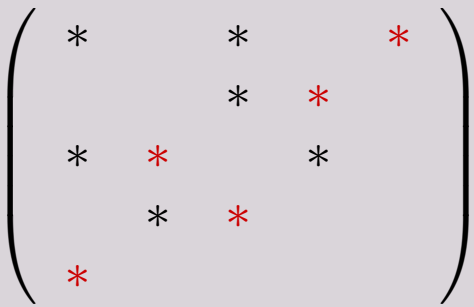


$\pi = (5, 3, 4, 2, 1)$

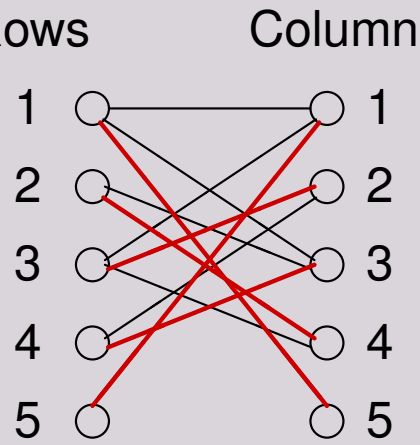
- Associated permutation

What are Matchings

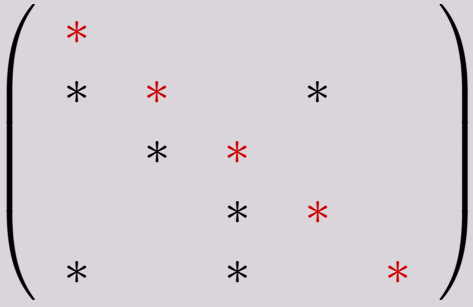
Example



original A



$\pi = (5, 3, 4, 2, 1)$



permuted $\Pi^\top A$

- Permuted matrix

Maximum Weight Matchings

Find matching such that $\prod_{i=1}^{n} |a_{\pi(i),i}|$ is maximized

- reformulation: minimize $\sum_{i=1}^{n} |c_{\pi(i),i}|$, where

$$c_{ij} = \begin{cases} \max_j |a_{ij}| - \log |a_{ij}| & a_{ij} \neq 0 \\ \infty & \text{otherwise} \end{cases}$$

- known linear–sum assignment problem,
  can be solved by sparse variant of Kuhn–Munkres algorithm

- dense case [Olschowka,Neumeier'96], sparse case [Duff,Koster'99]

Maximum Weight Matchings

Find matching such that $\prod_{i=1}^{n} |a_{\pi(i),i}|$ is maximized

- reformulation: minimize $\sum_{i=1}^{n} |c_{\pi(i),i}|$, where

$$c_{ij} = \begin{cases} \max_j |a_{ij}| - \log |a_{ij}| & a_{ij} \neq 0 \\ \infty & \text{otherwise} \end{cases}$$

- known linear–sum assignment problem,
  can be solved by sparse variant of Kuhn–Munkres algorithm
- solution leads to two vectors $(u_i)_i$, $(v_j)_j$ such that
  - $u_i + u_j \leqslant c_{ij}$ and if (i, j) are part of the matching $u_i + u_j = c_{ij}$
  - diagonal scalings $D_r = \mathrm{dgl}\,(e^{u_i})_i$, $D_c = \mathrm{dgl}\,(e^{v_i} / \max_k |a_{jk}|)_j$

- dense case [Olschowka,Neumeier'96], sparse case [Duff,Koster'99]

Maximum Weight Matchings

Find matching such that $\prod_{i=1}^{n} |a_{\pi(i),i}|$ is maximized

- reformulation: minimize $\sum_{i=1}^{n} |c_{\pi(i),i}|$, where

$$c_{ij} = \begin{cases} \max_j |a_{ij}| - \log |a_{ij}| & a_{ij} \neq 0 \\ \infty & \text{otherwise} \end{cases}$$

- known linear–sum assignment problem,
  can be solved by sparse variant of Kuhn–Munkres algorithm
- solution leads to two vectors $(u_i)_i$, $(v_j)_j$ such that
  - $u_i + u_j \leqslant c_{ij}$ and if (i,j) are part of the matching $u_i + u_j = c_{ij}$
  - diagonal scalings $D_r = \mathrm{dgl}\,(e^{u_i})_i$, $D_c = \mathrm{dgl}\,(e^{v_i} / \max_k |a_{jk}|)_j$
- yields permutation Π and diagonal scalings D_r and D_c.

$$A \to \Pi^\top D_r A D_c \text{ such that } |a_{ii}| = 1, |a_{ij}| \leqslant 1 \text{ for all } i,j$$

- dense case [Olschowka,Neumeier'96], sparse case [Duff,Koster'99]

Maximum Weight Matchings

MATLAB demo using the ILUPACK toolbox

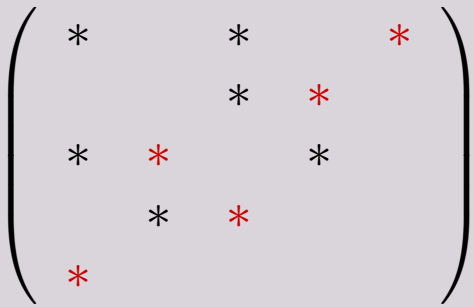
To download ILUPACK, visit http://ilupack.tu-bs.de

```
>> % add ILUPACK MEX interface drivers
>> addpath 'ilupack/'
>> % load sample matrix
>> load west0479; A=west0479;
>> % display matrix
>> figure(1);spy(A)
>> % reorder and rescale matrix using maximum weight matchings
>> [pl,pr,Dl,Dr]=mwm(A); M=Dl*A(pl,pr)*Dr;
>> % show improved block diagonal part
>> full(M(1:10,1:10))
>> % display reordered matrix
>> figure(2);spy(M)
>> % maxmimum weight matching + Metis nested dissection
>> [pl,pr,Dl,Dr]=mwmmetisn(A); M=Dl*A(pl,pr)*Dr;
>> figure(3);spy(M)
```

Basic Idea of Symmetric Matchings

Symmetric matchings [Duff,Gilbert'02], [Duff,Pralet'04]

Example



original A

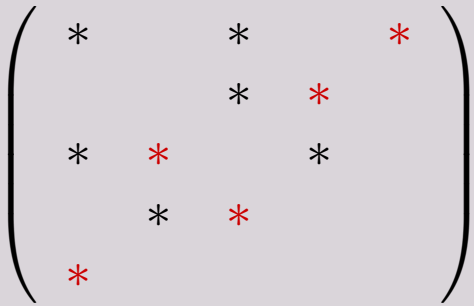
$$\pi \quad = \quad (5, 3, 4, 2, 1)$$

Basic Idea of Symmetric Matchings

Symmetric matchings [Duff,Gilbert'02], [Duff,Pralet'04]

- Decompose permutation as product of cycles

Example



original A

$$\pi = (5,3,4,2,1) = (5,1)(3,4,2)$$

Basic Idea of Symmetric Matchings

Symmetric matchings [Duff,Gilbert'02], [Duff,Pralet'04]

- Decompose permutation as product of cycles
- each cycle is associated with a diagonal block, pivots stay there

Example

$$\begin{pmatrix} * & & * & & * \\ & & * & * & \\ * & * & & * & \\ & * & * & & \\ * & & & & \end{pmatrix}$$

original A

$$\pi = (5,3,4,2,1) = (5,1)(3,4,2)$$

$$\begin{pmatrix} * & & & & \\ * & * & & * & \\ & * & * & & \\ & & * & * & \\ * & & * & & * \end{pmatrix}$$

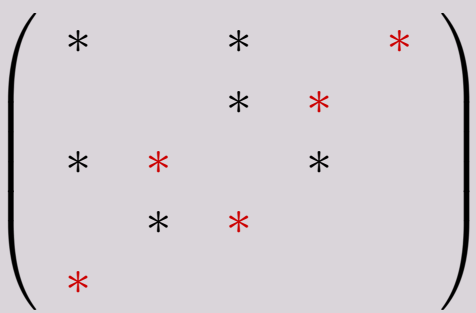
permuted $\Pi^\top A$

Basic Idea of Symmetric Matchings

Symmetric matchings [Duff,Gilbert'02], [Duff,Pralet'04]

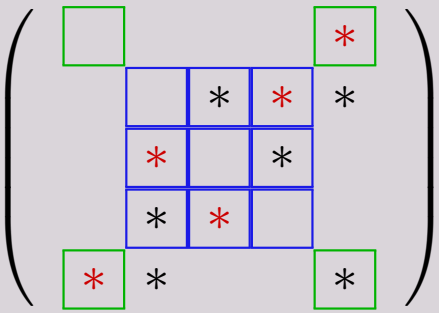
- Decompose permutation as product of cycles
- each cycle is associated with a diagonal block, pivots stay there
- pivots stay inside if related SYMMETRIC permutation is applied

Example



original A

$$\pi = (5,3,4,2,1) = (5,1)(3,4,2)$$

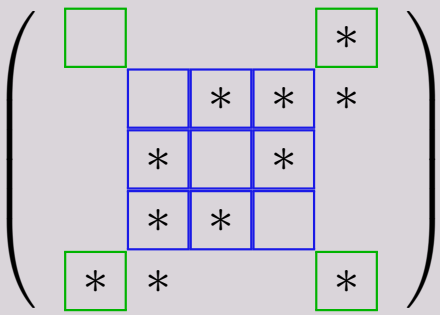


permuted $\Pi^\top A \Pi$

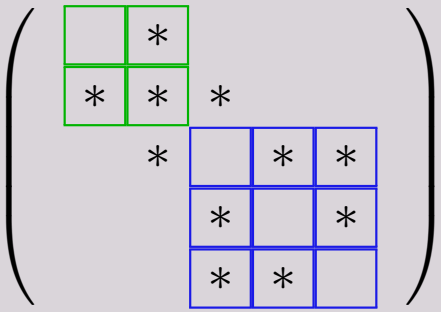
Basic Idea of Symmetric Matchings

- break up cycles into pairs of 2–cycles and group them together
  - various strategies of breaking up the cycles [Duff,Gilbert'02],[Duff,Pralet'04],[Schenk,Hagemann'04]
  - long cycles rarely show up in practice [Duff,Pralet'04]
- Scaling replaced by symmetric scaling $D = (D_c D_r)^{1/2}$ ensures that entries are still at most 1 in modulus

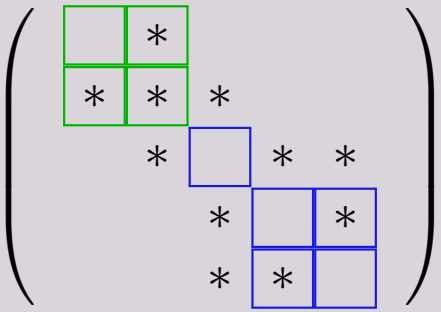
Example



$\Pi^\top A \Pi$

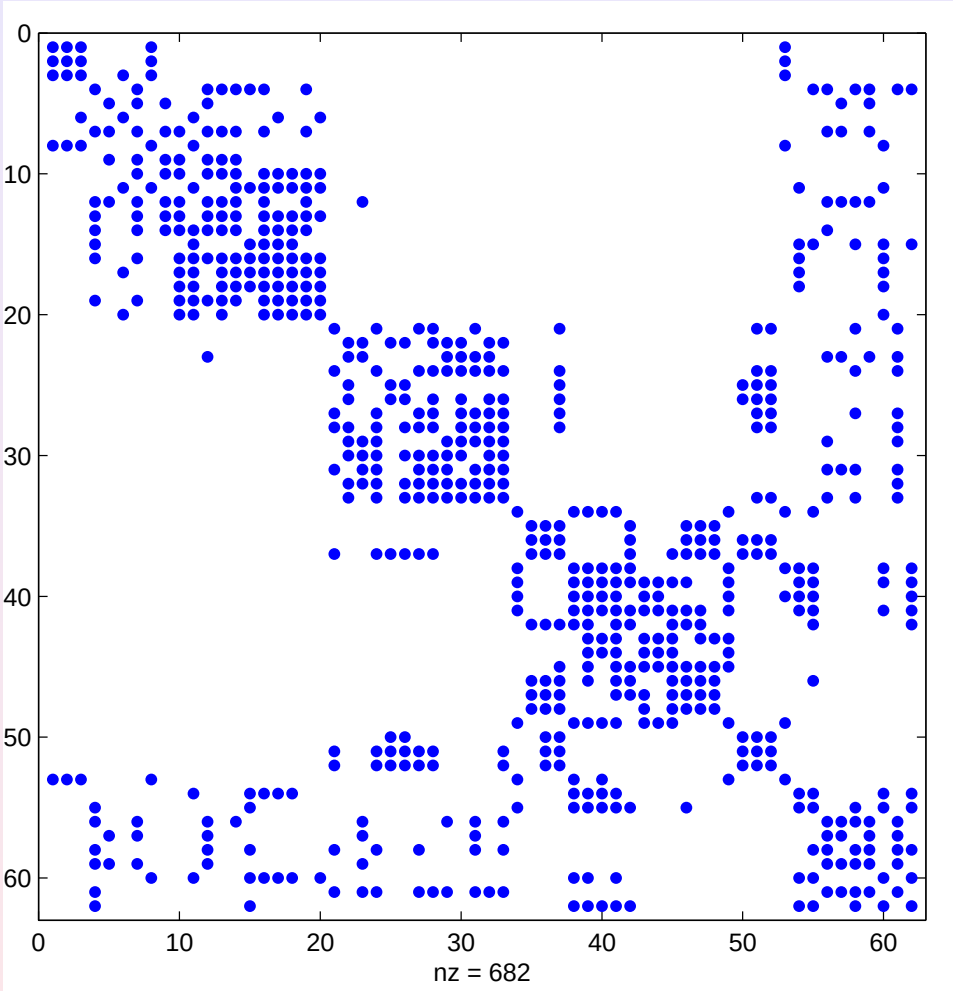


rearranged $\Pi_1^\top A \Pi_1$



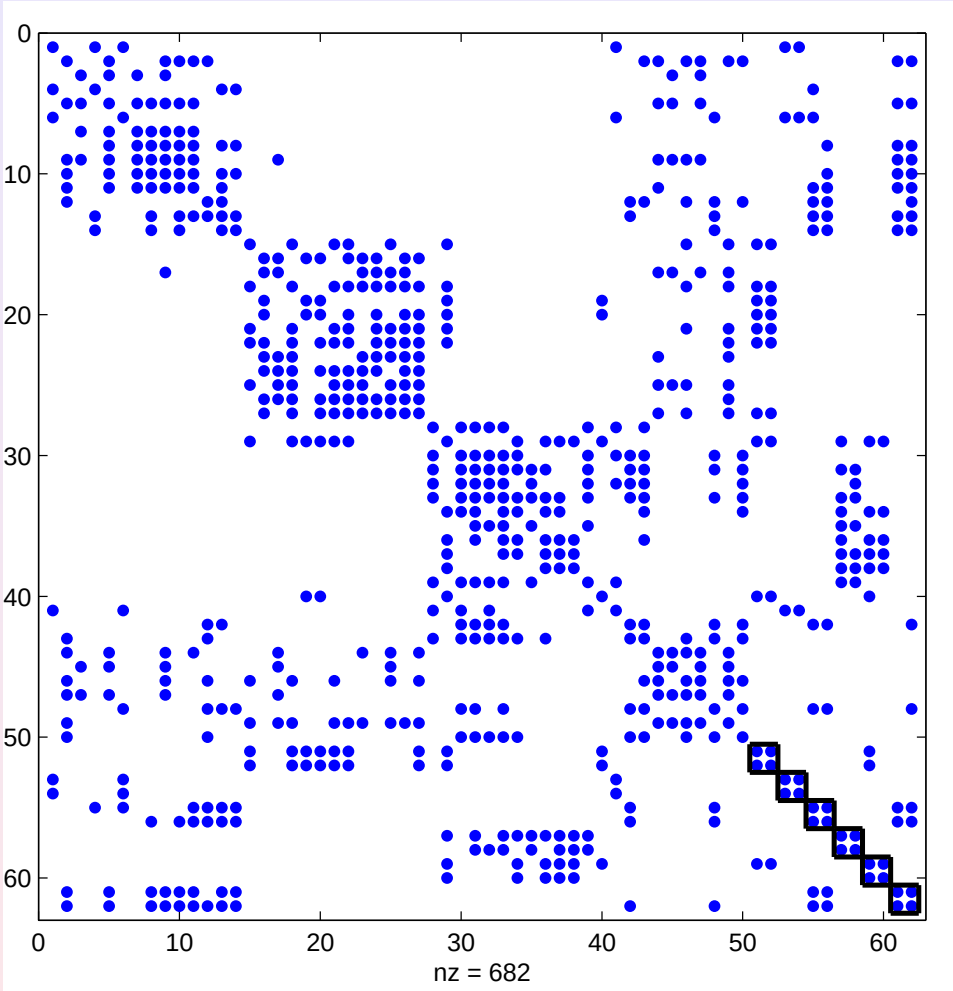
final $\Pi_2^\top A \Pi_2$

Reordering



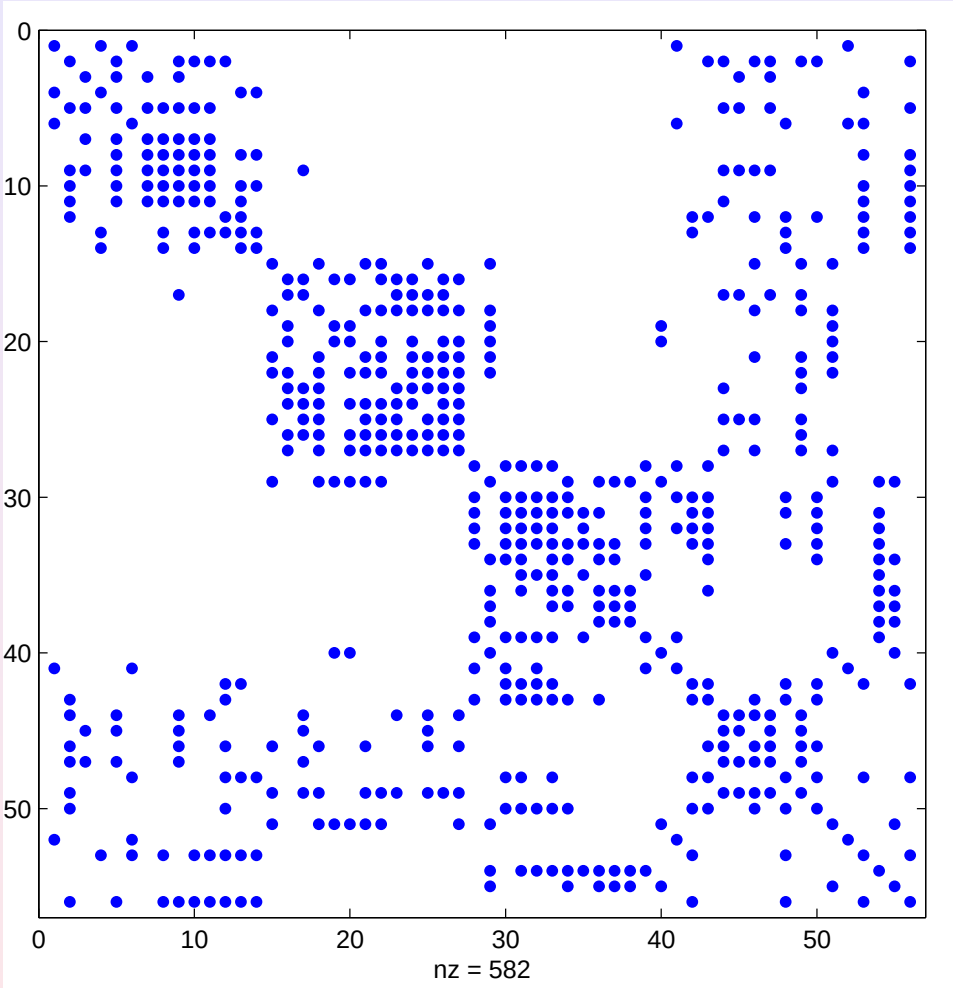
Original matrix

Reordering



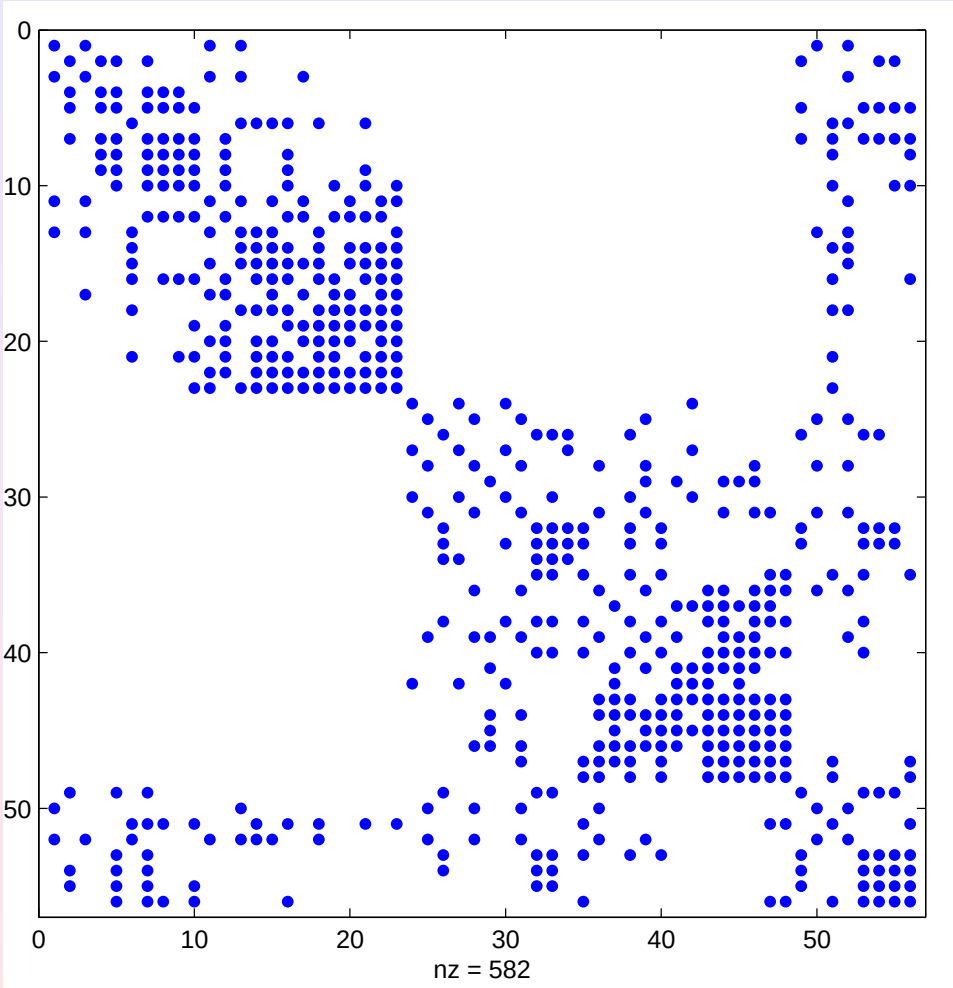
1. step. Apply symmetric maximum weight matching

Reordering



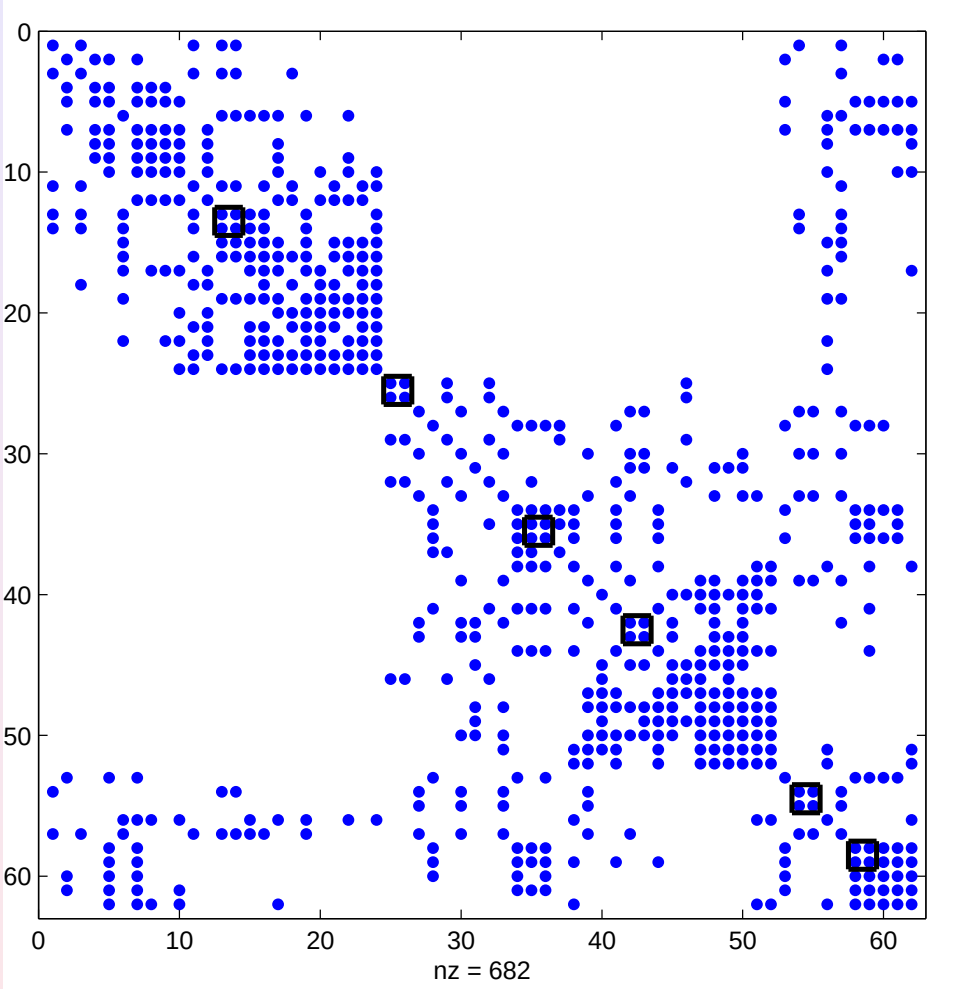
2. step. Compress 2×2 blocks

Reordering



3. step. Reorder the system with some reordering (AMD, METIS,RCM,. . .)

Reordering



4. step. Expand 2×2 blocks

Reordering

MATLAB demo using the ILUPACK toolbox

```
>> % create Anderson model matrix
>> m=50; w=0.1; A=anderson(m,w);
>> % reorder and rescale matrix using symmetric maximum weight
>> % matchings + RCM
>> [p,D]=symmwmrcm(A);
>> % show improved block diagonal part
>> M=D*A(p,p)*D; full(M(1:10,1:10))
>> % display reordered matrix
>> figure(1);spy(A(p,p))
>> % compare with the scalar variant by adding a diagonal shift
>> [p,D]=symmwmrcm(A+5*norm(A,1)*speye(m^3));
>> figure(2);spy(A(p,p))
>> % reorder and rescale now with Metis nested dissection
>> [p,D]=symmwmmetisn(A);figure(1);spy(A(p,p))
>> [p,D]=symmwmmetisn(A+5*norm(A,1)*speye(m^3));
>> figure(2);spy(A(p,p))
```

Technische Universität Braunschweig

Outline

Multilevel ILU

- rescaling + reordering

$$A \to \Pi^\top DAD\Pi$$

Multilevel ILU

- rescaling + reordering

$$A \to \Pi^\top DAD\Pi$$

- Select 1×1 and 2×2 block diagonal pivots

Multilevel ILU

- rescaling + reordering

$$A \to \Pi^\top DAD\Pi$$

- Select 1×1 and 2×2 block diagonal pivots
- postpone updates if “small” pivots occur

$$A \longrightarrow \begin{pmatrix} A_{\mathcal{F}\mathcal{F}} & A_{\mathcal{F}\mathcal{C}} \\ A_{\mathcal{C}\mathcal{F}} & A_{\mathcal{C}\mathcal{C}} \end{pmatrix}$$

Multilevel ILU

- rescaling + reordering

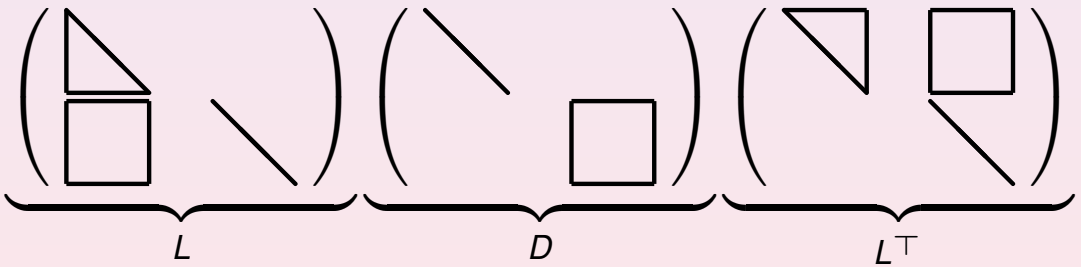
$$A \to \Pi^\top DAD\Pi$$

- Select 1×1 and 2×2 block diagonal pivots
- postpone updates if “small” pivots occur

$$A \longrightarrow \begin{pmatrix} A_{\mathcal{F}\mathcal{F}} & A_{\mathcal{F}\mathcal{C}} \\ A_{\mathcal{C}\mathcal{F}} & A_{\mathcal{C}\mathcal{C}} \end{pmatrix}$$

- approximate block decomposition

$$\begin{pmatrix} A_{\mathcal{F}\mathcal{F}} & A_{\mathcal{F}\mathcal{C}} \\ A_{\mathcal{C}\mathcal{F}} & A_{\mathcal{C}\mathcal{C}} \end{pmatrix} = \begin{pmatrix} L_{\mathcal{F}\mathcal{F}} & 0 \\ L_{\mathcal{C}\mathcal{F}} & I \end{pmatrix} \begin{pmatrix} D_{\mathcal{F}\mathcal{F}} & 0 \\ 0 & S_{\mathcal{C}\mathcal{C}} \end{pmatrix} \begin{pmatrix} L_{\mathcal{F}\mathcal{F}}^\top & L_{\mathcal{C}\mathcal{F}}^\top \\ 0 & I \end{pmatrix} + E$$



Multilevel ILU

- rescaling + reordering

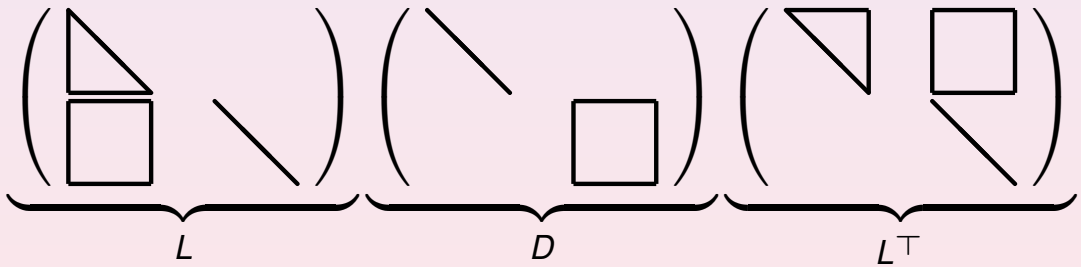
$$A \to \Pi^\top DAD\Pi$$

- Select 1×1 and 2×2 block diagonal pivots
- postpone updates if “small” pivots occur

$$A \longrightarrow \begin{pmatrix} A_{\mathcal{F}\mathcal{F}} & A_{\mathcal{F}\mathcal{C}} \\ A_{\mathcal{C}\mathcal{F}} & A_{\mathcal{C}\mathcal{C}} \end{pmatrix}$$

- approximate block decomposition

$$\begin{pmatrix} A_{\mathcal{F}\mathcal{F}} & A_{\mathcal{F}\mathcal{C}} \\ A_{\mathcal{C}\mathcal{F}} & A_{\mathcal{C}\mathcal{C}} \end{pmatrix} = \begin{pmatrix} L_{\mathcal{F}\mathcal{F}} & 0 \\ L_{\mathcal{C}\mathcal{F}} & I \end{pmatrix} \begin{pmatrix} D_{\mathcal{F}\mathcal{F}} & 0 \\ 0 & S_{\mathcal{C}\mathcal{C}} \end{pmatrix} \begin{pmatrix} L_{\mathcal{F}\mathcal{F}}^\top & L_{\mathcal{C}\mathcal{F}}^\top \\ 0 & I \end{pmatrix} + E$$



Multilevel approach: analogous principle recursively applied to $S_{\mathcal{C}\mathcal{C}}$, i.e., repeat

- matching and reordering
- postpone “bad” pivots while computing ILU

Matchings and Multilevel ILU

Several codes using unsymmetric/symmetric matchings, e.g.,

- PARDISO [Schenk, Gärtner]
- MUMPS [Amestoy et al.]
- HSL library [Duff, Scott, Reid, . . .]
- ILUPACK [Bo. et al.]

Many approaches to multilevel ILU, e.g.,

- (p)ARMS [Saad et al.]
- MRILU [Wubs et al.]
- ILUPACK [Bo. et al.]

Common believe

$$A \to \begin{pmatrix} A_{\mathcal{F}\mathcal{F}} & A_{\mathcal{F}\mathcal{C}} \\ A_{\mathcal{C}\mathcal{F}} & A_{\mathcal{C}\mathcal{C}} \end{pmatrix}$$

such that $A_{\mathcal{F}\mathcal{F}}$ is easy to factorize, (independent sets, (block) diagonal dominance, . . .)

Block diagonal dominance

Example Anderson model

Rows that satisfy $\sum_{j:j\neq i} |a_{ij}|/|a_{ii}| \leqslant 0.9$:

ω	14.0	$\omega_c = 16.5$	21.0
%	4.8%	19.2%	36.5%

Now scratch out one out of seven rows/columns to make $A_{\mathcal{F}\mathcal{F}}$ more diagonally dominant.

Rows of $A_{\mathcal{F}\mathcal{F}}$ that satisfy $\sum_{j:j\neq i} |a_{ij}|/|a_{ii}| \leqslant 0.9$:

ω	14.0	$\omega_c = 16.5$	21.0
%	20.6%	32.7%	47.1%

Block diagonal dominance

Example Anderson model

Rows that satisfy $\sum_{j:j\neq i} |a_{ij}|/|a_{ii}| \leqslant 0.9$:

ω	14.0	$\omega_c = 16.5$	21.0
%	4.8%	19.2%	36.5%

Now scratch out one out of seven rows/columns to make $A_{\mathcal{F}\mathcal{F}}$ more diagonally dominant.

Rows of $A_{\mathcal{F}\mathcal{F}}$ that satisfy $\sum_{j:j\neq i} |a_{ij}|/|a_{ii}| \leqslant 0.9$:

ω	14.0	$\omega_c = 16.5$	21.0
%	20.6%	32.7%	47.1%

Remark. For Helmholtz we could similarly scratch out some rows/columns

Block diagonal dominance

Example Anderson model

Rows that satisfy $\sum_{j:j\neq i} |a_{ij}|/|a_{ii}| \leqslant 0.9$:

ω	14.0	$\omega_c = 16.5$	21.0
%	4.8%	19.2%	36.5%

Now scratch out one out of seven rows/columns to make $A_{\mathcal{F}\mathcal{F}}$ more diagonally dominant.

Rows of $A_{\mathcal{F}\mathcal{F}}$ that satisfy $\sum_{j:j\neq i} |a_{ij}|/|a_{ii}| \leqslant 0.9$:

ω	14.0	$\omega_c = 16.5$	21.0
%	20.6%	32.7%	47.1%

Remark. For Helmholtz we could similarly scratch out some rows/columns

Warning

There is no guarantee that the (approximate) Schur complement will have a similar diagonal dominance structure!

Inverse–Based Pivoting

Generalizing block diagonal dominance

- Suppose that $A = A^T$, $A = LDL^T$ and

$$\sum_{i:i\neq j} \left| A_{ij} A_{jj}^{-1} \right| \leqslant \frac{\kappa - 1}{\kappa} \Longrightarrow \|L^{-1}\|_1 \leqslant \kappa.$$

- Idea! Instead of constructing some block diagonal dominant block $A_{\mathcal{F}\mathcal{F}}$, directly construct L such that $\|L^{-1}\| \leqslant \kappa$.
- $\|L^{-1}\|$ can be efficiently estimated [Cline,Moler,Stewart,Wilkinson'77]

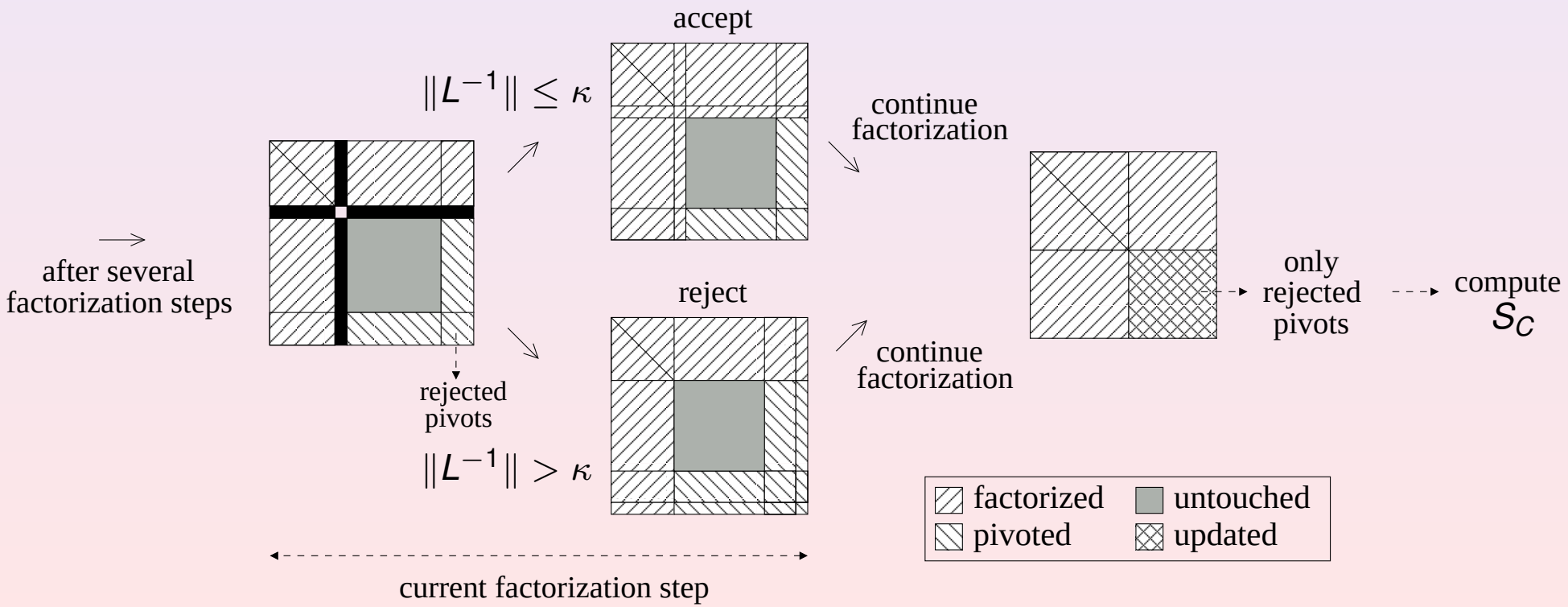
Inverse–Based Pivoting

Generalizing block diagonal dominance

- Suppose that $A = A^T$, $A = LDL^T$ and

$$\sum_{i:i\neq j} \left| A_{ij} A_{jj}^{-1} \right| \leqslant \frac{\kappa - 1}{\kappa} \Longrightarrow \|L^{-1}\|_1 \leqslant \kappa.$$

- Idea! Instead of constructing some block diagonal dominant block $A_{\mathcal{F}\mathcal{F}}$, directly construct L such that $\|L^{-1}\| \leqslant \kappa$.
- $\|L^{-1}\|$ can be efficiently estimated [Cline,Moler,Stewart,Wilkinson'77]



Inverse-based Pivoting

MATLAB demo using the ILUPACK toolbox

```
>> % create Anderson model matrix
>> m=50; w=16.5; A=anderson(m,w);
>> % init parameters to their default values
>> options=AMGinit(A)
>> % factorize A using multilevel ILU and inverse-based pivoting
>> tic;[P,options]=AMGfactor(A,options);toc
>> % display reordered matrix
>> figure(1);AMGspy(P); figure(2);AMGspy(A,P);
>> % fill-in of the preconditioner
>> AMGnnz(P)/nnz(A)
>> P
>> % iterative solution
>> tic;[x,options]=AMGsolver(A,P,options,A*ones(m^3,1));toc
>> options.niter
>> % remove preconditioner
>> P=AMGdelete(P)
```

Inverse–Based Pivoting

- Original background [Bo., Saad'06]. Error introduced by ILU

$$A = LDU + E,$$

preconditioning requires small "inverse error"

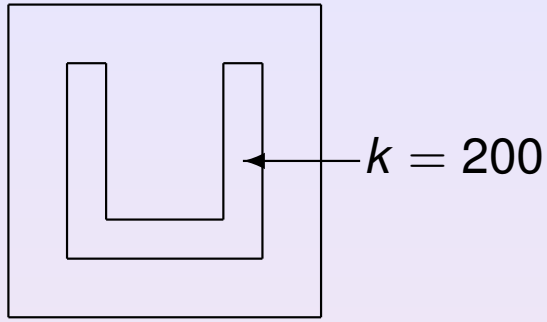
$$L^{-1}AU^{-1} = D + L^{-1}EU^{-1} \equiv D + F.$$

D is not taken into account, but a major approach to bound the error.

- Also connections to factored approximate inverses were one root.
- More recently, observations indicated that a tight κ such as $\kappa = 3$ (Anderson) let the inverse-based pivoting process act like a coarsening strategy.

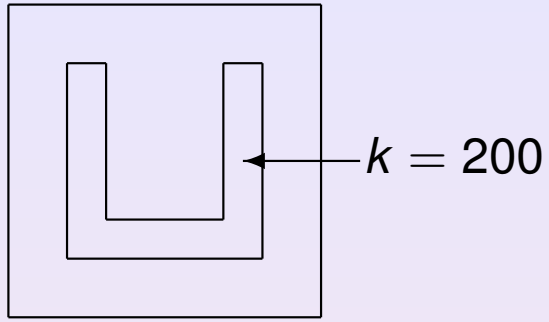
Inverse-based coarsening in the Helmholtz case

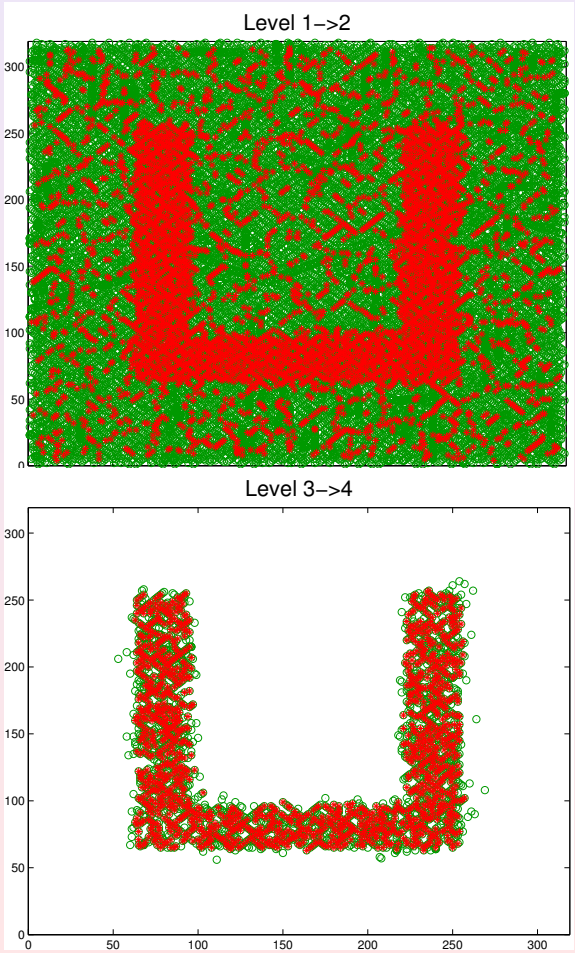
$$-\Delta u - k^2 u = b$$

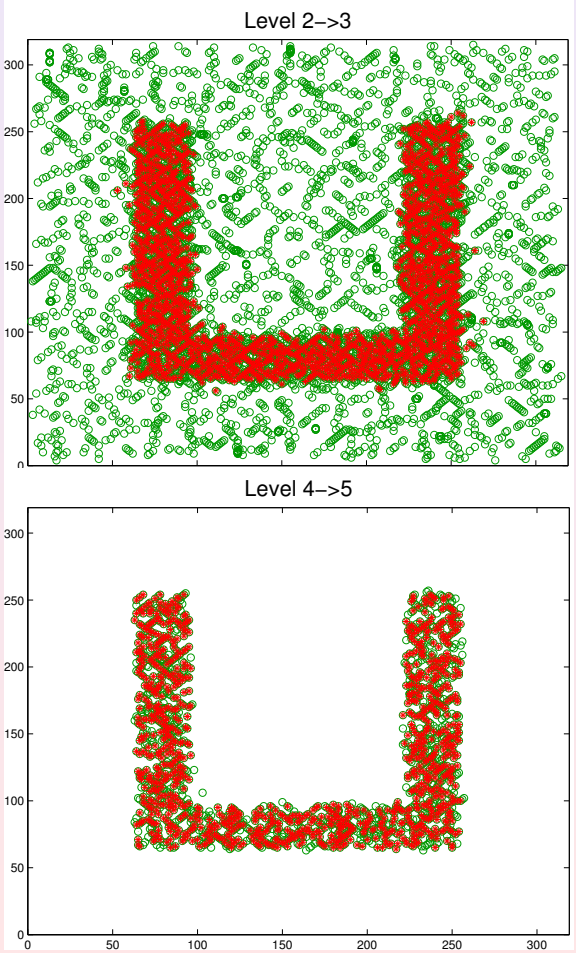


Inverse-based coarsening in the Helmholtz case

$$-\Delta u - k^2 u = b$$







Coarse Grid System

$$A^{-1} \approx \underbrace{L^{-T}}_{\approx \kappa} D^{-1} \underbrace{L^{-1}}_{\approx \kappa}$$

Coarse Grid System

$$
\begin{aligned}
A^{-1} &\approx \underbrace{L^{-T}}_{\approx\kappa} D^{-1} \underbrace{L^{-1}}_{\approx\kappa} \\
&= \begin{pmatrix} (L_{\mathcal{F}\mathcal{F}} D_{\mathcal{F}\mathcal{F}} L_{\mathcal{F}\mathcal{F}}^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \underbrace{\begin{pmatrix} -L_{\mathcal{F}\mathcal{F}}^{-T} L_{\mathcal{C}\mathcal{F}}^T \\ I \end{pmatrix}}_{I_h} S_{\mathcal{C}\mathcal{C}}^{-1} \underbrace{\begin{pmatrix} -L_{\mathcal{C}\mathcal{F}} L_{\mathcal{F}\mathcal{F}}^{-1} & I \end{pmatrix}}_{I_h^T}
\end{aligned}
$$

Coarse Grid System

$$
\begin{aligned}
A^{-1} &\approx \underbrace{L^{-T}}_{\approx\kappa} D^{-1} \underbrace{L^{-1}}_{\approx\kappa} \\
&= \begin{pmatrix} (L_{\mathcal{F}\mathcal{F}} D_{\mathcal{F}\mathcal{F}} L_{\mathcal{F}\mathcal{F}}^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \underbrace{\begin{pmatrix} -L_{\mathcal{F}\mathcal{F}}^{-T} L_{\mathcal{C}\mathcal{F}}^T \\ I \end{pmatrix}}_{I_h} S_{\mathcal{C}\mathcal{C}}^{-1} \underbrace{\begin{pmatrix} -L_{\mathcal{C}\mathcal{F}} L_{\mathcal{F}\mathcal{F}}^{-1} & I \end{pmatrix}}_{I_h^T}
\end{aligned}
$$

$$
Ax = \varepsilon x
$$

Coarse Grid System

$$A^{-1} \approx \underbrace{L^{-T}}_{\approx\kappa} D^{-1} \underbrace{L^{-1}}_{\approx\kappa}$$

$$= \begin{pmatrix} (L_{\mathcal{F}\mathcal{F}} D_{\mathcal{F}\mathcal{F}} L_{\mathcal{F}\mathcal{F}}^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \underbrace{\begin{pmatrix} -L_{\mathcal{F}\mathcal{F}}^{-T} L_{\mathcal{C}\mathcal{F}}^T \\ I \end{pmatrix}}_{I_h} S_{\mathcal{C}\mathcal{C}}^{-1} \underbrace{\begin{pmatrix} -L_{\mathcal{C}\mathcal{F}} L_{\mathcal{F}\mathcal{F}}^{-1} & I \end{pmatrix}}_{I_h^T}$$

$$Ax = \varepsilon x$$

$$\underbrace{\frac{1}{\varepsilon} x}_{\text{LARGE}} = A^{-1} x \approx \begin{pmatrix} \underbrace{(L_{\mathcal{F}\mathcal{F}} D_{\mathcal{F}\mathcal{F}} L_{\mathcal{F}\mathcal{F}}^T)^{-1}}_{\approx C} & 0 \\ 0 & 0 \end{pmatrix} x + \underbrace{I_h}_{\approx\kappa} S_{\mathcal{C}\mathcal{C}}^{-1} \underbrace{I_h^T}_{\approx\kappa} x$$

Coarse Grid System

$$A^{-1} \approx \underbrace{L^{-T}}_{\approx\kappa} D^{-1} \underbrace{L^{-1}}_{\approx\kappa}$$

$$= \begin{pmatrix} (L_{\mathcal{FF}} D_{\mathcal{FF}} L_{\mathcal{FF}}^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix} + \underbrace{\begin{pmatrix} -L_{\mathcal{FF}}^{-T} L_{\mathcal{CF}}^T \\ I \end{pmatrix}}_{I_h} S_{\mathcal{CC}}^{-1} \underbrace{\begin{pmatrix} -L_{\mathcal{CF}} L_{\mathcal{FF}}^{-1} & I \end{pmatrix}}_{I_h^T}$$

$$Ax = \varepsilon x$$

$$\underbrace{\frac{1}{\varepsilon} x}_{\text{LARGE}} = A^{-1}x \approx \begin{pmatrix} \underbrace{(L_{\mathcal{FF}} D_{\mathcal{FF}} L_{\mathcal{FF}}^T)^{-1}}_{\approx C} & 0 \\ 0 & 0 \end{pmatrix} x + \underbrace{I_h}_{\approx\kappa} \underbrace{S_{\mathcal{CC}}^{-1}}_{\text{LARGE}} \underbrace{I_h^T}_{\approx\kappa} x$$

$\Rightarrow$ by inverse-based coarsening $S_{\mathcal{CC}}$ captures the eigenvalues with small modulus

Eigenvalues

Theorem (Bo., Grote, Schenk'08)

Let A be a nonsingular, real, symmetric.
Under some moderate assumptions [. . .] we have that all small positive eigenvalues μ_j *of* S_C *satisfy*

$$\lambda_j \leqslant \mu_j \lessapprox \kappa^2 \lambda_j,$$

where λ_j *are their counter parts w.r.t. A.*

A similar result holds for the largest negative eigenvalues close to 0*.*

Remark. This result does not apply dropping.

Analogy to Multigrid Methods

Eigenvalues of coarse grid systems

$$-\Delta u - k^2 u = f \text{ in } \Omega = [0,1]^2, k = 50, \quad N = 80 = [5k/\pi]$$

Standard "Geometric" Multigrid, Eigenvalue Distribution

Level	N	size	negative ev.	positive ev.
1	80	6'400	187	6213
2	40	1'600	205	1395
3	20	400	313	87
4	10	100	100	0

Algebraic Multilevel ILU, Eigenvalue Distribution

Level	size	negative ev.	positive ev.
1	6'400	187	6213
2	2'276	187	2089
3	1'284	190	1094
4	801	164	637
5	350	—	—

- For standard multigrid, eigenvalues cross the origin while coarsening
- For drop tolerance $\tau = 0$, this cannot happen to the algebraic multilevel ILU
- Experiment uses $\tau = 10^{-2}$

Technische Universität Braunschweig

Outline

Shifted Systems

- To improve the diagonal dominance of A we could introduce imaginary shifts

$$A \to A + \mathbf{i}\beta I, \text{ for some } \beta > 0.$$

- systems change from real symmetric to complex symmetric
- Helmholtz case

$$A - k^2 M \to A - (1 - \mathbf{i}\beta)k^2 M.$$

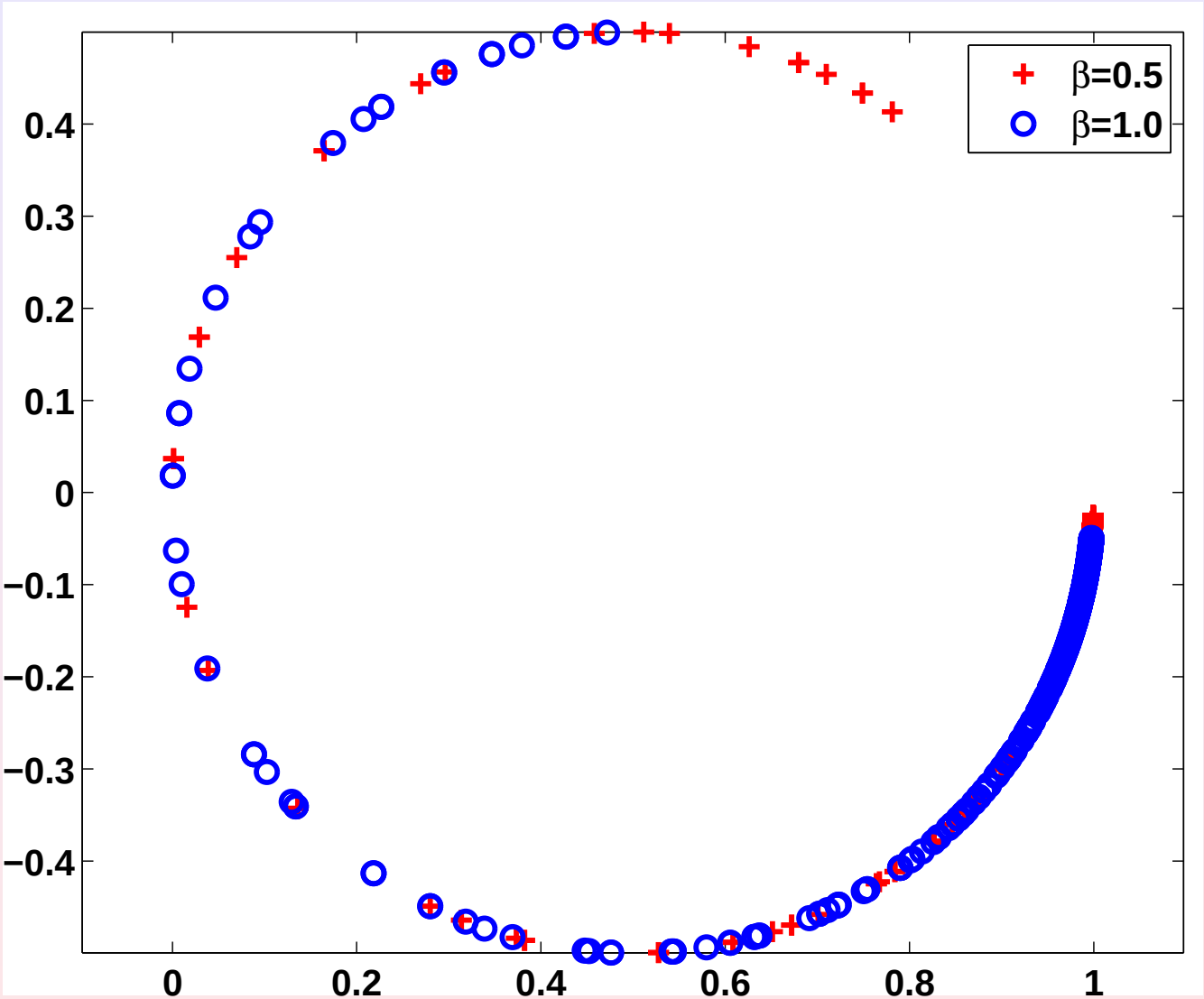
- [Magolu '01] was among the first people to use shifts for ILUs to prevent large fill

- [Erlangga, Vuik, Oosterlee'04,'06] successfully use shifts to set up multigrid for the shifted system and apply it to the original system
- [van Gijzen, Erlangga, Vuik'07] show that the eigenvalues of

$$(A - (1 - \beta i)k^2 M)^{-1}(A - k^2 M)$$

are located on a disc around $z = \frac{1}{2}$ with radius $r = \frac{1}{2}$ in the complex plane.

Eigenvalues of the preconditioned system



Shifted Systems

MATLAB demo using the ILUPACK toolbox

```
>> % create 3D Helmholtz matrix with zero b.c.
>> k=50; L=5; [A,m]=helmholtz(k,L); size(A,1)
>> % shift matrix with a complex shift
>> M=A+sqrt(-1)*0.1*k^2*speye(m^3);
>> % init parameters to their default values
>> options=AMGinit(M); options.maxit=1000
>> % factorize shifted matrix M using multilevel ILU
>> tic;[P,options]=AMGfactor(M,options);toc
>> AMGnnz(P)/nnz(A), length(P), P(1), AMGspy(P)
>> % iterative solution
>> tic;[x,options]=AMGsolver(A,P,options,A*ones(m^3,1));toc
>> options.niter
>> % remove preconditioner
>> P=AMGdelete(P)
```

Technische Universität Braunschweig

Outline

Iterative Solver

- system A is real or complex symmetric
- preconditioner P is also real (resp. complex) symmetric.
- The simplified QMR method [Freund, Jarre'99] is a quasi-minimum residual method
- formally a two-sided Lanczos method with a special left initial guess
- sQMR can make use of symmetry
- one step sQMR is almost as cheap as one step of CG (i.e., only one mat-vec, one prec step)

Anderson model

$n = m^3$, $m = 100, 130, 160, 190$. $\omega = \omega_c = 16.5$.

Compute 5 eigenvalues and associated eigenvectors closest to $\lambda = 0$.

	time			
size $n = m^3$	10^6	$2 \cdot 10^6$	$4 \cdot 10^6$	$7 \cdot 10^6$
Cullum–Willoughby Lanczos	71 : 04	—	—	—

Anderson model

$n = m^3$, $m = 100, 130, 160, 190$. $\omega = \omega_c = 16.5$.

Compute 5 eigenvalues and associated eigenvectors closest to $\lambda = 0$.

	time			
size $n = m^3$	10^6	$2 \cdot 10^6$	$4 \cdot 10^6$	$7 \cdot 10^6$
Cullum–Willoughby Lanczos	71 : 04	—	—	—
ARPACK + PARDISO	5 : 37	—	—	—

Anderson model

$n = m^3$, $m = 100, 130, 160, 190$. $\omega = \omega_c = 16.5$.

Compute 5 eigenvalues and associated eigenvectors closest to $\lambda = 0$.

	time			
size $n = m^3$	10^6	$2 \cdot 10^6$	$4 \cdot 10^6$	$7 \cdot 10^6$
Cullum–Willoughby Lanczos	71 : 04	—	—	—
ARPACK + PARDISO	5 : 37	—	—	—
ARPACK + ILUPACK	0 : 45	5 : 46	13 : 58	—

Anderson model

$n = m^3$, $m = 100, 130, 160, 190$. $\omega = \omega_c = 16.5$.

Compute 5 eigenvalues and associated eigenvectors closest to $\lambda = 0$.

	time			
size $n = m^3$	10^6	$2 \cdot 10^6$	$4 \cdot 10^6$	$7 \cdot 10^6$
Cullum–Willoughby Lanczos	71 : 04	—	—	—
ARPACK + PARDISO	5 : 37	—	—	—
ARPACK + ILUPACK	0 : 45	5 : 46	13 : 58	—
Jacobi–Davidson + ILUPACK	0 : 18	1 : 01	5 : 03	9 : 06

results from [Schenk,Bo.,Römer'05,'08]

Anderson model

$n = m^3$, $m = 100, 130, 160, 190$. $\omega = \omega_c = 16.5$.

Compute 5 eigenvalues and associated eigenvectors closest to $\lambda = 0$.

	memory [GB]			
size $n = m^3$	10^6	$2 \cdot 10^6$	$4 \cdot 10^6$	$7 \cdot 10^6$
Cullum–Willoughby Lanczos	0.3	—	—	—
ARPACK + PARDISO	14.3	—	—	—
ARPACK + ILUPACK	1.4	3.0	5.8	—
Jacobi–Davidson + ILUPACK	1.4	3.0	5.8	9.6

results from [Schenk,Bo.,Römer'05,'08]

3D seismic imaging problem

- Helmholtz equation $-\Delta u - (1 - i\alpha)k^2 u = b$ in 3D ($20km \times 20km \times 4km$)
- inverse-based multilevel ILU applied to shifted system $A + i \cdot 0.1k^2 I$

3D seismic imaging problem

- Helmholtz equation $-\Delta u - (1 - i\alpha)k^2 u = b$ in 3D ($20km \times 20km \times 4km$)
- inverse-based multilevel ILU applied to shifted system $A + i \cdot 0.1k^2 I$

Example

freq. [Hz]	grid points	damping α	Multilevel ILU iteration (secs)
2.5	$101 \times 101 \times 24$	0.0	17 (25)
		0.025	14 (22)
		0.050	12 (15)
5	$201 \times 201 \times 48$	0.0	35 (410)
		0.025	21 (321)
		0.050	17 (268)
10	$401 \times 401 \times 94$	0.0	78 (4,243)
		0.025	54 (3,188)
		0.050	30 (1,602)

results from [Bo.,Grote,Schenk'09]

Shifted Maxwell equations

- add artificial conductivity for **E** and similar contribution for **H** such that the reduced second order system refers to a shifted Helmholtz equation

$$\left\{ \begin{array}{rcl} +\mathbf{i}\omega\varepsilon\mathbf{E} - \mathrm{curl}(\mathbf{H}) & = & -\beta_{\mathbf{E}}\omega\varepsilon\mathbf{E}, \\ -\mathbf{i}\omega\mu\mathbf{H} - \mathrm{curl}(\mathbf{E}) & = & +\beta_{\mathbf{H}}\omega\mu\mathbf{H}, \end{array} \right. \quad + \text{b.c.}$$

- Reduced second order system

$$\rightarrow \left\{ \begin{array}{c} -(1 - \beta_{\mathbf{E}}\beta_{\mathbf{H}} - (\beta_{\mathbf{E}} + \beta_{\mathbf{H}})\mathbf{i})\omega^2 \mathrm{div}(\varepsilon \mathrm{grad})\varphi, \\ \left(-(1 - \beta_{\mathbf{E}}\beta_{\mathbf{H}} - (\beta_{\mathbf{E}} + \beta_{\mathbf{H}})\mathbf{i})\omega^2\varepsilon - \mathrm{div}(\frac{1}{\mu}\mathrm{grad})\right)\mathbf{U} \end{array} \right.$$

  Essentially reduces to a Helmholtz equation

Technische Universität Braunschweig

Outline

Conclusions

- structure-preserving ILU-type methods for indefinite systems
- combinatorial approach using maximum weight matchings to improve block diagonal pivots
- multilevel and inverse-based approach to deal with small pivots

Software

- ILUPACK, multilevel preconditioning package and solvers for various type of problems.
  To download *ILUPACK*, visit `http://ilupack.tu-bs.de`

- JADAMILU, Jacobi-Davidson eigenvalue solver for generalized symmetric / Hermitian eigenvalue problems including ILUPACK's preconditioner.
  To download *JADAMILU*, visit `http://homepages.ulb.ac.be/~jadamilu/`