Acceleration strategies and applications

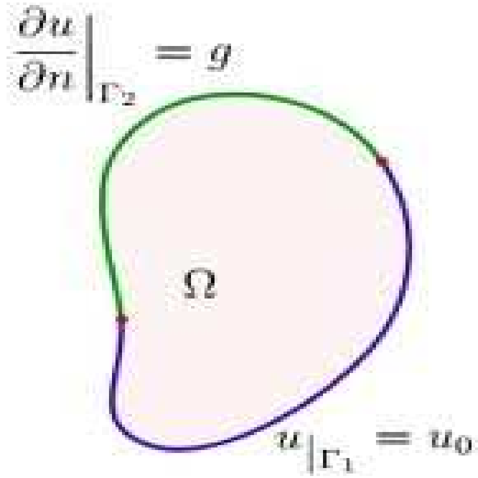
Outline

- Some common elliptic operators
- Finite Difference schemes for 2D operators
- Sparse matrices
- General preconditioning strategies
- Saddle point problems

Some common elliptic operators

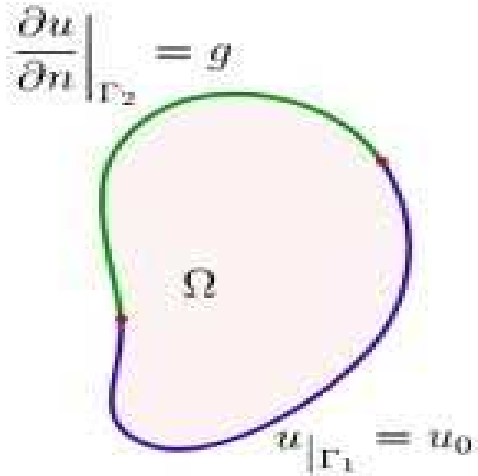
Given $\Omega \subset \mathbb{R}^2$ bounded, open domain, $\Gamma = \partial\Omega$. Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f, \qquad (x, y) \in \Omega$$



equipped with *boundary* conditions,

Some common elliptic operators



Given $\Omega \subset \mathbb{R}^2$ bounded, open domain, $\Gamma = \partial\Omega$. Poisson equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = f, \qquad (x, y) \in \Omega$$

equipped with *boundary* conditions, that is, for (x, y) on Γ, e.g.:

Dirichlet conditions: $u(x, y) = \phi(x, y)$

Neumann conditions: $\frac{\partial u}{\partial \mathbf{n}} = 0 \qquad (\nabla u \cdot \mathbf{n} = 0)$

Cauchy conditions: $\frac{\partial u}{\partial \mathbf{n}} + \alpha(x, y)u(x, y) = \gamma(x, y)$

Note: possibly mixed conditions on parts of the domain (e.g., $\Gamma = \Gamma_1 \cup \Gamma_2$, with Dirichlet cond. on Γ_1, Neumann cond on Γ_2)

Some common elliptic operators

More general,

$$Lu = f, \qquad L = \frac{\partial}{\partial x}\left(a_1 \frac{\partial}{\partial x}\right) + \frac{\partial}{\partial y}\left(a_2 \frac{\partial}{\partial y}\right)$$

(or, more compactly, $L = \nabla \cdot (\mathbf{a} \cdot \nabla)$)

In case of an anisotropic and inhomogeneous medium. In general

$$Lu = \nabla \cdot (\mathbb{A}\nabla)u, \qquad \mathbb{A} \in \mathbb{R}^{2\times 2}$$

$\mathbb{A}$: tensor acting on both components of ∇

The (steady-state) convection diffusion equation:

$$-\nabla \cdot (\mathbf{a}.\nabla)u + \mathbf{b} \cdot \nabla u = f$$

the magnitude of the vector $\mathbf{b}$ is a measure of non-selfadjointness of the equation.

Finite difference: basic approximations

$$\frac{du}{dx} = \frac{u(x+h) - u(x)}{h} - \frac{h}{2}\frac{d^2u(x)}{dx^2} + O(h^2), \quad h \to 0$$
$$\frac{du}{dx} = \frac{u(x) - u(x-h)}{h} + \frac{h}{2}\frac{d^2u(x)}{dx^2} + O(h^2), \quad h \to 0$$

Centered approximation: Combining these two approximations,

$$\frac{du}{dx} = \frac{u(x+h) - u(x-h)}{2h} + O(h^2), \quad h \to 0$$

second order accuracy!

$\Rightarrow$ Two-point stencils

Some common elliptic operators

Approximating the second derivative:

$$\frac{d^2u}{dx^2} = \frac{u_x(x+h) - u_x(x)}{h}, \qquad h > 0, h \to 0$$

Combining forward and backward approximation of u_x,

$$\frac{d^2u}{dx^2} = \frac{u(x+h) - 2u(x) + u(x-h)}{h^2} + O(h^2), \quad h \to 0$$

⇒ Three-point stencil

More general second order operator:

$$\frac{d}{dx}\left(a(x)\frac{du}{dx}\right) = \frac{a_{i+\frac{1}{2}}(u_{i+1} - u_i) - a_{i-\frac{1}{2}}(u_i - u_{i-1})}{h^2} + O(h^2), \quad h \to 0$$

where $u_{i+1} = u(x+h)$, $a_{i+\frac{1}{2}} = a(x + \frac{1}{2}h)$, etc.

Difference schemes for the 2D Laplace operator

Using h_1 in x-direction and h_2 in y-direction,

$$\begin{aligned}\Delta u &\equiv u_{xx} + u_{yy} \\ &\approx \frac{u(x+h_1,y) - 2u(x,y) + u(x-h_1,y)}{h_1^2} \\ &\qquad + \frac{u(x,y+h_2) - 2u(x,y) + u(x,y-h_2)}{h_2^2}\end{aligned}$$

that is, for $h_1 = h_2 = h$,

$$\Delta u \approx \frac{1}{h^2}\left(u(x+h,y) + u(x-h,y) + u(x,y+h) + u(x,y-h) - 4u(x,y)\right)$$

Actual implementation. 1D

Consider the 1D problem

$$-u''(x) = f(x), \quad x \in (0,1),$$
$$u(0) = u(1) = 0$$

Discretization of interval $[0, 1]$ with $n + 2$ *nodes*:
$x_i = ih, i = 0, 1, \ldots, n + 1$

Note: $h = \frac{1}{n+1}$

Note: Dirichlet b.c., $u(0) = u(x_0)$ and $u(1) = u(x_{n+1})$ known

Write $u(x_i) \equiv u_i$. Then the *discrete* version of the diff.equation is

$$-u_{i-1} + 2u_i - u_{i+1} = h^2 f_i, \qquad i = 1, \ldots, n$$

Actual implementation. 1D

$$(-u_{i-1} + 2u_i - u_{i+1}) = h^2 f_i, \qquad i = 1, \ldots, n$$

Collecting all i's, we obtain $\quad A\mathbf{u} = \mathbf{f}$ with

$$A = \begin{bmatrix} 2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}, \qquad \mathbf{f} = \begin{bmatrix} f_0 + u(0) \\ f_1 \\ \vdots \\ f_n \\ f_{n+1} + u(1) \end{bmatrix}$$

Neumann boundary conditions

Assume: $u'(0) = 0$. Therefore $u(x_1) - u(x_0) = 0 \Leftrightarrow u(x_0) = u(x_1)$

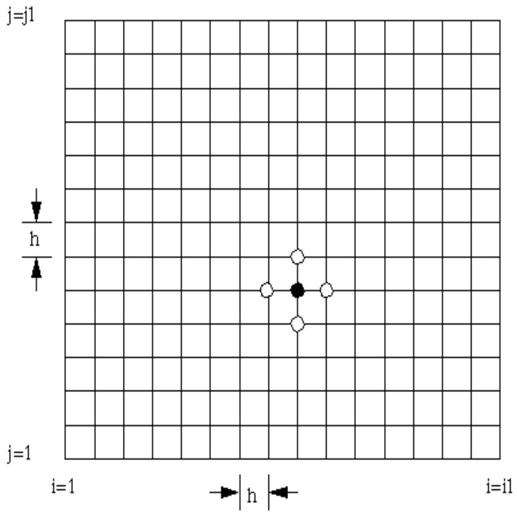
In the generic equation $\frac{1}{h^2}(-u_{i-1} + 2u_i - u_{i+1}) = f_i,\; i = 1, \ldots, n$

For $i = 1$ we obtain $\frac{1}{h^2}(-u_1 + 2u_1 - u_2) = \frac{1}{h^2}(u_1 - u_2)$

Therefore,

$$A = \begin{bmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 2 \end{bmatrix}, \qquad f = \begin{bmatrix} f_0 \\ f_1 \\ \vdots \\ f_n \\ f_{n+1} + u(1) \end{bmatrix}$$

Actual implementation. Poisson equation in a square

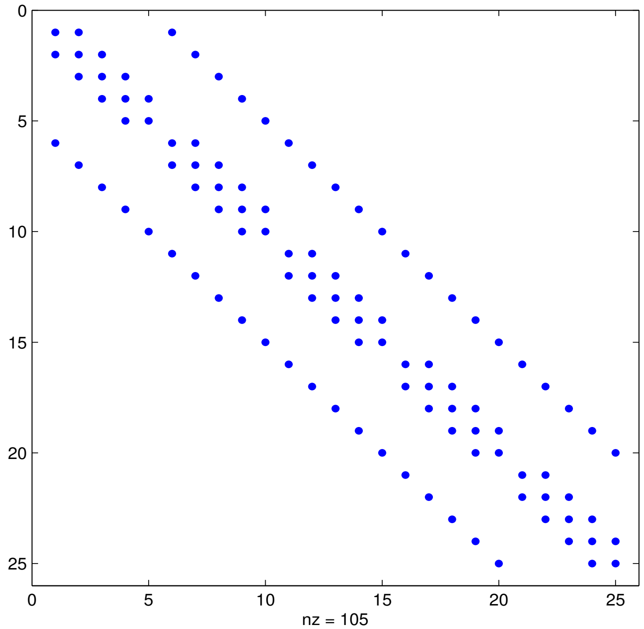


$$\text{lexicographic order : } \mathbf{u} = \begin{bmatrix} u_{11} \\ \vdots \\ u_{n,1} \\ u_{2,1} \\ \vdots \\ u_{2,n} \\ \vdots \end{bmatrix}$$

$$\text{So that: } A = \begin{bmatrix} & & & \cdots & & & \\ \cdots \underbrace{-1}_{i,j-1} & 0 \ldots \underbrace{-1}_{i-1,j} & \underbrace{4}_{i,j} & \underbrace{-1}_{i+1,j} & \ldots 0 & \underbrace{-1}_{i,j+1} \cdots \\ & & & \cdots & & & \end{bmatrix}$$

2D Poisson equation. The coefficient matrix

$$A = \begin{bmatrix} & & & & \cdots & & \\ \cdots \underbrace{-1}_{i,j-1} & 0 \cdots \underbrace{-1}_{i-1,j} & \underbrace{4}_{i,j} & \underbrace{-1}_{i+1,j} \cdots 0 & \underbrace{-1}_{i,j+1} \cdots \\ & & & & \cdots & & \end{bmatrix}$$



$$\Rightarrow \qquad A\mathbf{u} = \mathbf{f}$$

Spectral properties of discretized operators in 2D

M: "mass" matrix, discretization of 0-order operator

A: "diffusion" matrix, discretizazion of self-ajdoint 2nd-order operator

- Finite Differences: n nodes each direction, $A \in \mathbb{R}^{n^2 \times n^2}$, $h = \frac{1}{n-1}$

$$M = I, \qquad \kappa(M) = 1$$

A such that $\quad ch^2 \leq \lambda_i(A) \leq C$, $\kappa(A) = O(\frac{1}{h^2})$ (c, C constants)

- Finite Elements:

M such that $\quad ch^2 \leq \lambda_i(M) \leq Ch^2$, $\kappa(M) = C/c$ (c, C constants)

A such that $\quad ch \leq \lambda_i(A) \leq \frac{1}{h}C$, $\kappa(A) = O(\frac{1}{h^2})$ (c, C constants)

Finite Differences: n nodes each direction, $A \in \mathbb{R}^{n^2 \times n^2}$, $h = \frac{1}{n-1}$

n	$\lambda_{\min}$	$\lambda_{\max}$	κ
10	1.6203e-01	7.8380e+00	4.8374e+01
20	4.4677e-02	7.9553e+00	1.7806e+02
30	2.0523e-02	7.9795e+00	3.8881e+02
40	1.1737e-02	7.9883e+00	6.8062e+02
50	7.5867e-03	7.9924e+00	1.0535e+03
60	5.3036e-03	7.9947e+00	1.5074e+03
70	3.9151e-03	7.9961e+00	2.0424e+03

Structured and Sparse matrices

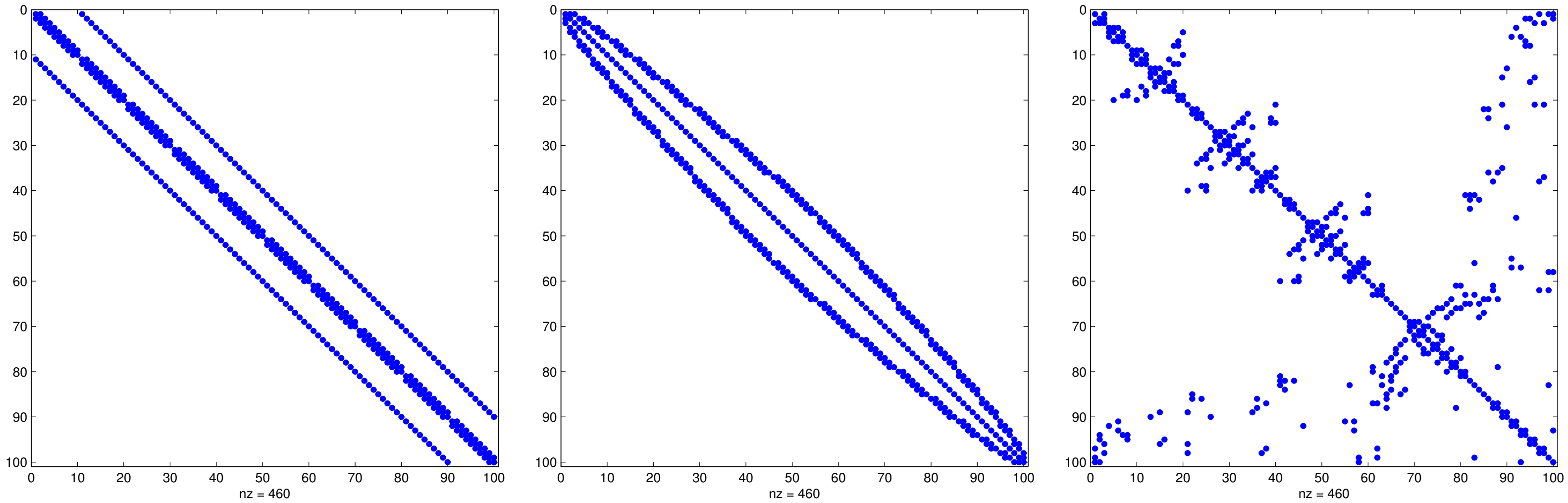
Finite Difference/Element discretization of 1D operator: banded matrices

⇒ Exploiting banded structure with banded solvers

However: higher degree operators and general domains determine matrices with different structure ⇒ Sparse matrices

Sparse matrices. I

Matrices stemming from discretizations have special pattern:



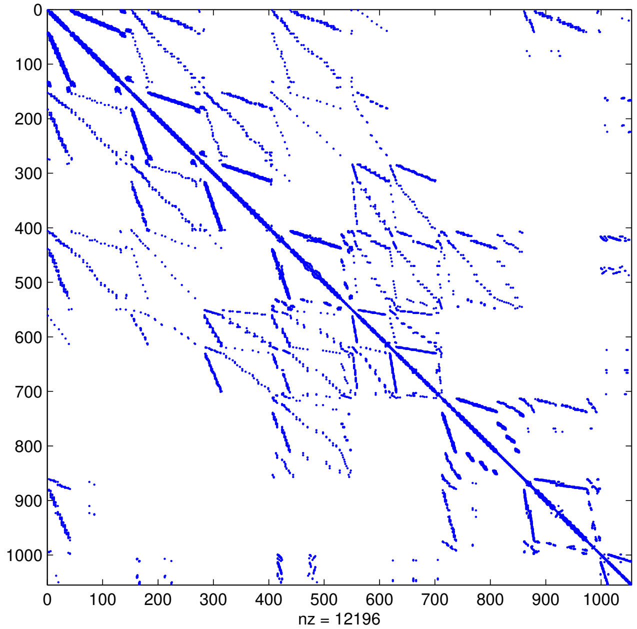
Same matrix, different ordering of the unknowns

large dimensions, only low percentage of nonzero elements per row

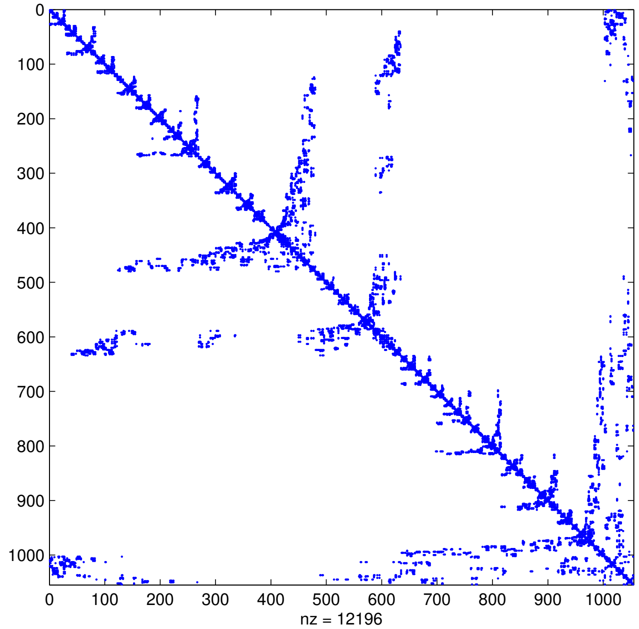
Sparse matrices. An Example

Matrix market. matrix CAN_1072 (structure problem in aircraft design)

Original sparsity pattern



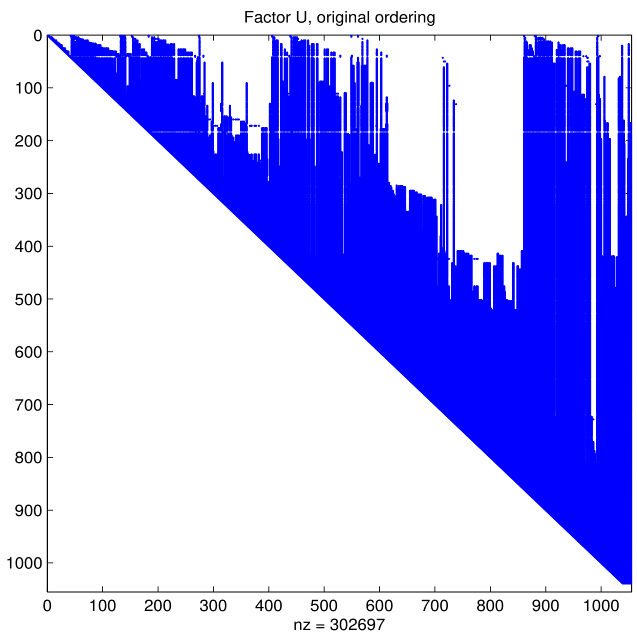
`symamd` reordering



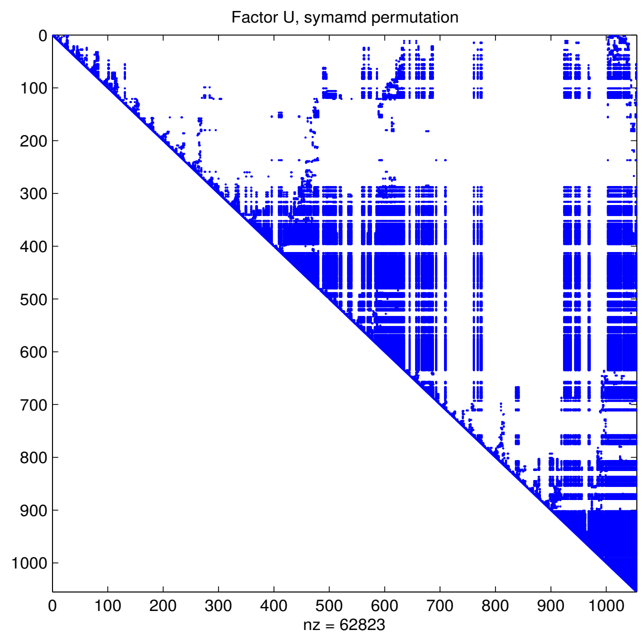
Sparse matrices. An Example

Factor U in LU factorization $A = LU$:

A with original sparsity pattern



A with **`symamd`** reordering



Solution methods for large matrices

Discretization of 2D and 3D problems leads to large matrices A

(size $O(10^k)$, $k = 5 - 8$)

$\Rightarrow$ (Optimized) LU decomposition too expensive

- Iterative methods: Projection-type methods (*)
- Geometric multigrid methods
- Algebraic multigrid methods
- Problem-related optimized methods

Discretization and linear system solves

A symmetric and positive definite.

CG: Number of iterations k depends on $\text{cond}(A) := \frac{\lambda_{\max}(A)}{\lambda_{\min}(A)}$

A 2D Poisson operator:

number of nodes per dimension	cond(A)	# its tol $= 10^{-10}$
2^3	32.16	10
2^4	116.46	31
2^5	440.69	66
2^6	1711.17	132

Stopping criterion: $r_k := b - Ax_k$ small enough in some norm

Preconditioning techniques

Determine matrix P such that

$$(PA)x = Pb$$

is "easier" to solve than $Ax = b$, that is

- Takes less CPU time
- P is cheap to construct
- P is reasonably cheap to apply

Note: Typically, P used in operators such as $y \leftarrow Pv$

Preconditioning techniques

Determine matrix P such that

$$(PA)x = Pb$$

is "easier" to solve than $Ax = b$, that is

- Takes less CPU time
- P is cheap to construct
- P is reasonably cheap to apply

Note: Typically, P used in operators such as $y \leftarrow Pv$

Choice criteria :

- P s.t. $PA \approx \alpha I$, with I identity matrix

Preconditioning techniques

Determine matrix P such that

$$(PA)x = Pb$$

is "easier" to solve than $Ax = b$, that is

- Takes less CPU time
- P is cheap to construct
- P is reasonably cheap to apply

Note: Typically, P used in operators such as $y \leftarrow Pv$

Choice criteria :

- P s.t. $PA \approx \alpha I$, with I identity matrix
- P s.t. P spectral properties similar to those of A^{-1}

Preconditioning techniques

Determine matrix P such that

$$(PA)x = Pb$$

is "easier" to solve than $Ax = b$, that is

- Takes less CPU time
- P is cheap to construct
- P is reasonably cheap to apply

Note: Typically, P used in operators such as $y \leftarrow Pv$

Choice criteria :

- P s.t. $PA \approx \alpha I$, with I identity matrix
- P s.t. P spectral properties similar to those of A^{-1}
- P "mimicks" the operator behind A
- ...

Preconditioning. 2

$$(PA)x = Pb$$

Classical strategy:

Determine P as $P = \mathcal{P}^{-1}$ con $\mathcal{P} \approx A$

$$\mathcal{P}^{-1}Ax = \mathcal{P}^{-1}b$$

Preconditioning. 2

$$(PA)x = Pb$$

Classical strategy:

Determine P as $P = \mathcal{P}^{-1}$ con $\mathcal{P} \approx A$

$$\mathcal{P}^{-1}Ax = \mathcal{P}^{-1}b$$

hoping that:

$\Rightarrow$ $\mathcal{P} \approx A$ then $\mathcal{P}^{-1} \approx A^{-1}$ so that $\mathcal{P}^{-1}A \approx I$

$\Rightarrow$ $\mathcal{P}^{-1}$ cheap to apply (via $y \leftarrow \mathcal{P}^{-1}v$), that is, solving

$$\mathcal{P}y = v$$

is far less expensive than $Ax = b$

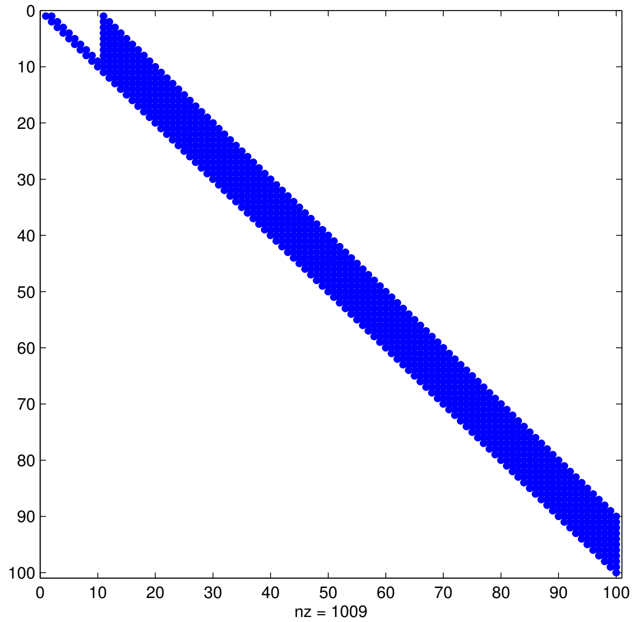
$\star$ Example: $\mathcal{P} = \mathrm{diag}(A)$: cheap, but little effective....

An example: Cholesky incomplete decomposition

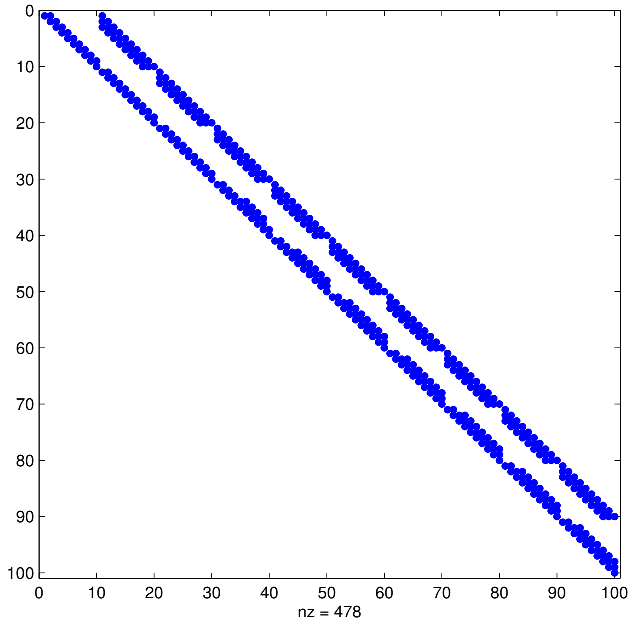
A sym.pos.def. $\qquad A = LL^T \approx L_0 L_0^T$

L_0 obtained from L by threshold chopping (element values below `tol` zeroed out)

L Original



approximation L_0



A corresponds to the Poisson operator, and $\texttt{tol} = 10^{-2}$

A possible strategy for incomplete LU

(ILUT, Algorithm 10.6, Saad)

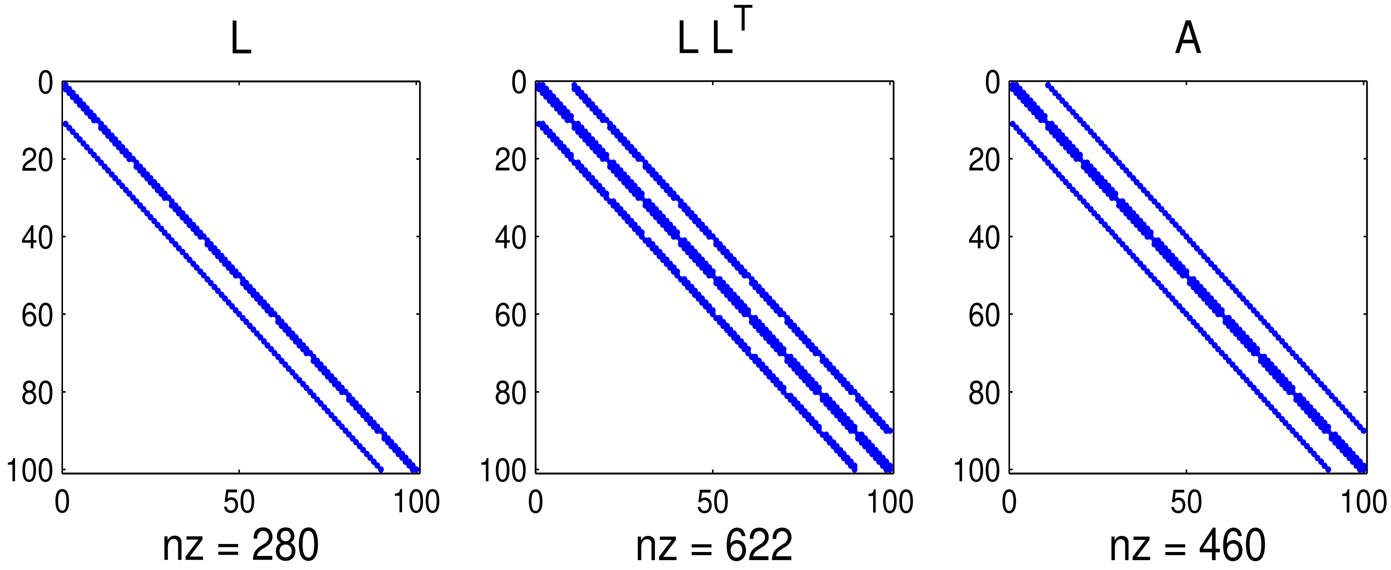
A $n \times n$, "threshold dropping" strategy

```
1.  for i = 1...n do
2.      w = a_{i,:} (with w = (w_1, ..., w_n))
3.      for k = 1....i-1 and w_k ≠ 0 do
4.          w_k := w_k / a_{k,k}
5.          Apply the ``dropping rule'' to w_k
6.          If w_k ≠ 0, w := w - w_k u_{k,:},end
7.      endfor
8.      Apply the ``dropping rule'' to the row w
9.      l_{i,1:i-1} = w_{1:i-1}, u_{i,i:n} = w_{i:n}
10. endfor
```

zero threshold: ILU(0) and CHOLINC(0)

$A \approx LU$ such that L and U have the same sparsity pattern as A

(`nnz(L + U - speye(size(A))) = nnz(A)`)



...also other strategies...

THEOREM. If A is a P-matrix, then there exists an incomplete factorization of A with fixed zero sparsity pattern, such that $A = LU - R$ with LU non-singular

PCG, maintaing symmetry

For A sym pos.def., $A \approx P = LL^T$. The preconditioned problem:

$$Ax = b \quad \Rightarrow \quad \underbrace{L^{-1}AL^{-T}}_{\widetilde{A}}\underbrace{L^Tx}_{\tilde{x}} = \underbrace{L^{-1}b}_{\tilde{b}},$$

PCG, maintaing symmetry

For A sym pos.def., $A \approx P = LL^T$. The preconditioned problem:

$$Ax = b \quad \Rightarrow \quad \underbrace{L^{-1}AL^{-T}}_{\widetilde{A}}\underbrace{L^{T}x}_{\tilde{x}} = \underbrace{L^{-1}b}_{\tilde{b}},$$

For $\tilde{p}^{(0)} = \tilde{r}^{(0)} = \tilde{b} - \widetilde{A}\tilde{x}^{(0)} = L^{-1}(b - Ax^{(0)}) = L^{-1}r^{(0)}$, we have

$$\tilde{x}^{(j+1)} = \tilde{x}^{(j)} + \alpha_j \tilde{p}^{(j)}, \text{ with} \qquad \alpha_j = \frac{(\tilde{r}^{(j)}, \tilde{r}^{(j)})}{(\widetilde{A}\tilde{p}^{(j)}, \tilde{p}^{(j)})}$$

$$\tilde{r}^{(j+1)} = \tilde{r}^{(j)} - \alpha_j \widetilde{A}\tilde{p}^{(j)}$$

$$\tilde{p}^{(j+1)} = \tilde{r}^{(j+1)} + \beta_j \tilde{p}^{(j)}, \text{ con} \qquad \beta_j = \frac{(\tilde{r}^{(j+1)}, \tilde{r}^{(j+1)})}{(\tilde{r}^{(j)}, \tilde{r}^{(j)})}$$

PCG, maintaing symmetry

For A sym pos.def., $A \approx P = LL^T$. The preconditioned problem:

$$Ax = b \quad \Rightarrow \quad \underbrace{L^{-1}AL^{-T}}_{\widetilde{A}}\underbrace{L^T x}_{\tilde{x}} = \underbrace{L^{-1}b}_{\tilde{b}},$$

For $\tilde{p}^{(0)} = \tilde{r}^{(0)} = \tilde{b} - \widetilde{A}\tilde{x}^{(0)} = L^{-1}(b - Ax^{(0)}) = L^{-1}r^{(0)}$, we have

$\tilde{x}^{(j+1)} = \tilde{x}^{(j)} + \alpha_j \tilde{p}^{(j)}$, with $\qquad \alpha_j = \frac{(\tilde{r}^{(j)},\tilde{r}^{(j)})}{(\widetilde{A}\tilde{p}^{(j)},\tilde{p}^{(j)})}$

$L^T x^{(j+1)} = L^T x^{(j)} + \alpha_j L^{-1} p^{(j)}$, with $\qquad \alpha_j = \frac{(L^{-1}r^{(j)},L^{-1}r^{(j)})}{(L^{-1}AL^{-T}L^{-1}p^{(j)},L^{-1}p^{(j)})}$

$\tilde{r}^{(j+1)} = \tilde{r}^{(j)} - \alpha_j \widetilde{A}\tilde{p}^{(j)}$

$L^{-1}r^{(j+1)} = L^{-1}r^{(j)} - \alpha_j L^{-1}AL^{-T}L^{-1}p^{(j)}$

$\tilde{p}^{(j+1)} = \tilde{r}^{(j+1)} + \beta_j \tilde{p}^{(j)}$, with $\qquad \beta_j = \frac{(\tilde{r}^{(j+1)},\tilde{r}^{(j+1)})}{(\tilde{r}^{(j)},\tilde{r}^{(j)})}$

$L^{-1}p^{(j+1)} = L^{-1}r^{(j+1)} + \beta_j L^{-1}p^{(j)}$, with $\qquad \beta_j = \frac{(L^{-1}r^{(j+1)},L^{-1}r^{(j+1)})}{(L^{-1}r^{(j)},L^{-1}r^{(j)})}$

PCG, maintaing symmetry

For A sym pos.def., $A \approx P = LL^T$. The preconditioned problem:

$$Ax = b \quad \Rightarrow \quad \underbrace{L^{-1}AL^{-T}}_{\widetilde{A}}\underbrace{L^T x}_{\tilde{x}} = \underbrace{L^{-1}b}_{\tilde{b}},$$

For $\tilde{p}^{(0)} = \tilde{r}^{(0)} = \tilde{b} - \widetilde{A}\tilde{x}^{(0)} = L^{-1}(b - Ax^{(0)}) = L^{-1}r^{(0)}$, we have

$$\tilde{x}^{(j+1)} = \tilde{x}^{(j)} + \alpha_j \tilde{p}^{(j)}, \text{ with} \qquad \alpha_j = \frac{(\tilde{r}^{(j)}, \tilde{r}^{(j)})}{(\widetilde{A}\tilde{p}^{(j)}, \tilde{p}^{(j)})}$$

$$x^{(j+1)} = x^{(j)} + \alpha_j L^{-T}L^{-1}p^{(j)}, \text{ with} \qquad \alpha_j = \frac{(r^{(j)}, L^{-T}L^{-1}r^{(j)})}{(AL^{-T}L^{-1}p^{(j)}, L^{-T}L^{-1}p^{(j)})}$$

$$\tilde{r}^{(j+1)} = \tilde{r}^{(j)} - \alpha_j \widetilde{A}\tilde{p}^{(j)}$$

$$r^{(j+1)} = r^{(j)} - \alpha_j AL^{-T}L^{-1}p^{(j)}$$

$$\tilde{p}^{(j+1)} = \tilde{r}^{(j+1)} + \beta_j \tilde{p}^{(j)}, \text{ with} \qquad \beta_j = \frac{(\tilde{r}^{(j+1)}, \tilde{r}^{(j+1)})}{(\tilde{r}^{(j)}, \tilde{p}^{(j)})}$$

$$L^{-T}L^{-1}p^{(j+1)} = L^{-T}L^{-1}r^{(j+1)} + \beta_j L^{-T}L^{-1}p^{(j)}, \text{ with} \quad \beta_j = \frac{(r^{(j+1)}, L^{-T}L^{-1}r^{(j+1)})}{(r^{(j)}, L^{-T}L^{-1}r^{(j)})}$$

PCG, maintaing symmetry

For A sym pos.def., $A \approx P = LL^T$. The preconditioned problem:

$$Ax = b \quad \Rightarrow \quad \underbrace{L^{-1}AL^{-T}}_{\widetilde{A}}\underbrace{L^T x}_{\tilde{x}} = \underbrace{L^{-1}b}_{\tilde{b}},$$

For $\tilde{p}^{(0)} = \tilde{r}^{(0)} = \tilde{b} - \widetilde{A}\tilde{x}^{(0)} = L^{-1}(b - Ax^{(0)}) = L^{-1}r^{(0)}$, we have

With $\hat{p}^{(0)} = L^{-T}L^{-1}p^{(0)} = P^{-1}p^{(0)}$ and $z^{(j)} = L^{-T}L^{-1}r^{(j)} = P^{-1}r^{(j)}$:

$$x^{(j+1)} = x^{(j)} + \alpha_j \hat{p}^{(j)} \text{ with } \qquad \alpha_j = \frac{(r^{(j)}, z^{(j)})}{(A\hat{p}^{(j)}, \hat{p}^{(j)})}$$

$$r^{(j+1)} = r^{(j)} - \alpha_j A\hat{p}^{(j)}$$

$$\hat{p}^{(j+1)} = z^{(j+1)} + \beta_j \hat{p}^{(j)}, \text{ with } \beta_j = \frac{(r^{(j+1)}, z^{(j+1)})}{(r^{(j)}, z^{(j)})}$$

Practical preconditioning strategies

- LU-type approx decomposition of A: $\rightarrow Pv = U^{-1}L^{-1}v$
- Algebraic multigrid (approximate representation of A on smaller version of the matrix - recursive procedure)
- Geometric multigrid (operator and domain dependent)
- Functional approximation of the underlying operator

A comparison :
Incomplete Cholesky and Algebraic Multigrid

Poisson, 2D problem on $[0,1]^2$. Matrices of dim $n = 2^k \times 2^k$

grid	incomplete Chol		AMG	
nodes per dim	# it's	CPU time	# it's	CPU time
2^4	11	0.008	6	0.18
2^5	18	0.007	6	0.20
2^6	33	0.04	7	0.22
2^7	58	0.29	7	0.32
2^8	106	2.27	8	0.71

For 2^8, $\dim(A) = 65536 \times 65536$

!! Preconditioned CG with AMG gives grid independent # it's !!

A comparison :
Incomplete Cholesky and Algebraic Multigrid

Poisson, 2D problem on $[0,1]^2$. Matrices of dim $n = 2^k \times 2^k$

grid	incomplete Chol		AMG	
nodes per dim	# it's	CPU time	# it's	CPU time
2^4	11	0.008	6	0.18
2^5	18	0.007	6	0.20
2^6	33	0.04	7	0.22
2^7	58	0.29	7	0.32
2^8	106	2.27	8	0.71

For 2^8, $\dim(A) = 65536 \times 65536$

!! Preconditioned CG with AMG gives grid independent # it's !!

Remark: For 2^8, `tic;`$A\backslash b$`;toc`, gives: `Elapsed time is 0.588393 seconds.`

Algebraic Multigrid (AMG)

Consider the original system

$$A_h u^h = f^h \qquad (\star)$$

The error vector is split in two parts: an *oscillatory* component (high freq.) and a *regular* component (smooth, low freq.)

A Multigrid (or multilevel) type method for a linear system is made of two ingredients:

- A smoothing step of the oscillatory portion:

  usually a few iterations of a classical method (e.g., Jacobi, Gauss-Seidel)

- A correction on a coarser grid for the smooth part

  The system $(\star)$ is approximated by a system on a coarser grid: A^H, f^H such that

$$A_H = I_h^H A_h I_H^h, \qquad f^H = I_h^H f^h$$

Conceptually similar to a Galerkin projection type procedure:

I_h^H: restriction operator, full rank

I_H^h: prolongation operator, rull rank

with

$$I_h^H = (I_H^h)^T \qquad \text{(transposition)}$$

Remark: *Geometric* Multigrid uses the physical grid. *Algebraic* Multigrid use the matrix elements

(matrix indexes $\equiv$ grid nodes)

Algebraic Multigrid (AMG)

General procedure (on two grids):

1. Perform n_1 steps of smoothing (e.g., Jacobi) on $A_h u^h = f^h$
2. Compute the residual $r^h = f^h - A_h u^h \equiv A e^h$
3. Project (restrict) to the coarse grid $r^H = I_h^H r^h$
4. Solve on coarse grid: $A_H e^H = r^H$
5. Add (prolong) $u^h := u^h + I_H^h e^H$
6. Take n_2 steps of smoothing on $A_h u^h = f^h$

Algebraic Multigrid (AMG). The coarse grid

Determine A_H from A_h, A_H is a subset of the rows/columns of A_h

(strong connection among the elements of A_H)

DEF. Let $\theta \in (0,1]$ be a fixed threshold. The variable u_i *strongly* depends on the variable u_j if

$$-a_{ij} \geq \theta \max_{k \neq i} \{-a_{ik}\}$$

$\Rightarrow$ non-diagonal positive elements have a weak connection

The following steps should be taken (where: node= pair of indexes)

1. Define a "strength" matrix (A_f) by eliminating the weak connections
2. Choose an independent set of strong nodes of A_f
3. Add possible nodes to have a correct proloungation operator

Spectral equivalence

Under particular conditions[a] on the matrix A, it can be proved that the AMG preconditioner is spectrally equivalent to A, that is:

There exist $\alpha_1, \alpha_2 > 0$ independent of the dimension of A such that

$$\alpha_1(x, Px) \leq (x, Ax) \leq \alpha_2(x, Px), \qquad \forall x \neq 0$$

[a] e.g., if A is Hpd is an M-matrix, that is with $a_{ii} > 0 \; \forall i$ and $a_{ij} \leq 0 \; \forall i \neq j$, with non-negative inverse - the usual discretization of the Laplacian.

Spectral equivalence

Under particular conditions[a] on the matrix A, it can be proved that the AMG preconditioner is spectrally equivalent to A, that is:

There exist $\alpha_1, \alpha_2 > 0$ independent of the dimension of A such that

$$\alpha_1 (x, Px) \leq (x, Ax) \leq \alpha_2 (x, Px), \qquad \forall x \neq 0$$

In our context:

$$P^{-1} A v = \lambda v \qquad \Leftrightarrow \qquad Av = \lambda P v$$

so that

$$\lambda = \frac{(v, Av)}{(v, Pv)}, \qquad \min_{x \neq 0} \frac{(x, Ax)}{(x, Px)} \leq \lambda \leq \max_{x \neq 0} \frac{(x, Ax)}{(x, Px)}$$

$\Rightarrow$ The spectral interval of the preconditioned problems **does not** depend on the problem dimension (or on the grid!)

[a] e.g., if A is Hpd is an M-matrix, that is with $a_{ii} > 0 \ \forall i$ and $a_{ij} \leq 0 \ \forall i \neq j$, with non-negative inverse - the usual discretization of the Laplacian.

Saddle point linear systems

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

- Computational Fluid Dynamics (Elman, Silvester, Wathen 2005)
- Elasticity problems
- Mixed (FE) formulations of II and IV order elliptic PDEs
- Linearly Constrained Programs
- Linear Regression in Statistics
- Image restoration
- ... Survey: Benzi, Golub and Liesen, Acta Num 2005

The problem. Simplifications

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

To make things simple:

- ⋆ A symmetric positive (semi)definite
- ⋆ B^T tall, possibly rank deficient
- ⋆ C symmetric positive (semi)definite

Spectral properties

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & O \end{bmatrix} \qquad \begin{array}{ll} 0 < \lambda_n \le \cdots \le \lambda_1 & \text{eigs of } A \\ 0 < \sigma_m \le \cdots \le \sigma_1 & \text{sing. vals of } B \end{array}$$

$\sigma(\mathcal{A})$ subset of (Rusten & Winther 1992)

$$\left[\frac{1}{2}(\lambda_n - \sqrt{\lambda_n^2 + 4\sigma_1^2}), \frac{1}{2}(\lambda_1 - \sqrt{\lambda_1^2 + 4\sigma_m{}^2})\right] \quad \cup \quad \left[\lambda_n, \frac{1}{2}(\lambda_1 + \sqrt{\lambda_1^2 + 4\sigma_1^2})\right]$$

A nonsingular , B full rank

(other hypotheses are possible)

Spectral properties

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & O \end{bmatrix} \qquad \begin{array}{ll} 0 < \lambda_n \leq \cdots \leq \lambda_1 & \text{eigs of } A \\ 0 < \sigma_m \leq \cdots \leq \sigma_1 & \text{sing. vals of } B \end{array}$$

$\sigma(\mathcal{A})$ subset of (Rusten & Winther 1992)

$$\left[\frac{1}{2}(\lambda_n - \sqrt{\lambda_n^2 + 4\sigma_1^2}), \frac{1}{2}(\lambda_1 - \sqrt{\lambda_1^2 + 4\sigma_m^2})\right] \quad \cup \quad \left[\lambda_n, \frac{1}{2}(\lambda_1 + \sqrt{\lambda_1^2 + 4\sigma_1^2})\right]$$

Good (= slim) spectrum: $\lambda_1 \approx \lambda_n, \quad \sigma_1 \approx \sigma_m$

Spectral properties

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & O \end{bmatrix} \qquad \begin{array}{ll} 0 < \lambda_n \le \cdots \le \lambda_1 & \text{eigs of } A \\ 0 < \sigma_m \le \cdots \le \sigma_1 & \text{sing. vals of } B \end{array}$$

$\sigma(\mathcal{A})$ subset of (Rusten & Winther 1992)

$$\left[\frac{1}{2}(\lambda_n - \sqrt{\lambda_n^2 + 4\sigma_1^2}), \frac{1}{2}(\lambda_1 - \sqrt{\lambda_1^2 + 4\sigma_m^2})\right] \quad \cup \quad \left[\lambda_n, \frac{1}{2}(\lambda_1 + \sqrt{\lambda_1^2 + 4\sigma_1^2})\right]$$

Good (= slim) spectrum: $\lambda_1 \approx \lambda_n, \quad \sigma_1 \approx \sigma_m$

EXAMPLE:

$$\mathcal{A} = \begin{bmatrix} I & U^T \\ U & O \end{bmatrix}, \quad UU^T = I, \qquad \lambda_i(I) = 1\,\forall i, \sigma_j(U) = 1, \forall j$$

$$\sigma(A) \subset \{\frac{1}{2}(1 - \sqrt{5})\} \cup [1, \frac{1}{2}(1 + \sqrt{5})]$$

Which method for this problem?

$\mathcal{A}$ is symmetric **but** indefinite!

$\Rightarrow$ CG will not work...

$\Rightarrow$ GMRES? it is for nonsymmetric problems... however, we said:

$$\boxed{\text{If } \mathcal{A} \text{ were Hpd} \quad \Rightarrow \quad V_k^* \mathcal{A} V_k \text{ also Hpd} \quad \Rightarrow \quad \text{tridiagonal}}$$

Which method for this problem?

$\mathcal{A}$ is symmetric **but** indefinite!

$\Rightarrow$ CG will not work...

$\Rightarrow$ GMRES? it is for nonsymmetric problems... however, we said:

$$\boxed{\text{If } \mathcal{A} \text{ were Hpd} \quad \Rightarrow \quad V_k^* \mathcal{A} V_k \text{ also Hpd} \quad \Rightarrow \quad \text{tridiagonal}}$$

This implies (details omitted) that

$$\min_y \|r_0 - AV_k y\| \quad \Leftrightarrow \quad \min_y \|e_1 \beta_0 - \underline{H}_k y\|$$

with $\underline{H}_k = V_{k+1}^* \mathcal{A} V_k$ tridiagonal, so that

$$x_{k+1} = x_k + q_k \eta_k$$

(for some q_k, η_k) short-term recurrence, MINRES

Block diagonal Preconditioner

⋆ A nonsing., $C = 0$:

$$\mathcal{P}_0 = \begin{bmatrix} A & 0 \\ 0 & BA^{-1}B^T \end{bmatrix}$$

$$\Rightarrow \qquad \mathcal{P}_0^{-\frac{1}{2}} \mathcal{A} \mathcal{P}_0^{-\frac{1}{2}} = \begin{bmatrix} I & A^{-\frac{1}{2}} B^T (BA^{-1}B^T)^{-\frac{1}{2}} \\ (BA^{-1}B^T)^{-\frac{1}{2}} BA^{-\frac{1}{2}} & 0 \end{bmatrix}$$

MINRES converges in at most 3 iterations. $\sigma(\mathcal{P}_0^{-\frac{1}{2}} \mathcal{A} \mathcal{P}_0^{-\frac{1}{2}}) = \{1, \frac{1}{2} \pm \frac{\sqrt{5}}{2}\}$

Block diagonal Preconditioner

⋆ A nonsing., $C = 0$:

$$\mathcal{P}_0 = \begin{bmatrix} A & 0 \\ 0 & BA^{-1}B^T \end{bmatrix}$$

$$\Rightarrow \qquad \mathcal{P}_0^{-\frac{1}{2}} \mathcal{A} \mathcal{P}_0^{-\frac{1}{2}} = \begin{bmatrix} I & A^{-\frac{1}{2}} B^T (BA^{-1}B^T)^{-\frac{1}{2}} \\ (BA^{-1}B^T)^{-\frac{1}{2}} BA^{-\frac{1}{2}} & 0 \end{bmatrix}$$

MINRES converges in at most 3 iterations. $\sigma(\mathcal{P}_0^{-\frac{1}{2}} \mathcal{A} \mathcal{P}_0^{-\frac{1}{2}}) = \{1, \frac{1}{2} \pm \frac{\sqrt{5}}{2}\}$

A more practical choice:

$$\mathcal{P} = \begin{bmatrix} \widetilde{A} & 0 \\ 0 & \widetilde{S} \end{bmatrix} \qquad \text{spd.} \quad \widetilde{A} \approx A \qquad \widetilde{S} \approx BA^{-1}B^T$$

eigs in $\quad [-a, -b] \cup [c, d], \qquad a, b, c, d > 0$

Still an Indefinite Problem $\Rightarrow$ MINRES

Giving up symmetry ...

- Change the preconditioner: *Mimic the LU factors*

$$\mathcal{A} = \begin{bmatrix} I & O \\ BA^{-1} & I \end{bmatrix} \begin{bmatrix} A & B^T \\ O & BA^{-1}B^T + C \end{bmatrix} \quad \Rightarrow \mathcal{P} \approx \begin{bmatrix} A & B^T \\ O & BA^{-1}B^T + C \end{bmatrix}$$

- Change the preconditioner: *Mimic the Structure*

$$\mathcal{A} = \begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \quad \Rightarrow \mathcal{P} \approx \mathcal{A}$$

- Change the matrix: *Eliminate indef.* $\quad \mathcal{A}_- = \begin{bmatrix} A & B^T \\ -B & C \end{bmatrix}$

- Change the matrix: *Regularize* $(C = 0)$

$$\mathcal{A} \ \Rightarrow \ \mathcal{A}_\gamma = \begin{bmatrix} A & B^T \\ B & -\gamma W \end{bmatrix} \text{ or } \mathcal{A}_\gamma = \begin{bmatrix} A + \frac{1}{\gamma} B^T W^{-1} B & B^T \\ B & O \end{bmatrix}$$

Application of the preconditioners. 1

At each iteration of CG, MINRES or GMRES, compute $y = \mathcal{P}^{-1} z$, that is solve

$$\mathcal{P} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

$\mathcal{P} = \begin{bmatrix} \widetilde{A} & 0 \\ 0 & \widetilde{S} \end{bmatrix}$ that is Solve $\widetilde{A} y_1 = z_1, \quad \widetilde{S} y_2 = z_2.$

$\mathcal{P} = \begin{bmatrix} \widetilde{A} & B^T \\ 0 & \widetilde{S} \end{bmatrix}$ that is Solve $\widetilde{S} y_2 = z_2, \quad \widetilde{A} y_1 = z_1 - B^T y_2.$

Application of the preconditioners. 2

Indefinite preconditioner:

$$\mathcal{P} = \begin{bmatrix} \widetilde{A} & B^T \\ B & -\widetilde{S} \end{bmatrix} = \begin{bmatrix} I & 0 \\ B\widetilde{A}^{-1} & I \end{bmatrix} \begin{bmatrix} \widetilde{A} & 0 \\ 0 & -\widehat{S} \end{bmatrix} \begin{bmatrix} I & \widetilde{A}^{-}B^T \\ 0 & I \end{bmatrix} = \mathcal{P}_1 \mathcal{D} \mathcal{P}_2$$

with $\widehat{S} = \widetilde{S} + B\widetilde{A}^{-}B^T$

(In practice $\widehat{S}$ is an approximation to this quantity)

Application of the indefinite preconditioner:

$$\mathcal{P}y = z \quad \Leftrightarrow \quad \mathcal{P}_1 \underbrace{\mathcal{D}\, \underbrace{\mathcal{P}_2 y}_{=z_1}}_{=z_2} = z$$