Resolution of sparse linear systems of equations: the RPK strategy

G. Montero

in collaboration with R. Montenegro, J.M. Escobar and E. Rodríguez

University Institute of Intelligent Systems and Numerical Applications in Engineering
University of Las Palmas de Gran Canaria

Partially Supported by MCYT and FEDER (Grant number REN2001-0925-C03-02/CLI)

September, 2004

The problem

To solve a linear system of equations

$$\mathbf{Ax} = \mathbf{b}$$

where matrix **A** is large, sparse and non singular (symmetric or nonsymmetric).

Direct or iterative solvers?

- Rounding errors affect to direct methods
- More memory requirement due to the fill-in related to the factorization
- In time-dependent problems, iterative methods can take advantage of the solution obtained in the previous time step, using it as initial guess

The problem

To solve a linear system of equations

$$\mathbf{Ax} = \mathbf{b}$$

where matrix $\mathbf{A}$ is large, sparse and non singular (symmetric or nonsymmetric).

Direct or iterative solvers?

- Rounding errors affect to direct methods
- More memory requirement due to the fill-in related to the factorization
- In time-dependent problems, iterative methods can take advantage of the solution obtained in the previous time step, using it as initial guess

The problem

To solve a linear system of equations

$$\mathbf{A}\mathbf{x} = \mathbf{b}$$

where matrix **A** is large, sparse and non singular (symmetric or nonsymmetric).

Direct or iterative solvers?

- Rounding errors affect to direct methods
- More memory requirement due to the fill-in related to the factorization
- In time-dependent problems, iterative methods can take advantage of the solution obtained in the previous time step, using it as initial guess

The problem

To solve a linear system of equations

$$\mathbf{Ax} = \mathbf{b}$$

where matrix **A** is large, sparse and non singular (symmetric or nonsymmetric).

Direct or iterative solvers?

- Rounding errors affect to direct methods
- More memory requirement due to the fill-in related to the factorization
- In time-dependent problems, iterative methods can take advantage of the solution obtained in the previous time step, using it as initial guess

Reordering techniques

Reordering

- Based on Graph Theory
- Firstly, they were applied with direct solvers
  - ⋆ Reduce the fill-in ⇒ less memory requirement and computational cost
- Nowadays, they are used with iterative solvers
  - ⋆ Improve the effect of preconditioning on the convergence

I.S. Duff and G.A. Meurant, L. Dutto, M. Benzi et al, E. Flórez et al

References

Based on the location of the entries

- ⋆ Reverse Cuthill McKee
- ⋆ Minimum Degree
- ⋆ Minimum Neighbouring
- ⋆ Minimum Fill-in
- ⋆ Spiral
- ⋆ Red black
- ⋆ Nested dissection

Based also on the magnitude of the entries

- ⋆ Minimum discarded fill (E.F. D'Azevedo et al)
- ⋆ TPABLO (H. Choi and D.B. Szyld)
- ⋆ Simulated Annealing (R.R. Lewis)
- ⋆ Genetic Algorithms (D. Greiner et al)

Reordering techniques

Reordering

- Based on Graph Theory
- Firstly, they were applied with direct solvers
  - ⋆ Reduce the fill-in ⇒ less memory requirement and computational cost
- Nowadays, they are used with iterative solvers
  - ⋆ Improve the effect of preconditioning on the convergence

I.S. Duff and G.A. Meurant, L. Dutto, M. Benzi et al, E. Flórez et al

References

Based on the location of the entries

- ⋆ Reverse Cuthill McKee
- ⋆ Minimum Degree
- ⋆ Minimum Neighbouring
- ⋆ Minimum Fill-in
- ⋆ Spiral
- ⋆ Red black
- ⋆ Nested dissection

Based also on the magnitude of the entries

- ⋆ Minimum discarded fill (E.F. D'Azevedo et al)
- ⋆ TPABLO (H. Choi and D.B. Szyld)
- ⋆ Simulated Annealing (R.R. Lewis)
- ⋆ Genetic Algorithms (D. Greiner et al)

Reordering techniques

Reordering

- Based on Graph Theory
- Firstly, they were applied with direct solvers
  - ⋆ Reduce the fill-in ⇒ less memory requirement and computational cost
- Nowadays, they are used with iterative solvers
  - ⋆ Improve the effect of preconditioning on the convergence

I.S. Duff and G.A. Meurant, L. Dutto, M. Benzi et al, E. Flórez et al

References

Based on the location of the entries

- ⋆ Reverse Cuthill McKee
- ⋆ Minimum Degree
- ⋆ Minimum Neighbouring
- ⋆ Minimum Fill-in
- ⋆ Spiral
- ⋆ Red black
- ⋆ Nested dissection

Based also on the magnitude of the entries

- ⋆ Minimum discarded fill (E.F. D'Azevedo et al)
- ⋆ TPABLO (H. Choi and D.B. Szyld)
- ⋆ Simulated Annealing (R.R. Lewis)
- ⋆ Genetic Algorithms (D. Greiner et al)

Reordering techniques

Reordering

- Based on Graph Theory
- Firstly, they were applied with direct solvers
  - ⋆ Reduce the fill-in $\Rightarrow$ less memory requirement and computational cost
- Nowadays, they are used with iterative solvers
  - ⋆ Improve the effect of preconditioning on the convergence

I.S. Duff and G.A. Meurant, L. Dutto, M. Benzi et al, E. Flórez et al

References

Based on the location of the entries

- ⋆ Reverse Cuthill McKee
- ⋆ Minimum Degree
- ⋆ Minimum Neighbouring
- ⋆ Minimum Fill-in
- ⋆ Spiral
- ⋆ Red black
- ⋆ Nested dissection

Based also on the magnitude of the entries

- ⋆ Minimum discarded fill (E.F. D'Azevedo et al)
- ⋆ TPABLO (H. Choi and D.B. Szyld)
- ⋆ Simulated Annealing (R.R. Lewis)
- ⋆ Genetic Algorithms (D. Greiner et al)

Reordering techniques

Reordering

- Based on Graph Theory
- Firstly, they were applied with direct solvers
  - ⋆ Reduce the fill-in $\Rightarrow$ less memory requirement and computational cost
- Nowadays, they are used with iterative solvers
  - ⋆ Improve the effect of preconditioning on the convergence

I.S. Duff and G.A. Meurant, L. Dutto, M. Benzi et al, E. Flórez et al

References

Based on the location of the entries

- ⋆ Reverse Cuthill McKee
- ⋆ Minimum Degree
- ⋆ Minimum Neighbouring
- ⋆ Minimum Fill-in
- ⋆ Spiral
- ⋆ Red black
- ⋆ Nested dissection

Based also on the magnitude of the entries

- ⋆ Minimum discarded fill (E.F. D'Azevedo et al)
- ⋆ TPABLO (H. Choi and D.B. Szyld)
- ⋆ Simulated Annealing (R.R. Lewis)
- ⋆ Genetic Algorithms (D. Greiner et al)

Preconditioning techniques

Preconditioning patterns

- ⋆ Left $\mathbf{MAx} = \mathbf{Mb}$
- ⋆ Right $\mathbf{AMM}^{-1}\mathbf{x} = \mathbf{b}$
- ⋆ Both side $\mathbf{M}_1\mathbf{AM}_2\mathbf{M}_2^{-1}\mathbf{x} = \mathbf{M}_1\mathbf{b}$

Preconditioners

Implicit Preconditioners

- ⋆ SSOR
- ⋆ ILUT
- ⋆ ILU(m)

Explicit Preconditioners

- ⋆ Jacobi
- ⋆ Sparse Approximate Inverses
  - SPAI, Grote et al
  - AINV, Benzi et al
  - Generalization of SPAI (Montero et al)
- ⋆ Optimum Diagonal

Preconditioning techniques

Preconditioning patterns

- ⋆ Left $\mathbf{MAx} = \mathbf{Mb}$
- ⋆ Right $\mathbf{AMM}^{-1}\mathbf{x} = \mathbf{b}$
- ⋆ Both side $\mathbf{M}_1\mathbf{A}\mathbf{M}_2\mathbf{M}_2^{-1}\mathbf{x} = \mathbf{M}_1\mathbf{b}$

Preconditioners

Implicit Preconditioners

- ⋆ SSOR
- ⋆ ILUT
- ⋆ ILU(m)

Explicit Preconditioners

- ⋆ Jacobi
- ⋆ Sparse Approximate Inverses
  - SPAI, Grote et al
  - AINV, Benzi et al
  - Generalization of SPAI (Montero et al)
- ⋆ Optimum Diagonal

Preconditioning techniques

Preconditioning patterns

- ⋆ Left $\mathbf{MAx} = \mathbf{Mb}$
- ⋆ Right $\mathbf{AMM}^{-1}\mathbf{x} = \mathbf{b}$
- ⋆ Both side $\mathbf{M}_1\mathbf{AM}_2\mathbf{M}_2^{-1}\mathbf{x} = \mathbf{M}_1\mathbf{b}$

Preconditioners

Implicit Preconditioners

- ⋆ SSOR
- ⋆ ILUT
- ⋆ ILU(m)

Explicit Preconditioners

- ⋆ Jacobi
- ⋆ Sparse Approximate Inverses
  - SPAI, Grote et al
  - AINV, Benzi et al
  - Generalization of SPAI (Montero et al)
- ⋆ Optimum Diagonal

Krylov subspace methods

Orthogonalisation	Biorthogonalisation	Normal equation
ORTHOMIN (Vinsome '76)	BiCG (Fletcher '76)	CGN (Hestenes & Stiefel '52)
ORTHORES (Young & Jea '80)	CGS (Sonneveld '89)	CGNE (Craig '55)
ORTHODIR (Young & Jea '80)	BiCGSTAB (Van der Vorst '90)	LSQR (Paige & Saunders '82)
FOM (Saad '81)	QMR (Freund & Nachtigal '90)	
GMRES (Saad & Schultz '86)	TFQMR (Freund '93)	Multigrid method
FGMRES (Saad '93)	QMRCGSTAB (Chan et al '94)	Domain decomposition
VGMRES (Galán et al '94)		

Krylov subspace methods

Orthogonalisation	Biorthogonalisation	Normal equation
ORTHOMIN (Vinsome '76)	BiCG (Fletcher '76)	CGN (Hestenes & Stiefel '52)
ORTHORES (Young & Jea '80)	CGS (Sonneveld '89)	CGNE (Craig '55)
ORTHODIR (Young & Jea '80)	BiCGSTAB (Van der Vorst '90)	LSQR (Paige & Saunders '82)
FOM (Saad '81)	QMR (Freund & Nachtigal '90)	
GMRES (Saad & Schultz '86)	TFQMR (Freund '93)	Multigrid method
FGMRES (Saad '93)	QMRCGSTAB (Chan et al '94)	Domain decomposition
VGMRES (Galán et al '94)		

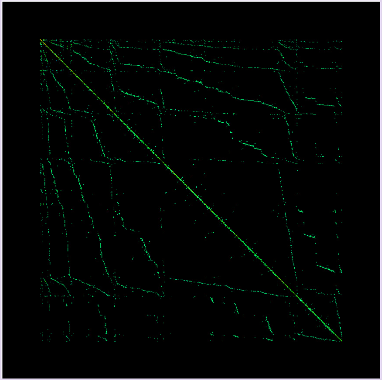
Krylov subspace methods

Orthogonalisation	Biorthogonalisation	Normal equation
ORTHOMIN (Vinsome '76)	BiCG (Fletcher '76)	CGN (Hestenes & Stiefel '52)
ORTHORES (Young & Jea '80)	CGS (Sonneveld '89)	CGNE (Craig '55)
ORTHODIR (Young & Jea '80)	BiCGSTAB (Van der Vorst '90)	LSQR (Paige & Saunders '82)
FOM (Saad '81)	QMR (Freund & Nachtigal '90)	
GMRES (Saad & Schultz '86)	TFQMR (Freund '93)	Multigrid method
FGMRES (Saad '93)	QMRCGSTAB (Chan et al '94)	Domain decomposition
VGMRES (Galán et al '94)		

Krylov subspace methods

Orthogonalisation	Biorthogonalisation	Normal equation
ORTHOMIN (Vinsome '76)	BiCG (Fletcher '76)	CGN (Hestenes & Stiefel '52)
ORTHORES (Young & Jea '80)	CGS (Sonneveld '89)	CGNE (Craig '55)
ORTHODIR (Young & Jea '80)	BiCGSTAB (Van der Vorst '90)	LSQR (Paige & Saunders '82)
FOM (Saad '81)	QMR (Freund & Nachtigal '90)	
GMRES (Saad & Schultz '86)	TFQMR (Freund '93)	Multigrid method
FGMRES (Saad '93)	QMRCGSTAB (Chan et al '94)	Domain decomposition
VGMRES (Galán et al '94)		

Sparsity pattern of a 7520 FEM convection-diffusion matrix with initial ordering



Pseudo-peripheral node searching algorithm

1. Choose any node r in V.
2. Generate a nested level structure in r, $\{L_0(r), L_1(r), \ldots, L_{\varepsilon(r)}(r)\}$. being $L_i(r) = \{x/d(x,r) = i\}$
3. Choose a minimum degree node x in $L_{\varepsilon(r)}(r)$.
4. Generate a nested level structure in x, $\{L_0(x), L_1(x), \ldots, L_{\varepsilon(x)}(x)\}$
5. If $\varepsilon(x) > \varepsilon(r)$, establish $x \to r$ and go to step 3
6. Otherwise, select x as initial node
7. End

$\star$ $d(x,y)$: distance between two nodes x and y in a graph
$\star$ $g(x) = \langle V, E \rangle$: length of the shortest trajectory that joins both nodes
$\star$ $\varepsilon(x) = max\,\{d(x,y)/x, y \in V\}$: eccentricity of node x

Reverse Cuthill-Mckee algorithm

1 - Build the graph associated to matrix $\mathbf{A}$, $g(x) = \langle V, E \rangle$, being V the set of nodes and $E = \{\{a, b\} : a \neq b \ / \ a, b \in V\}$

2 - Find an initial node (pseudo-peripheral) and order it as x_1

3 - Order the nodes connected to x_i in increasing degree order

4 - Carry out the inverse ordering.

5 - End

Reverse Cuthill-Mckee algorithm

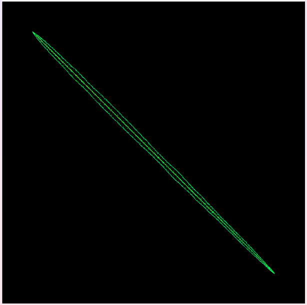
1 - Build the graph associated to matrix $\mathbf{A}$, $g(x) = \langle V, E \rangle$, being V the set of nodes and $E = \{\{a, b\} : a \neq b \ / \ a, b \in V\}$

2 - Find an initial node (pseudo-peripheral) and order it as x_1

3 - Order the nodes connected to x_i in increasing degree order

4 - Carry out the inverse ordering.

5 - End



Minimum Degree algorithm

1 - Build the graph associated to matrix **A**,
$g(x) = \langle V, E \rangle$, being V the set of nodes
and $E = \{\{a, b\} : a \neq b \ / \ a, b \in V\}$

2 - While $V \neq \emptyset$:

2.1- Choose a minimum degree node v in
$g(x) = \langle V, E \rangle$ and order it as next node

2.2 - Define:
$V_v = V - \{v\}$,
$E_v = \{\{a, b\} \in E \ / \ a, b \in V_v\} \cup$
$\{\{a, b\} \ a \neq b \ / \ a, b \in Adj_g(v)\}$,
being $Adj_g(v)$ the set of nodes connected
to v in the graph $g(x)$ and do
$V = V_v, \ \ E = E_v, \ g(x) = \langle V, E \rangle$

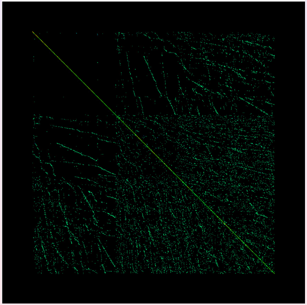
3 - End

Minimum Degree algorithm

1 - Build the graph associated to matrix **A**,
$g(x) = \langle V, E \rangle$, being V the set of nodes
and $E = \{\{a, b\} : a \neq b \ / \ a, b \in V\}$

2 - While $V \neq \emptyset$:
2.1- Choose a minimum degree node v in
$g(x) = \langle V, E \rangle$ and order it as next node
2.2 - Define:
$V_v = V - \{v\}$,
$E_v = \{\{a, b\} \in E \ / \ a, b \in V_v\} \cup$
$\{\{a, b\} \ a \neq b \ / \ a, b \in Adj_g(v)\}$,
being $Adj_g(v)$ the set of nodes connected
to v in the graph $g(x)$ and do
$V = V_v, \ \ E = E_v, \ g(x) = \langle V, E \rangle$

3 - End



Minimum Neighbouring algorithm

1 - Build the graph associated to matrix $\mathbf{A}$,
$g(x) = \langle V, E \rangle$, being V the set of nodes and
$E = \{\{a, b\} : a \neq b \ / \ a, b \in V\}$

2 - While $V \neq \emptyset$:

2.1- Choose a minimum degree node v in
$g(x) = \langle V, E \rangle$ and order it as next node

2.2 - Define:

$V_v = V - \{v\}$

$E_v = \{\{a, b\} \in E \ a, b \in V_v\}$ and do

$V = V_v, \ \ E = E_v, \ g(x) = \langle V, E \rangle$

3 - End

Minimum Neighbouring algorithm

1 - Build the graph associated to matrix **A**,
$g(x) = \langle V, E \rangle$, being V the set of nodes and
$E = \{\{a, b\} : a \neq b \ / \ a, b \in V\}$

2 - While $V \neq \emptyset$:

2.1- Choose a minimum degree node v in
$g(x) = \langle V, E \rangle$ and order it as next node

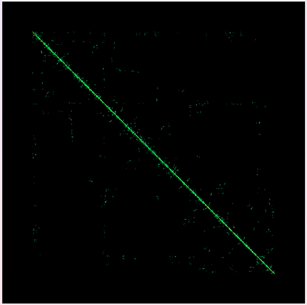
2.2 - Define:

$V_v = V - \{v\}$

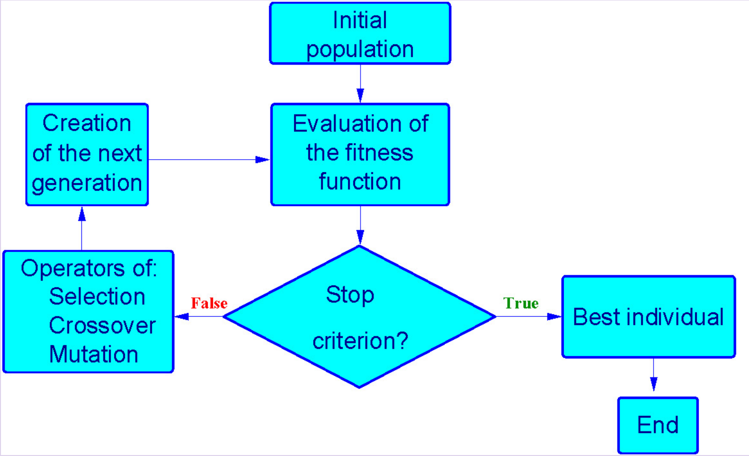
$E_v = \{\{a, b\} \in E \ a, b \in V_v\}$ and do

$V = V_v, \ \ E = E_v, \ g(x) = \langle V, E \rangle$

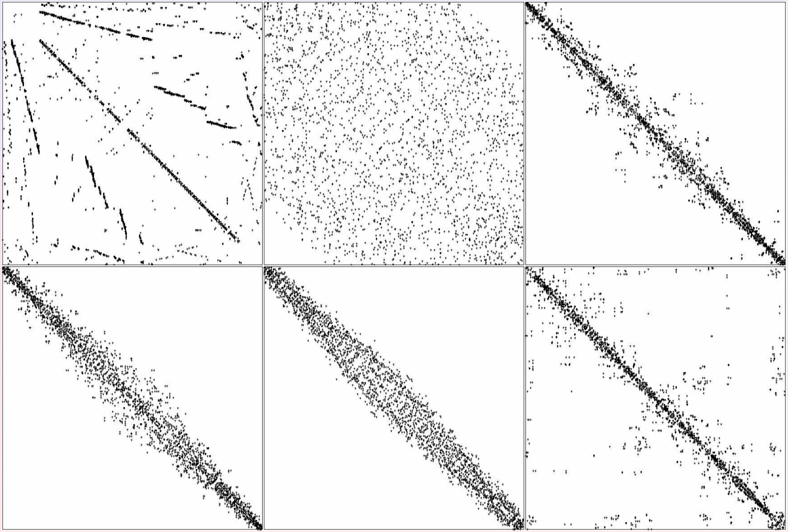
3 - End



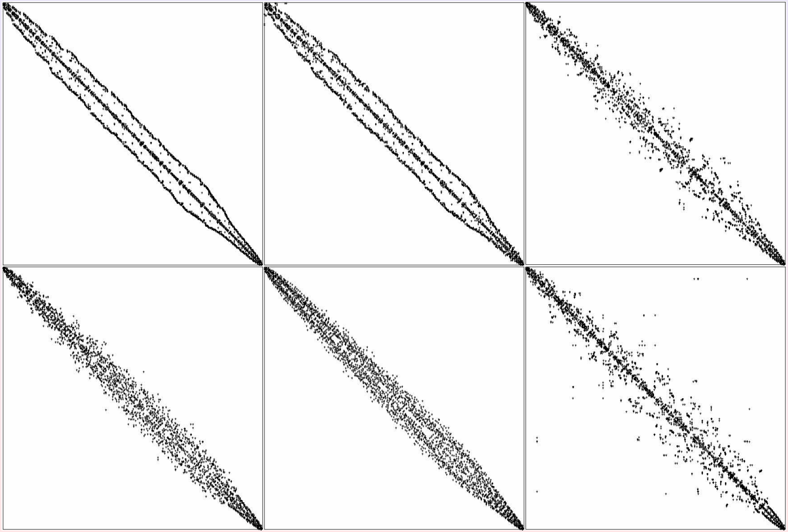
Genetic algorithms



Sparsity patterns with different GAs strategies



Sparsity patterns with different GAs strategies (RCM)



Preconditioning

Quality indicators

- Closeness of the condition number to 1,

$$K_2(\mathbf{MA}) \leq \frac{1+\|\mathbf{MA}-\mathbf{I}\|_2}{1-\|\mathbf{MA}-\mathbf{I}\|_2}$$

- Departure from normality ($\{\lambda_k\}_{k=1}^n, \{\sigma_k\}_{k=1}^n$ the eigen and singular values)

$$\frac{1}{n}\sum_{k=1}^{n}(|\lambda_k| - \sigma_k)^2 \leq \frac{2}{n}\|\mathbf{MA}\|_F^2\,(1-\sigma_n)$$

- Clustering of eigenvalues,

$$\sum_{k=1}^{n}(1-\lambda_k)^2 \leq \|\mathbf{MA}-\mathbf{I}\|_F^2$$

- Clustering of singular values

$$\sum_{k=1}^{n}(1-\sigma_k)^2 \leq \|\mathbf{MA}-\mathbf{I}\|_F^2$$

Preconditioning

Quality indicators

- Closeness of the condition number to 1,

$$K_2(\mathbf{MA}) \leq \frac{1+\|\mathbf{MA}-\mathbf{I}\|_2}{1-\|\mathbf{MA}-\mathbf{I}\|_2}$$

- Departure from normality ($\{\lambda_k\}_{k=1}^n, \{\sigma_k\}_{k=1}^n$ the eigen and singular values)

$$\frac{1}{n}\sum_{k=1}^{n}(|\lambda_k| - \sigma_k)^2 \leq \frac{2}{n}\|\mathbf{MA}\|_F^2\,(1-\sigma_n)$$

- Clustering of eigenvalues,

$$\sum_{k=1}^{n}(1-\lambda_k)^2 \leq \|\mathbf{MA}-\mathbf{I}\|_F^2$$

- Clustering of singular values

$$\sum_{k=1}^{n}(1-\sigma_k)^2 \leq \|\mathbf{MA}-\mathbf{I}\|_F^2$$

Preconditioning

Quality indicators

- Closeness of the condition number to 1,

$$K_2(\mathbf{MA}) \leq \frac{1+\|\mathbf{MA}-\mathbf{I}\|_2}{1-\|\mathbf{MA}-\mathbf{I}\|_2}$$

- Departure from normality ($\{\lambda_k\}_{k=1}^n, \{\sigma_k\}_{k=1}^n$ the eigen and singular values)

$$\frac{1}{n}\sum_{k=1}^{n}(|\lambda_k| - \sigma_k)^2 \leq \frac{2}{n}\,\|\mathbf{MA}\|_F^2\,(1-\sigma_n)$$

- Clustering of eigenvalues,

$$\sum_{k=1}^{n}(1-\lambda_k)^2 \leq \|\mathbf{MA}-\mathbf{I}\|_F^2$$

- Clustering of singular values

$$\sum_{k=1}^{n}(1-\sigma_k)^2 \leq \|\mathbf{MA}-\mathbf{I}\|_F^2$$

Preconditioning

Quality indicators

- Closeness of the condition number to 1,

$$K_2(\mathbf{MA}) \leq \frac{1+\|\mathbf{MA}-\mathbf{I}\|_2}{1-\|\mathbf{MA}-\mathbf{I}\|_2}$$

- Departure from normality ($\{\lambda_k\}_{k=1}^n, \{\sigma_k\}_{k=1}^n$ the eigen and singular values)

$$\frac{1}{n}\sum_{k=1}^{n}(|\lambda_k| - \sigma_k)^2 \leq \frac{2}{n}\|\mathbf{MA}\|_F^2\,(1-\sigma_n)$$

- Clustering of eigenvalues,

$$\sum_{k=1}^{n}(1-\lambda_k)^2 \leq \|\mathbf{MA}-\mathbf{I}\|_F^2$$

- Clustering of singular values

$$\sum_{k=1}^{n}(1-\sigma_k)^2 \leq \|\mathbf{MA}-\mathbf{I}\|_F^2$$

Standard preconditioners

$$\mathbf{M}^{-1}\mathbf{x}_{i+1} = \mathbf{M}^{-1}\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

Jacobi

$$\mathbf{D}\mathbf{x}_{i+1} = \mathbf{D}\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

$$\mathbf{M}^{-1} = D = diag(\mathbf{A})$$

Optimum diagonal

$$\mathbf{M} = diag\left(\frac{a_{11}}{\|\mathbf{e}_1^T\mathbf{A}\|_2^2}, \frac{a_{22}}{\|\mathbf{e}_2^T\mathbf{A}\|_2^2}, \ldots, \frac{a_{nn}}{\|\mathbf{e}_n^T\mathbf{A}\|_2^2}\right)$$

$$\|\mathbf{MA} - \mathbf{I}\|_F^2 = n - \sum_{i=1}^{n} \frac{a_{ii}}{\|\mathbf{e}_i^T\mathbf{A}\|_2^2}$$

SSOR(ω)

$$\frac{1}{\omega(2-\omega)}(\mathbf{D}-\omega\mathbf{E})\,\mathbf{D}^{-1}(\mathbf{D}-\omega\mathbf{F})\,\mathbf{x}_{i+1}$$

$$= \frac{1}{\omega(2-\omega)}(\mathbf{D}-\omega\mathbf{E})\,\mathbf{D}^{-1}(\mathbf{D}-\omega\mathbf{F})\,\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

$$\mathbf{M}^{-1} = \left(\mathbf{I}-\omega\mathbf{E}\mathbf{D}^{-1}\right)\left(\frac{\mathbf{D}-\omega\mathbf{F}}{\omega(2-\omega)}\right)$$

In the case of symmetric matrices,

$$\mathbf{M}^{-1} = \left(\frac{(\mathbf{D}-\omega\mathbf{E})\mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}}\right)\left(\frac{(\mathbf{D}-\omega\mathbf{E})\mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}}\right)^T$$

Standard preconditioners

$$\mathbf{M}^{-1}\mathbf{x}_{i+1} = \mathbf{M}^{-1}\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

Jacobi

$$\mathbf{D}\mathbf{x}_{i+1} = \mathbf{D}\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

$$\mathbf{M}^{-1} = D = diag(\mathbf{A})$$

Optimum diagonal

$$\mathbf{M} = diag\left(\frac{a_{11}}{\|\mathbf{e}_1^T\mathbf{A}\|_2^2}, \frac{a_{22}}{\|\mathbf{e}_2^T\mathbf{A}\|_2^2}, \ldots, \frac{a_{nn}}{\|\mathbf{e}_n^T\mathbf{A}\|_2^2}\right)$$

$$\|\mathbf{MA} - \mathbf{I}\|_F^2 = n - \sum_{i=1}^{n} \frac{a_{ii}}{\|\mathbf{e}_i^T\mathbf{A}\|_2^2}$$

SSOR(ω)

$$\frac{1}{\omega(2-\omega)}(\mathbf{D}-\omega\mathbf{E})\,\mathbf{D}^{-1}(\mathbf{D}-\omega\mathbf{F})\,\mathbf{x}_{i+1}$$

$$= \frac{1}{\omega(2-\omega)}(\mathbf{D}-\omega\mathbf{E})\,\mathbf{D}^{-1}(\mathbf{D}-\omega\mathbf{F})\,\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

$$\mathbf{M}^{-1} = \left(\mathbf{I}-\omega\mathbf{E}\mathbf{D}^{-1}\right)\left(\frac{\mathbf{D}-\omega\mathbf{F}}{\omega(2-\omega)}\right)$$

In the case of symmetric matrices,

$$\mathbf{M}^{-1} = \left(\frac{(\mathbf{D}-\omega\mathbf{E})\mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}}\right)\left(\frac{(\mathbf{D}-\omega\mathbf{E})\mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}}\right)^T$$

Standard preconditioners

$$\mathbf{M}^{-1}\mathbf{x}_{i+1} = \mathbf{M}^{-1}\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

Jacobi

$$\mathbf{D}\mathbf{x}_{i+1} = \mathbf{D}\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

$$\mathbf{M}^{-1} = D = diag(\mathbf{A})$$

Optimum diagonal

$$\mathbf{M} = diag\left(\frac{a_{11}}{\|\mathbf{e}_1^T\mathbf{A}\|_2^2}, \frac{a_{22}}{\|\mathbf{e}_2^T\mathbf{A}\|_2^2}, \ldots, \frac{a_{nn}}{\|\mathbf{e}_n^T\mathbf{A}\|_2^2}\right)$$

$$\|\mathbf{MA} - \mathbf{I}\|_F^2 = n - \sum_{i=1}^{n} \frac{a_{ii}}{\|\mathbf{e}_i^T\mathbf{A}\|_2^2}$$

SSOR(ω)

$$\frac{1}{\omega\,(2-\omega)}\,(\mathbf{D}-\omega\mathbf{E})\,\mathbf{D}^{-1}\,(\mathbf{D}-\omega\mathbf{F})\,\mathbf{x}_{i+1}$$

$$= \frac{1}{\omega\,(2-\omega)}\,(\mathbf{D}-\omega\mathbf{E})\,\mathbf{D}^{-1}\,(\mathbf{D}-\omega\mathbf{F})\,\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

$$\mathbf{M}^{-1} = \left(\mathbf{I}-\omega\mathbf{E}\mathbf{D}^{-1}\right)\left(\frac{\mathbf{D}-\omega\mathbf{F}}{\omega(2-\omega)}\right)$$

In the case of symmetric matrices,

$$\mathbf{M}^{-1} = \left(\frac{(\mathbf{D}-\omega\mathbf{E})\mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}}\right)\left(\frac{(\mathbf{D}-\omega\mathbf{E})\mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}}\right)^T$$

Standard preconditioners

$$\mathbf{M}^{-1}\mathbf{x}_{i+1} = \mathbf{M}^{-1}\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

Jacobi

$$\mathbf{D}\mathbf{x}_{i+1} = \mathbf{D}\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

$$\mathbf{M}^{-1} = D = diag(\mathbf{A})$$

Optimum diagonal

$$\mathbf{M} = diag\left(\frac{a_{11}}{\|\mathbf{e}_1^T\mathbf{A}\|_2^2}, \frac{a_{22}}{\|\mathbf{e}_2^T\mathbf{A}\|_2^2}, \ldots, \frac{a_{nn}}{\|\mathbf{e}_n^T\mathbf{A}\|_2^2}\right)$$

$$\|\mathbf{MA} - \mathbf{I}\|_F^2 = n - \sum_{i=1}^{n} \frac{a_{ii}}{\|\mathbf{e}_i^T\mathbf{A}\|_2^2}$$

SSOR(ω)

$$\frac{1}{\omega\,(2-\omega)}\,(\mathbf{D}-\omega\mathbf{E})\,\mathbf{D}^{-1}\,(\mathbf{D}-\omega\mathbf{F})\,\mathbf{x}_{i+1}$$

$$= \frac{1}{\omega\,(2-\omega)}\,(\mathbf{D}-\omega\mathbf{E})\,\mathbf{D}^{-1}\,(\mathbf{D}-\omega\mathbf{F})\,\mathbf{x}_i + (\mathbf{b} - \mathbf{A}\mathbf{x}_i)$$

$$\mathbf{M}^{-1} = \left(\mathbf{I}-\omega\mathbf{E}\mathbf{D}^{-1}\right)\left(\frac{\mathbf{D}-\omega\mathbf{F}}{\omega(2-\omega)}\right)$$

In the case of symmetric matrices,

$$\mathbf{M}^{-1} = \left(\frac{(\mathbf{D}-\omega\mathbf{E})\mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}}\right)\left(\frac{(\mathbf{D}-\omega\mathbf{E})\mathbf{D}^{-1/2}}{\sqrt{\omega(2-\omega)}}\right)^T$$

Standard preconditioners

ILUT

For $i = 1, ..., n$, do
  $w = a_{i*}$
  For $k = 1, ..., i-1$, if $w_k \neq 0$ do
    $w_k = w_k / a_{kk}$
    If $w_k \neq 0$ then $w = w - w_{k*} u_{k*}$
    End
  End
  $l_{i,j} = w_j$ for $j = 1, ..., i-1$
  $u_{i,j} = w_j$ for $j = i, ..., n$
  $w = 0$
End

ILU(0)

$\mathbf{A} = \mathbf{LU} \approx \mathbf{ILU}(0) = \mathbf{M}^{-1}$
where m_{ij} are the entries of $\mathbf{M}^{-1}$ such that,
$m_{ij} = 0$ if $a_{ij} = 0$
$\{\mathbf{A} - \mathbf{LU}\}_{ij} = 0$ if $a_{ij} \neq 0$

Incomplete factorizations after reordering

- The number of CG iterations is not related to the number of fill-ins we are dropping, but is almost directly related to the norm of the residual matrix $MA - I$
- In general, local reorderings (RCM) give the best results (sufficient condition)
- However, local reorderings are more affected by the choice of the initial node and by the ordering of nodes within level sets
- The harder is the problem (non regular and unstructure meshes, discontinuous coefficients, anisotropy, strong nonsymmetry, ...), the more important is the reordering
- Many of the reorderings which are better suited for parallel computations do not give very good results. An alternative is to use domain decomposition techniques and within each local subdomain a local reordering of the nodes
- The effect of reordering is much more important in nonsymmetric problems than in symmetric ones

Incomplete factorizations after reordering

- The number of CG iterations is not related to the number of fill-ins we are dropping, but is almost directly related to the norm of the residual matrix $MA - I$
- In general, local reorderings (RCM) give the best results (sufficient condition)
- However, local reorderings are more affected by the choice of the initial node and by the ordering of nodes within level sets
- The harder is the problem (non regular and unstructure meshes, discontinuous coefficients, anisotropy, strong nonsymmetry, ...), the more important is the reordering
- Many of the reorderings which are better suited for parallel computations do not give very good results. An alternative is to use domain decomposition techniques and within each local subdomain a local reordering of the nodes
- The effect of reordering is much more important in nonsymmetric problems than in symmetric ones

Incomplete factorizations after reordering

- The number of CG iterations is not related to the number of fill-ins we are dropping, but is almost directly related to the norm of the residual matrix $MA - I$
- In general, local reorderings (RCM) give the best results (sufficient condition)
- However, local reorderings are more affected by the choice of the initial node and by the ordering of nodes within level sets
- The harder is the problem (non regular and unstructure meshes, discontinuous coefficients, anisotropy, strong nonsymmetry, ...), the more important is the reordering
- Many of the reorderings which are better suited for parallel computations do not give very good results. An alternative is to use domain decomposition techniques and within each local subdomain a local reordering of the nodes
- The effect of reordering is much more important in nonsymmetric problems than in symmetric ones

Incomplete factorizations after reordering

- The number of CG iterations is not related to the number of fill-ins we are dropping, but is almost directly related to the norm of the residual matrix $MA - I$
- In general, local reorderings (RCM) give the best results (sufficient condition)
- However, local reorderings are more affected by the choice of the initial node and by the ordering of nodes within level sets
- The harder is the problem (non regular and unstructure meshes, discontinuous coefficients, anisotropy, strong nonsymmetry, ...), the more important is the reordering
- Many of the reorderings which are better suited for parallel computations do not give very good results. An alternative is to use domain decomposition techniques and within each local subdomain a local reordering of the nodes
- The effect of reordering is much more important in nonsymmetric problems than in symmetric ones

Incomplete factorizations after reordering

- The number of CG iterations is not related to the number of fill-ins we are dropping, but is almost directly related to the norm of the residual matrix $MA - I$
- In general, local reorderings (RCM) give the best results (sufficient condition)
- However, local reorderings are more affected by the choice of the initial node and by the ordering of nodes within level sets
- The harder is the problem (non regular and unstructure meshes, discontinuous coefficients, anisotropy, strong nonsymmetry, ...), the more important is the reordering
- Many of the reorderings which are better suited for parallel computations do not give very good results. An alternative is to use domain decomposition techniques and within each local subdomain a local reordering of the nodes
- The effect of reordering is much more important in nonsymmetric problems than in symmetric ones

Incomplete factorizations after reordering

- The number of CG iterations is not related to the number of fill-ins we are dropping, but is almost directly related to the norm of the residual matrix $MA - I$
- In general, local reorderings (RCM) give the best results (sufficient condition)
- However, local reorderings are more affected by the choice of the initial node and by the ordering of nodes within level sets
- The harder is the problem (non regular and unstructure meshes, discontinuous coefficients, anisotropy, strong nonsymmetry, ...), the more important is the reordering
- Many of the reorderings which are better suited for parallel computations do not give very good results. An alternative is to use domain decomposition techniques and within each local subdomain a local reordering of the nodes
- The effect of reordering is much more important in nonsymmetric problems than in symmetric ones

SPAI

Find $\mathbf{M} \in \mathcal{S}$ such that

$$\mathbf{M} = \underset{\mathbf{M}' \in \mathcal{S}}{\arg\min} \|\mathbf{A}\mathbf{M}' - \mathbf{I}\|_F$$

$\mathbf{r}_k = \mathbf{A}\mathbf{m}_k - \mathbf{e}_k$, $\mathcal{I}_k = \{i \in \{1, 2, \ldots, n\} r_{ik} \neq 0\}$, $\mathcal{L}_k = \{l \in \{1, 2, \ldots, n\} \;/\; m_{lk} \neq 0\}$
a new entry is searched in $\mathcal{J}_k = \{j \in \mathcal{L}_k^c \;/\; a_{ij} \neq 0, \forall i \in \mathcal{I}_k\}$
$\mathcal{L}_k \cup \{j\} = \{i_1^k, i_2^k, \ldots, i_{p_k}^k\}$ is not empty, being p_k the current number of nonzero entries in $\mathbf{m}_k$, and $i_{p_k}^k = j$, $\forall j \in \mathcal{J}_k$. For each j, compute,

$$||\mathbf{A}\mathbf{m}_\mathbf{k} - \mathbf{e}_\mathbf{k}||_2^2 = 1 - \sum_{l=1}^{p_k} \frac{[\det(\mathbf{D}_l^k)]^2}{\det(\mathbf{G}_{l-1}^k)\det(\mathbf{G}_l^k)}$$

where, $\forall k$, $\det(\mathbf{G}_0^k) = 1$ and $\mathbf{G}_l^k$ is the Gram matrix of columns $i_1^k, i_2^k, \ldots, i_l^k$ of matrix $\mathbf{A}$, $\mathbf{D}_l^k$ results from replacing the last row of $\mathbf{G}_l^k$ by $a_{k\,i_1^k}, a_{k\,i_2^k}, \ldots, a_{k\,i_l^k}$, with $1 \leq l \leq p_k$. We select the index j_k that minimises $\|\mathbf{A}\mathbf{m}_\mathbf{k} - \mathbf{e}_\mathbf{k}\|_2$. Thus, $\mathbf{m}_k$ is searched in the set $\mathcal{S}_k = \{\mathbf{m}_k \in R^n / m_{ik} = 0; \forall i \notin \mathcal{L}_k \cup \{j_k\}\}$

$$\mathbf{m}_k = \sum_{l=1}^{p_k} \frac{\det(\mathbf{D}_l^k)}{\det(\mathbf{G}_{l-1}^k)\det(\mathbf{G}_l^k)} \tilde{\mathbf{m}}_l$$

where $\tilde{\mathbf{m}}_l$ is the vector with nonzero entries i_h^k $(1 \leq h \leq l)$, obtained from replacing the las row of $\det(\mathbf{G}_l^k)$ by $\mathbf{e}_h^t$, with $1 \leq l \leq p_k$.

SPAI

Find $\mathbf{M} \in \mathcal{S}$ such that

$$\mathbf{M} = \underset{\mathbf{M}' \in \mathcal{S}}{\arg\min} \|\mathbf{AM}' - \mathbf{I}\|_F$$

$\mathbf{r}_k = \mathbf{A}\mathbf{m}_k - \mathbf{e}_k$, $\mathcal{I}_k = \{i \in \{1, 2, \dots, n\} r_{ik} \neq 0\}$, $\mathcal{L}_k = \{l \in \{1, 2, \dots, n\} \ / \ m_{lk} \neq 0\}$
a new entry is searched in $\mathcal{J}_k = \{j \in \mathcal{L}_k^c \ / \ a_{ij} \neq 0, \forall i \in \mathcal{I}_k\}$
$\mathcal{L}_k \cup \{j\} = \{i_1^k, i_2^k, \dots, i_{p_k}^k\}$ is not empty, being p_k the current number of nonzero entries in $\mathbf{m}_k$, and $i_{p_k}^k = j$, $\forall j \in \mathcal{J}_k$. For each j, compute,

$$||\mathbf{Am_k} - \mathbf{e_k}||_2^2 = 1 - \sum_{l=1}^{p_k} \frac{[\det(\mathbf{D}_l^k)]^2}{\det(\mathbf{G}_{l-1}^k)\det(\mathbf{G}_l^k)}$$

where, $\forall k$, $\det(\mathbf{G}_0^k) = 1$ and $\mathbf{G}_l^k$ is the Gram matrix of columns $i_1^k, i_2^k, \dots, i_l^k$ of matrix $\mathbf{A}$, $\mathbf{D}_l^k$ results from replacing the last row of $\mathbf{G}_l^k$ by $a_{k\,i_1^k}, a_{k\,i_2^k}, \dots, a_{k\,i_l^k}$, with $1 \leq l \leq p_k$. We select the index j_k that minimises $\|\mathbf{Am_k} - \mathbf{e_k}\|_2$. Thus, $\mathbf{m}_k$ is searched in the set $\mathcal{S}_k = \{\mathbf{m}_k \in R^n / m_{ik} = 0; \forall i \notin \mathcal{L}_k \cup \{j_k\}\}$

$$\mathbf{m}_k = \sum_{l=1}^{p_k} \frac{\det(\mathbf{D}_l^k)}{\det(\mathbf{G}_{l-1}^k)\det(\mathbf{G}_l^k)} \tilde{\mathbf{m}}_l$$

where $\tilde{\mathbf{m}}_l$ is the vector with nonzero entries i_h^k $(1 \leq h \leq l)$, obtained from replacing the las row of $\det(\mathbf{G}_l^k)$ by $\mathbf{e}_h^t$, with $1 \leq l \leq p_k$.

SPAI

Find $\mathbf{M} \in \mathcal{S}$ such that

$$\mathbf{M} = \underset{\mathbf{M}' \in \mathcal{S}}{\arg\min} \|\mathbf{A}\mathbf{M}' - \mathbf{I}\|_F$$

$\mathbf{r}_k = \mathbf{A}\mathbf{m}_k - \mathbf{e}_k$, $\mathcal{I}_k = \{i \in \{1, 2, \ldots, n\} r_{ik} \neq 0\}$, $\mathcal{L}_k = \{l \in \{1, 2, \ldots, n\} \;/\; m_{lk} \neq 0\}$
a new entry is searched in $\mathcal{J}_k = \{j \in \mathcal{L}_k^c \;/\; a_{ij} \neq 0, \forall i \in \mathcal{I}_k\}$
$\mathcal{L}_k \cup \{j\} = \{i_1^k, i_2^k, \ldots, i_{p_k}^k\}$ is not empty, being p_k the current number of nonzero entries in $\mathbf{m}_k$, and $i_{p_k}^k = j$, $\forall j \in \mathcal{J}_k$. For each j, compute,

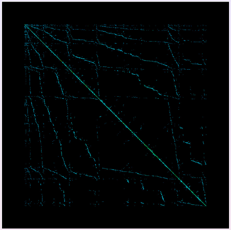
$$||\mathbf{Am_k} - \mathbf{e_k}||_2^2 = 1 - \sum_{l=1}^{p_k} \frac{[\det(\mathbf{D}_l^k)]^2}{\det(\mathbf{G}_{l-1}^k)\det(\mathbf{G}_l^k)}$$

where, $\forall k$, $\det(\mathbf{G}_0^k) = 1$ and $\mathbf{G}_l^k$ is the Gram matrix of columns $i_1^k, i_2^k, \ldots, i_l^k$ of matrix $\mathbf{A}$, $\mathbf{D}_l^k$ results from replacing the last row of $\mathbf{G}_l^k$ by $a_{k\,i_1^k}, a_{k\,i_2^k}, \ldots, a_{k\,i_l^k}$, with $1 \leq l \leq p_k$. We select the index j_k that minimises $\|\mathbf{Am_k} - \mathbf{e_k}\|_2$. Thus, $\mathbf{m}_k$ is searched in the set $\mathcal{S}_k = \{\mathbf{m}_k \in R^n / m_{ik} = 0; \forall i \notin \mathcal{L}_k \cup \{j_k\}\}$

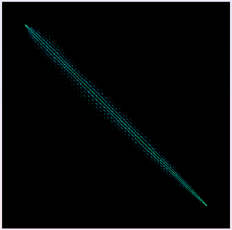
$$\mathbf{m}_k = \sum_{l=1}^{p_k} \frac{\det(\mathbf{D}_l^k)}{\det(\mathbf{G}_{l-1}^k)\det(\mathbf{G}_l^k)} \tilde{\mathbf{m}}_l$$

where $\tilde{\mathbf{m}}_l$ is the vector with nonzero entries i_h^k $(1 \leq h \leq l)$, obtained from replacing the las row of $\det(\mathbf{G}_l^k)$ by $\mathbf{e}_h^t$, with $1 \leq l \leq p_k$.

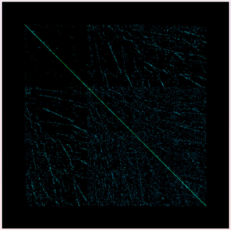
SPAI



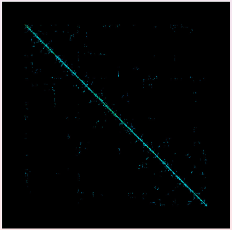
Initial Ordering



Reverse Cuthill McKee



Minimum Degree



Minimum Neighbouring

Approximate inverse after reordering

Let $\mathbf{P}$ be the permutation matrix related to a reordering algorithm
$(\mathbf{P^T A P})^{-1} = \mathbf{P^T A^{-1} P} \Rightarrow$ The inverse of a reordered matrix $=$ the reordered inverse
Let ε be the tolerance in subspace $\mathcal{S} \subset M_n(R)$,

$$\min_{\mathbf{M} \in \mathcal{S}} \|\mathbf{MA} - \mathbf{I}\|_F = \|\mathbf{NA} - \mathbf{I}\|_F < \varepsilon$$

$$\min_{\mathbf{M}' \in \mathbf{P}^T \mathcal{S} \mathbf{P}} \|\mathbf{M}'\mathbf{P}^T\mathbf{AP} - \mathbf{I}\|_F = \min_{\mathbf{M}' \in \mathbf{P}^T \mathcal{S} \mathbf{P}} \|\mathbf{PM}'\mathbf{P}^T\mathbf{A} - \mathbf{I}\|_F = \min_{\mathbf{M} \in \mathcal{S}} \|\mathbf{MA} - \mathbf{I}\| < \varepsilon$$

Let $\mathcal{S}'$ be a subspace of $M_n(R)$ with the same amount of non null entries as $\mathcal{S}$, related to a reordering algorithm. If the reordering verifies,

$$\|\mathbf{N}'\mathbf{P}^T\mathbf{AP} - \mathbf{I}\|_F = \min_{\mathbf{M}' \in \mathcal{S}'} \|\mathbf{M}'\mathbf{P}^T\mathbf{AP} - \mathbf{I}\|_F \leq \min_{\mathbf{M}' \in \mathbf{P}^T \mathcal{S} \mathbf{P}} \|\mathbf{M}'\mathbf{P}^T\mathbf{AP} - \mathbf{I}\|_F < \varepsilon$$

In such cases we can reduce the number of nonzero entries in the inverse of the reordered matrix (for a similar quality to that without reordering)

Wind field simulation

Preconditioned Conjugate Gradient Algorithm

Initial guess $\mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$;

$\mathbf{z}_0 = \mathbf{M}\mathbf{r}_0$, $\mathbf{p}_0 = \mathbf{z}_0$;

While $\| \mathbf{r}_j \| / \| \mathbf{r}_0 \| \geq \varepsilon \quad (j = 0, 1, 2, 3, \ldots)$, do

$$\alpha_j = \frac{\langle \mathbf{r}_j, \mathbf{z}_j \rangle}{\langle \mathbf{A}\mathbf{p}_j, \mathbf{p}_j \rangle};$$

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j;$$

$$\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j \mathbf{A}\mathbf{p}_j;$$

$$\mathbf{z}_{j+1} = \mathbf{M}\mathbf{r}_{j+1};$$

$$\beta_j = \frac{\langle \mathbf{r}_{j+1}, \mathbf{z}_{j+1} \rangle}{\langle \mathbf{r}_j, \mathbf{z}_j \rangle};$$

$$\mathbf{p}_{j+1} = \mathbf{z}_{j+1} + \beta_j \mathbf{p}_j;$$

End

Wind field simulation

Preconditioned Conjugate Gradient Algorithm

Initial guess $\mathbf{x}_0$, $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$;

$\mathbf{z}_0 = \mathbf{M}\mathbf{r}_0$, $\mathbf{p}_0 = \mathbf{z}_0$;

While $\| \mathbf{r}_j \| / \| \mathbf{r}_0 \| \geq \varepsilon \quad (j = 0, 1, 2, 3, \ldots)$, do

$$\alpha_j = \frac{\langle \mathbf{r}_j, \mathbf{z}_j \rangle}{\langle \mathbf{A}\mathbf{p}_j, \mathbf{p}_j \rangle};$$

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j;$$

$$\mathbf{r}_{j+1} = \mathbf{r}_j - \alpha_j \mathbf{A}\mathbf{p}_j;$$

$$\mathbf{z}_{j+1} = \mathbf{M}\mathbf{r}_{j+1};$$

$$\beta_j = \frac{\langle \mathbf{r}_{j+1}, \mathbf{z}_{j+1} \rangle}{\langle \mathbf{r}_j, \mathbf{z}_j \rangle};$$

$$\mathbf{p}_{j+1} = \mathbf{z}_{j+1} + \beta_j \mathbf{p}_j;$$

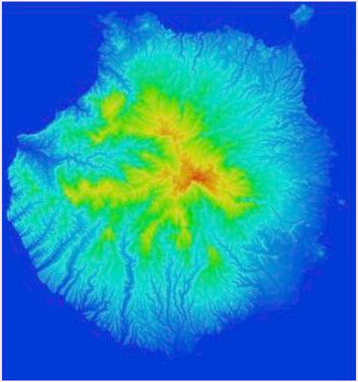
End

Wind field simulation

Wind modeling facilities

- Wind farm location
- Wind maps
- Definition of a measure grid
- Computation of a velocity field within a more complex problem

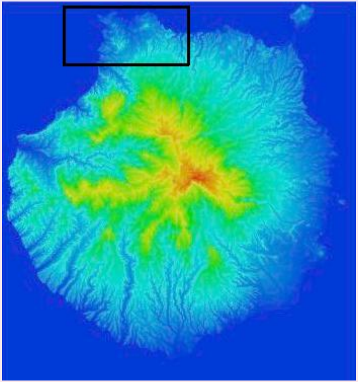
Wind field simulation



Gran Canaria simulation

- A 3-D complex terrain problem
- Unstructured mesh with elements of very different sizes
- Large linear system of equations with positive definite matrix
- One simulation for each set of measurements (often given every ten minutes)
- With the same parameters, the matrix does not change along the simulation

Wind field simulation



A mass consistent model

Based on the continuity equation,

$$\vec{\nabla} \cdot \vec{u} = 0 \qquad \text{in} \quad \Omega$$

No-flow-through conditions on the terrain

$$\vec{n} \cdot \vec{u} = 0 \qquad \text{on} \quad \Gamma_b$$

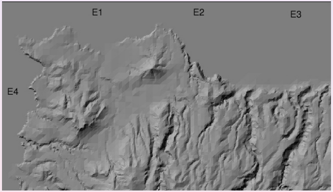
Adjust $\vec{u}(\widetilde{u}, \widetilde{v}, \widetilde{w})$ *to* $\vec{v}_0(u_0, v_0, w_0)$

$$E(\widetilde{u}, \widetilde{v}, \widetilde{w}) = \int_\Omega \left[\alpha_1^2 \left((\widetilde{u} - u_0)^2 + (\widetilde{v} - v_0)^2 \right) + \alpha_2^2 (\widetilde{w} - w_0)^2 \right] \, d\Omega$$

Equivalent to find the saddle point $(\vec{v}(u, v, w), \phi)$ of

$$F(\vec{u}, \lambda) = E(\vec{u}) + \int_\Omega \lambda \vec{\nabla} \cdot \vec{u} \, d\Omega$$

Wind field simulation



Elliptic problem

Lagrange multiplier leads to the Euler-Lagrange equations

$$\vec{v} = \vec{v}_0 + T\,\vec{\nabla}\phi$$
$$T = \left(\frac{1}{2\alpha_1^2}, \frac{1}{2\alpha_1^2}, \frac{1}{2\alpha_2^2}\right)$$

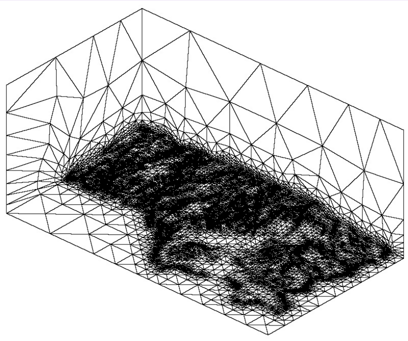
which yield the following elliptic problem

$$\vec{\nabla}\cdot T\,\vec{\nabla}\phi = -\vec{\nabla}\cdot\vec{v}_0 \quad \text{in} \quad \Omega$$

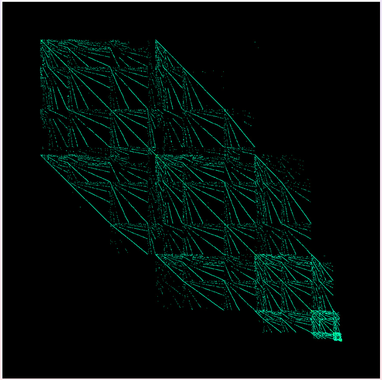
$$\phi = 0 \quad \text{on} \quad \Gamma_a$$

$$\vec{n}\cdot T\,\vec{\nabla}\phi = -\vec{n}\cdot\vec{v}_0 \quad \text{on} \quad \Gamma_b$$

Wind field simulation

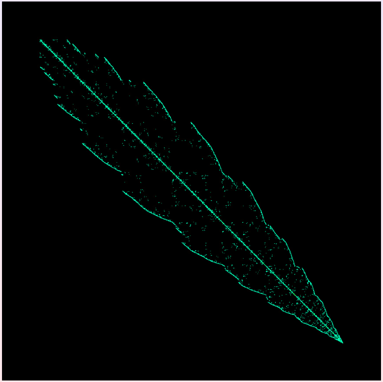


Sparsity pattern of a 43954 matrix

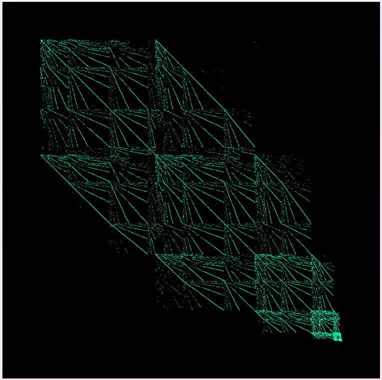


Wind field simulation

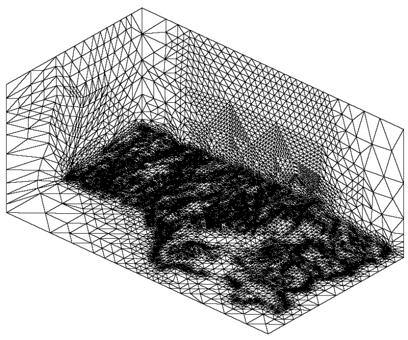
Sparsity pattern of a 43954 matrix reordered with RCM



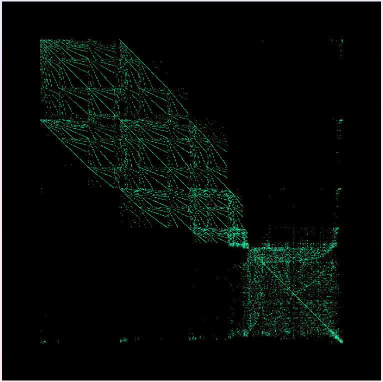
Sparsity pattern of a 43954 matrix



Wind field simulation

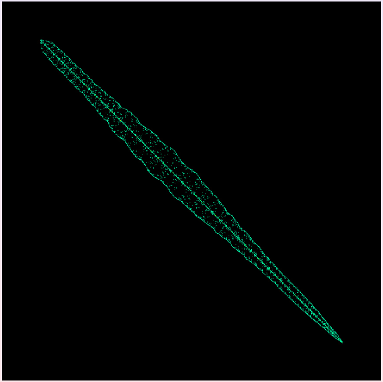


Sparsity pattern of a 63746 matrix

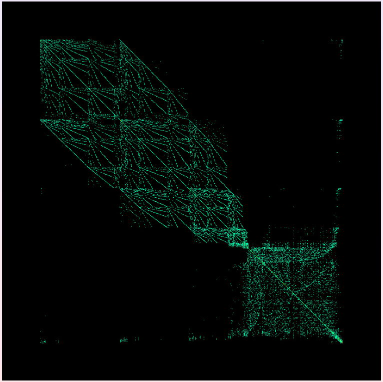


Wind field simulation

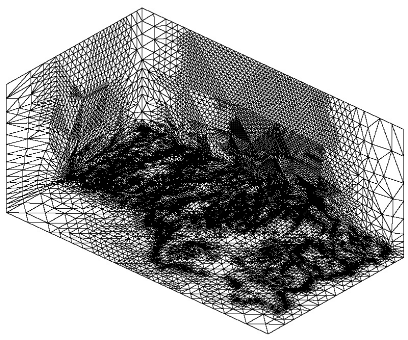
Sparsity pattern of a 63746 matrix reordered with RCM



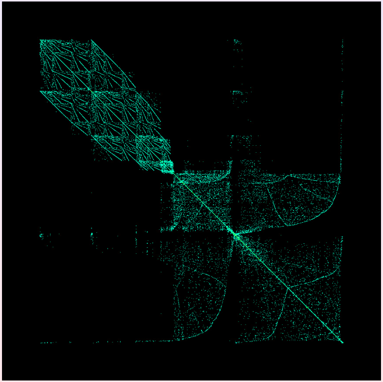
Sparsity pattern of a 63746 matrix



Wind field simulation

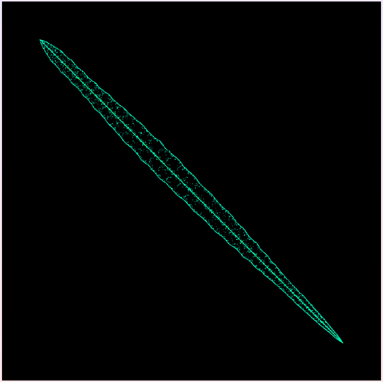


Sparsity pattern of a 98999 matrix

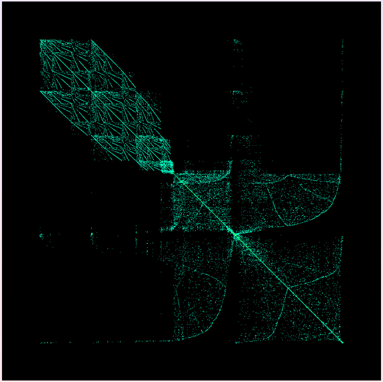


Wind field simulation

Sparsity pattern of a 98999 matrix reordered with RCM

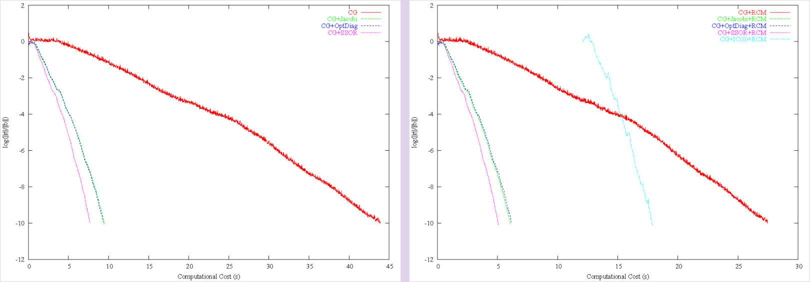


Sparsity pattern of a 98999 matrix



Wind field simulation

Convergence of Krylov Subspace Methods

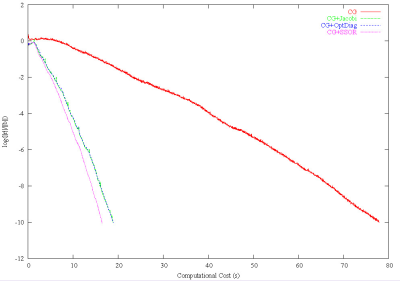


Performance of PCG in a 43954 matrix

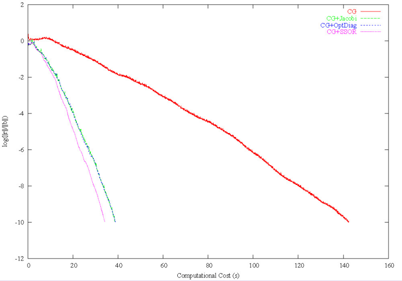
Effect of reordering

Wind field simulation

Convergence of Krylov Subspace Methods



Performance of PCG in a 63746 matrix



Performance of PCG in a 98999 matrix

Nonsymmetric linear systems

Preconditioned VGMRES algorithm

Initial guess $\mathbf{x}_0$. $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$;

Choose k_{init}, k_{top}, $\delta \in [0,1]$, $k = k_{init}$

While $\| \hat{\mathbf{r}}_{i-1} \| / \| \mathbf{r}_0 \| \geq \varepsilon \quad (i = 1, 2, 3, \ldots)$, do

$\beta_{i-1} = \| \mathbf{r}_{i-1} \|$, $\mathbf{v}_i = \mathbf{r}_{i-1}/\beta_{i-1}$;

If $\| \mathbf{r}_{i-1} \| / \| \mathbf{r}_0 \| \geq \delta$ and $k < k_{top}$ then $k = k+1$;

For $j = 1, \ldots, k$

$\mathbf{z}_j = \mathbf{M}\mathbf{v}_j; \quad \mathbf{w} = \mathbf{A}\mathbf{z}_j$;

For $n = 1, \ldots, j$

$\{\mathbf{H}\}_{nj} = \langle \mathbf{w}, \mathbf{v}_n \rangle$;

$\mathbf{w} = \mathbf{w} - \{\mathbf{H}\}_{nj}\, \mathbf{v}_n$;

End

$\{\mathbf{H}\}_{j+1,j} = \| \mathbf{w} \|$;

$\mathbf{v}_{j+1} = \mathbf{w} / \{\mathbf{H}\}_{j+1,j}$;

End

Solve $\mathbf{U}_k^t \bar{\mathbf{p}} = \mathbf{d}_k$ and $\mathbf{U}_k \mathbf{p} = \bar{\mathbf{p}}$;

with $\begin{array}{l} \{\mathbf{d}_k\}_m = \{\mathbf{H}\}_{1m} \\ \{\mathbf{U}_k\}_{lm} = \{\mathbf{H}\}_{l+1,m} \end{array}$

$l, m = 1, \ldots, k$;

$\lambda_i = \dfrac{\beta_{i-1}}{1 + \mathbf{d}_k^t\, \mathbf{p}}$; $\mathbf{u}_k = \lambda_i\, \mathbf{p}$;

$\mathbf{x}_i = \mathbf{x}_{i-1} + \mathbf{Z}_k \mathbf{u}_k$;

being $\mathbf{Z}_k = [\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_k]$;

$\mathbf{r}_i = \mathbf{Z}_{k+1} \hat{\mathbf{r}}_i$;

with $\begin{array}{l} \{\hat{\mathbf{r}}_i\}_1 = \lambda_i \\ \{\hat{\mathbf{r}}_i\}_{l+1} = -\lambda_i \{\bar{\mathbf{p}}\}_l \end{array}$

$l = 1, \ldots, k$;

End

Nonsymmetric linear systems

Preconditioned Bi-CGSTAB algorithm

Initial guess $\mathbf{x}_0$. $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$;

Choose any $\mathbf{r}_0^*$ such that $\mathbf{r}_0, \quad \mathbf{r}_0^* \neq 0$;

$\mathbf{z}_0 = \mathbf{M}\mathbf{r}_0$;

$\mathbf{p}_0 = \mathbf{z}_0$;

While $\| \mathbf{r}_{j-1} \| \,/\, \| \mathbf{r}_0 \| \geq \varepsilon \quad (j = 1, 2, 3, \ldots)$, do

$$\mathbf{z}_j = \mathbf{M}\mathbf{r}_j;$$

$$\mathbf{y}_j = \mathbf{A}\mathbf{p}_j;$$

$$\mathbf{v}_j = \mathbf{M}\mathbf{y}_j;$$

$$\alpha_j = \frac{\langle \mathbf{z}_j, \mathbf{r}_0^* \rangle}{\langle \mathbf{v}_j, \mathbf{r}_0^* \rangle};$$

$$\mathbf{s}_j = \mathbf{r}_j - \alpha_j \mathbf{y}_j;$$

$$\mathbf{u}_j = \mathbf{A}\mathbf{s}_j;$$

$$\mathbf{t}_j = \mathbf{M}\mathbf{u}_j;$$

$$\tilde{\omega}_j = \frac{\langle \mathbf{t}_j, \mathbf{s}_j \rangle}{\langle \mathbf{t}_j, \mathbf{t}_j \rangle};$$

$$\mathbf{x}_{j+1} = \mathbf{x}_j + \alpha_j \mathbf{p}_j + \tilde{\omega}_j \mathbf{u}_j;$$

$$\mathbf{z}_{j+1} = \mathbf{s}_j - \tilde{\omega}_j \mathbf{t}_j;$$

$$\beta_j = \frac{\langle \mathbf{z}_{j+1}, \mathbf{r}_0^* \rangle}{\langle \mathbf{z}_j, \mathbf{r}_0^* \rangle} \, \frac{\alpha_j}{\tilde{\omega}_j};$$

$$\mathbf{p}_{j+1} = \mathbf{z}_{j+1} + \beta_j \left(\mathbf{p}_j - \tilde{\omega}_j \mathbf{v}_j \right);$$

End

Nonsymmetric linear systems

Preconditioned QMRCGSTAB algorithm

Initial guess $\mathbf{x}_0$. $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0$;

$\mathbf{z}_0 = \mathbf{M}\mathbf{r}_0$;

Choose $\tilde{\mathbf{r}}_0$ such that $\langle \mathbf{z}_0, \tilde{\mathbf{r}}_0 \rangle \neq 0$

$\mathbf{p}_0 = \mathbf{v}_0 = \mathbf{d}_0 = \mathbf{0}$, $\rho_0 = \alpha_0 = \tilde{\omega}_0 = 1$;

$\tau_0 = \|\mathbf{z}_0\|$, $\theta_0 = 0$, $\eta_0 = 0$;

While $\sqrt{j+1}\,|\tilde{\tau}|\,/\,\|\mathbf{r}_0\| \geq \varepsilon (j = 1, 2, \ldots)$, do:

$\rho_j = \langle \mathbf{z}_{j-1}, \tilde{\mathbf{r}}_0 \rangle$;

$\beta_j = (\rho_j/\rho_{j-1})(\alpha_{j-1}/\tilde{\omega}_{j-1})$;

$\mathbf{p}_j = \mathbf{z}_{j-1} + \beta_j(\mathbf{p}_{j-1} - \tilde{\omega}_{j-1}\mathbf{v}_{j-1})$;

$\mathbf{y}_j = \mathbf{A}\mathbf{p}_j$; $\quad \mathbf{v}_j = \mathbf{M}\mathbf{y}_j$;

$\alpha_j = \rho_j / \langle \mathbf{v}_j, \tilde{\mathbf{r}}_0 \rangle$; $\quad \mathbf{s}_j = \mathbf{z}_{j-1} - \alpha_j \mathbf{v}_j$;

$\tilde{\theta}_j = \|\mathbf{s}_j\| / \tau$; $\quad c = \dfrac{1}{\sqrt{1+\tilde{\theta}_j^2}}$;

$\tilde{\tau} = \tau\tilde{\theta}_j c$; $\quad \tilde{\eta}_j = c_j^2 \alpha_j$;

$$\tilde{\mathbf{d}}_j = \mathbf{p}_j + \frac{\theta_{j-1}^2 \eta_{j-1}}{\alpha_j}\mathbf{d}_{j-1};$$

$\tilde{\mathbf{x}}_j = \mathbf{x}_{j-1} + \tilde{\eta}_j \tilde{\mathbf{d}}_j$;

$\mathbf{u}_j = \mathbf{A}\mathbf{s}_j$; $\quad \mathbf{t}_j = \mathbf{M}\mathbf{u}_j$;

$$\tilde{\omega}_j = \frac{\langle \mathbf{s}_j, \mathbf{t}_j \rangle}{\langle \mathbf{t}_j, \mathbf{t}_j \rangle};$$

$\mathbf{z}_j = \mathbf{s}_j - \tilde{\omega}_j \mathbf{t}_j$;

$\theta_j = \|\mathbf{z}_j\| / \tilde{\tau}$, $c = \dfrac{1}{\sqrt{1+\theta_j^2}}$;

$\tau = \tilde{\tau}\theta_j c$; $\quad \eta_j = c^2 \tilde{\omega}_j$;

$$\mathbf{d}_j = \mathbf{s}_j + \frac{\tilde{\theta}_j^2 \tilde{\eta}_j}{\tilde{\omega}_j}\tilde{\mathbf{d}}_j;$$

$\mathbf{x}_j = \tilde{\mathbf{x}}_j + \eta_j \mathbf{d}_j$;

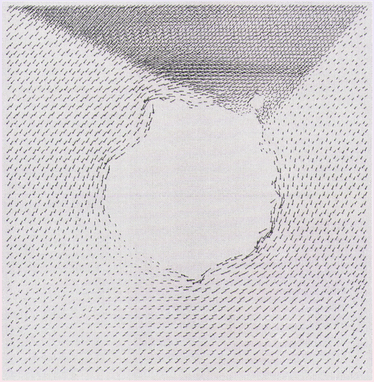
End

Convection-diffusion simulation

Sea pollution modelling facilities

- Evaluation of environmental impacts
- Location of pollutant sources
- Actuation in case of a pollutant disaster

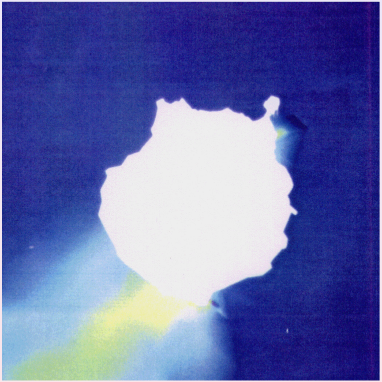
Convection-diffusion simulation



Simulation in the sea of Gran Canaria

- A 2-D problem
- The velocity field is computed with a mass consistent model
- A convection-diffusion problem

Convection-diffusion simulation



A convection-diffusion model

$$\vec{v}\cdot\vec{\nabla}c-\vec{\nabla}\cdot(K\vec{\nabla}c)=f \text{ in } \Omega$$

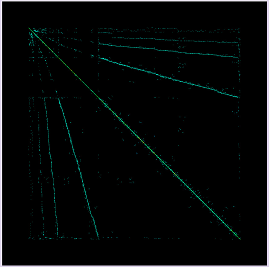
$$\vec{n}\cdot\left[\vec{v}c-K\vec{\nabla}c\right]=\vec{n}\cdot\vec{v}c^{\Gamma_{a0}} \text{ on } \Gamma_{a_0}\ (\vec{n}\cdot\vec{v}\le 0)$$

$$-\vec{n}\cdot K\vec{\nabla}c=0 \text{ on } \Gamma_{a_1}\ (\vec{n}\cdot\vec{v}>0)$$

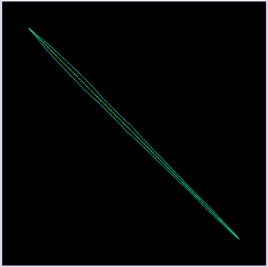
$$c=c^{e}(x,y,z,t) \text{ on } \Gamma_{b_0}$$

Convection-diffusion simulation

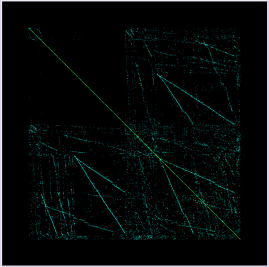
Sparsity patterns of a 12666 matrix



Initial Ordering



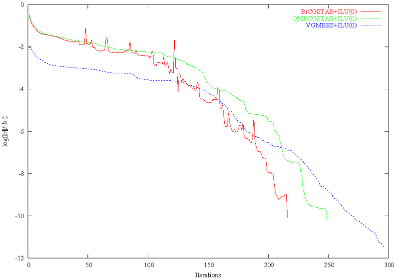
Reverse Cuthill McKee



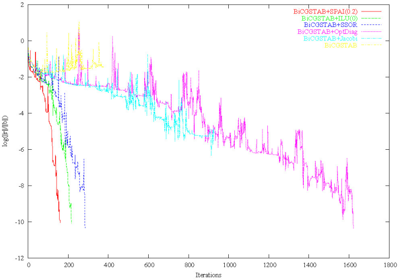
Minimum Degree

Convection-diffusion simulation

Convergence of Krylov Subspace Methods



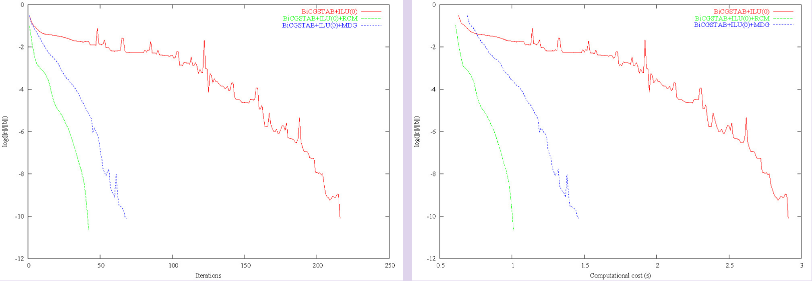
Performance of Krylov methods



Effect of preconditioning

Convection-diffusion simulation

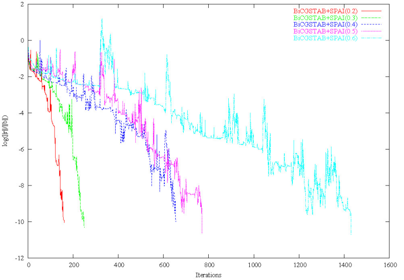
Convergence of Krylov Subspace Methods



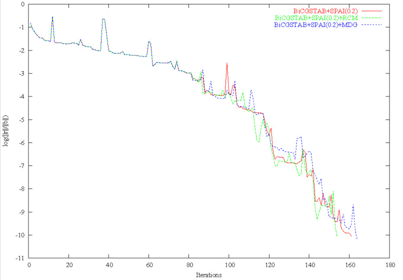
Effect of reordering on BiCGSTAB preconditioned with ILU(0)

Convection-diffusion simulation

Convergence of Krylov Subspace Methods



Performance of SPAI with several levels of fill-in



Effect of reordering on BiCGSTAB preconditioned with SPAI(0.2)

Conclusions

- Krylov subspace methods provides a wide set of possibilities for solving linear systems of equations. Although, in the symmetric case the selection is clear (CG), for nonsymmetric problems the choice depends on several factors: time-dependent or non-linear problems, available computer memory, parallel computation, very ill-conditioned problems, ...
- The available algorithms for computing a sparse approximate inverse M of a sparse nonsymmetric matrix A, may be implemented in parallel since the columns (or rows) of M are obtained independently. The sparsity pattern of these preconditioners are dynamically built. Evidently, this type of preconditioner may compete with implicit ones in a parallel environment.
- The reordering techniques slightly affect the convergence of conjugate gradient method preconditioned with an incomplete factorization. However, in nonsymmetric linear systems, the reordering improves the effect of preconditioning on the convergence of Krylov subspace methods. On the other hand, in most of the cases, the usual reorderings does not change the convergence behaviour of Krylov methods with SPAI, except for rounding errors.

Conclusions

- Krylov subspace methods provides a wide set of possibilities for solving linear systems of equations. Although, in the symmetric case the selection is clear (CG), for nonsymmetric problems the choice depends on several factors: time-dependent or non-linear problems, available computer memory, parallel computation, very ill-conditioned problems, ...
- The available algorithms for computing a sparse approximate inverse M of a sparse nonsymmetric matrix A, may be implemented in parallel since the columns (or rows) of M are obtained independently. The sparsity pattern of these preconditioners are dynamically built. Evidently, this type of preconditioner may compete with implicit ones in a parallel environment.
- The reordering techniques slightly affect the convergence of conjugate gradient method preconditioned with an incomplete factorization. However, in nonsymmetric linear systems, the reordering improves the effect of preconditioning on the convergence of Krylov subspace methods. On the other hand, in most of the cases, the usual reorderings does not change the convergence behaviour of Krylov methods with SPAI, except for rounding errors.

Conclusions

- Krylov subspace methods provides a wide set of possibilities for solving linear systems of equations. Although, in the symmetric case the selection is clear (CG), for nonsymmetric problems the choice depends on several factors: time-dependent or non-linear problems, available computer memory, parallel computation, very ill-conditioned problems, ...
- The available algorithms for computing a sparse approximate inverse M of a sparse nonsymmetric matrix A, may be implemented in parallel since the columns (or rows) of M are obtained independently. The sparsity pattern of these preconditioners are dynamically built. Evidently, this type of preconditioner may compete with implicit ones in a parallel environment.
- The reordering techniques slightly affect the convergence of conjugate gradient method preconditioned with an incomplete factorization. However, in nonsymmetric linear systems, the reordering improves the effect of preconditioning on the convergence of Krylov subspace methods. On the other hand, in most of the cases, the usual reorderings does not change the convergence behaviour of Krylov methods with SPAI, except for rounding errors.

Future research

- More research must be carried out on the effect of other reordering techniques which take into account the values of the entries in A and not only their positions. Although these techniques are expensive, in the case of unsteady problems that yield a linear system with multiple right hand side, this techniques may compete in parallel machines.
- We have preliminary results of a SSPAI for symmetric problems, where some difficulties about parallel computation and dynamical sparsity pattern have been found.
- In our wind model we obtain in each step i a linear system of equations as $A_i x_i = b_i$, where $A_i = A_1 + \alpha_i A_2$. Since A_1 and A_2 are the same along the whole process, it would be interesting to construct a preconditioner that could be updated as a function of α_i.
- Finally, we are now solving time-dependent (convection-diffusion-reaction) problems which lead to thousand of linear systems of equations. These equations must be solved in an efficient, fast and accurate way. Here, we shall have to use one suitable RPK strategy.

Future research

- More research must be carried out on the effect of other reordering techniques which take into account the values of the entries in A and not only their positions. Although these techniques are expensive, in the case of unsteady problems that yield a linear system with multiple right hand side, this techniques may compete in parallel machines.
- We have preliminary results of a SSPAI for symmetric problems, where some difficulties about parallel computation and dynamical sparsity pattern have been found.
- In our wind model we obtain in each step i a linear system of equations as $A_i x_i = b_i$, where $A_i = A_1 + \alpha_i A_2$. Since A_1 and A_2 are the same along the whole process, it would be interesting to construct a preconditioner that could be updated as a function of α_i.
- Finally, we are now solving time-dependent (convection-diffusion-reaction) problems which lead to thousand of linear systems of equations. These equations must be solved in an efficient, fast and accurate way. Here, we shall have to use one suitable RPK strategy.

Future research

- More research must be carried out on the effect of other reordering techniques which take into account the values of the entries in A and not only their positions. Although these techniques are expensive, in the case of unsteady problems that yield a linear system with multiple right hand side, this techniques may compete in parallel machines.
- We have preliminary results of a SSPAI for symmetric problems, where some difficulties about parallel computation and dynamical sparsity pattern have been found.
- In our wind model we obtain in each step i a linear system of equations as $A_i x_i = b_i$, where $A_i = A_1 + \alpha_i A_2$. Since A_1 and A_2 are the same along the whole process, it would be interesting to construct a preconditioner that could be updated as a function of α_i.
- Finally, we are now solving time-dependent (convection-diffusion-reaction) problems which lead to thousand of linear systems of equations. These equations must be solved in an efficient, fast and accurate way. Here, we shall have to use one suitable RPK strategy.

Future research

- More research must be carried out on the effect of other reordering techniques which take into account the values of the entries in A and not only their positions. Although these techniques are expensive, in the case of unsteady problems that yield a linear system with multiple right hand side, this techniques may compete in parallel machines.
- We have preliminary results of a SSPAI for symmetric problems, where some difficulties about parallel computation and dynamical sparsity pattern have been found.
- In our wind model we obtain in each step i a linear system of equations as $A_i x_i = b_i$, where $A_i = A_1 + \alpha_i A_2$. Since A_1 and A_2 are the same along the whole process, it would be interesting to construct a preconditioner that could be updated as a function of α_i.
- Finally, we are now solving time-dependent (convection-diffusion-reaction) problems which lead to thousand of linear systems of equations. These equations must be solved in an efficient, fast and accurate way. Here, we shall have to use one suitable RPK strategy.