Iterative Methods and Multigrid

Part 3: Preconditioning

Preconditioning

The general idea behind preconditioning is that convergence of some method for the linear system $Ax = b$ can be improved by applying the method to the preconditioned system

1) $P^{-1}Ax = P^{-1}b$ or
2) $AP^{-1}\tilde{x} = b$ and $x = P^{-1}\tilde{x}$ or
3) $P_1^{-1}AP_2^{-1}\tilde{x} = P_1^{-1}b$ and $x = P_2^{-1}\tilde{x}$.

Although this is not really necessary, one easy way to think about preconditioning is that if

$P \approx A$ (case 1 and 2) or $P_1 P_2 \approx A$ (case 3) then the preconditioned system is close to the identity (in some appropriate sense) and convergence will be rapid.

Preconditioned Krylov methods

For non-Hermitian problems the Krylov method can be applied immediately to the preconditioned problem.

For Hermitian problems and corresponding solvers the preconditioned system must also be Hermitian (positive definite).

Two ways:
'Symmetric' preconditioning: $P^{-1}AP^{-H}\tilde{x} = P^{-1}b$,

Use inner product based on (HPD) preconditioner: $\langle x, y\rangle_P = y^H P x$
$P^{-1}A$ is Hermitian (self-adjoint) wrt this inner product.

$\langle P^{-1}Ax, y\rangle_P = y^H P P^{-1} A x = y^H A^H x = y^H A^H P^{-H} P^H x = \langle x, P^{-1}Ay\rangle_P$.

Now apply CG to $P^{-1}A$ but use P-inner product.

Preconditioners from splittings (1)

One way to derive preconditioners is to use basic iterative methods (fixed point methods).

Consider $Ax = b$ and use matrix splitting:

$$A = M - N \longrightarrow Mx_{k+1} = Nx_k + b \Leftrightarrow x_{k+1} = M^{-1}Nx_k + M^{-1}b$$

Converges iff $\rho(M^{-1}N) < 1$

Fixpoint: $x = M^{-1}Nx + M^{-1}b \Leftrightarrow (I - M^{-1}N)x = M^{-1}b$

$(I - M^{-1}N) = M^{-1}A$ is preconditioned matrix.

$(I - M^{-1}N)x = M^{-1}Ax = M^{-1}b$ is preconditioned system

Preconditioners from splittings (2)

Some well-known choices for M:

$$A = \begin{pmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$

$M = diag(A_{11}, A_{22}, A_{33})$ (block) Jacobi

$$M = \begin{pmatrix} A_{11} & & \\ A_{21} & A_{22} & \\ A_{31} & A_{32} & A_{33} \end{pmatrix}$$ (block) Gauss-Seidel

$$M = \begin{pmatrix} \omega^{-1} A_{11} & & \\ A_{21} & \omega^{-1} A_{22} & \\ A_{31} & A_{32} & \omega^{-1} A_{33} \end{pmatrix}$$ (block) SOR

$$x_{k+1}^{SOR} = \omega x_{k+1}^{GS} + (1-\omega) x_k^{SOR}$$

Preconditioners from splittings (3)

Properly chosen ω yields greatly improved convergence rate.

For example, for the Poisson equation $\Delta u = f$ the convergence rate goes from $1 - O(h^2)$ for $\omega = 1$ (Gauss-Seidel) to $1 - O(h)$.

For analysis see book.
For special problems SOR (as a method in itself) is still very popular.

For symmetric problems we need symmetric preconditioner:

$$M = \frac{\omega}{2-\omega} \begin{pmatrix} \omega^{-1} A_{11} & & \\ A_{21} & \omega^{-1} A_{22} & \\ A_{31} & A_{32} & \omega^{-1} A_{33} \end{pmatrix} \begin{pmatrix} A_{11}^{-1} & & \\ & A_{22}^{-1} & \\ & & A_{33}^{-2} \end{pmatrix} \begin{pmatrix} \omega^{-1} A_{11} & A_{12} & A_{13} \\ & \omega^{-1} A_{22} & A_{23} \\ & & \omega^{-1} A_{33} \end{pmatrix}$$

(block) SSOR

Diagonal preconditioning

If A is HPD and $A = \begin{pmatrix} I_{n_1} & B \\ B^H & I_{n_2} \end{pmatrix}$.

Then $\kappa(A) \le \kappa(D^H A D)$ for any nonsingular $D = diag(D_{n_1}, D_{n_2})$.

If a HPD matrix has all diagonal elements equal, then $\kappa(A) \le m.\min_D \kappa(D^H A D)$ for all D diagonal and HPD and m is max number of nonzeros in any row of A. If A is HPD and

$$A = \begin{pmatrix} I_{n_1} & A_{12} & \cdots & A_{1m} \\ A_{12}^H & I_{n_2} & & \vdots \\ \vdots & & \ddots & A_{m-1m} \\ A_{1m}^H & \cdots & A_{m-1m}^H & I_{n_m} \end{pmatrix}.$$

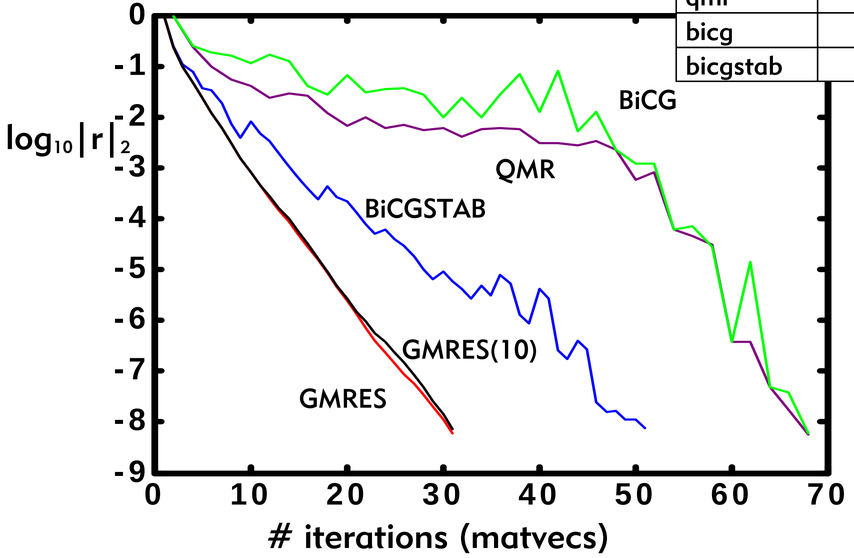
Then $\kappa(A) \le m.\min_D \kappa(D^H A D)$ for all nonsingular, block-diagonal D (same block sizes as A) and m is the number of diagonal blocks in A.

Unpreconditioned Convergence

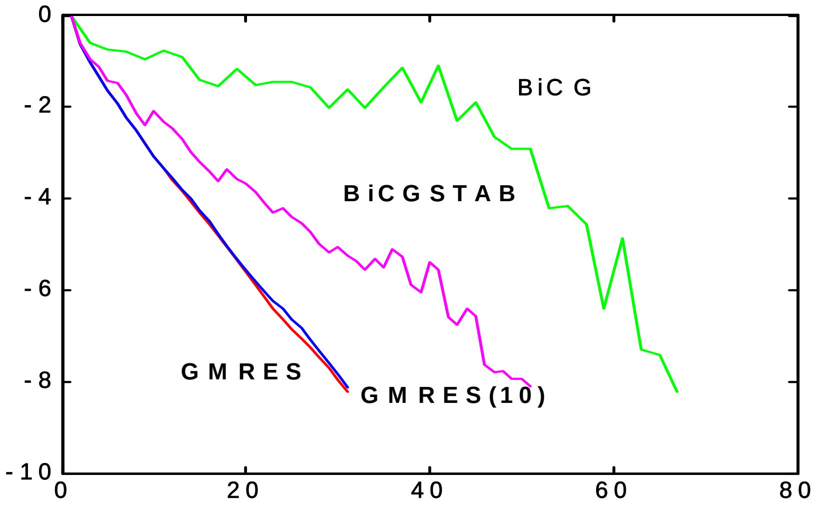
p=q=1; r=s=70; h=1/31;

u_s = 0; u_w =0; u_n = 1; u_e = 1;

	#matvec	time (s)
gmres	30	3.57
gmres(10)	40	0.5
qmr	66	0.38
bicg	66	0.28
bicgstab	50	0.33

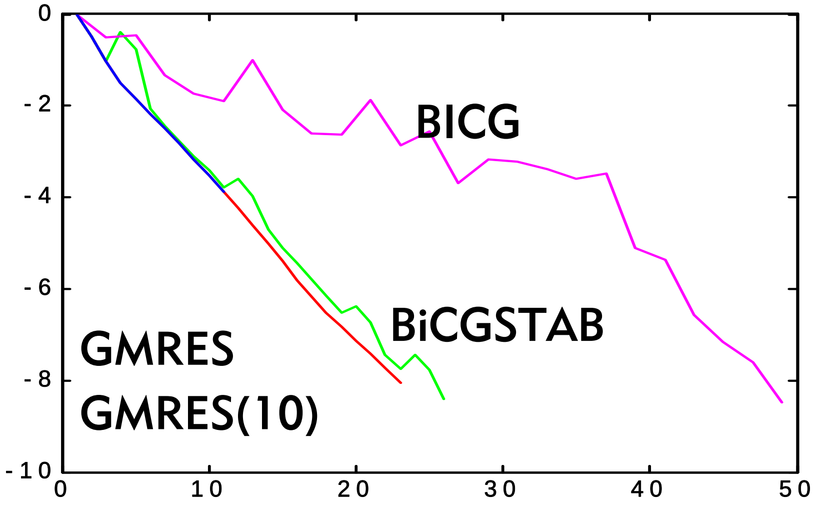


Diagonal preconditioner

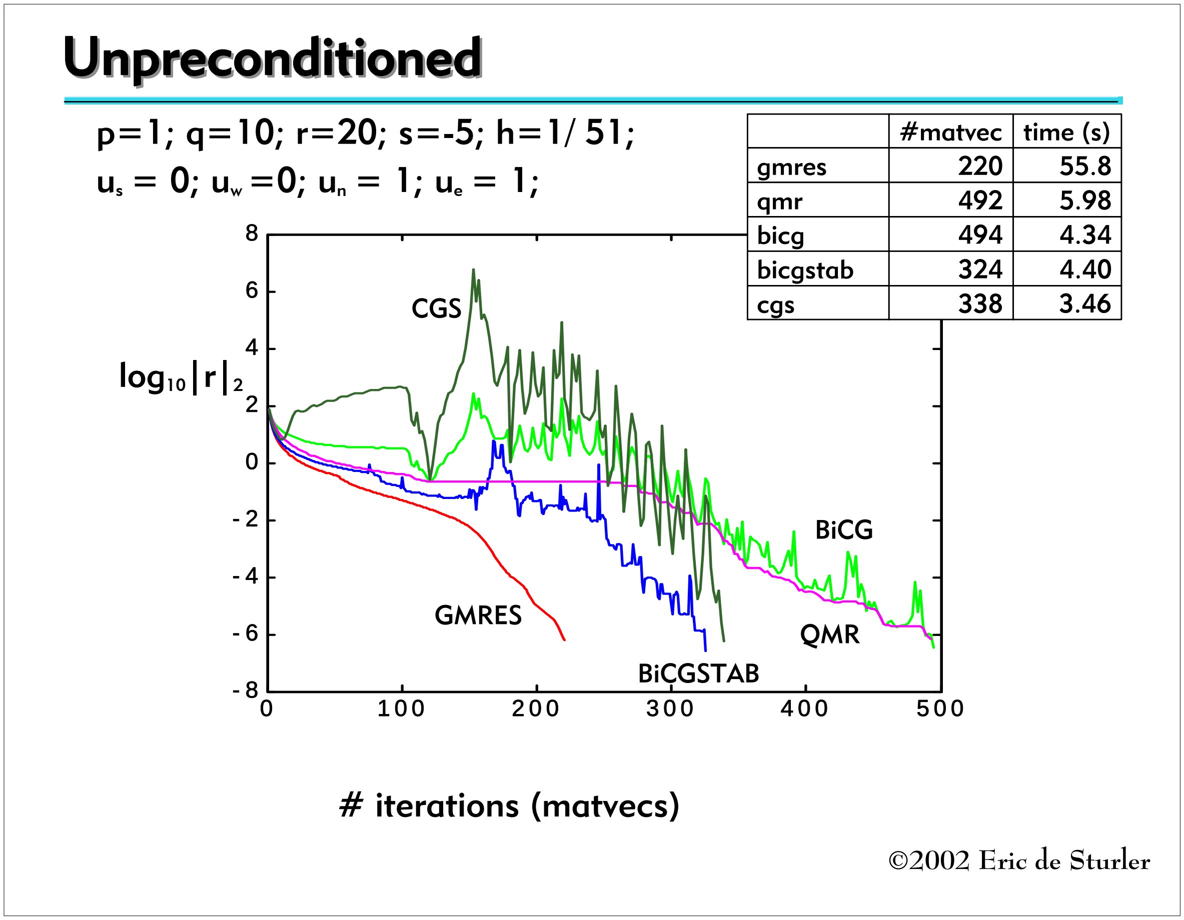


©2002 Eric de Sturler

Tridiagonal preconditiner



©2002 Eric de Sturler

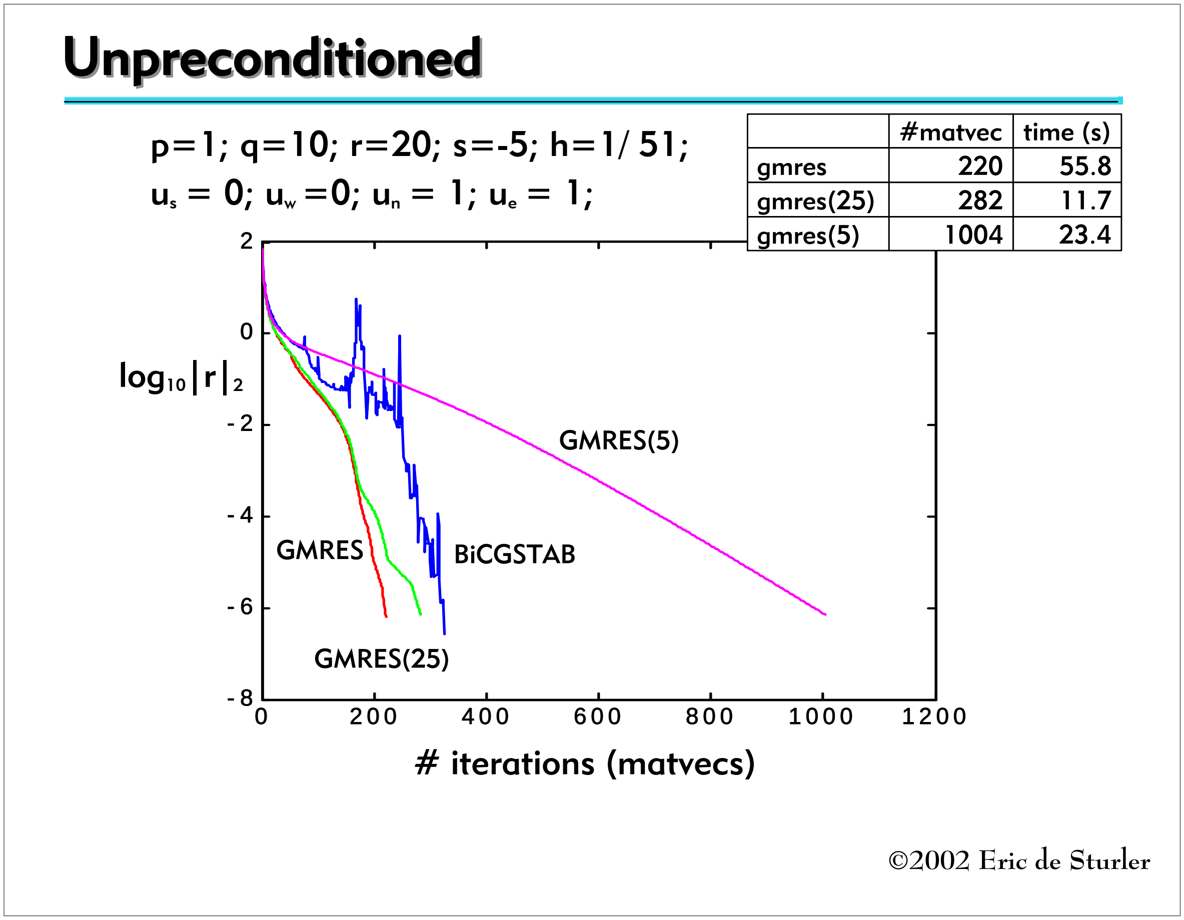


Unpreconditioned

p=1; q=10; r=20; s=-5; h=1/ 51;
u_s = 0; u_w =0; u_n = 1; u_e = 1;

	#matvec	time (s)
gmres	220	55.8
qmr	492	5.98
bicg	494	4.34
bicgstab	324	4.40
cgs	338	3.46

©2002 Eric de Sturler



Unpreconditioned

p=1; q=10; r=20; s=-5; h=1/ 51;
u_s = 0; u_w =0; u_n = 1; u_e = 1;

	#matvec	time (s)
gmres	220	55.8
gmres(25)	282	11.7
gmres(5)	1004	23.4

©2002 Eric de Sturler

Incomplete Decompositions

An important class of preconditioners is the one based on incomplete decompositions.

Consider the linear system $Ax = b$.

We carry out an LU decomposition (for example). However, at every point where we introduce an non-zero coefficient in A we ignore it. Let $R = LU - A$. This gives, $A = LU - R$, which is a matrix splitting.

Clearly we won't get the exact LU decomposition this way, but it turns out that for M-matrices this splitting has nice properties.

Incomplete Decompositions

Theorem:
Let A be an M-matrix. Then for every subset P of off-diagonal indices, there exists a lower triangular matrix L with unit diagonal and an upper triangular matrix U such that $A = LU - R$, and $l_{ij} = 0$ if $(i,j) \in P$, $u_{ij} = 0$ if $(i,j) \in P$, and $r_{ij} = 0$ if $(i,j) \notin P$. The factors L and U are unique and $A = LU - R$ is a regular splitting.

Proof: (see book)

If A is a symmetric M-matrix and $(j,i) \in P \Leftrightarrow (i,j) \in P$ then we have an incomplete Cholesky decomposition $A = LL^T - R$ which is a regular splitting.

The incomplete decompositions can be used as preconditioners. If we allow no fill-in we call the preconditioners ILU(0), IC(0) respectively.

Incomplete Decompositions

For hard problems it may be useful to allow more fill-in (non-zeros in L or U that are zeros in A).

The corresponding preconditioners are referred to as ILU(k), IC(k), where k indicates the level of fill-in allowed.
Level 1 fill-in is fill-in caused by non-zero coefficients of A. Level 2 fill-in is fill-in caused by the level 1 fill-in (that is by non-zeros not originally in A), etc.

Another variation of ILU is to use a drop tolerance. That is we only allow fill-in if, in absolute sense, it is larger than the drop tolerance times the norm of the column. The two approaches can also be mixed.

Incomplete Decompositions

The IC preconditioner tends to reduce the number of iterations significantly. However, for the mesh width h going to zero, the condition number of an elliptic problem still behaves as $O(h^{-2})$, just as for the unpreconditioned problem. Hence, the number of iterations of CG still behaves as $O(h^{-1})$.

It turns out we can improve the condition number of the preconditioned matrix by a factor of h, that is it behaves as $O(h)$.
Consequently, the number of iterations of CG behaves as $O(h^{-1/2})$.
For details see the book. This type of preconditioner is referred to as MIC (or MILU) for Modified Incomplete Cholesky (or Modified ILU).

Often a parametrized form of is used, where based on the parameter the preconditioner varies between IC (ILU) and MIC (MILU).

Incomplete Decompositions

MIC/MILU have about the same complexity to compute and to apply as IC/MILU.

It is important to keep in mind that preconditioning introduces extra costs:
1) the cost to compute the incomplete decomposition,
2) the cost to apply the preconditioner at each iteration.

The second is typically the most important. For example, ILU(0) introduces floating point overhead equal to another matrix vector product. In practice, especially on parallel (and vector) computers the extra cost in runtime can be even worse, because the solution of triangular systems is hard to parallelize (vectorize).

Hence, the preconditioner should reduce the number of iterations sufficiently (and it usually does).

Incomplete Decompositions

On parallel computers we often use incomplete block (diagonal) decompositions. This decouples parts of the domain. Unfortunately, this generally increases the number of iterations again, so that we must look for some balance. A variety of strategies to prevent deterioration of the number of iterations has been developed.
Some of these are related to so-called domain decomposition based preconditioners (next topic), sometimes referred to as interface preconditioners.

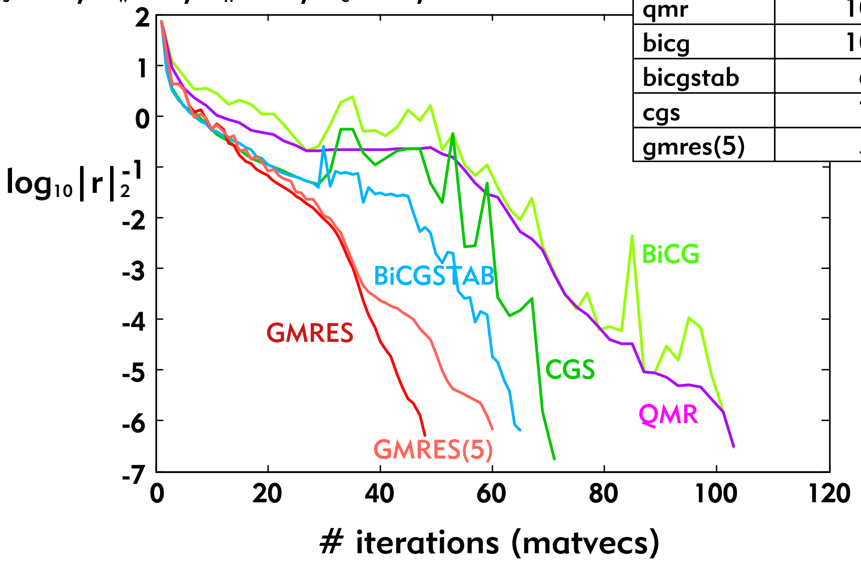
Nevertheless incomplete decompositions are the most popular preconditioners today. Largely, because they can more or less be used as black box preconditioners and are nevertheless very effective for a wide range of problems.

ILU(0) Preconditioner

p=1; q=10; r=20; s=-5; h=1/ 51;

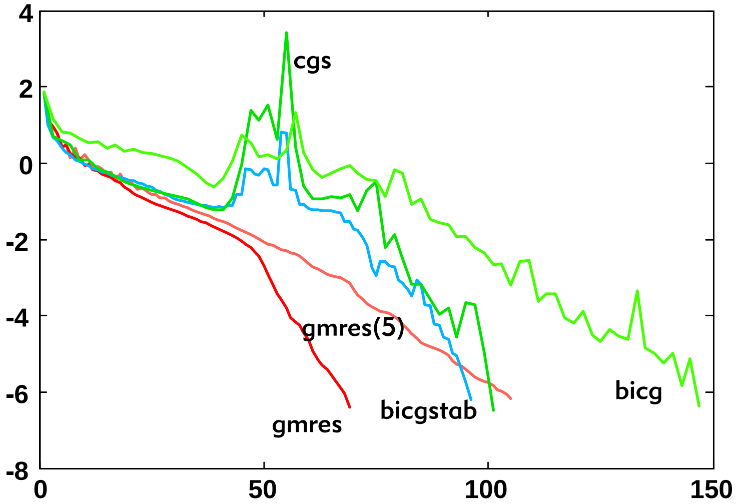
u_s = 0; u_w =0; u_n = 1; u_e = 1;

	#matvec	time (s)
gmres	47	5.83
qmr	101	3.95
bicg	101	3.52
bicgstab	64	2.58
cgs	70	2.41
gmres(5)	59	3.03



©2002 Eric de Sturler

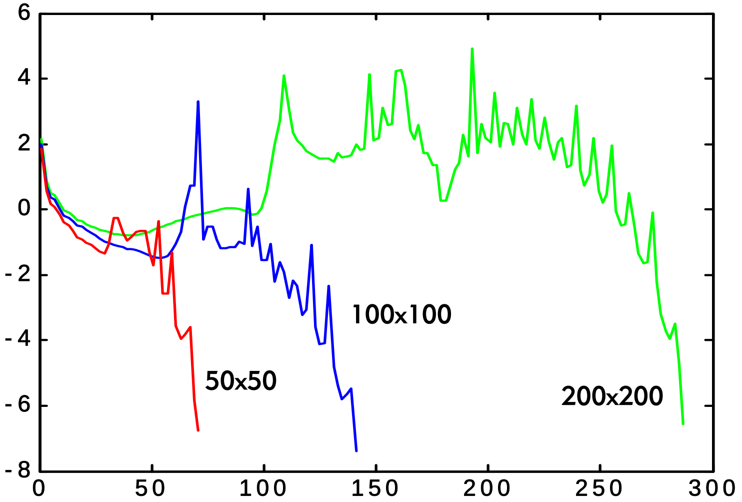
MILU(0)



©2002 Eric de Sturler

Convergence CGS for various h

	#matvec	time (s)
50x50	70	2.41
100x100	70	18.1
200x200	143	172


6
4
2
0
-2
-4
-6
-8
0
50
100
150
200
250
300
50x50
100x100
200x200