Krylov subspace methods

Hermitian Problems

Eric de Sturler
Department of Computer Science
University of Illinois at Urbana-Champaign
sturler@cs.uiuc.edu www-faculty.cs.uiuc.edu/sturler

MINRES

Consider again how GMRES builds an orthogonal basis for the Krylov space $K^{m+1}(A, r_0)$:

$v_1 = r_0 / \| r_0 \|_2;$
for $k = 1 : m,$
  $\tilde{v}_{k+1} = A v_k;$
  for $j = 1 : k,$
    $h_{j,k} = v_j^H \tilde{v}_{k+1};$
    $\tilde{v}_{k+1} = \tilde{v}_{k+1} - h_{j,k} v_k;$
  end
  $h_{k+1,k} = \| \tilde{v}_{k+1} \|_2;$
  $v_{k+1} = \tilde{v}_{k+1} / h_{k+1,k};$
end

Verify that the (Arnoldi) algorithm generates the following recurrence:

$AV_m = V_{m+1} H_{m+1,m}.$

What does $H_{m+1,m}$ look like?

Prove V_{m+1} is orthogonal.

Note $H_{m+1,m} = V_{m+1}^H A V_m.$

$\text{range}(V_m) = K^m(A, r_0)$ and $\text{range}(V_{m+1}) = K^{m+1}(A, r_0)$. So both $\text{range}(U_m)$ and $\text{range}(C_m)$ from GCR contained in $\text{range}(V_{m+1})$.

MINRES

Now consider A being Hermitian: $A^H = A$

Another way to write the recurrence relation from Arnoldi:

$$AV_m = V_{m+1}\underline{H}_m = V_m H_m + v_{m+1} e_m^T h_{m+1,m},$$

where H_m is the upper $m \times m$ part of $\underline{H}_m$.

So, $V_m^H A V_m = V_m^H V_m H_m + V_m^H v_{m+1} e_m^T h_{m+1,m} = H_m$.

$(V_m^H A V_m)^H = V_m^H A^H V_m = V_m^H A V_m$ since $A^H = A$, and so

H_m must be Hermitian as well.

This has some important consequences ...

MINRES

A Hermitian upper Hessenberg matrix is tridiagonal!

This means that (in exact arithmetic) we need to orthogonalize each new vector Av_i only against the vectors v_{i-1} and v_i.

We could solve the least squares problem in the same way as for GMRES, except that we save on orthogonalizations (inner products and vector updates).

What is the computational cost of m iterations of GMRES?

Theorem: Let A be Hermitian and let $v_1, v_2, \dots, v_m$ be the vectors generated by the Arnoldi algorithm (so they span $K^m(A, v_1)$). Then $Av_i \perp v_1, v_2, \dots, v_{i-2}$ and so $Av_i \perp \text{span}\{v_1, v_2, \dots, v_{i-2}\}$.

Proof:

MINRES

The algorithm now proceeds as follows:

Lanczos recurrence: $AV_m = V_{m+1}\underline{T}_m$ (T for tridiagonal).
Lanczos is Arnoldi in the Hermitian case (2 orthogonalizations).

Solve $y_m = \arg\min \| r_0 - AV_m y \|_2$ just as in GMRES:

We have $AV_m = V_{m+1}\underline{T}_m = V_{m+1}\underline{Q}_m R_m$,
and we compute $y_m = R_m^{-1}\underline{Q}_m^H V_{m+1}^H r_0$ (solving least squares problem).
Every step we update the QR-decomposition of $\underline{T}_i$ and solve
$R_i y_i = \underline{Q}_i^H e_1 \| r_0 \|_2$.

At end we update $x_m = x_0 + V_m y_m$ and $r_m = r_0 - V_{m+1} y_m$.

Note that each step we only orthogonalize on previous 2 vectors.
What would seem an obvious improvement. Can we do that here?

MINRES

Since we only orthogonalize on the previous two vectors, we would like to discard the other vectors.

However, we need them for the update at the end.

Can we update every step and discard the vectors v_i?

The problem is that R_m changes and hence y_m changes (in general) completely. So we need all previous v_i.

We need a trick.

MINRES

A cunning plan:
Since y_m changes completely every step, apply a change of variables
Alternative for update $V_m y_m$:
Take $W_m = V_m R_m^{-1}$ and $\hat{y}_m = R_m y_m = R_m R_m^{-1} \underline{Q}_m^H e_1 \| r_0 \|_2 = \underline{Q}_m^H e_1 \| r_0 \|_2$.

Then $W_m \hat{y}_m = V_m y_m$ and each iteration only the last component of $\hat{y}_m$ changes. So we can update $W_m \hat{y}_m$ without keeping all w_i.

R_m from the Givens QR decomposition of a tridiagonal matrix is uppertriangular with 2 upper diagonals.

W_m columns are found by solving $W_m R_m = V_m$ each iteration.
So looking at the last (=the new) column we have:

$w_m r_{m,m} + w_{m-1} r_{m-1,m} + w_{m-2} r_{m-2,m} = v_m$, only w_m not known:
$w_m = r_{m,m}^{-1}(v_m - w_{m-1} r_{m-1,m} - w_{m-2} r_{m-2,m})$

MINRES

Update solution: $x_m = x_0 + V_m y_m = x_0 + W_m \hat{y}_m$

Since $\hat{y}_m$, contrary to y_m, changes only in its last position we can do the update iteration-wise:

$$x_m = x_0 + \sum_{i=1}^{m} w_i \hat{y}_{i,m} = x_0 + \sum_{i=1}^{m-1} w_i \hat{y}_{i,m} + w_m \hat{y}_{m,m} = x_{m-1} + w_m \hat{y}_{m,m}$$

How many vectors do we need to keep (indepedenf of # iterations)?

Do we need r_m to continue the iteration?

What would be an update formula for r_m?

MINRES

MINRES: $Ax = b$

choose $x_0 \rightarrow r_0 = b - Ax_0$ and tol, set $k = 0$;
while $\|r_k\| > tol$ do
$k = k + 1;$
$\tilde{v}_{k+1} = Av_k - t_{k,k}v_k - t_{k-1,k}v_{k-1};$
$t_{k+1,k} = \|\tilde{v}_{k+1}\|_2;\ v_{k+1} = \tilde{v}_{k+1}/t_{k+1,k};$

Update QR: $Q_{k+1} = Q_k G_k; \underline{R}_k = G_k^H(Q_k^H \underline{T}_k); \hat{y}_{k,k} = q_k^H \ell_1 \|r_0\|_2$
$\rightarrow \underline{Q}_k,\ R_k,\ \hat{y}_k \equiv \underline{Q}_k^H \ell_1 \|r_0\|_2;$
$w_k = r_{k,k}^{-1}(v_k - w_{k-1}r_{k-1,k} - w_{k-2}r_{k-2,k});$
$x_k = x_{k-1} + w_k \hat{y}_{k,k}$
end

Conjugate Gradients

Hermitian matrices: Error minimization in the A-norm

We are solving $Ax = b$ with initial guess $x_0 \rightarrow r_0 = b - Ax_0$ and $\hat{x}$ is the solution to $Ax = b$.
The error at iteration i is $e_i = \hat{x} - (x_0 + z_i)$, where $z_i \in K^i(A, r_0)$ is the ith update to the initial guess.

Theorem:
Let A be Hermitian, then the vector $z_i \in K^i(A, r_0)$ satisfies $z_i = \arg\min\{\|\hat{x} - (x_0 + z)\|_A : z \in K^i(A, r_0)\}$ iff $r_i \equiv r_0 - Az_i$ satisfies $r_i \perp K^i(A, r_0)$.

The most important algorithm of this class is the Conjugate Gradient Algorithm.

Conjugate Gradients

Proof:

$z_i = \arg\min\{\|\hat{x} - (x_0 + z)\|_A : z \in K^i(A, r_0)\} \iff (\hat{x} - x_0) - z_i \perp_A K^i(A, r_0)$

We know $K^i(A, r_0) = \text{span}\{r_0, r_1, \ldots, r_{i-1}\}$.
This gives $r_k \perp_A (\hat{x} - x_0 - z_i)$ for $k = 0, \ldots, i-1 \quad \Leftrightarrow$

$\langle A(\hat{x} - x_0 - z_i), r_k \rangle$ for $k = 0, \ldots, i-1 \quad \Leftrightarrow$

$\langle b - Ax_0 - Az_i, r_k \rangle$ for $k = 0, \ldots, i-1 \quad \Leftrightarrow$

$\langle r_0 - Az_i, r_k \rangle$ for $k = 0, \ldots, i-1 \quad \Leftrightarrow$

$\langle r_i, r_k \rangle$ for $k = 0, \ldots, i-1 \quad \Leftrightarrow$

$r_i \perp K^i(A, r_0)$

Conjugate Gradients

Lanczos iteration:
Choose q_1; $\beta_0 = 0$; $q_0 = 0$;
for $i = 1, 2, \ldots$ do
$\tilde{q}_{i+1} = Aq_i$;
$\alpha_i = \langle Aq_i, q_i \rangle$; $\tilde{q}_{i+1} = \tilde{q}_{i+1} - \alpha_i q_i$; $\tilde{q}_{i+1} = \tilde{q}_{i+1} - \beta_{i-1} q_{i-1}$;
$\beta_i = \|\tilde{q}_{i+1}\|_2$; $q_{i+1} = \tilde{q}_{i+1}/\beta_i$;
end

Show $\tilde{q}_{i+1} = \tilde{q}_{i+1} - \beta_{i-1} q_{i-1}$ sets $\tilde{q}_{i+1} \perp q_{i-1}$.
(one argument is the symmetry of the Hessenberg matrix for Arnoldi, give another)
This generates the recurrence relation:

$AQ_i = Q_i T_i + \beta_i q_{i+1} e_i^T$, where $Q_i = [q_1\ q_2 \cdots q_i]$, $T_i = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots \\ \beta_1 & \alpha_2 & \beta_2 & \ddots \\ 0 & \beta_2 & \ddots & \ddots \\ \vdots & \ddots & \ddots & \ddots \end{bmatrix}$.

Conjugate Gradients

Use Lanczos orthonormal basis for minimizing A-norm of error.
$z_i = \arg\min\{\|\hat{x} - (x_0 + z)\|_A : z \in K^i(A, r_0)\}$ iff $r_i \equiv r_0 - Az_i$ satisfies
$r_i \perp K^i(A, r_0)$.

$q_1 = r_0 / \|r_0\|_2$;
Lanczos method: $AQ_i = Q_i T_i + \beta_i q_{i+1} \ell_i^T$

Solve $r_0 - AQ_i y_i \perp Q_i \Leftrightarrow Q_i^H(\|r_0\|_2 q_1 - AQ_i y_i) = 0 \Leftrightarrow$
$Q_i^H(\|r_0\|_2 q_1 - AQ_i y_i) = 0 \Leftrightarrow \|r_0\|_2 \ell_1 - Q_i^H A Q_i y_i = 0$.

Notice $\mathrm{range}(Q_i) = \mathrm{span}\{r_0, r_1, \ldots, r_{i-1}\}$.
$AQ_i = Q_i T_i + \beta_i q_{i+1} \ell_i^T \Rightarrow Q_i^H A Q_i = T_i$

So we reduced the problem to solve $\|r_0\|_2 \ell_1 - T_i y_i = 0$:

$y_i = T_{i-1}^{-1} \ell_1 \|r_0\|_2$

Conjugate Gradients

In order to update step-by-step we use same trick as in MINRES:

Let $T_i = L_i D_i L_i^H$ then $y_i = L_i^{-H} D_i^{-1} L_i^{-1} \ell_1 \|r_0\|_2$, where L_i is unit lower bi-diagonal with lower diagonal coeff.s $l_1, l_2, \ldots, l_{i-2}$, index → column

Change of variables:
$P_i = Q_i L_i^{-H}$ and $\hat{y}_i = D_i^{-1} L_i^{-1} \ell_1 \|r_0\|_2$: $Q_i y_i = P_i \hat{y}_i$

Notice that each iteration only the last component of $\hat{y}_i$ changes.
From $P_i L_i^H = Q_i$ we get a recurrence for p_i: $p_i + l_{i-1} p_{i-1} = q_i \ (p_1 = q_1)$

So every new step we compute a new q_{i+1}, we update the decomposition of T_i and from that $\hat{y}_{i+1}$ and p_{i+1}.

$x_i = x_{i-1} + p_i \hat{y}_{i,i}$
$r_i = r_{i-1} - Ap_i \hat{y}_{i,i} = q_{i+1} \beta_i \hat{y}_{i,i}$ (where $\hat{y}_{i,i}$ is ith comp of vector $\hat{y}_i$)

Conjugate Gradients

(Easier form of) CG algorithm: $Ax = b$

Choose $x_0 \rightarrow r_0 = b - Ax_0$;

$p_1 = r_0$; $i = 0$

while $\|r_i\|_2 > tol$ do

$i = i + 1$;

$\alpha_i = \frac{\langle r_{i-1}, r_{i-1} \rangle}{\langle p_{i-1}, Ap_{i-1} \rangle}$;

$x_i = x_i + \alpha_i p_i$;

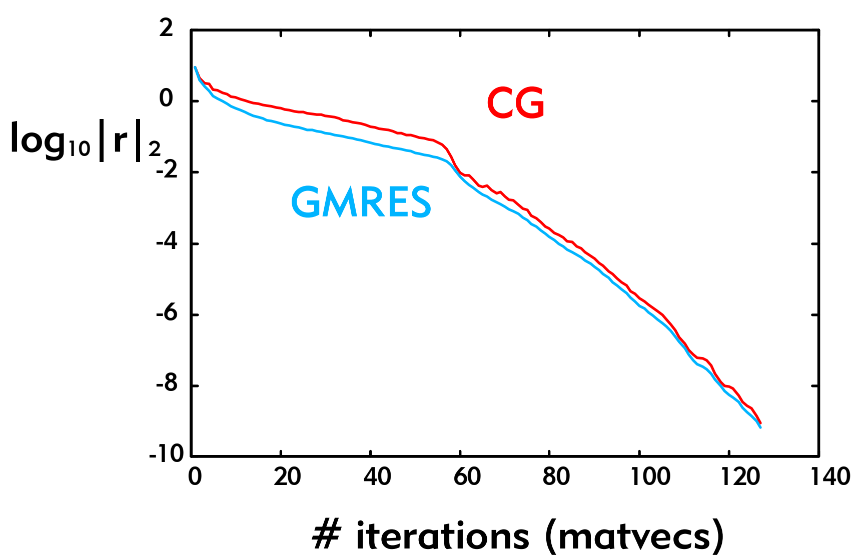
$r_i = r_{i-1} - \alpha_i A p_i$;

$\beta_i = \frac{\langle r_i, r_i \rangle}{\langle r_{i-1}, r_{i-1} \rangle}$;

$p_i = r_i - \beta_i p_{i-1}$;

end

Conjugate Gradients



Krylov subspace methods

Comparing Methods

Eric de Sturler
Department of Computer Science
University of Illinois at Urbana-Champaign
sturler@cs.uiuc.edu www-faculty.cs.uiuc.edu/sturler

GMRES

GMRES: $Ax = b$
choose x_0 (e.g. $x_0 = 0$) and tol

$r_0 = b - Ax_0;\ k = 0;\ v_1 = r_0/\|r_0\|_2;$
while $\|r_k\|_2 > tol$
  $k = k + 1;$
  $\tilde{v}_{k+1} = Av_k;$
  for $j = 1 : k,$
    $h_{j,k} = v_j^H \tilde{v}_{k+1};\ \tilde{v}_{k+1} = \tilde{v}_{k+1} - h_{j,k} v_k;$
  end
  $h_{k+1,k} = \|\tilde{v}_{k+1}\|_2;\ v_{k+1} = \tilde{v}_{k+1}/h_{k+1,k};$
  update QR-dec: $\underline{H}_k = Q_{k+1}\underline{R}_k$
  $\|r_k\|_2 = |\tilde{q}_{k+1}^H e_1| \|r_0\|_2$
end
$y_k = R_k^{-1} \underline{Q}_k^H e_1 \|r_0\|_2;\ x_k = x_0 + V_k y_k;$
$r_k = r_0 - V_{k+1}\underline{H}_k y_k = V_{k+1}\left(I - \underline{Q}_k \underline{Q}_k^H\right) e_1 \|r_0\|_2;$ (or simply $r_k = b - Ax_k$)

Iterative Methods: Cost

- **Many Cheap Iterations versus Minimum Number of Expensive Iterations**
  - same as sequential but issues determining cost change

- four main kernels
  - matrix-vector product: comp: 2*N*nz1 comm: “neighbour”
  - preconditioner: comp: 2*N*nz2 comm: “neighbour” (& global)
  - vector update: comp: 2*N comm: none
  - inner product: comp: 2*N comm: global

  - Methods
    - GMRES, GCR, FOM, BiCG, CGS, BiCGSTAB(l)
    - short recurrence: cheap iteration / many iterations
    - full orthogonalization: minimal number of iterations / expensive

  - Matrix vector product often linked with grid/domain partitioning
    - partition scheme to minimize comm. volume/number of messages
    - separate local and nonlocal references
    - overlap communication (latency hiding)

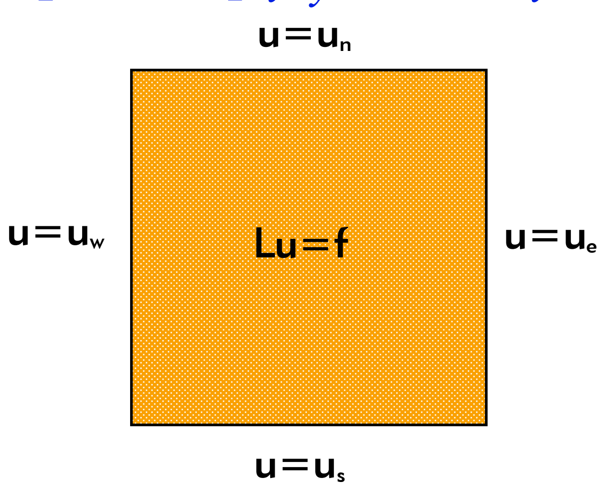
 Swiss Center for Scientific Computing

Model Problem

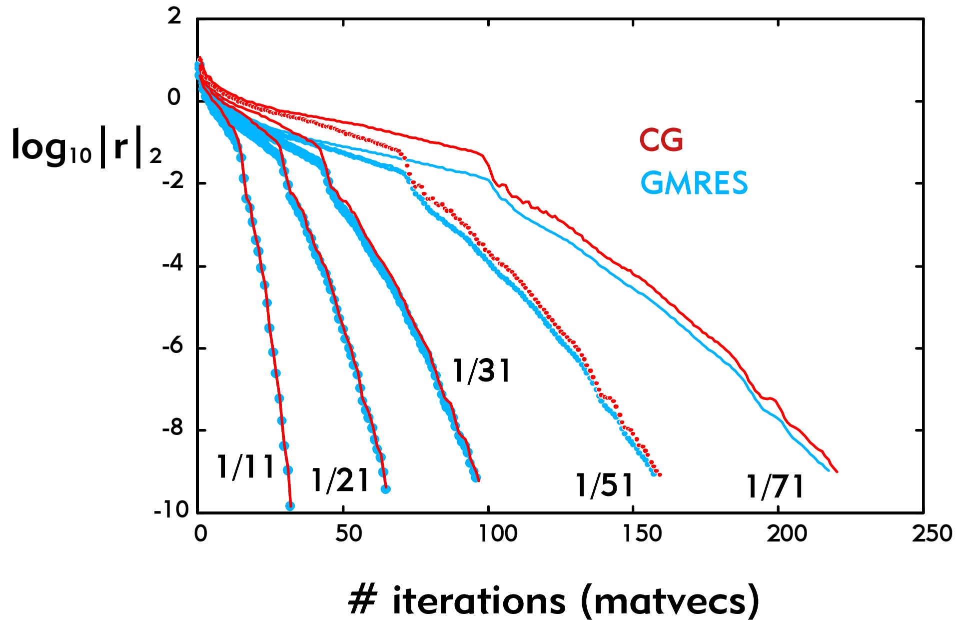
Convection-Diffusion(-Reaction) Equation

Dirichlet boundary conditions

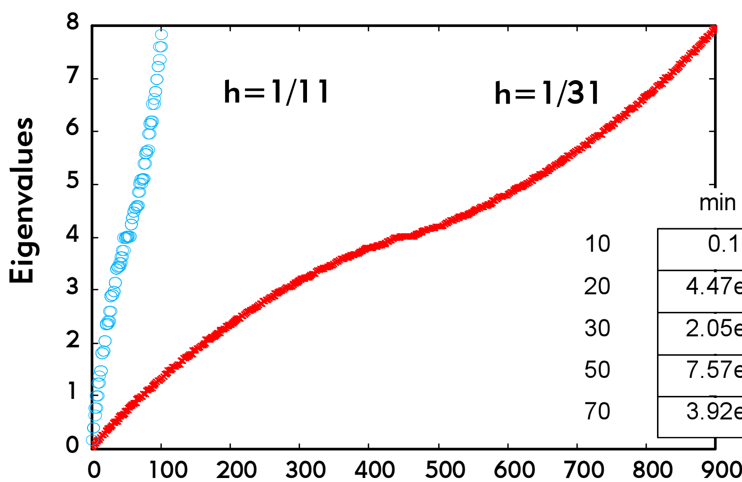
$$Lu = -(pu_x)_x - (qu_y)_y + ru_x + su_y + tu = f$$



CG vs GMRES for various mesh widths (h)



Eigenvalues for various h

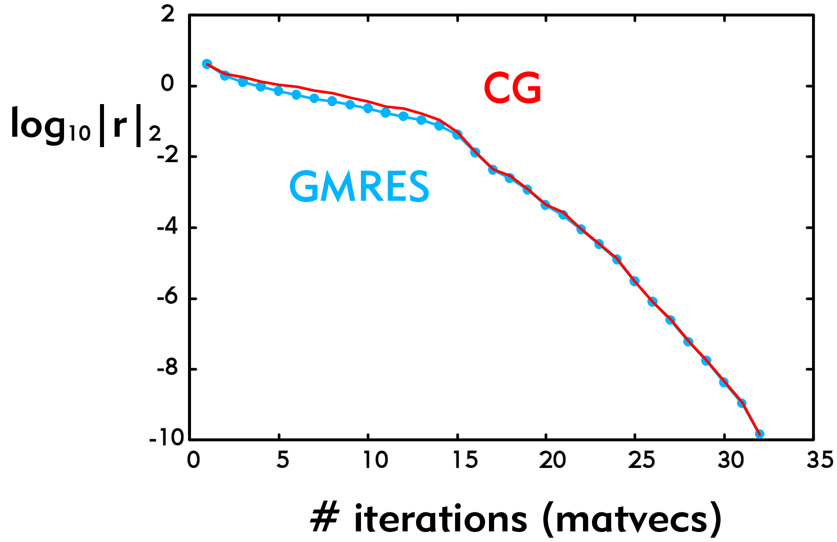


	min	max	cond. nr.
10	0.162	7.84	48.4
20	4.47e-2	7.95	178
30	2.05e-2	7.98	389
50	7.57e-3	7.99	1.06e3
70	3.92e-3	8.00	2.04e3

CG vs GMRES

p=q=1; t = 0; f = 0; h=1/ 11;

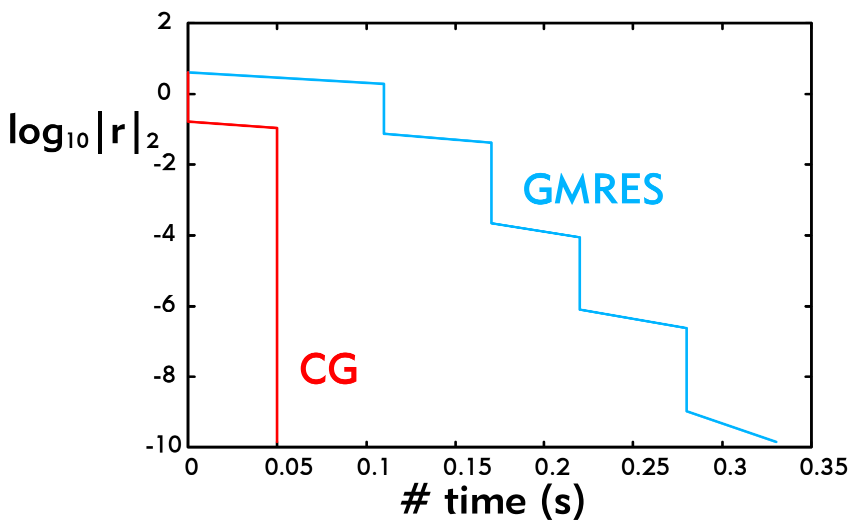
u_s = 0; u_w =1; u_n = 1; u_e = 0;



CG vs GMRES

p=q=1; t = 0; f = 0; h=1/ 11;

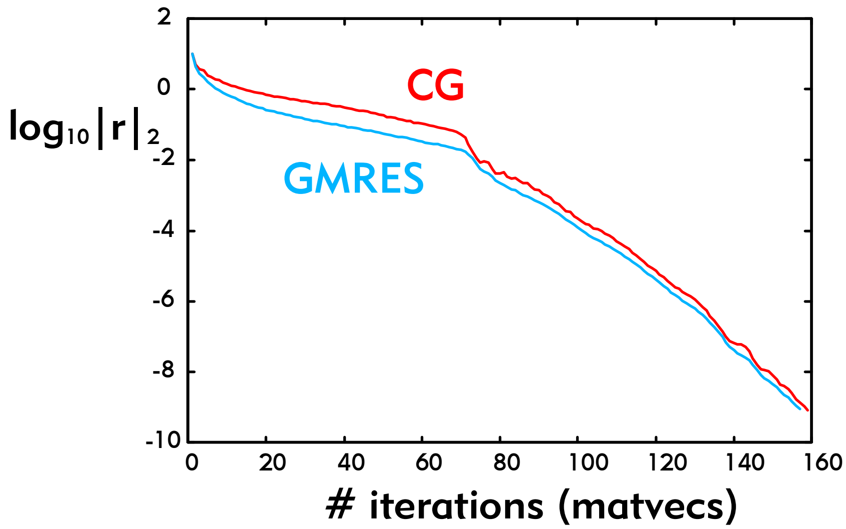
u_s = 0; u_w =1; u_n = 1; u_e = 0;



CG vs GMRES

$p=q=1;\ t = 0;\ f = 0;\ h=1/51;$

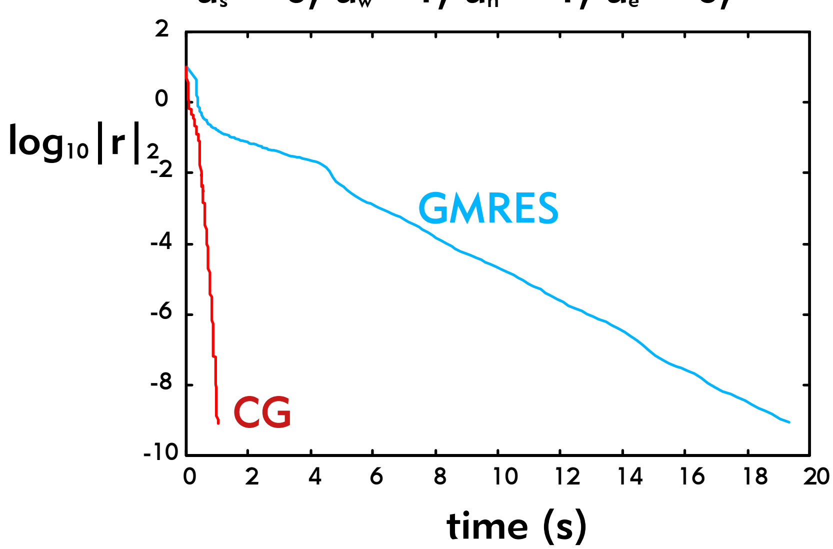
$u_s = 0;\ u_w = 1;\ u_n = 1;\ u_e = 0;$



CG vs GMRES (time)

$p=q=1;\ t = 0;\ f = 0;\ h=1/51;$

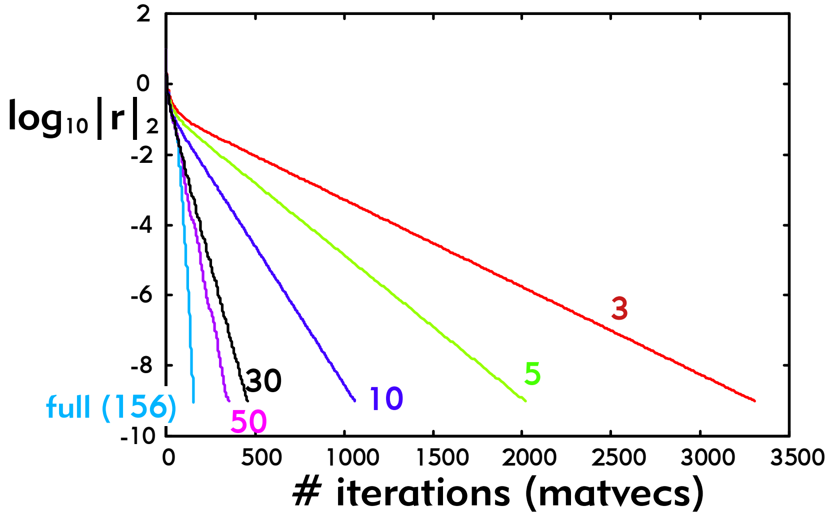
$u_s = 0;\ u_w = 1;\ u_n = 1;\ u_e = 0;$



Iterations for GMRES(m)

p=q=1; t = 0; f = 0; h=1/ 51;

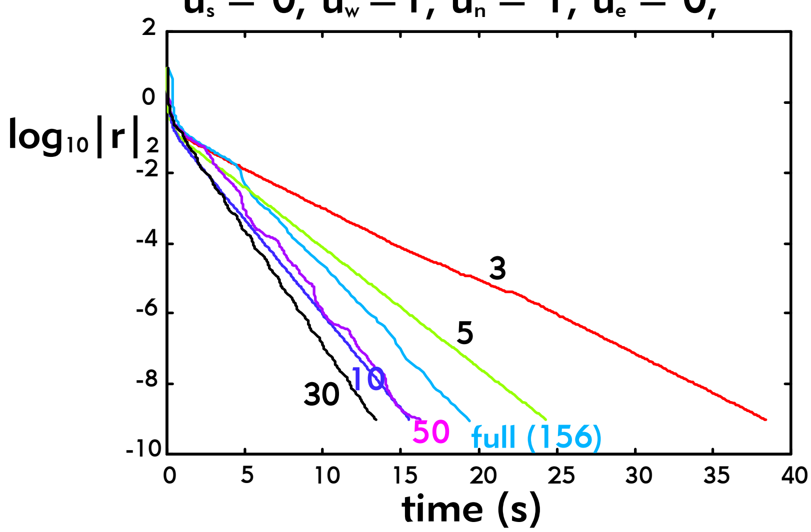
u_s = 0; u_w =1; u_n = 1; u_e = 0;



Time for GMRES(m)

p=q=1; t = 0; f = 0; h=1/ 51;

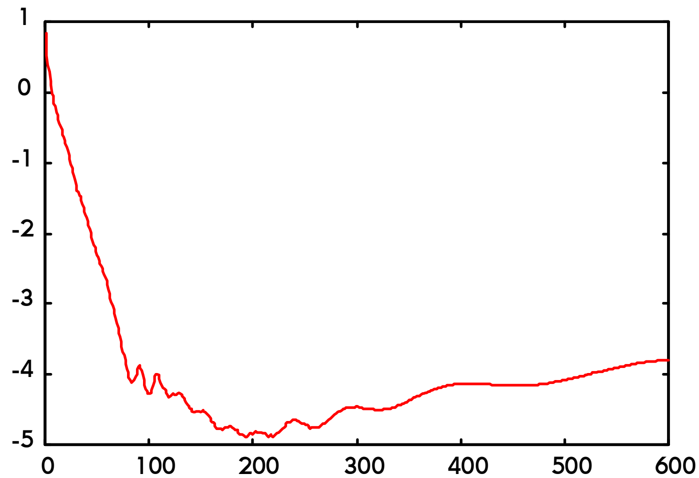
u_s = 0; u_w =1; u_n = 1; u_e = 0;



CG for a non-Hermitian Problem

p=q=1; r=s=5; h=1/31;

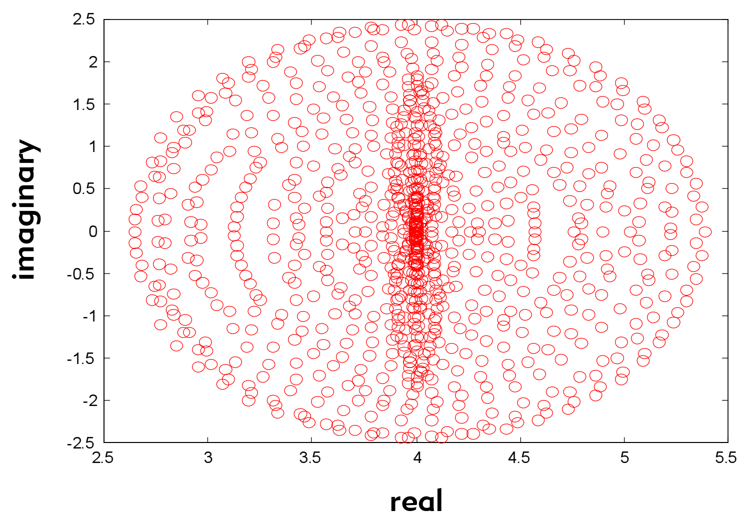
u_s=0; u_w=0; u_n=1; u_e=1;



Eigenvalues

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

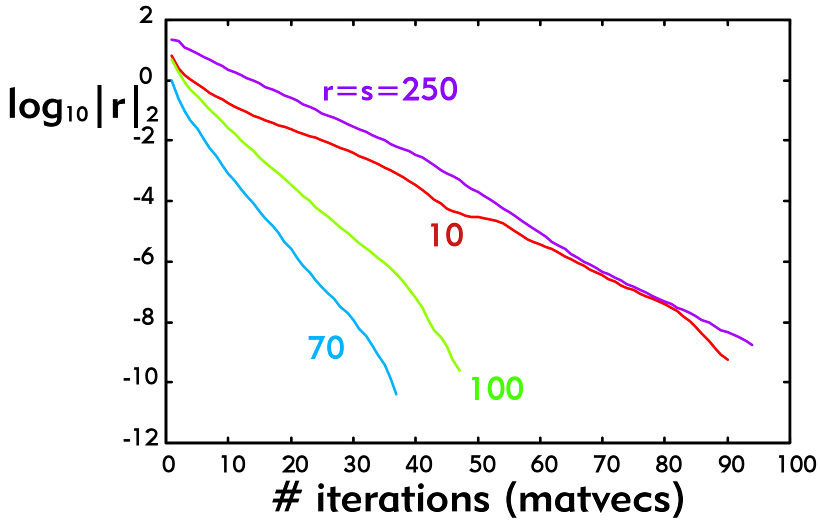
u_s=0; u_w=0; u_n=1; u_e=1;



GMRES for varying convection

p=q=1; r=s: given; h=1/31;

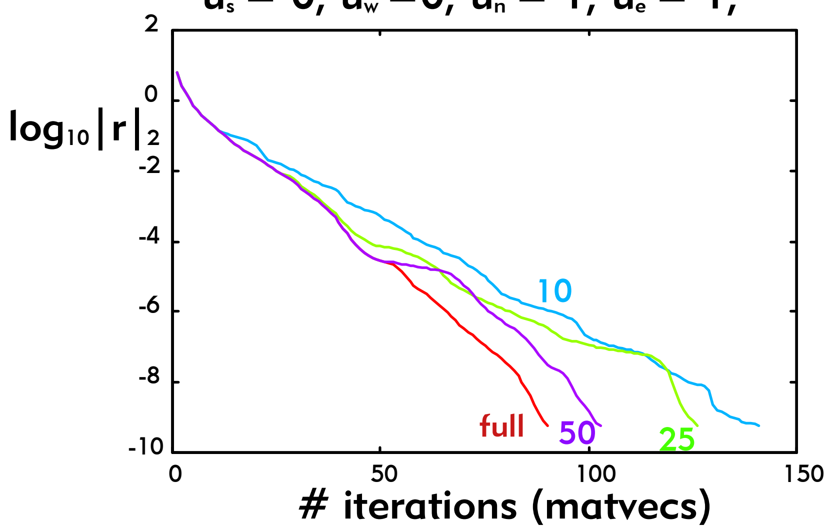
$u_s = 0$; $u_w = 0$; $u_n = 1$; $u_e = 1$;



GMRES(m) with r=s=10

p=q=1; r=s=10; h=1/31;

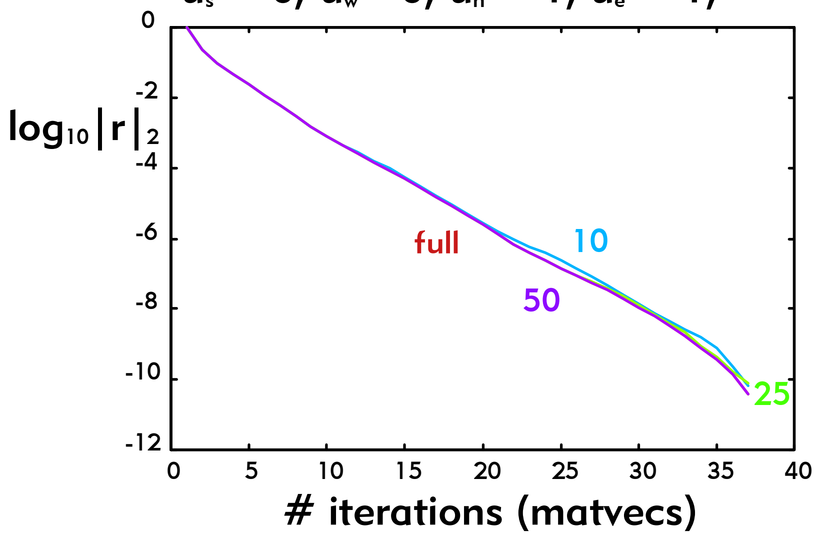
$u_s = 0$; $u_w = 0$; $u_n = 1$; $u_e = 1$;



GMRES(m) with r=s=70

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

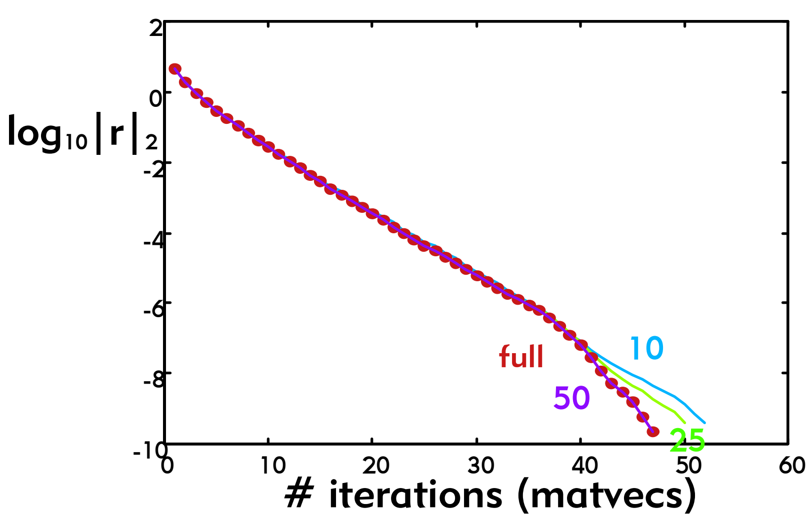
u_s = 0; u_w =0; u_n = 1; u_e = 1;



GMRES(m) with r=s=100

p=q=1; r=s=100; h=1/ 31;

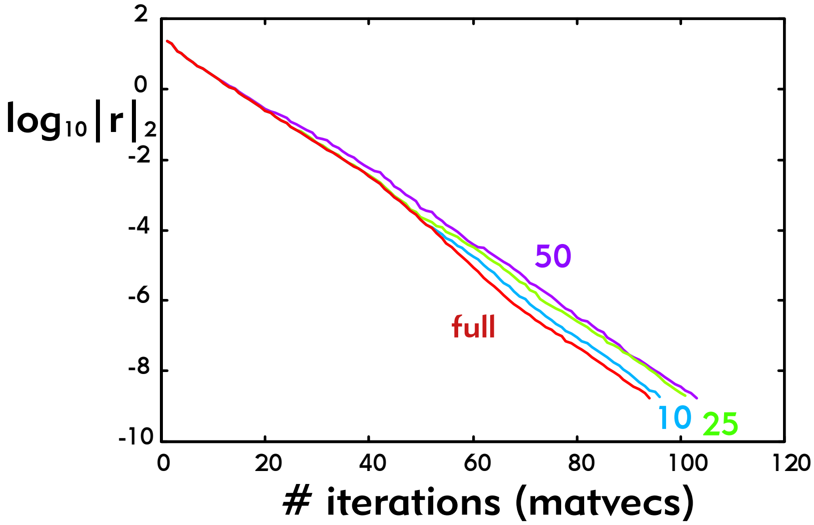
u_s = 0; u_w =0; u_n = 1; u_e = 1;



GMRES(m) with r=s=250

p=q=1; r=s=250; h=1/ 31;

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

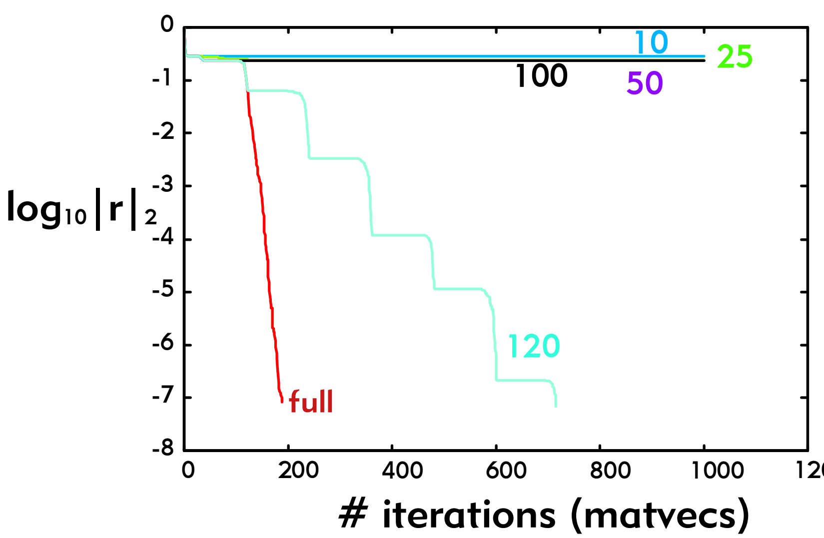


GMRES(m) after shifting spectrum

A=A-3.65*I

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

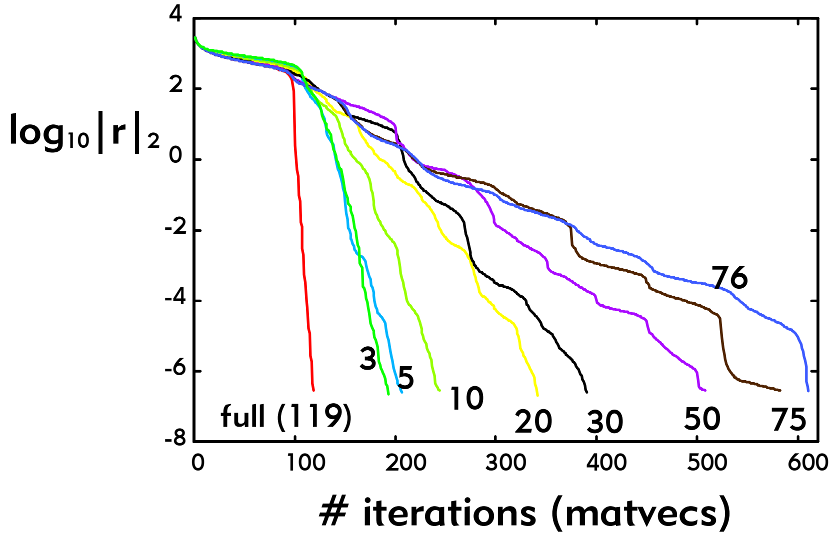
u_s = 0; u_w =0; u_n = 1; u_e = 1;



GMRES(m)

p=q=1; r=200; s=-200; t=0; f=0; h=1/51;

u_s = 0; u_w =100; u_n = 100; u_e = 0;



GMRES(m)

p=q=1; r=200; s=-200; t=0; f=0; h=1/51;

u_s = 0; u_w =100; u_n = 100; u_e = 0;