Krylov subspace methods

Introduction

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

Large Sparse Linear Systems

- Scientific and engineering simulations require the solution of (many) very large, sparse, linear systems.
- The matrices arise from finite element/volume discretization of partial differential or integral equations (and other areas) describing the physical behavior of complex systems.
- Accurate solution requires millions of unknowns.
- Time-dependent nonlinear problem: Solve a nonlinear system each timestep, which (Newton iteration) requires many linear systems to be solved.
- Very large optimization problems: each iteration requires the solution of a linear system.
- New fields of application: Financial modeling, Econometry, Biology.

Why iterative solvers?

Consider N x N matrix with k nonzeros/row (average), $k \ll N$:

- direct solver (LU): work: $O(N^3)$ storage: $O(N^2)$
- idem for band matrix: work: $O(b^2N)$ storage: $O(bN)$
  - 2D: $b = O(N^{1/2})$: work: $O(N^2)$ storage: $O(N^{3/2})$
  - 3D: $b = O(N^{2/3})$: work: $O(N^{7/3})$ storage: $O(N^{5/3})$
- sparse matrix $\times$ vector: work: $2Nk$ storage: Nk

For large problems direct methods are impossible; even for moderate problems they are much more expensive than iterative methods (if they converge).

Why iterative solvers?

Consider N x N matrix with k nonzeros/row (average), $k \ll N$:

Iterative methods; convergence in m iterations:

- typically $m \ll N$ (independent of 2D, 3D, ... problem),
- m depends on characteristics of problem rather than size,
- in general m increases only as a moderate function of N,
- for several problem classes constant m algorithms are known (multigrid O(N) work (optimal), multilevel O(1) iterations),
- Krylov subspace methods convergence in $m \leq N$ steps (in exact arithmetic).

Basic Iterative Methods (1)

System of nonlinear equations: $f(x) = 0$
Rewrite as $x = F(x)$, and iterate $x_{i+1} = F(x_i)$ (fixed-point iteration)
Converges if $\rho(\nabla F^T) < 1$ and ∇F^T Lip. cont. in neighborhood of solution

Linear system: $Ax = b$
Matrix splitting: $[P + (A-P)]x = b \Leftrightarrow Px = (P-A)x + b \Leftrightarrow$
$x = (I - P^{-1}A)x + P^{-1}b$
Iterate: $x_{i+1} = (I - P^{-1}A)x_i + P^{-1}b$
Converges if $\rho(I - P^{-1}A) < 1$

Methods: Jacobi iteration, Gauss-Seidel, (S)SOR, ...

Fixed-point: $x = (I - P^{-1}A)x + P^{-1}b \Leftrightarrow P^{-1}Ax = P^{-1}b$
Fixed-point is solution of the preconditioned system: $P^{-1}Ax = P^{-1}b$

Basic Iterative Methods (2)

$$x_{i+1} = (I - P^{-1}A)x_i + P^{-1}b = x_i + P^{-1}b - P^{-1}Ax_i$$

Linear System: $Ax = b$ Prec. system: $P^{-1}Ax = P^{-1}b$
Residual: $r_i = b - Ax_i$ Prec. residual: $\tilde{r}_i = P^{-1}b - P^{-1}Ax_i$

$$x_{i+1} = x_i + \tilde{r}_i \quad \Rightarrow \quad x_{i+1} = x_0 + \tilde{r}_0 + \tilde{r}_1 + \cdots + \tilde{r}_i$$

Update $x_{i+1} - x_0 = \tilde{r}_0 + \tilde{r}_1 + \cdots + \tilde{r}_i$

$$\tilde{r}_{i+1} = P^{-1}b - P^{-1}Ax_{i+1} = P^{-1}b - P^{-1}Ax_i - P^{-1}A\tilde{r}_i = \tilde{r}_i - P^{-1}A\tilde{r}_i$$
$$\tilde{r}_{i+1} = (I - P^{-1}A)\tilde{r}_i = (I - P^{-1}A)^{i+1}\tilde{r}_0$$

$$\tilde{r}_i \in \text{span}\{\tilde{r}_0, P^{-1}A\tilde{r}_0, \ldots, (P^{-1}A)^i\tilde{r}_0\} \equiv K^{i+1}(P^{-1}A, \tilde{r}_0)$$ Krylov subspace

$$x_i - x_0 \in \text{span}\{\tilde{r}_0, \tilde{r}_1, \ldots, \tilde{r}_{i-1}\} = K^i(P^{-1}A, \tilde{r}_0)$$

Basic Iterative Methods (3)

Solution to $Ax = b$: $\hat{x}$ Error: $e_i = \hat{x} - x_i$

Residual and error: $r_i = b - Ax_i = A\hat{x} - Ax_i = Ae_i$ $(\tilde{r}_i = P^{-1}Ae_i)$

Theorem: $\hat{x}$ is a fixed point of $x_{i+1} = (I - P^{-1}A)x_i + P^{-1}b$ iff
$\hat{x}$ is solution of $P^{-1}Ax = P^{-1}b$ $(\Leftrightarrow Ax = b)$

Proof: $x = (I - P^{-1}A)x + P^{-1}b = x - P^{-1}Ax + P^{-1}b \Leftrightarrow$
$P^{-1}Ax = P^{-1}b$

$$e_{i+1} = \hat{x} - x_{i+1} = (I - P^{-1}A)\hat{x} + P^{-1}b - (I - P^{-1}A)x_i - P^{-1}b$$
$$= (I - P^{-1}A)e_i$$

$e_{i+1} = (I - P^{-1}A)e_i = (I - P^{-1}A)^{i+1}e_0$ and $\tilde{r}_{i+1} = (I - P^{-1}A)^{i+1}\tilde{r}_0$

$$e_{i+1} \in \operatorname{span}\{e_0, P^{-1}Ae_0, (P^{-1}A)^2 e_0, \ldots, (P^{-1}A)^{i+1}e_0\}$$
$$e_{i+1} \in \operatorname{span}\{e_0, \tilde{r}_0, P^{-1}A\tilde{r}_0, \ldots, (P^{-1}A)^i \tilde{r}_0\}$$

Methods based on Projection (1)

Assume that in $Ax = b$, A is an explicitly preconditioned matrix

From original system $Ku = f$ we derive preconditioned system

$P^{-1}Ku = P^{-1}f$
or $KP^{-1}u = f$
or $P_1^{-1}KP_2^{-1}\tilde{u} = P_1^{-1}f$ and $P_2^{-1}\tilde{u} = u$

Iteration becomes
$x_{i+1} = (I - A)x_i + b = x_i + (b - Ax_i)$
$x_{i+1} = x_i + r_i$

Simple way to improve the iteration.
Is there a better update in same direction?

$x_{i+1} = x_i + \alpha_i(b - Ax_i)$ best α_i?

Methods based on Projection (2)

First question: Best in what sense?

a) minimum residual in 2-norm:

$x_{i+1} = x_i + \alpha_i(b - Ax_i) \quad \Rightarrow \quad r_{i+1} = r_i - \alpha_i A r_i$

minimum $\|r_{i+1}\|_2$: find point in $\text{span}\{Ar_i\}$ closest to r_i

Orthogonal projection of r_i on $\text{span}\{Ar_i\}$
Orthogonal in corresponding inner product: $\langle x, y\rangle_2 = y^H x$

$\alpha_i : Ar_i \perp r_i - Ar_i \quad \Leftrightarrow \quad \langle r_i - \alpha_i A r_i, Ar_i\rangle_2 = 0$

$\langle r_i, Ar_i\rangle_2 - \alpha_i\langle Ar_i, Ar_i\rangle_2 = 0 \quad \Leftrightarrow \quad \alpha_i = \frac{\langle r_i, Ar_i\rangle_2}{\langle Ar_i, Ar_i\rangle_2}$

$x_{i+1} = x_i + \frac{\langle r_i, Ar_i\rangle_2}{\langle Ar_i, Ar_i\rangle_2} r_i \quad \Rightarrow \quad r_{i+1} = r_i - \frac{\langle r_i, Ar_i\rangle_2}{\langle Ar_i, Ar_i\rangle_2} Ar_i$ Orthomin(1)

Methods Based on Projection (9)

First question: Best in what sense?
b) minimum error in A-norm if A is Hermitian positive definite:

$x_{i+1} = x_i + \alpha_i(b - Ax_i) \quad \Rightarrow \quad e_{i+1} = e_i - \alpha_i r_i$

minimum $\|e_{i+1}\|_A$: find point in $\text{span}\{r_i\}$ closest to e_i (in A-norm)

Orthogonal projection of e_i on $\text{span}\{r_i\}$
Orthogonal in corresponding inner product: $\langle x, y\rangle_A = y^H A x$

$\alpha_i : r_i \perp_A e_i - \alpha_i r_i \quad \Leftrightarrow \quad \langle e_i - \alpha_i r_i, r_i\rangle_A = 0$

$\langle r_i, Ae_i\rangle_2 - \alpha_i\langle r_i, Ar_i\rangle_2 = 0 \quad \Leftrightarrow \quad \alpha_i = \frac{\langle r_i, r_i\rangle_2}{\langle r_i, Ar_i\rangle_2}$

$x_{i+1} = x_i + \frac{\langle r_i, r_i\rangle_2}{\langle r_i, Ar_i\rangle_2} r_i \quad \Rightarrow \quad r_{i+1} = r_i - \frac{\langle r_i, r_i\rangle_2}{\langle r_i, Ar_i\rangle_2} Ar_i$ (steepest descent)

Note that we do not need to know error to minimize it in A-norm

Methods Based on Projection (10)

Steepest descent because of relation to quadratic problem:

$f(x) = \frac{1}{2}x^T Ax - b^T x + c$ for symmetric positive definite (SPD) A

$f(x + \varepsilon p) = f(x) + \varepsilon(Ax - b)^T p$, fastest decrease in direction of negative gradient: residual

Note that quadratic problem has same solution;
minimum if $f(x + \varepsilon p) = f(x)$ for any direction p
(note that stationary point must be minimum)

Hence, $\forall p : (Ax - b)^T p = 0$; his implies $Ax - b = 0$

Why does A SPD prove x is a minimum of $f(x)$ if $Ax - b = 0$?

Compare with classification of stationary point general problem.

Methods Based on Projection (12)

Note the following properties of Orthomin(1) and Steepest Descent:

Orthomin(1): $r_{i+1} = r_i - \alpha_i A r_i$ with $\alpha_i = \frac{\langle r_i, Ar_i \rangle}{\langle Ar_i, Ar_i \rangle}$

$$\langle r_{i+1}, Ar_i \rangle = \langle r_i, Ar_i \rangle - \alpha_i \langle Ar_i, Ar_i \rangle = 0$$
$$r_{i+1} \perp Ar_i$$

Steepest Descent: $r_{i+1} = r_i - \alpha_i A r_i$ with $\alpha_i = \frac{\langle r_i, r_i \rangle_2}{\langle r_i, Ar_i \rangle_2}$

$$\langle r_{i+1}, r_i \rangle = \langle r_i, r_i \rangle - \alpha_i \langle Ar_i, r_i \rangle = \langle r_i, r_i \rangle - \alpha_i \langle r_i, Ar_i \rangle = 0$$
$$r_{i+1} \perp r_i$$

What can we say about α_i in the steepest descent case?
(A is HPD)

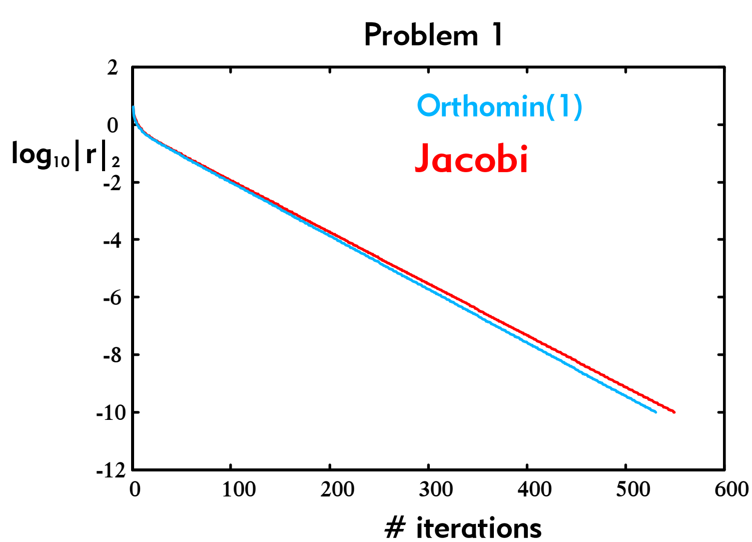
Orthomin(1) vs Jacobi iteration

We will use the Jacobi iteration, a basic iteration with P^{-1} the inverse of the diagonal of the matrix A, and Orthomin(1) on a simple PDE on the unit square, discretized on a 10×10 grid.

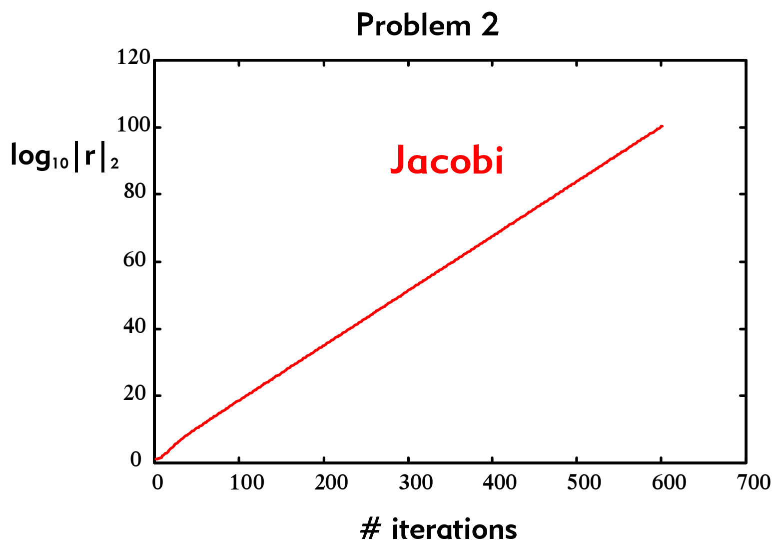
The PDE is $-u_{xx} - u_{yy} + ru_x - ru_y = 0$ with Direchlet boundary conditions $u = 0$ on the south and east boundary, and $u = 1$ on the west and north boundary.

In the first problem we take $r = 0$, in the second problem we take $r = 40$.

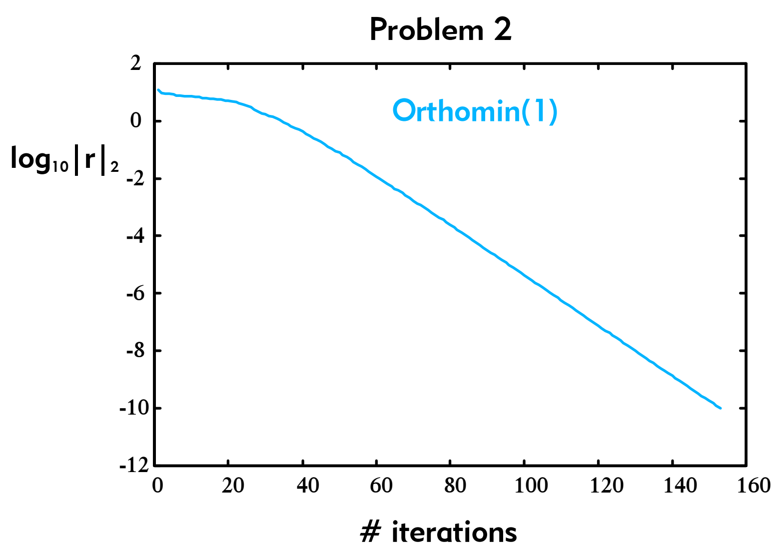
Orthomin(1) vs Jacobi iteration



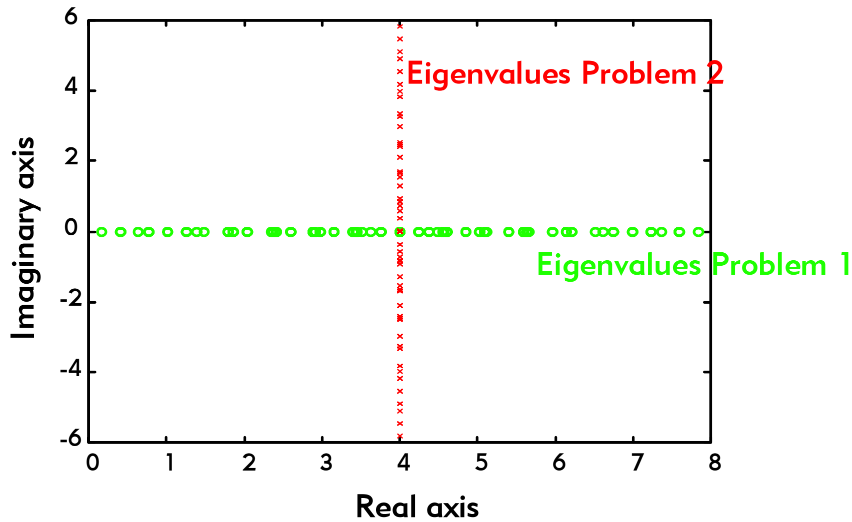
Orthomin(1) vs Jacobi iteration



Orthomin(1) vs Jacobi iteration



Eigenvalues of Test Problems



Optimal Projection Methods

Basic iterations generate (preconditioned) Krylov subspaces:

$x_i - x_0 \in K^i(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \ldots, A^{i-1}r_0\}$

$r_i \in K^{i+1}(A, r_0) = \text{span}\{r_0, Ar_0, A^2r_0, \ldots, A^i r_0\}$

Note that Orthomin(1) and steepest descent generate approximations and residuals from these same spaces.

1. $x_{i+1} = x_i + \frac{\langle r_i, Ar_i\rangle_2}{\langle Ar_i, Ar_i\rangle_2} r_i \quad \Rightarrow \quad r_{i+1} = r_i - \frac{\langle r_i, Ar_i\rangle_2}{\langle Ar_i, Ar_i\rangle_2} Ar_i$ Orthomin(1)
2. $x_{i+1} = x_i + \frac{\langle r_i, r_i\rangle_2}{\langle r_i, Ar_i\rangle_2} r_i \quad \Rightarrow \quad r_{i+1} = r_i - \frac{\langle r_i, r_i\rangle_2}{\langle r_i, Ar_i\rangle_2} Ar_i$ (steepest descent)

These two methods 'improve' convergence using 1-dimensional minimization. Hence, these methods have also been called accelerators.

The obvious question arises whether we can extend the idea and find the best approximation over a larger space; for example the entire subspace generated so far.

Optimal Projection Methods

Minimize the 2-norm of the residual:
Find $z_i \in K^i(A, r_0)$: $\|b - A(x_0 + z_i)\|_2$ is minimum, and set $x_i = x_0 + z_i$

Theorem: We obtain the minimum for z_i if $b - A(x_0 + z_i) \perp AK^i(A, r_0)$
Proof: Note that $\|b - A(x_0 + z_i)\|_2$ minimum is equivalent to $\|r_0 - Az_i)\|_2$ minimum. Let $\hat{z} \in K^i(A, r_0)$ such that $\|r_0 - A\hat{z}\|_2$ is minimum. Then $\hat{z}$ must be a stationary point of the function $f(z) = \|r_0 - Az\|_2^2$.
Hence for any unit vector $p \in K^i(A, r_0)$ we must have $f_p(\hat{z}) = 0$:

$$\lim_{\varepsilon \in \mathbb{R}, \varepsilon \to 0} \frac{f(\hat{z}+\varepsilon p) - f(\hat{z})}{\varepsilon} = 0 \Leftrightarrow$$

$$\lim_{\varepsilon \to 0} \frac{\|r_0 - A\hat{z} - \varepsilon p\|_2^2 - \|r_0 - A\hat{z}\|_2^2}{\varepsilon} = \lim_{\varepsilon \to 0} \frac{-\varepsilon p^H A^H (r_0 - A\hat{z}) - \varepsilon (r_0 - A\hat{z})^H Ap + \varepsilon^2 \|Ap\|_2^2}{\varepsilon} = 0 \Leftrightarrow$$

$p^H A^H (r_0 - A\hat{z}) + (r_0 - A\hat{z})^H Ap = 0$ for any unit $p \in K^i(A, r_0)$. This means $(r_0 - A\hat{z})^H Ap = 0$ for any unit $p \in K^i(A, r_0)$ (why?), and so, by definition, $(r_0 - A\hat{z}) \perp AK^i(A, r_0)$.

Optimal Projection Methods

So, to find optimal approximation ($\|r_i\|_2$) we try to find $z_i \in K^i(A, r_0)$ such that $r_0 - Az_i \perp AK^i(A, r_0)$: $z_i \in K^i(A, r_0) \Rightarrow z_i = \sum_{j=0}^{i-1} A^j r_0 \zeta_{j+1}$

So, $Az_i = \sum_{j=1}^{i} A^j r_0 \zeta_j = [Ar_0 \;\; A^2 r_0 \;\; A^3 r_0 \;\; \cdots \;\; A^i r_0]\zeta$ approximates r_0

We can rewrite problem in least squares form:
$[Ar_0 \;\; A^2 r_0 \;\; A^3 r_0 \;\; \cdots \;\; A^i r_0]\zeta \approx r_0 \quad \equiv \quad K\zeta \approx r_0$

This can be solved using
a) normal equations (accuracy problems)
b) QR decomposition

We have (min. 2-norm) unique decomposition: $r_0 = f_1 + f_2$
such that $K\zeta = f_1$ and $f_2 \perp \text{range}(K)$
Solve: $Q^{n \times i} R^{i \times i} = K$, where $Q^H Q = I$ and R upper triangular.
$f_2 = (I - QQ^H) r_0$, $f_1 = r_0 - f_2 = QQ^H r_0$, and $\zeta = R^{-1} Q^H f_1 = R^{-1} Q^H r_0$
$z_i = [r_0 \;\; A^1 r_0 \;\; A^2 r_0 \;\; \cdots \;\; A^{i-1} r_0]\zeta$

Optimal Projection Methods

Iteration-wise the problem is solved in three steps:

1) Extend the Krylov spaces $K^i(A, r_0)$ and $AK^i(A, r_0)$ by adding the respective next vectors $A^i r_0$ and $A^{i+1} r_0$ (only 1 matvec)

2) Compute orthogonal basis for $AK^i(A, r_0)$: QR-decomp. of K

3) Project r_0 (orthog) onto $AK^i(A, r_0)$ and solve the small problem $R\zeta = f_1 = Q^H r_0$. Note that this problem is only $i \times i$ irrespective of the actual size of the problem (say $n \times n$).

We would like to carry out these steps efficiently.
The GCR method (Generalized Conjugate Residuals) illustrates these steps well

GCR

GCR: $Ax = b$
Choose x_0 (e.g. $x_0 = 0$), and tol

$r_0 = b - Ax_0;\ i = 0;$
while $\|r_i\|_2 > tol$ do
    $i = i + 1;$ r_{i-1} adds search vector to $K^{i-1}(A, r_0)$
    $u_i = r_{i-1}; c_i = Au_i;$ Ar_{i-1} extends $AK^{i-1}(A, r_0)$
    for $j = 1, i-1$ do
        $u_i = u_i - u_j c_j^H c_i;$ Orthog. c_i against previous c_j and
        $c_i = c_i - c_j c_j^H c_i;$ update u_i such that $Au_i = c_i$ maintained
    end do
    $u_i = u_i / \|c_i\|_2;\ c_i = c_i / \|c_i\|_2;$ Normalize; end QR decomposition
    $x_i = x_{i-1} + u_i c_i^H r_{i-1};$ Project new c_i out of residual and
    $r_i = r_{i-1} - c_i c_i^H r_{i-1};$ update solution accordingly
end do   Note that $r_i \perp c_j$ for $j \le i$

What can go wrong with this algorithm?

GCR

Recapitulation of GCR; after m iterations:

$u_i \in K^i(A, r_0)$, $c_i \in K^i(A, Ar_0)$, for $i = 1 \ldots m$
$r_i \in K^{i+1}(A, r_0) = \text{span}\{r_0, r_1, \ldots, r_i\}$, for $i = 0 \ldots m$

Let $U_m = [u_1\; u_2 \cdots u_m]$; $\quad C_m = [c_1\; c_2 \cdots c_m]$; $\quad AU_m = C_m$; $\quad C_m^H C_m = I$
$\text{range}(U_m) = K^m(A, r_0)$

$\|r_m\|_2 = \min\{\|r_0 - Az\| : z \in \text{range}(U_m)\}$; minimum obtained for z_m

$r_0 - Az_m \perp C_m \Rightarrow C_m^*(r_0 - Az_m) = 0$; set $z_m = U_m \zeta$.
$C_m^H r_0 - C_m^H C_m \zeta = 0 \Rightarrow \zeta = C_m^H r_0$ and $z_m = U_m C_m^H r_0 = A^{-1} C_m C_m^H r_0$.

$r_m = r_0 - AU_m C_m^H r_0 = r_0 - C_m C_m^H r_0 = (I - C_m C_m^H) r_0$

Note that $r_0 = r_m + \sum_{j=1}^{m} c_j c_j^H r_0$ is a decomposition on orthog. basis.

GMRES

First the GMRES method generates an orthogonal basis for the Krylov space $K^{m+1}(A, r_0)$:

$v_1 = r_0 / \|r_0\|_2$;
for $k = 1 : m$,
  $\tilde{v}_{k+1} = Av_k$;
  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
  $h_{k+1,k} = \|\tilde{v}_{k+1}\|_2$;
  $v_{k+1} = \tilde{v}_{k+1} / h_{k+1,k}$;
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 $\text{range}(V_{m+1})$.

GMRES

So we have generated the Krylov subspace (step 1), and we have an orthogonal basis for it (step 2, more or less). However, we do not have an orthogonal basis for $K^m(A, Ar_0) = \mathrm{range}(C_m)$. (why not?)

Step 3 is the orthogonal projection of the residual on $K^m(A, Ar_0) = \mathrm{range}(C_m)$ and computing the update to the approximate solution from $K^m(A, r_0) = \mathrm{range}(U_m)$.

Obviously we don't want to orthogonalize $K^m(A, Ar_0)$ as well.

QR-decomposition $\underline{H}_m \equiv H_{m+1,m} = Q_{m+1}\underline{R}_m$ (m Givens rotations), where $\underline{R}_m$ is upper triangular and has last row entirely zero.

So we can drop last row of $\underline{R}_m$ and last column of Q_{m+1} giving:

$\underline{H}_m = Q_{m+1}\underline{R}_m = \underline{Q}_m R_m$. (dimensions?)

GMRES

Using this QR-decomposition we have a QR-decomp. of AV_m:

$AV_m = V_{m+1}\underline{H}_m = \left(V_{m+1}\underline{Q}_m\right)R_m$; $V_{m+1}\underline{Q}_m$ is unitary and R_m is uppertri.

So for the cost of m Givens rotations we get the orthogonal basis for $K^m(A, r_0)$ implicitly, since $\mathrm{range}(AV_m) = K^m(A, Ar_0)$.

New residual and approximate solution:

$$r_m = \left(I - (V_{m+1}\underline{Q}_m)(V_{m+1}\underline{Q}_m)^H\right)r_0 = r_0 - V_{m+1}\underline{Q}_m \underline{Q}_m^H V_{m+1}^H r_0 =$$

$$r_0 - V_{m+1}\underline{Q}_m R_m R_m^{-1}\underline{Q}_m^H e_1 \|r_0\|_2 \qquad \text{(note } v_1 = r_0/\|r_0\|_2\text{.)}$$

$$r_0 - V_{m+1}\underline{H}_m R_m^{-1}\underline{Q}_m^H e_1 \|r_0\|_2$$

and

$$x_m = x_0 + A^{-1}(r_m - r_0) = x_0 + V_m R_m^{-1}\underline{Q}_m^H e_1 \|r_0\|_2$$

GMRES

Comparing with GCR, we see that apart from scaling each column with a unit scalar:

$C_m = V_{m+1}\underline{Q}_m$ and $U_m = V_m R_m^{-1}$ (note the relation $AU_m = C_m$)

The solution to the least squares problem (ζ in GCR) is given by

$$\underline{Q}_m^H V_{m+1}^H r_0 = \underline{Q}_m^H e_1 \| r_0 \|_2$$

Note that $R_m^{-1}\underline{Q}_m^H$ is the left inverse of $\underline{H}_m$.
So, multiplying an equation $\underline{H}_m y \approx f$ from the left by $R_m^{-1}\underline{Q}_m^H$ will give the least squares solution: $y = R_m^{-1}\underline{Q}_m^H f$.

GMRES

Alternative derivation:

We have generated the recurrence $AV_m = V_{m+1}\underline{H}_m$

Now solve $\min\{\| r_0 - Az \|_2 : z \in K^m(A, r_0)\}$; write $z = V_m y$

Minimize $\| r_0 - AV_m y \|_2$ over all m-vectors y

Now substitute for $r_0 = V_{m+1} e_1 \| r_0 \|_2$ and $AV_m y = V_{m+1}\underline{H}_m y$.
So we minimize

$$\| r_0 - AV_m y \|_2 = \left\| V_{m+1} e_1 \| r_0 \|_2 - V_{m+1}\underline{H}_m y \right\|_2 = \left\| e_1 \| r_0 \|_2 - \underline{H}_m y \right\|_2$$

So we must solve an $m+1 \times m$ least squares problem
We will exploit the structure of $\underline{H}_m$ to
1. do this efficiently
2. compute the residual norm without the residual

GMRES

By construction $\underline{H}_m$ has the following structure

$$\underline{H}_m = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,m-1} & h_{1,m} \\ h_{2,1} & h_{2,2} & h_{2,3} & & h_{2,m-1} & h_{1,m} \\ & h_{3,2} & h_{3,3} & & \vdots & \vdots \\ & & h_{4,3} & \ddots & h_{m-1,m-1} & h_{m-1,m} \\ & & & \ddots & h_{m,m-1} & h_{m,m} \\ & & & & & h_{m+1,m} \end{bmatrix} \quad \text{(Upper Hessenberg)}$$

Cheapest QR decomp. is by Givens rotations to zero lower diagonal.

$$G_1^H \underline{H}_m = \begin{bmatrix} c_1 & \bar{s}_1 & \\ -s_1 & \bar{c}_1 & \\ & & I_{m-1} \end{bmatrix} = \begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ & h_{3,2} & \cdots & h_{3,m} \\ & & \ddots & \vdots \end{bmatrix}$$

GMRES

Next step we compute:

$$G_2^H G_1^H \underline{H}_m = \begin{bmatrix} 1 & & & \\ & c_2 & \bar{s}_2 & \\ & -s_2 & \bar{c}_2 & \\ & & & I \end{bmatrix} \begin{bmatrix} * & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ & h_{3,2} & h_{3,3} & \cdots & h_{3,m} \\ & & h_{4,3} & \cdots & h_{4,m} \\ & & & \ddots & \vdots \end{bmatrix} = \begin{bmatrix} * & * & * & \cdots & * \\ 0 & * & * & \cdots & * \\ & 0 & * & \cdots & * \\ & & h_{4,3} & \cdots & h_{m,3} \\ & & & \ddots & \vdots \end{bmatrix}$$

After m Givens rotations:

$$G_m^H \cdots G_1^H \underline{H}_m = Q_{m+1}^H \underline{H}_m = \begin{bmatrix} r_{1,1} & & \cdots & & r_{1,m} \\ 0 & r_{2,2} & & & \\ & 0 & r_{3,3} & & \vdots \\ \vdots & & 0 & \ddots & \\ & & & \ddots & r_{m,m} \\ 0 & & \cdots & & 0 \end{bmatrix} = \underline{R}_m$$

GMRES

So the least squares problem

$$y_m = \arg\min\left\{ \left\| e_1 \| r_0 \|_2 - \underline{H}_m y \right\|_2 : y \in \mathbb{C}^m \right\}$$

can be solved by multiplying $\underline{H}_m y \approx e_1 \| r_0 \|_2$ from left by $R_m^{-1} \underline{Q}_m^H$:

$$y_m = R_m^{-1} \underline{Q}_m^H e_1 \| r_0 \|_2$$

In practice:
We stepwise compute $G_i^H (G_{i-1}^H \cdots G_1^H \underline{H}_i)$ and $G_i^H (G_{i-1}^H \cdots G_1^H e_1 \| r_0 \|_2)$
This means updating $\underline{H}_{i-1}$ with new column, carry out previous Givens rotations on new column.
Compute new Givens rotation and update $\underline{H}_i$ and right hand side (of small least squares problem): $G_i^H (G_{i-1}^H \cdots G_1^H e_1 \| r_0 \|_2)$

GMRES

Least squares system looks like $\underline{R}_i y_i = Q_{i+1}^H e_1 \| r_0 \|_2$.
We may assume $\underline{R}_i$ has no zeros on diagonal (see later)

Since bottom row of $\underline{R}_i$ is zero we can only solve for $(Q_{i+1}^H e_1 \| r_0 \|_2)_{1\ldots i}$
(first i coeff.s)

This is exactly what we do in: get y_i by solving $R_i y_i = \underline{Q}_i^H e_1 \| r_0 \|_2$

Note from derivation that norm residual from LS problem is norm actual residual: $\| r_i \|_2 = | \tilde{q}_{i+1}^H e_1 | \| r_0 \|_2$ ($\tilde{q}_{i+1}$ since it changes with i):

$$\| r_0 - A V_m y \|_2 = \left\| V_{m+1} e_1 \| r_0 \|_2 - V_{m+1} \underline{H}_m y \right\|_2 = \left\| e_1 \| r_0 \|_2 - \underline{H}_m y \right\|_2$$

This way we can monitor convergence without actually computing updates to solution and residual (cheap).

©2001 Eric de 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: $\underline{H}_k = Q_{k+1}\underline{R}_k$

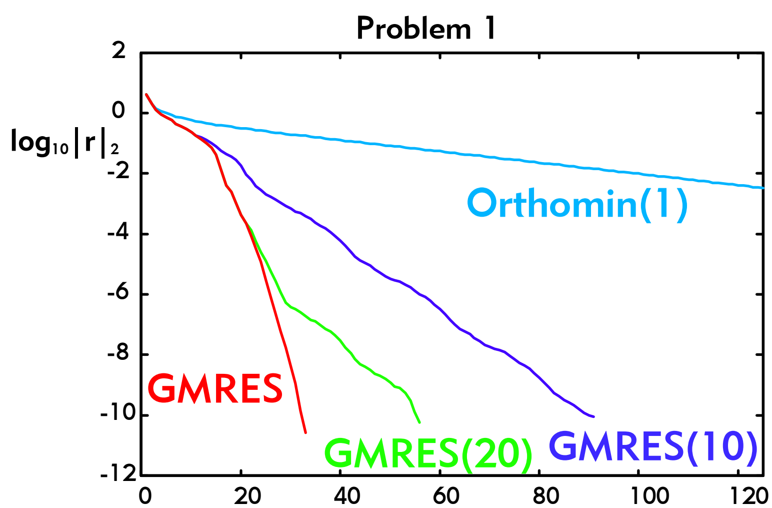
$\|r_k\|_2 = |\tilde{q}_{k+1}^H e_1| \, \|r_0\|_2$

end

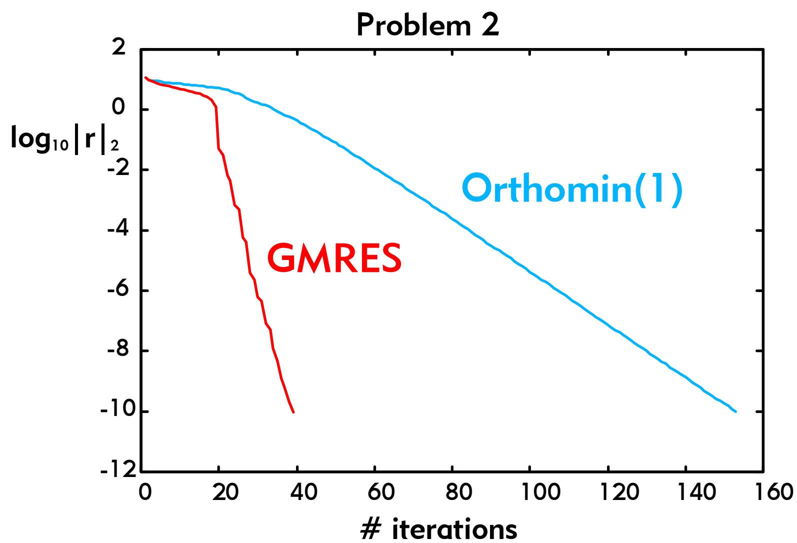
$y_k = R_k^{-1} \underline{Q}_k^H e_1 \|r_0\|_2;\ x_k = x_0 + V_k y_k;$

$r_k = r_0 - V_{k+1}\underline{H}_k y_k = V_{k+1}\left(I - \underline{Q}_k \underline{Q}_k^H\right) e_1 \|r_0\|_2;$ (or simply $r_k = b - Ax_k$)

GMRES

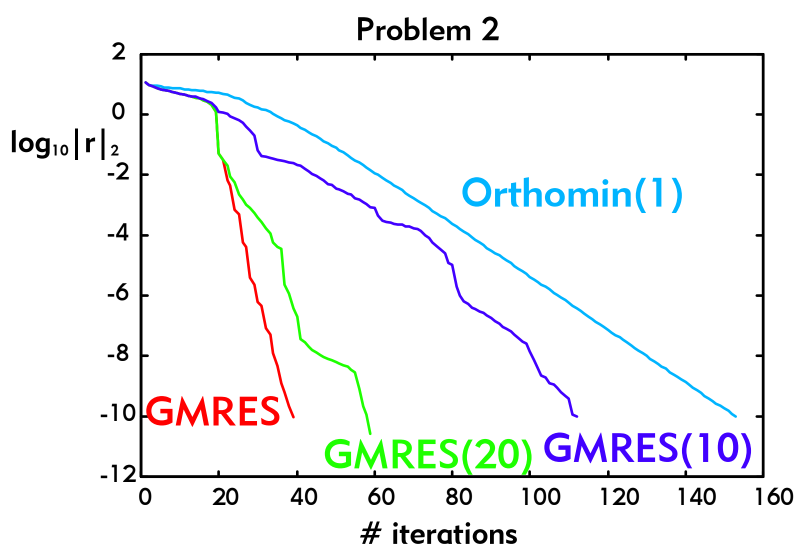


GMRES



©2001 Eric de Sturler

GMRES



©2001 Eric de Sturler

Givens rotations (1)

Complex Givens rotations:

$$G = \begin{pmatrix} c & \bar{s} \\ -s & \bar{c} \end{pmatrix},\ G^H G = \begin{pmatrix} \bar{c} & -\bar{s} \\ s & c \end{pmatrix}\begin{pmatrix} c & \bar{s} \\ -s & \bar{c} \end{pmatrix} = \begin{pmatrix} \bar{c}c + \bar{s}s & \bar{c}\bar{s} - \bar{c}\bar{s} \\ sc - sc & s\bar{s} + c\bar{c} \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

Verify $GG^H = I$. So, G is unitary.

Givens rotation so that $\begin{pmatrix} c & \bar{s} \\ -s & \bar{c} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \tilde{x} \\ 0 \end{pmatrix}$ where $|\tilde{x}| = \left\| \begin{pmatrix} x \\ y \end{pmatrix} \right\|_2$.

What degrees of freedom (assuming same purpose) in G?

How can we use those degrees of freedom?
What properties of G can we ensure?

Givens rotations (2)

Computing complex Givens rotations:

$$\begin{pmatrix} c & \bar{s} \\ -s & \bar{c} \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \tilde{x} \\ 0 \end{pmatrix}$$

Note that if we make G unitary, then making the second equation hold automatically makes the first hold. So with requirement of G unitary, only one of the equations is essential.

Second equation: $|c|^2 + |s|^2 = 1$.

$-sx + \bar{c}y = 0 \Leftrightarrow \bar{c} = s\frac{x}{y}$ (if $y \neq 0$) or $-sx + \bar{c}y = 0 \Leftrightarrow s = \bar{c}\frac{y}{x}$ (if $x \neq 0$).

For numerical accuracy it is not a good idea to divide a large number by a small one.

Givens rotations (3)

$y = 0 \to c = 1; s = 0;$

$|y| \geq |x| \to$
$\tilde{z} = x/y;\ z = |\tilde{z}|;$
$|c| = z|s| \to |c|^2 + |s|^2 = z^2|s|^2 + |s|^2 = 1 \Rightarrow |s| = (z^2+1)^{-1/2}$

Now we *choose* $c = z|s| \in \mathbb{R}$.
From $\bar{c} = c = s\frac{x}{y}$ we see that $\arg s = -\arg\frac{x}{y}$.
So we set $s = (z^2+1)^{-1/2}(z/\tilde{z})$

$|y| < |x| \to$
$\tilde{z} = y/x;\ z = |\tilde{z}|;$
$|s| = z|c| \to |c|^2 + |s|^2 = |c|^2 + z^2|c|^2 = 1 \Rightarrow |c| = (z^2+1)^{-1/2}$
Now we *choose* $c = (z^2+1)^{-1/2} \in \mathbb{R}$.
Following $s = \frac{y}{x}\bar{c}$ we set $s = \tilde{z}c$.