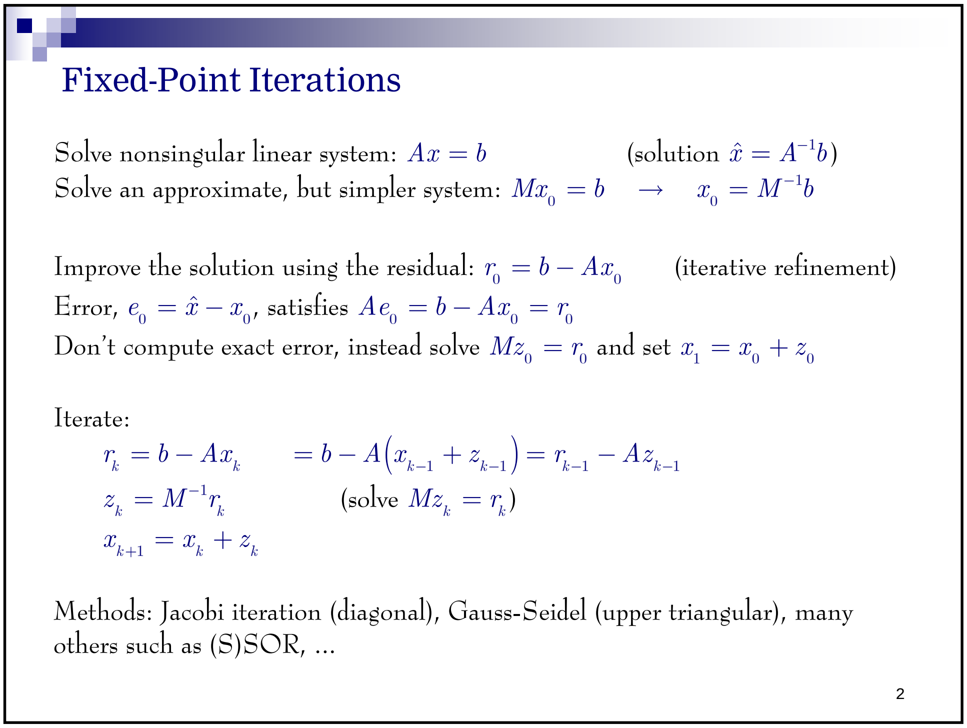
Fixed-Point Iterations,
Krylov Spaces, and Krylov Methods



Fixed-Point Iterations

Solve nonsingular linear system: $Ax = b$ (solution $\hat{x} = A^{-1}b$)
Solve an approximate, but simpler system: $Mx_0 = b \quad \rightarrow \quad x_0 = M^{-1}b$

Improve the solution using the residual: $r_0 = b - Ax_0$ (iterative refinement)
Error, $e_0 = \hat{x} - x_0$, satisfies $Ae_0 = b - Ax_0 = r_0$
Don't compute exact error, instead solve $Mz_0 = r_0$ and set $x_1 = x_0 + z_0$

Iterate:

$$r_k = b - Ax_k \qquad = b - A\left(x_{k-1} + z_{k-1}\right) = r_{k-1} - Az_{k-1}$$
$$z_k = M^{-1}r_k \qquad \text{(solve } Mz_k = r_k\text{)}$$
$$x_{k+1} = x_k + z_k$$

Methods: Jacobi iteration (diagonal), Gauss-Seidel (upper triangular), many others such as (S)SOR, …

Fixed-Point Iterations

Convergence of such iterations?
Let $A = M - N$ (matrix splitting).
Then $Mz_k = b - Ax_k \Leftrightarrow Mz_k = b - Mx_k + Nx_k \Leftrightarrow Mx_{k+1} = Nx_k + b$
(fixed-point iteration $x_{k+1} = M^{-1}Nx_k + M^{-1}b$)

Note that the fixed-point is the solution (proof?)

Error:

$$\begin{aligned} e_k &= \hat{x} - x_k = M^{-1}N\hat{x} + M^{-1}b - M^{-1}Nx_{k-1} - M^{-1}b \\ &= M^{-1}N\left(\hat{x} - x_{k-1}\right) = M^{-1}Ne_{k-1} \\ &= \left(M^{-1}N\right)^k e_0 \end{aligned}$$

Residual: $r_k = \left(NM^{-1}\right)^k r_0$ and $M^{-1}r_k = \left(M^{-1}N\right)^k \left(M^{-1}r_0\right)$ (proof?)
To analyze convergence we need to introduce/review a number of concepts

Rate of Convergence

Let $\hat{x}$ be the solution of $Ax = b$, and we have iterates $x_0, x_1, x_2, \dots$

$\{x_k\}$ converges (q-)linearly to $\hat{x}$ if there are $N \geq 0$ and $c \in [0,1)$ such that for $k \geq N$: $\| x_{k+1} - \hat{x} \| \leq c \| x_k - \hat{x} \|$,

$\{x_k\}$ converges (q-)superlinearly to $\hat{x}$ if there are $N \geq 0$ and a sequence $\{c_k\}$ that converges to 0 such that for $k \geq N$: $\| x_{k+1} - \hat{x} \| \leq c_k \| x_k - \hat{x} \|$

$\{x_k\}$ converges to $\hat{x}$ with (q-)order at least p if there are $p > 1$, $c \geq 0$, and $N \geq 0$ such that $k \geq N$: $\| x_{k+1} - \hat{x} \| \leq c \| x_k - \hat{x} \|^p$ (quadratic if $p = 2$, cubic if $p = 3$, and so on)

$\{x_k\}$ converges to $\hat{x}$ with j-step (q-)order at least p if there are a fixed integer $j \geq 1$, $p > 1$, $c \geq 0$, and $N \geq 0$, such that $k \geq N$: $\| x_{k+j} - \hat{x} \| \leq c \| x_k - \hat{x} \|^p$

Norms

A norm on a vector space V is any function $f : V \to \mathbb{R}$ such that

1. $f\left(x\right) \geq 0$ and $f\left(x\right) = 0 \Leftrightarrow x = 0$,
2. $f\left(\alpha x\right) = \left|\alpha\right| f\left(x\right)$,
3. $f\left(x + y\right) \leq f\left(x\right) + f\left(y\right)$,

where $x \in V$ and $\alpha \in \mathbb{R}$.

Important vector spaces in this course: $\mathbb{R}^n$, $\mathbb{C}^n$, and $\mathbb{R}^{m\times n}$, $\mathbb{C}^{m\times n}$ (matrices).
Note that the set of all m-by-n matrices (real or complex) is a vector space.

Many matrix norms possess the submultiplicative or consistency property:

$f\left(AB\right) \leq f\left(A\right) f\left(B\right)$ for all $A \in \mathbb{C}^{m\times k}$ and $B \in \mathbb{C}^{k\times n}$ (or real matrices).

Note that strictly speaking this is a property of a *family of norms*, because in general 'each' f is defined on a different vector space.

Norms

We can define a matrix norm using a vector norm (an induced matrix norm):

$$\| A \|_{\alpha} = \max_{x \neq 0} \frac{\| Ax \|_{\alpha}}{\| x \|_{\alpha}} = \max_{\|x\|_{\alpha}=1} \| Ax \|_{\alpha}$$

Induced norms are always consistent (satisfy consistency property).

Two norms $\| . \|_{\alpha}$ and $\| . \|_{\beta}$ are equivalent if there exist positive, real constants a and b such that

$$\forall x : a \| x \|_{\alpha} \leq \| x \|_{\beta} \leq b \| x \|_{\alpha}$$

The constants depend on the two norms but not on x.

All norms on a finite dimensional vector space are equivalent.

Norms

Some useful norms on $\mathbb{R}^n$, $\mathbb{C}^n$, $\mathbb{R}^{m\times n}$, $\mathbb{C}^{m\times n}$:

p-norms: $\left\|x\right\|_p = \left(\sum_{i=1}^{n}\left|x_i\right|^p\right)^{\frac{1}{p}}$, especially $p = 1, 2, \infty$, where $\left\|x\right\|_\infty = \max_i \left|x_i\right|$.

Induced matrix p-norms are:

$\left\|A\right\|_1 = \max_j \sum_{i=1}^{n}\left|a_{ij}\right|$ (max absolute column sum)

$\left\|A\right\|_2 = \sigma_{\max}\left(A\right)$ (max singular value – harder to compute than others)

$\left\|A\right\|_\infty = \max_i \sum_{j=1}^{n}\left|a_{ij}\right|$ (max absolute row sum)

Matrix Frobenius norm:

$\left\|A\right\|_F = \left(\sum_{i,j=1}^{n}\left|a_{ij}\right|^2\right)^{\frac{1}{2}}$ (similar to vector 2-norm for a matrix)

All these norms are consistent (satisfy the submultiplicative property)

Eigenvalues and Eigenvectors

Let $Ax = \lambda x$ and $\breve{y}A = \lambda\breve{y}$ (for same λ).

We call the column vector x a (right) eigenvector, the row vector $\breve{y}$ a left eigenvector, and λ an eigenvalue, the triple together is called an eigentriple, and $\left(\lambda, x\right)$ and $\left(\lambda, \breve{y}\right)$ a (right) eigenpair or left eigenpair.

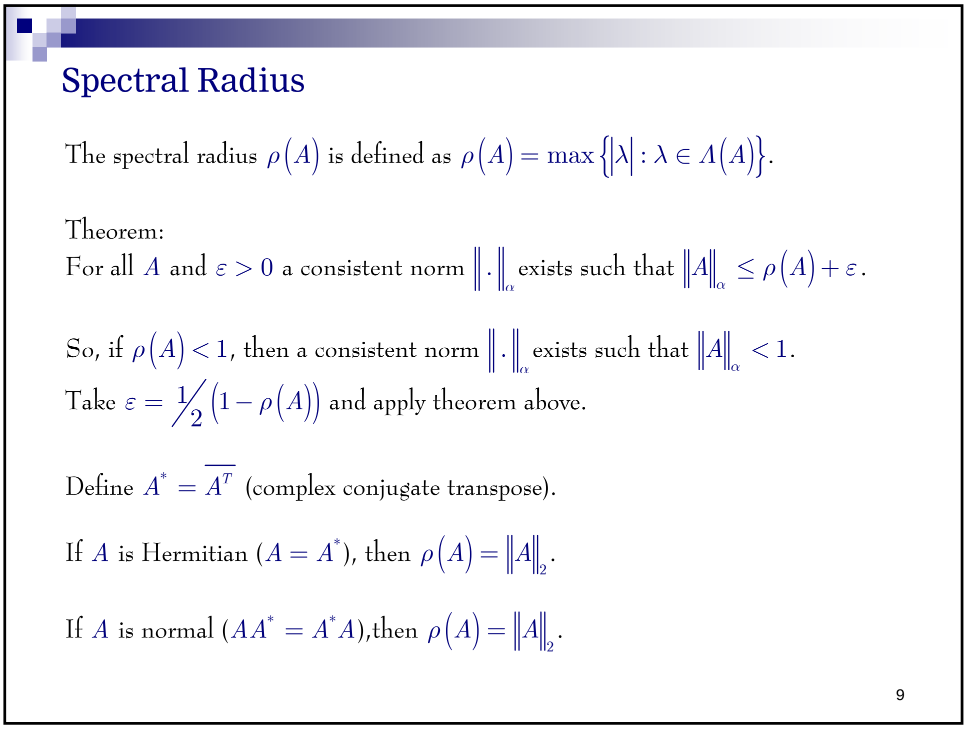
The set of all eigenvalues of A, $\Lambda\left(A\right)$, is called the spectrum of A.

If the matrix A is diagonalizable (has a complete set of eigenvectors) we have

$A = V\Lambda V^{-1} \Leftrightarrow AV = V\Lambda$,

where V is a matrix with the right eigenvectors as columns and Λ is a diagonal matrix with the eigenvalues as coefficients.

A similar decomposition can be given for the left eigenvectors.



Spectral Radius

The spectral radius $\rho(A)$ is defined as $\rho(A) = \max\{|\lambda| : \lambda \in \Lambda(A)\}$.

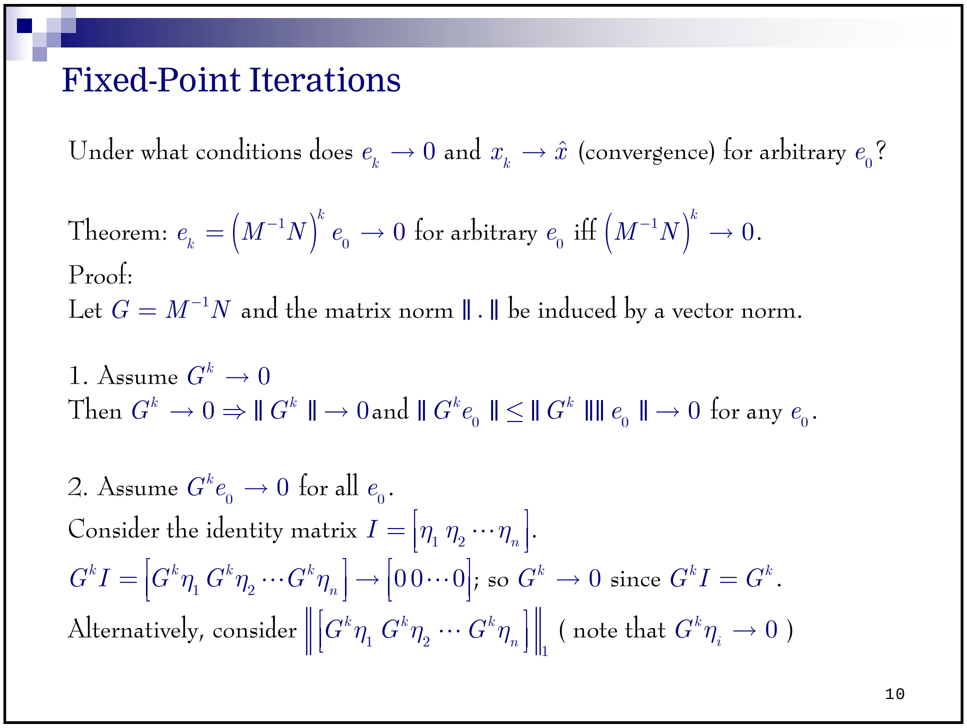
Theorem:
For all A and $\varepsilon > 0$ a consistent norm $\|.\|_\alpha$ exists such that $\|A\|_\alpha \le \rho(A) + \varepsilon$.

So, if $\rho(A) < 1$, then a consistent norm $\|.\|_\alpha$ exists such that $\|A\|_\alpha < 1$.
Take $\varepsilon = {}^1\!/_2 \left(1 - \rho(A)\right)$ and apply theorem above.

Define $A^* = \overline{A^T}$ (complex conjugate transpose).

If A is Hermitian ($A = A^*$), then $\rho(A) = \|A\|_2$.

If A is normal ($AA^* = A^*A$),then $\rho(A) = \|A\|_2$.



Fixed-Point Iterations

Under what conditions does $e_k \to 0$ and $x_k \to \hat{x}$ (convergence) for arbitrary e_0?

Theorem: $e_k = \left(M^{-1}N\right)^k e_0 \to 0$ for arbitrary e_0 iff $\left(M^{-1}N\right)^k \to 0$.
Proof:
Let $G = M^{-1}N$ and the matrix norm $\| . \|$ be induced by a vector norm.

1. Assume $G^k \to 0$
Then $G^k \to 0 \Rightarrow \| G^k \| \to 0$and $\| G^k e_0 \| \le \| G^k \| \| e_0 \| \to 0$ for any e_0.

2. Assume $G^k e_0 \to 0$ for all e_0.
Consider the identity matrix $I = \left[\eta_1\ \eta_2 \cdots \eta_n\right]$.
$G^k I = \left[G^k\eta_1\ G^k\eta_2 \cdots G^k\eta_n\right] \to \left[0\,0 \cdots 0\right]$; so $G^k \to 0$ since $G^k I = G^k$.
Alternatively, consider $\left\| \left[G^k\eta_1\ G^k\eta_2\ \cdots\ G^k\eta_n\right] \right\|_1$ (note that $G^k\eta_i \to 0$)

Norms

Note that we can generalize the result for the one-norm to all norms by using the equivalence of norms on finite dimensional vector spaces.

Similarly, the results are readily generalized for inconsistent matrix norms (with $\| AB \| > \| A \| \| B \|$possible), by using the equivalence of norms on finite dimensional spaces.

Fixed-Point Iterations

Theorem: $G^k \to 0$ iff $\rho\left(G\right) < 1$.

Proof:
1. $G^k \to 0 \Rightarrow \rho\left(G\right) < 1$.
For each eigenvalue λ of G there exists at least one eigenvector v s.t. $Gv = \lambda v$.
Then $\| G^k v \| = \| \lambda^k v \| = | \lambda |^k \| v \|$ and $G^k v \to 0$. So, $| \lambda |^k \| v \| \to 0 \Rightarrow | \lambda | < 1$.
Since this holds for each eigenvalue, $\rho\left(G\right) < 1$ must hold.

2. $\rho\left(G\right) < 1 \Rightarrow G^k \to 0$.
There exists a consistent norm $\| . \|_\alpha$ s.t. $\| G \|_\alpha < 1$.
Hence, $\| G^k \|_\alpha \leq \| G \|_\alpha^k \to 0$. Therefore $\| G^k \|_\alpha \to 0 \Rightarrow G^k \to 0$.

So $e_k \to 0$ ($x_k \to \hat{x}$) for arbitrary e_0 iff $\rho\left(M^{-1}N\right) < 1$.

Krylov Spaces

Given x_0, set $r_0 = b - Ax_0$.
For $k = 0,1,2,\ldots$

$$z_k = M^{-1} r_k,$$
$$x_{k+1} = x_k + z_k,$$
$$r_{k+1} = b - Ax_{k+1} = r_k - Az_k.$$

Note that $x_{k+1} - x_k = z_k = M^{-1} r_k$ and hence $x_{k+1} - x_0 = z_0 + z_1 + \cdots + z_k$.

This implies $x_{k+1} - x_0 = M^{-1} r_0 + \left(M^{-1}N\right) M^{-1} r_0 + \cdots + \left(M^{-1}N\right)^k M^{-1} r_0$.

So, correction $x_{k+1} - x_0$ is given by polynomial $S_k\left(t\right) = 1 + t + t^2 + \cdots + t^k$:

$$x_{k+1} - x_0 = \sum\nolimits_{i=0}^{k} \left(M^{-1}N\right)^i M^{-1} r_0 = S_k\left(M^{-1}N\right) \cdot M^{-1} r_0.$$

Note also $e_k = \left(M^{-1}N\right)^k e_0$ and $r_k = \left(NM^{-1}\right)^k r_0 = M\left(M^{-1}N\right)^k M^{-1} r_0$.

Similarity Transformation

Let A have eigenpairs $\left(\lambda_i, v_i\right)$: $Av_i = \lambda_i v_i$

Define the *similarity transformation*: BAB^{-1}

The matrix BAB^{-1} has the same eigenvalues, λ_i, as A and eigenvectors Bv_i:

$$BAB^{-1}\left(Bv_i\right) = BAv_i = \lambda_i Bv_i$$

Show that $\left(M^{-1}N\right)^k$ has the same eigenvalues as $\left(NM^{-1}\right)^k$.

Polynomials and Spaces

Main computational cost is in the multiplication by A and solving for M.
So, we can try to generate better polynomials (faster convergence) at same cost.

Also correction $x_{k+1} - x_0 \in \text{span}\left\{M^{-1}r_0, \left(M^{-1}N\right)M^{-1}r_0, \dots, \left(M^{-1}N\right)^k M^{-1}r_0\right\}$

We call a space $K_m\left(B, y\right) \equiv \text{span}\left\{y, By, B^2y, \dots, B^{m-1}y\right\}$

the Krylov (sub)space of dimension m associated with B and y.

So, $x_m - x_0 \in K_m\left(M^{-1}N, M^{-1}r_0\right)$

$$e_m = \left(M^{-1}N\right)^m e_0 \in K_{m+1}\left(M^{-1}N, e_0\right) \text{ and}$$

$$M^{-1}r_m = \left(M^{-1}N\right)^m M^{-1}r_0 \in K_{m+1}\left(M^{-1}N, M^{-1}r_0\right).$$

Therefore, alternatively we can compute better approximate solutions from the same space (faster convergence) at same cost.

Krylov Spaces

So, a Krylov space is a *space of polynomials in a matrix times a vector*.

These spaces inherit the many important approximation properties that polynomials on the real line or in the complex plane possess.

For simplicity let the matrix B be diagonalizable, $B = V\Lambda V^{-1}$.
Then $B^2 = V\Lambda V^{-1}V\Lambda V^{-1} = V\Lambda^2 V^{-1}$ and generally $B^k = V\Lambda^k V^{-1}$.
So, the polynomial $p_m\left(t\right) = \alpha_0 + \alpha_1 t + \dots + \alpha_m t^m$ applied to B gives

$$p_m\left(B\right) = V\left(\alpha_0 I + \alpha_1 \Lambda + \alpha_2 \Lambda^2 + \dots + \alpha_m \Lambda^m\right)V^{-1} \quad \text{and hence}$$

$$p_m\left(B\right) = Vp_m\left(\Lambda\right)V^{-1} = V\,\text{diag}\left(p_m\left(\lambda_1\right), \dots, p_m\left(\lambda_n\right)\right)V^{-1}$$

So, the polynomial is applied to the eigenvalues individually.
This allows us to approximate solutions to linear systems, eigenvalue problems, and more general problems using polynomial approximation.

Approximation by Matrix Polynomials

Let $B = V\Lambda V^{-1}$, let $\Lambda(B) \subset \Omega \subset \mathbb{C}$.

If $p_m(t) \approx \frac{1}{t}$ for all $t \in \Omega$, then $p_m(B) \approx B^{-1}$.

Let $y = V\zeta$. Then $p_m(B)y = \sum_i v_i p_m(\lambda_i)\zeta_i \approx \sum_i v_i \frac{\zeta_i}{\lambda_i} = B^{-1}y$

Furthermore, let $\varepsilon \approx 0$ and $|\lambda_i - \lambda_j| > \delta$ (for some eigenvalue λ_i)

If $p_m(t) = \begin{cases} \varepsilon, & t \in \Omega \text{ and } |t - \lambda_i| > \delta, \\ 1, & t = \lambda_i, \end{cases}$

then $p_m(B)y \approx v_i\zeta_i$.

If we can construct such polynomials for modest m we have an efficient linear solver or eigensolver.

Krylov Spaces

We have $M^{-1}N = M^{-1}(M - A) = I - M^{-1}A$.

So, $x_{k+1} - x_k = \left(I - M^{-1}A\right)^k M^{-1}r_0 \in K_{k+1}\left(M^{-1}A, M^{-1}r_0\right)$, and

$x_{k+1} = M^{-1}Nx_k + M^{-1}b$,

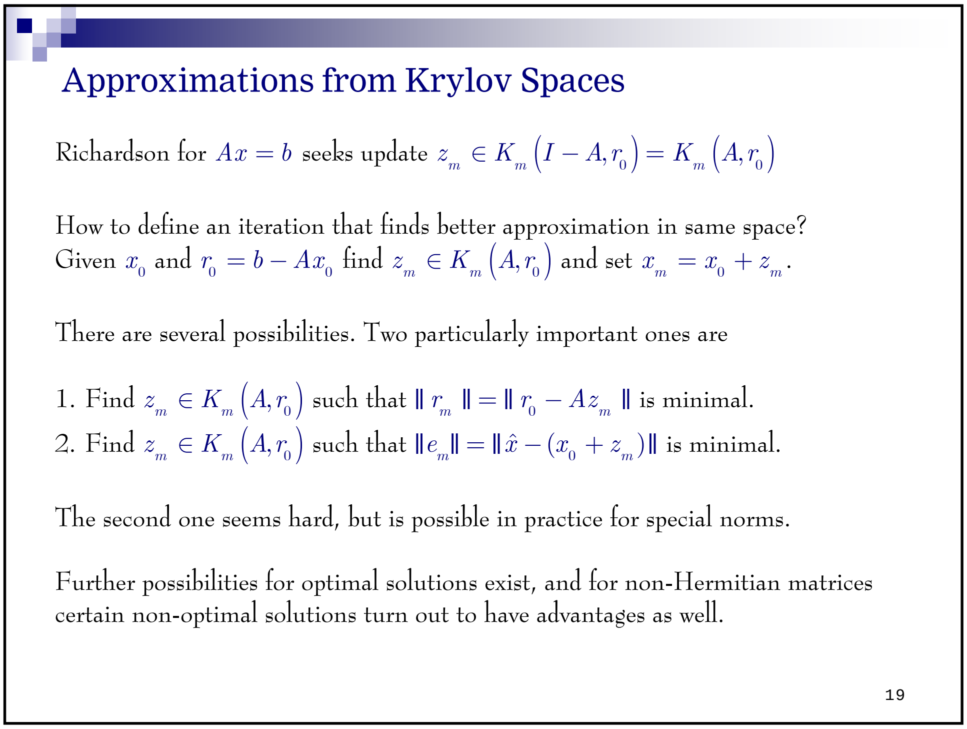
with fixed-point $\hat{x} = \left(I - M^{-1}A\right)\hat{x} + M^{-1}b \Leftrightarrow M^{-1}A\hat{x} = M^{-1}b$.

So, we solve the *preconditioned* problem $M^{-1}Ax = M^{-1}b$.
Preconditioning aims to improve the convergence of Richardson's iteration

$$x_{k+1} = (I - A)x_k + b$$

However, with $\tilde{A} = M^{-1}A$, $\tilde{M} = I$, and $\tilde{b} = M^{-1}b$ our iteration becomes Richardson's iteration, and we have Krylov spaces and polynomials based on $\tilde{A}$.

Hence, for simplicity we consider A and b as an explicitly preconditioned matrix and vector, and work with Krylov spaces in A and r_0 (most of the time).



Approximations from Krylov Spaces

Richardson for $Ax = b$ seeks update $z_m \in K_m\left(I - A, r_0\right) = K_m\left(A, r_0\right)$

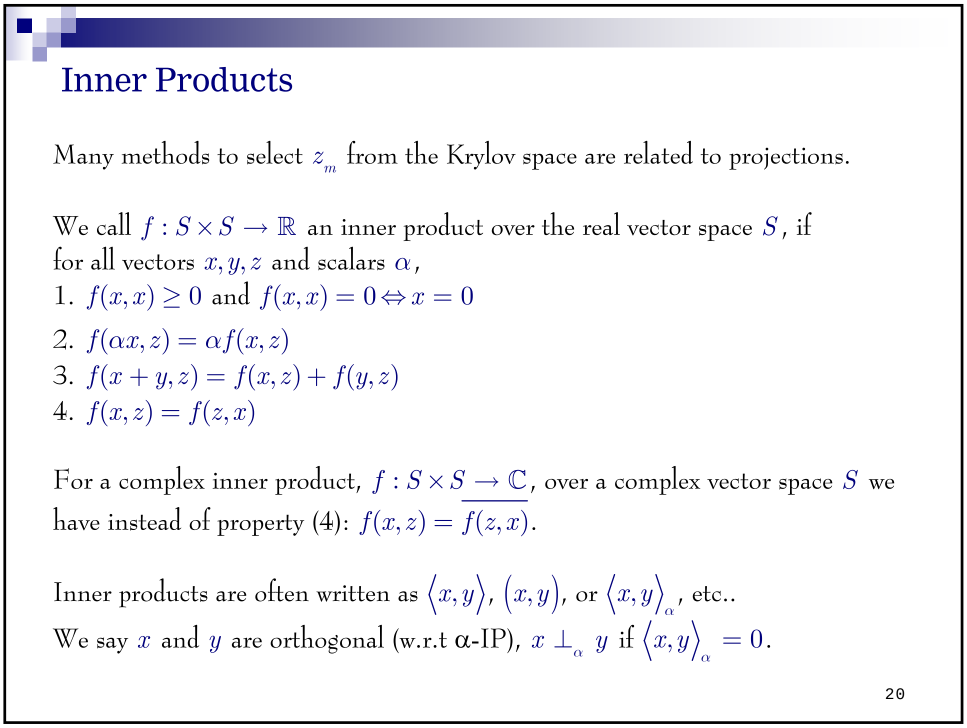
How to define an iteration that finds better approximation in same space?
Given x_0 and $r_0 = b - Ax_0$ find $z_m \in K_m\left(A, r_0\right)$ and set $x_m = x_0 + z_m$.

There are several possibilities. Two particularly important ones are

1. Find $z_m \in K_m\left(A, r_0\right)$ such that $\| r_m \| = \| r_0 - Az_m \|$ is minimal.
2. Find $z_m \in K_m\left(A, r_0\right)$ such that $\|e_m\| = \|\hat{x} - (x_0 + z_m)\|$ is minimal.

The second one seems hard, but is possible in practice for special norms.

Further possibilities for optimal solutions exist, and for non-Hermitian matrices certain non-optimal solutions turn out to have advantages as well.



Inner Products

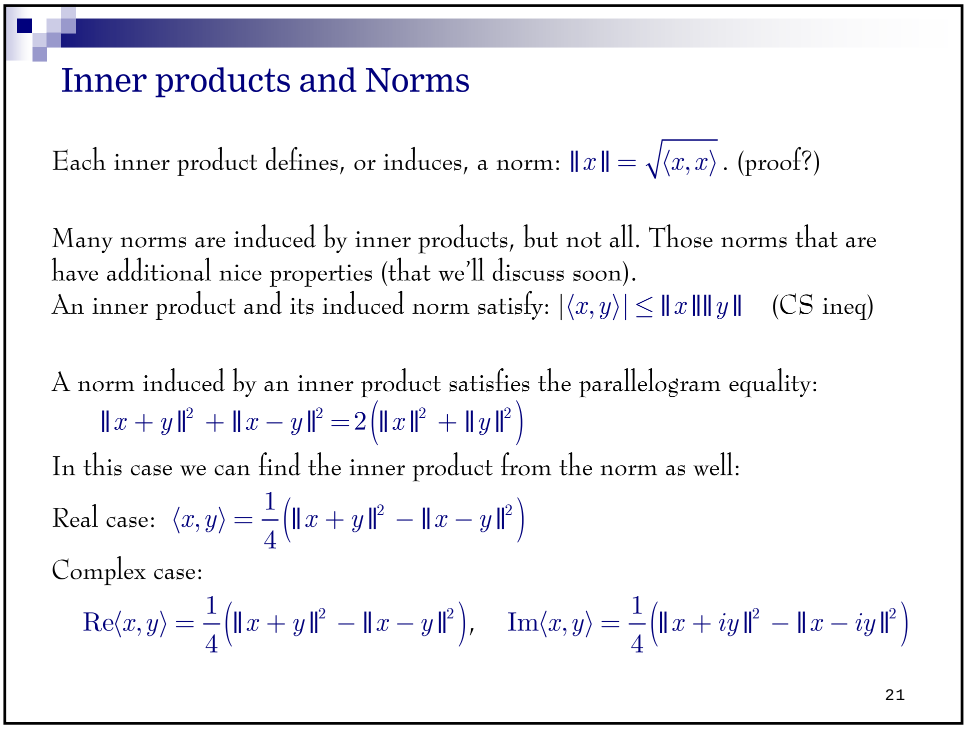
Many methods to select z_m from the Krylov space are related to projections.

We call $f : S \times S \to \mathbb{R}$ an inner product over the real vector space S, if
for all vectors x, y, z and scalars α,

1. $f(x,x) \geq 0$ and $f(x,x) = 0 \Leftrightarrow x = 0$
2. $f(\alpha x, z) = \alpha f(x,z)$
3. $f(x+y, z) = f(x,z) + f(y,z)$
4. $f(x,z) = f(z,x)$

For a complex inner product, $f : S \times S \to \mathbb{C}$, over a complex vector space S we have instead of property (4): $f(x,z) = \overline{f(z,x)}$.

Inner products are often written as $\left\langle x, y \right\rangle$, $\left(x, y\right)$, or $\left\langle x, y \right\rangle_\alpha$, etc..
We say x and y are orthogonal (w.r.t α-IP), $x \perp_\alpha y$ if $\left\langle x, y \right\rangle_\alpha = 0$.



Inner products and Norms

Each inner product defines, or induces, a norm: $\| x \| = \sqrt{\langle x, x \rangle}$. (proof?)

Many norms are induced by inner products, but not all. Those norms that are have additional nice properties (that we'll discuss soon).
An inner product and its induced norm satisfy: $|\langle x, y \rangle| \le \| x \| \| y \|$ (CS ineq)

A norm induced by an inner product satisfies the parallelogram equality:

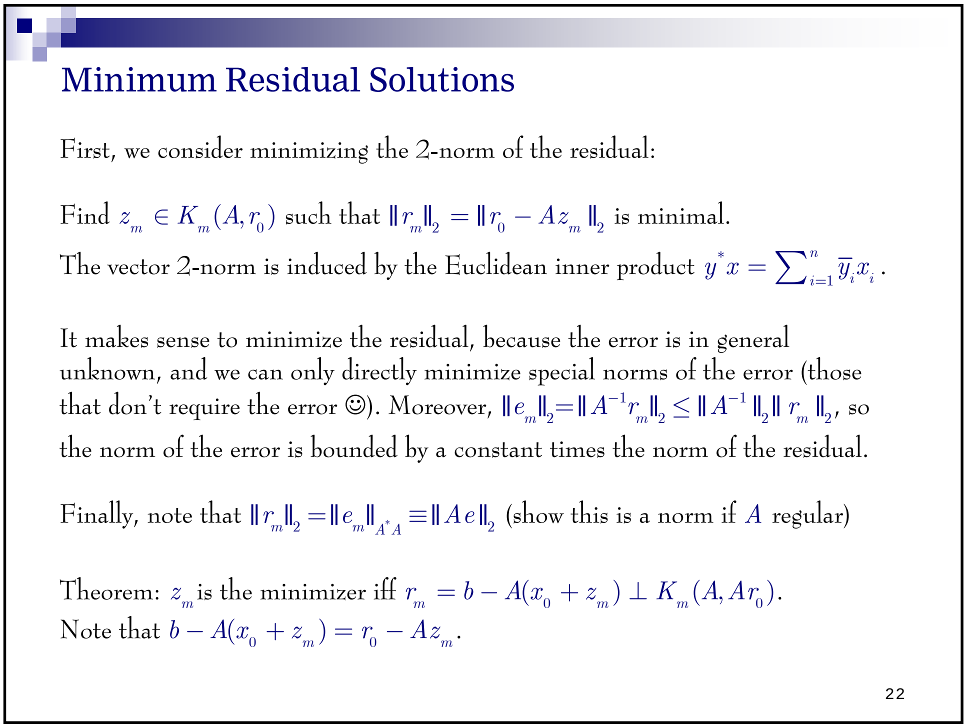
$$\| x + y \|^2 + \| x - y \|^2 = 2\left(\| x \|^2 + \| y \|^2 \right)$$

In this case we can find the inner product from the norm as well:

Real case: $\langle x, y \rangle = \frac{1}{4}\left(\| x + y \|^2 - \| x - y \|^2 \right)$

Complex case:

$$\mathrm{Re}\langle x, y \rangle = \frac{1}{4}\left(\| x + y \|^2 - \| x - y \|^2 \right), \quad \mathrm{Im}\langle x, y \rangle = \frac{1}{4}\left(\| x + iy \|^2 - \| x - iy \|^2 \right)$$



Minimum Residual Solutions

First, we consider minimizing the 2-norm of the residual:

Find $z_m \in K_m(A, r_0)$ such that $\| r_m \|_2 = \| r_0 - A z_m \|_2$ is minimal.

The vector 2-norm is induced by the Euclidean inner product $y^* x = \sum_{i=1}^{n} \overline{y}_i x_i$.

It makes sense to minimize the residual, because the error is in general unknown, and we can only directly minimize special norms of the error (those that don't require the error ☺). Moreover, $\| e_m \|_2 = \| A^{-1} r_m \|_2 \le \| A^{-1} \|_2 \| r_m \|_2$, so the norm of the error is bounded by a constant times the norm of the residual.

Finally, note that $\| r_m \|_2 = \| e_m \|_{A^* A} \equiv \| A e \|_2$ (show this is a norm if A regular)

Theorem: z_m is the minimizer iff $r_m = b - A(x_0 + z_m) \perp K_m(A, A r_0)$.
Note that $b - A(x_0 + z_m) = r_0 - A z_m$.

Minimum Residual Solutions

Proof: Let $f(z) = \| r_0 - Az \|_2^2$.

Then $\hat{z} \in K_m(A, r_0) : \| r_0 - A\hat{z} \|_2^2 \ \min$, implies $\hat{z}$ is a stationary point of $f(z)$:

For any unit vector $p \in K_m(A, r_0)$ we have $\dfrac{\partial f(\hat{z})}{\partial p} = 0$.

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

$$\lim_{\varepsilon \to 0} \frac{\| r_0 - A\hat{z} - \varepsilon A p \|_2^2 - \| r_0 - A\hat{z} \|_2^2}{\varepsilon} =$$

$$\lim_{\varepsilon \to 0} \frac{-\varepsilon p^* A^* (r_0 - A\hat{z}) - \varepsilon (r_0 - A\hat{z})^* A p + \varepsilon^2 p^* A^* A p}{\varepsilon} = 0 \quad \Leftrightarrow$$

$p^* A^* (r_0 - A\hat{z}) - (r_0 - A\hat{z})^* A p = 0$ for any unit vector $p \in K_m(A, r_0)$.

So, $(r_0 - A\hat{z})^* A p = 0 \Leftrightarrow (r_0 - A\hat{z}) \perp Ap$ for any unit $p \in K_m(A, r_0)$ (why?)

Since $Ap \in K_m(A, Ar_0)$ we have $(r_0 - A\hat{z}) \perp K_m(A, Ar_0) \equiv AK_m(A, r_0)$.

Minimum Residual Solutions

Minimizing a norm (cont. function) is in general complicated. However, the orthogonality conditions lead naturally to a linear system of equations.

Let $\left\{ w_1, w_2, \ldots, w_m \right\}$ form a basis for $K_m\left(A, r_0\right)$ and $\left\{ A w_i \right\}_{i=1}^{m}$ for $K_m\left(A, Ar_0\right)$.
Let $W_m = [w_1 \, w_2 \ldots w_m]$. Then $z_m = W_m \zeta$ (for some unknown ζ).

Now the orthogonality conditions $Aw_i \perp (r_0 - A W_m \zeta)$
yield the linear equations $\sum_{j=1}^{m} (w_i^* A^* A w_j) \zeta_j = w_i^* A^* r_0$.

In matrix form: $W_m^* A^* A W_m \zeta = W_m^* A^* r_0$ normal equations (accuracy problems)
LS problem: $\min_\zeta \| r_0 - K_m \zeta \|_2$ where $K_m = A W_m$
More accurate to solve LS problem using QR-decomposition.

Solving for ζ requires only the solution of an $m \times m$ system independent of n.

Minimum Residual Solutions

We have seen that $\min_\zeta \|r_0 - K_m\zeta\|_2 \Leftrightarrow K_m \perp (r_0 - K_m\zeta)$

Compute $K_m = Q_m R_m$ (QR-decomposition) where

$Q_m \in \mathbb{C}^{n\times m}$ $\left(\mathbb{R}^{n\times m}\right)$ s.t. $Q_m^* Q_m = I_m$ and $R_m \in \mathbb{C}^{n\times m}$ uppertriangular

If $\mathrm{rank}(K_m) = m$, then R_m is nonsingular and $\mathrm{range}(Q_m) = \mathrm{range}(K_m)$

Now $Q_m \perp (r_0 - K_m\zeta)$ gives $Q_m^* r_0 - Q_m^* Q_m R_m\zeta = 0 \Leftrightarrow R_m\zeta = Q_m^* r_0$ (easy solve)

Note that $K_m\zeta = Q_m Q_m^* r_0 \perp r_0 - K_m\zeta = r_0 - Q_m Q_m^* r_0$

$K_m\zeta$ is the orthogonal projection of r_0 onto $\mathrm{range}\left(K_m\right) = K_m(A, Ar_0)$.

$r_m = r_0 - K_m\zeta$ is the orthogonal projection of r_0 onto $\left(K_m(A, Ar_0)\right)^\perp$.

$Q_m Q_m^*$ and $I - Q_m Q_m^*$ are orthogonal projectors.

P is projector if $P^2 = P$, orthogonal projector if $R(\mathrm{P}) \perp N(\mathrm{P}) \Leftrightarrow \mathrm{P}^* = P$.

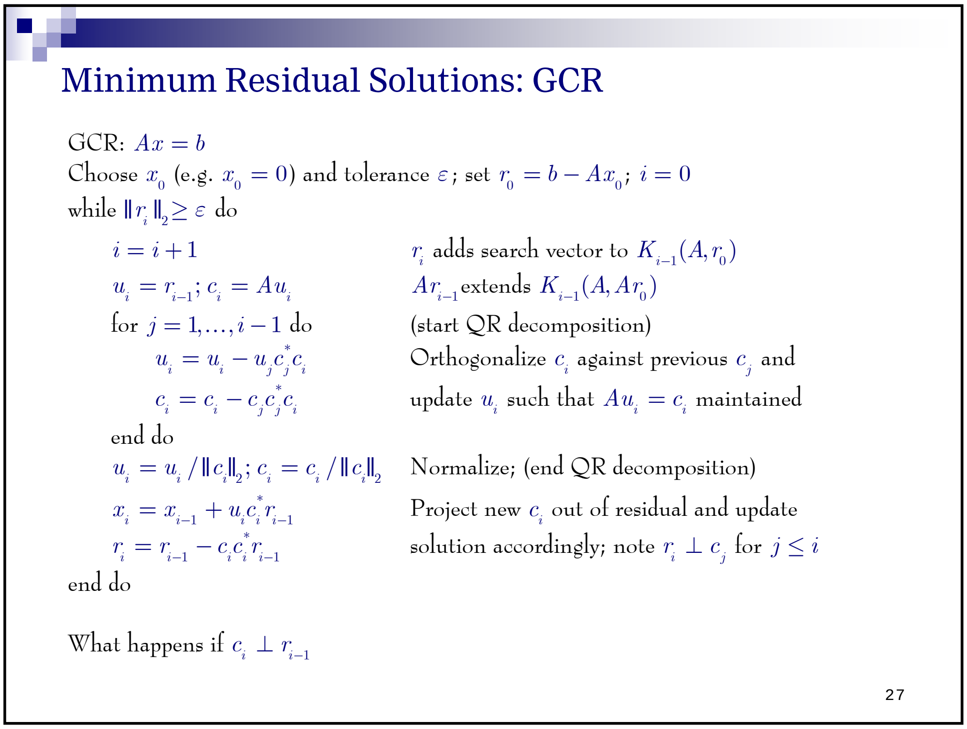
Minimum Residual Solutions

Iteration-wise the problem is solved in four steps:

1. Extend the Krylov spaces $K_m(A, r_0)$ and $K_m(A, Ar_0)$ by adding the respective next vectors $A^m r_0$ and $A^{m+1} r_0$ (only 1 matvec)
2. Update orthogonal basis for $K_m(A, Ar_0)$: QR-decomp. of K_m
3. Update projected matrix and projection of r_0 (orthog) onto $K_m(A, Ar_0)$
4. Solve the projected problem, e.g. $R\zeta = Q_m^* r_0$. Note that this problem is only $m \times m$ or $(m+1) \times m$ irrespective of the size of the linear system.

These steps vary somewhat for different methods.
We would like to carry out these steps efficiently.

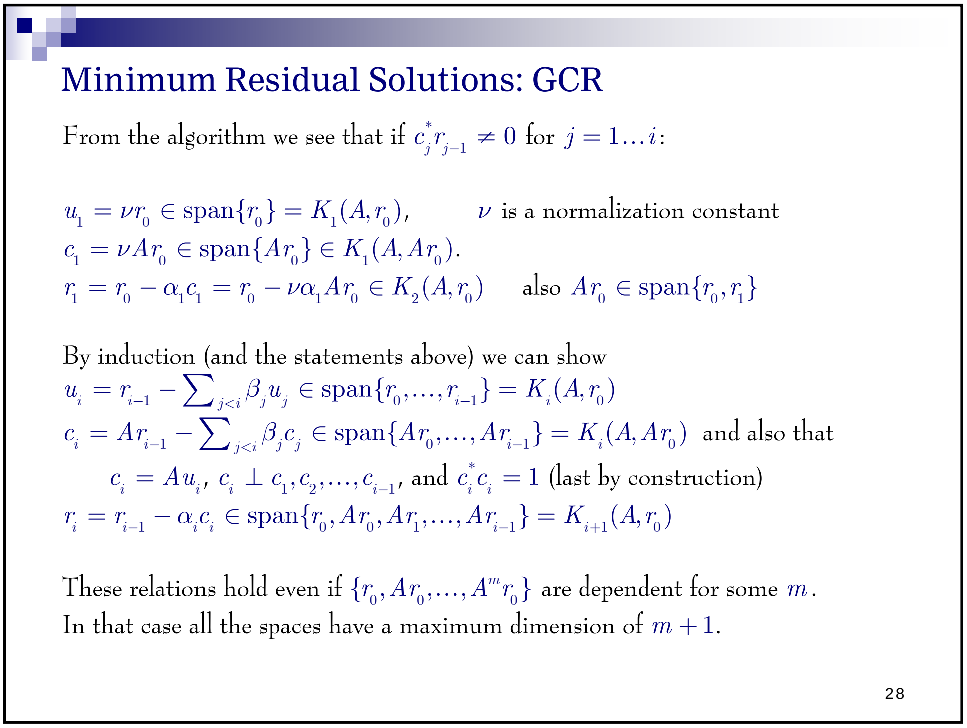
The GCR method (Generalized Conjugate Residuals) illustrates these steps well.
(Eisenstat, Elman, and Schulz 1983)



Minimum Residual Solutions: GCR

```
GCR: $Ax = b$
Choose $x_0$ (e.g. $x_0 = 0$) and tolerance $\varepsilon$; set $r_0 = b - Ax_0$; $i = 0$
while $\| r_i \|_2 \geq \varepsilon$ do
    $i = i + 1$                                  $r_i$ adds search vector to $K_{i-1}(A, r_0)$
    $u_i = r_{i-1}$; $c_i = Au_i$                $Ar_{i-1}$ extends $K_{i-1}(A, Ar_0)$
    for $j = 1, \ldots, i-1$ do                  (start QR decomposition)
        $u_i = u_i - u_j c_j^* c_i$              Orthogonalize $c_i$ against previous $c_j$ and
        $c_i = c_i - c_j c_j^* c_i$              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)
    $x_i = x_{i-1} + u_i c_i^* r_{i-1}$          Project new $c_i$ out of residual and update
    $r_i = r_{i-1} - c_i c_i^* r_{i-1}$          solution accordingly; note $r_i \perp c_j$ for $j \leq i$
end do
```

What happens if $c_i \perp r_{i-1}$



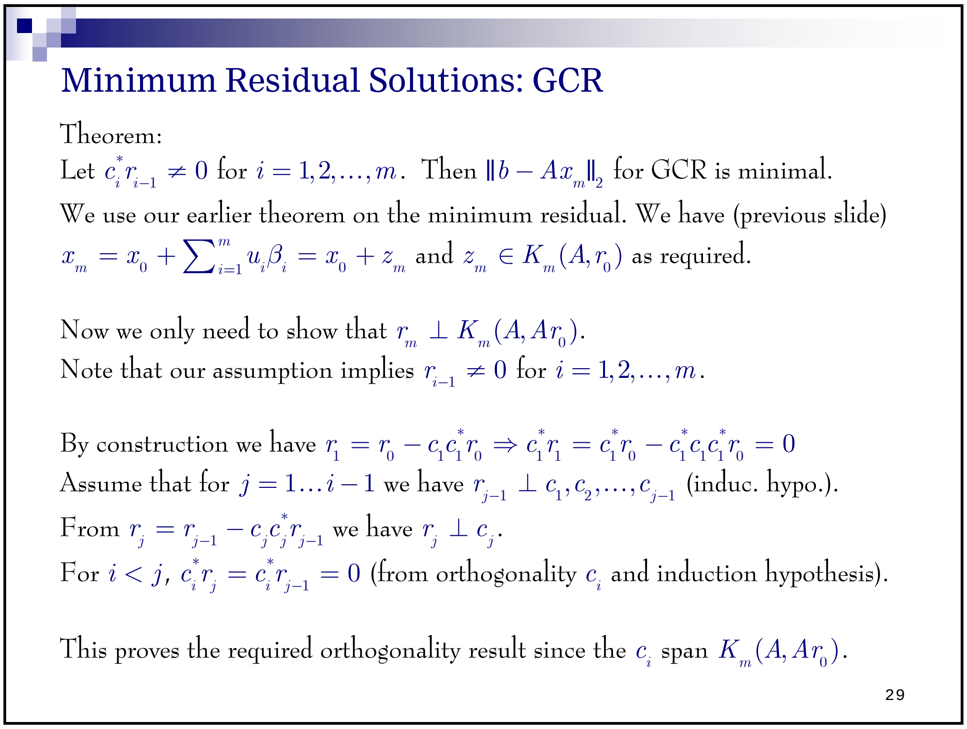
Minimum Residual Solutions: GCR

From the algorithm we see that if $c_j^* r_{j-1} \neq 0$ for $j = 1 \ldots i$:

$u_1 = \nu r_0 \in \text{span}\{r_0\} = K_1(A, r_0)$, ν is a normalization constant
$c_1 = \nu A r_0 \in \text{span}\{Ar_0\} \in K_1(A, Ar_0)$.
$r_1 = r_0 - \alpha_1 c_1 = r_0 - \nu \alpha_1 A r_0 \in K_2(A, r_0)$ also $Ar_0 \in \text{span}\{r_0, r_1\}$

By induction (and the statements above) we can show
$u_i = r_{i-1} - \sum_{j<i} \beta_j u_j \in \text{span}\{r_0, \ldots, r_{i-1}\} = K_i(A, r_0)$
$c_i = Ar_{i-1} - \sum_{j<i} \beta_j c_j \in \text{span}\{Ar_0, \ldots, Ar_{i-1}\} = K_i(A, Ar_0)$ and also that
$c_i = Au_i$, $c_i \perp c_1, c_2, \ldots, c_{i-1}$, and $c_i^* c_i = 1$ (last by construction)
$r_i = r_{i-1} - \alpha_i c_i \in \text{span}\{r_0, Ar_0, Ar_1, \ldots, Ar_{i-1}\} = K_{i+1}(A, r_0)$

These relations hold even if $\{r_0, Ar_0, \ldots, A^m r_0\}$ are dependent for some m.
In that case all the spaces have a maximum dimension of $m + 1$.



Minimum Residual Solutions: GCR

Theorem:
Let $c_i^* r_{i-1} \neq 0$ for $i = 1,2,\ldots,m$. Then $\|b - Ax_m\|_2$ for GCR is minimal.

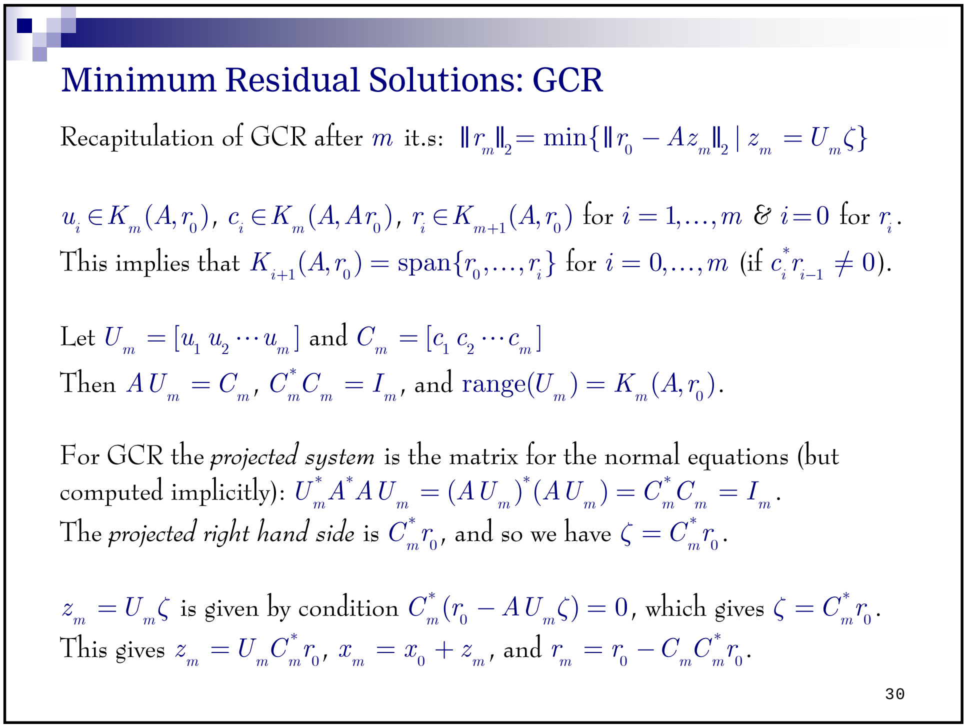
We use our earlier theorem on the minimum residual. We have (previous slide) $x_m = x_0 + \sum_{i=1}^{m} u_i \beta_i = x_0 + z_m$ and $z_m \in K_m(A, r_0)$ as required.

Now we only need to show that $r_m \perp K_m(A, Ar_0)$.
Note that our assumption implies $r_{i-1} \neq 0$ for $i = 1,2,\ldots,m$.

By construction we have $r_1 = r_0 - c_1 c_1^* r_0 \Rightarrow c_1^* r_1 = c_1^* r_0 - c_1^* c_1 c_1^* r_0 = 0$
Assume that for $j = 1 \ldots i-1$ we have $r_{j-1} \perp c_1, c_2, \ldots, c_{j-1}$ (induc. hypo.).
From $r_j = r_{j-1} - c_j c_j^* r_{j-1}$ we have $r_j \perp c_j$.
For $i < j$, $c_i^* r_j = c_i^* r_{j-1} = 0$ (from orthogonality c_i and induction hypothesis).

This proves the required orthogonality result since the c_i span $K_m(A, Ar_0)$.

29



Minimum Residual Solutions: GCR

Recapitulation of GCR after m it.s: $\|r_m\|_2 = \min\{\|r_0 - Az_m\|_2 \mid z_m = U_m \zeta\}$

$u_i \in K_m(A, r_0)$, $c_i \in K_m(A, Ar_0)$, $r_i \in K_{m+1}(A, r_0)$ for $i = 1,\ldots,m$ & $i = 0$ for r_i.
This implies that $K_{i+1}(A, r_0) = \text{span}\{r_0, \ldots, r_i\}$ for $i = 0,\ldots,m$ (if $c_i^* r_{i-1} \neq 0$).

Let $U_m = [u_1\ u_2 \cdots u_m]$ and $C_m = [c_1\ c_2 \cdots c_m]$
Then $AU_m = C_m$, $C_m^* C_m = I_m$, and $\text{range}(U_m) = K_m(A, r_0)$.

For GCR the *projected system* is the matrix for the normal equations (but computed implicitly): $U_m^* A^* A U_m = (AU_m)^* (AU_m) = C_m^* C_m = I_m$.
The *projected right hand side* is $C_m^* r_0$, and so we have $\zeta = C_m^* r_0$.

$z_m = U_m \zeta$ is given by condition $C_m^* (r_0 - AU_m \zeta) = 0$, which gives $\zeta = C_m^* r_0$.
This gives $z_m = U_m C_m^* r_0$, $x_m = x_0 + z_m$, and $r_m = r_0 - C_m C_m^* r_0$.

30

Minimum Residual Solutions: GCR

Note that $C_m C_m^*$ and $I - C_m C_m^*$ are both orthogonal projectors.

$C_m C_m^*$ is a projection since $C_m C_m^* C_m C_m^* = C_m C_m^*$.
$C_m C_m^*$ is an orthogonal projection since $C_m C_m^*$ is Hermitian.

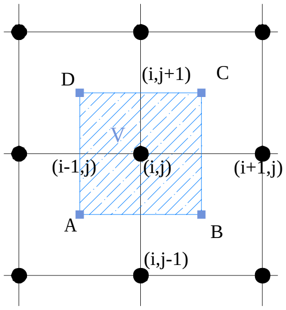
$I - C_m C_m^*$ is a projection since

$$(I - C_m C_m^*)(I - C_m C_m^*) = I - C_m C_m^* - C_m C_m^* + C_m C_m^* = I - C_m C_m^*.$$

$I - C_m C_m^*$ is an orthogonal projection since $I - C_m C_m^*$ is Hermitian.

Model Problems

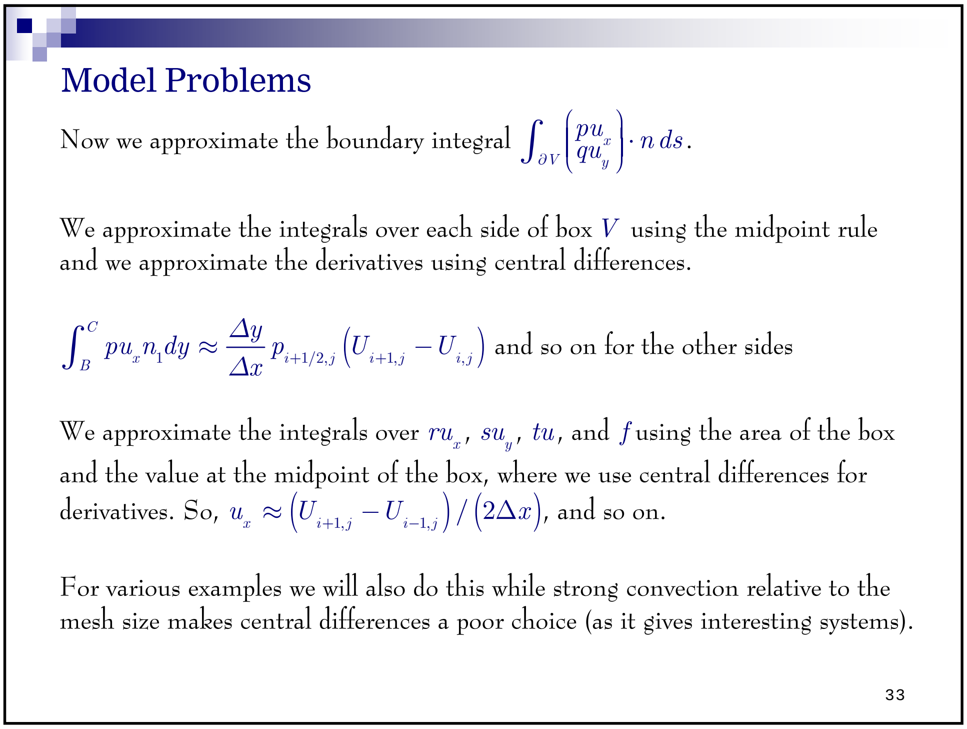
Discretize $-\left(pu_x\right)_x - \left(qu_y\right)_y + ru_x + su_y + tu = f$.



Integrate equality over box V. Use Gauss' divergence theorem to get

$$\int_V \left(pu_x\right)_x + \left(qu_y\right)_y \, dxdy = \int_{\partial V} \begin{pmatrix} pu_x \\ qu_y \end{pmatrix} \cdot n \, ds$$

And approximate the line integral numerically.



Model Problems

Now we approximate the boundary integral $\int_{\partial V} \begin{pmatrix} pu_x \\ qu_y \end{pmatrix} \cdot n\, ds$.

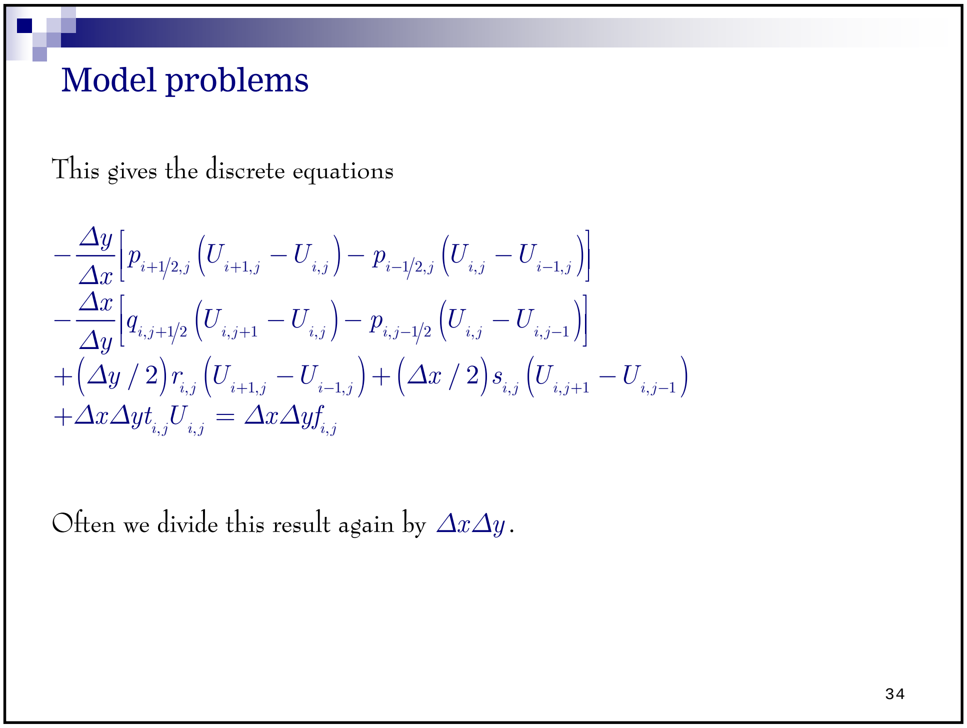
We approximate the integrals over each side of box V using the midpoint rule and we approximate the derivatives using central differences.

$\int_B^C pu_x n_1 dy \approx \frac{\Delta y}{\Delta x} p_{i+1/2,j} \left(U_{i+1,j} - U_{i,j}\right)$ and so on for the other sides

We approximate the integrals over ru_x, su_y, tu, and f using the area of the box and the value at the midpoint of the box, where we use central differences for derivatives. So, $u_x \approx \left(U_{i+1,j} - U_{i-1,j}\right) / \left(2\Delta x\right)$, and so on.

For various examples we will also do this while strong convection relative to the mesh size makes central differences a poor choice (as it gives interesting systems).

33



Model problems

This gives the discrete equations

$$
\begin{aligned}
&-\frac{\Delta y}{\Delta x}\left[p_{i+1/2,j}\left(U_{i+1,j} - U_{i,j}\right) - p_{i-1/2,j}\left(U_{i,j} - U_{i-1,j}\right)\right] \\
&-\frac{\Delta x}{\Delta y}\left[q_{i,j+1/2}\left(U_{i,j+1} - U_{i,j}\right) - p_{i,j-1/2}\left(U_{i,j} - U_{i,j-1}\right)\right] \\
&+\left(\Delta y / 2\right) r_{i,j}\left(U_{i+1,j} - U_{i-1,j}\right) + \left(\Delta x / 2\right) s_{i,j}\left(U_{i,j+1} - U_{i,j-1}\right) \\
&+\Delta x \Delta y t_{i,j} U_{i,j} = \Delta x \Delta y f_{i,j}
\end{aligned}
$$

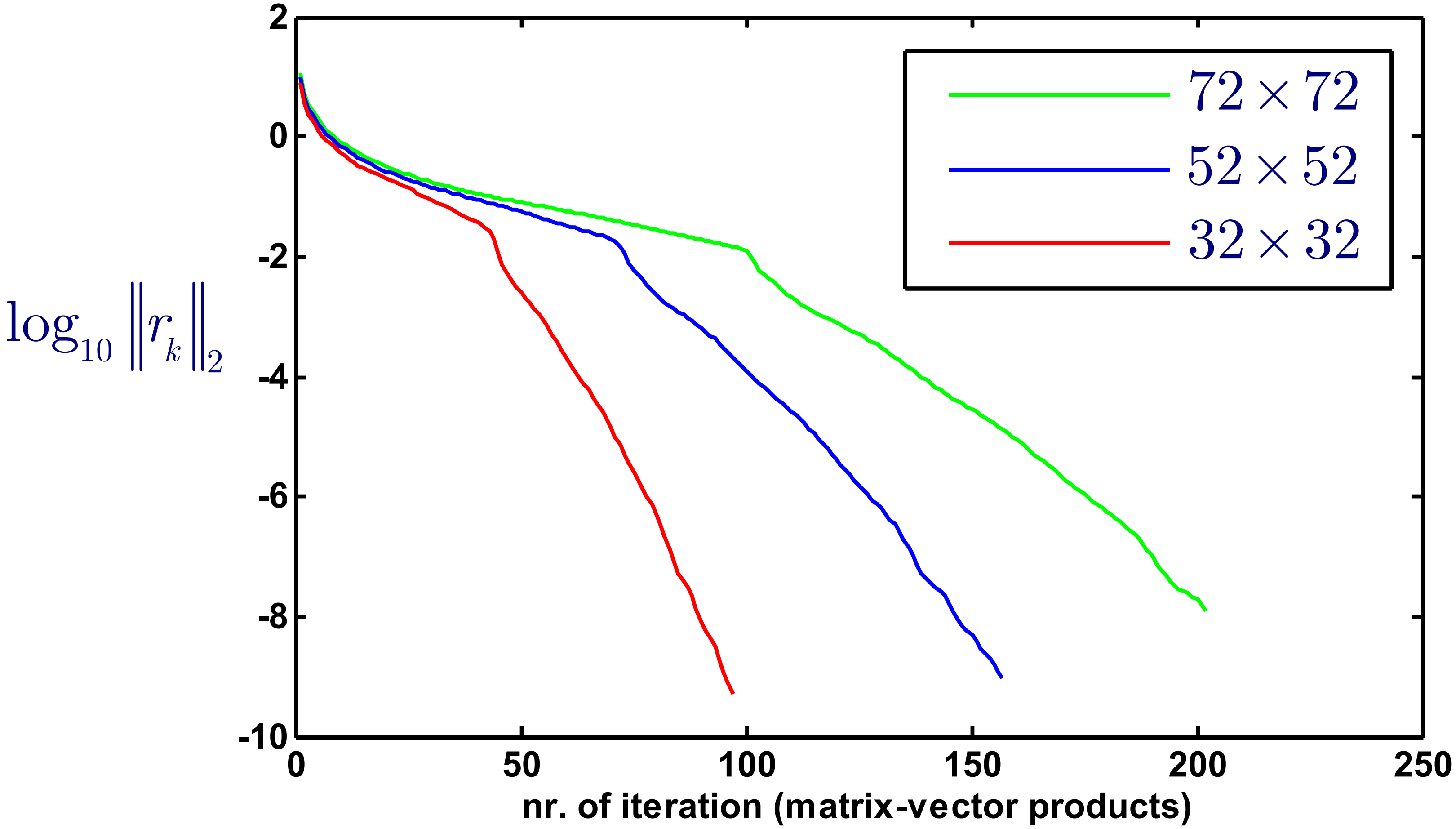
Often we divide this result again by $\Delta x \Delta y$.

34

Experiments with GCR

Solve $\Delta u = 0$ on unit square with Dirichlet boundary conditions.

Note the modest deterioration of convergence rate as discretization becomes finer (so cost slightly worse than linear)

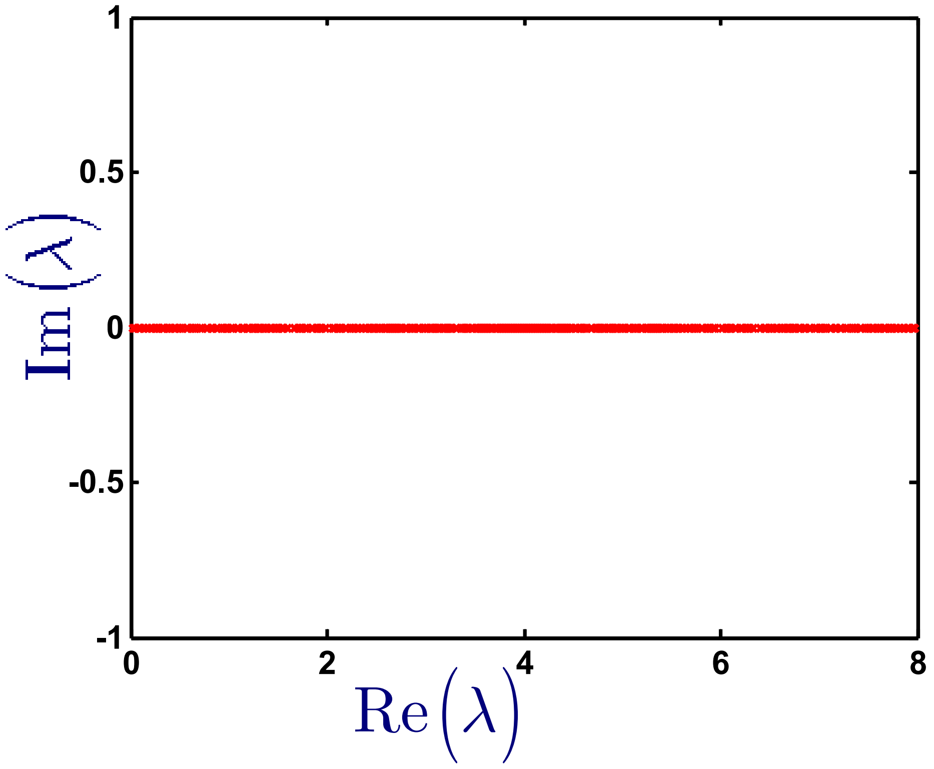


Experiments with GCR

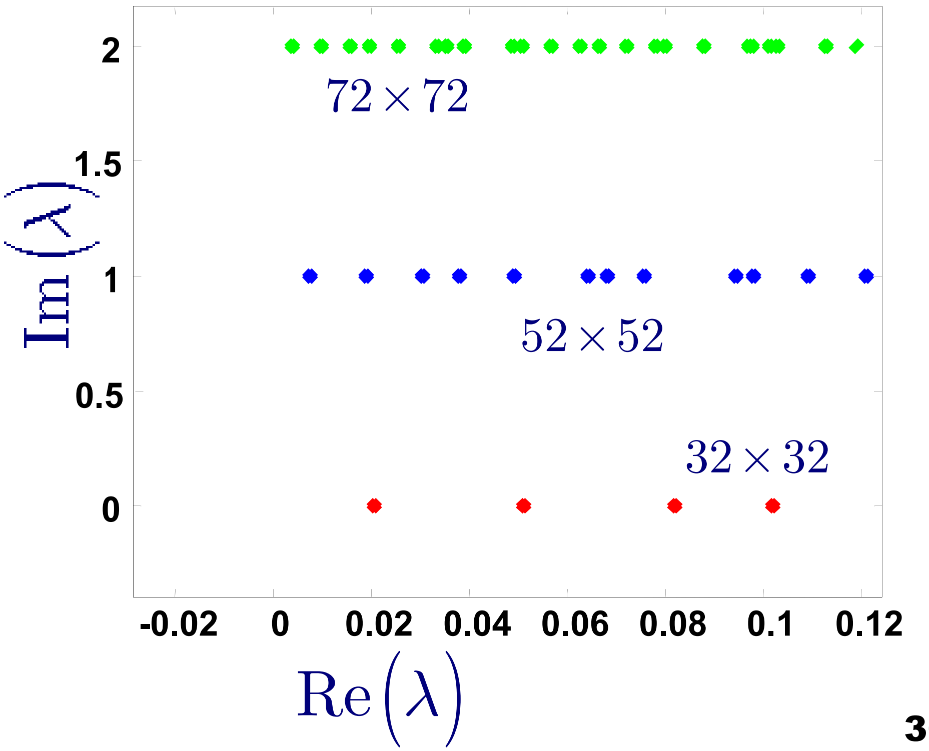
Eigenvalues of the discrete Laplacian for three discretization sizes

For this problem, the smallest eigenvalues and the number of small eigenvalues matter most (as we shall consider later in more detail)

All eigenvalues for 32 x 32 mesh



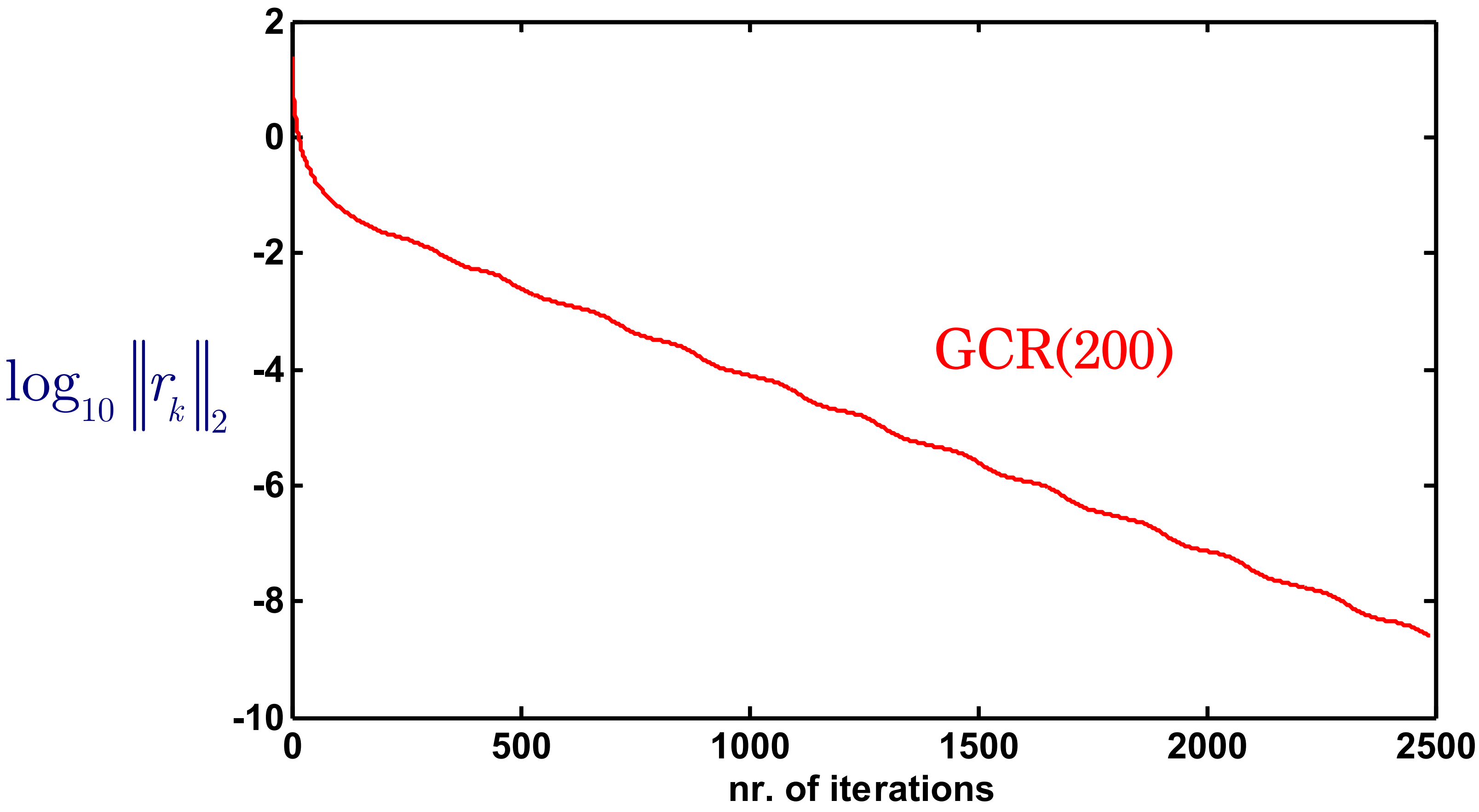
Eigenvalues close to 0 for all three meshes



Experiments with GCR

- Convergence for 322 x 322 problem using GCR restarting every 200 it.s
- Restarting saves memory and time (from excessive orthogonalizations)
- No preconditioning

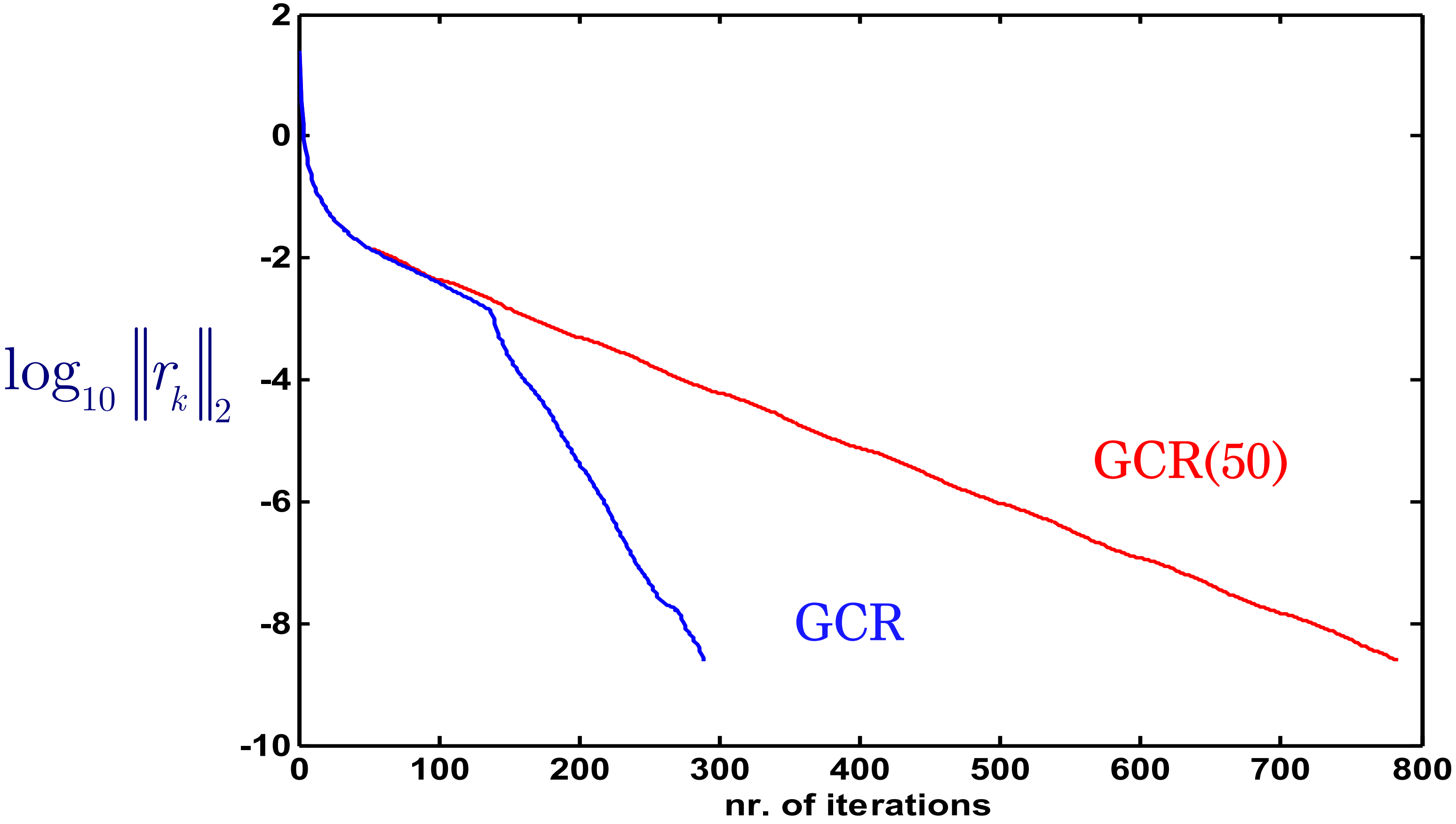
- Runtime: ~15 minutes on laptop (Intel Core 2 Duo P8700 @ 2.53 GHz)

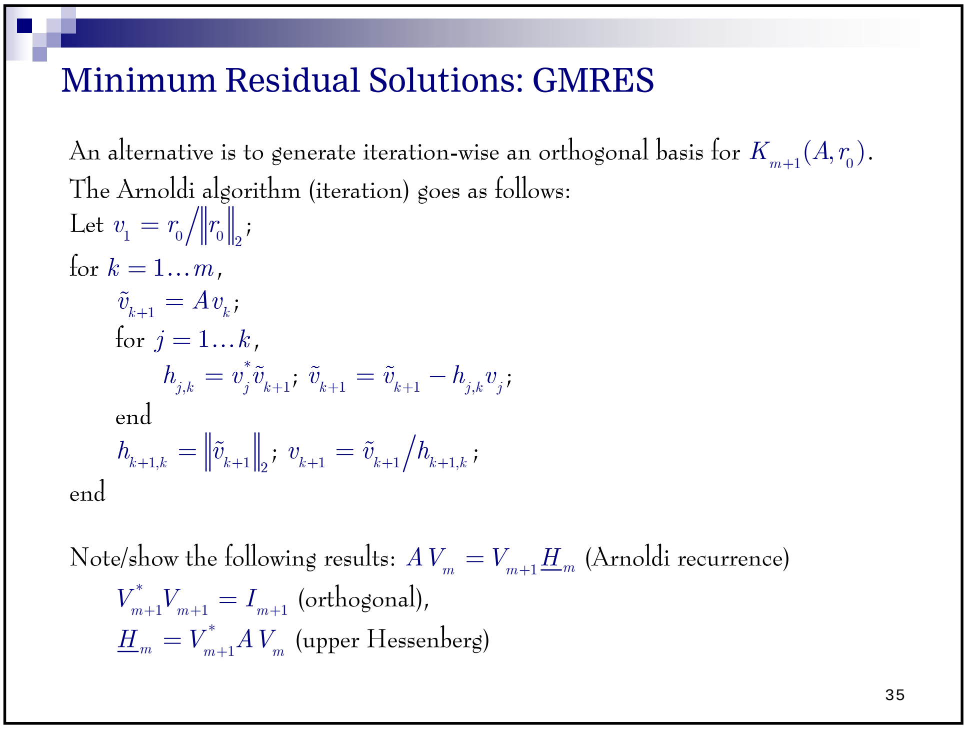


Experiments with GCR

- Convergence for 322 x 322 problem using preconditioned GCR
- Restarting saves memory and time (from excessive orthogonalizations)
- Preconditioner: Incomplete ILU without fill-in

- Runtimes: GCR(50) ~ 2 min., GCR (without restart) ~ 4 min.





Minimum Residual Solutions: GMRES

An alternative is to generate iteration-wise an orthogonal basis for $K_{m+1}(A, r_0)$.

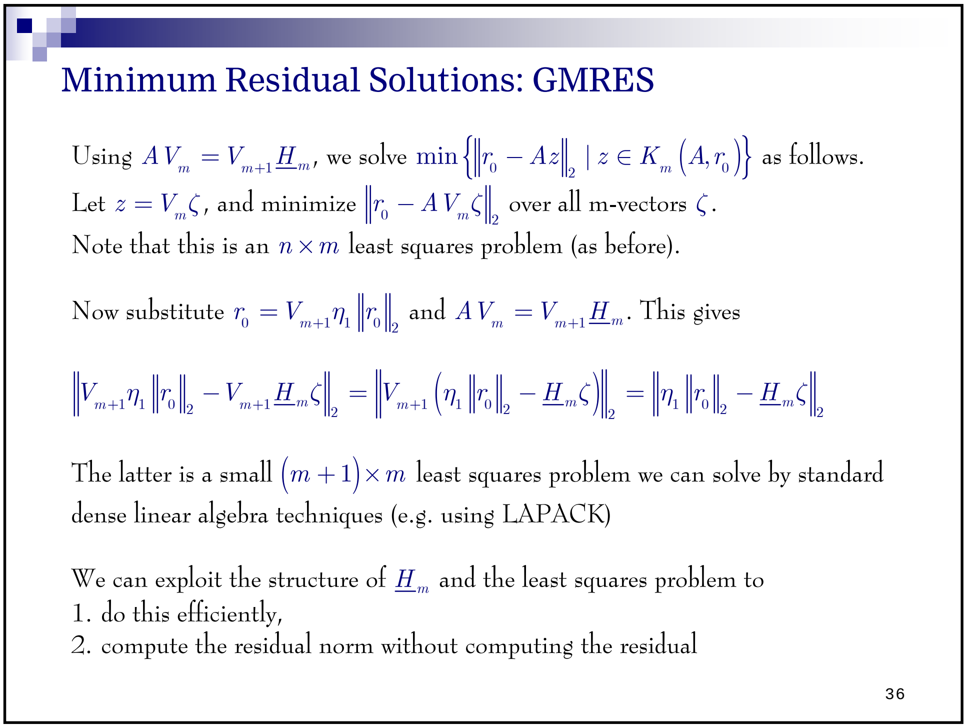
The Arnoldi algorithm (iteration) goes as follows:

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

Note/show the following results: $A V_m = V_{m+1} \underline{H}_m$ (Arnoldi recurrence)

$V_{m+1}^* V_{m+1} = I_{m+1}$ (orthogonal),

$\underline{H}_m = V_{m+1}^* A V_m$ (upper Hessenberg)



Minimum Residual Solutions: GMRES

Using $A V_m = V_{m+1} \underline{H}_m$, we solve $\min \left\{ \|r_0 - A z\|_2 \mid z \in K_m(A, r_0) \right\}$ as follows.

Let $z = V_m \zeta$, and minimize $\|r_0 - A V_m \zeta\|_2$ over all m-vectors ζ.

Note that this is an $n \times m$ least squares problem (as before).

Now substitute $r_0 = V_{m+1} \eta_1 \|r_0\|_2$ and $A V_m = V_{m+1} \underline{H}_m$. This gives

$$\left\| V_{m+1} \eta_1 \|r_0\|_2 - V_{m+1} \underline{H}_m \zeta \right\|_2 = \left\| V_{m+1} \left(\eta_1 \|r_0\|_2 - \underline{H}_m \zeta \right) \right\|_2 = \left\| \eta_1 \|r_0\|_2 - \underline{H}_m \zeta \right\|_2$$

The latter is a small $(m+1) \times m$ least squares problem we can solve by standard dense linear algebra techniques (e.g. using LAPACK)

We can exploit the structure of $\underline{H}_m$ and the least squares problem to

1. do this efficiently,
2. compute the residual norm without computing 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 $\boldsymbol{m}$ Givens rotations:

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

GMRES

Theorem: An unreduced $(m+1)\times m$ Hessenberg matrix is nonsingular.
(unreduced means no zeros on subdiagonal)

Proof: ?

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:
Stepwise compute $G_i^H(G_{i-1}^H \cdots G_1^H \underline{H}_i)$ and $G_i^H(G_{i-1}^H \cdots G_1^H e_1 \|r_0\|_2)$
In Arnoldi step, update $\underline{H}_{i-1}$ with new column; then 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

The least squares system now 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 by solving $R_i y_i = \underline{Q}_i^H e_1 \|r_0\|_2$

Note LS residual norm equals the norm of the actual residual:
$\|r_i\|_2 = |\tilde{q}_{i+1}^H e_1| \, \|r_0\|_2$ ($\tilde{q}_{i+1}$ since it changes with i):

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

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

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-DECOMP.: $\underline{H}_k = Q_{k+1} \underline{R}_k$
    $\|r_k\|_2 = |\tilde{q}_{k+1}^H e_1| \, \|r_0\|_2$
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

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 $K^m(A, Ar_0) = \text{range}(C_m)$. (why not?)

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

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

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.

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:

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

where $V_{m+1}\underline{Q}_m$ is unitary and R_m is uppertriangular

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 possibly 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: $y = R_m^{-1}\underline{Q}_m^H f$.

Minimum Residual Solutions: GMRES

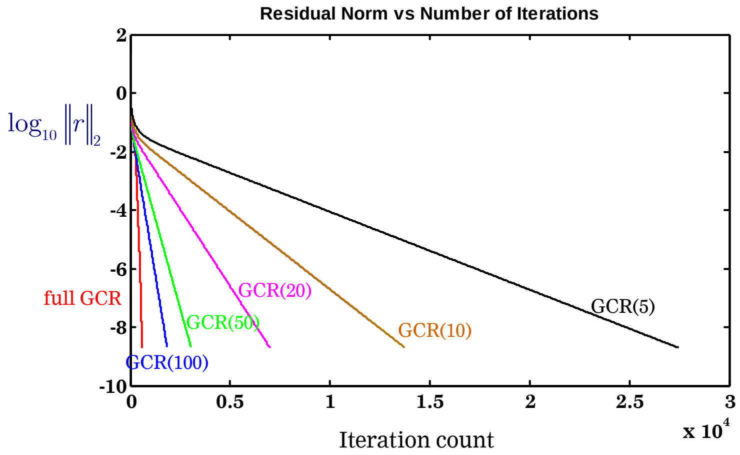
```
GMRES: $Ax = b$
Choose $x_0$, tolerance $\varepsilon$; set $r_0 = b - Ax_0$; $v_1 = r_0 / \left\|r_0\right\|_2$, $k = 0$.
while $\| r_k \|_2 \geq \varepsilon$ do
    $k = k + 1$
    $\tilde{v}_{k+1} = A v_k$;
    for $j = 1 \ldots k$,
        $h_{j,k} = v_j^* \tilde{v}_{k+1}$; $\tilde{v}_{k+1} = \tilde{v}_{k+1} - h_{j,k} v_j$;
    end
    $h_{k+1,k} = \left\|\tilde{v}_{k+1}\right\|_2$; $v_{k+1} = \tilde{v}_{k+1} / h_{k+1,k}$;
    Solve LS $\min_\zeta \left\| \eta_1 \left\|r_0\right\|_2 - \underline{H}_k \zeta \right\|_2 \;\left( = \left\|r_k\right\|_2 \right)$ by construction
    (actually we update the solution rather than solve from scratch – see later)
end
$x_k = x_0 + V_k \zeta_k$;
$r_k = r_0 - V_{k+1} \underline{H}_k \zeta_k = V_{k+1} \left( \eta_1 \left\|r_0\right\| - \underline{H}_k \zeta_k \right)$ or simply $r_k = b - Ax_k$
```

Convergence Restarted GCR

Test problem on unit square: 202×202 grid points

Interior: $-\nabla \cdot \left(\nabla u \right) = 0$ Boundary $u = 1$ for $x = 0$ and $y = 1$

$u = 0$ elsewhere

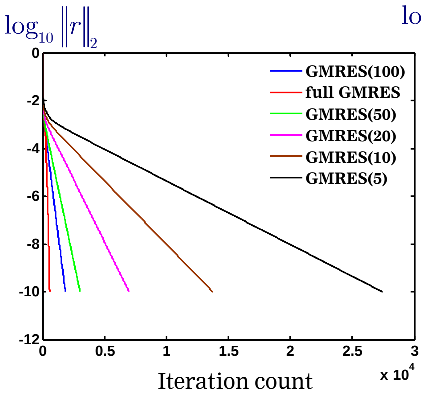


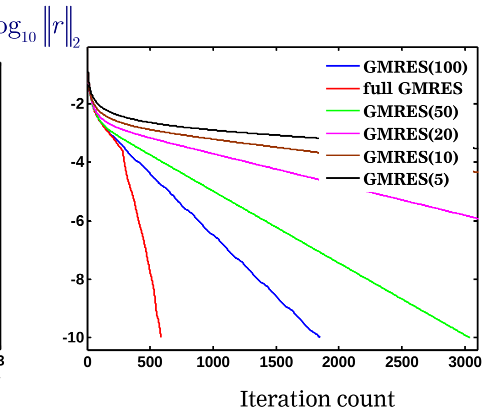
Convergence restarted GMRES

Test problem on unit square: 202×202 grid points

Interior: $-\nabla \cdot (\nabla u) = 0$ Boundary $u = 1$ for $x = 0$ and $y = 1$

$u = 0$ elsewhere





GMRES vs GCR

GMRES(m)	200 x 200 unknowns		
	time (s)	iterations	log10(||r||/||b||)
full	72.888	587	-10
100	40.256	1851	-10
50	41.087	3043	-10
20	63.604	6985	-10
10	111.26	13761	-10
5	199.42	27451	-10

rGCR(m)	200 x 200 unknowns		
	time (s)	iterations	log10(||r||/||b||)
full	215.87	587	-10
100	114.04	1851	-10
50	97.89	3043	-10
20	103.56	6985	-10
10	131.69	13761	-10
5	180.88	27451	-10

Iterative Methods and Multigrid

4. Optimal Krylov Subspace Methods with short recurrences

MINRES (I)

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

```
$v_1 = r_0 / \| r_0 \|_2;$
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 $H_{m+1,m} = V_{m+1}^H A V_m$.

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

MINRES (2)

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 (3)

A Hermitian upper Hessenberg matrix is tridiagonal!

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

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 $\boldsymbol{m}$ iterations of GMRES?

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

Proof:

MINRES (4)

Proof:

Consider $v_j^H A v_i = \overline{v_i^H A^H v_j} = \overline{v_i^H A v_j}$.

Since $v_j \in K^j(A, v_1)$, we have $A v_j \in K^{j+1}(A, v_j)$.

We know $v_i \perp \mathbf{span}\{v_1, \ldots, v_{i-1}\} = K^{i-1}(A, v_1)$;

so if $j+1 \le i-1 \Leftrightarrow j \le i-2$ then $v_i \perp A v_j$ and $v_i^H A v_j = 0$.

MINRES (5)

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 $y_m = \textbf{arg min} \| r_0 - AV_m y \|_2$ 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 (6)

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 $\boldsymbol{v_i}$?

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

We need a trick.

MINRES (7)

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 (9)

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 (independent of # iterations)?

Do we need r_m to continue the iteration?

What would be an update formula for r_m?

MINRES (10)

MINRES: $Ax = b$

```
CHOOSE $x_0 \rightarrow r_0 = b - Ax_0$ AND $tol$, SET $k = 0$;
$v_1 = r_0/\|r_0\|_2$
WHILE $\|r_k\| > tol$ DO
    $k = k + 1$;
    $\tilde{v}_{k+1} = Av_k - t_{k,k}v_k - t_{k-1,k}v_{k-1}$; (ignore indices less than zero)
    $t_{k+1,k} = \|\tilde{v}_{k+1}\|_2$; $v_{k+1} = \tilde{v}_{k+1}/t_{k+1,k}$;

    UPDATE QR:   $Q_{k+1} = Q_k G_k$; $\underline{R}_k = G_k^H(Q_k^H \underline{T}_k)$;
                 $\hat{y}_{k,k} = q_k^H e_1 \|r_0\|_2$
                 $\rightarrow \underline{Q}_k, R_k, \hat{y}_k \equiv \underline{Q}_k^H e_1 \|r_0\|_2$;
    $w_k = r_{k,k}^{-1}(v_k - w_{k-1}r_{k-1,k} - w_{k-2}r_{k-2,k})$;
    $x_k = x_{k-1} + w_k \hat{y}_{k,k}$
END
```

Conjugate Gradients (1)

Hermitian matrices: Error minimization in the A-norm

We are solving $Ax = b$ with initial guess $x_0 \to r_0 = b - Ax_0$ and
$\hat{x}$ is the solution to $Ax = b$.
The error at iteration i is $\varepsilon_i = \hat{x} - (x_0 + z_i)$,
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
$z_i = \mathbf{arg\ min}\{\|\hat{x} - (x_0 + z)\|_A : z \in K^i(A, r_0)\}$ 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 (2)

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

We know $K^i(A, r_0) = \mathbf{span}\{r_0, r_1, \ldots, r_{i-1}\}$.

This gives $r_k \perp_A (\hat{x} - x_0 - z_i)$ **for** $k = 0, \ldots, i-1 \quad \Leftrightarrow$

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

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

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

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

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

Conjugate Gradients (3)

So we can minimize the error by choosing the update such that the new residual is orthogonal to all previous residuals. Hence, the name orthogonal residual methods.

(note the comparison between Orthomin(1) and Steepest Descent)

We can generate an orthogonal basis for the Krylov subspace using the Lanczos iteration, the 3-term recurrence version of the Arnoldi-iteration.

Conjugate Gradients (4)

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 algorithm generates the recurrence relation:

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

Conjugate Gradients (5)

Use Lanczos orthonormal basis for minimizing A-norm of error.

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

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: $AQ_i = Q_i T_i + \beta_i q_{i+1} e_i^T$

Solve $r_0 - AQ_i y_i \perp Q_i \Leftrightarrow Q_i^H(\|r_0\|_2 q_1 - AQ_i y_i) = \mathbf{0} \Leftrightarrow$

$Q_i^H(\|r_0\|_2 q_1 - AQ_i y_i) = \mathbf{0} \Leftrightarrow \|r_0\|_2 e_1 - Q_i^H AQ_i y_i = \mathbf{0}$

Notice $\textbf{range}(Q_i) = \textbf{span}\{r_0, r_1, \ldots, r_{i-1}\}$.

$AQ_i = Q_i T_i + \beta_i q_{i+1} e_i^T \Rightarrow Q_i^H AQ_i = T_i$

So we reduced our problem to solving $\|r_0\|_2 e_1 - T_i y_i = \mathbf{0}$:

$y_i = T_i^{-1} e_1 \|r_0\|_2$

Conjugate Gradients (6)

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 $y_i = L_i^{-H} D_i^{-1} L_i^{-1} e_1 \|r_0\|_2$, where L_i is unit lower bi-diagonal with lower diagonal coeff.s, $l_1, l_2, \ldots, l_{i-2}$ (index gives the column)

Change of variables: $P_i = Q_i L_i^{-H}$ and $\hat{y}_i = D_i^{-1} L_i^{-1} e_1 \|r_0\|_2$: $Q_i y_i = P_i \hat{y}_i$

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 i*th* comp of vector $\hat{y}_i$)

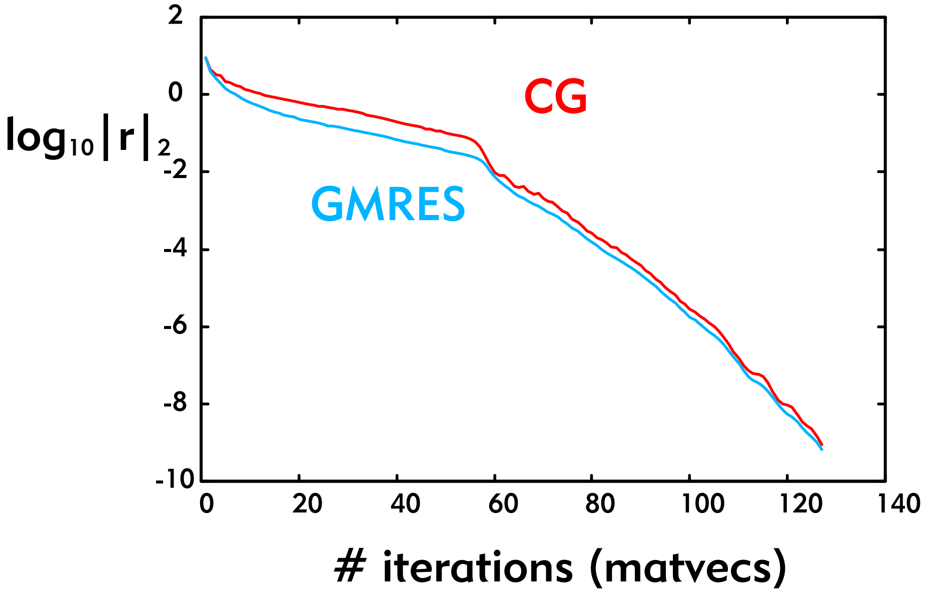
Conjugate Gradients (7)

This leads to the coupled two-term recurrence form of CG

```
CG algorithm: $Ax = b$
CHOOSE $x_0 \longrightarrow r_0 = b - Ax_0$;
$p_1 = r_0; i = 0$

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



A SPD, Solve $Ax = b$ by iterative method
$A \in \mathbb{R}^{n \times n}$, $b \in \mathbb{R}^n$. Let $\hat{x}$ denote the solution $(A\hat{x} = b)$.

Conjugate Gradients (CG)

A defines inner product $\langle x, y \rangle_A = y^* A x$ and

$$\|x\|_A = (x^* A x)^{1/2}$$

Often matrices from discretized PDE's are very sparse, so matrix-vector product very cheap: $O(n)$ work. However, std direct method, Gaussian elim. takes $O(n^3)$ work. That's where iterative methods come in.

At iteration m, method picks approx. solution

$$x_m \in K_m(A, b) = \operatorname{span}\{b, Ab, A^2 b, \dots, A^{m-1} b\}$$

such that

$\|\hat{x} - x_m\|_A$ minimal. (in this sense, the method is optimal)

Theo: $x_m = \arg\min\limits_{x \in K_m} \|\hat{x} - x_m\|_A \iff \hat{x} - x_m \perp_A K_m(A, b)$

We first consider a general approach to implement this method, and then derive a very efficient algorithm. We follow the recipe in lemma 8.9.

Pick basis $p_0, p_2, \dots, p_{m-1}$ for $K_m(A, b)$ and set $x_m = \sum_{j=0}^{m-1} p_j \xi_j$,

such that $\hat{x} - x_m \perp_A K_m(A, b)$

Since $\{p_0, \dots, p_{m-1}\}$ basis for $K_m(A,b)$, the orthog. relation gives

$$\langle \hat{x} - \sum_{j=0}^{m-1} p_j \xi_j , p_i \rangle_A = 0 \quad \text{for } i = 0 .. m-1$$

Form Gram matrix and solve the system

$$p_i^* A \left(\hat{x} - \sum_{j=0}^{m-1} p_j \xi_j\right) = 0 \iff p_i^* b - \sum_{j=0}^{m-1} p_i^* A p_j \xi_j = 0$$

$$\text{for } i = 0 .. m-1$$

Note that writing out these equations, we get

$$G : G_{ij} = p_i^* A p_j = \langle p_j , p_i \rangle_A \quad \text{and } f : f_i = p_i^* A \hat{x} = p_i^* b$$
$$= \langle \hat{x}, p_i \rangle_A$$

Solve for the unknown expansion coefficients ξ_j.

$$G \xi = f \iff \xi = G^{-1} f$$

$$x_m = \sum_{j=0}^{m-1} p_j (G^{-1} f)_j \qquad \text{(optimal approx. from } K_m(A,b))$$

This also provides some other useful/interesting prop.s.

We cannot monitor the error $\hat{x} - x_m$, so we monitor the residual $r_m = b - A x_m = A(\hat{x} - x_m) = A e_m$ (e_m error).

$$\langle \hat{x} - x_m , p \rangle_A = 0 \quad \text{for all } p \in K_m(A,b) \iff$$
$$p^* A (\hat{x} - x_m) = p^* r_m = 0 \quad \text{for all } p \in K_m(A,b)$$

So, $\quad r_m \perp K_m(A,b) \quad$ (in std. inner product)

Next, we consider an efficient algorithm.

At each step, we extend the search space $K_m(A,b)$ to $K_{m+1}(A,b)$, pick additional p_m s.t. $\{p_0, p_1, \dots, p_m\}$

is a basis for $K_{m+1}(A, b)$.

However, we need to update and solve the Gram matrix system at each step. In general, ξ changes in each component, so we must form x_{m+1} from scratch, and store all vectors $p_0, p_1, \dots$. We can do better.

If we pick the vectors p_j A-orthogonal (conjugate), that is $\langle p_j, p_i \rangle_A = p_i^* A p_j = 0$, for $i \neq j$, G is diagonal. Solving for G is then cheap and solving

$$\begin{pmatrix} g_{11} & & & \\ & \ddots & & \\ & & g_{mm} & \\ & & & g_{m+1,m+1} \end{pmatrix} \begin{pmatrix} \xi_0 \\ \vdots \\ \xi_{m-1} \\ \xi_m \end{pmatrix} = \begin{pmatrix} p_0^* b \\ \vdots \\ p_{m-1}^* b \\ p_m^* b \end{pmatrix}$$

leaves $\xi_0 \cdots \xi_{m-1}$ the same. This suggests that we can update x_m as we go : $x_{m+1} = x_m + p_m \xi_m$

and discard the p_j vectors. However, we need to orthogonalize the p vectors. We also need to extend the Krylov space.

We saw that $r_m \perp K_m(A, b)$. In addition $x_m \in K_m(A,b)$ and therefore $r_m = b - Ax_m \in K_{m+1}(A,b)$. Since

$$r_m \perp K_m(A, b), \quad r_m \notin K_m(A,b)$$

so $K_{m+1}(A,b) = \operatorname{span}\{r_m\} \oplus K_m(A,b)$

Note the direction $\operatorname{span}\{r_m\}$ in $K_{m+1}(A,b)$ and orthog. to $K_m(A,b)$ is unique.

This suggests we pick $p_m = r_m + \sum_{j=0}^{m-1} p_j \gamma_j$ s.t. $p_j^* A p_m = 0$ for $j = 0 .. m-1$.

It turns out we get this mostly for free:

$$p_i^* A r_m + \sum_{j=0}^{m-1} p_i^* A p_j \gamma_j = 0 \iff p_i^* A r_m + p_i^* A p_i \gamma_i = 0$$

$\hookrightarrow p_i^* A p_j = 0 \quad i \neq j$

So, $\gamma_i = -\dfrac{p_i^* A r_m}{p_i^* A p_i}$

Note that $p_i^* A r_m = \overline{r_m^* A p_i}$. Since $p_i \in K_{i+1}(A, b)$, $A p_i \in K_{i+2}(A, b)$ and $r_m \perp K_m(A, b)$.

So, $r_m \perp A p_i$ for $m \geqslant i+2 \iff i \leq m-2$.

So, only need to orthog. against p_{m-1}: $p_m = r_m + \gamma_{m-1} p_{m-1}$

$$\iff \quad p_m = r_m - \frac{p_{m-1}^* A r_m}{p_{m-1}^* A p_{m-1}} p_{m-1}$$

$$x_{m+1} = x_m + p_m \xi_m = x_m + p_m \frac{p_m^* b}{p_m^* A p_m}$$

$$r_{m+1} = r_m - A p_m \xi_m$$

Few more simplifications:

$$r_{m-1} = r_{m-2} - A p_{m-2} \xi_{m-2} = r_{m-3} - A p_{m-3} \xi_{m-3} - A p_{m-2} \xi_{m-2}$$

$$\implies \quad b = r_{m-1} + \sum_{i=0}^{m-2} A p_i \xi_i$$

$$p_{m-1}^* b = p_{m-1}^* r_{m-1} + \sum_{i=0}^{m-2} \cancel{p_{m-1}^* A p_i} \xi_i$$

Moreover, $p_{m-1} = r_{m-1} + p_{m-2}\gamma_{m-2} \Rightarrow$

$$r_{m-1}^* p_{m-1} = r_{m-1}^* r_{m-1} \quad \left(= \overline{p_{m-1}^* b} \right)$$

Analogously $p_{m-1}^* A r_m = \overline{r_m^* A p_{m-1}}$ and

$$r_m^* r_m = \cancel{r_m^* r_{m-1}} - r_m^* A p_{m-1} \xi_{m-1} \iff$$

$$r_m^* A p_{m-1} = \frac{r_m^* r_m}{\xi_{m-1}} = r_m^* r_m \cdot \frac{p_{m-1}^* A p_{m-1}}{p_{m-1}^* b}$$

$$= r_m^* r_m \cdot \frac{p_{m-1}^* A p_{m-1}}{r_{m-1}^* r_{m-1}}$$

So, $\dfrac{p_{m-1}^* A r_m}{p_{m-1}^* A p_{m-1}} = \dfrac{r_m^* r_m}{r_{m-1}^* r_{m-1}} \cdot \cancel{\dfrac{p_{m-1}^* A p_{m-1}}{p_{m-1}^* A p_{m-1}}}$ (is real)

$$x_0 = 0, \; r_0 = b, \; p_0 = r_0$$

$$x_m = x_{m-1} + p_{m-1} \cdot \frac{r_{m-1}^* r_{m-1}}{p_{m-1}^* A p_{m-1}} = x_{m-1} + \alpha_{m-1} p_{m-1}$$

$$r_m = r_{m-1} - A p_{m-1} \cdot \frac{r_{m-1}^* r_{m-1}}{p_{m-1}^* A p_{m-1}} = r_{m-1} - \alpha_{m-1} A p_{m-1}$$

$$p_m = r_m - p_{m-1} \frac{r_m^* r_m}{r_{m-1}^* r_{m-1}} = r_m - \beta_{m-1} p_{m-1}$$

Very efficient algorithm. It keeps only a few vectors, and per iteration does 1 matvec, a few dot products, and a few vector additions.