SCSC seminar 1.

based on Trefethen &

Alex Barnett : GMRES generalised minimum residual

- an iterative solver
- goal : given vector $b \in \mathbb{C}^N$
  ↳ complex, more general

$$A \in \mathbb{C}^{N \times N}$$ general (not necessarily symm)

"solve" $Ax = b$

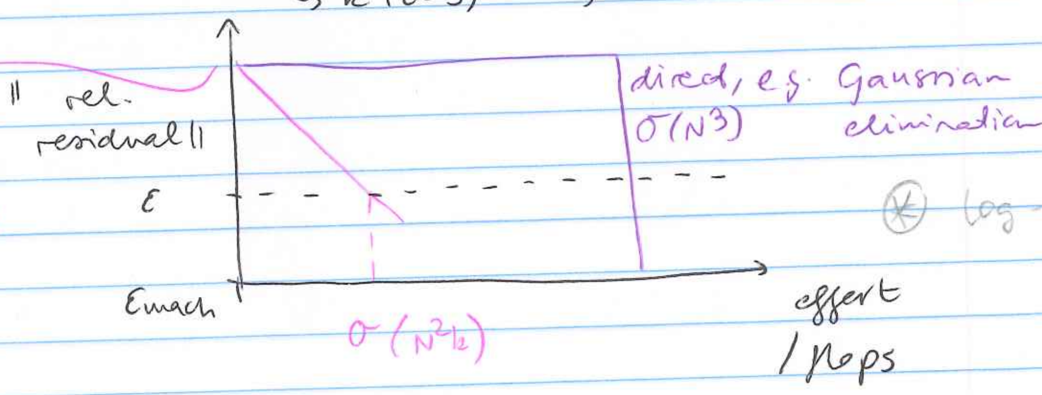
↖ find x with small

$$r := Ax - b$$

$$\frac{\|r\|}{\|b\|} < \varepsilon \leftarrow \text{user supplied tolerance}$$

direct v. iterative

↳ k iters, $O(1) \ll N$



...ding pre-conditioning may be ...d bit at the start

⊛ log-linear?

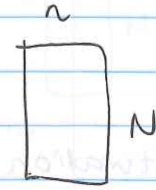
accesses A only through $x \to Ax$

matvec can be fast

- one of the gold-standard methods

- Krylov (space) matrix :

$$K_n = [\, b \quad Ab \quad A^2b \quad \dots \quad A^{n-1}b \,]$$
$$\downarrow \quad \downarrow \quad \downarrow \qquad \downarrow$$



K space $(= \text{col}\, K_n$ space spanned by their columns

c tells what lin comb of K's cols $= \{ K_n c : c \in \mathbb{C}^n \}$

to generate, matvecs : $b \curvearrowright Ab \curvearrowright A^2b \quad \dots$ (no AA!)

In exact arithmetic :

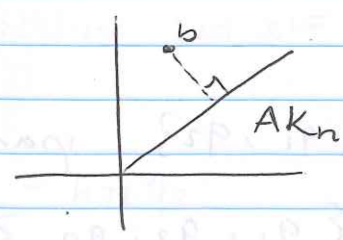
GMRES solves for c_n st

$$c_n := \underset{c \in \mathbb{C}^n}{\arg\min} \| A \underbrace{K_n c}_{x} - b \|$$

finding min residual in Krylov space

return $x_n = K_n c_n$

it's a least squares



algorithmically, not touching krylov space, bc it's ill-conditioned

this happens when there is one dominant eigenvalue

all pointing in almost the same dir

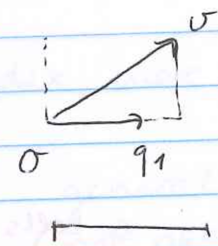
Leslie: find best polynomial approx of A^{-1}. Residual is guaranteed to decrease

$\Rightarrow$ build orthonormal basis for K_n without building K_n

v is a temp variable gets overwritten

iteration 1

$$\begin{cases} q_1 = b/\|b\| \\ v \leftarrow A q_1 \quad \text{set equal to} \\ v \leftarrow v - (q_1^* v) q_1 \\ q_2 \leftarrow v/\|v\| \end{cases}$$



$\vec{q}_1 \cdot \vec{v} = q_1^* v$

"stable" Gram-Schmidt,

in normal GS this is done sequentially

iteration 2

$$\begin{cases} v \leftarrow A q_2 \\ v \leftarrow v - (q_1^* v) q_1 - (q_2^* v) q_2 \qquad v \perp q_1, q_2 \\ q_3 \leftarrow v/\|v\| \end{cases}$$

Arnoldi process

etc.

growing amount of work / iter

$O(Nn^2)$ work, $O(Nn)$ RAM

$\{q_1, q_2\}$ spans $K_2 = \{b, Ab\}$

$\{q_1, q_2, q_3\}$ spans K_3

comment: QR decomposition ⊛ ?

now constructed onb for K_n, Q_n

let $Q_n = [q_1 \; \dots \; q_n]$ stack orthogonal matrix, $Q_n^* Q_n = I_n$

$\downarrow$ $\downarrow$

n

N

$q_1^* v$ h_{11}

$q_2 \leftarrow v/\|v\|$ h_{21}

matrix elements of a new matrix

$q_1^* v$ h_{12}

$q_2^* v$ h_{22}

$q_3 \leftarrow v/\|v\|$ $\leftarrow h_{32}$

rewrite iteration 1:

$$h_{21} q_2 = A q_1 - h_{11} q_1 \Rightarrow A q_1 = h_{11} q_1 + h_{21} q_2$$

$$= [q_1 \; q_2] \begin{bmatrix} h_{11} \\ h_{21} \end{bmatrix}$$

stack

iteration 2:

$$h_{32} q_3 = A q_2 - h_{12} q_1 - h_{22} q_2$$

$$\Rightarrow A q_2 = [q_1 \; q_2 \; q_3] \begin{bmatrix} h_{12} \\ h_{22} \\ h_{32} \end{bmatrix}$$

tells you what coeffs are generated when A hits any element of the q-basis

$$A Q_2 = Q_3 \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ 0 & h_{32} \end{bmatrix}$$

$$(*) \quad A Q_n = Q_{n+1} \tilde{H}_n \longrightarrow \overbrace{\begin{bmatrix} * & * & * & \\ * & * & * & \\ 0 & * & * & \ddots \\ 0 & 0 & * & \\ \vdots & \vdots & & \end{bmatrix}}^{n} \Big\} n+1$$

diagonal

subdiagonal

upper triangular + 1 subdiag

"Hessenberg" form

Comment: if Q is symmetric, then by hitting LHS of $(*)$ w/ Q_n^*, LHS is symm, so RHS is symm, so $\tilde{H}_n$ is symm

GMRES: solve for y_n (vector) the Q-coeffs of x_n:

$$x_n = Q_n y_n$$

$$y_n = \underset{y \in \mathbb{C}^n}{\operatorname{argmin}} \| \underbrace{A Q_n}_{\downarrow (*)\ Q_{n+1} \tilde{H}_n} y - b \|$$

both in the span of Q_{n+1}

$\Rightarrow$ can hit from left with Q_{n+1}^* & norm is the same (just recomputes coeffs)

$\Rightarrow$

$$= \underset{y \in \mathbb{C}^n}{\operatorname{argmin}} \| \tilde{H}_n y - Q_{n+1}^{*} b \|$$

↓ by orthogonality,

$$\begin{bmatrix} \|b\| \\ 0 \\ 0 \\ \vdots \end{bmatrix}$$

this is a small least-squares problem, can solve via QR or anything, $O(n^3)$

(n is no of iterations!)

$= \|r_n\|$

$= \| Ax_n - b \|$

where $x_n = Q_n y_n$

showed performance against spectrum of matrix A.
* Spectrum (rather than size of A) determines performance.

$$\|r_{n+1}\| \leq \|r_n\| \quad \text{because}$$

$$\underbrace{\operatorname{col} K_{n+1} \supset \operatorname{col} K_n}_{}$$

↑ encloses

it's a least squares in a larger space