This week and next: Krylov subspace methods for $Ax=b$

- $A \in \mathbb{C}^{n\times n}$, nonsingular ($\det A \neq 0$), $b \in \mathbb{C}^n$, solve $Ax = b$
- Krylov subspace: $y \in \mathbb{C}^n$

$$K_k(A, y) = \text{span}\{y, Ay, \cdots, A^{k-1}y\}$$

<u>Goal</u>: Iterative solvers for $Ax=b$ using Krylov subspaces.

- Note that $K_k(A, y) \subseteq K_{k+1}(A, y)$ but there is no guarantee that $\dim K_k(A, y) = n$ for large enough k. ($k \geq n$)

  (think about x_0 being an eig. vector of A)

  i.e., $K_k(A, y)$ may not span the whole $\mathbb{C}^n$.

– Why is it possible, at least in theory, to find a good approximation of $x = A^{-1}b$ in some Krylov subspace?

This is guaranteed by the following theorem:

<u>Thm (Cayley–Hamilton)</u>

Let $p(\lambda)$ be the characteristic polynomial of A,

i.e. $p(\lambda) = \det(\lambda I - A)$, then $p(A) = 0$ ▨

– Using this theorem, we show that $x = A^{-1}b \in K_n(A, b)$:

Indeed, let $p(\lambda)$ be characteristic polynomial of A,

i.e., $p(\lambda) = \lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0$

then $a_0 = p(0) = \det(-A) = (-1)^n \det(A) \neq 0$

so by $p(A) = 0$, i.e. $A^n + a_{n-1}A^{n-1} + \cdots + a_1 A + a_0 = 0$

multiply both sides by A^{-1}, we get

$$A^{n-1} + a_{n-1} A^{n-2} + \cdots + a_1 I + a_0 A^{-1} = 0$$

$$\Rightarrow A^{-1} = -\frac{1}{a_0} A^{n-1} - \frac{a_{n-1}}{a_0} A^{n-2} - \cdots - \frac{a_1}{a_0} I$$

$$\Rightarrow x = A^{-1} b = -\frac{1}{a_0} A^{n-1} b - \cdots - \frac{a_1}{a_0} b \in K_n(A, b)$$

- That is to say, $x = A^{-1} b \in K_n(A, b)$

  If we design our algorithm "good" enough, there is a hope that the algorithm terminates in n steps.

- Projection methods

- Given a subspace $K \subseteq \mathbb{C}^n$, projection methods aim to find an approximate solution to $Ax = b$ from K.
  (search subspace.)

- If $\dim K = k$, then we need k constraints to be able to extract a unique approximation.

  A typical way is to impose k linearly (independent) orthogonality conditions, which force the residual $b - Ax$ to be orthogonal to k linearly independent vectors.
  (test subspace $\mathcal{L}$)

  i.e. find $\tilde{x} \in K$, such that $b - A\tilde{x} \perp \mathcal{L}$

- To make use of the initial guess x_0, we want to formulate the problem as

$$\boxed{\text{find } \tilde{x} \in x_0 + K, \text{ such that } b - A\tilde{x} \perp \mathcal{L}}$$

- Let $V = [v_1 \cdots v_k]$ be a basis of K

$W = [w_1 \cdots w_k]$ be a basis of $\mathcal{L}$

then $b - A\tilde{x} \perp \mathcal{L}$ (let $\tilde{x} = x_0 + Vy$)

$$\iff \quad W^*(b - A(x_0 + Vy)) = 0$$

$$\iff \quad (W^*AV)y = W^*\underbrace{(b - Ax_0)}_{=:r_0} \qquad (*)$$

if W^*AV is invertible, then

$$\tilde{x} = x_0 + (W^*AV)^{-1} W^* r_0$$

Krylov subspace methods set $K = K_k(A, r_0)$.

the approximate solution has the form

$$\tilde{x} = x_0 + p_{k-1}(A) r_0 \qquad p_{k-1} : (k-1)\text{-degree polynomial}$$

When $x_0 = 0$, we have $\tilde{x} = p_{k-1}(A)\, b$.

There are many Krylov subspace methods. Depending different choices of $\mathcal{L}$, the methods can be classified into three different categories.

1) $\mathcal{L} = K_k(A, r_0)$

Algorithm: general A Hermitian A

Full orthogonalization method (FOM) Lanczos / D-Lanczos

conjugate gradient (CG)

2) $\mathcal{L} = A\mathcal{K}$

Algorithm: general A | Hermitian A

GMRES | MINRES

Generalized Conjugate Residual (GCR) | Conjugate Residual (CR)

ORTHOMIN

3) $\mathcal{L} = \mathcal{K}_k(A^*, r_0)$

Algorithm: non-symmetric Lanczos, Biorthogonal CG (BiCG), Quasi-Minimal Residual (QMR), Stablized BiCG

In order to compute $(*)$, we need to find an (orthonormal) basis for $\mathcal{K}_k(A, r_0)$

Recall Arnoldi's iteration.

Given $r_0 \in \mathbb{C}^n$, Arnoldi's iteration compute

$Q_k = [q_1 \cdots q_k] \in \mathbb{C}^{n\times k}$ orthonormal basis for $\mathcal{K}_k(A, r_0)$

such that

$$A Q_k = Q_k H_k + h_{k+1,k}\, q_{k+1} e_k^* \quad (\star)$$

where $H_k \in \mathbb{C}^{k\times k}$ is upper Hessenberg, $q_{k+1} \in \mathbb{C}^n$

$q_{k+1}^* Q_k = 0$, $\quad e_k = (0, \cdots, 0, 1)^T \in \mathbb{C}^n$ unit vector

we also have $Q_k^* A Q_k = H_k$

Suppose now we take $\mathcal{L} = \mathcal{K}_k(A, r_0)$

then we can take $V = W = Q_k \Rightarrow W^* A V = Q_k^* A Q_k = H_k$

from Arnoldi, $r_0 = \beta q_1 \Rightarrow W^* r_0 = \beta Q_k^* q_1 = \beta e_1$
$(\beta = \|r_0\|_2)$ $\quad (e_1 = (1, 0, \cdots, 0)^T)$

$$(*) \Rightarrow \quad H_k y_k = \beta e_1 \qquad (**)$$

$$\Rightarrow \quad y_k = H_k^{-1}(\beta e_1)$$

$$\Rightarrow \quad x_k = x_0 + Q_k y_k$$

- How accurate is x_k? (when to stop?)

Compute residual

$$r_k = b - A x_k = r_0 - A Q_k y_k$$

$$= r_0 - Q_k H_k y_k - h_{k+1,k}\, q_{k+1} e_k^* y_k$$

$$= \underbrace{r_0 - Q_k(\beta e_1)}_{=0} - h_{k+1,k} \underbrace{(e_k^* y_k)}_{=: y_k^{(k)}} q_{k+1}$$

$$\|r_k\|_2 = \|b - A x_k\|_2 = |h_{k+1,k}|\, |y_k^{(k)}| \quad \leftarrow \text{no overhead in computing residual}$$

- What if Arnoldi breakdown? ($h_{j+1,j} = 0$ for some $j \le k$)

(assume $j = k$ for simplicity)

 - In this case, $(**)$ is uniquely solvable.

 b.c. $A Q_k = Q_k H_k$ and if λ, w is an eigen pair of H_k

 then $A Q_k w = Q_k H_k w = \lambda Q_k w \Rightarrow (\lambda, Q_k w)$ is an eig. pair of A

 but since A is nonsingular, we know $\lambda \neq 0$

 - From the residual formula we know $b - A x_k = 0 \leftarrow$ exact soln!

This is lucky breakdown!

- Cost:
  - Arnoldi: Flops: $\underbrace{O(k^2 n)}$ dominant when k large. ($k \ll n$) Memory: $O(kn)$
    $+ \; k \times$ cost of Aq
  - Solve (**): since H_k is upper Hessenberg, we use QR fact. to solve (**), i.e. compute $H_k = \tilde{Q}\tilde{R}$, then solve $\tilde{R}y = \tilde{Q}^*(\beta e_1)$.

    $O(k^2)$ by Givens

    $O(k^2)$ for back sub + mat. vec. mult.

- What if k is already large but residual is still large?
  We can use restart method, i.e., set a new $x_0 \leftarrow x_k$ and rerun the algorithm

Algorithm (FOM (k))

Given $x_0 \in \mathbb{C}^n$, $k \geq 1$.

Compute $r_0 = b - Ax_0$, $\beta = \|r_0\|_2$,

Step 1: Run Arnoldi to compute $Q_k \in \mathbb{C}^{n\times k}$, $H_k \in \mathbb{C}^{k\times k}$
$h_{k+1,k} \in \mathbb{R}$, if $|h_{j+1,j}| < \varepsilon$ during iteration, move to step 2

Step 2: Compute $y_k = H_k^{-1}(\beta e_1)$, and $x_k = x_0 + Q_k y_k$
if $|h_{k+1,k}| \, |y_k^{(k)}| < \varepsilon$, break
else let $x_0 \leftarrow x_k$, recompute r_0 and β. goes to step 1.

- A key problem we didn't touched above: does H_k alway invertible?
  Unfortunately, the answer is no:

$$A = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & & \vdots & \vdots \\ \vdots & \vdots & \ddots & & \\ 0 & 0 & & 1 & 0 \end{bmatrix}, \quad A^T A = I. \text{ unitary} \Rightarrow |\lambda| = 1$$

let $b = (1, 0, \cdots, 0)^T = e_1$, $x_0 = 0$,

if we run Arnoldi, $q_1 = b = e_1$

and $Aq_1 = e_2 \perp q_1$, $\Rightarrow$ $q_i = e_i$, $i = 2 \cdots, k$

so $Q_k = \begin{bmatrix} I_k \\ 0 \end{bmatrix} \in \mathbb{R}^{n \times k}$

so $H_k = \underbrace{Q_k^* A Q_k}_{\substack{k \times k \text{ submatrix} \\ \text{in } A}} = \begin{bmatrix} 0 & & & & \\ 1 & 0 & & & \\ & \ddots & \ddots & & \\ & & & 0 & \\ & & & 1 & 0 \end{bmatrix}$ ← singular for all $1 \le k \le n-1$!

only when $k = n$, $H_k = A$, invertible, and $x_n = A^{-1} b$

FOM(k) totally fails !

When A is Hermitian positive definite, H_k is guaranteed to be invertible ($\Leftrightarrow$ CG)