New topic: Krylov subspace methods

Goal: approximately solve $Ax = \lambda x$ or $Ax = b$ quickly for large sparse $A \in \mathbb{C}^{n\times n}$ ($n >> 10^4$)

(Ax can be evaluated efficiently)

Previous:

- For $Ax = b$
  - Direct methods, GE cost $O(n^3)$
  - Stationary iterative methods, convergence is usually slow!
- For $Ax = \lambda x$
  - Power iteration: only single eig. val., convergence?
  - QR iteration: find all eig. val., but sparsity is not preserved during iteration, and need access to full matrix A (not available in some applications)

In practice, only need a few of A's largest/smallest eigenvalues

Today: Krylov subspace methods for $Ax = \lambda x$

Recall: Power iteration, A has eig. val. $\lambda_1 > \lambda_2 \geq \cdots$, eig. vec. $v_1, v_2, \cdots$

At k^{th} stage, given $x_{k-1} \in \mathbb{C}^n$

compute $\hat{x}_k = A x_{k-1}$

then $x_k = \hat{x}_k / \|\hat{x}_k\|_2$

Convergence: $\|x_k - \pm v_1\|_2 = O\left(\left|\frac{\lambda_2}{\lambda_1}\right|^k\right)$

Issue: 1) Slow when $\left|\frac{\lambda_2}{\lambda_1}\right| \approx 1$

2) Power iteration visits all the directions

$x_0, Ax_0, A^2x_0, \dots, A^k x_0, \dots$

but only makes use of the most recent $A^k x_0$

A lot of information is disregard.

We need a method that "learns from history"

- Krylov subspaces

Given $x_0 \in \mathbb{C}^n$, $A \in \mathbb{C}^{n\times n}$, the Krylov subspace is defined by

$$\mathcal{K}_k(A, x_0) = \operatorname{span}\{x_0, Ax_0, \cdots, A^{k-1}x_0\}$$

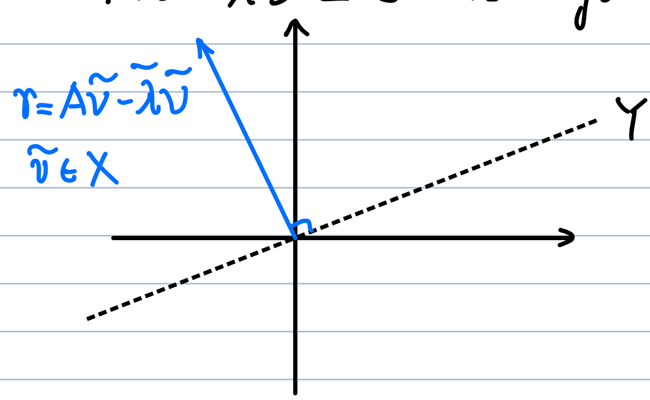
$$= \{ p(A)x_0 \in \mathbb{C}^n : p(A) = \sum_{l=0}^{k-1} t_l A^l \text{ is a polynomial of degree} \le k-1 \}$$

Krylov subspace methods: find <u>best</u> solution in $\mathcal{K}_k(A, x_0)$

- Projection method:

Idea: Find $\tilde{v} \in X \subseteq \mathbb{C}^n$, $\tilde{\lambda} \in \mathbb{C}$

s.t. $A\tilde{v} - \tilde{\lambda}\tilde{v} \subseteq \mathbb{C}^n$ is "good" enough



Petrov–Galerkin condition: $A\tilde{v} - \tilde{\lambda}\tilde{v} \perp Y$

X: search space, Y: test space

Special case: $X = Y = \mathbb{C}^n$, original eig. val. problem

When $X = Y$, let $Q_k = [\, q_1 \cdots q_k \,] \in \mathbb{C}^{n\times k}$

collect orthonormal basis spanning X, then

Galerkin condition $\iff$ $Q_k Q_k^*(A Q_k z - \tilde{\lambda} Q_k z) = 0$

$$\iff Q_k(\underbrace{Q_k^* A Q_k}_{B_k \in \mathbb{C}^{k\times k}} z - \tilde{\lambda} z) = 0$$

$$\iff B_k z = \tilde{\lambda} z$$

<u>Rayleigh-Ritz projection</u>

the problem size is reduced to $k \times k$

We call $\tilde{\lambda}$ Ritz value, $\tilde{v} = Q_k z$ Ritz vector

- Krylov subspace methods for eig. val. problem

  Use $X = Y = K_k(A, x_0)$.

Q: How to construct Q_k, i.e. an orthonormal basis for $K_k(A, x_0)$? (and B_k)

- Arnoldi's method

  Idea: apply Gram-Schmidt to $x_0, Ax_0, \dots, A^{k-1}x_0$

  Algorithm: Arnoldi's iteration

$$x_0 \longrightarrow Ax_0 \longrightarrow A^2x_0 \longrightarrow \cdots \longrightarrow A^{k-1}x_0$$

Orthonormalize Aq_i to $\{q_1, \dots, q_i\}$

$$\begin{array}{ccccccccc}
x_0 & & Aq_1 & & Aq_2 & & \cdots & & Aq_{k-1} \\
\Downarrow & \nearrow & \Downarrow & \nearrow & \Downarrow & \nearrow & \cdots & \nearrow & \Downarrow \\
q_1 & & q_2 & & q_3 & & & & q_k \\
\cap & & \cap & & \cap & & & & \cap \\
K_1(A, x_0) & & K_2(A, x_0) & & K_3(A, x_0) & & & & K_k(A, x_0)
\end{array}$$

By induction $\operatorname{span}\{q_1, \dots, q_k\} = K_k(A, x_0)$

Implementation:

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

$q_1 \leftarrow x_0 / \|x_0\|_2$

For $i = 1, \cdots, k$

$w_i \leftarrow A q_i$

For $j = 1, \cdots, i$

$h_{ji} \longleftarrow q_j^* w_i$

end

$w_i \leftarrow w_i - \sum_{j=1}^{i} h_{ji} q_j$

Classical GS
unstable
can be improved
by Modified GS

For $j = 1, \ldots, i$

$h_{ji} \leftarrow q_j^* w_i$

$w_i \leftarrow w_i - h_{ji} q_j$

end

$h_{i+1,i} = \|w_i\|_2$

$q_{i+1} = w_i / h_{i+1,i}$

end

Cost $\approx 2k^2 n$ for Classical/Modified G-S

$+ \underbrace{k \times \text{cost to evaluate } Aq_i}$

assume N_z nonzero elements per row
cost $\approx 2k N_z n$

Output: $(q_i)_{i=1}^{k+1}$, $(h_{ji})_{\substack{1 \leq i \leq k \\ 1 \leq j \leq i+1}}$

- $Q_k = [\, q_1 \; \cdots \; q_k \,] \in \mathbb{C}^{n \times k}$ forms a basis for $K_k(A, x_0)$

The Arnoldi iteration is essentially QR factorization for $K_k(A, x_0)$

$$[\, x_0 \;\; Ax_0 \; \cdots \; A^{k-1}x_0 \,] = [\, q_1 \; \cdots \; q_k \,] \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1k} \\ & r_{22} & \cdots & r_{2k} \\ & & \ddots & \\ & & & r_{kk} \end{bmatrix}$$

- From $Aq_i = \sum_{j=1}^{i+1} h_{ji} q_j$. $i = 1, \ldots, k$

$$A[\, q_1 \; \cdots \; q_k \,] = [\, q_1 \cdots q_k \mid q_{k+1} \,] \underbrace{\begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1k} \\ h_{21} & h_{22} & & \vdots \\ & h_{32} & \ddots & \\ & & \ddots & h_{kk} \\ \hline & & & h_{k+1,k} \end{bmatrix}}_{k \text{ columns}} \Bigg\} k+1 \text{ rows}$$

$$=: [Q_k \;\; q_{k+1}] \begin{bmatrix} H_k \\ h_{k+1,k} e_k^* \end{bmatrix} \quad (e_k = (0,\cdots,0,1)^* \in \mathbb{C}^k)$$

$$\Rightarrow \quad A Q_k = Q_k H_k + h_{k+1,k} \underbrace{q_{k+1}}_{Q_k^* q_{k+1} = 0} e_k^*$$ ← k-step Arnoldi decomposition

$$\Rightarrow \quad B_k := Q_k^* A Q_k = H_k$$ ← obtain Raleigh-Ritz projection onto $K_k(A, x_0)$ directly

Let $(\tilde{\lambda}, z)$ with $\|z\|_2 = 1$ be eig. pair of H_k. i.e. $H_k z = \tilde{\lambda} z$

$\tilde{v} = Q_k z$. then ← Ritz value / vector

$$\Rightarrow \quad A\tilde{v} = \tilde{\lambda}\tilde{v} + h_{k+1,k}(e_k^* z) q_{k+1}$$

$$\Rightarrow \quad (A - \underbrace{h_{k+1,k}(e_k^* z) q_{k+1} \tilde{v}^*}_{=: E}) \tilde{v} = \tilde{\lambda}\tilde{v}$$

$z = (z_1, \cdots, z_k)^T$

$=: E$, $\|E\|_2 = |h_{k+1,k}| \, |z_k|$ ← can be easily computed

Ritz value and vector solve the perturbed eig. val. problem

$$(A + E)\tilde{v} = \tilde{\lambda}\tilde{v}, \quad \|\tilde{v}\|_2 = 1$$

<u>Remark</u>: Ritz value and vectors can be found by QR iteration.

Since $H_k \in \mathbb{C}^{k \times k}$ and upper Hesenberg, cost $\approx O(k^2)$

<u>Remark</u>: Since $[x_0 \mid Ax_0 \cdots A^{k-1}x_0]$ becomes more and more singular,

G-S could be unstable. A Householder version of Arnoldi iteration can be applied to enforce orthogonality. But cost $\approx 4k^2 n - \frac{4}{3}k^3$. ← complete reorthogonalization

usually a selective reorthogonalization technique + MGS is good enough

<u>Remark</u>: If $h_{k+1,k} = 0$ for some k, Arnoldi iteration breakdowns

But this is a lucky breakdown because the Ritz values and Ritz vectors are eig. values and vectors of A.

If other eig. values are wanted, restart the iteration with vector ⊥ found eig. vec