Arnoldi iteration

4, 9, 11 May $\rightarrow$ no lecture!

Arnoldi iteration

Definition: given $A \in \mathbb{C}^{m\times m}$, $b \in \mathbb{C}^{m}$ $\quad m > n$
(usually m large, A sparse)

$$K_n(A,b) = \operatorname{span}\left(b, Ab, A^2b, \dots A^{n-1}b\right)$$
$$= \left\{ p(A)b \;,\; p \text{ polynomial with degree } < n \right\}$$

Indeed, if $p(z) = \alpha_0 + \alpha_1 z + \dots + \alpha_{n-1} z^{n-1}$, then

$$p(A)b = b\alpha_0 + Ab\alpha_1 + \dots + A^{n-1}b\,\alpha_{n-1}.$$

$K_n(A,b)$ is the span of n different vectors, so if they are lin. independent it has dimension n.

Many problems with large A, b have solutions that are well approximated by projecting them onto $K_n(A,b)$.

Even if there is no b, the problem of computing eigenvalues is also well approximated by its projection on $K_n(A,b)$

Let V_n ~~be~~ whose columns are an orthonormal basis for $K_n(A,b)$.
$(V_n^* V_n = I_n)$

Then, $P = V_n V_n^*$ is the orthogonal projection matrix onto $K_n(A,b)$

$$PAP = V_n \boxed{V_n^* A V_n} V_n^*$$

We can work with $v \in K_n(A,b)$ through their coordinates in the basis V_n, i.e. $y \in \mathbb{C}^n$ such that $v = V_n y$

$A_n = V_n^* A V_n \in \mathbb{C}^{n \times n}$ is the $n \times n$ matrix that represents the operator A restricted to that subspace

For many matrices, $\Lambda(A_n)$ is a good approximation of the <u>outer</u> eigenvalues of A, i.e., those at the extremes of the complex region where the spectrum $\Lambda(A)$ lies.

Same for eigenvectors: y eigenvector of $A_n \Rightarrow v = V_n y$ is a good approximation of an eigenvector of A

<u>Intuition:</u>

If one writes b in the eigenvector basis of A,

$$b = x_1\alpha_1 + x_2\alpha_2 + \dots + x_m\alpha_m \qquad Ax_i = x_i\lambda_i$$

λ_i, α_i scalars x_i vectors

$$A^k b = A^k x_1\alpha_1 + A^k x_2\alpha_2 + \dots A^k x_m \alpha_m$$
$$= \lambda_1^k x_1\alpha_1 + \lambda_2^k x_2\alpha_2 + \dots + \lambda_m^k x_m \alpha_m$$

larger for the λ_i that have larger absolute value

So the vector $A^k b$ are very close to being in $\text{span}(x_i : |\lambda_i| \text{ is large})$

Another example of using Krylov spaces to solve problems with large A: the solution of $Ax = b$ is well approximated (for some A) by $x \approx V_n y$

such that y solves $A_n y = V_n^* b$ ← $n\times n$ rather than $m \times m$

Problem: how to compute an o.n. basis matrix $V_n = \left[v_1 \middle| v_2 \middle| \dots v_n \right]$

$v_1, \dots v_n$ are an o.n. basis of $K_n(A,b)$?

First idea: V_n is the Q factor of

$$qr\left(\left[b, Ab, \dots A^{n-1}b\right]\right)$$

or (equivalently) the result of orthonormalizing these vector by Gram-Schmidt

But $\left[b, Ab, \dots A^{n-1}b\right]$ is terribly ill-conditioned:

$A^k b$ converge (up to good normalization) to the eigenvector x_i with largest $|\lambda_i|$.

Instead, we are going to do something slightly different.

Idea: do this in an incremental fashion

$$K_1(A,b) \subset K_2(A,b) \subset K_3(A,b) \subset \dots$$

$$\underbrace{\mathrm{span}(b)}_{v_1} \subset \underbrace{\mathrm{span}(b, Ab)}_{v_1, v_2} \subset \underbrace{\mathrm{span}(b, Ab, A^2 b, \dots)}_{v_1, v_2, v_3} \quad \dots$$

So we can just solve the problem incrementally:

given $(v_1, v_2, \dots v_j)$ basis of $K_j(A,b)$,

compute v_{j+1} such that $(v_1, v_2, \dots v_j, v_{j+1})$ is a basis of $K_{j+1}(A,b)$

To do this, it is sufficient to take $w \in K_{j+1}(A,b) \setminus K_j(A,b)$ and orthonormalize it w.r.t. all the previous v_i:

$$w = \underbrace{v_1\alpha_1 + v_2\alpha_2 + \dots + v_j\alpha_j}_{\text{known}} + \underbrace{v_{j+1}}_{\hookrightarrow \text{ unknown vector to}}\alpha_{j+1}$$

determine, orthonormal to the other v_i

<u>Gram-Schmidt</u>: compute $\alpha_1 = v_1^* w_0$

update $w_1 \leftarrow w_0 - v_1\alpha_1 = w - v_1 v_1^* w$

(Modified GS, MGS)

$$= v_2\alpha_2 + v_3\alpha_3 + \dots + v_j\alpha_j + v_{j+1}\alpha_{j+1}$$

compute $\alpha_2 = v_2^* w$

$$w_2 \leftarrow w_1 - v_2\alpha_2$$

$\vdots$

After j steps, I am left with $w_j = v_{j+1}\alpha_{j+1}$

so I can set $v_{j+1} = \frac{1}{\|w_j\|} \cdot w_j \qquad \alpha_{j+1} = \|w_j\|$

$$w = \alpha_{j+1} v_{j+1}$$

(Traditional GS)

(Note that I could also have computed $\alpha_1 = v_1^* w_0$, $\alpha_2 = v_2^* w_0$, ... $\alpha_j = v_j^* w_0$ and subtracted them all at the same time, but this is more unstable, so it is preferred to subtract them one by one)

Idea behind Arnoldi: take $w_0 = A v_j$.

<u>Lemma</u>: Suppose that all the vectors $b, Ab, \dots A^j b$ are linearly independent. Then,

1. $v_j \in K_j(A,b) \setminus K_{j-1}(A,b)$
2. $Av_j \in K_{j+1}(A,b) \setminus K_j(A,b)$
3. α_{j+1} in the algorithm above is $\neq 0$.

<u>Proof</u>: induction:

For $j=1$, $v_1 = \frac{b}{\|b\|}$ is a multiple of b, and we can take $K_0(A,b) = \{0\}$

Let us assume 1. holds for a certain j.

1. means that $v_j \in p(A)b$ where p is a polynomial of

degree exactly $j-1$, i.e.

$$v_j = b\gamma_1 + Ab\gamma_2 + A^2 b\gamma_3 + \dots + A^{j-1} b\gamma_j$$

with $\gamma_j \neq 0$

(Note that this expension is unique because we assumed that $b, Ab, \dots A^{j-1}b$ are lin. independent)

$$w_0 = Av_j = Ab\gamma_1 + A^2 b\gamma_2 + \dots + A^j b\gamma_j \quad \text{with } \gamma_j \neq 0.$$

Hence $w_0 \in K_{j+1}(A,b) \setminus K_j(A,b)$

Note that each update $w_i = w_{i-1} - v_i(v_i^* w_{i-1})$ subtracts a vector $v_i \in K_j(A,b)$.

So after each update $w_i \notin K_{j+1}(A,b) \setminus K_j(A,b)$.

After all updates, $w_j \in K_{j+1}(A,b) \setminus K_j(A,b)$, hence $w_j \neq 0$

$\alpha_{j+1} = \|w_j\| \neq 0$. $\quad v_{j+1} = \frac{1}{\|w_j\|} w_j \in K_{j+1}(A,b) \setminus K_j(A,b)$.

So we can continue. $\square$

---

3 things we want to return:

$V = [v_1 | v_2 | \dots \; v_{n+1}]$ nested bases: $v_1, v_2, \dots v_j$ basis for $K_j(A,b)$

H contains all the coefficients α_i that we computed:

$$w_{0,j} = Av_j = v_1\alpha_{1,j} + v_2\alpha_{2,j} + \dots + v_j\alpha_{jj} + v_j\alpha_{j+1,j}$$

$$H_{ij} = \alpha_{ij} \qquad \begin{bmatrix} \alpha_{1,1} & \alpha_{1,2} & \dots \\ \alpha_{2,1} & \alpha_{2,2} & \\ 0 & \alpha_{3,2} & \ddots \\ 0 & 0 & \\ \vdots & 0 & \\ 0 & 0 & \ddots \end{bmatrix}$$

$B = \|b\|$

At the j-th step,

$$A v_j = v_1 \alpha_{1,j} + v_2 \alpha_{2,j} + \dots + v_j \alpha_{j,j} + v_{j+1} \alpha_{j+1,j}$$

$$= \underbrace{\left[v_1 \middle| v_2 \middle| \dots \middle| v_{n+1} \right]}_{V_{n+1}} \cdot \underbrace{\begin{bmatrix} \alpha_{1,j} \\ \alpha_{2,j} \\ \vdots \\ \alpha_{j+1,j} \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{H_{:,j}}$$

Stacking together these relations,

$$A \underbrace{\left[v_1 \middle| v_2 \middle| \dots \middle| v_n \right]}_{V_n} = V_{n+1} \cdot H_{n+1,n}$$

$$\boxed{A V_n = V_{n+1} H_{n+1,n}}$$

```
function [V, H, beta] = arnoldi(A, b, n)
m = length(b);
V = zeros(m, n+1);
H = zeros(n+1, n);

beta = norm(b);
V(:, 1) = b / beta;
for j = 1:n
    w = A * V(:,j);  % continuation vector
    for i = 1:j
        H(i,j) = V(:,i)' * w;
        w = w - V(:,i) * H(i,j);
    end
    H(j+1,j) = norm(w);
    V(:,j+1) = w / H(j+1,j);
end
```

n products $A \cdot$ vector

$O(nnz(A))$

$O(m) \cdot O(n^2)$ times

$O(mn^2)$

Arnoldi is a "black-box" algorithm: I just need to have a function that computes $v \mapsto Av$.
I do not need to access A otherwise.

$$\boxed{AV_n = V_{n+1} H_{n+1,n}}$$

From this equality, we can compute

$$V_n^* A V_n = \underbrace{V_n^* V_{n+1}}_{\begin{bmatrix} 1 & & & \\ & \ddots & & \\ 0 & 0 \cdots & 1 & 0 \end{bmatrix}} H_{n+1,n} = \text{the first } n\times n \text{ block of } H_{n+1,n}$$

<u>Lemma</u>: if p is a polynomial of degree $\deg p < n$, then

$$p(A)b = V_n\, p(A_n) \underbrace{V_n^* b}_{= e_1\beta = \begin{bmatrix}\beta\\0\\0\\ \vdots\\0\end{bmatrix} \text{ in Arnoldi}} = V_n\, p(A_n)\, e_1 \beta$$

<u>Proof</u>: it is enough to prove it for $p(z) = z^k$, for $k = 0, 1, 2, \dots n-1$, by linearity

$$\text{RHS} = V_n A_n^k V_n^* b = V_n (V_n^* A V_n)^k V_n^* b = V_n V_n^* A V_n V_n^* A V_n V_n^* \cdots A \underbrace{V_n V_n^* b}_{=b}$$

$V_n V_n^* b = b$, because $V_n V_n^*$ is the projection on $K_n(A,b)$, and $b \in K_n(A,b)$

$$= V_n V_n^* A \cdots \cdots A \underbrace{V_n V_n^* A b}$$

$V_n V_n^* A b = Ab$, because $Ab \in K_n(A,b)$

$= \dots = A^k b \,{}^{= \text{LHS}}$, since all projections act on vectors in $K_n(A,b)$.

and $k<n$. $\square$

We proved that

$$p(A)b = V_n\, p(A_n)\, e_1 \beta \quad \text{for polynomials of degree } < n.$$

This suggests the approximation

$$\underset{\substack{\uparrow \\ \text{large sparse } A}}{f(A)}\, b \approx V_n \underbrace{f(A_n)}_{\substack{\uparrow \\ \text{same matrix function, but on a smaller } n\times n \text{ matrix}}} e_1 \beta$$