Termination in Arnoldi

Earlier, we wrote: "Assume for now that $\dim K_n(A, b) = n$, i.e., $b, Ab, \ldots, A^{n-1}b$ are linearly independent."

At some point this assumption must become false. For instance, assume we arrive at step $m = n$: $q_1, q_2, \ldots, q_m$ are a basis of $\mathbb{R}^m$, so

$$Aq_m = \beta_1 q_1 + \cdots + \beta_m q_m + 0$$

(without an additional term $\beta_{m+1} q_{m+1}$.)

We get a complete factorization of A:

$$AQ_m = Q_m H_m \implies A = Q_m H_m Q_m^T.$$

We can use it to solve linear systems, too, but most of the time we want to stop the iteration much earlier.

Breakdown in Arnoldi

Note that the 'breakdown' can happen much earlier: for instance, if b is an eigenvector of A, it happens already at $n = 1$.

Suppose that at step n

$$A^n b \in \text{span}(b, Ab, \ldots, A^{n-1} b).$$

Then,

$$Aq_n = \beta_1 q_1 + \cdots + \beta_n q_n + 0$$

and the vector q_{n+1} that continues the orthonormal basis is not uniquely determined.

We take $\beta_{n+1} = 0$, $q_{n+1} =$ any vector orthonormal to $q_1, \ldots, q_n$, and continue.

(At this point $q_1, \ldots, q_n, q_{n+1}$ is not anymore a basis of $K_{n+1}(A, b)$ — but we continue nevertheless.)

After the breakdown

In the end, we get a full factorization

$$AQ_m = H_m Q_m,$$

$$H_m = \left[\begin{array}{cccc|cccc} * & \cdots & * & * & * & \cdots & \cdots & * \\ * & \cdots & * & * & * & \cdots & \cdots & * \\ 0 & \ddots & * & * & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & * & * & * & \cdots & \cdots & * \\ \hline & & & 0 & * & \cdots & * & * \\ & & & & * & \cdots & * & * \\ & & & & 0 & \ddots & * & * \\ & & & & 0 & 0 & * & * \end{array}\right].$$

Arnoldi completed to a full factorization

$$A = Q_m H_m Q_m^T = \begin{bmatrix} Q_n & \widehat{Q} \end{bmatrix} \begin{bmatrix} H_n & L \\ 0 & M \end{bmatrix} \begin{bmatrix} Q_n & \widehat{Q} \end{bmatrix}^T$$

Consequence #1 The eigenvalues of A are the union of those of H_n and those of M.

Consequence #2 We can use this factorization to solve the linear system $Ax = b$:

$$\begin{aligned} x &= \begin{bmatrix} Q_n & \widehat{Q} \end{bmatrix} \begin{bmatrix} H_n & L \\ 0 & M \end{bmatrix}^{-1} \begin{bmatrix} Q_n & \widehat{Q} \end{bmatrix}^T b \\ &= \begin{bmatrix} Q_n & \widehat{Q} \end{bmatrix} \begin{bmatrix} H_n & L \\ 0 & M \end{bmatrix}^{-1} \begin{bmatrix} \|b\| e_1 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} Q_n & \widehat{Q} \end{bmatrix} \begin{bmatrix} \|b\| H_n^{-1} e_1 \\ 0 \end{bmatrix} = Q_n \|b\| H_n^{-1} e_1. \end{aligned}$$

(Check the computations, they are not obvious.)

These results as a theorem

Suppose that there is breakdown at step n in Arnoldi, i.e., $A^n b \in K_n(A, b)$. Then,

- if $H_n v = \lambda v$ is an eigenvalue of H_n, then $w = Q_n v$ satisfies $Aw = \lambda w$, i.e., it is an eigenvector of A with the same eigenvalue. (And, in particular, the eigenvalues of H_n are a subset of those of A.)
- the exact solution of $Ax = b$ is given by $Q_n \|b\| H_n^{-1} e_1$.

We can stop immediately at breakdown: we only need Q_n, H_n.

'Lucky breakdown': if it happens at an early step, we can solve linear systems (or compute eigenvalues) cheaply: costs n matrix-vector products + $O(mn^2)$.

When there is no breakdown

Let (λ, v) be an eigenpair of H_n; then $(\lambda, Q_n v)$ is called a Ritz pair (Ritz value, Ritz vector) of A.

Amazing fact: even without 'lucky breakdown', Ritz values are often good approximations of the eigenvalues of A.

Typically, they approximate the largest (in modulus) eigenvalues of A.

Example

```
% Generates 1000x1000 A with 10 larger eigenvalues
>> A = sprandn(1000,1000,0.01);
>> A(1:20,1:20) = 20*sprandn(20,20,0.2);
>> ev = eig(full(A)); plot(real(ev), imag(ev), 'x');
>> [Q, H] = arnoldi(A, randn(size(A,1),1), 15);
>> rv = eig(H(1:end-1,1:end));
>> plot(real(ev),imag(ev),'x', real(rv),imag(rv),'o');
```

The convergence improves with the dimension n of the Krylov space.

Convergence of Arnoldi

Bad explanation The eigenvalues of H_n are eigenvalues of a 'nearby matrix' obtained by taking H_m (full process) and replacing $H_{n+1,n}$ with zero.

$$H_m = \left[\begin{array}{cccc|cccc} * & \cdots & * & * & * & \cdots & \cdots & * \\ * & \cdots & * & * & * & \cdots & \cdots & * \\ 0 & \ddots & * & * & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & * & * & * & \cdots & \cdots & * \\ \hline & & & 0 & * & \cdots & * & * \\ & & & & * & \cdots & * & * \\ & & & & 0 & \ddots & * & * \\ & & & & 0 & 0 & * & * \end{array}\right].$$

Convergence of Arnoldi

Better explanation The space $K_n(A, b) = \text{span}(b, Ab, \ldots, A^{n-1}b)$ contains the right 'features' to represent the eigenvectors of A with largest eigenvalues: if $A = V\Lambda V^{-1}$ is diagonalizable, then

$$\begin{aligned} A^k b &= (V\Lambda V^{-1})\ldots(V\Lambda V^{-1})b = V\Lambda^k V^{-1} b \\ &= v_1\lambda_1^k c_1 + v_2\lambda_2^k c_2 + \cdots + v_m\lambda_m^k c_m \end{aligned}$$

where $c = V^{-1}b$.
$A^k b$ is a linear combination of the eigenvectors v_i in which those with largest $|\lambda_i|$ are 'more represented'.

Exception: v_i won't appear if the corresponding c_i is very small or zero. Note that the c_i's are the coefficients in the linear combination $b = v_1 c_1 + \cdots + v_m c_m$.

Matlab's eigs

`eigs(A, n)` or `[V, D] = eigs(A, n)` (**eig** **s**parse) computes approximations of the top-n (largest in modulus) eigenvalues.

A can be either a matrix or a function (syntax: `@fname` or `@(x) A*x`).

Uses Arnoldi + some tricks (may use a Krylov space of size larger than n and 'restart' the computation if there are too many iterations).

Convergence not always fast; can return much worse results than `eig`. There is no free lunch: to work on large matrices we have to make approximations / tradeoffs.

Variants

What if I want all eigenvalues?

1. Are you sure you want them all?
2. Convert A to a full matrix and use normal `eig`.

What if I want eigenvalues that are not the largest, e.g., the smallest, or the leftmost, or the closest to 1?

Theorem

Let $(\lambda_1, v_1), \ldots, (\lambda_k, v_k)$ be the eigenvalues/vectors of A.

- $(\lambda_i + \alpha, v_i)$ are eigenvalues/vectors of $A + \alpha I$.
- $(\frac{1}{\lambda_i}, v_i)$ are eigenvalues/vectors of A^{-1}.
- (λ_i^k, v_i) are eigenvalues/vectors of A^k.

Shifted/inverted Arnoldi

Let $(\lambda_1, v_1), \ldots, (\lambda_k, v_k)$ be the eigenvalues/vectors of A. Then, $B = (A - \mu I)^{-1}$ has eigenvalues $\frac{1}{\lambda_i - \mu}$.
The largest eigenvalues of B are those for which $|\lambda_i - \mu|$ is smallest, i.e., the closest ones to μ.

Run Arnoldi on $B = (A - \mu I)^{-1}$ to get the eigenvalues of A closest to a given $\mu \in \mathbb{C}$.

You don't need to compute $B = (A - \mu I)^{-1}$ (and you don't *want* to: it's a full matrix).

Instead, compute a factorization of $A - \mu I$, and then use a function to solve linear systems with it.

Example

```
>> A = bucky();
>> B = inv(A-mu*eye(size(A)));
>> size(A), nnz(A), nnz(B)
ans =
    60 60
ans =
   180
ans =
        3600
% computes 5 eigs of B (bad idea: B is full)
>> fl = eigs(B, 5);
% recover eigs of A
>> lambdas = 1./fl + mu;
```

Example

```
% better idea
>> [L,U] = lu(A - mu*speye(size(A)));
>> f = @(v) U \ (L \ v); % lambda function
>> fl = eigs(f, length(A), 5);
```

Matlab (and scipy) do it automatically for you.

```
% computes 5 eigenvalues closest to mu=2
>> fl = eigs(A, 5, mu);
>> fl = eigs(A, 5, 'SM'); %smallest magnitude
```

Exercises

- Show that if there is breakdown at step n in Arnoldi, i.e., $A^n b \in K_n(A, b)$, then $A^{n+1} b \in K_n(A, b)$, too. Hint: use the relation $A^{n+1} b = A(A^n b)$, and write everything as linear combinations.
- Expand the previous argument (using induction, for instance) to show that $A^k b \in K_n(A, b)$ for all $k \geq n$.
- Take a matrix A, and $b = \sum_{i=1}^{10} v_i$, where v_i are the eigenvectors (in some order). What happens in Arnoldi? At which step do we have breakdown? Why?
- Run Arnoldi on

$$\begin{bmatrix} 0 & & & & 1 \\ 1 & 0 & & & \\ & 1 & \ddots & & \\ & & \ddots & & \\ & & & 1 & 0 \end{bmatrix}$$

(on a computer, by hand, or both). What are the Ritz values at each step? Do they approximate well the eigenvalues?