EECS 275 Matrix Computation

Ming-Hsuan Yang

Electrical Engineering and Computer Science
University of California at Merced
Merced, CA 95344
http://faculty.ucmerced.edu/mhyang

Lecture 18

Overview

- Overview of iterative methods
- Arnoldi algorithm
- Krylov subspace

Reading

- Chapter 32-34 of *Numerical Linear Algebra* by Llyod Trefethen and David Bau
- Chapter 9-10 of *Matrix Computations* by Gene Golub and Charles Van Loan

Direct and iterative methods

- Direct methods:
  - solve the problem by a finite sequence of operations,
  - and in the absence of rounding errors, would deliver an exact solution (like solving a linear system of equation $A\mathbf{x} = \mathbf{b}$ by Gaussian elimination)
  - operate directly on elements of a matrix
  - for general matrices require $O(m^3)$
- Iterative methods:
  - solve a problem by finding successive approximations to the solution starting from an initial guess
  - usually the only choice for nonlinear equations
  - often useful even for linear problems involving a large number of variables where direct methods would be prohibitively expensive
  - exploit sparsity structure that operate in $O(m^2)$

Matrix computation

- Thumbnail history of matrix computation for "very large" dense direct matrix computation

1950:	$m = 20$	Wilkinson
1960:	$m = 200$	Forsythe and Moler
1980:	$m = 2000$	LINPACK
1995:	$m = 20000$	LAPACK
2010:	$m =?$	?

- Matrix dimensions: increase by a factor of 10^3
- Computer hardware: increase by a factor of 10^9 (FLOPS)
- Roughly the $O(m^3)$ bottleneck of direct matrix algorithms
- If matrix problems could be solved in $O(m^2)$ instead, some of the matrices might be 10 to 100 times larger

Structure, sparsity, and black boxes

- For example, a finite difference discretization of a partial differential equation may lead to matrix of dimension $m = 10^5$ with only $\nu = 10$ non-zero entries per row
- Iterative methods exploit sparsity structure
- Iterative methods use a matrix in the form of a black box

- The iterative algorithm requires nothing more than the ability to determine $A\mathbf{x}$ for any $\mathbf{x}$
- For sparse matrix A, easy to design a procedure to compute $A\mathbf{x}$ in only $O(\nu m)$ rather than $O(m^2)$ operations
- Marked contrast to direct methods such as Gaussian or Householder triangularization (which explicitly manipulate matrix entries to introduce zeros, but may destroy sparsity structure)

Projection into Krylov subspaces

- The iterative methods are based on the idea of projecting an m-dimensional problem into a lower-dimensional Krylov subspace
- Given a matrix A and a vector $\mathbf{b}$, the associated Krylov sequence is the set of vector $\mathbf{b}$, $A\mathbf{b}$, $A(A\mathbf{b})$, $A(A(A\mathbf{b}))$, ...
- The corresponding Krylov subspaces are the spaces spanned by successively larger groups of these vectors
- The algorithms can be categorized as

	$A\mathbf{x} = \mathbf{b}$	$A\mathbf{x} = \lambda\mathbf{x}$
$A = A^*$	Conjugate gradients	Lanczos
$A \neq A^*$	GMRES, CGN, BCG, et al.	Arnoldi

- The result of projection into the Krylov subspaces is that the original matrix problem is reduced to a sequence of matrix problems of dimension $n = 1, 2, 3, \ldots$
- When A is Hermitian, the reduced matrices are tridiagonal, otherwise they have Hessenberg form

Number of steps, work per step, and preconditioning

- Gaussian elimination, QR factorization, and most other algorithms of dense linear algebra: there are $O(m)$ steps, each requiring $O(m^2)$ work, for a total work estimate of $O(m^3)$
- For iterative methods, the same figures still apply, but now they represent a typical worst-case behavior
- When iterative methods succeed, they may reduce one or both factors
- The ideal iterative method reduces the number of steps from m to $O(1)$ and the work per step from $O(m^2)$ to $O(m)$, reducing the total work from $O(m^3)$ to $O(m)$
- A more typical improvement is from $O(m^3)$ to $O(m^2)$
- In a practical large-scale engineering computation of the mid-1990s (e.g., $m = 20,000$), they beat direct algorithms by a factor on the order of 10

Exact vs. approximate solutions

- Iterative methods are approximate in the sense that in principle they do not deliver exact answers
- Even direct methods are inexact when carried out on a computer, i.e., up to machine precision
- Under favorable circumstances, iterative methods converge geometrically until the residual is on the oder of machine precision, $\varepsilon_{machine}$
- The direct method makes no progress at all until $O(m^3)$ operations are computed, at which point the residual is again on the oder of $\varepsilon_{machine}$
- Note that there are direct methods that beat $O(m^3)$, however they do not scale well

The Arnoldi iteration

- Most iterative methods are built upon Arnoldi process
- Gram-Schmidt style iteration for transforming a matrix into Hessenberg form
- Recall for QR factorization, we can use
  - Householder reflections (batch algorithm)
  - Gram-Schmidt orthogonalization (anytime algorithm)
- Recall we use similarity transforms to reduce a matrix into Hessenberg form, $A = QHQ^*$, and we can use
  - Householder reflections (batch algorithm)
  - Arnoldi method (anytime algorithm)

	$A = QR$	$A = QHQ^*$
orthogonal structuring	Householder	Householder
structured orthogonalization	Gram-Schmidt	Arnoldi

- Consider a $m \times m$ real or complex matrix A and $m > n$ and $\|\cdot\| = \|\cdot\|_2$

Mechanics of the Arnoldi iteration

- A complete reduction of A to Hessenberg form by an orthogonal similarity transformation can be written as $A = QHQ^*$ or $AQ = QH$
- For iterative methods, we take the view that m is huger of infinite (so computing the full reduction is not feasible)
- Instead, consider the first n columns of $AQ = QH$ and let Q_n be the $m \times n$ matrix whose columns are the first columns of Q

$$Q_n = \left[\begin{array}{c|c|c|c} \mathbf{q}_1 & \mathbf{q}_2 & \cdots & \mathbf{q}_n \end{array}\right]$$

- Let $\widetilde{H}_n$ be the $(n+1) \times n$ upper left section of H, which is also a Hessenberg matrix

$$\widetilde{H}_n = \begin{bmatrix} h_{11} & & \cdots & h_{1n} \\ h_{21} & h_{22} & & \\ & \ddots & \ddots & \vdots \\ & & h_{n,n-1} & h_{nn} \\ & & & h_{n+1,n} \end{bmatrix}$$

Mechanics of the Arnoldi iteration (cont'd)

- We have

$$AQ_n = Q_{n+1}\widetilde{H}_n$$

$$A\left[\mathbf{q}_1\middle|\cdots\middle|\mathbf{q}_n\right] = \left[\mathbf{q}_1\middle|\cdots\middle|\mathbf{q}_{n+1}\right]\begin{bmatrix} h_{11} & & \cdots & h_{1n} \\ h_{21} & h_{22} & & \\ & \ddots & \ddots & \vdots \\ & & h_{n,n-1} & h_{nn} \\ & & & h_{n+1,n} \end{bmatrix}$$

$$A\mathbf{q}_1 = h_{11}\mathbf{q}_1 + h_{21}\mathbf{q}_2$$

$$A\mathbf{q}_2 = h_{12}\mathbf{q}_1 + h_{22}\mathbf{q}_2 + h_{32}\mathbf{q}_3$$

- The n-th column of this equation can be written as

$$A\mathbf{q}_n = h_{1n}\mathbf{q}_1 + \cdots + h_{nn}\mathbf{q}_n + h_{n+1,n}\mathbf{q}_{n+1}$$

- That is, $\mathbf{q}_{n+1}$ satisfies an $(n+1)$-term recurrence relation involving itself and the previous Krylov vectors
- The Arnoldi iteration is simply the modified Gram-Schmidt iteration that implements the above equation

Arnoldi iteration

- Arnoldi iteration:

  Initialize $\mathbf{b}$ as a random vector, $\mathbf{q}_1 = \frac{\mathbf{b}}{\|\mathbf{b}\|}$
  for $n = 1, 2, 3, \ldots$ **do**
    $\mathbf{v} = A\mathbf{q}_n$
    for $j = 1$ to n **do**
      $h_{jn} = \mathbf{q}_j^* \mathbf{v}$
      $\mathbf{v} = \mathbf{v} - h_{jn}\mathbf{q}_j$
    end for
    $h_{n+1,n} = \|\mathbf{v}\|$
    $\mathbf{q}_{n+1} = \mathbf{v}/h_{n+1,n}$
  end for

- Can be implemented in a few lines using MATLAB
- The matrix A appears only in the product of $A\mathbf{q}_n$ which can be computed efficiently (e.g., as a black box procedure)

QR factorization of a Krylov matrix

- The power of the Arnoldi process lies in its interpretations

$$A\mathbf{q}_n = h_{1n}\mathbf{q}_1 + \cdots + h_{nn}\mathbf{q}_n + h_{n+1,n}\mathbf{q}_{n+1}$$

- The vectors $\{\mathbf{q}_i\}$ form bases of the successive Krylov subspaces generated by A and $\mathbf{b}$

$$\mathcal{K}_n = \langle \mathbf{b}, A\mathbf{b}, \ldots, A^{n-1}\mathbf{b} \rangle = \langle \mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n \rangle \subseteq \mathbb{C}^m$$

- Since the vectors $\mathbf{q}_j$ are orthonormal, these are orthonormal bases
- The Arnoldi process can be described as the systematic construction of orthonormal bases for successive Krylov subspaces
- Define K_n as the $m \times n$ Krylov matrix

$$K_n = \left[\mathbf{b} \middle| A\mathbf{b} \middle| \cdots \middle| A^{n-1}\mathbf{b}\right] \tag{1}$$

- Then K_n must have a reduced QR factorization

$$K_n = Q_n R_n \tag{2}$$

where Q_n is the same matrix as before

QR factorization of a Krylov matrix (cont'd)

- In the Arnoldi process, neither K_n nor R_n is formed explicitly
- Working with an explicit approach would make for an unstable algorithm, since these are exceedingly ill-conditioned matrices in general, as the columns of K_n all tend to approximate the same dominant eigenvector of A
- Clearly K_n might be expected to contain good information about the eigenvalues of A with largest modulus
- The QR factorization might be expected to reveal the information by peeling off one approximate eigenvector after another, starting with dominant one

	direct	iterative
straightforward but unstable	simultaneous iteration	(1)-(2)
subtle but stable	QR algorithm	Arnoldi

Projection onto Krylov subspaces

- Another way to view the Arnoldi process is as a computation of projections onto successive Krylov subspaces
- Note that the product $Q_n^* Q_{n+1}$ is the $n \times (n+1)$ matrix with 1 on the main diagonal and 0 elsewhere
- Thus $Q_n^* Q_{n+1} \widetilde{H_n}$ is the $n \times n$ Hessenberg matrix obtained by removing the last row of $\widetilde{H_n}$

$$H_n = \begin{bmatrix} h_{11} & & \cdots & h_{1n} \\ h_{21} & h_{22} & & \\ & \ddots & \ddots & \vdots \\ & & h_{n,n-1} & h_{nn} \end{bmatrix}$$

and with $AQ_n = Q_{n+1}\widetilde{H_n}$, we have

$$H_n = Q_n^* A Q_n$$

- The matrix H_n can be interpreted as the representation in the basis $\{\mathbf{q}_q, \ldots, \mathbf{q}_n\}$ of the orthogonal projection of A onto $\mathcal{K}_n$

Projection onto Krylov subspaces (cont'd)

- Consider the linear operator $\mathcal{K}_n \to \mathcal{K}_n$ defined as follows: given $\mathbf{v} \in \mathcal{K}_n$, apply A to it, then orthogonally project $A\mathbf{v}$ back into the space $\mathcal{K}_n$
- Since the orthogonal projector of $\mathbb{C}^M$ onto $\mathcal{K}_n$ is $Q_n Q_n^*$, this operator can be written $Q_n Q_n^* A$ with respect to the standard basis of $\mathbb{C}^m$
- With respect to the basis of columns of Q_n, it can therefore be written $Q_n^* A Q_n$
- Used frequently in applied and numerical mathematics
- Known as Rayleigh-Ritz procedure in another context
- Not coincidentally, in the diagonal elements of H_n one recognizes the Rayleigh quotients of A with respect to the vectors $\mathbf{q}_j$
- Also one of the ideas underlying finite element methods for solution of partial differential equations, and spectral methods

Projection onto Krylov subspaces (cont'd)

- Since H_n is a projection of A, one might imagine that its eigenvalues would be related to those of A in a useful fashion
- These n numbers
$$\{\theta_j\} = \{\text{eigenvalues of } H_n\}$$
are called the Arnoldi eigenvalue estimates (at step n) or Ritz values (with respect to $\mathcal{K}_n$ of A)
- Some of these numbers may be extraordinarily accurate approximations to some of the eigenvalues of A, even for $n \ll m$

Theorem

The matrices Q_n generated by the Arnoldi iteration are reduced QR factors of the Krylov matrix
$$K_n = Q_n R_n$$
The Hessenberg matrices H_n are the corresponding projections
$$H_n = Q_n^* A Q_n$$
and the successive iterates are related by the formula
$$AQ_n = Q_{n+1} \widetilde{H}_n$$

Computing eigenvalues by the Arnoldi iteration

- The Arnoldi iteration has two roles
  - the basis of many of the iterative algorithms of numerical linear algebra
  - find eigenvalues of non-Hermitian matrices
- At each step n, or at occasional steps, the eigenvalues of the Hessenberg matrix H_n are computed by standard methods such as the QR algorithm
- These are the Arnoldi estimates or Ritz values
- Since $n \ll m$ for feasible computation, one cannot expect to compute all the eigenvalues of A by this process
- Typically, it finds extreme eigenvalues, i.e., eigenvalues near the edge of the spectrum of A
- Physical significance of the eigenvalues of non-Hermitian matrices is sometimes not as great as supposed
- If a matrix is far from normal, i.e., if its eigenvectors are far from orthogonal, implies that its eigenvalues are ill-conditioned
- Then the eigenvalues may have little to do with how a physical system governed by the matrix actually behaves

Arnoldi and polynomial approximation

- Let $\mathbf{x}$ be a vector in the Krylov subspace $\mathcal{K}_n$ which can be written as a linear combination of powers of A times $\mathbf{b}$

$$\mathbf{x} = c_0\mathbf{b} + c_1 A\mathbf{b} + c_2 A^2\mathbf{b} + \cdots + c_{n-1}A^{n-1}\mathbf{b}$$

  i.e., $\mathbf{x}$ is a polynomial in A times $\mathbf{b}$
- That is, if q is the polynomial $q(\mathbf{z}) = c_0 + c_1\mathbf{z} + \cdots + c_{n-1}\mathbf{z}^{n-1}$, then we have

$$\mathbf{x} = q(A)\mathbf{b}$$

- Krylov subspace iterations can always be analyzed in terms of matrix polynomials
- Define

$$P^n = \{\text{monic polynomials of degrees } n\}$$

  (Note superscript n here does not indicate power)
- Arnoldi/Lanczos approximation problem
  Find $p^n \in P^n$ such that

$$\|p^n(A)\mathbf{b}\| = \text{minimum}$$

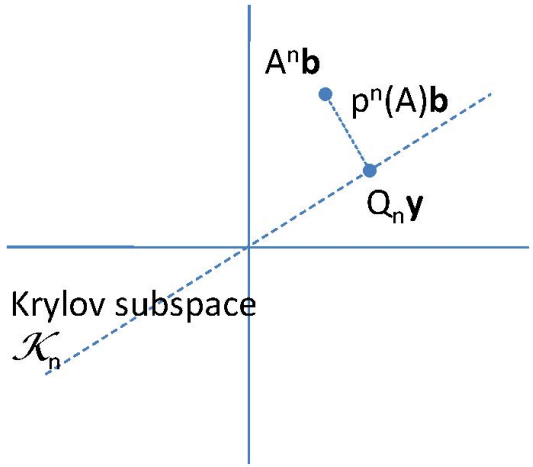
Arnoldi and polynomial approximation (cont'd)

Theorem

As long as the Arnoldi iteration does not break down (i.e., $\mathcal{K}_n$ is of full rank n), it has a unique solution p^n, namely, the characteristic polynomial of H_n

- First note that if $p \in P^n$, then the vector $p(A)\mathbf{b}$ can be written $p(A)\mathbf{b} = A^n\mathbf{b} - Q_n\mathbf{y}$ for some $\mathbf{y} \in \mathbb{C}^n$ where Q_n is defined as before (Q_n is the orthogonal matrix in similarity transform)
- Equivalent to a linear least squares problem: find the point in the $\mathcal{K}_n$ closest to $A^n\mathbf{b}$, or in the matrix terms, find $\mathbf{y}$ such that $\|A^n\mathbf{b} - Q_n\mathbf{y}\|$ is minimal
- The solution is characterized by the orthogonality condition $p^n(A)\mathbf{b} \perp \mathcal{K}_n$, or equivalently $Q_n^* p^n(A)\mathbf{b} = 0$

Arnoldi and polynomial approximation (cont'd)

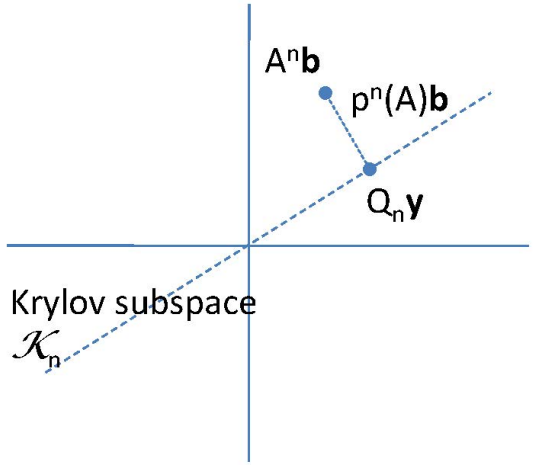


- Now consider the factorization $A = QHQ^*$ as discussed before
- At step n of the Arnoldi process, we have computed the first n columns of Q and H, and thus

$$Q = \begin{bmatrix} Q_n & U \end{bmatrix}, \quad H = \begin{bmatrix} H_n & X_1 \\ X_2 & X_3 \end{bmatrix}$$

for some $m \times (m-n)$ matrix U with orthonormal columns that are also orthogonal to the columns Q_n and some matrices X_1, X_2, and X_3 of dimensions $n \times (m-n)$, $(m-n) \times n$, and $(m-n) \times (m-n)$, respectively with all but the upper right entry of X_2 equal to 0

Arnoldi and polynomial approximation (cont'd)



- The orthogonality condition becomes $Q_n^* Q p^n(H) Q^* \mathbf{b} = 0$, which amounts to the condition that the first n entries of the first column of $p^n(H)$ are zero (as only the first entry of $Q^*\mathbf{b}$ is nonzero)
- Because of the structure of H, these are also the first n entries of the first column of $p^n(H_n)$
- By the Cayley-Hamilton theorem, that these are zero if p^n is the characteristic polynomial of H_n
- Conversely, suppose there were another polynomial $p^n(A)\mathbf{b} \perp \mathcal{K}_n$
- Taking the difference would give a nonzero polynomial q of degree $n-1$ with $q(A)\mathbf{b} = 0$, violating the assumption that $\mathcal{K}_n$ is of full rank

Arnoldi and polynomial approximation (cont'd)

- The goal of the Arnoldi iteration is to solve a polynomial approximation problem, or equivalently a least squares problem involving a Krylov subspace
- If the Arnoldi iteration tends to find eigenvalues, it must be a by-product of achieving this goal
- Suppose that A is diagonalizable and has only $n \ll m$ distinct eigenvalues, hence a minimal polynomial of degree n
- After n steps, all of these eigenvalues will be found exactly at least if the vector $\mathbf{b}$ contains components in directions associated with every eigenvalue
- Thus, after n steps, the Arnoldi iteration has computed the minimal polynomial of A exactly
- In practical applications, the agreement of Ritz values with eigenvalues is approximate instead of exact, and instead of minimal polynomial, the result is a pseudo minimal, i.e., a polynomial p^n such that $\|p^n(A)\|$ is small