AMSC 600 /CMSC 760 Advanced Linear Numerical Analysis

Fall 2007

Arnoldi Methods (cont.)

Dianne P. O'Leary

Algorithms that use the Arnoldi Basis

Reference: Chapter 6 of Saad

So far: Arnoldi for general matrices.

Now: Arnoldi for symmetric matrices.

- Generating the Arnoldi Basis: The Lanczos algorithm
- CGNR, CGNE, and SYMMLQ
- FOM when A is symmetric positive definite = CG
- GMRES when A is symmetric = CR
- The Lanczos - CG relation
- Practicalities
- Convergence Results for CG

And: A few notes

- Applications to Linear Operators
- The Faber-Manteuffel Theorem
- Block Iterative Methods
- Arnoldi methods use the "wrong" subspace

Generating the Arnoldi Basis: The Lanczos algorithm

When $\mathbf{A}$ is symmetric, we have seen that the relation

$$\mathbf{AV} = \mathbf{VH}$$

with $\mathbf{H}$ upper Hessenberg and $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ implies that $\mathbf{H}$ is tridiagonal.

In this case, the Arnoldi iteration is usually called the (symmetric) Lanczos iteration.

The complexity of the iteration is greatly reduced, since we only need to orthogonalize against 1 old vector at iteration k, instead of $k-1$ vectors.

The storage is also reduced.

Orthogonal polynomials

Sec 6.6.2

If $\mathbf{AV} = \mathbf{TV}$ where $\mathbf{T}$ is tridiagonal, then

$$\mathbf{A}\mathbf{v}_j = t_{j-1,j}\mathbf{v}_{j-1} + t_{jj}\mathbf{v}_j + t_{j+1,j}\mathbf{v}_{j+1},$$

or

$$t_{j+1,j}\mathbf{v}_{j+1} = \mathbf{A}\mathbf{v}_j - t_{jj}\mathbf{v}_j - t_{j-1,j}\mathbf{v}_{j-1}.$$

We have seen a recurrence like this before, the Chebyshev semi-iterative method:

$$\bar{\mathbf{x}}_{j+1} = w_{j+1}(\mathbf{A}\bar{\mathbf{x}}_j + \mathbf{b} - \bar{\mathbf{x}}_{j-1}) + \bar{\mathbf{x}}_{j-1}$$

where w_{j+1} involved values of Chebyshev polynomials.

The Lanczos sequence also defines a sequence of polynomials, and, like the Chebyshev polynomials, they are orthogonal in a particular inner product.

The (rather bizarre) inner product is

$$(f, g) = \int_a^b f(x)g(x)\, dw(x)$$

where

- $0 < a < \lambda_1$, $\lambda_n < b$.
- The distribution function w is constant except at λ_i.
- The jump in w at λ_i is proportional to $\mathbf{b}^T\mathbf{u}_i$, where $\mathbf{u}_i$ is the eigenvector corresponding to the eigenvalue λ_i.
- This is a Lebesgue integral rather than the more usual Riemann integral.

This has allowed researchers to apply the formidable machinery of orthogonal polynomials and Gaussian quadrature, with very fruitful results, most of which are beyond the scope of the course.

CGNR and CGNE

We can convert any problem $\mathbf{Ax} = \mathbf{b}$ with a nonsingular nonsymmetric matrix to a symmetric positive definite one in either of two ways:

- $\mathbf{A}^T\mathbf{Ax} = \mathbf{A}^T\mathbf{b}$. This is called the normal equations, and the Arnoldi method that minimizes the norm of the residual $\mathbf{b} - \mathbf{Ax}$ is called CGNR (Sec 8.3.1).
- $\mathbf{AA}^T\mathbf{u} = \mathbf{b}$, where $\mathbf{x} = \mathbf{A}^T\mathbf{u}$, and the Arnoldi method that minimizes the norm of the error $\mathbf{x}_k - \mathbf{x}_{true}$ is called CGNE or Craig's method (Sec 8.3.2).

Advantage:

- We now have a short recurrence for the $\mathbf{v}$ vectors because the matrix is symmetric.

Disadvantages:

- Convergence depends on the eigenvalues of $\mathbf{A}^T\mathbf{A}$, which are equal to those of $\mathbf{A}\mathbf{A}^T$, and these can be much more spread out than those of $\mathbf{A}$.
- Each iteration involves multiplication by both $\mathbf{A}$ and $\mathbf{A}^T$, twice the work of the Arnoldi iteration.

SYMMLQ for symmetric indefinite problems

If we use the Arnoldi iteration on a symmetric matrix, we obtain

$$\mathbf{A}\mathbf{V}_m = \mathbf{V}_{m+1}\bar{\mathbf{T}}_m,$$

where $\bar{\mathbf{T}}_m$ is $(m+1) \times m$.

Case 1: (GMRES-like) If we decide to minimize the residual, we need to compute $\mathbf{x}_m = \mathbf{V}_m\mathbf{y}_m$ using

$$\begin{aligned} \|\mathbf{r}_m\|^2 &= \|\mathbf{b} - \mathbf{A}\mathbf{x}_m\|^2 \\ &= \|\mathbf{b} - \mathbf{V}_{m+1}\bar{\mathbf{T}}_m\mathbf{y}_m\|^2 \\ &= \|\beta\mathbf{e}_1 - \bar{\mathbf{T}}_m\mathbf{y}_m\|^2. \end{aligned}$$

Case 2: (FOM-like) If we decide to make the residual orthogonal to the Krylov subspace then we use

$$\mathbf{V}_m^T\mathbf{r}_m = \beta\mathbf{e}_1 - \mathbf{T}_m\mathbf{y}_m = \mathbf{0}.$$

We need to factor $\mathbf{T}_m$.

One stable way to solve either of these two problems, regardless of whether $\mathbf{A}$ is positive definite or not, is to use an $\mathbf{LQ}$ factorization, where $\mathbf{L}$ is lower triangular and $\mathbf{Q}$ is orthogonal.

For Case 2, this leads to the very useful algorithm SYMMLQ (Paige and Saunders).

For Case 1, this leads to the MINRES algorithm of Paige and Saunders, based on solving the normal equations.

$$\bar{\mathbf{T}}_m^T\bar{\mathbf{T}}_m\mathbf{y}_m = \beta\bar{\mathbf{T}}_m^T\mathbf{e}_1$$

Whenever you have a symmetric matrix and don't know whether or not it is positive definite, I advise using one of these two algorithms, available from a website at Stanford University.

FOM when A is symmetric positive definite = CG

If $\mathbf{A}$ is not positive definite, then for stability we need to factor $\mathbf{T}_m$ using LQ, QR, or LU with pivoting,

If $\mathbf{A}$ is positive definite, then so is $\mathbf{T}_m$, so we can use the more inexpensive Cholesky decomposition.

More commonly, though, instead of the three-term recurrence, we use the algorithm given in the KMP notes (p. 8), one involving a recurrence for $\mathbf{p}_m$ and $\mathbf{r}_m$. The residuals are still mutually orthogonal, but the $\mathbf{p}$-vectors are $\mathbf{A}$-conjugate, so that if they form the columns of a matrix $\mathbf{P}$, then

$$\mathbf{PAP} = \mathbf{I}.$$

This is what enables us to express $\mathbf{x}_m$ as

$$\mathbf{x}_m = \mathbf{P}\boldsymbol{\alpha},$$

where $\alpha_m = (\mathbf{r}_m^T \mathbf{r}_m)/(\mathbf{p}_m^T \mathbf{A}\mathbf{p}_m)$.

GMRES when A is symmetric = CR

Sec 6.8

The GMRES algorithm, specialized to $\mathbf{A}$ symmetric, gives an algorithm called conjugate residual, in which it can be shown that

$$\mathbf{r}_i^T \mathbf{A}\mathbf{r}_j = 0$$

for $i \neq j$.

The Lanczos - CG relation

The parameters α and β computed by the $\mathbf{p}, \mathbf{r}$ form of cg can be used to compute $\mathbf{T}_m$. See p. 194 for the formulas.

The residual $\mathbf{r}_k$, normalized to length 1, is the vector $\mathbf{v}_k$.

Practicalities

There are many mathematically equivalent but computationally different forms of these algorithms.

I recommend:

- $\mathbf{A}$ symmetric, positive definite: CG in its $\mathbf{p}, \mathbf{r}$ form or MINRES.
- $\mathbf{A}$ symmetric indefinite: SYMMLQ or MINRES.

Convergence Results for CG

See pp. 9-10 of the KMP notes, or Sec 6.11.3.

A Few Notes about Arnoldi Methods

Applications to Linear Operators

We have stated all of these algorithms for matrix problems:

$\mathbf{A}$ is an $n \times n$ matrix.

We could have stated them for linear operators $\mathbf{A}$ which map some domain space into itself.

This can be an important conceptual tool:

- Given an infinite dimensional problem (e.g., an integral equation), we can apply an Arnoldi method to it.
- If we actually want to compute with it, we can use a discretization in order to approximate $\mathbf{A}\mathbf{v}_k$.
- Alternatively, given an infinite dimensional problem, we can discretize it to a finite dimensional one and then apply an Arnoldi method to it,

---

The Faber-Manteuffel Theorem

We see that:

- For general matrices, $\mathbf{v}_k$ in the Arnoldi iteration depends on all $k-1$ previous vectors $\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}$.
- For symmetric matrices, $\mathbf{v}_k$ depends on only two previous vectors.

So (Section 6.10), the question arose, for what class of matrices is the recurrence "short"?

The answer was given by Faber and Manteuffel in 1984: There is an s-term recurrence for every choice of $\mathbf{b}$ iff

- The Arnoldi iteration always breaks down at or before iteration s, so that $\mathcal{K}_s(\mathbf{A}, \mathbf{b}) = \mathcal{K}_{s+1}(\mathbf{A}, \mathbf{b})$; or
- The matrix $\mathbf{A}$ is normal and there is a polynomial p of degree $s-1$ for which $\mathbf{A}^H = p(\mathbf{A})$.

(A matrix $\mathbf{A}$ is normal if $\mathbf{A}^H\mathbf{A} = \mathbf{A}\mathbf{A}^H$.
Examples: symmetric matrices, Hermitian matrices, orthogonal matrices.)

Saad doesn't give the proof, since the Faber-Manteuffel one is messy, but there is a clean version in a 2006 preprint by Faber, Liesen, and Tichy.

Block Iterative Methods

Suppose we want to solve

$$\mathbf{A}\mathbf{X} = \mathbf{B}$$

where $\mathbf{X}$ and $\mathbf{B}$ are $n \times p$, $p > 1$.

For definiteness, suppose we want to use GMRES.

- We could run GMRES on the first rhs, then repeat $p-1$ times. But this wastes a lot of effort.
- We could make better use of our Krylov subspaces this way:
  - We could run GMRES on the first rhs, and update the $p-1$ other problems by making their residuals orthogonal to the resulting subspace.
  - If the solution to the 2nd system is not good enough, we could apply GMRES to it, updating the $p-2$ other problems.
  - etc.
- We could look for an Arnoldi method that treats all of the rhs vectors equally.

The result of the 3rd option is the Block Arnoldi methods: block-GMRES, block-CG, etc. (Sec 6.12)

At the kth step, they work with the subspace spanned by the columns $\mathbf{B}, \mathbf{A}\mathbf{B}, \ldots, \mathbf{A}^{k-1}\mathbf{B}$, abbreviated as $\mathcal{K}_k(\mathbf{A}, \mathbf{B})$. This subspace has dimension at most kp.

Mechanics

The algebra is derived from $\mathbf{A}\mathbf{V} = \mathbf{V}\mathbf{H}$, but now we specify the first p columns of $\mathbf{V}$ to be an orthonormal basis for the space spanned by the columns of $\mathbf{B}$.

We move forward p columns at a time:

- Compute $\mathbf{A}$ times the p previous columns,
- Then use modified Gram-Schmidt to make these new vectors orthogonal to all of the old ones, The resulting coefficients are placed in $\mathbf{H}$ in the blocks on and above the main diagonal.
- Then make the resulting vectors orthogonal to each other, usually by using the Householder QR factorization. The Q-matrix becomes the next block of $\mathbf{V}$, and the R-matrix becomes the block of $\mathbf{H}$ below the main diagonal $(j+1, j)$.

The resulting matrix $\mathbf{H}$ has p diagonals below the main diagonal.

Breakdown

For the Arnoldi method, breakdown ($h_{j+1,j} = 0$) means that we have solved the problem.

For the block-Arnoldi method, breakdown ($H_{j+1,j}$ rank-deficient) means that our Krylov subspace has dimension less than kp.

Diagnosing and handling this condition, by reducing the size of the block used from now on, is messy.

Two advantages of the block iterations

Advantage 1:

Arnoldi: To get a subspace of dimension k requires k matrix vector products, which means k accesses of the matrix $\mathbf{A}$.

block Arnoldi: To get a subspace of dimension k requires k matrix vector products, which means k/p accesses of the matrix $\mathbf{A}$.

Thus block Arnoldi has a big advantage when the main cost of matrix-vector product is due to accessing the matrix (e.g., $\mathbf{A}$ is on secondary storage, or spread across multiple processors, etc.) In fact, current machines do arithmetic much faster than accessing memory, so this is a widespread advantage.

Advantage 2:

Arnoldi: Convergence depends on how well a polynomial of degree k behaves at the n eigenvalues of $\mathbf{A}$.

block Arnoldi: Convergence depends on how well a polynomial of degree kp behaves at the n eigenvalues of $\mathbf{A}$.

We can derive error bounds that ignore $p-1$ of the eigenvalues, so this is a powerful property.

Therefore, sometimes it is advantageous to use block iterations by making up additional right-hand sides, even if you only want to solve one problem!

Arnoldi methods use the "wrong" subspace.

Recall what you know about systems of equations:

If $\mathbf{A} : \mathcal{C} \rightarrow \mathcal{D}$, and we want to solve $\mathbf{Ax} = \mathbf{b}$, then

- $\mathbf{b} \in \mathcal{D}$,
- $\mathbf{x} \in \mathcal{C}$.

Example 1: If $\mathbf{A}$ is $m \times n$ with $m \neq n$, then the linear system may still have a solution, but we can't approximate it in a subspace built upon $\mathbf{b}$.

Example 2: If $\mathbf{A}$ is $n \times n$ but rank deficient, then the solution obtained from a Krylov subspace approximation will never have a component in the nullspace of $\mathbf{A}$ (assuming exact arithmetic). This may be good or bad, depending on the application.

The bottom line is that Arnoldi methods search in a Krylov subspace that is a subset of the range space of $\mathbf{A}$, when the solution really lives in the domain.

This leads us to a different set of Krylov methods, based on Lanczos bi-diagonalization.