WI4014TU – Numerical Analysis for PDE's

Lecture 15
Krylov subspace, spectral polynomials, Arnoldi algorithm, polynomial approximation problem, GMRES algorithm, preconditioning

Krylov Subspace Methods

A very simple iteration

Simple iteration has the form: $\boldsymbol{x}_{k+1} = H\boldsymbol{x}_k + \boldsymbol{v}$.
One of the possible iteration matrices is $H = I - A$. In this case $\boldsymbol{v} = (I - H)A^{-1}\boldsymbol{b} = \boldsymbol{b}$ and

$$\boldsymbol{x}_{k+1} = (I - A)\boldsymbol{x}_k + \boldsymbol{b}$$

As we know, this scheme will converge iff $\rho(H) = \rho(I - A) < 1$, which means $\rho(A) < 1$, and it is very seldom the case.

We want a robust iterative algorithm that works with any matrix A

Simple iteration for residuals

Simple iterations can also be written as:

$$A\boldsymbol{x}_{k+1} = A(I - A)\boldsymbol{x}_x + A\boldsymbol{b}$$
$$\boldsymbol{b} - A\boldsymbol{x}_{k+1} = \boldsymbol{b} - A(I - A)\boldsymbol{x}_k - A\boldsymbol{b} = (I - A)(\boldsymbol{b} - A\boldsymbol{x}_k)$$

The *residual* vector $\boldsymbol{r}_k$ that measures how well the system is solved by $\boldsymbol{x}_k$ is defined as

$$\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$$

The goal of an iterative algorithm is to minimize the residual.

The simple iterations lead to the following iterations for $\boldsymbol{r}_k$:

$$\boldsymbol{r}_{k+1} = (I - A)\boldsymbol{r}_k = (I - A)^{k+1}\boldsymbol{r}_0$$

Krylov subspace

The k-dimensional *Krylov subspace* $\mathcal{K}_k(A, \boldsymbol{b})$ of the linear system

$$A\boldsymbol{x} = \boldsymbol{b}$$

is the space spanned by the sequence of Krylov vectors $\{\boldsymbol{b}, A\boldsymbol{b}, A^2\boldsymbol{b}, \ldots, A^{k-1}\boldsymbol{b}\}$:

$$\mathcal{K}_k(A, \boldsymbol{b}) = \text{span}\,\{\boldsymbol{b}, A\boldsymbol{b}, A^2\boldsymbol{b}, \ldots, A^{k-1}\boldsymbol{b}\}$$

Residuals live in Krylov subspace

We see that

$$\boldsymbol{r}_1 = (I - A)\boldsymbol{r}_0 = A^0\boldsymbol{r}_0 - A^1\boldsymbol{r}_0 \in \text{span}\{\boldsymbol{r}_0, A\boldsymbol{r}_0\} = \mathcal{K}_2(A, \boldsymbol{r}_0)$$

Hence,

$$\boldsymbol{r}_k \in \mathcal{K}_{k+1}(A, \boldsymbol{r}_0)$$

Spectral polynomials of simple residual iterations

Let there be $A = V\Lambda V^{-1}$ with eigenvalues $\{\lambda_m\}$ and eigenvectors $\{\boldsymbol{v}_m\}$. Then, the simple iteration for $\boldsymbol{r}_k$ can also be written as

$$\boldsymbol{r}_k = (I - A)^k \boldsymbol{r}_0 = V(I - \Lambda)^k V^{-1} \boldsymbol{r}_0 = \sum_{m=1}^{n} \alpha_m (1 - \lambda_m)^k \boldsymbol{v}_m$$

The iterations converge, i.e., $\boldsymbol{r}_k \to \boldsymbol{0}$ as $k \to \infty$, only if $|1 - \lambda_m| < 1$, for all $m = 1, \ldots, n$, in other words, if $\rho(I - A) < 1$.

Expression for $\boldsymbol{r}_k$ contains the *polynomial* $(1 - \lambda_m)^k$ in the eigenvalues of A (spectral polynomial). However, this particular polynomial gives convergence only if $\rho(A) < 1$.

Is it possible to find a spectral polynomial that gives convergence for any matrix A?

Spectral polynomials of general iterations

A general iteration with some polynomial $P_k(z) = c_0 + c_1 z + \cdots + c_k z^k$ can be expressed as

$$\boldsymbol{r}_k = P_k(A)\boldsymbol{r}_0 = \sum_{m=1}^{n} \alpha_m P_k(\lambda_m)\boldsymbol{v}_m$$

How to chose this polynomial $P_k(z)$? Since we want to *minimize* the residual $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k$, such a polynomial should be *minimal on the spectrum of A*, i.e., when λ_m are substituted for z.

The polynomial $(1 - z)^k$ of the simple iteration is small on the spectrum of A, only if $|\lambda_m| < 1$.

Hessenberg decomposition of matrices

For any matrix $A \in \mathbb{C}^{n \times n}$ there exists the Hessenberg decomposition:

$$A = QHQ^H$$

where $Q \in \mathbb{C}^{n \times n}$ is a unitary matrix $Q^H Q = I$, $Q^H = \overline{Q}^T$, and $H \in \mathbb{C}^{n \times n}$ is an upper *Hessenberg* matrix – an almost upper-triangular matrix with zeros below the first sub-diagonal.

$$H = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix}$$

Arnoldi algorithm

There is an efficient *Arnoldi* algorithm that allows to build $Q_k \in \mathbb{C}^{n\times k}$ and $H_k \in \mathbb{C}^{k\times k}$ iteratively using the modified Gram-Schmidt process starting with arbitrary vector $\boldsymbol{b}$:

$$
\begin{aligned}
&\boldsymbol{q}_1 = \frac{1}{\|\boldsymbol{b}\|}\boldsymbol{b} \\
&\text{for } k = 1, 2, \cdots : \\
&\qquad \boldsymbol{v} = A\boldsymbol{q}_k \\
&\qquad \text{for } j = 1 \text{ to } k : \\
&\qquad\qquad h_{j,k} = \boldsymbol{q}_j^H \boldsymbol{v} \\
&\qquad\qquad \boldsymbol{v} = \boldsymbol{v} - h_{j,k}\boldsymbol{q}_j \\
&\qquad h_{k+1,k} = \|\boldsymbol{v}\| \\
&\qquad \boldsymbol{q}_{k+1} = \frac{1}{h_{k+1,k}}\boldsymbol{v}
\end{aligned}
$$

Partial Hessenberg decomposition

A full Hessenberg decomposition yields $AQ = QH$. The Arnoldi iterations produce the partial decompositions

$$AQ_k = Q_{k+1}\tilde{H}_k,$$

where $\tilde{H}_k \in \mathbb{C}^{(k+1)\times k}$ is H_k with an extra row at the bottom containing all zeros except for the last element $h_{k+1,k}$, e.g.

$$\tilde{H}_4 = \begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \\ 0 & 0 & 0 & \bullet \end{bmatrix}$$

Arnoldi algorithm and Krylov subspace

- If Arnoldi algorithm is started with $\boldsymbol{b}$ from $A\boldsymbol{x} = \boldsymbol{b}$, then the vectors $\{\boldsymbol{q}_j,\ j = 1, \ldots, k\}$ form an orthonormal basis for the Krylov subspace $\mathcal{K}_k(A, \boldsymbol{b})$. In fact, the relation between the Krylov matrix $K_k = [\boldsymbol{b}\ A\boldsymbol{b}\ \cdots\ A^{k-1}\boldsymbol{b}]$ and the Arnoldi matrix $Q_k = [\boldsymbol{q}_1\ \boldsymbol{q}_2\ \cdots\ \boldsymbol{q}_k]$ is the (reduced) QR factorization:

$$K_k = Q_k R_k$$

  where R_k is upper-triangular. However, neither K_k nor R_k are formed explicitly by the Arnoldi algorithm, since it would be numerically unstable.
- The matrix $H_k = Q_k^H A Q_k$ is the *orthogonal projection* of A onto $\mathcal{K}_k(A, \boldsymbol{b})$. The eigenvalues of H_k are called the *Ritz values* of A with respect to $\mathcal{K}_k(A, \boldsymbol{b})$. As $k \to n$ the Ritz values tend to the eigenvalues of A

Iterative algorithm as a polynomial

We are looking for an iterative algorithm that would find an approximation $\boldsymbol{x}_k$ to the solution of $A\boldsymbol{x} = \boldsymbol{b}$ over the Krylov subspace $\mathcal{K}_k(A, \boldsymbol{b})$:

$$\boldsymbol{x}_k = c_0\boldsymbol{b} + c_1 A\boldsymbol{b} + c_2 A^2\boldsymbol{b} + \cdots + c_{k-1}A^{k-1}\boldsymbol{b}$$
$$\boldsymbol{x}_k = P_{k-1}(A)\boldsymbol{b}$$

where $P_{k-1}(A) = c_0 I + c_1 A + \cdots + c_{k-1}A^{k-1}$. To define the iterative algorithm we need to find the coefficients of the polynomial $\{c_j \mid j = 0, 1, \ldots, k-1\}$.

Since $\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k = \boldsymbol{b} - AP_{k-1}(A)\boldsymbol{b} = (I - AP_{k-1}(A))\boldsymbol{b} = \tilde{P}_k(A)\boldsymbol{b}$, we need to find $P_{k-1}(A)$ such that $\tilde{P}_k(A)\boldsymbol{b}$ is minimized.

Monic and characteristic polynomials

A *monic* polynomial has the form:

$$P_k(z) = c_0 + c_1 z + \cdots + c_{k-1} z^{k-1} + z^k,$$

i.e., $c_k = 1$.

The *characteristic* polynomial of the matrix A is:

$$P_A(z) = \det(zI - A), \quad z \in \mathbb{C}$$

It is a monic polynomial.

Cayley-Hamilton Theorem

$$P_A(A) = 0 \quad \text{(zero matrix)}$$

Polynomial approximation problem and Arnoldi

Polynomial approximation problem: Let $\mathcal{P}_k$ be the space of all monic polynomials of degree $\leq k$. We want to find $P_k \in \mathcal{P}_k$ such that

$$\|P_k(A)\boldsymbol{b}\| = \text{ minimum}$$

Theorem: *As long as the Arnoldi iteration does not break down (Krylov matrix K_k is of full column rank), the polynomial approximation problem has a unique solution P_k, namely, the characteristic polynomial of H_k*

Proof (1):

$$\begin{aligned} P_k(A)\boldsymbol{b} &= \left(c_0 I + c_1 A + \cdots + c_{k-1}A^{k-1} + A^k\right)\boldsymbol{b} \\ &= A^k\boldsymbol{b} + \left(c_0 I + c_1 A + \cdots + c_{k-1}A^{k-1}\right)\boldsymbol{b} \end{aligned}$$

Since

$$\begin{aligned} &\left(c_0 I + c_1 A + \cdots + c_{k-1}A^{k-1}\right)\boldsymbol{b} \\ &= c_0\boldsymbol{b} + c_1 A\boldsymbol{b} + \cdots + c_{k-1}A^{k-1}\boldsymbol{b} \in \text{span}\{\boldsymbol{b}, A\boldsymbol{b}, \ldots, A^{k-1}\boldsymbol{b}\} = \mathcal{K}_k(A, \boldsymbol{b}) \end{aligned}$$

and the vectors $\{\boldsymbol{q}_1, \boldsymbol{q}_2, \ldots, \boldsymbol{q}_k\}$ are the basis for the Krylov subspace $\mathcal{K}_k(A, \boldsymbol{b})$, we can express $P_k(A)\boldsymbol{b}$ as

$$P_k(A)\boldsymbol{b} = A^k\boldsymbol{b} - Q_k\boldsymbol{y}$$

where $\boldsymbol{y}$ is a vector of unknown coefficients.

Proof (2): Hence, we arrive at the equivalent minimization problem:

$$\min_{\mathcal{P}_k} \|P_k(A)\boldsymbol{b}\| \quad \Leftrightarrow \quad \min_{\boldsymbol{y}} \|A^k\boldsymbol{b} - Q_k\boldsymbol{y}\|$$

i.e., the polynomial approximation problem is equivalent to a least-squares (LS) minimization problem and we need to find a point in $\mathcal{K}^k(A, \boldsymbol{b})$ closest to $A^k\boldsymbol{b}$.

The least-squares residual $P_k(A)\boldsymbol{b}$ should then satisfy the orthogonality condition $P_k(A)\boldsymbol{b} \perp \mathcal{K}^k(A, \boldsymbol{b})$, or equivalently, one can require that the residual is orthogonal to every basis vector of $\mathcal{K}^k(A, \boldsymbol{b})$. Since these basis vectors are the columns of Q_k, we require:

$$Q_k^H P_k(A)\boldsymbol{b} = \boldsymbol{0}$$

Proof (3): The matrices of the exact Hessenberg factorization $A = Q_n H_n Q_n^H$ can be represented in block form as

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

where Q_k and H_k have been generated by the Arnoldi algorithm, $U \in \mathbb{C}^{n \times (n-k)}$ has orthonormal columns that are also orthogonal to the columns of Q_k, and $X_1 \in \mathbb{C}^{k \times (n-k)}$, $X_2 \in \mathbb{C}^{(n-k) \times k}$, $X_3 \in \mathbb{C}^{(n-k) \times (n-k)}$ are some matrices. Note that all but the upper-right entry of X_2 are equal to zero.

The orthogonality condition can be written as

$$Q_k^H P_k(A)\boldsymbol{b} = Q_k^H P_k(Q_n H_n Q_n^H)\boldsymbol{b} = Q_k^H Q_n P_k(H_n) Q_n^H \boldsymbol{b} = \mathbf{0}$$

By construction all Arnoldi vectors, except for the first one, are orthogonal to $\boldsymbol{b}$. Hence, only the first entry of the vector $Q_n^H \boldsymbol{b}$ is nonzero. Therefore, the orthogonality condition demands that the first k entries of the first column of $P_k(H_n)$ are zero. Due to the upper Hessenberg structure of H_n, these are also the upper k entries of the first column of $P_k(H_k)$.

Proof (4): By the *Cayley-Hamilton* Theorem these entries will be zero if $P_k(H_k)$ is the characteristic polynomial of H_k.

Conversely, suppose there is another polynomial $\tilde{P}_k \in \mathcal{P}_k$ with the property $\tilde{P}_k(A)\boldsymbol{b} \perp \mathcal{K}_k(A, \boldsymbol{b})$. As both P_k and $\tilde{P}_k$ are monial, their difference $R_{k-1} = P_k - \tilde{P}_k$ would be a nonzero polynomial of degree $k-1$:

$$\begin{aligned} R_{k-1}(A) &= c_0 I + \cdots + c_{k-1}A^{k-1} + A^k - (\tilde{c}_0 I + \cdots + \tilde{c}_{k-1}A^{k-1} + A^k) \\ &= (c_0 - \tilde{c}_0)I + \cdots + (c_{k-1} - \tilde{c}_{k-1})A^{k-1} \end{aligned}$$

Moreover, $R_{k-1}(A)\boldsymbol{b} = \boldsymbol{0}$, since $R_{k-1}(A)\boldsymbol{b} \in \mathcal{K}_k(A, \boldsymbol{b})$ and $R_{k-1}(A)\boldsymbol{b} \perp \mathcal{K}_k(A, \boldsymbol{b})$. However, then the Krylov matrix K_k would not have a full column rank. Indeed,

$$R_{k-1}(A)\boldsymbol{b} = (c_0 - \tilde{c}_0)\boldsymbol{b} + \cdots + (c_{k-1} - \tilde{c}_{k-1})A^{k-1}\boldsymbol{b} = K_k(\boldsymbol{c} - \tilde{\boldsymbol{c}}) = \boldsymbol{0}, \quad \boldsymbol{c} - \tilde{\boldsymbol{c}} \neq \boldsymbol{0},$$

which contradicts the assumptions of the Theorem. $\square$

Minimizing residuals over Krylov subspace

Our goal is to solve the system $A\boldsymbol{x} = \boldsymbol{b}$ with nonsingular A. The idea of the Generalized Minimum Residual (GMRES) algorithm is to find $\boldsymbol{x}_k$ that solves the LS minimization problem over the Krylov subspace:

$$\min_{\boldsymbol{x}_k \in \mathcal{K}_k(A,\boldsymbol{b})} \|\boldsymbol{b} - A\boldsymbol{x}_k\|$$

Since any $\boldsymbol{x}_k \in \mathcal{K}_k(A, \boldsymbol{b})$ can be represented as $\boldsymbol{x}_k = Q_k \boldsymbol{y}$, where Q_k has been generated by the Arnoldi algorithm and $AQ_k = Q_{k+1}\tilde{H}_k$, we can also pose this problem as

$$\begin{aligned}
&\min_{\boldsymbol{y}} \|\boldsymbol{b} - AQ_k\boldsymbol{y}\|, \quad \boldsymbol{x}_k = Q_k\boldsymbol{y} \\
&\min_{\boldsymbol{y}} \|\boldsymbol{b} - Q_{k+1}\tilde{H}_k\boldsymbol{y}\|, \quad \boldsymbol{x}_k = Q_k\boldsymbol{y} \\
&\min_{\boldsymbol{y}} \|Q_{k+1}^H\boldsymbol{b} - \tilde{H}_k\boldsymbol{y}\|, \quad \boldsymbol{x}_k = Q_k\boldsymbol{y}
\end{aligned}$$

where the last step is permitted since $\boldsymbol{b} \in \mathrm{Col}(Q_{k+1})$

GMRES algorithm

Finally, we recall that by construction $Q_{k+1}^H \boldsymbol{b} = \|\boldsymbol{b}\| \boldsymbol{e}_1$, where $\boldsymbol{e}_1 = (1, 0, 0, \ldots, 0)^T$ and arrive at the minimization problem at the heart of the GMRES algorithm

$$\min_{\boldsymbol{y}} \left\| \|\boldsymbol{b}\| \boldsymbol{e}_1 - \tilde{H}_k \boldsymbol{y} \right\|, \quad \boldsymbol{x}_k = Q_k \boldsymbol{y}$$

GMRES:

$$\boldsymbol{q}_1 = \frac{1}{\|\boldsymbol{b}\|} \boldsymbol{b}$$

for $k = 1, 2, \cdots$:

step k of Arnoldi algorithm

find $\boldsymbol{y}$ to minimize $\left\| \|\boldsymbol{b}\| \boldsymbol{e}_1 - \tilde{H}_k \boldsymbol{y} \right\|$

$\boldsymbol{x}_k = Q_k \boldsymbol{y}$

The inner $(k+1) \times k$ LS problem can be solved by the QR factorization of $\tilde{H}_k$ that can be obtained by updating the QR of $\tilde{H}_{k-1}$ in $\mathcal{O}(k)$ operations

GMRES and polynomial approximation problem

Let $\tilde{\mathcal{P}}_k$ be the space of polynomials $\tilde{P}_k(z)$ of degree $\leq k$ with the property $\tilde{P}_k(0) = 1$. In terms of coefficients we have now $c_0 = 1$ instead of $c_n = 1$.

GMRES iterations produce

$$\boldsymbol{x}_k = P_{k-1}(A)\boldsymbol{b}$$

where the coefficients of $P_{k-1}(A)$ are the entries of the vector $\boldsymbol{c}$, which minimizes $\|\boldsymbol{b} - AK_k\boldsymbol{c}\|$, with $K_k\boldsymbol{c} = Q_kR_k\boldsymbol{c} = Q_k\boldsymbol{y}$.

The corresponding residual is

$$\boldsymbol{r}_k = \boldsymbol{b} - A\boldsymbol{x}_k = \boldsymbol{b} - AP_{k-1}(A)\boldsymbol{b} = (I - AP_{k-1}(A))\boldsymbol{b} = \tilde{P}_k(A)\boldsymbol{b}$$

where $\tilde{P}_k(z) = 1 - zP_{k-1}(z) \in \tilde{\mathcal{P}}_k$, since $\tilde{P}_k(0) = 1$. Hence, GMRES solves the following polynomial approximation problem:

$$\min_{\tilde{P}_k \in \tilde{\mathcal{P}}_k} \|\tilde{P}_k(A)\boldsymbol{b}\|$$

Convergence of GMRES

GMRES will always find the exact solution in n steps. Can it find a good approximation at a step $k < n$? Since $\|\boldsymbol{r}_k\| = \left\|\tilde{P}_k(A)\boldsymbol{b}\right\| \leq \left\|\tilde{P}_k(A)\right\| \|\boldsymbol{b}\|$, GMRES residuals satisfy

$$\frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{b}\|} \leq \inf_{\tilde{P}_k \in \tilde{\mathcal{P}}_k} \left\|\tilde{P}_k(A)\right\|$$

Theorem: *Let $A = V\Lambda V^{-1}$ be diagonalizable. At step k of the GMRES algorithm, the residual $\boldsymbol{r}_k$ satisfies*

$$\frac{\|\boldsymbol{r}_k\|}{\|\boldsymbol{b}\|} \leq \kappa(V) \inf_{\tilde{P}_k \in \tilde{\mathcal{P}}_k} \|\tilde{P}_k(\Lambda)\|$$

where $\kappa(V) = \|V\| \, \|V^{-1}\|$ is the condition number of V.

In words: if A is not too far from normal, and if properly normalized degree k polynomials can be found that are small on the spectrum of A, then GMRES converges quickly (e.g. clustered spectra removed from zero - why?)

Other notable Krylov subspace methods

GMRES has the fastest convergence of all Krylov subspace methods, but it may take a lot of memory if it does not converge fast enough, since it requires to store $Q_k \in \mathbb{C}^{n \times k}$. Some alternatives:

- Conjugate Gradient (CG) method: works with symmetric (Hermitian) matrices, short recurrence relation
- Lanczos method: works with symmetric (Hermitian) matrices, based on Arnoldi, three-term recurrence relation, tri-diagonal H_k
- Stabilized Bi-Conjugate Gradient (Bi-CGSTAB): works with any matrix, short recurrence relation, norm of the residual noes not reduce monotonically
- Restarted GMRES(m): saves space by restarting Arnoldi iterations after m steps, stores only $Q_m \in \mathbb{C}^{n \times m}$, convergence is slower though

Preconditioning

Instead of solving $A\boldsymbol{x} = \boldsymbol{b}$ we could also solve

$$M^{-1}A\boldsymbol{x} = M^{-1}\boldsymbol{b}$$

where M is any nonsingular matrix. If M^{-1} is not available, this may require solving systems of the form $M\boldsymbol{y} = \boldsymbol{c}$ at each step of an iterative algorithm.

Matrix M is called *precoditioner*. If well chosen, it may significantly accelerate convergence of an iterative algorithm

Common preconditioners

- *Diagonal scaling or Jacobi*: $M = \text{diag}(A)$ or $M = \text{diag}(\boldsymbol{c})$
- *ILU*: $M = \tilde{L}\tilde{U}$, where $\tilde{L}$ and $\tilde{U}$ are triangular and sparse (M^{-1} is easy to compute), and approximate the exact LU factors of A
- *Coarse-grid approximation*: $M = \langle \text{to fine grid} \rangle \circ \tilde{A} \circ \langle \text{to coarse grid} \rangle$, $\tilde{A}$ is a coarse-grid discretization of the operator, small matrix
- *Geometric and algebraic multigrid*: iterated application of coarse- and fine-grid discretizations or judicious 'grouping' of matrix elements
- *Block preconditioners, domain decomposition, low-order discretization, constant coefficient and symmetric approximations, matrix splitting, single step of classical iterations, convolution approximation (FFT), polynomial preconditioners, deflation, shifted Laplacian*