NUMERICAL LINEAR ALGEBRA

Claudio Tessa

October 7, 2025

Contents

1 Introduction **2**
1.1 Basic matrix decomposition . . . 2
1.1.1 LU factorization with (partial) pivoting . . . 2
1.1.2 Cholesky decomposition . . . 2
1.1.3 QR decomposition . . . 2
1.2 Sparse matrices . . . 3
1.2.1 Coordinate format (COO) . . . 3
1.2.2 Coordinate Compressed Sparse Row format (CSR) . . . 3

2 Iterative methods for large and sparse linear systems **4**
2.1 The Jacobi method . . . 4
2.2 The Gauss-Seidel method . . . 5
2.3 Linear iterative methods . . . 5
2.3.1 necessary and sufficient condition for convergence . . . 6
2.3.2 Convergence of Jacobi (J) and Gauss-Seidel (GS) methods . . . 6
2.4 Stopping criteria . . . 7
2.5 The stationary Richardson method . . . 7
2.5.1 Convergence of the stationary Richardson method . . . 8
2.6 Preconditioning techniques . . . 8
2.7 The gradient method . . . 9
2.8 The conjugate gradient method . . . 9
2.9 Krylov-space methods . . . 9
2.10 Solving nonsymmetric linear systems iteratively with Krylov space solvers 10
2.10.1 The biconjugate gradiend (BiCG) method . . . 10
2.10.2 the BiCGSTAB method . . . 11
2.10.3 The GMRES method . . . 11

3 Solving large-scale eigenvalue problems **11**
3.1 Similarity transformations . . . 11

1 Introduction

1.1 Basic matrix decomposition

1.1.1 LU factorization with (partial) pivoting

If $A \in \mathbb{R}^{n \times n}$ is nonsingular (invertible), then

$$PA = LU$$

- P is a permutation matrix
- L is unit lower matrix
- U is upper triangular

Linear system solution:

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

1. Factor $PA = LU$ (expensive, $O(n^3)$)
2. Solve $L\mathbf{y} = P\mathbf{b}$ (lower triangular system, $O(n^2)$)
3. Solve $U\mathbf{x} = \mathbf{y}$ (upper triangular system, $O(n^2)$)

1.1.2 Cholesky decomposition

If $A \in \mathbb{R}^{n \times n}$ is symmetric ($A^T = A$) and positive definite ($\mathbf{z}^T A\mathbf{z} > 0$ for all $\mathbf{z} \neq \mathbf{0}$), then

$$A = L^T L$$

where L is lower triangular (with positive entries on the diagonal).

Linear system solution

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

1. Factor $A = L^T L$ (expensive, $O(N^3)$)
2. Solve $L^T\mathbf{y} = \mathbf{b}$ (lower triangular system, $O(n^2)$)
3. Solve $L\mathbf{x} = \mathbf{y}$ (upper triangular system, $O(n^2)$)

1.1.3 QR decomposition

If $A \in \mathbb{R}^{n \times n}$ is nonsingular (invertible), then

$$A = QR$$

- Q is an orthogonal
- R is upper triangular

Linear system solution:

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

1. Factor $A = QR$ (expensive, $O(n^3)$)

2. Multiply $\mathbf{c} = Q^T\mathbf{b}$ ($O(n^2)$)
3. Solve $R\mathbf{x} = \mathbf{c}$ (lower triangular system, $O(n^2)$)

1.2 Sparse matrices

A sparse matrix is a matrix in which most elements are zero. Roughly speaking, given $A \in \mathbb{R}^{n\times n}$, the number of non-zero entries of A, called nnz(A), is $O(n)$.

Many matrices arising from real applications are sparse. If we need to store A, we can exploit the sparse structure using different **storage schemes**:

Name	Easy insertion	Fast $A\mathbf{x}$
Coordinate (COO)	Yes	No
CSR	No	Yes

1.2.1 Coordinate format (COO)

The data structure consists of three arrays of length nnz(A):

- `AA`: all the values of the nonzero elements of A *in any order*
- `JR`: integer array containing their row indices
- `JC`: integer array containing their column indices

Example

$$A = \begin{pmatrix} 1. & 0. & 0. & 2. & 0. \\ 3. & 4. & 0. & 5. & 0. \\ 6. & 0. & 7. & 8. & 9. \\ 0. & 0. & 10. & 11. & 0. \\ 0. & 0. & 0. & 0. & 12. \end{pmatrix}$$

AA	12.	9.	7.	5.	1.	2.	11.	3.	6.	4.	8.	10.
JR	5	3	3	2	1	1	4	2	3	2	3	4
JC	5	5	3	4	1	4	4	1	1	2	4	3

1.2.2 Coordinate Compressed Sparse Row format (CSR)

If the elements of A are listed by row, the array `JC` might be replaced by an array that points to the beginning of each row.

- `AA`: all the values of the nonzero elements of A, *stored row by row* from $1, \ldots, n$
- `JA`: contains the column indices
- `IA`: contains the pointers to the beginning of each row in the arrays `AA` and `JA`. Thus $IA(i)$ contains the position in the arrays `AA` and `JA` where the i-th row starts. The length of `IA` is $n + 1$.

Example

$$A = \begin{pmatrix} 1. & 0. & 0. & 2. & 0. \\ 3. & 4. & 0. & 5. & 0. \\ 6. & 0. & 7. & 8. & 9. \\ 0. & 0. & 10. & 11. & 0. \\ 0. & 0. & 0. & 0. & 12. \end{pmatrix}$$

AA	1.	2.	3.	4.	5.	6.	7.	8.	9.	10.	11.	12.
JA	1	4	1	2	4	1	3	4	5	3	4	5
IA	1	3	6	10	12	13						

2 Iterative methods for large and sparse linear systems

Let's consider the following linear system of equations

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

where:

- $A \in \mathbb{R}^{n \times n}$
- $\mathbf{b} \in \mathbb{R}^n$
- $\mathbf{x} \in \mathbb{R}^n$
- $\det(A) \neq 0$

n general, direct methods (i.e., methods based on a *manipulation* of A) are not suitable whenever n is large, or A is sparse. The average cost of direct methods scales as n^3. Therefore, we use iterative methods.

We introduce a sequence $\mathbf{x}^{(k)}$ of vectors determined by a recursive relation that identifies the method. In order to initialize the iterative process, it is necessary to provide an initial vector $\mathbf{x}^{(0)}$.

For the method to make sense, it must satisfy the **convergence property**

$$\lim_{k \to +\infty} \mathbf{x}^{(k)} = \mathbf{x}$$

Convergence must not depend on the choice of $\mathbf{x}^{(0)}$.

Since the convergence is only guaranteed after an infinite number of iterations, from a practical point of view we will have to stop the iterative process after a finite number of iterations, when we believe we have arrived *sufficiently close* to the solution. For this, we use specific **stopping criteria**.

Note that, even in exact arithmetic, an iterative method will inevitably be affected by a **numerical error**.

2.1 The Jacobi method

Starting from the i-th line of the linear system:

$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad \Longrightarrow \quad a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n = b_i$$

Formally, the solution x_i for each i is given by

$$x_i = \frac{b_i - \sum_{j \neq i} a_{ij} x_j}{a_{ii}}$$

Obviously we cannot use it because we do not know x_j for $j \neq i$. We can introduce an iterative method that updates $x_i^{(k+1)}$ using the others $x_j^{(k)}$ obtained in the previous step k

$$x_i^{(k+1)} = \frac{b_i - \sum_{j \neq i} a_{ij} x_j^{(k)}}{a_{ii}} \qquad \forall i = 1, \ldots, n$$

In general, each iteration costs $\sim n^2$ operations, so the Jacobi method is competitive if the number of iterations is less that n. If A is a sparse matrix, then the cost is only $\sim n$ per iteration.

It should be noted that the solutions $x_i^{(k+1)}$ can be computed fully in parallel, which is very competitive for large scale systems.

2.2 The Gauss-Seidel method

Let's start from Jacobi's method: $x_i^{(k+1)} = \frac{b_i - \sum_{j \neq i} a_{ij} x_j^{(k)}}{a_{ii}}$

At iteration $(k+1)$, let's consider the computation of $x_i^{(k+1)}$. We observe that, for $j < i$ (for $i > 2$), $x_j^{(k+1)}$ is known (we have already calculated it). We can therefore think of using the quantities at step $(k+1)$ if $j < i$ and (as in the Jacobi method) those at the previous step k if $j > i$.

$$x_i^{(k+1)} = \frac{b_i - \sum_{j<i} a_{ij} x_j^{(k+1)} - \sum_{j>i} a_{ij} x_j^{(k)}}{a_{ii}}$$

The computational costs are comparable to those of the Jacobi method. However, unlike the Jacobi, GS is not fully parallelizable.

2.3 Linear iterative methods

n general, we consider linear iterative methods of the following form:

$$\mathbf{x}^{(k+1)} = B\mathbf{x}^{(k)} + \mathbf{f} \qquad k \geq 0$$

where $B \in \mathbb{R}^{n \times n}$, $\mathbf{f} \in \mathbb{R}^n$

B is called iteration matrix. Its choice, together with $\mathbf{f}$, uniquely identify the method. So how to choose B and $\mathbf{f}$?

1. **Consistency**: If $\mathbf{x}^{(k)}$ is the exact solution $\mathbf{x}$, then $\mathbf{x}^{(k+1)}$ is again equal to $\mathbf{x}$

$$\mathbf{x} = B\mathbf{x} + \mathbf{f} \quad \Longrightarrow \quad \mathbf{f} = (I - B)\mathbf{x} = (I - B)A^{-1}\mathbf{b}$$

   The former identity gives a relationship between B and $\mathbf{f}$ as a function of the data.

2. **Convergence**: Let's introduce the error at step $(k+1)$: $\mathbf{e}^{(k+1)} = \mathbf{x} - \mathbf{x}^{(k+1)}$ and a suitable vector norm $\|\cdot\|$ (for example, the Euclidian norm). Then we have:

$$\|\mathbf{e}^{(k+1)}\| = \|\mathbf{x} - \mathbf{x}^{(k+1)}\| = \|\mathbf{x} - B\mathbf{x}^{(k)} - \mathbf{f}\| = \qquad \leftarrow \text{consistency}$$

$$= \|\mathbf{x} - B\mathbf{x}^{(k)} - (I - B)\mathbf{x}\| = \|B\mathbf{e}^{(k)}\|$$

$$\leq \|B\|\|\mathbf{e}^{(k)}\|$$

By recursion we obtain

$$\|\mathbf{e}^{(k-1)}\| \leq \|B\|\ \|B\|\ \|e^{(k+1)}\|$$

$$\leq \|B\|\ \|B\|\ \|B\|\ \|\mathbf{e}^{(k-2)}\| \leq \cdots \leq \|B\|^{k+1}\|\mathbf{e}^{(0)}\|$$

$$\implies \lim_{k\to\infty} \|\mathbf{e}^{(k+1)}\| \leq \left(\lim_{k\to\infty} \|B\|^{k+1}\right) \|\mathbf{e}^{(0)}\|$$

If $\|B\| < 1 \implies \lim_{k\to\infty} \|\mathbf{e}^{(k+1)}\| = 0$. Therefore, $\|B\| < 1$ is a **sufficient condition for convergence**.

2.3.1 necessary and sufficient condition for convergence

We define $\rho(B) = \max_j |\lambda_j(B)|$, where $\lambda_j(B)$ are the eigenvalues of B. $\rho(B)$ is called **spectral radius** of B. Remember that if B is SPD, then $\rho(B) = \|B\|$.

Theorem

A consistent iterative method with iteration matrix B converges **if and only if** $\rho(B) < 1$.

2.3.2 Convergence of Jacobi (J) and Gauss-Seidel (GS) methods

Let

$$A = \begin{bmatrix} \ddots & & -F \\ & D & \\ -E & & \ddots \end{bmatrix}$$

- D: diagonal part of A
- $-E$: lower triangular part of A
- $-F$: upper triangular part of A

The Jacobi method can be rewritten as

$$D\mathbf{x}^{(k+1)} = (E + F)\mathbf{x}^{(k)} + \mathbf{b}$$

its iteration matrix is given by

$$B_J = D^{-1}(E + F) = F^{-1}(D - A) = I - D^{-1}A$$

For the Gauss-Seidel method we have

$$(D - E)\mathbf{x}^{(k+1)} = F\mathbf{x}^{(k)} + \mathbf{b}$$

The iteration matrix is given by

$$B_{GS} = (D - E)^{-1}F$$

Both methods are consistent.

- If A is strictly diagonally dominant by rows/columns, then J and GS converge.
- If A is SPD then the GS method is convergent.
- If A is tridiagonal, it can be shown that $\rho^2(B_J) = \rho(B_{GS})$. Therefore both methods converge or fail to converge at the same time. If they converge, GS is faster than J.

2.4 Stopping criteria

A practical test is needed to determine when to stop the iteration. The idea is to terminate the iterations when $\dfrac{\|\mathbf{x} - \mathbf{x}^k\|}{\|\mathbf{x}^k\|} \leq \varepsilon$, where ε is a user-defined tolerance. Unfortunately, the error is not known. We can use two criterion:

1. **Residual-based stopping criterion**.

$$\frac{\|\mathbf{x} - \mathbf{x}^{(k)}\|}{\|\mathbf{x}\|} \leq K(A)\frac{\|\mathbf{r}^{(k)}\|}{\|\mathbf{b}\|} \quad \Longrightarrow \quad \boxed{\frac{\|\mathbf{r}^k\|}{\|\mathbf{b}\|} \leq \varepsilon}$$

   This is a good stopping criterion whenever $K(A)$ is "small".

2. **Distance between consecutive iterates**. Define the relative residual $\boldsymbol{\delta}^{(k)} = \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$

$$\boxed{\|\boldsymbol{\delta}^{(k)}\| \leq \varepsilon}$$

   It can be shown that the relation between the true error and $\boldsymbol{\delta}^{(k)}$ is

$$\|\mathbf{e}^{(k)}\| \leq \frac{1}{1 - \rho(B)}\|\boldsymbol{\delta}^{(k)}\|$$

   This is a good stopping criterion only if $\rho(B) \ll 1$.

2.5 The stationary Richardson method

Given $\mathbf{x}^{(0)}$, $\alpha \in \mathbb{R}$:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha \underbrace{(\mathbf{b} - A\mathbf{x}^{(k)})}_{\text{residual } \mathbf{r}^{(k)}}$$

The idea is to update the numerical solution by adding a quantity proportional to the residual (e.g., if the residual is large, the solution at step k should be corrected a lot).

The stationary Richardson method is characterized by

$$B_\alpha = I - \alpha A \qquad \mathbf{f} = \alpha\mathbf{b}$$

2.5.1 Convergence of the stationary Richardson method

Let A be symmetric and positive definite matrix (SPD), then the stationary Richardson method is convergent if and only if

$$0 < \alpha < \frac{2}{\lambda_{max}(A)}$$

with the optimal parameter α_{opt}:

$$\alpha_{opt} = \frac{2}{\lambda_{min}(A) + \lambda_{max}(A)}$$

and optimal spectral radius ρ_{opt} (maximum convergence speed):

$$\rho_{opt} = \frac{K(A) - 1}{K(A) + 1}$$

2.6 Preconditioning techniques

The optimal value $\rho_{opt} = \frac{K(A)-1}{K(A)+1}$ expresses the maximum convergence speed that can be attained with the stationary Richardson method.

Badly conditioned matrices ($K(A) \gg 1$) are characterized by a very low convergence rate. To improve the speed of convergence, we introduce a SPD matrix P^{-1} (called *preconditioner*). Then, solving $A\mathbf{x} = \mathbf{b}$ is equivalent to the following preconditioned system:

$$P^{-\frac{1}{2}} A \underbrace{P^{-\frac{1}{2}}\mathbf{z}}_{\mathbf{x}} = P^{-\frac{1}{2}}\mathbf{b}$$

where $\mathbf{x} = P^{-\frac{1}{2}}\mathbf{z}$.

Suppose that P^{-1} has real and positive eigenvalues. We apply the stationary Richardson method to $P^{-1}A$, i.e.,

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha P^{-1}\left(\mathbf{b} - A\mathbf{x}^{(k)}\right) = \mathbf{x}^{(k)} + \alpha P^{-1}\mathbf{r}^{(k)}$$

We obtain the same results as in the non preconditioned case, provided we replace A with $P^{-1}A$:

- Convergence: $\quad 0 < \alpha < \dfrac{2}{\lambda_{max}(P^{-1}A)}$
- Optimal values:

$$\alpha_{opt} = \frac{2}{\lambda_{min}(P^{-1}A) + \lambda_{max}(P^{-1}A)} \qquad \rho_{opt} = \frac{K(P^{-1}A) - 1}{K(P^{-1}A) + 1}$$

Therefore, if $K(P^{-1}A) \ll K(A)$, we obtain a higher convergence rate with respect to the unpreconditioned case.

To have $K(P^{-1}A) \ll K(A)$ we need that the linear system in P must be "easily solvable". To this aim, P should have a special structure (e.g., diagonal, triangular, ...).

2.7 The gradient method

Let $A \in \mathbb{R}^n$ be SPD. In this case, solving the linear system $A\mathbf{x} = \mathbf{b}$ is equivalent to minimizing the quadratic function $\Phi : \mathbb{R}^n \to \mathbb{R}$

$$\Phi(\mathbf{y}) = \frac{1}{2}\mathbf{y}^T A\mathbf{y} - \mathbf{y}^T\mathbf{b}$$

As A is positive definite, the hyperplane given by $\mathbf{z} = \Phi(\mathbf{y})$ defines a paraboloid in $\mathbb{R}^{n+1}$ and global minimum in $\mathbf{x}$. Since $\nabla\Phi(\mathbf{y}) = A\mathbf{y} - \mathbf{b}$ we have that the minimum ($\nabla\Phi(\mathbf{x}) = \mathbf{0}$) coincides with the solution of $A\mathbf{x} - \mathbf{b}$.

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \nabla\Phi(\mathbf{x}^{(k)})$$

where α_k is a parameter. The optimal parameter α_k is

$$\alpha_k = \frac{\left(\mathbf{r}^{(k)}\right)^T \mathbf{r}^{(k)}}{\left(\mathbf{r}^{(k)}\right)^T A\mathbf{r}^{(k)}}$$

Note that $\nabla\Phi(\mathbf{y}) = A\mathbf{y} - \mathbf{b} = -\mathbf{r}$, we obtain

$$-\nabla\Phi(\mathbf{x}^{(k)}) = \mathbf{b} - A\mathbf{x}^{(k)} = \mathbf{r}^{(k)}$$

Therefore, the gradient method can be interpreted as a Richardson method with dynamic parameter α_k.

2.8 The conjugate gradient method

We introduce a new updating direction $\mathbf{d}^{(k+1)}$ in such a way that it is A-conjugate to all the previous directions $\mathbf{d}^{(j)}$, $j \leq k$ (i.e., orthogonal with respect to the scalar product induced by A)

$$\left(\mathbf{d}^{(k+1)}, \mathbf{d}^{(j)}\right)_A = \left(\mathbf{d}^{(k+1)}, A\mathbf{d}^{(j)}\right) = 0 \qquad \forall j \leq k \quad \begin{matrix}\text{conjugate}\\ \text{directions}\end{matrix}$$

In exact arithmetic, the CG method converges to the exact solution in at most n iterations. At each iteration k, the error $\mathbf{e}^{(k)} = \mathbf{x} - \mathbf{x}^{(k)}$ can be bounded by

$$\left\|\mathbf{e}^{(k)}\right\|_A \leq \frac{2c^k}{1 + c^{2k}} \left\|\mathbf{e}^{(0)}\right\|_A \qquad \text{with} \qquad c = \frac{\sqrt{K(A)} - 1}{\sqrt{K(A)} + 1}$$

We can also use preconditioning on the CG method.

2.9 Krylov-space methods

For linear iterative methods (with $P = I,\ \alpha_k = 1\ \forall k$), we have:

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{r}^{(k)} \qquad k \geq 1 \tag{1}$$

The following recursive relation for the residuals holds

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - A\mathbf{r}^{(k)} \qquad k \geq 1$$

From the above identity, it follows by induction that

$$\mathbf{r}^{(k)} = p_{k-1}(A)\mathbf{r}^{(0)} \in \text{span}\ \{\mathbf{r}^{(0)},\ A\mathbf{r}^{(0)},\ \ldots,\ A^{k-1}\mathbf{r}^{(0)}\}$$

where $p_r() = (1-)^r$ is a polynomial of exact degree r.

From (1) we have

$$\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + \mathbf{r}^{(0)} + \cdots + \mathbf{r}^{(k-1)} = \mathbf{x}^{(0)} + p_{k-1}(A)\mathbf{r}^{(0)}$$

so $\mathbf{x}^{(0)}$ lies in the affine space $\mathbf{x}^{(0)} + \text{span}\ \{\mathbf{r}^{(0)},\ A\mathbf{r}^{(0)},\ \ldots,\ A^{k-1}\mathbf{r}^{(0)}\}$ obtained by shifting the subspace of $\mathbf{r}^{k-1}$.

Definition of Krylov (sub)space

Given a nonsingular $A \in \mathbb{R}^{n\times n}$ and $\mathbf{y} \in \mathbb{R}^n,\ \mathbf{y} \neq 0$, the k-th Kyrlov (sub)space $\mathscr{K}_k(A, \mathbf{y})$ generated by A from $\mathbf{y}$ is

$$\mathscr{K}_k(A, \mathbf{y}) := \text{span}\ (\mathbf{y},\ A\mathbf{y},\ \ldots,\ A^{k-1}\mathbf{y})$$

Clearly, it holds

$$\mathscr{K}_1(A, \mathbf{y}) \subseteq \mathcal{K}_2(A, \mathbf{y}) \subseteq \ldots$$

We can choose the k-th approximate solution $\mathbf{x}^{(k)} \in \mathbf{x}^{(0)} + \mathscr{K}_k(A, \mathbf{r}^{(0)})$. In particular

$$\mathbf{x}^{(k)} = \mathbf{x}^{(0)} + p_{k-1}(A)\mathbf{r}^{(0)}$$

with a polynomial $p_{k-1}(A)$ of exact degree $k-1$.

When applied to large real-world problems, Kyrlov space solvers often converge very slowly, if at all. In practice, they are therefore nearly always used with preconditioning:

$$A\mathbf{x} = \mathbf{b} \qquad \Longleftrightarrow \qquad P^{-1}A\mathbb{P}^{-1}\mathbf{x} = P^{-1}\mathbf{b} \qquad \Longleftrightarrow \qquad \hat{A}\hat{\mathbf{z}} = \hat{\mathbf{b}},\quad \hat{P}\hat{\mathbf{z}} = \mathbf{x}$$

Applying a preconditioned Kyrlov space solver means applying the method to $\hat{A}\hat{\mathbf{z}} = \hat{b}$.

2.10 Solving nonsymmetric linear systems iteratively with Krylov space solvers

2.10.1 The biconjugate gradiend (BiCG) method

While CG (for spd A) has mutually orthogonal residuals $\mathbf{r}^{(k)}$ with

$$\mathbf{r}^{(k)} = p_k(A)\mathbf{r}^{(0)} \in \text{span}\{\mathbf{r}^{(0)}, A\mathbf{r}^{(0)}, \ldots, A^k\mathbf{r}^{(0)}\} = \mathscr{K}_{k+1}(A, \mathbf{r}^{(0)})$$

BiCG constructs in the same spaces residuals that are orthogonal to a dual Krylov space spanned by *shadow residuals*

$$\tilde{\mathbf{r}}^{(k)} = p_k(A)\tilde{\mathbf{r}}^{(0)} \in \text{span}\{\tilde{\mathbf{r}}^{(0)}, A^T\tilde{\mathbf{r}}^{(0)}, \ldots, (A^T)^k\tilde{\mathbf{r}}^{(0)}\} = \mathscr{K}_{k+1}(A^T, \tilde{\mathbf{r}}^{(0)}) \equiv \tilde{\mathscr{K}}_{k+1}$$

The initial shadow residual $\tilde{\mathbf{r}}(0)$ can be chosen freely. So, BiCG requires two matrix-vector multiplications to extend $\mathscr{K}_k$ and $\tilde{\mathscr{K}}_k$: one multiplication by A and one by A^T.

2.10.2 the BiCGSTAB method

The biconjugate gradient stabilized method (BiCGSTAB) is a variant of the biconjugate gradient method (BiCG) and has faster and smoother convergence than the original BiCG.

It is unnecessary to explicitly keep track of the residuals and search directions of BiCG. In other words, the BiCG iterations can be performed implicitly. Unlike the original BiCG method, it doed not require multiplication by A^T.

2.10.3 The GMRES method

The Generalized Minimum Residual method (GMRES) is a projection method. The method approximates the solution by the vector in a Krylov subspace with minimal residual. The arnoldi iteration is used to find this vector.

GMRES approximates the exact solution of $A\mathbf{x} = \mathbf{b}$ by the vector $\mathbf{x}^{(k)} \in \mathbf{x}^{(0)} + \mathscr{K}_k(A, \mathbf{r}^{(0)})$ that minimizes the Euclidian norm of the residual $\mathbf{r}^{(k)} = \mathbf{b} - A\mathbf{x}^{(k)}$

3 Solving large-scale eigenvalue problems

Eigenvalue problem

Given a matrix $A \in \mathbb{C}^{n\times n}$, find $(\lambda, \mathbf{v}) \in \mathbb{C} \times \mathbb{C}^n \setminus \{\mathbf{0}\}$ such that

$$A\mathbf{v} = \lambda\mathbf{v}$$

where λ is an eigenvalue of A, and $\mathbf{v}$ (non-zero) is the corresponding eigenvector.

The set of all the eigenvalues of a matrix A is calle the **spectrum** of A.

The maximum modulus of all the eigenvalues is called the **spectral radius** of A: $\rho(A) = \max\{|\lambda| : \lambda \in \lambda(A)\}$.

The problem $A\mathbf{v} = \lambda\mathbf{v}$ is equivalent to $(A - \lambda I)\mathbf{v} = 0$. $\det(A - \lambda I) = 0$ is a polynomial of degree n in λ: it is called the **characteristic polynomial** of A and its roots are the eigenvalues of A.

3.1 Similarity transformations

We first need to identify what types of transformations preserve eigenvalues, and for what types of matrices the eigenvalues are easily determined.

Definition

The matrix B is similar to the matrix A if there exists a nonsingulat matrix T such taht $B = T^{-1}AT$.

With the above definition, it is trivial to show that

$$B\mathbf{y} = \lambda\mathbf{y} \implies T^{-1}AT\mathbf{y} = \lambda\mathbf{y} \implies A(T\mathbf{y}) = \lambda(T\mathbf{y})$$

so that A and B have the same eigenvalues, and if $\mathbf{y}$ is an eigenvector of B, then $\mathbf{v} = T\mathbf{y}$ is an eigenvector of A.