# Lecture 21

# The Jacobi Method

## Learning Outcomes

- **Understanding the Jacobi Method:**
  - Acquire knowledge of the Jacobi method for solving linear systems, including the decomposition of matrix $A$ into its diagonal, lower, and upper triangular parts, and the iterative formula for finding the solution.
- **Analyzing Convergence Criteria:**
  - Develop the ability to analyze the convergence of the Jacobi method, understanding the conditions under which the method converges, such as the strong row and column sum criteria and the weak row sum criterion for irreducible matrices.
- **Applying Graph Theory to Matrices:**
  - Learn to apply graph theory concepts to study the properties of matrices, understanding the notion of irreducibility and connectivity in the context of the oriented graph of a matrix, and utilizing these concepts to assess the convergence of the Jacobi method.

Let us split our matrix $A \in \mathbb{C}^{n\times n}$ in the form $A = D + L + U$ where $D = \mathrm{diag}(a_{1,1}, \dots, a_{n,n})$ is the diagonal part and $L$ and $U$ are the lower and upper triangular parts given by

$$l_{i,j} := \begin{cases} a_{i,j} & \text{if } i > j, \\ 0 & \text{else,} \end{cases} \qquad u_{i,j} := \begin{cases} a_{i,j} & \text{if } i < j, \\ 0 & \text{else.} \end{cases}$$

The Jacobi method is the linear iterative method that consists in choosing $M = D$ and $N = L + U$, whence

$$x^{(k)} = D^{-1}\big(b - (L+U)x^{(k-1)}\big).$$

**Example:** For

$$A = \begin{pmatrix} 2 & -1 & 0 & \cdots & 0 \\ -1 & 2 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 2 \end{pmatrix}$$

we have

$$D = \begin{pmatrix} 2 & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & 2 \end{pmatrix}, L = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ -1 & \ddots & & & \vdots \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & -1 & 0 \end{pmatrix}, U = \begin{pmatrix} 0 & -1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ \vdots & & & \ddots & -1 \\ 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}$$

so that

$$x^{(k)} = \begin{pmatrix} \frac{1}{2} & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \frac{1}{2} \end{pmatrix} \left( b - \begin{pmatrix} 0 & -1 & 0 & \cdots & 0 \\ -1 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & -1 \\ 0 & \cdots & 0 & -1 & 0 \end{pmatrix} x^{(k-1)} \right).$$

Writing $x^{(k)} = (x_1^{(k)}, \ldots, x_n^{(k)})^T$, this means that

$$\begin{aligned} x_i^{(k)} &= \frac{1}{2}(b_i + x_{i-1}^{(k-1)} + x_{i+1}^{(k-1)}), \quad 2 \le i \le n-1, \\ x_1^{(k)} &= \frac{1}{2}(b_1 + x_2^{(k-1)}), \\ x_n^{(k)} &= \frac{1}{2}(b_n + x_{n-1}^{(k-1)}). \end{aligned}$$

For convergence it is sufficient if the spectral radius of the iteration matrix $R = -M^{-1}N = -D^{-1}(L+U)$ is smaller than one.

**Theorem 21.1.** *The Jacobi method is convergent if $A$ satisfies*

*(1) $|a_{i,i}| > \sum_{j \neq i} |a_{i,j}|$ for all $i$ (strong row sum criterion),* **or**

*(2) $|a_{j,j}| > \sum_{i \neq j} |a_{i,j}|$ for all $j$ (strong column sum criterion).*

*Proof.* In the first case we have that the entries of $R$ satisfy,

$$r_{i,j} = \begin{cases} -\frac{a_{i,j}}{a_{i,i}}, & \text{if } j \neq i \\ 0, & \text{otherwise} \end{cases}$$

for all $i$. Using the definition of induced by infinity norm,

$$\|R\|_\infty = \max_{1 \le i \le n} \frac{1}{|a_{ii}|} \sum_{j \neq i}^{n} |a_{ij}|.$$

Therefore, the convergence condition, $\|R\|_\infty < 1$, implies that $\rho(R) = \max_i \frac{1}{|a_{i,i}|} \sum_{j \neq i} |a_{i,j}| < 1$ and the method converges. The other case can be proved similarly. □

**Exercise:** is the Jacobi method convergent for the following system?

$$\begin{aligned}7x_1 + 2x_2 + 3x_3 &= -4\\ 2x_1 + 4x_2 + 0.3x_3 &= 1\\ 5x_1 + 0.2x_2 - 6x_3 &= 3\end{aligned}$$

**Answer:**

$$\begin{aligned}|7| &> |2| + |3| = 5\\ |4| &> |2| + |0.3| = 2.3\\ |-6| &> |5| + |0.2| = 5.2\end{aligned}$$

$\implies$ Jacobi converges.

**Exercise:** is the Jacobi method convergent for the following system?

$$\begin{aligned}2x_1 + 4x_2 + 0.3x_3 &= 1\\ 7x_1 + 2x_2 + 3x_3 &= -4\\ 5x_1 + 0.2x_2 - 6x_3 &= 3\end{aligned}$$

**Answer:**

$$\begin{aligned}|2| &< |4| + |0.3| = 4.3\\ |2| &< |7| + |3| = 10\\ |-6| &> |5| + |0.2| = 5.2\end{aligned}$$

$\implies$ Jacobi does not converge.

One can weaken this criterion, just a little bit so that it becomes much more useful for quite a few applications. We need the notion of irreducibility for this purpose.

**Definition 21.1.** *A matrix $A \in \mathbb{C}^{n\times n}$ is called irreducible if there is no permutation matrix $P$ such that*

$$P^T AP = \begin{pmatrix}\tilde{A}_{1,1} & \tilde{A}_{1,2}\\ 0 & \tilde{A}_{2,2}\end{pmatrix}$$

*where $\tilde{A}_{1,1} \in \mathbb{C}^{p\times p}$ and $\tilde{A}_{2,2} \in \mathbb{C}^{q\times q}$ both are square blocks with $p, q > 0$ and $(p+q) = n$, $\tilde{A}_{1,2} \in \mathbb{C}^{p\times q}$, and $0$ is the $q \times p$ vector.*

**Exercise:** is the following matrix irreducible?

$$A_1 = \begin{pmatrix}2 & 0 & 0\\ 3 & 2 & -1\\ 1 & 6 & 2\end{pmatrix}$$

**Answer:** Let us take,

$$P = \begin{pmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{pmatrix}$$

then,

$$\begin{pmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{pmatrix}\begin{pmatrix}2 & 0 & 0\\ 3 & 2 & -1\\ 1 & 6 & 2\end{pmatrix}\begin{pmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{pmatrix}=\begin{pmatrix}2 & 6 & 1\\ -1 & 2 & 3\\ 0 & 0 & 2\end{pmatrix}$$

This is in the form $P^TAP=\begin{pmatrix}\tilde{A}_{1,1} & \tilde{A}_{1,2}\\ 0 & \tilde{A}_{2,2}\end{pmatrix}$ where $\tilde{A}_{1,1}=\begin{pmatrix}2 & 6\\ -1 & 2\end{pmatrix}$, $\tilde{A}_{1,2}=\begin{pmatrix}1\\ 3\end{pmatrix}$, and $\tilde{A}_{2,2}=(2)$

**Exercise:** is the following matrix irreducible?

$$A_2=\begin{pmatrix}2 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 2\end{pmatrix}$$

**Answer:**

- Let us take,

$$P_1=\begin{pmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{pmatrix}$$

then,

$$\begin{pmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{pmatrix}\begin{pmatrix}2 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 2\end{pmatrix}\begin{pmatrix}0 & 0 & 1\\ 0 & 1 & 0\\ 1 & 0 & 0\end{pmatrix}=\begin{pmatrix}2 & -1 & 1\\ -1 & 2 & -1\\ 0 & -1 & 2\end{pmatrix}$$

This is not in the form $P^TAP=\begin{pmatrix}\tilde{A}_{1,1} & \tilde{A}_{1,2}\\ 0 & \tilde{A}_{2,2}\end{pmatrix}$.

- Let us take,

$$P_2=\begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{pmatrix}$$

then,

$$\begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{pmatrix}\begin{pmatrix}2 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 2\end{pmatrix}\begin{pmatrix}1 & 0 & 0\\ 0 & 0 & 1\\ 0 & 1 & 0\end{pmatrix}=\begin{pmatrix}2 & 0 & -1\\ 0 & -1 & 2\\ -1 & 2 & -1\end{pmatrix}$$

This is not in the form $P^TAP$.

- Let us take,

$$P_3=\begin{pmatrix}0 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}$$

then,

$$\begin{pmatrix}0 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}\begin{pmatrix}2 & -1 & 0\\ -1 & 2 & -1\\ 0 & -1 & 2\end{pmatrix}\begin{pmatrix}0 & 0 & 1\\ 1 & 0 & 0\\ 0 & 1 & 0\end{pmatrix}=\begin{pmatrix}2 & 0 & 2\\ 0 & 2 & -1\\ -1 & -1 & 0\end{pmatrix}$$

This is not in the form $P^TAP$

- Let us take,

  $$P_4 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

  then,

  $$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix}$$

  This is not in the form $P^TAP$.

- Let us take,

  $$P_5 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

  then,

  $$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix} \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & -1 & -1 \\ -1 & 2 & 0 \\ -1 & 0 & 2 \end{pmatrix}$$

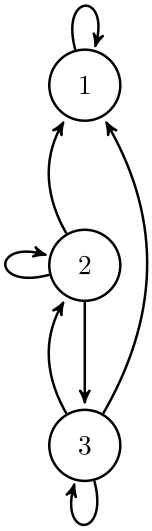
  This is not in the form $P^TAP$.

Now that we checked all the possible permutation matrices we can say that $A_2$ is irreducible.

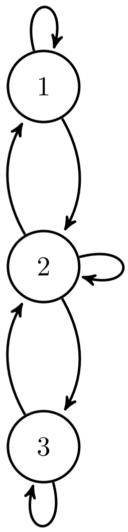
## Graph of a matrix

As can be clearly seen from the previous examples, it can be very tedious to check this irreducibility criterion using the above method. Fortunately, there is another way based on studying the $\underline{\text{oriented graph}}$ $G(A)$ of the matrix $A$. It consists of the vertices $1, \ldots, n$, and there is an (oriented) edge from vertex $i$ to vertex $j$ (denoted by $i \to j$) if $a_{i,j} \neq 0$.

The graph for the matrix $A_1 = \begin{pmatrix} 2 & 0 & 0 \\ 3 & 2 & -1 \\ 1 & 6 & 2 \end{pmatrix}$,



The graph for the matrix $A_2 = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$,



We say that two vertices $i, j$ are connected if there is a chain of connecting edges (or direct connections) $i = i_0 \to i_1 \to \cdots \to i_k = j$ with some $k \in \mathbb{N}$. The graph $G(A)$ then is called connected if any two vertices $i, j$ of it are connected. Irreducibility of the matrix $A$ now may be checked using the following lemma.

**Lemma 21.2.** *$A$ is irreducible if and only if $G(A)$ is connected.*

This means that $G(A_1)$ is not connected. This is because there is no way to go from node 1 to node 2 or node 1 to node 3. Thus $A_1$ is reducible.
However, $G(A_2)$ is connected. This is because,

$$\begin{aligned} &v_1 \to v_2 \to v_3 \to v_2 \to v_1 \\ &v_2 \to v_3 \to v_2 \\ &v_3 \to v_2 \to v_1 \to v_2 \to v_3 \\ &v_3 \to v_3 \\ &v_2 \to v_2 \end{aligned}$$

Thus $A_2$ is irreducible.

Back to the question of convergence of the Jacobi method:

**Theorem 21.2.** *If $A$ is irreducible and satisfies the weak row sum criterion*

*(1) $|a_{i,i}| \geq \sum_{j \neq i} |a_{i,j}|$ for all $i = 1, \ldots, n$,* **and**

*(2) $|a_{k,k}| > \sum_{j \neq k} |a_{k,j}|$ for at least one index $k \in \{1, \ldots, n\}$*

*then the Jacobi method converges.*

*Proof.* Recall that we need to prove that $\rho(R) < 1$. Let $e := (1, \dots, 1)^T \in \mathbb{C}^n$ and $|R| = (|r_{i,j}|)_{i,j}$. Then thanks to the first condition

$$0 \leq \big(|R|e\big)_i = \sum_{j=1}^{n} |r_{i,j}| = \sum_{j \neq i} \frac{|a_{i,j}|}{|a_{i,i}|} \leq 1 = e_i$$

so that $e \geq |R|e \geq |R|^2 e \geq \dots$ where the inequality for vectors here and in the following has to be understood component-wise.
Let $t^{(l)} := e - |R|^l e \geq 0$, $l \in \mathbb{N}$. Assume now that there is a positive number of non-vanishing components of $t^{(l)}$ that become stationary. We may assume that these are the first $m$ entries where $m > 0$ thanks to the second condition, i.e.,

$$t^{(l)} = \begin{pmatrix} b^{(l)} \\ 0 \end{pmatrix}, \quad t^{(l+1)} = \begin{pmatrix} b^{(l+1)} \\ 0 \end{pmatrix}$$

where $b^{(l)}, b^{(l+1)} \in \mathbb{R}^m$ have positive entries, $b^{(l)} > 0, b^{(l+1)} > 0$.
Suppose that $m < n$. Then

$$\begin{pmatrix} b^{(l+1)} \\ 0 \end{pmatrix} = e - |R|^{l+1} e \geq |R|e - |R|^{l+1} e = |R|(e - |R|^l e) = |R| \begin{pmatrix} b^{(l)} \\ 0 \end{pmatrix} = \begin{pmatrix} |R_{1,1}| & |R_{1,2}| \\ |R_{2,1}| & |R_{2,2}| \end{pmatrix} \begin{pmatrix} b^{(l)} \\ 0 \end{pmatrix}$$

with $R_{1,1} \in \mathbb{R}^{m \times m}$ and the other blocks accordingly. Since $b^{(l)} > 0$ necessarily $|R_{2,1}| = 0$. Therefore $R$ is not irreducible. And since $r_{i,j} = a_{i,j}/a_{i,i}$ if $i \neq j$ we obtain that $A$ is not irreducible in contradiction to the assumption. Hence $m = n$.
Consequently, $t^{(l)} > 0$ as long as $l$ is big enough (using the above contradiction argument again we see that $l > n$ is sufficient). This means that $e > |R|^l e$, whence

$$\rho(R)^l \leq \rho(R^l) \leq \|R^l\|_\infty \leq \||R|^l\|_\infty = \max_i \big(|R|^l e\big)_i < \max_i e_i = 1$$

so that $\rho(R) < 1$ as desired. □

**Exercise:** Consider the matrix,

$$A_2 = \begin{pmatrix} 2 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 2 \end{pmatrix}$$

$$\begin{aligned} |2| &> |-1| + |0| = 1 \\ |2| &= |-1| + |-1| = 2 \\ |2| &> |0| + |-1| = 1 \end{aligned}$$

Thus, $A_2$ is irreducibly diagonally dominant.

# Lecture 22

# Computational Complexity of Linear Iterative Methods

## Learning Outcomes

- **Understanding Computational Complexity:**
  - Gain insights into the computational complexity of linear iterative methods, focusing on achieving a specified relative backward error and understanding the impact of the iteration matrix $R$ and other data $A$, $b$, and $e^{(0)}$ on the number of steps required.
- **Analyzing the Jacobi Method:**
  - Develop proficiency in analyzing the computational complexity of the Jacobi method, understanding the cost involved in each iteration step, and how sparsity of the matrix can affect the number of operations required.
- **Exploring Variants of Jacobi Method:**
  - Explore and understand the variants of the Jacobi method, such as the Successive Over Relaxation (SOR) method and the Gauss-Seidel method, and learn how the relaxation parameter $\omega$ can influence the spectral properties of the iteration matrix $R$ and the convergence speed of the method.

In some applications, the goal will be to decrease the relative forward error $\|e^{(k)}\|/\|x\|$ below a given threshold while in others it is sufficient to decrease the relative backward error $\|r^{(k)}\|/\|b\|$. We will concentrate on the latter goal but recall that by (20.1) the two goals are related. Of course, the knowledge of the condition number is required to deduce an estimate for the forward error from the backward error.
So our goal is

$$\|r^{(k)}\| \leq \varepsilon_r \|b\| \tag{22.1}$$

where $\varepsilon_r > 0$ is a given tolerance. Recall that $r^{(k)}$ is related to $e^{(k)}$ as follows,

$$\begin{aligned}
&e^{(k)} := x - x^{(k)} \quad \text{multiplying both sides by A} \implies \\
&Ae^{(k)} := A(x - x^{(k)}) \implies \\
&Ae^{(k)} := b - Ax^{(k)} = r^{(k)}
\end{aligned}$$

It was also shown that,

$$e^{(k)} = R^k e^{(0)}$$

Hence,

$$AR^k e^{(0)} = r^{(k)}$$

Taking the norm of the above equation we get,

$$\|r^{(k)}\| \leq \|A\| \|R^k\| \|e^{(0)}\|$$

Using this in (22.1) yields,

$$\|A\| \|R\|^k \|e^{(0)}\| \leq \varepsilon_r \|b\| \quad \Leftrightarrow \quad \underbrace{\left(\frac{1}{\|R\|}\right)}_{>1}^k \geq \frac{\|A\| \|e^{(0)}\|}{\|b\| \varepsilon_r}$$

Taking the log of both sides yields the following condition that needs to be satisfied in order to achieve our goal,

$$k \geq \frac{\log(\|A\|) + \log(\|e^{(0)}\|) - \log(\|b\|) - \log(\varepsilon_r)}{\log(\|R\|^{-1})} =: k^\sharp(n, \varepsilon_r). \tag{22.2}$$

In practice, the iteration matrix $R$ often depends on $n$ in a very unfavourable way while the other data $A$, $b$ and $e^{(0)}$ do not affect the number of steps that much.

**Assumption 22.1.**

1. *The calculation of $Rx$ involves a cost of $\Theta(n^\alpha)$ as $n \to \infty$ with some $\alpha > 0$.*
2. *$\|R\| = 1 - h(n)$ with a positive function $h$ such that $h(n) = \Theta(n^{-\beta})$ as $n \to \infty$ with some $\beta > 0$.*
3. *$\|A\|$, $\|b\|$, and $\|e^{(0)}\|$ are uniformly bounded in $n$.*

**Theorem 22.1.** *Under Assumption 22.1, the computational cost to achieve (22.1) is bounded by a function $C(n, \varepsilon_r)$ satisfying*

$$C(n, \varepsilon_r) = \Theta(n^{\alpha+\beta} \log(\varepsilon_r^{-1})) \quad \textit{as } (n, \varepsilon_r) \to (\infty, 0).$$

*Proof.* Recall that $\log(1/(1-x)) = x + x^2/2 + x^3/3 + \dots$. Thus, by Assumption 22.1.2

$$\log(\|R\|^{-1}) = \log(1/(1 - h(n))) = h(n) + \text{ higher order terms } = \Theta(n^{-\beta})$$

so that $(\log(\|R\|^{-1}))^{-1} = \Theta(n^\beta)$ as $n \to \infty$. From (22.2) and Assumption 22.1.3 we get

$$k^\sharp(n, \varepsilon_r) = \Theta(n^\beta \log(\varepsilon_r^{-1})) \quad \text{as } (n, \varepsilon_r) \to (\infty, 0)$$

for the number of steps. Taking the cost per step into account (Assumption 22.1.1), the total cost is

$$k\sharp(n, \varepsilon_r) C_{\text{one\_step}}(n) = \Theta(n^{\alpha+\beta} \log(\varepsilon_r^{-1})).$$

□

Assuming a polynomial dependence in Assumption 22.1.3 one would obtain additional $\log(n)$ terms in the cost estimate.

## Computational complexity of the Jacobi method

In each iteration step:

- computing $(L+U)x^{(k-1)}$ involves at most $O(n^2)$ operations,
- computing $b - \dots$ is $O(n)$,
- computing $D^{-1}(\dots)$ is $O(n)$, too.

So the essential cost is coming from the first step. If the matrix is sparse, i.e., the number of non-vanishing entries in each row is $\sim n^\eta$ with some $\eta < 1$ then the number of operations is $O(n^{1+\eta})$. For instance, $\eta = 0$ if $A$ is tridiagonal (has nonzero elements on the main diagonal, on the subdiagonal/lower diagonal, and on the supradiagonal/upper diagonal only ) as the number of non-vanishing entries in each row is bounded by a constant (namely 3). In any case, with $\alpha \in [1,2]$ the general result on the computational cost in Theorem 22.1 is applicable.

## Variants of the Jacobi method

The decomposition of $A = L + D + U$ in the Jacobi method is beneficial for the Gauss-Seidel (AKA Liebmann method) where the iterations are computed as follows,

$$x^{(k)} = D^{-1}(b - Ux^{(k-1)} - Lx^{(k)})$$

This variant of the Jacobi method require less memory and converges faster.

Another variant of the Jacobi method is called the successive over relaxation (SOR) method which generalises the Jacobi method by setting,

$$M := \omega L + D, \quad N := \omega U + (\omega - 1)D, \qquad \omega \in \mathbb{R}.$$

which yields the following iterations,

$$x^{(k)} = (D + \omega L)^{-1}(\omega b - [\omega U + (\omega - 1)D]x^{(k-1)})$$

For $\omega = 1$ we obtain the Gauss-Seidel method.

Results for convergence criteria and computations cost read similarly. Yet the relaxation parameter $\omega$ can have a massive influence on the spectral properties of the iteration matrix $R$ and speed up the convergence a lot. The notion *over-relaxation* refers to choosing $\omega > 1$, hence bigger than in the Gauss-Seidel method.

# Lecture 23

# Nonlinear Iterative Methods, Steepest Descent

## Learning Outcomes

- **Understanding Nonlinear Iterative Methods:**
  - Acquire knowledge on the application of nonlinear iterative methods to solve Symmetric Linear Equations (SLEs) with positive definite matrices, focusing on minimizing the residual in the $\| \cdot \|_{A^{-1}}$ norm or the error in the $\| \cdot \|_A$ norm.
- **Mastering Steepest Descent Method:**
  - Develop an understanding of the Steepest Descent Method, learning how to choose the search direction and step length to minimize the target function $g$, and how to implement the method efficiently with only one matrix-vector multiplication per iteration step.
- **Analyzing Convergence and Condition Number:**
  - Gain insights into the convergence behavior of the Steepest Descent Method, understanding the role of the condition number $\kappa_2(A)$ in the elongation of the level sets of $g$ and its impact on the convergence of the method.

Steepest Descent is one of the iterative methods for solving the least square problem where the objective is to model some measured data, $b \in \mathbb{R}^m$ for a system model represented in $A \in \mathbb{R}^{m \times n}$ operating on some parameter vector, $x \in \mathbb{R}^n$

$$b = Ax + \epsilon$$

Since the measured data is noisy, there will in general be a discrepancy between $Ax$ and the measured data, $b$. This means that there exists no model that can match the measured data due to the noise and other inconsistencies. Thus, the least square problem is formulated as finding a least square estimate of the vector parameter, $x$ with the objective of minimising the discrepancy between the actual data of the system and the estimated one,

$$\hat{x} = \underset{x}{\operatorname{argmin}} \frac{1}{2} \|b - Ax\|_2$$

Similarly, the steepest descent method defines some differentiable function of the unknown parameter vector $x$, and solves the minimum of that function.

Here we restrict our attention to positive definite (symmetric) matrices $A \in \mathbb{R}^{n\times n}$.

Recall the notation $\langle x, y\rangle_A := \langle x, Ay\rangle$ and $\|x\|_A := \sqrt{\langle x, x\rangle_A}$.

The differentiable function in the steepest descent method is defined as,

$$g : \mathbb{R}^n \to \mathbb{R}, \quad g(y) = \frac{1}{2}\|Ax - b\|^2_{A^{-1}}. \tag{23.1}$$

Clearly, $x \in \mathbb{R}^n$ solves $Ax = b$ if and only if $x$ is minimiser of this function.

Recalling that $e^{(k)} = x - x^{(k)}$, $r^{(k)} = b - Ax^{(k)} = Ae^{(k)}$ one can easily show that

$$g(x^{(k)}) = \frac{1}{2}\|r^{(k)}\|^2_{A^{-1}} = \frac{1}{2}\|e^{(k)}\|^2_A. \tag{23.2}$$

Hence, minimising $g$ means

- minimising the residual in the $\|\cdot\|_{A^{-1}}$ norm or
- minimising the error in the $\|\cdot\|_A$ norm (energy norm).

Steepest descent is a nonlinear iterative method for solving SLE's which considers solutions that are in a certain direction from the current point,

$$x^{(k)} = x^{(k-1)} + \alpha^{(k-1)} d^{(k-1)}, \tag{23.3}$$

where $d^{(k-1)} \in \mathbb{R}^n$ is the search direction and $\alpha^{(k-1)} \in \mathbb{R}$ is the step length. The step length is chosen such that

$$f(\alpha) := g(x^{(k-1)} + \alpha d^{(k-1)})$$

is minimal. This uniquely determines $\alpha^{(k-1)}$ since $f$ is convex and tends to infinity as $|\alpha| \to \infty$. This also allows to derive an explicit formula for the step length:

$$\begin{aligned}
f(\alpha) &= \frac{1}{2}\langle \alpha A d^{(k-1)} + \underbrace{Ax^{(k-1)} - b}_{-r^{(k-1)}}, A^{-1}(\alpha A d^{(k-1)} + \underbrace{Ax^{(k-1)} - b}_{-r^{(k-1)}})\rangle \\
&= \frac{1}{2}\alpha^2\langle Ad^{(k-1)}, d^{(k-1)}\rangle + \frac{1}{2}\alpha\langle Ad^{(k-1)}, -A^{-1}r^{(k-1)}\rangle \\
&\quad + \frac{1}{2}\alpha\langle -r^{(k-1)}, d^{(k-1)}\rangle + \frac{1}{2}\langle -r^{(k-1)}, -A^{-1}r^{(k-1)}\rangle \\
&= \frac{1}{2}\alpha^2\|d^{(k-1)}\|^2_A - \alpha\langle d^{(k-1)}, r^{(k-1)}\rangle + \frac{1}{2}\|r^{(k-1)}\|^2_{A^{-1}} \qquad (23.4) \\
\Rightarrow \quad f'(\alpha) &= \alpha\|d^{(k-1)}\|^2_A - \langle d^{(k-1)}, r^{(k-1)}\rangle
\end{aligned}$$

As the minimiser fulfils $f'(\alpha^{(k-1)}) = 0$ we obtain

$$\alpha^{(k-1)} = \frac{\langle r^{(k-1)}, d^{(k-1)}\rangle}{\|d^{(k-1)}\|^2_A}. \tag{23.5}$$

Before we start looking at possible search directions we make two observations:

1. The residual is subject to the iterative formula

$$r^{(k)} = b - Ax^{(k)} = b - Ax^{(k-1)} - \alpha^{(k-1)}Ad^{(k-1)} = r^{(k-1)} - \alpha^{(k-1)}Ad^{(k-1)}. \tag{23.6}$$

2. It can be shown that,

$$\nabla g(x^{(k-1)}) = -r^{(k-1)}$$

### Steepest Descent Method

The idea of this method is to choose $d^{(k-1)} = r^{(k-1)} = -\nabla g(x^{(k-1)})$. This choice is motivated from the fact that the gradient points in the direction of the fastest increase, hence a sufficiently small step in direction $-\nabla g(x^{(k-1)})$ will decrease the value of our target function $g$ that is to be minimised. According to (23.5) the optimal step length then is $\alpha^{(k-1)} = \|r^{(k-1)}\|_2^2/\|r^{(k-1)}\|_A^2$.

**Algorithm 12 SD** (steepest descent method)

```
input: $A = (a_{ij})_{i,j=1}^n \in \mathbb{R}^{n\times n}$ positive definite, $b, x^{(0)} \in \mathbb{R}^n$, $\varepsilon_r > 0$.
output: $x \in \mathbb{R}^n$ with $\|Ax - b\|_2 \leq \varepsilon_r$.
for $k = 1, 2, \ldots$ do
    $r^{(k-1)} := b - Ax^{(k-1)}$
    if $\|r^{(k-1)}\|_2 \leq \varepsilon_r$ then
        return $x^{(k-1)}$
    else
        $\alpha^{(k-1)} := \|r^{(k-1)}\|_2^2/\|r^{(k-1)}\|_A^2$
        $x^{(k)} := x^{(k-1)} + \alpha^{(k-1)}r^{(k-1)}$
    end if
end for
```

Note that in the above algorithm the following two matrix vector multiplications are required,

- To compute the denominator in $\alpha^{(k-1)}$, one need to compute $d^{(k-1)^T}Ad^{(k-1)}$,
- To compute $r^{(k-1)}$ one need to compute $Ax^{(k-1)}$.

Thus to reduce the number of matrix vector multiplications, we can use the fact that $r^{(k)} = r^{(k-1)} - \alpha^{(k-1)}Ad^{(k-1)}$ as stated in (23.6) and then introduce a help variable $h^{(k-1)} = Ad^{(k-1)}$ to formulate the algorithm such that only one matrix-vector multiplication per iteration step is required (exercise).

One observes that subsequent search directions are orthogonal with respect to the standard scalar product: Thanks to (23.6) and $d^{(k-1)} = r^{(k-1)}$

$$\langle r^{(k-1)}, r^{(k)}\rangle = \langle r^{(k-1)}, r^{(k-1)} - \alpha^{(k-1)}Ar^{(k-1)}\rangle$$
$$= \langle r^{(k-1)}, r^{(k-1)}\rangle - \frac{\|r^{(k-1)}\|_2^2}{\|r^{(k-1)}\|_A^2}\langle r^{(k-1)}, Ar^{(k-1)}\rangle = 0.$$

The effect is a zig-zag path of the iterates when approaching the minimum of $g$ as illustrated in Figure 23.1. Since $g$ is a quadratic function the level sets of $g$ are ellipsoids. As an observation, the longer the ellipsoids are stretched, the longer it takes for the algorithms to obtain an iterate close to the minimum. It turns out that the main axes of the level set ellipsoids are the

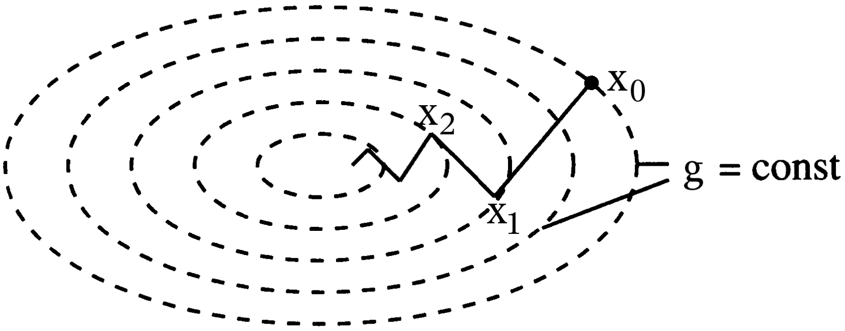


Figure 23.1: Behaviour of **SD**, zigzag path to the minimum due to orthogonal (w.r.t. the Euclidean scalar product) search directions.

eigenspaces, and the stretching of the ellipsoids depends on the ratio of the eigenvalues, most prominently $\lambda_{\max}/\lambda_{\min}$. Recalling that for positive definite matrices $\|A\|_2 = \lambda_{\max}$ and $\|A^{-1}\|_2 = 1/\lambda_{\min}$ we see that it is exactly the condition number $\kappa_2(A) = \|A\|_2\|A^{-1}\|_2 = \lambda_{\max}/\lambda_{\min}$ that can serve as a measure how elongated the level sets are. We will see later on how the condition number influences the convergence of **SD**.