Numerical methods course

chapiter 3:

Iterative Methods for Solving Linear Systems

By Dr, Asmaa Ourdighi

USTO MB

2023-2024

Numerical methods course
chapter 3:
Iterative methods for solving linear systems

- **Introduction**
- Jacobi and Relaxation Methods
- Gauss-Seidel and Successive Relaxation Methods
- Remarks on the Implementation of Iterative Methods
- Convergence of Jacobi and Gauss-Seidel Methods

Numerical methods course

chapter 3:

Iterative methods for solving linear systems

- **Introduction**

- Direct methods: are good for dense systems of small to moderate size. This generally means the entire coefficient matrix for the linear system can be stored in the computers memory. Then Gaussian elimination can be applied to solve the linear system.
- Iteration methods: used with large sparse systems and large dense systems. Discretizations of boundary value problems for PDEs(partial differential equations) often lead to large sparse systems, and there is generally a pattern to the sparsity.
    - Traditional iteration - Jacobi, Gauss-Seidel, SOR, red/black iteration, line iteration
    - Multigrid iteration - Uses many levels of discretization
    - Conjugate gradient iteration and variants of it.

Numerical methods course
chapiter 3:
Iterative methods for solving linear systems

- **Introduction**

- Iterative methods formally yield the solution x of a linear system after an infinite number of steps. At each step they require the computation of the residual of the system.

- Iterative methods have the advantage of being easier to program and require less memory space.

- Iterative linear systems
- We call iterative method of resolution of the linear system $Ax=b$ a method which constructs a sequence $(x^{(k)})_{k\in IN}$

where the iterate $x^{(k)}$ is calculated from the iterates $x^{(0)} \ldots x^{(k-1)}$ supposed to converge towards x solution of the linear system $Ax=b$

- The usual strategy consists of constructing the sequence $x^{(k)}$ by one recurrence relation of the form

$$x^{(k+1)} = B\,x^{(k)} + C$$

The iteration matrix of method is B

- Iterative linear systems
- A general technique to devise consistent linear iterative methods is based on an additive splitting of the matrix A of the form

$$A=P-N$$

where P and N are two suitable matrices and P is nonsingular.

P is called preconditioning matrix or preconditioner. Precisely, given x(0), one can compute x(k) for k ≥ 1, solving the systems

$$\mathbf{P}\mathbf{x}^{(k+1)} = \mathbf{N}\mathbf{x}^{(k)} + \mathbf{b}, \quad k \geq 0.$$

$$x^{(k+1)} = P^{-1}N\, x^{(k)} + P^{-1}b$$

The iteration matrix of method is $B = P^{-1}N = Id - P^{-1} A$ (N=P-A)

- Iterative linear systems
- In practice, the most classic splittings are based on writing

$$A = D - E - F$$

- Where:
  - D is the diagonal matrix of the diagonal entries of A
  - E is the lower triangular matrix of entries $E_{ij} = -A_{ij}$ if $i > j$, $E_{ij} = 0$ if $i \leq j$
  - F is the upper triangular matrix of entries $F_{ij} = -A_{ij}$ if $j > i$, $F_{ij} = 0$ if $j \leq i$

As a consequence, $A = D - (E + F)$.

Numerical methods course
chapiter 3:
Iterative methods for solving linear systems

- Iterative linear systems

$$D=\begin{pmatrix} a_{11} & 0 & 0 & 0 & 0 \\ 0 & \ddots & 0 & 0 & 0 \\ 0 & 0 & a_{ii} & 0 & 0 \\ 0 & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & 0 & a_{nn} \end{pmatrix} \qquad -E=\begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ a_{21} & \ddots & 0 & 0 & 0 \\ . & & & 0 & 0 \\ . & & & \ddots & 0 \\ a_{n1} & \ldots.. & & a_{n,n-1} & 0 \end{pmatrix}$$

$$-F=\begin{pmatrix} 0 & a_{12} & \ldots. & \ldots. & a_{1n} \\ 0 & \ddots & & & . \\ . & 0 & 0 & & . \\ . & 0 & 0 & \ddots & a_{n-1,n} \\ 0 & \ldots.. & 0 & 0 & 0 \end{pmatrix}$$

- ## Jacobi method
- It consists of choosing the simplest splitting with $P = D$ et $N = E + F$.
- By replacing in the equation $x^{(k+1)} = P^{-1}N\, x^{(k)} + P^{-1}b$, we obtain

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

We call the Jacobi matrix the iteration matrix of the method

$$J = D^{-1}(E + F) = Id - D^{-1}A$$

This matrix form will only be used for the analysis of convergence of the method,

Numerical methods course

chapiter 3:

Iterative methods for solving linear systems

- Jacobi method
- The implementation of the corresponding algorithm consists of calculating consider the following linear system

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots\ldots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots\ldots + a_{2n}x_n = b_2 \\ \\ \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots\ldots + a_{nn}x_n = b_n \end{cases}$$

- Jacobi method

$$\begin{cases} x_1 = (-a_{12}x_2 - \ldots - a_{1n}x_n + b_1)/a_{11} \\ x_2 = (-a_{21}x_1 - a_{23}x_3 - \ldots - a_{2n}x_n + b_2)/a_{22} \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ x_i = (-a_{i1}x_1 - \ldots - a_{ii-1}x_{i-1} - a_{ii+1}x_{i+1} - \ldots - a_{in}x_n + b_i))/a_{ii} \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ x_n = (-a_{n1}x_1 - a_{n2}x_2 - \ldots - a_{nn-1}x_{n-1} + b_n)/a_{nn} \end{cases}$$

Numerical methods course
chapiter 3:
Iterative methods for solving linear systems

- Jacobi method

$$\begin{cases} x_1^{(k+1)} = (-a_{12}x_2^{(k)} - \dots - a_{1n}x_n^{(k)} + b_1)/a_{11} \\ x_2^{(k+1)} = (-a_{21}x_1^{(k)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)} + b_2)/a_{22} \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ x_i^{(k+1)} = (-a_{i1}x_1^{(k)} - \dots - a_{ii-1}x_{i-1}^{(k)} - a_{ii+1}x_{i+1}^{(k)} - \dots - a_{in}x_n^{(k)} + b_i))/a_{ii} \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ x_n^{(k+1)} = (-a_{n1}x_1^{(k)} - a_{n2}x_2^{(k)} - \dots - a_{nn-1}x_{n-1}^{(k)} + b_n)/a_{nn} \end{cases}$$

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

- Jacobi method
- Exemple 1

$$\begin{cases} 3x_1 + x_2 - x_3 = 2 \\ x_1 + 5x_2 + 2x_3 = 17 \\ 2x_1 - x_2 - 6x_3 = -18 \end{cases}$$

$$\begin{cases} x_1^{k+1} = \frac{1}{3}(2 - x_2^k + x_3^k) \\ x_2^{k+1} = \frac{1}{5}(17 - x_1^k - 2x_3^k) \\ x_3^{k+1} = -\frac{1}{6}(-18 - 2x_1^k + x_2^k) \end{cases}$$

- First iteration $X^{(0)}=(0,0,0)^T$

$$\begin{cases} x_1^1 = \frac{1}{3}(2 - 0 + 0) = \frac{2}{3} \\ x_2^1 = \frac{1}{5}(17 - 0 - 0) = \frac{17}{5} \\ x_3^1 = -\frac{1}{6}(-18 - 0 + 0) = 3 \end{cases}$$

- Jacobi method
- Exemple 1
- Second iteration $X^{(1)}=(2/3,17/5,3)^T$

$$\begin{cases} x_1^1 = \frac{1}{3}\left(2 - \frac{17}{5} + 3\right) = \frac{8}{15} \\ x_2^1 = \frac{1}{5}\left(17 - \frac{2}{3} - 2(3)\right) = \frac{31}{15} \\ x_3^1 = -\frac{1}{6}\left(-18 - 2\left(\frac{2}{3}\right) + \frac{17}{5}\right) = 2{,}655556 \end{cases}$$

- Jacobi method
- Exemple 1
- The solution is X=(1,2,3)^T

$$|x^{(k+1)} - x^{(k)}| < \varepsilon$$

k	x_1^k	x_2^k	x_3^k
0	0,00000	0,00000	0,00000
1	0,66667	3.40000	3.00000
2	0.53333	2.06667	2.65556
3	0.86296	2.23111	2.83333
4	0.86741	2.09407	2.91580
5	0.94057	2.06020	2.94012
6	0.95997	2.03583	2.97016
7	0.97811	2.01994	2.98069
8	0.98691	2.01210	2.98938
9	0.99242	2.00686	2.99362
10	0.99558	2.00407	2.99633

- ## Gauss seidel method
- It consists of choosing the simplest splitting with $P = D-E$ et $N = F$.
- By replacing in the equation $x^{(k+1)} = P^{-1}N\, x^{(k)} + P^{-1}b$, we obtain

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

We call the Jacobi matrix the iteration matrix of the method

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

Numerical methods course

chapiter 3:

Iterative methods for solving linear systems

- Gauss seidel method

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

$$\begin{cases} x_1 = (-a_{12}x_2 - \dots - a_{1n}x_n + b_1)/a_{11} \\ x_2 = (-a_{21}x_1 - a_{23}x_3 - \dots - a_{2n}x_n + b_2)/a_{22} \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ x_i = (-a_{i1}x_1 - \dots - a_{ii-1}x_{i-1} - a_{ii+1}x_{i+1} - \dots - a_{in}x_n + b_i))/a_{ii} \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ x_n = (-a_{n1}x_1 - a_{n2}x_2 - \dots - a_{nn-1}x_{n-1} + b_n)/a_{nn} \end{cases}$$

Numerical methods course

chapiter 3:

Iterative methods for solving linear systems

- Gauss seidel method

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

$$\begin{cases} x_1^{(k+1)} = (-a_{12}x_2^{(k)} - \dots - a_{1n}x_n^{(k)} + b_1)/a_{11} \\ x_2^{(k+1)} = (-a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)} + b_2)/a_{22} \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ x_i^{(k+1)} = (-a_{i1}x_1^{(k+1)} - \dots - a_{ii-1}x_{i-1}^{(k+1)} - a_{ii+1}x_{i+1}^{(k)} - \dots - a_{in}x_n^{(k)} + b_i)/a_{ii} \\ \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \quad \cdots \\ x_n^{(k+1)} = (-a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \dots - a_{nn-1}x_{n-1}^{(k+1)} + b_n)/a_{nn} \end{cases}$$

Numerical methods course

chapiter 3:

Iterative methods for solving linear systems

- Gauss seidel method

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

$$\begin{cases} x_1^{(k+1)} = (-a_{12}x_2^{(k)} - \dots - a_{1n}x_n^{(k)} + b_1)/a_{11} \\ x_2^{(k+1)} = (-a_{21}x_1^{(k+1)} - a_{23}x_3^{(k)} - \dots - a_{2n}x_n^{(k)} + b_2)/a_{22} \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ x_i^{(k+1)} = (-a_{i1}x_1^{(k+1)} - \dots - a_{ii-1}x_{i-1}^{(k+1)} - a_{ii+1}x_{i+1}^{(k)} - \dots - a_{in}x_n^{(k)} + b_i)/a_{ii} \\ \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \quad \dots \\ x_n^{(k+1)} = (-a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \dots - a_{nn-1}x_{n-1}^{(k+1)} + b_n)/a_{nn} \end{cases}$$

- # Gauss seidel method
- Exemple 1

$$\begin{cases} 3x_1 + x_2 - x_3 = 2 \\ x_1 + 5x_2 + 2x_3 = 17 \\ 2x_1 - x_2 - 6x_3 = -18 \end{cases}$$

- Initial vector

$x^{(0)}=(0,0,0)^T$

$$x_1^1 = \frac{1}{3}(2 - 0 + 0) = \frac{2}{3}$$

$$x_2^1 = \frac{1}{5}\left(17 - \frac{2}{3} - 0\right) = \frac{49}{15}$$

$$x_3^1 = -\frac{1}{6}\left(-18 - 2\left(\frac{2}{3}\right) + \frac{49}{15}\right) = \frac{241}{90}$$

$$x_1^2 = \frac{1}{3}(2 - \frac{49}{15} + \frac{241}{90}) = 0.47$$

$$x_2^2 = \frac{1}{5}\left(17 - 0.47 - 2\left(\frac{241}{90}\right)\right) = 2.235$$

$$x_3^2 = -\frac{1}{6}(-18 - 2(0.47) + 2.235) = 2,784$$

- # Gauss seidel method
- Exemple 1

k	x_1^k	x_2^k	x_3^k
0	0,00000	0,00000	0,00000
1	0,66667	3.26667	2.67778
2	0.47037	2.23481	2.78432
3	0.84983	2.11630	2.93056
4	0.93808	2.04016	2.97267
5	0.97750	2.01543	2.98993
6	0.99150	2.00573	2.99621
7	0.99683	2.00515	2.99858
8	0.99881	2.00080	2.99947
9	0.99955	2.00030	2.99980
10	0.99983	2.00011	2.99993

x= [1 2 3]

Successive Relaxation Method (SOR)

- In this method, we slightly modify the previous method by introducing a parameter w, the relaxation coefficient. This parameter is generally constant. Relaxation in the Jacobi method typically does not provide any significant gains. However, when applied to the Gauss-Seidel method, it improves the speed of convergence. The update formula becomes:

$$\begin{cases} X^{(0)} \in IR^n \\ x_i^{(k+1)} = (1-\omega)x_i^{(k)} + \frac{\omega}{a_{ii}}(b_i - \sum_{j<i} a_{ij}\, x_j^{(k+1)} - \sum_{j>i} a_{ij}\, x_j^{(k)}) \end{cases}$$

- # Successive Relaxation Method (SOR)

$$X^{(k+1)} = \left(\frac{D}{\omega} - E\right)^{-1}\left(\left(\frac{1-\omega}{\omega}\right)D + F\right)X^{(k)} + \left(\frac{D}{\omega} - E\right)^{-1} b$$

$$\boldsymbol{B_{SOR}} = \left(\frac{D}{\omega} - E\right)^{-1}\left(\left(\frac{1-\omega}{\omega}\right)D + F\right) \qquad and \qquad C = \left(\frac{D}{\omega} - E\right)^{-1} b$$

- Let Solve the following sytem based on A=D-E-F splitting

$$\begin{cases} 3x_1 + x_2 - x_3 = 2 \\ x_1 + 5x_2 + 2x_3 = 17 \\ 2x_1 - x_2 \pm 6x_3 = -18 \end{cases}$$

- Successive Relaxation Method (SOR)

$$D = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & -6 \end{pmatrix} \qquad E = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -2 & 1 & 0 \end{pmatrix} \qquad F = \begin{pmatrix} 0 & -1 & 1 \\ 0 & 0 & -2 \\ 0 & 0 & 0 \end{pmatrix}$$

Calculate $\boldsymbol{B_{SOR}} = \left(\frac{D}{\omega} - E\right)^{-1}\left(\left(\frac{1-\omega}{\omega}\right)D + F\right)$

$$\left(\frac{D}{\omega} - E\right) = \begin{pmatrix} \frac{3}{1.1} & 0 & 0 \\ 0 & \frac{5}{1.1} & 0 \\ 0 & 0 & -\frac{6}{1.1} \end{pmatrix} - \begin{pmatrix} 0 & 0 & 0 \\ -1 & 0 & 0 \\ -2 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 2{,}72 & 0 & 0 \\ 1 & 4{,}54 & 0 \\ 2 & -1 & -5{,}45 \end{pmatrix}$$

$$\left(\frac{D}{\omega} - E\right)^{-1} = \begin{pmatrix} 0.37 & 0 & 0 \\ -0.08 & 0.22 & 0 \\ 0.15 & -0.04 & -0.18 \end{pmatrix}$$

- Successive Relaxation Method (SOR)

$$\left(\left(\frac{1-\omega}{\omega}\right)D+F\right)=\left(\left(\frac{1-1.1}{1.1}\right)D+F\right)=\left((-0{,}09)D+F\right)=\begin{pmatrix}-0.27. & -1 & 1\\ 0 & -0.45 & -2\\ 0 & 0 & 0.54\end{pmatrix}$$

- $\boldsymbol{B_j}=\left(\frac{D}{\omega}-E\right)^{-1}\left(\left(\frac{1-\omega}{\omega}\right)D+F\right)\rightarrow \boldsymbol{B_j}=$

$$\begin{pmatrix}0.37 & 0 & 0\\ -0.08 & 0.22 & 0\\ 0.15 & -0.04 & -0.18\end{pmatrix}\begin{pmatrix}-0.27. & -1 & 1\\ 0 & -0.45 & -2\\ 0 & 0 & 0.54\end{pmatrix}=\begin{pmatrix}-0.09 & -0.37 & 0.37\\ 0.02 & 0.02 & -0.52\\ -0.04 & -0.13 & 0.13\end{pmatrix}$$

- $C=\left(\frac{D}{\omega}-E\right)^{-1}b=\begin{pmatrix}0.37 & 0 & 0\\ -0.08 & 0.22 & 0\\ 0.15 & -0.04 & -0.18\end{pmatrix}\begin{pmatrix}2\\ 17\\ -18\end{pmatrix}=\begin{pmatrix}0.74\\ 3.58\\ 2.86\end{pmatrix}$

then $\boldsymbol{B_j}=\begin{pmatrix}-0.09 & -0.37 & 0.37\\ 0.02 & 0.02 & -0.52\\ -0.04 & -0.13 & 0.13\end{pmatrix}$ and C=$\begin{pmatrix}0.74\\ 3.58\\ 2.86\end{pmatrix}$

- Remarks on the Implementation of Iterative Methods

When implementing iterative methods for solving linear systems, several factors can significantly impact their efficiency and effectiveness. Here are some key considerations:

1. Initial Guess:

Choice: The initial guess can significantly influence convergence speed. A good initial guess can accelerate convergence, while a poor one may lead to slow convergence or even divergence.

Strategies: Common strategies include using a zero vector, averaging previous solutions, or leveraging prior knowledge about the system.

- **Zero Vector**:

Example: For a system $Ax = b$, starting with $x = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ is a common default. This can work well for many problems but may not be optimal for every system.

- **Averaging Previous Solutions**:

Example: If past solutions were $x^{(k-1)} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$ and $x^{(k-2)} = \begin{pmatrix} 0.5 \\ 1.5 \\ 2.5 \end{pmatrix}$, an average can be $x^{(0)} = \frac{1}{2}(x^{(k-1)}+x^{(k-2)}) = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$.

- **Leveraging Prior Knowledge**:

Example: If a temperature distribution in a rod is known to stabilize around a certain value based on physical properties, starting near that temperature can lead to faster convergence.