Lecture 3
Jacobi's Method (JM)

Jinn-Liang Liu
2017/4/18

Jacobi's method is the easiest iterative method for solving a system of linear equations

$$A_{NxN}\overrightarrow{x} = \overrightarrow{b} \tag{3.1}$$

For any equation, the i^{th} equation

$$\sum_{j=1}^{N} a_{ij}x_j = b_i \tag{3.2}$$

we solve for the value x_i while assuming that the other entries of $\overrightarrow{x} = (x_1, x_2, x_3, \cdots, x_N)^T$ remain fixed and hence we obtain

$$x_i = (b_i - \sum_{\substack{j=1 \\ j\neq i}}^{N} a_{ij}x_j)/a_{ii} \tag{3.3}$$

This suggests an iterative method by

$$x_i^{(k)} = (b_i - \sum_{\substack{j=1 \\ j\neq i}}^{N} a_{ij}x_j^{(k-1)})/a_{ii} \tag{3.4}$$

where $x_i^{(k)}$ means the value of k^{th} iteration for unknown x_i with $k = 1, 2, 3, \cdots$, and $\overrightarrow{x}^{(0)}$ is an initial guess vector, e.g., we can guess that

$$\overrightarrow{x}^{(0)} = (0, 0, 0, \cdots, 0)^T \tag{3.5}$$

This is so called **Jacobi's method**. Note that the order in which the equations are examined is irrelevant.

Example 3.1. Consider the system

$$\begin{bmatrix} 3 & 2 \\ 1 & 5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix} \tag{3.6}$$

The solution is $\vec{x} = (x_1, x_2)^T = (1, 1)^T$.
Jacobi's Iteration: Let the initial guess be $x_1^{(0)} = x_2^{(0)} = 0$.

$$\begin{aligned}
k &= 1, \quad & & 3x_1 + 2x_2 = 5 \\
& & & x_1^{(1)} = (5 - 2x_2^{(0)})/3 = (5 - 2 \cdot 0)/3 = \tfrac{5}{3} \\
& & & x_1 + 5x_2 = 6 \\
& & & x_2^{(1)} = (6 - x_1^{(0)})/5 = (6 - 0)/5 = \tfrac{6}{5} \\
k &= 2, & & x_1^{(2)} = (5 - 2x_2^{(1)})/3 = (5 - 2 \cdot \tfrac{6}{5})/3 = \tfrac{13}{15} \\
& & & x_2^{(2)} = (6 - x_1^{(1)})/5 = (6 - \tfrac{5}{3})/5 = \tfrac{13}{15} \\
k &= 3, & & x_1^{(3)} = (5 - 2x_2^{(2)})/3 = (5 - 2 \cdot \tfrac{13}{15})/3 = \tfrac{49}{45} \\
& & & x_2^{(3)} = (6 - x_1^{(2)})/5 = (6 - \tfrac{13}{15})/5 = \tfrac{77}{75}
\end{aligned}$$

Table 3.1. Jacobi's Iteration

k	0	1	2	3	$\cdots$	∞
$x_1^{(k)}$	0	$\frac{5}{3}$	$\frac{13}{15}$	$\frac{49}{45}$	$\cdots$	1
$x_2^{(k)}$	0	$\frac{6}{5}$	$\frac{13}{15}$	$\frac{77}{75}$	$\cdots$	1

$\left(x_1^{(3)}, x_2^{(3)}\right) = \left(\frac{49}{45}, \frac{77}{75}\right)$ is an approximation of the exact solution $(x_1, x_2) = (1, 1)$.

Jacobi's method is highly parallel.

In matrix form, Jacobi's method can be expressed as

$$\vec{x}^{(k)} = -D^{-1}(L + U)\vec{x}^{(k-1)} + D^{-1}\vec{b}, \qquad k = 1, 2, 3, \cdots \tag{3.7}$$

where $A = D + L + U$. Here D, L, and U are the diagonal, the strictly lower-triangular, and the strictly upper-triangular parts of A, respectively.

Example 3.2.

$$\begin{aligned}
A &= \begin{bmatrix} 3 & 2 \\ 1 & 5 \end{bmatrix} \\
&= \begin{bmatrix} 3 & 0 \\ 0 & 5 \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 2 \\ 0 & 0 \end{bmatrix} \\
&= D + L + U \qquad (3.8) \\
D\vec{x}^{(1)} &= -(L+U)\vec{x}^{(0)} + \vec{b} \qquad (3.9) \\
\begin{bmatrix} 3 & 0 \\ 0 & 5 \end{bmatrix}\begin{bmatrix} x_1^{(1)} \\ x_2^{(1)} \end{bmatrix} &= \begin{bmatrix} 0 & 0 \\ -1 & 0 \end{bmatrix}\begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \end{bmatrix} + \begin{bmatrix} 0 & -2 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1^{(0)} \\ x_2^{(0)} \end{bmatrix} \\
&\quad + \begin{bmatrix} 5 \\ 6 \end{bmatrix} \qquad (3.10)
\end{aligned}$$

In general, any iterative method can be expressed as

$$\vec{x}^{(k)} = B\vec{x}^{(k-1)} + \vec{c}, \qquad k = 1, 2, 3, \cdots \tag{3.11}$$

Hence, for JM, we have

$$B = -D^{-1}(L+U)\ ,\vec{c} = D^{-1}\vec{b} \tag{3.12}$$

Algorithm JM: Jacobi's Method Solve $A\vec{x} = \vec{b}$.

Input: N: Number of unknowns and equations; a_{ij}: Entries of A, $i, j = 1 \cdots N$; b_i: Entries of $\vec{b}$, $i = 1 \cdots N$; TOL: Error Tolerance.

Output: $x_i^{(k)}$: Entries of $\vec{x}^{(k)}$(approximate solution) or Error Message.

Step 1. Choose an arbitrary initial guess $\vec{x}^{(0)} = \left(x_1^{(0)}, \cdots, x_N^{(0)}\right)^T$ to the solution $\vec{x}$.

Step 2. For $k = 1, 2, 3 \cdots, k_{\max}$

Step 3. For $i = 1, 2, \cdots, N$

Step 4. sum $= 0$ (sum represents a summation $\sum$)

Step 5. For $j = 1, 2, \cdots, i-1, i+1, \cdots, N$

Step 6. sum = sum $+ a_{ij} x_j^{(k-1)}$

Step 7. End j loop

Step 8. $x_i^{(k)} = (b_i -$ sum$)/a_{ii}$

Step 9. End i loop

Step 10. $\vec{x}^{(k)} = (x_1^{(k)}, \cdots, x_N^{(k)})^T$

Step 11. If $||\vec{r}^{(k)}||_\infty < \text{TOL} = 10^{-6}$ then Stop otherwise Set $\vec{x}^{(k-1)} = \vec{x}^{(k)}$ and Go To Step 2.

Step 12. End k loop

Step 13. Error: Not convergent with the max number of iterations $k_{\max}$ and TOL.

$\vec{x}^{(k)} \approx \vec{x}$: An approximate solution.
$\vec{r}^{(k)} = A\vec{x}^{(k)} - \vec{b}$: Residual vector.
$E^{\vec{x}} = ||\vec{r}^{(k)}||_\infty := \max_{1 \le i \le N} |r_i^{(k)}|$: Residual error in maximum norm.

Project 3.1. Consider Example 3.1 and implement the JM.

Input: $N = 2$, A, $\vec{b}$, $k_{\max}$, TOL (write the input in the program).

Output:

k	$x_1^{(k)}$	$x_2^{(k)}$
0		
1		
2		
$\vdots$		
$k_{\max}$		

Project 3.2. Consider the 1D Poisson Problem (1.1) (with $f(x) = 2$, $g_D = 0$, and $g_N = 0$) and implement the methods FDM and JM.

Input: N, A, $\vec{b}$, $k_{\max}$, TOL (write the input in the program).

Output:

N	k	$E^{\vec{x}}$	E^u	α
5				
9				
17				
33				
65				
129				

Summary: Methods for $A\vec{x} = \vec{b}$

1. Direct Methods: GE etc.

2. Iterative Methods:

(A) Stationary Iterative Methods

Neither B nor $\vec{c}$ depend upon the iteration count k in (3.11).

Eg: JM, Gauss-Seidel Method (GS),

Successive Overrelaxation (SOR) Method,

Symmetric SOR (SSOR) Method

(B) Nonstationary Iterative Method

Eg: Conjugate Gradient (CG) Method