Solution of Linear Set of Equations – 05

Dr. Serhat Hosder

Associate Professor of Aerospace Engineering

Mechanical and Aerospace Engineering

290B Toomey Hall

Missouri S&T

Rolla, MO 65409

Phone: 573-341-7239

E-mail: hosders@mst.edu

Outline

In previous lectures, we discussed direct methods for solving linear problems in which $\boldsymbol{A}$ was full. We will now turn our attention to indirect (iterative) methods.

Indirect Methods

1. Jacobi iteration
2. Gauss-Seidel iteration (Chapter 11)
3. Over/under-relaxation

Indirect (Iterative) Methods

Indirect methods use a numerical estimate to $\boldsymbol{x}$, the solution to the linear problem $\boldsymbol{Ax=b}$ and try to iteratively improve the result until a prescribed error tolerance is achieved.

Indirect Methods for discussion:

1. Jacobi Iteration
2. Gauss-Seidel Iteration
3. Over/under relaxation (simple extension to iterative method)

For a matrix to be diagonally dominant, on each row, the diagonal element has to be greater than the sum of the off-diagonal elements on that row. ***Diagonal dominance plays an important role in the theory and practical application of iterative methods***

For the matrix $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$, diagonal dominance implies

$$|a_{11}| > |a_{12}|+|a_{13}|$$
$$|a_{22}| > |a_{21}|+|a_{23}|$$
$$|a_{33}| > |a_{31}|+|a_{32}|$$

Jacobi and Gauss-Seidel Example (1)

Consider the system of equations $2x_1+x_2=3 \;\Rightarrow\; x_1=(3-x_2)/2$

$x_1-3x_2=-2 \;\Rightarrow\; x_2=(-2-x_1)/(-3)$

Re-writing in matrix form, $Ax=b$, we have $\begin{bmatrix} 2 & 1 \\ 1 & -3 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$

where $A = \begin{bmatrix} 2 & 1 \\ 1 & -3 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad b = \begin{bmatrix} 3 \\ -2 \end{bmatrix}$

The approach to the methods are described below:

Jacobi iteration	**Gauss-Seidel iteration**
$x_1^{(k+1)} = (3 - x_2^{(k)})/2$	$x_1^{(k+1)} = (3 - x_2^{(k)})/2$
$x_2^{(k+1)} = (-2 - x_1^{(k)})/(-3)$	$x_2^{(k+1)} = (-2 - x_1^{(k+1)})/(-3)$

Starting with an initial guess for x_1 and x_2, we can iteratively apply the algorithms and tabulate the results.

Jacobi and Gauss-Seidel Example (3)

Table for numerical iteration:

Jacobi iteration			Gauss-Seidel iteration		
$x_1^{(k+1)} = (3 - x_2^{(k)})/2\ ;\ x_2^{(k+1)} = (-2 - x_1^{(k)})/(-3)$			$x_1^{(k+1)} = (3 - x_2^{(k)})/2\ ;\ x_2^{(k+1)} = (-2 - x_1^{(k+1)})/(-3)$		
k	**x_1**	**x_2**	**k**	**x_1**	**x_2**
0	0	0	0	0	0
1	3/2	2/3	1	3/2	7/6
2	7/6	7/6	2	11/12	35/36
…	…	…	…	…	…

In general, Gauss-Seidel converges twice as fast as Jacobi Iteration.

However, many factors have to be considered when choosing a method including the computer architecture.

The matrix in this example is diagonally dominant. If the order of the equations are reversed, the values of x_1 and x_2 will diverge. Why? Because A is no longer diagonally dominant.

Jacobi Iteration (1)

Consider the system of equations:

$$a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1$$

$$a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2$$

$$a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + \ldots + a_{3n}x_n = b_3 \qquad Ax = b \quad \Rightarrow \quad r(x) = b - Ax = 0$$

.

$$a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n$$

Note that one can define a residual vector, with the *ith* component

$$r_i = b_i - \sum_{j=1}^{n} a_{ij}x_j, \text{ where } \{r\} \text{ is a vector with 'n' components}$$

Jacobi iteration:

$$x_i^{k+1} = \frac{b_i - \sum\limits_{\substack{j=1 \\ j \neq i}}^{n} a_{ij}x_j^k}{a_{ii}}$$

(subtract $x^k{}_i$ from both sides) $\Rightarrow x_i^{k+1} - x_i^k = \dfrac{\left[b_i - \sum\limits_{j=1}^{n} a_{ij}x_j^k \right]}{a_{ii}} = \dfrac{r_i^k}{a_{ii}}$

Jacobi Iteration (2)

The successive difference Δx^k_i is given by

$$\Delta x_i^k = x_i^{k+1} - x_i^k = \frac{\left[b_i - \sum_{j=1}^{n} a_{ij} x_j^k\right]}{a_{ii}}$$

$$\Rightarrow \Delta x_i^k = \frac{r_i^k}{a_{ii}}$$

$$\Rightarrow x_i^{k+1} = x_i^k + \Delta x_i^k$$

The iterative process is continued until a convergence criteria is satisfied. We are going to see different convergence criteria for the Jacobi and Gauss-Seidel Iterations in our next lecture.

Gauss-Seidel Iterative Algorithm

The Gauss-Seidel method differs from Jacobi iteration by always using the latest estimate for the unknown quantities. The algorithm is:

$$x_i^{k+1} = \frac{\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i+1}^{n} a_{ij} x_j^k\right]}{a_{ii}} \Rightarrow \Delta x_i^k = \frac{\left[b_i - \sum_{j=1}^{i-1} a_{ij} x_j^{k+1} - \sum_{j=i}^{n} a_{ij} x_j^k\right]}{a_{ii}}$$

Note that

$$\sum_{j=i}^{n} a_{ij} x_j^k = \sum_{j=1}^{n} a_{ij} x_j^k - \sum_{j=1}^{i-1} a_{ij} x_j^k$$

Then

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

$$\Rightarrow \Delta x_i^k = \frac{\left[r_i^k - \sum_{j=1}^{i-1} a_{ij} \Delta x_j^k\right]}{a_{ii}}$$ *(i=1,2,…n)* → *(1)*

$$\Rightarrow x_i^{k+1} = x_i^k + \Delta x_i^k$$ → *(2)*

Matrix Form of Jacobi and Gauss-Seidel Iterations (1)

It is frequently convenient to examine (and code) the matrix form of these methods. Consider a different decomposition of A

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}$$

such that ***A=D-L-U*** where

$$A = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & & 0 \\ \vdots & & \ddots & \\ 0 & & & a_{nn} \end{bmatrix} - \begin{bmatrix} 0 & \cdots\cdots & & 0 \\ -a_{21} & \ddots & & \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n1} & \cdots & -a_{nn-1} & 0 \end{bmatrix} - \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -a_{n-1n} \\ 0 & \cdots\cdots & & 0 \end{bmatrix}$$

$$\boldsymbol{A} = \boldsymbol{D} - \boldsymbol{L} - \boldsymbol{U}$$

Matrix Form of Jacobi and Gauss-Seidel Iterations (2)

The problem

$$Ax=b$$

$$(D - L - U)x=b$$

$$Dx=(L+U)x+b$$

$$D^{-1} = \begin{bmatrix} a_{11}^{-1} & 0 & .... & 0 \\ 0 & a_{22}^{-1} & & 0 \\ \vdots & & \ddots & \\ 0 & & & a_{nn}^{-1} \end{bmatrix}$$

If D^{-1} exists, then $x= D^{-1}(L+U)x + D^{-1}b$

Jacobi iteration: $x^{(k+1)} = D^{-1}(L+ U)x^{(k)} + D^{-1}b$

$$= x^{(k)} + D^{-1}r^{(k)}$$

$$\boldsymbol{\Delta x^{(k)}=D^{-1}r^{(k)}}$$

Where $r^{(k)} =b - A x^{(k)}$

Gauss-Seidel iteration: $(D - L) x^{(k+1)} = U x^{(k)} + b$

$$(D - L) (x^{(k+1)} - x^{(k)})= r^{(k)}$$

$$\boldsymbol{(D - L) \Delta x^{(k)}= r^{(k)}}$$

Solution strategy for the Matrix Form of the Gauss-Seidel Iteration

$$(D-L)\,\Delta x^k = r^k$$

$$\begin{bmatrix} a_{11} & 0 & 0 & \dots & 0 \\ a_{21} & a_{22} & 0 & \dots & 0 \\ a_{31} & a_{32} & a_{33} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & a_{n3} & \dots & a_{nn} \end{bmatrix} \begin{Bmatrix} \Delta x_1^k \\ \Delta x_2^k \\ \Delta x_3^k \\ \vdots \\ \Delta x_n^k \end{Bmatrix} = \begin{Bmatrix} r_1^k \\ r_2^k \\ r_3^k \\ \vdots \\ r_n^k \end{Bmatrix}$$

Note that ***(D-L)*** is a lower triangular matrix. Therefore we can use forward substitution to find Δx_i^k

$$\Delta x_1^k = \frac{r_1^k}{a_{11}}$$

$$\Delta x_i^k = \frac{r_i^k - \sum_{j=1}^{i-1} a_{ij}\Delta x_j^k}{a_{ii}} \qquad (i = 2,3,\dots,n)$$

Summary

In this lecture

- We developed the Jacobi and Gauss-Seidel iterative methods for solving $\boldsymbol{Ax=b}$. Talked about the importance of ordering the matrix to achieve diagonal dominance if possible.

Next lecture,

- We will define different vector norms and the convergence criteria for the iterative methods
- Learn over/under relaxation