Numerical Analysis

(ENME 602)

Spring 2021

Lecture 7

Dr. Hesham H. Ibrahim

Associate Professor, Mechatronics Department.
hesham.hamed-ahmed@guc.edu.eg
Office C7.04

Lecture 7

Linear Systems: Iterative Methods

7.1 Successive Over-Relaxation (SOR) Method

7.2 Compact Matrix Form

7.3 Optimum Relaxation Parameter

7.4 Convergence and Error Bounds

7.1 Successive Over-Relaxation (SOR) Method

7.1 Successive Over-Relaxation (SOR) Method

Example 1

A matrix **A** is diagonally dominated if, in each row, the absolute value of the entry on the diagonal is greater than the sum of the absolute values of the other entries. More compactly, **A** is diagonally dominated if

$$|A_{ii}| > \sum_{j,j\neq i} |A_{ij}| \quad \text{for all } i$$

$$\begin{aligned} 2x_1 - x_2 \phantom{{}-x_3} &= 0 \\ -x_1 + 2x_2 - x_3 &= 1 \\ - x_2 + 2x_3 &= 2 \end{aligned}$$

Jacobi-iteration $\longrightarrow$

$$\begin{aligned} x_1^k &= \frac{1}{2} x_2^{k-1} \\ x_2^k &= \frac{1}{2}\left(1 + x_1^{k-1} + x_3^{k-1}\right) \\ x_3^k &= \frac{1}{2}\left(2 + x_2^{k-1}\right) \end{aligned}$$

$$x_i^o = \frac{b_i}{a_{ii}}$$

$$x^o = (0, 0.5, 1)^t$$

$\longrightarrow$

$$\begin{aligned} x^1 &= (0.25, 1, 1.25)^t \\ x^2 &= (0.5, 1.25, 1.5)^t \\ x^3 &= (0.625, 1.5, 1.625)^t \end{aligned}$$

$\downarrow$

<u>Exact Solution</u> $x = (1, 2, 2)^t$

7.1 Successive Over-Relaxation (SOR) Method

Example 1

$$\begin{aligned} 2x_1 - x_2 \phantom{{}-x_3} &= 0 \\ -x_1 + 2x_2 - x_3 &= 1 \\ - x_2 + 2x_3 &= 2 \end{aligned}$$

Gauss-Seidel iteration $\longrightarrow$

$$x_1^k = \frac{1}{2} x_2^{k-1}$$
$$x_2^k = \frac{1}{2}\left(1 + x_1^k + x_3^{k-1}\right)$$
$$x_3^k = \frac{1}{2}\left(2 + x_2^k\right)$$

$$x^o = (0, 0.5, 1)^t \longrightarrow \begin{aligned} x^1 &= (0.25, 1.125, 1.563)^t \\ x^2 &= (0.5625, 1.5625, 1.78125)^t \end{aligned}$$

$\downarrow$

Exact Solution $x = (1, 2, 2)^t$

7.1 Successive Over-Relaxation (SOR) Method

Jacobi's iteration →

$$x_i^{(k)} = \frac{1}{a_{ii}}\left[\sum_{\substack{j=1\\ j\neq i}}^{n}\left(-a_{ij}x_j^{(k-1)}\right) + b_i\right], \qquad \text{for } i = 1, 2, \ldots, n$$

- The components of $\mathbf{x}^{(k-1)}$ are used to compute all the components $x_i^{(k)}$ of $\mathbf{x}^{(k)}$.
- But, for $i > 1$, the components $x_1^{(k)}, \ldots, x_{i-1}^{(k)}$ of $\mathbf{x}^{(k)}$ have already been computed and are expected to be better approximations to the actual solutions $x_1, \ldots, x_{i-1}$ than are $x_1^{(k-1)}, \ldots, x_{i-1}^{(k-1)}$.

Gauss-Seidel iteration →

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

for each $i = 1, 2, \ldots, n$.

7.1 Successive Over-Relaxation (SOR) Method

Jacobi's iteration $\longrightarrow$

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-a_{ij} x_j^{(k-1)} \right) + b_i \right], \qquad \text{for } i = 1, 2, \ldots, n$$

Gauss-Seidel iteration $\longrightarrow$

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

for each $i = 1, 2, \ldots, n$.

SOR - iteration $\longrightarrow$

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

- ω is a relaxation parameter
- $0 < \omega < 2$ in order to guarantee convergence
- $\omega < 1$ is called under-relaxed
- $\omega = 1$ reduces to Gauss-Seidel
- $\omega > 1$ is called Over-relaxed

7.1 Successive Over-Relaxation (SOR) Method

Example 1

$$
\begin{aligned}
2x_1 - \quad x_2 \qquad &= 0 \\
-x_1 + 2x_2 - x_3 &= 1 \\
- x_2 + 2x_3 &= 2
\end{aligned}
$$

Gauss-Seidel iteration →

$$
\begin{aligned}
x_1^k &= \frac{1}{2} x_2^{k-1} \\
x_2^k &= \frac{1}{2}\left(1 + x_1^k + x_3^{k-1}\right) \\
x_3^k &= \frac{1}{2}\left(2 + x_2^k\right)
\end{aligned}
$$

SOR iteration →

$$
\begin{aligned}
x_1^k &= (1-\omega)\, x_1^{k-1} + \omega\left(\frac{1}{2} x_2^{k-1}\right) \\
x_2^k &= (1-\omega) x_2^{k-1} + \omega\left(\frac{1}{2}\left(1 + x_1^k + x_3^{k-1}\right)\right) \\
x_3^k &= (1-\omega) x_3^{k-1} + \omega\left(\frac{1}{2}\left(2 + x_2^k\right)\right)
\end{aligned}
$$

7.1 Successive Over-Relaxation (SOR) Method

Example 1

$$
\begin{aligned}
2x_1 - \quad x_2 \qquad &= 0\\
-x_1 + 2x_2 - x_3 \ &= 1\\
- x_2 + 2x_3 \ &= 2
\end{aligned}
$$

SOR iteration →

$$
\begin{aligned}
x_1^k &= (1-\omega)\, x_1^{k-1} + \omega\left(\frac{1}{2}x_2^{k-1}\right)\\
x_2^k &= (1-\omega)x_2^{k-1} + \omega\left(\frac{1}{2}\left(1 + x_1^k + x_3^{k-1}\right)\right)\\
x_3^k &= (1-\omega)x_3^{k-1} + \omega\left(\frac{1}{2}\left(2 + x_2^k\right)\right)
\end{aligned}
$$

$x^o = (0, 0.5, 1)^t$ →

$\omega = 1.2$

$$
\begin{aligned}
x^1 &= (0.3, 1.28, 1.708)^t\\
x^2 &= (0.708, 1.829, 1.944)^t
\end{aligned}
$$

↓

<u>Exact Solution</u> $x = (1, 2, 2)^t$

7.1 Successive Over-Relaxation (SOR) Method

Example 2

- The linear system $A\mathbf{x} = \mathbf{b}$ given by

$$\begin{aligned} 4x_1 + 3x_2 \qquad &= 24 \\ 3x_1 + 4x_2 - \quad x_3 &= 30 \\ - \quad x_2 + 4x_3 &= -24 \end{aligned}$$

  has the solution $(3, 4, -5)^t$.

- Compare the iterations from the Gauss-Seidel method and the SOR method with $\omega = 1.25$ using $\mathbf{x}^{(0)} = (1, 1, 1)^t$ for both methods.

7.1 Successive Over-Relaxation (SOR) Method

Example 2

For each $k = 1, 2, \ldots,$ the equations for the Gauss-Seidel method are

$$\begin{aligned}
x_1^{(k)} &= -0.75x_2^{(k-1)} + 6 \\
x_2^{(k)} &= -0.75x_1^{(k)} + 0.25x_3^{(k-1)} + 7.5 \\
x_3^{(k)} &= 0.25x_2^{(k)} - 6
\end{aligned}$$

and the equations for the SOR method with $\omega = 1.25$ are

$$\begin{aligned}
x_1^{(k)} &= -0.25x_1^{(k-1)} - 0.9375x_2^{(k-1)} + 7.5 \\
x_2^{(k)} &= -0.9375x_1^{(k)} - 0.25x_2^{(k-1)} + 0.3125x_3^{(k-1)} + 9.375 \\
x_3^{(k)} &= 0.3125x_2^{(k)} - 0.25x_3^{(k-1)} - 7.5
\end{aligned}$$

7.1 Successive Over-Relaxation (SOR) Method

Example 2

Gauss-Seidel Iterations

k	0	1	2	3	$\cdots$	7
$x_1^{(k)}$	1	5.250000	3.1406250	3.0878906		3.0134110
$x_2^{(k)}$	1	3.812500	3.8828125	3.9267578		3.9888241
$x_3^{(k)}$	1	-5.046875	-5.0292969	-5.0183105		-5.0027940

SOR Iterations ($\omega = 1.25$)

k	0	1	2	3	$\cdots$	7
$x_1^{(k)}$	1	6.312500	2.6223145	3.1333027		3.0000498
$x_2^{(k)}$	1	3.5195313	3.9585266	4.0102646		4.0002586
$x_3^{(k)}$	1	-6.6501465	-4.6004238	-5.0966863		-5.0003486

For the iterates to be accurate to 7 decimal places,

- the Gauss-Seidel method requires 34 iterations,
- as opposed to 14 iterations for the SOR method with $\omega = 1.25$.

7.2 Compact Matrix Form

7.2 Compact Matrix Form

Jacobi's Method

- In general, iterative techniques for solving linear systems involve a process that converts the system $A\mathbf{x} = \mathbf{b}$ into an equivalent system of the form
$$\mathbf{x} = T\mathbf{x} + \mathbf{c}$$
for some fixed matrix T and vector $\mathbf{c}$.
- After the initial vector $\mathbf{x}^{(0)}$ is selected, the sequence of approximate solution vectors is generated by computing
$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$$
for each $k = 1, 2, 3, \ldots$ (reminiscent of the fixed-point iteration for solving nonlinear equations).

7.2 Compact Matrix Form

Jacobi's Method

- The Jacobi method can be written in the form

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$$

  by splitting A into its diagonal and off-diagonal parts.

- To see this, let D be the diagonal matrix whose diagonal entries are those of A, $-L$ be the strictly lower-triangular part of A, and $-U$ be the strictly upper-triangular part of A where

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

7.2 Compact Matrix Form

Jacobi's Method

We then write $A = D - L - U$ where

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & 0 \\ 0 & a_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & a_{nn} \end{bmatrix} \qquad L = \begin{bmatrix} 0 & \cdots & \cdots & 0 \\ -a_{21} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ -a_{n1} & \cdots & -a_{n,n-1} & 0 \end{bmatrix}$$

and

$$U = \begin{bmatrix} 0 & -a_{12} & \cdots & -a_{1n} \\ \vdots & \ddots & \ddots & \vdots \\ \vdots & & \ddots & -a_{n-1,n} \\ 0 & \cdots & \cdots & 0 \end{bmatrix}$$

7.2 Compact Matrix Form

Jacobi's Method

The equation $A\mathbf{x} = \mathbf{b}$, or $(D - L - U)\mathbf{x} = \mathbf{b}$, is then transformed into

$$D\mathbf{x} = (L + U)\mathbf{x} + \mathbf{b}$$

and, if D^{-1} exists, that is, if $a_{ii} \neq 0$ for each i, then

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

This results in the matrix form of the Jacobi iterative technique:

$$\mathbf{x}^{(k)} = D^{-1}(L + U)\mathbf{x}^{(k-1)} + D^{-1}\mathbf{b}, \quad k = 1, 2, \ldots$$

7.2 Compact Matrix Form

Jacobi's Method

Introducing the notation $T_j = D^{-1}(L+U)$ and $\mathbf{c}_j = D^{-1}\mathbf{b}$ gives the Jacobi technique the form

$$\mathbf{x}^{(k)} = T_j \mathbf{x}^{(k-1)} + \mathbf{c}_j$$

In practice, this form is only used for theoretical purposes while

$$x_i^{(k)} = \frac{1}{a_{ii}} \left[\sum_{\substack{j=1 \\ j \neq i}}^{n} \left(-a_{ij} x_j^{(k-1)} \right) + b_i \right], \qquad \text{for } i = 1, 2, \ldots, n$$

is used in computation.

7.2 Compact Matrix Form

Example 3

Express the Jacobi iteration method for the linear system $A\mathbf{x} = \mathbf{b}$ given by

$$
\begin{aligned}
E_1 &: & 10x_1 &- & x_2 &+ & 2x_3 & & &= 6 \\
E_2 &: & -x_1 &+ & 11x_2 &- & x_3 &+ & 3x_4 &= 25 \\
E_3 &: & 2x_1 &- & x_2 &+ & 10x_3 &- & x_4 &= -11 \\
E_4 &: & & & 3x_2 &- & x_3 &+ & 8x_4 &= 15
\end{aligned}
$$

in the form $\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$.

7.2 Compact Matrix Form

Example 3

We saw earlier that the Jacobi method for this system has the form

$$
\begin{aligned}
x_1 &= \phantom{-\frac{1}{5}x_1 +} \frac{1}{10}x_2 - \frac{1}{5}x_3 \phantom{+ \frac{1}{10}x_4} + \frac{3}{5} \\
x_2 &= \frac{1}{11}x_1 \phantom{+ \frac{1}{10}x_2} + \frac{1}{11}x_3 - \frac{3}{11}x_4 + \frac{25}{11} \\
x_3 &= -\frac{1}{5}x_1 + \frac{1}{10}x_2 \phantom{+ \frac{1}{11}x_3} + \frac{1}{10}x_4 - \frac{11}{10} \\
x_4 &= \phantom{-\frac{1}{5}x_1} - \frac{3}{8}x_2 + \frac{1}{8}x_3 \phantom{+ \frac{1}{10}x_4} + \frac{15}{8}
\end{aligned}
$$

Hence, we have

$$
T = \begin{bmatrix} 0 & \frac{1}{10} & -\frac{1}{5} & 0 \\ \frac{1}{11} & 0 & \frac{1}{11} & -\frac{3}{11} \\ -\frac{1}{5} & \frac{1}{10} & 0 & \frac{1}{10} \\ 0 & -\frac{3}{8} & \frac{1}{8} & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{c} = \begin{bmatrix} \frac{3}{5} \\ \frac{25}{11} \\ -\frac{11}{10} \\ \frac{15}{8} \end{bmatrix}
$$

$$
\mathbf{x}^{(k)} = D^{-1}(L+U)\mathbf{x}^{(k-1)} + D^{-1}\mathbf{b}, \quad k = 1, 2, \ldots
$$

7.2 Compact Matrix Form

Gauss-Seidel Method

Writing all n equations gives

$$
\begin{aligned}
a_{11}x_1^{(k)} & & &= -a_{12}x_2^{(k-1)} - a_{13}x_3^{(k-1)} - \cdots - a_{1n}x_n^{(k-1)} + b_1 \\
a_{21}x_1^{(k)} &+ a_{22}x_2^{(k)} & &= -a_{23}x_3^{(k-1)} - \cdots - a_{2n}x_n^{(k-1)} + b_2 \\
&\vdots & & \\
a_{n1}x_1^{(k)} &+ a_{n2}x_2^{(k)} + \cdots + a_{nn}x_n^{(k)} & &= b_n
\end{aligned}
$$

With the definitions of D, L, and U given previously, we have the Gauss-Seidel method represented by

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

7.2 Compact Matrix Form

Gauss-Seidel Method

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

Re-Writing the Equations (Cont'd)

Solving for $\mathbf{x}^{(k)}$ finally gives

$$\mathbf{x}^{(k)} = (D - L)^{-1}U\mathbf{x}^{(k-1)} + (D - L)^{-1}\mathbf{b}, \quad \text{for each } k = 1, 2, \ldots$$

Letting $T_g = (D - L)^{-1}U$ and $\mathbf{c}_g = (D - L)^{-1}\mathbf{b}$, gives the Gauss-Seidel technique the form

$$\mathbf{x}^{(k)} = T_g\mathbf{x}^{(k-1)} + \mathbf{c}_g$$

For the lower-triangular matrix $D - L$ to be nonsingular, it is necessary and sufficient that $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$.

7.2 Compact Matrix Form

SOR Method

The SOR Method

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

Letting

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

and

$$\mathbf{c}_\omega = \omega(D - \omega L)^{-1}\mathbf{b}$$

gives the SOR technique the form

$$\mathbf{x}^{(k)} = T_\omega \mathbf{x}^{(k-1)} + \mathbf{c}_\omega$$

7.3 Optimum Relaxation Parameter

7.3 Optimum Relaxation Parameter

- An obvious question to ask is how the appropriate value of ω is chosen when the SOR method is used?
- Although no complete answer to this question is known for the general $n \times n$ linear system, the following results can be used in certain important situations.

Theorem (Kahan)

If $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$, then $\rho(T_\omega) \geq |\omega - 1|$. This implies that the SOR method can converge only if $0 < \omega < 2$.

Theorem (Ostrowski-Reich)

If A is a positive definite matrix and $0 < \omega < 2$, then the SOR method converges for any choice of initial approximate vector $\mathbf{x}^{(0)}$.

7.3 Optimum Relaxation Parameter

Theorem (Kahan)

If $a_{ii} \neq 0$, for each $i = 1, 2, \ldots, n$, then $\rho(T_\omega) \geq |\omega - 1|$. This implies that the SOR method can converge only if $0 < \omega < 2$.

Theorem (Ostrowski-Reich)

If A is a positive definite matrix and $0 < \omega < 2$, then the SOR method converges for any choice of initial approximate vector $\mathbf{x}^{(0)}$.

Theorem

If A is positive definite and tridiagonal, then $\rho(T_g) = [\rho(T_j)]^2 < 1$, and the optimal choice of ω for the SOR method is

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}}$$

With this choice of ω, we have $\rho(T_\omega) = \omega - 1$.

7.3 Optimum Relaxation Parameter

Example 4

Find the optimal choice of ω for the SOR method for the matrix

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}$$

- This matrix is clearly tridiagonal, so we can apply the result in the SOR theorem if we can also show that it is positive definite.
- Because the matrix is symmetric, the theory tells us that it is positive definite if and only if all its leading principle submatrices has a positive determinant.
- This is easily seen to be the case because

$$\det(A) = 24, \quad \det\left(\begin{bmatrix} 4 & 3 \\ 3 & 4 \end{bmatrix}\right) = 7 \quad \text{and} \quad \det([4]) = 4$$

7.3 Optimum Relaxation Parameter

Example 4

Find the optimal choice of ω for the SOR method for the matrix

$$A = \begin{bmatrix} 4 & 3 & 0 \\ 3 & 4 & -1 \\ 0 & -1 & 4 \end{bmatrix}$$

We compute

$$\begin{aligned} T_j &= D^{-1}(L+U) \\ &= \begin{bmatrix} \frac{1}{4} & 0 & 0 \\ 0 & \frac{1}{4} & 0 \\ 0 & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} 0 & -3 & 0 \\ -3 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} 0 & -0.75 & 0 \\ -0.75 & 0 & 0.25 \\ 0 & 0.25 & 0 \end{bmatrix} \end{aligned}$$

so that

$$T_j - \lambda I = \begin{bmatrix} -\lambda & -0.75 & 0 \\ -0.75 & -\lambda & 0.25 \\ 0 & 0.25 & -\lambda \end{bmatrix}$$

7.3 Optimum Relaxation Parameter

Example 4

Therefore

$$\det(T_j - \lambda I) = \begin{vmatrix} -\lambda & -0.75 & 0 \\ -0.75 & -\lambda & 0.25 \\ 0 & 0.25 & -\lambda \end{vmatrix} = -\lambda(\lambda^2 - 0.625)$$

Thus

$$\rho(T_j) = \sqrt{0.625}$$

and

$$\omega = \frac{2}{1 + \sqrt{1 - [\rho(T_j)]^2}} = \frac{2}{1 + \sqrt{1 - 0.625}} \approx 1.24.$$

7.4 Convergence and Error Bounds

7.4 Convergence and Error Bounds

Theorem

For any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}, \quad \text{for each } k \geq 1$$

converges to the unique solution of

$$\mathbf{x} = T\mathbf{x} + \mathbf{c}$$

if and only if $\rho(T) < 1$.

7.4 Convergence and Error Bounds

Corollary

$\|T\| < 1$ for any natural matrix norm and $\mathbf{c}$ is a given vector, then the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by

$$\mathbf{x}^{(k)} = T\mathbf{x}^{(k-1)} + \mathbf{c}$$

converges, for any $\mathbf{x}^{(0)} \in \mathbb{R}^n$, to a vector $\mathbf{x} \in \mathbb{R}^n$, with $\mathbf{x} = T\mathbf{x} + \mathbf{c}$, and the following error bounds hold:

(i) $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \|T\|^k \|\mathbf{x}^{(0)} - \mathbf{x}\|$

(ii) $\|\mathbf{x} - \mathbf{x}^{(k)}\| \leq \frac{\|T\|^k}{1-\|T\|} \|\mathbf{x}^{(1)} - \mathbf{x}^{(0)}\|$

Using the Matrix Formulations

We have seen that the Jacobi and Gauss-Seidel iterative techniques can be written

$$\begin{aligned} \mathbf{x}^{(k)} &= T_j\mathbf{x}^{(k-1)} + \mathbf{c}_j \quad \text{and} \\ \mathbf{x}^{(k)} &= T_g\mathbf{x}^{(k-1)} + \mathbf{c}_g \end{aligned}$$

using the matrices

$$T_j = D^{-1}(L+U) \quad \text{and} \quad T_g = (D-L)^{-1}U$$

respectively. If $\rho(T_j)$ or $\rho(T_g)$ is less than 1, then the corresponding sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ will converge to the solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$.

7.4 Convergence and Error Bounds

The following are easily verified sufficiency conditions for convergence of the Jacobi and Gauss-Seidel methods.

Theorem

If A is strictly diagonally dominant, then for any choice of $\mathbf{x}^{(0)}$, both the Jacobi and Gauss-Seidel methods give sequences $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ that converge to the unique solution of $A\mathbf{x} = \mathbf{b}$.

Thank You

"Numerical Differentiation and Integration "