Chapter 5. Linear Systems of Equations

Section 1. Gaussian Elimination

1. Gaussian Elimination (Burden & Faires, 6.1)

Consider the $n \times n$ liner system of equations,

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ &a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ &\cdots\cdots \\ &a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{aligned}$$

or in matrix form

$$A\mathbf{x} = \mathbf{b}$$

where

$$A = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

The augmented matrix for this system is

$$[A, \mathbf{b}] = \left[\begin{array}{cccc:c} a_{11} & a_{12} & \cdots & a_{1n} & b_1 \\ a_{21} & a_{22} & \cdots & a_{2n} & b_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} & b_n \end{array}\right]$$

It is well known that the following three row operations to $[A, \mathbf{b}]$ will not change the solution of the system,

1. Interchange two rows.
2. Mutiply a row by a nonzero constant.
3. Mutiply a row by a nonzero constant and add to another row.

The Gaussian Elimination uses the three row operations to eliminate all the entries below the diagonals column by column, and then the system can be solved by back

substitution. Let

$$\tilde{A}^{(k)} = \begin{bmatrix} a_{11}^{(1)} & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1,k-1}^{(1)} & a_{1k}^{(1)} & \cdots & a_{1n}^{(1)} & \vdots & a_{1,n+1}^{(1)} \\ 0 & a_{22}^{(2)} & a_{23}^{(2)} & \cdots & a_{2,k-1}^{(2)} & a_{2k}^{(2)} & \cdots & a_{2n}^{(2)} & \vdots & a_{2,n+1}^{(2)} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ \vdots & \ddots & \ddots & \ddots & a_{k-1,k-1}^{(k-1)} & a_{k-1,k}^{(k-1)} & \cdots & a_{k-1,n}^{(k-1)} & \vdots & a_{k-1,n+1}^{(k-1)} \\ \vdots & \ddots & \ddots & \ddots & 0 & a_{k,k}^{(k)} & \cdots & a_{k,n}^{(k)} & \vdots & a_{k,n+1}^{(k)} \\ \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \cdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & \cdots & 0 & a_{n,k}^{(k)} & \cdots & a_{n,n}^{(k)} & \vdots & a_{n,n+1}^{(k)} \end{bmatrix}$$

where $\tilde{A}^{(1)} = [A, \mathbf{b}]$. For $k = 2, 3, \cdots, n$,

$$a_{ij}^{(k)} = \begin{cases} a_{ij}^{(k-1)}, & i = 1, 2, \cdots, k-1, \quad j = 1, 2, \cdots, n+1 \\ 0, & i = k, k+1, \cdots, n, \quad j = 1, 2, \cdots, k-1 \\ a_{ij}^{(k-1)} - \frac{a_{i,k-1}^{(k-1)}}{a_{k-1,k-1}^{(k-1)}} a_{k-1,j}^{(k-1)}, & i = k, k+1, \cdots, n, \quad j = k, k+1, \cdots, n+1 \end{cases}$$

$\tilde{A}^{(k)}$ represents the equivalent linear system for which the columns $1, 2, \cdots, k-1$ below the diagonals have been eliminated. The elements $a_{ii}^{(k)}, k = 1, 2 \cdots, n,\ i = 1, 2, \cdots, n$, on the diagonal are called the pivot elements. Clearly, the procedure will fail if $a_{kk}^{(k)} = 0$.

2. Operations count (Burden & Faires, 6.1)

To compare different algorithms, the excution time on computers is a very important factor. The excution time is different for different computers. Thus, we usually count the basic operations required to be excuted. In most algorithms, the Multiplication, Division, Addition and Subtraction are the main operations. Amount these, Multiplication and Division require much more computer time than Addition and Subtraction. For Gaussian Elimination, at the ith step, we have

Multiplication/Division (ith step):

$$(n-i) + (n-i)(n-i+1) = (n-i)(n-i+2)$$

Addition/Subtraction (ith step):

$$(n-i)(n-i+1)$$

Adding up we obtain the operations for elimination

Multiplication/Division:

$$\begin{aligned} \sum_{i=1}^{n-1} (n-i)(n-i+2) &= \sum_{i=1}^{n-1} (n^2 - 2ni + i^2 + 2n - 2i) \\ &= \frac{2n^3 + 3n^2 - 5n}{6} \end{aligned}$$

Addition/Subtraction:

$$\begin{aligned}\sum_{i=1}^{n-1}(n-i)(n-i+1) &= \sum_{i=1}^{n-1}(n^2-2ni+i^2+n-i)\\ &= \frac{n^3-n}{3}\end{aligned}$$

For backward substitution we have

Multiplication/Division:

$$1+\sum_{i=1}^{n-1}(n-i+1)=\frac{n^2+n}{2}$$

Addition/Subtraction:

$$\sum_{i=1}^{n-1}(n-i-1+1)=\frac{n^2-n}{2}$$

Thus, the total operation for solving the linear system is

Multiplication/Division (Total)

$$\frac{2n^3+3n^2-5n}{6}+\frac{n^2+n}{2}=\frac{n^3}{3}+n^2-\frac{n}{3}$$

Addition/Subtraction (Total)

$$\frac{n^3-n}{3}+\frac{n^2-n}{2}=\frac{n^3}{3}+\frac{n^2}{2}-\frac{5n}{6}$$

3. Pivoting (Burden & Faires, 6.2)

Partial pivoting. When the element $a_{kk}^{(k)}$ is zero, a row interchange is needed. Theoretically, the elimination procedure can continue if $a_{kk}^{(k)} \neq 0$. However, if $a_{kk}^{(k)}$ is relatively small, the elimination procedure can produce large round-off error. Thus, row interchanges are performed even when the pivit elements are not zero. To reduce the round-off error, we determine the smallest $p \geq k$ such that

$$|a_{pk}^{(k)}| = \max_{k\leq i\leq n} |a_{ik}^{(k)}|$$

and then perform the row interchange of row k and row p. This technique is called partial pivoting.

Scaled pivoting. Since any row can be multiplied by a nonzero constant, the partial pivoting may not improve the performence for some problems. The better way is to scale the rows uniformly, and then select the pivoting element. Let

$$s_i = \max_{1\leq j\leq n} |a_{ij}|$$

and then we select the smallest integer $p \geq i$ such that

$$\frac{|a_{pi}|}{s_p} = \max_{i \leq k \leq n} |a_{ki}|$$

and then perform the row interchange of row i and row p. This technique is called scaled pivoting. The scale factors $s_1, s_2, \cdots, s_n$ are computed only once before the elimination procedure. The additional operations required for scaled pivoting are

$$n(n-1) + \sum_{k=1}^{n-1} k = \frac{3}{2} n(n-1)$$

comparisons, and

$$\sum_{k=2}^{n} k = \frac{n(n+1)}{2} - 1$$

divisions. The computational time to perform a comparison is about the same as an addition/subtraction. Thus, the scaled pivoting does not add significantly to the computational time required to solve a system for large n.

Complete pivoting. The scaled pivoting generally gives satisfactory results. However, since the scale factors are computed only once, there is no guarantee that the rows are uniformly scaled at each step of the elimination. To ensure the accuracy of the solution, the complete pivoting should be used. The complete pivoting at kth step searches all the entries $a_{ij}, i = k, k+1, \cdots, n, j = k, k+1, \cdots, n$, to find the entry with the largest magnitude. Both row and column interchanges are then performed to bring this entry to the pivot position. The total comparisons are

$$\sum_{k=2}^{n} (k^2 - 1) = \frac{n(n-1)(2n+5)}{6}$$

4. Matrix factorization (Burden & Faires, 6.5)

LU Factorization

Since the three row operations are equivalent to multiply the matrix A by three elementary matrices, each step of the Gaussian elimination is equivalent to multiply A by a matrix. At the first step we have

$$M^{(1)} A\mathbf{x} = M^{(1)}\mathbf{b}$$

where

$$M^{(1)} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -m_{21} & 1 & \ddots & & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 0 \\ -m_{n1} & 0 & \cdots & 0 & 1 \end{bmatrix}$$

with

$$m_{j1} = \frac{a_{j1}^{(1)}}{a_{11}^{(1)}}$$

If the elimination process can be continued, we have

$$M^{(n-1)}M^{(n-2)}\cdots M^{(1)}A\mathbf{x} = M^{(n-1)}M^{(n-2)}\cdots M^{(1)}\mathbf{b}$$

where

$$M^{(k)} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & & & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & & & \vdots \\ \vdots & & 0 & \ddots & \ddots & & & \vdots \\ \vdots & & \vdots & -m_{k+1,k} & \ddots & \ddots & & \vdots \\ \vdots & & \vdots & \vdots & 0 & \ddots & \ddots & 0 \\ \vdots & & \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & -m_{nk} & 0 & \cdots & 0 & 1 \end{bmatrix}$$

with

$$m_{jk} = \frac{a_{jk}^{(k)}}{a_{kk}^{(k)}}$$

The matrix

$$U = M^{(n-1)}M^{(n-2)}\cdots M^{(1)}A$$

is the upper triangular matrix. Let

$$L = (M^{(1)})^{-1}(M^{(2)})^{-1}\cdots(M^{(n-2)})^{-1}(M^{(n-1)})^{-1}$$

then

$$A = LU$$

L is a lower triangular matrix. Thus, we have the following theorem,

Theorem 6.17. (LU factorization) If Gaussian elimination can be performed on the linear system $A\mathbf{x} = \mathbf{b}$ without row interchanges, then the matrix A can be factored into the product of a lower triangular matrix L and an upper triangular matrix U, i.e., $A = LU$. ∎

The advantage of the factorization is that we just need to factor the matrix A once, then any system with A as the coefficient matrix can be easily solved.

5. Stability of Gaussian elimination (Burden & Faires, 6.6)

Theorem 6.19. (Gaussian elimination) A strictly diagonally dominant matrix A is nonsingular. Moreover, in this case, Gaussian elimination can be performed on any linear system of the form $A\mathbf{x} = \mathbf{b}$ to obtain its unique solution without row or column

interchanges, and the computations will be stable with respect to the growth of round off errors. ■

Theorem 6.24. (Gaussian elimination) The symmetric matrix A is positive definite if and only if Gaussian elimination without row interchanges can be performed on the linear system $A\mathbf{x} = \mathbf{b}$ with all pivot elements positive. Moreover, in this case, the computations are stable with respect to the growth of round off errors. ■

Corollary 6.25. (LDL^t **factorization)** The matrix A is positive definite if and only if A can be factored in the form LDL^t, where L is lower triangular with 1's on its diagonal and D is a diagonal matrix with positive diagonal entries. ■

Corollary 6.26. (LL^t **factorization)** The matrix A is positive definite if and only if A can be factored in the form LL^t, where L is lower triangular with nonzero diagonal entries. ■

6. Solution of tridiagonal systems (Burden & Faires, 6.6) If

$$A = \begin{bmatrix} a_{11} & a_{12} & 0 & \cdots & \cdots & 0 \\ a_{21} & a_{22} & a_{23} & \ddots & & \vdots \\ 0 & a_{32} & a_{33} & a_{34} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & 0 & a_{n,n-1} & a_{nn} \end{bmatrix}$$

then the system $A\mathbf{x} = \mathbf{b}$ is called a tridiagonal system. In this case, the LU factorization algorithm can be simplified considerably. Let

$$L = \begin{bmatrix} l_{11} & 0 & \cdots & \cdots & 0 \\ l_{21} & l_{22} & \ddots & & \vdots \\ 0 & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & l_{n,n-1} & l_{nn} \end{bmatrix}, \quad U = \begin{bmatrix} 1 & u_{12} & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & u_{n-1,n} \\ 0 & \cdots & \cdots & 0 & 1 \end{bmatrix}$$

then $A = LU$ gives

$$\begin{aligned} a_{11} &= l_{11} \\ a_{i,i-1} &= l_{i,i-1}, \quad i = 2, 3, \cdots, n \\ a_{i,i} &= l_{i,i-1}u_{i-1,i} + l_{i,i}, \quad i = 2, 3, \cdots, n \\ a_{i,i+1} &= l_{ii}u_{i,i+1}, \quad i = 2, 3, \cdots, n-1 \end{aligned}$$

All the entries of L and U can be obtained from these equations.

Section 2. Iterative methods

1. Vector and matrix norm (Burden & Faires, 7.1)

Definition 7.1. (vector norm) A vector norm on R^n is a function, $\|\cdot\|$, from R^n to R with the following properties:

(i) $\|\mathbf{x}\| \geq 0, \quad \forall \mathbf{x} \in R^n$

(ii) $\|\mathbf{x}\| = 0$, if and only if $\mathbf{x} = 0$

(iii) $\|\alpha\mathbf{x}\| = |\alpha|\|\mathbf{x}\|, \quad \forall \alpha \in R$ and $\mathbf{x} \in R^n$

(iv) $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|, \quad \forall \mathbf{x}, \mathbf{y} \in R^n$

The commonly used norms are the following three norms:

l_1 **norm**

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|$$

l_2 **norm**

$$\|\mathbf{x}\|_2 = \left(\sum_{i=1}^{n} x_i^2 \right)^{1/2}$$

l_∞ **norm**

$$\|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|$$

Definition 7.8. (matrix norm) A matrix norm on the set of all $n \times n$ matries is a real valued function, $\|\cdot\|$, defined on this set, satisfying for all $n \times n$ matries A and B and all real numbers α:

(i) $\|A\| \geq 0$

(ii) $\|A\| = 0$, if and only if $A = 0$

(iii) $\|\alpha A\| = |\alpha|\|A\|$

(iv) $\|A + B\| \leq \|A\| + \|B\|$

(v) $\|AB\| \leq \|A\|\|B\|$

It is easy to prove the following theorem,

Theorem 7.9. (matrix norm) For a given vector norm

$$\|A\| = \max_{\|\mathbf{x}\|=1} \|A\mathbf{x}\|$$

is a matrix norm. ■

Corresponding to l_1, l_2 and l_∞ vector norms, we have the matrix norms

$$\|A\|_1 = \max_{\|\mathbf{x}\|_1=1} \|A\mathbf{x}\|_1$$
$$\|A\|_2 = \max_{\|\mathbf{x}\|_2=1} \|A\mathbf{x}\|_2$$
$$\|A\|_\infty = \max_{\|\mathbf{x}\|_\infty=1} \|A\mathbf{x}\|_\infty$$

Theorem 7.11. (matrix norm) If $A = (a_{ij})$ is an $n \times n$ matrix, then

$$\|A\|_1 = \max_{1\le j\le n} \sum_{i=1}^{n} |a_{ij}|$$
$$\|A\|_\infty = \max_{1\le i\le n} \sum_{j=1}^{n} |a_{ij}|$$

Theorem 7.14. (matrix norm) If $A = (a_{ij})$ is an $n \times n$ matrix, then

$$\|A\|_2 = [\rho(A^t A)]^{1/2}$$

where $\rho(A)$ is the spectral radius of A, i.e., the largest eigenvalue of A in norm:

$$\rho(A) = \max |\lambda| \quad ■$$

2. Jacobi iteration (Burden & Faires, 7.3)

Consider the $n \times n$ liner system of equations,

$$\begin{aligned}
&a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1\\
&a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2\\
&\cdots\cdots\\
&a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n
\end{aligned}$$

or $A\mathbf{x} = \mathbf{b}$. Suppose $a_{ii} \neq 0$, solving for $x_1, x_2, \cdots, x_n$ from each equation we obtain

$$x_1 = \frac{1}{a_{11}}(-a_{12}x_2 - \cdots - a_{1n}x_n + b_1) \tag{1}$$
$$x_2 = \frac{1}{a_{22}}(-a_{21}x_1 - \cdots - a_{2n}x_n + b_2) \tag{2}$$
$$\cdots\cdots \tag{3}$$
$$x_n = \frac{1}{a_{nn}}(-a_{n1}x_1 - \cdots - a_{n,n-1}x_{n-1} + b_n) \tag{4}$$

Suppose an initial approximation $\mathbf{x}^{(0)}$ to the solution $\mathbf{x}$ is given, then we can form an iterative procedure,

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

for $k = 1, 2, \cdots,$. Let $A = D - L - U$ with D, $-L$ and $-U$ denoting the diagonal, lower triangular and upper triangular parts of the matrix A,

$$D = \begin{bmatrix} a_{11} & 0 & \cdots & \cdots & 0 \\ 0 & a_{22} & \ddots & & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots \\ 0 & \cdots & \cdots & 0 & a_{nn} \end{bmatrix}$$

$$-L = \begin{bmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ -a_{21} & \ddots & & & \vdots \\ \vdots & & \ddots & \ddots & \vdots \\ -a_{n1} & \cdots & \cdots & -a_{n,n-1} & 0 \end{bmatrix} \qquad -U = \begin{bmatrix} 0 & -a_{12} & \cdots & \cdots & -a_{1n} \\ \vdots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & a_{n-1,n} \\ 0 & \cdots & \cdots & \cdots & 0 \end{bmatrix}$$

Then equations (**??**)-(**??**) can be written as

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

The iterative procedure (**??**)-(**??**) can be written as

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

It is easy to see that if the iteration coverges, then the limit is the solution $\mathbf{x}$.

3. Gauss-Seidel iteration (Burden & Faires, 7.3) In the Jacobi iteration, when we compute $x_i^{(k)}$, all values $x_1^{(k)}, x_2^{(k)}, \cdots, x_{i-1}^{(k)}$ have been computed. Thus, we may use these new values instead of the old values. This improvement gives the Gauss-Seidel iteration,

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

In matrix form,

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

or

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

4. SOR iteration (Burden & Faires, 7.3) The SOR (**Successive Over-Relaxation**) iteration is a further improvement on the Gauss-Seidel iteration. At each step of the iteration, the SOR value takes the linear combination of Gauss-Seidel value and the value from the previous iteration, i.e.,

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

for $i = 1, 2, \cdots, n$. In matrix form,

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

or

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

When $0 < \omega < 1$, it is called **under-relaxation**. When $1 < \omega < 2$, it is called **over-relaxation**.

4. Convergence of the iterative methods (Burden & Faires, 7.3)

Theorem 7.19. (General case) For any $\mathbf{x}^{(0)} \in R^n$, the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by

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

converges to the unique solution of $\mathbf{x} = T\mathbf{x} + \mathbf{c}$ if and only if $\rho(T) < 1$.

Corollary 7.20. (General case) If $\|T\| < 1$ for any matrix norm and $\mathbf{c}$ is a given vector, then the sequence $\{\mathbf{x}^{(k)}\}_{k=0}^{\infty}$ defined by

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

converges, for any $\mathbf{x}^{(0)} \in R^n$, to a vector $\mathbf{x} \in R^n$, and the following error bounds hold,

(i)

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

(ii)

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

Theorem 7.21. (Jacobi and Gauss-Seidel) If A is strictly diagonally dominant, then for any choice of $\mathbf{x}^{(0)}$, both Jacobi and Gauss-Seidel methods converge.

Theorem 7.24. (SOR) If $a_{ii} \neq 0$, for each $i = 1, 2, \cdots, n$, then $\rho(T_\omega) \geq |\omega - 1|$. this implies that the SOR method can converge only if $0 < \omega < 2$.

Theorem 7.25. (SOR) If A is positive definite and $0 < \omega < 2$, then the SOR method converges for any choice of $\mathbf{x}^{(0)}$.

Section 3. Error analysis and condition number

Suppose $\tilde{\mathbf{x}}$ is an approximation to the solution $\mathbf{x}$ of $A\mathbf{x} = \mathbf{b}$. Let

$$\mathbf{r} = \mathbf{b} - A\tilde{\mathbf{x}}$$

If $\|\mathbf{r}\|$ is small, it seems natural to say that $\tilde{\mathbf{x}}$ is an good approximation to $\mathbf{x}$. However, for some systems this is not true.

Theorem 7.27. (Error bounds) Suppose that $\tilde{\mathbf{x}}$ is an approximation to the solution of $A\mathbf{x} = \mathbf{b}$. A is a nonsingular matrix, and $\mathbf{r}$ is the residual vector for $\tilde{\mathbf{x}}$. Then for any norm

$$\|\mathbf{x} - \tilde{\mathbf{x}}\| \leq \|\mathbf{r}\| \cdot \left\|A^{-1}\right\|$$

and if $\mathbf{x} \neq 0$ and $\mathbf{b} \neq 0$,

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \|A\| \cdot \left\|A^{-1}\right\| \frac{\|\mathbf{r}\|}{\|\mathbf{b}\|}$$

Definition 7.28. (Condition number) The condition number of the nonsigular matrix A relative to a norm $\|\cdot\|$ is

$$K(A) = \|A\| \cdot \left\|A^{-1}\right\| \quad \blacksquare$$

In the real applications, the matrix A and the right hand side $\mathbf{b}$ are usually not exact, but with small perturbations. So the system

$$A\mathbf{x} = \mathbf{b}$$

becomes

$$(A + \delta A)\tilde{\mathbf{x}} = \mathbf{b} + \delta\mathbf{b}$$

Theorem 7.29. (Error bounds) Suppose A is nonsingular and

$$\|\delta A\| < \frac{1}{\left\|A^{-1}\right\|}$$

then

$$\frac{\|\mathbf{x} - \tilde{\mathbf{x}}\|}{\|\mathbf{x}\|} \leq \frac{K(A)\|A\|}{\|A\| - K(A)\|\delta A\|} \left(\frac{\|\delta\mathbf{b}\|}{\|\mathbf{b}\|} + \frac{\|\delta A\|}{\|A\|}\right)$$