4. Solution of Linear Equation Systems

NUMERICAL ANALYSIS. Prof. Y. Nishidate (323-B, nisidate@u-aizu.ac.jp)
http://web-int.u-aizu.ac.jp/∼nisidate/na/

Vectors and Matrices

Matrix is represented with a rectangular table of numbers. *Vector* is a column matrix. Matrix and vector representations are shown below:

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{m1} & a_{m2} & ... & a_{mn} \end{bmatrix} \qquad \mathbf{v} = \begin{Bmatrix} v_1 \\ v_2 \\ ... \\ v_n \end{Bmatrix}$$

Common operations defined on vector and matrices are summarized in the following equations.
Vector sum:

$$\mathbf{w} = \mathbf{u} + \mathbf{v} \qquad w_i = u_i + v_i \qquad i = 1..n$$

The product of a vector by a number:

$$\mathbf{w} = \alpha\mathbf{v} \qquad w_i = \alpha v_i \qquad i = 1..n$$

The scalar product of two vectors:

$$s = \mathbf{u} \cdot \mathbf{v} \qquad s = \sum u_i v_i \qquad i = 1..n$$

The norm of a vector:

$$|\mathbf{v}| = \sqrt{\mathbf{v} \cdot \mathbf{v}}$$

The sun of two matrices of the same dimensions

$$\mathbf{C} = \mathbf{A} + \mathbf{B} \qquad c_{ij} = a_{ij} + b_{ij} \qquad i = 1..m \;\; j = 1..n$$

The product of a matrix by a number:

$$\mathbf{B} = \alpha\mathbf{A} \qquad b_{ij} = \alpha a_{ij} \qquad i = 1..m \;\; j = 1..n$$

The transpose of a matrix:

$$\mathbf{B} = \mathbf{A}^T \qquad b_{ij} = a_{ji} \qquad i = 1..m \;\; j = 1..n$$

The product of a matrix with a vector:

$$\mathbf{u} = \mathbf{A}\mathbf{v} \qquad u_i = \sum a_{ij} v_j \qquad i = 1..m \;\; j = 1..n$$

The product of a $m \times p$ matrix with a $p \times n$ matrix:

$$\mathbf{C} = \mathbf{A}\mathbf{B} \qquad c_{ij} = \sum_{k=1}^{p} a_{ik} b_{kj} \qquad i = 1..m \;\; j = 1..n$$

An *identity* matrix $\mathbf{I}$ is a square matrix such that

$$\mathbf{I}\mathbf{v} = \mathbf{v} \qquad \mathbf{I}\mathbf{A} = \mathbf{A}\mathbf{I} = \mathbf{A}$$

A *symmetrical* matrix is a matrix such that

$$\mathbf{A}^T = \mathbf{A} \qquad a_{ij} = a_{ji}$$

System of Linear Equation

A system of linear algebraic equations

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

can be represented in matrix-vector notation as

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

Gaussian Elimination

Solution of a linear equation system using Gaussian elimination consists of: a) forward elimination, and b) backward substitution. For simplicity, let us apply Gaussian elimination to a problem with no null coefficients. The forward elimination proceeds as follows.

The first equation times a_{21}/a_{11} is subtracted from the second equation to eliminate the first unknown from the second equation. In the same way, the first term of every equation $i > 2$ is eliminated by subtracting the first equation times a_{i1}/a_{11}. Then, the equation system looks like

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &= b_1 \\ a_{22}^{(1)}x_2 + ... + a_{2n}^{(1)}x_n &= b_2^{(1)} \\ ... &= ... \\ a_{n2}^{(1)}x_2 + ... + a_{nn}^{(1)}x_n &= b_n^{(1)} \end{aligned}$$

where $a_{ij}^{(1)} = a_{ij} - (a_{i1}/a_{11})a_{1j}$.
Next, the second term of each equation in the third through the last equation, $i > 2$, is eliminated by subtracting the second equation multiplied by $m_{i2} = a_{i2}^{(1)}/a_{22}^{(1)}$. After $(n-1)$ steps, the forward elimination process is finished and the set of equations has the following appearance:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n &= b_1 \\ a_{22}^{(1)}x_2 + ... + a_{2n}^{(1)}x_n &= b_2^{(1)} \\ ... &= ... \\ a_{nn}^{(n-1)}x_n &= b_n^{(n-1)} \end{aligned}$$

The backward substitution procedure starts with the last equation. The solution for $x_n$is obtained from the last equation:

$$x_n = b_n^{(n-1)}/a_{nn}^{(n-1)}$$

Subsequently,

$$\begin{aligned} & x_{n-1} = (b_{n-1}^{(n-2)} - a_{n-1\,n}^{(n-2)}x_n)/a_{n-1\,n-1}^{(n-2)} \\ & ... \\ & x_1 = \left(b_1 - \sum_{j=2}^{n} a_{1j}x_j\right)/a_{11} \end{aligned}$$

Thus, all unknowns are determined and the solution process is completed.

Algorithm of the Gaussian elimination solution procedure for the equation system $\mathbf{A}\mathbf{x} = \mathbf{b}$:

Forward elimination

for $i = 1$ **to** $n-1$
  for $j = i+1$ **to** n
    $r = a_{ji}/a_{ii}$
    $a_{ji} = 0$
    for $k = i+1$ **to** n
      $a_{jk} = a_{jk} - r a_{ik}$
    end for
    $b_j = b_j - r b_i$
  end for
end for

Back substitution

for $i = n$ **down to** 1
  $x_i = b_i/a_{ii}$
  $a_{ii} = 1$
  for $j = i-1$ **down to** 1
    $b_j = b_j - a_{ji}x_i$
    $a_{ji} = 0$
  end for
end for

The operation count for the Gaussian elimination is defined by the modification of a_{ik} coefficients inside triple loop and it is equal to $2n^3/3$ Multiply-Add operations.

Gaussian Elimination and Pivoting

Definition. The number a_{pp} that is used to eliminate x_p in rows $p+1, p+2, \ldots, n$ during forward Gaussian elimination is called the pth *pivotal element* and the pth row is called the *pivotal row*.

Pivoting to avoid $a_{pp}^{(p)} = 0$. If $a_{pp}^{(p)} = 0$ then row p cannot be used to eliminate the elements in column p below the diagonal. It is necessary to find row k where $a_{kp}^{(p)} \neq 0$ and $k > p$ and then to interchange row p and row k so that a nonzero pivot element is obtained. This process is called *pivoting*.

Pivoting to reduce error. If there is more than one nonzero element in column p that lies on or below the diagonal of the matrix, there is a choice which rows to interchange. To reduce the propagation of error, it is suggested to check the values of all matrix elements in column p that lie on or below the diagonal, and to select the row k in which the element $a_{kp}^{(p)}$ has the largest absolute value. The row k becomes pivotal row after interchange of row p with row k.

LU Decomposition

The nonsingular matrix $\mathbf{A}$ has a *triangular factorization* if it can be expressed as the product of a lower-triangular matrix $\mathbf{L}$ and an upper-triangular matrix $\mathbf{U}$:

$$\mathbf{A} = \mathbf{LU}$$

Triangular factorization for the matrix 4×4 is illustrated below.

$$A = \begin{bmatrix} m_{11} & 0 & 0 & 0 \\ m_{21} & m_{22} & 0 & 0 \\ m_{31} & m_{32} & m_{33} & 0 \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix} \begin{bmatrix} 1 & u_{12} & u_{13} & u_{14} \\ 0 & 1 & u_{23} & u_{24} \\ 0 & 0 & 1 & u_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Solution of a factorized linear system. If the coefficient matrix $\mathbf{A}$ for the linear system $\mathbf{Ax} = \mathbf{b}$ has a triangular factorization $\mathbf{A} = \mathbf{LU}$ then the solution to

$$\mathbf{LUx} = \mathbf{b}$$

can be obtained by defining $\mathbf{y} = \mathbf{Ux}$ and then by solving two systems:

$$\begin{aligned} \mathbf{Ly} &= \mathbf{b} \\ \mathbf{Ux} &= \mathbf{y} \end{aligned}$$

Triangular factorization. The general formula for computing coefficients of $\mathbf{L}$ and $\mathbf{U}$ matrices can be written as follows:

$$m_{ij} = a_{ij} - \sum_{k=1}^{j-1} m_{ik} u_{kj} \qquad j \leq i, \quad i = 1,\ 2,\ \ldots,\ n$$

$$u_{ij} = \left(a_{ij} - \sum_{k=1}^{i-1} m_{ik} u_{kj} \right) / m_{ii} \qquad i \leq j, \quad j = 2,\ 3,\ \ldots,\ n$$

Note that for $j = 1$ the coefficients for $\mathbf{L}$ are equal to
$m_{i1} = a_{i1}$
and for $i = 1$ the coefficients for $\mathbf{U}$ are equal to
$u_{1j} = a_{1j}/m_{11} = a_{1j}/a_{11}$.

Right-hand side reduction and back-substitution. The general equation for the reduction of the right-hand side $\mathbf{b}$ is

$$\begin{aligned} b_1' &= b_1/m_{11} \\ b_i' &= \left(b_i - \textstyle\sum_{k=1}^{i-1} m_{ik} b_k' \right) / m_{ii} \qquad i = 2,\ 3,\ \ldots,\ n \end{aligned}$$

The back-substitution procedure is described by the following relations

$$\begin{aligned} x_n &= b_n' \\ x_j &= b_j' - \textstyle\sum_{k=j+1}^{n} u_{jk} x_k \qquad j = n-1,\ n-2,\ \ldots,\ 1 \end{aligned}$$

Algorithm of the triangular decomposition of the matrix $\mathbf{A}$ into lower-triangular matrix $\mathbf{L}$ and an upper-triangular matrix $\mathbf{U}$:

```
for i = 1 to n
    m_i1 = a_i1
end for
for j = 1 to n
    u_1j = a_1j / m_11
end for
for j = 2 to n
    for i = j to n
        s = 0
        for k = 1 to j - 1
            s = s + m_ik u_kj
        end for
        m_ij = a_ij - s
    end for
    u_jj = 1
    for i = j + 1 to n
        s = 0
        for k = 1 to j - 1
            s = s + m_jk u_ki
        end for
        u_ji = (a_ji - s) / m_jj
    end for
end for
```

Algorithm of right-hand side reduction and back-substitution for a given right hand side vector $\mathbf{b}$:

```
for i = 1 to n
    s = 0
    for k = 1 to i - 1
        s = s + m_ik b_k
    end for
    b_i = (b_i - s) / m_ii
end for
x_n = b_n
for j = n - 1 to 1
    s = 0
    for k = j + 1 to n
        s = s + u_jk x_k
    end for
    x_j = b_j - s
end for
```

Computer implementation of LU decomposition. The reason the LU decomposition method is popular in programming is that storage space may be economized. There is no need to store the zeros in either $\mathbf{L}$ or $\mathbf{U}$, and the ones on the diagonal of $\mathbf{U}$ can also be omitted (since these values are always the same). One can store the essential coefficients of $\mathbf{L}$ and $\mathbf{U}$ in place of coefficients of matrix $\mathbf{A}$. This is possible because of the fact that, after any coefficient a_{ij} is once used, it never again appears in the calculations.

The matrix of the equation system can be transformed by the LU decomposition as follows

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix} \rightarrow \begin{bmatrix} m_{11} & u_{12} & u_{13} & u_{14} \\ m_{21} & m_{22} & u_{23} & u_{24} \\ m_{31} & m_{32} & m_{33} & u_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{bmatrix}$$

System ill-conditioning.

An equation system $\mathbf{Ax} = \mathbf{b}$ is called *ill-conditioned* if small changes in the coefficients of $\mathbf{A}$ or $\mathbf{b}$ will produce large changes in the solution $\mathbf{x}$.

In the case of ill-conditioning, numerical methods may produce system solution with considerable errors.

The system ill-conditioning occurs when $\mathbf{A}$ is "nearly singular" and the determinant of $\mathbf{A}$ is cloze to zero. For example, the system of two equations is ill-conditioned if two lines represented by equations are nearly parallel.