Linear Algebraic Systems

Lecture 1
Direct methods: Gaussian method and LU decomposition method

1. Example taken from mechanics

Why should we study linear algebraic systems?

Let us consider a one-dimensional array of three types of particles interacting by stiff springs

$$\begin{cases} m_1\ddot{u}_n = -k(v_n + w_{n-1} - 2u_n) - f_n^{(1)}(t), \\ m_2\ddot{v}_n = -k(w_n + u_n - 2v_n) - f_n^{(2)}(t), \\ m_3\ddot{w}_n = -k(u_{n+1} + v_n - 2w_n) - f_n^{(3)}(t), \end{cases} \tag{1}$$

where $u_n,\ \ v_n,\ \ w_n$ are the particles' vertical displacements, $m_1,\ \ m_2,\ \ m_3$ are their masses, k is the common stiffness of springs, $f_n^{(1)},\ \ f_n^{(2)},\ \ f_n^{(3)}$ are external forces and $n = -\infty, .., -2, -1, 0, 1, 2, ..., \infty$. We assume that the time dependence of external forces has the form $\exp(-i\omega t)$. Due to the translation symmetry we suppose

$$f_n^{(j)}(t) = f_j \exp(-i\omega t + iKLn)$$

where $j = 1, 2, 3,$ K is the "quasimomentum" - a free parameter, L is a period of the array. We seek a solution to the system in the form

$$\begin{cases} u_n = u\exp(-i\omega t + iKLn), \\ v_n = v\exp(-i\omega t + iKLn), \\ w_n = w\exp(-i\omega t + ikLn), \end{cases} \tag{2}$$

Substituting these formulae into the system, we obtain a 3×3 system of linear algebraic inhomogeneous equations for unknowns $u, \quad v, \quad w$

$$\begin{pmatrix} m_1\omega^2 + 2k & k & ke^{-iKL} \\ k & m_2\omega^2 + 2k & k \\ ke^{iKL} & k & m_3\omega^2 + 2k \end{pmatrix} \begin{pmatrix} u \\ v \\ w \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix}. \tag{3}$$

2. Kramer's method

Consider first a system of three linear algebraic equations with three unknowns

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 = f_1, \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 = f_2, \\ a_{31}x_1 + a_{32}x_2 + a_{33}x_3 = f_3. \end{cases} \tag{4}$$

This can be solved by Kramer's method

$$x_1 = \frac{\det A_1}{\det A}, \qquad x_2 = \frac{\det A_2}{\det A}, \qquad x_3 = \frac{\det A_3}{\det A}, \tag{5}$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \quad A_1 = \begin{pmatrix} f_1 & a_{12} & a_{13} \\ f_2 & a_{22} & a_{23} \\ f_3 & a_{32} & a_{33} \end{pmatrix}, \quad A_2 = \begin{pmatrix} a_{11} & f_1 & a_{13} \\ a_{21} & f_2 & a_{23} \\ a_{32} & f_3 & a_{33} \end{pmatrix},$$

$$A_3 = \begin{pmatrix} a_{11} & a_{12} & f_1 \\ a_{21} & a_{22} & f_2 \\ a_{31} & a_{32} & f_3 \end{pmatrix}.$$

2. Example of application of Gaussian method

We can also use the Gaussian method of successive elimination of unknowns. Let us consider the example

$$\begin{cases} x_1 + x_2 + 3x_4 = 4, \\ 2x_1 + x_2 - x_3 + x_4 = 1, \\ 3x_1 - x_2 - x_3 + 2x_4 = -3, \\ -x_1 + 2x_2 + 3x_3 - x_4 = 4, \end{cases}$$

and solve this system by the Gaussian method. Direct elimination yields

$$\left(\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 2 & 1 & -1 & 1 & 1 \\ 3 & -1 & -1 & 2 & -3 \\ -1 & 2 & 3 & -1 & 4 \end{array}\right), \quad \left(\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & -4 & -1 & -7 & -15 \\ 0 & 3 & 3 & 2 & 8 \end{array}\right),$$

$$\left(\begin{array}{cccc|c} 1 & 1 & 0 & 3 & 4 \\ 0 & -1 & -1 & -5 & -7 \\ 0 & 0 & 3 & 13 & 13 \\ 0 & 0 & 0 & -13 & -13 \end{array}\right).$$

Now consider the backward substitutions

$$-13x_4 = -13, \quad \Rightarrow x_4 = 1, \quad 3x_3 + 13x_4 = 13 \quad \Rightarrow x_3 = 0,$$

$$-x_2 - x_3 - 5x_4 = -7, \quad \Rightarrow x_2 = 2, \quad x_1 + x_2 + 3x_4 = 4 \quad \Rightarrow x_1 = -1.$$

3. Gaussian method in general case

Consider a system of linear algebraic equations

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = f_1, \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = f_2, \\ ...., \\ a_{n1}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = f_n, \end{cases} \tag{6}$$

which in matrix form looks like this

$$\begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{n1} & a_{n2} & ... & a_{nn} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ ... \\ x_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ ... \\ f_n \end{pmatrix}. \tag{7}$$

Here $A =\| a_{ij} \|$ is the matrix with $i, j = 1, .., n$, $x = (x_1, x_2, ..., x_n)^T$ and $f = (f_1, f_2, ..., f_n)$ are vectors. The vector x is unknown. We assume that $\det A \neq 0$.

If n is large Kramer's method requires a lot of time to perform calculations, as the approximate number of operations is of order $n!$. On the contrary the Gaussian method is much faster as it needs approximately $n^3/3$ operations. For example, if $n = 10$, then $10! = 3628800$ in comparison with $10^3/3 \approx 333$. Thus the Gaussian method is more efficient in numerical analysis.

After n steps of performing direct elimination of the unknowns we obtain a system with an upper triangular matrix U

$$\begin{pmatrix} 1 & u_{12} & u_{13} & ... & u_{1n} \\ 0 & 1 & u_{23} & ... & u_{2n} \\ 0 & 0 & 1 & ... & u_{2n} \\ ... & ... & ... & ... & ... \\ 0 & 0 & 0 & ... & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ ... \\ x_n \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ ... \\ y_n \end{pmatrix}, \qquad (8)$$

or in a short form $Ux = y$.

What is going on after we have made $k-1$ steps? We have a block of equations

$$\begin{cases} a_{kk}^{(k-1)} x_k + a_{k,k+1}^{(k-1)} x_{k+1} + ... + a_{kn}^{(k-1)} x_n = f_k^{(k-1)}, \\ a_{k+1,k}^{(k-1)} x_k + a_{k+1,k+1}^{(k-1)} x_{k+1} + ... + a_{k+1,n}^{(k-1)} x_n = f_{k+1}^{(k-1)}, \\ ...., \\ a_{n,k}^{(k-1)} x_k + a_{n,k+1}^{(k-1)} x_{k+1} + ... + a_{nn}^{(k-1)} x_n = f_n^{(k-1)}. \end{cases}$$

Making the next step we obtain

$$\begin{cases} x_k + u_{k,k+1} x_{k+1} + ... + u_{kn} x_n = y_k, \\ 0 \ + a_{k+1,k+1}^{(k)} x_{k+1} + ... + a_{k+1,n}^{(k)} x_n = f_{k+1}^{(k)}, \\ ...., \\ 0 \ + a_{n,k+1}^{(k)} x_{k+1} + ... + a_{nn}^{(k)} x_n = f_n^{(k)}. \end{cases} \qquad (9)$$

where

$$u_{kj} = \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}, \qquad y_k = \frac{f_k^{(k-1)}}{a_{kk}^{(k-1)}}, \qquad (10)$$

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - a_{ik}^{(k-1)} u_{kj}, \qquad i,j = k+1, ..., n, \qquad (11)$$

$$f_i^{(k)} = f_i^{(k-1)} - a_{ik}^{(k-1)} y_k, \qquad (12)$$

and $a_{1j}^{(0)} = a_{1j}$, $f_1^{(0)} = f_1$.

These formulae represent the first part of the algorithm. This is the forward elimination of the unknowns. The unknowns can be easily found from this system of backward substitutions

$$x_i = y_i - \sum_{j=i+1}^{n} u_{ij} x_j. \tag{13}$$

for $1 \leq i \leq n-1$, if $i = n$, we have $x_n = y_n$. These formulae provide the second part of the algorithm. Using this algorithm we can build up the numerical code.

Conclusion

The Gaussian method is simple, fast and stable if $\det A \neq 0$. However, the Gaussian method works if and only if the leading element $a_{kk}^{(k)} \neq 0$ at every step $k = 1, ..., n$ of the forward elimination.

4. Tridiagonal systems

Consider a 4×4 system

$$\begin{pmatrix} b_1 & c_1 & 0 & 0 \\ a_1 & b_2 & c_2 & 0 \\ 0 & a_2 & b_3 & c_3 \\ 0 & 0 & a_3 & b_4 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}. \tag{14}$$

It is called a tridiagonal system. Tridiagonal systems give rise to a particular simple results on the Gaussian elimination Let us apply the Gaussian method to solve the system. Forward elimination at each step yields a system of the following general form

$$\begin{pmatrix} 1 & c_1/d_1 & 0 & 0 \\ 0 & 1 & c_2/d_2 & 0 \\ 0 & 0 & 1 & c_3/d_3 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{pmatrix}, \tag{15}$$

where

At the first step $d_1 = b_1, \quad y_1 = f_1/d_1,$

At the second step $d_2 = b_2 - a_1c_1/d_1, \quad y_2 = (f_1 - y_1a_1)/d_2,$

At the third step $d_3 = b_3 - a_2c_2/d_2, \quad y_3 = (f_3 - y_2a_2)/d_3,$

At the fourth step $d_4 = b_4 - a_3c_3/d_3, \quad y_4 = (f_4 - y_3a_3)/d_4.$

Finally, the backward substitution procedure gives the answer

$$\begin{cases} x_4 = y_4, \\ x_3 = y_3 - x_4c_3/d_3, \\ x_2 = y_2 - x_3c_2/d_2, , \\ x_1 = y_1 - x_2c_1/d_1. \end{cases}$$

It is clear what will happen in general case of $n \times n$ tridiagonal system. The forward elimination procedure is

At the first step $d_1 = b_1, \quad y_1 = f_1/d_1,$

At the k-th step $d_k = b_k - a_{k-1}c_{k-1}/d_{k-1}, \quad y_k = (f_k - y_{k-1}a_{k-1})/d_k.$

The backward substitutions then look like this

$x_n = y_n, \qquad x_{k-1} = y_{k-1} - x_kc_{k-1}/d_{k-1}.$

5. LU decomposition method

Let us study the system $Ax = f$. As a result of applying the Gaussian method we obtained the system $Ux = y$, where U is the upper triangular matrix with unities on the main diagonal. Vector y is only dependant on vector f. Let us study a relation between y and f. In the case $n = 3$ we have

$$\begin{cases} y_1 = f_1/a_{11}, \\ y_2 = (f_2 - y_1 a_{21})/a_{22}^{(1)}, \\ y_3 = (f_3 - y_1 a_{31} - y_2 a_{32}^{(1)})/a_{33}^{(2)}. \end{cases}$$

Hence

$$\begin{cases} f_1 = a_{11} y_1, \\ f_2 = y_1 a_{21} + y_2 a_{22}^{(1)}, \\ f_3 = y_1 a_{31} + y_2 a_{32}^{(1)} + y_3 a_{33}^{(2)}. \end{cases}$$

It is clear that for arbitrary n we shall have

$$\begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ ... \\ f_n \end{pmatrix} = \begin{pmatrix} a_{11} & 0 & 0 & ... & 0 \\ a_{21} & a_{22}^{(1)} & 0 & ... & 0 \\ a_{31} & a_{32}^{(1)} & a_{33}^{(2)} & ... & 0 \\ ... & ... & ... & ... & \\ a_{n1} & a_{n2}^{(1)} & a_{n3}^{(2)} & ... & a_{nn}^{(n-1)} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ y_3 \\ ... \\ y_n \end{pmatrix}. \tag{16}$$

Thus, $f = Ly$, where L is a lower triangular matrix with elements

$$l_{ij} = \begin{cases} a_{ij}^{(j-1)}, & i \geq j, \\ 0, & i < j. \end{cases}$$

which can be calculated using the formulae of basic algorithm (10)-(12). As $Ax = f, \quad Ux = y, \quad Ly = f$ we get $Ax = Ly = LUx$. Hence, we deduce $A = LU$, where L is lower triangular matrix and U is upper triangular matrix. This representation is called LU decomposition. This is example for 3×3 matrix

$$\begin{pmatrix} 60 & 30 & 20 \\ 30 & 20 & 15 \\ 20 & 15 & 12 \end{pmatrix} = \begin{pmatrix} 60 & 0 & 0 \\ 30 & 5 & 0 \\ 20 & 5 & 1/3 \end{pmatrix} \begin{pmatrix} 1 & 1/2 & 1/3 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Suppose that for the system $Ax = f$ we know matrices L and U. Then we can easily solve the system in two easier steps: first $Ly = f$ solve for y, second $Ux = y$ solve for x.

The problem is, of course to obtain the matrices L and U. A has n^2 coefficients, while there are $n(n+1)$ coefficients in the two matrices L and U, so there is some redundancy. This gives us some freedom to choose some of the coefficients in L or U.

We mention some particular choices below:

• If the Gaussian elimination method implicitly chooses $l_{ii} = 1$, this is known as *Doolittle's method.*

• $u_{ii} = 1$ is known as *Crout's method.*

• If $A = A^T$ (i.e A is symmetric) and A is also positive definite (i.e has positive eigenvalues), then we can choose $u_{ij} = l_{ji}$. This is known as *Cholesky's method.*

Note that, strictly speaking an LU decomposition doesn't always exist. A theorem states that if all the principal minors of the matrix A are not equal to zero, then A is represented as $A = LU$ in unique way. Diagonal elements of L are different from zero.

On the other hand all Gaussian transformations can written in the matrix form. For example, if $n = 3$, we have

$$U = L_3 L_2 L_1 A,$$

where

$$L_1 = \begin{pmatrix} 1/a_{11} & 0 & 0 \\ -a_{21}/a_{11} & 1 & 0 \\ -a_{31}/a_{11} & 0 & 1 \end{pmatrix}, \quad L_2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1/a_{22}^{(1)} & 0 \\ 0 & -a_{32}^{(1)}/a_{22}^{(1)} & 1 \end{pmatrix}, \quad L_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1/a_{33}^{(2)} \end{pmatrix}.$$

L_1 performs elementary row operation to eliminate unknowns in the first column, L_2 - in the second column and so on. For the matrix L we have

$$L = L_1^{-1} L_2^{-1} L_3^{-1},$$

where

$$L_1^{-1} = \begin{pmatrix} a_{11} & 0 & 0 \\ a_{21} & 1 & 0 \\ a_{31} & 0 & 1 \end{pmatrix}, \quad L_2^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & a_{22}^{(1)} & 0 \\ 0 & a_{32}^{(1)} & 1 \end{pmatrix}, \quad L_3^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & a_{33}^{(2)} \end{pmatrix}.$$

In general case of $n \times n$ system we have

$$U = L_n L_{n-1} ... L_2 L_1 A, \qquad L = L_1^{-1} L_2^{-1} ... L_{n-1}^{-1} L_n^{-1},$$

where matrices L_k and L_k^{-1} $k = 1, ..., n$ look similar.

6. Gaussian elimination with scaled row pivoting

What should we do if at one step the leading element $a_{k+1,k+1}^{(k)} = 0$?

A theorem states that if $\det A \neq 0$, there exists a permutation matrix P such that the matrix PA has all principal minors not equal to zero. Hence $PA = LU$.

The basic idea of the Gaussian elimination with scaled row pivoting is that at every step of elimination forward procedure we arrange permutation of rows with i, $i = k+1, ..., n$ in such a way so that we have the leading element $a^{(k)}_{k+1,k+1}$ maximal on modules. We search for the row i with maximal $|a^{(k)}_{i,k+1}|$, $i = k+1, ..., n$ (see (9)), and then we arrange permutation between rows $k+1$ and i. Finally, we obtain

$$L = L_n L_{n-1} P_{n-1,j_{n-1}} ... L_1 P_{1,j_1} A,$$

where where P_{k,j_k} with $k = 1, ..., n-1$ is the permutation matrix or matrix of elementary row operation. For example, if $n = 3$,

$$P_{12} = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad P_{23} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad P_{13} = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix},$$

provide permutation between rows with 1 and 2, 2 and 3, 1 and 3.

7. Numerical calculation of determinant and universe matrix

Using the LU decomposition it is easily to calculate determinant of a matrix

$$\det A = \pm \det(PA) = \pm \det(LU) = \pm \det L \det U = \pm \det L = l_{11} l_{22} ... l_{nn}.$$

If A^{-1} is the universe matrix of A, then $AA^{-1} = I$, where I is the identity matrix. Let us denote $X = A^{-1}$. Actually, we have n systems of linear algebraic equations

$$\sum_{k=1}^{n} a_{ik} x_{kj} = \delta_{ij}$$

or $Ax^{(j)} = \delta^{(j)}$, where $x^{(j)} = (x_{1j}, x_{2j}, ..., , x_{nj})^T$ and $\delta^{(j)} = (\delta_{1j}, ..., \delta_{nj})^T$, $i, j = 1, ..., n$. Using the LU decomposition - $A = LU$, we have to solve first $Ly^{(j)} = \delta^{(j)}$ for $y^{(j)} = (y_{1j}, y_{2j}, ..., , y_{nj})^T$ and then $Ux^{(j)} = y^{(j)}$ for $x^{(j)}$.

8. Cholesky decomposition (for symmetric matrices)

Cholesky's method defines $LL^T = A$, where A is symmetric matrix and

$$L = \begin{pmatrix} l_{11} & 0 & 0 & ... & 0 \\ l_{21} & l_{22} & 0 & ... & 0 \\ l_{31} & l_{32} & l_{33} & ... & 0 \\ ... & ... & ... & ... & \\ l_{n1} & l_{n2} & l_{n3} & ... & l_{nn} \end{pmatrix}, \quad L^T = \begin{pmatrix} l_{11} & l_{21} & l_{31} & ... & l_{n1} \\ 0 & l_{22} & l_{32} & ... & l_{n2} \\ 0 & 0 & l_{33} & ... & l_{n3} \\ ... & ... & ... & ... & \\ 0 & 0 & 0 & ... & l_{nn} \end{pmatrix}.$$

If we write it out in the components for a 3×3 case we obtain:

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ .. & a_{22} & a_{23} \\ .. & .. & a_{33} \end{pmatrix} = \begin{pmatrix} l_{11}^2 & l_{11}l_{21} & l_{11}l_{31} \\ .. & l_{22}^2 + l_{21}^2 & l_{31}l_{21} + l_{22}l_{32} \\ .. & .. & l_{31}^2 + l_{32}^2 + l_{33}^2 \end{pmatrix}.$$

Now we get a set of equations for the l_{ij}'s. Thus:

$$\begin{aligned} l_{11}^2 &= a_{11} && \Rightarrow \; l_{11} = a_{11}^{1/2} \\ l_{11}l_{21} &= a_{12} && \Rightarrow \; l_{21} = \left(l_{11}^{-1}\right) a_{12} \\ l_{11}l_{31} &= a_{13} && \Rightarrow \; l_{31} = \left(l_{11}^{-1}\right) a_{13} \\ l_{22}^2 + l_{21}^2 &= a_{22} && \Rightarrow \; l_{22} = \left(a_{22} - l_{21}^2\right)^{1/2} \\ l_{31}l_{21} + l_{32}l_{22} &= a_{32} && \Rightarrow \; l_{32} = \left(l_{22}^{-1}\right)\left(a_{32} - l_{21}l_{31}\right) \\ l_{31}^2 + l_{32}^2 + l_{33}^2 &= a_{33} && \Rightarrow \; l_{33} = \left(a_{33} - l_{31}^2 + l_{32}^2\right)^{1/2} \end{aligned}$$

This algorithm uses values of l_{ij} that are already known to build up values of l_{ij} for higher values of i and j.

The general formula for an $n \times n$ matrix is

$$l_{ii} = \left(a_{ii} - \sum_{k=1}^{i-1} l_{ik}^2 \right)^{1/2}; \tag{17}$$

$$l_{ji} = \left(l_{ii}^{-1} \right) \left(a_{ij} - \sum_{k=1}^{i-1} l_{ik} l_{jk} \right) \quad j = i+1, i+2, \ldots n. \tag{18}$$

We note that (17) involves taking the square root and would clearly like to avoid complex arithmetic. Hence, the factorisation will only exist for a restricted class of matrices; these are the symmetric positive definite matrices.

A symmetric matrix A is positive definite if $x^T Ax > 0$ for all nonzero vectors x. A necessary and sufficient condition for positive definiteness is that all the eigenvalues are positive. We have the following theorem.

> A symmetric matrix A is positive definite if and only if there is a unique lower triangular matrix L with positive diagonal entries such that $A = LL^T$.

One can test a matrix for positive definiteness by attempting the computation of the Cholesky factorisation.

Actually the difficult step here is knowing that the matrix is positive definite! This involves finding the eigenvalues and may take more time than actually solving the equation! There is a theorem that the matrix is positive definite if:

1. The diagonal terms a_{ii} are all positive, and

2. The matrix is *strictly diagonally dominant*, i.e if

$$|a_{ii}| > \sum_{j=1 \top j \neq i}^{j=n} |a_{ij}| \quad i = 1, \ldots, n \tag{19}$$