LECTURE FOUR

Solving linear systems with special properties

After reading this lecture notes, you should be able to:

1. *Understand possible scenarios where systems of linear equations that contains multiple right-hand-side are relevant and learn how to solve these systems*
2. *Define, explain and determine systems that are either symmetric, positive definite, tri-diagonal and diagonally dominant.*
3. *Understand systems with special properties and learn the specific methods for solving these systems with more efficient methods.*

These systems occur very often in practical applications, such as in numerical solution of partial differential equations. There are hardly any systems in practical applications that are not one of the above types. Hence, these systems deserve special treatment.

The study includes:

- The method based on the Cholesky decomposition of a symmetric positive definite matrix
- Gaussian elimination for special systems: Hessenberg, positive definite, and diagonally dominant.

1. *Solving a Linear System with Multiple Right-hand Sides*

Consider the problem

$AX = B$

Where $B = (b_1, \ldots, b_m)$ is an $n \times m$ matrix $(m \le n)$ and $X = (x_1, x_2, \ldots, x_m)$. Here b_i and x_i, $i = 1, \ldots, m$ are $n-$vectors.

Problems of this type arise in many practical applications. Once the matrix *A* has been factored, the factorization can be used to solve the preceding m linear systems. We state the procedure only with partial pivoting.

ALGORITHM : Solving $AX = B$: Linear System with Multiple Right-hand Sides

Step 1: Factor $MA = U$, using Gaussian elimination with partial pivoting.

Step 2: Solve the *m* upper triangular systems:

$$Ux_i = b_i' = Mb_i \quad i = 1, \ldots, m$$

Example: Solve $AX = B$**,** where

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix},\ B = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix}$$

The factorization is $MA = U$, where

$$U = \begin{pmatrix} 7 & 8 & 9 \\ 0 & \frac{6}{7} & \frac{19}{7} \\ 0 & 0 & \frac{-1}{2} \end{pmatrix},\ M = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -\frac{1}{7} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \end{pmatrix}$$

Solve:

$$Ux_1 = Mb_1 = \begin{pmatrix} 5 \\ 0.2857 \\ 0 \end{pmatrix}$$

$$x_1 = \begin{pmatrix} 0.3333 \\ 0.3333 \\ 0 \end{pmatrix}$$

Solve:

$$Ux_2 = Mb_2 = \begin{pmatrix} 6 \\ 1.1429 \\ 0 \end{pmatrix}$$

$$x_2 = \begin{pmatrix} -0.6667 \\ 1.3333 \\ 0 \end{pmatrix}$$

The above can be used to find the inverse of matrix.

2. Computing the Inverse of a Matrix

Computing the inverse of a matrix *A* is equivalent to solving the sets of linear systems
$Ax_i = e_i,\ i = 1,\dots,n$
Then n linear systems now can be solved using any of the techniques discusses earlier. However, Gaussian elimination with partial pivoting will be used in practice.

Because the matrix *A* is the same for all the systems, *A* has to be factored only once. Thus, if Gaussian elimination with partial pivoting is used, then b_i is replaced by e_i in the case of multiple RHS.

ALGORITHM

Computing the inverse of *A* using Gaussian Elimination with partial pivoting

Step 1: Factor *A* into $MA = U$.

Step 2: Solve the n upper triangular systems: $Ux_i = e_i' = Me_i,\ \ i = 1\dots,n$

Example 2.4.3

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

Step 1:

$MA = U$ where

$$U = \begin{pmatrix} 7 & 8 & 9 \\ 0 & \frac{6}{7} & \frac{19}{7} \\ 0 & 0 & -\frac{1}{2} \end{pmatrix},\ M = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & -\frac{1}{7} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \end{pmatrix}$$

Step 2:

$Ux_i = e_i' = Me_i,\ \ i = 1,2,3$

$$x_1 = \begin{pmatrix} 1 & -2 & 1 \end{pmatrix}^T,\ x_2 = \begin{pmatrix} -4.6667 & 6.3333 & 2.000 \end{pmatrix}^T$$
$$x_3 = \begin{pmatrix} 2.6667 & -3.3333 & 1 \end{pmatrix}^T$$

Thus, $A^{-1} = X = \begin{pmatrix} 1 & -4.6667 & 2.6667 \\ -2 & 6.3333 & -3.3333 \\ 1 & -2.000 & 1 \end{pmatrix}$

3. Computing the Determinant of a Matrix

In theory, $\det(A) \neq 0$ if and only if *A* is invertible. However, in computational setting it is not a reliable measure of nonsingularity. Thus *the determinant of a matrix is seldom needed in practice.* However, if one needs to compute the determinant of *A,* then the ordinary Gaussian elimination (with partial and complete pivoting) or any triangularization method can be used.

If Gaussian elimination without pivoting is used, we have

$$A = LU, \quad \det(A) = \det(L) \cdot \det(U)$$

U is an upper triangular matrix, so $\det(U) = u_{11}u_{22}...u_{nn} = a_{11}a_{11}^{(1)}...a_{nn}^{(n-1)}$; *L* is a unit lower triangular matrix, so $\det(L) = 1$

Computing Det(*A*) From *LU* Factorization

$$\det(A) = a_{11}a_{22}^{(1)} \ldots a_{nn}^{(n-1)} = \text{product of pivots}$$

If Gaussian elimination with partial pivoting is used, we have

$$MA = U$$

Then, $\det(M)\det(A) = \det(U)$. Now, $M = M_{n-1}P_{n-1}...M_2P_2M_1P_1$. Because the determinant of each of the lower elementary matrices is 1 and the determinant of each of the permutation matrices ± 1, we have

$$\det(M) = (-1)^r$$

Where r is the number of row interchanges made during the pivoting process. So we have

$$\begin{aligned} \det(A) &= \frac{1}{\det(M)} \det(U) \\ &= (-1)^r \cdot u_{11} \cdot u_{22} \ldots u_{nn} \\ &= (-1)^r a_{11}a_{22}^{(1)} \ldots a_{nn}^{(n-1)} \end{aligned}$$

Computing Det(*A*) From *MA = U* Factorization

$\det(A) = (-1)^r a_{11} a_{22}^{(1)} \cdots a_{nn}^{(n-1)}$ where r is the number of interchanges.

If Gaussian elimination with complete pivoting is used, we have

$MAQ = U$ so, $\det(M) \cdot \det(A) \cdot \det(Q) = \det(U)$

Let r and s be, respectively, the number of row and column interchanges. Then

$$\det(M) = (-1)^r$$
$$\det(Q) = (-1)^s$$

Thus we state the following.

Computing Det(*A*) From *MAQ = U* Factorization

$$\begin{aligned}\det(A) &= (-1)^{r+s} \det(U) \\ &= (-1)^{r+s} a_{11} a_{22}^{(1)} \cdots a_{nn}^{(n-1)}\end{aligned}$$

where r and s are the number of row and column interchanges.

EXAMPLE

$$A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 1 & 1 \end{pmatrix}$$

Using Gaussian elimination with partial pivoting

$$U = \begin{pmatrix} 1 & 2 & 4 \\ 0 & 1 & 1 \\ 0 & 0 & -1 \end{pmatrix}$$

Only one interchange occurs; therefore, $r = 1$. $\det(A) = (-1)\det(U) = (-1)(-1) = 1$.

Using Gaussian elimination with complete pivoting,

$$U = \begin{pmatrix} 3 & 2 & 1 \\ 0 & \frac{2}{3} & \frac{1}{3} \\ 0 & 0 & \frac{1}{2} \end{pmatrix}$$

In the first step, one row interchange and one column interchange occurred. In the second step, one row interchange and one column interchange occurred. Thus, $r=2, s=2$, and

$$\det(A)=(-1)^{r+s}\det(U)=(-1)^4\cdot 3\cdot\tfrac{2}{3}\cdot\tfrac{1}{2}=1$$

4. Symmetric Positive Definite Systems

First, we show that for a symmetric positive definite matrix *A,* there exists a unique factorization

$A=HH^T$

where *H* is a lower triangular matrix with positive diagonal entries.

This factorization is called the **Cholesky factorization (decomposition)**. The existence of the Cholesky factorization for a symmetric positive definite matrix *A* can be seen either via *LU* factorization of *A* or by finding the matrix *H* directly from the preceding relation. In practical computations, the latter is preferable.

Computing the Cholesky Factorization

Gaussian elimination can be used to compute the Cholesky factorization of a symmetric positive definite matrix, and we have just seen that Gaussian elimination without pivoting is stable for such a matrix. However, in practice we do not use Gaussian elimination to compute the Cholesky factorization. Instead, we use a more efficient method, called the ***Cholesky algorithm,*** to do so.

The Cholesky Algorithm

From

$A=HH^T$

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} h_{11} & 0 & \cdots & 0 \\ h_{21} & h_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ h_{n1} & h_{n2} & \cdots & h_{nn} \end{pmatrix} \begin{pmatrix} h_{11} & h_{21} & \cdots & h_{n1} \\ 0 & h_{22} & \cdots & h_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & h_{nn} \end{pmatrix}$$

We have

$$h_{11}=\sqrt{a_{11}}, \qquad h_{i1}=\frac{a_{i1}}{h_{11}}, \quad i=2,\ldots,n$$

$$\sum_{k=1}^{i} h_{ik}^2 = a_{ii}, \qquad a_{ij}=\sum_{k=1}^{j} h_{ik}h_{jk} \quad j<i$$

which leads to the Cholesky algorithm.

Finding the matrix H directly

ALGORITHM: Cholesky Algorithm

Given an $n \times n$ symmetric positive definite matrix *A,* the following algorithm computes the Cholesky factor H . The matrix H is computed row by row.

Algorithm

for $k = 1, 2, \dots, n$ do

for $i = 1, 2, \dots, k-1$ do

$$h_{ki} = \frac{1}{h_{ii}}\left(a_{ki} - \sum_{j=1}^{i-1} h_{ij} h_{kj}\right)$$

end

$$h_{kk} = \sqrt{a_{kk} - \sum_{j=1}^{k-1} h_{kj}^2}$$

end

Remarks:

1. In the preceding pseudocode, $\sum_{j=1}^{0}(\ ) = 0$. Also when *k* = 1, the inner loop is skipped.
2. The Cholesky factor *H* is computed row by row.
3. The positive definiteness of *A* will make the quantities under the square root sign positive.

Solution of Ax=b Using the Cholosky Factorization

Having the Colesky factorization $A = HH^T$ at hand, the positive definite linear system $Ax = b$ can now be solved by solving the lower triangular system $Hy = b$ first, followed by the upper triangular system $H^T x = y$.

Procedure:

The Cholesky Algorithm for the Positive definite System $Ax = b$.

Step 1: Find the Cholesky factorization of $A = HH^T$

Step 2: Solve the lower triangular system for y : $Hy = b$

Step 3: Solve the upper triangular system for x: $H^T x = y$

Example

Let $A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 5 & 5 \\ 1 & 5 & 14 \end{pmatrix}, \qquad b = \begin{pmatrix} 3 \\ 11 \\ 20 \end{pmatrix}$

Step 1: Find the Cholesky factorization.

1st row $(k=1)$

$h_{11} = 1$

2nd row $(k=2)$

$$h_{21} = \frac{a_{21}}{h_{11}} = 1$$

$$h_{22} = \sqrt{a_{22} - h_{21}^2} = \sqrt{5-1} = \pm 2$$

(Because the diagonal entries of H have to be positive, we take the + sign,)

3rd row $(k=3)$

$$h_{31} = \frac{a_{31}}{h_{11}} = 1$$

$$h_{32} = \frac{1}{h_{22}}\left(a_{32} - h_{21}h_{31}\right) = \frac{1}{2}(5-1) = \sqrt{9}$$

(We take the + sign)

$h_{33} = +3$

$$H = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 2 & 3 \end{pmatrix}$$

Step 2: Solve $H_y = b$

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 1 & 2 & 3 \end{pmatrix}\begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 11 \\ 20 \end{pmatrix}$$

$y_1 = 3, \quad y_2 = 4, \quad y_3 = 3$

Step 3: Solve $H^T x = y$:

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 2 & 2 \\ 0 & 0 & 3 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 4 \\ 3 \end{pmatrix}$$

$x_3 = 1, \quad x_2 = 1, \quad x_3 = 1$

5. Tri-diagonal Systems

At the end of this section you will be able to solve linear system problems $Ax = b$ where *A* has special characteristics. The study includes Gaussian elimination for special systems: Tri-diagonal, and Block Tri-diagonal Systems.

The *LU* factorization of a tri-diagonal matrix *T,* when it exists, may yield *L* and *U* having very special simple structures: both bi-diagonal, *L* having 1's along the main diagonal and the superdiagonal entries of *U* the same as those of *T.* In particular, if we write

$$T = \begin{pmatrix} a_1 & b_1 & & & \\ c_2 & \ddots & \ddots & & \\ & \ddots & \ddots & b_{n-1} & \\ & & c_n & a_n & \end{pmatrix} = \begin{pmatrix} 1 & & & & \\ l_2 & \ddots & & 0 & \\ & \ddots & \ddots & & \\ & 0 & \ddots & \ddots & \\ & & & l_n & 1 \end{pmatrix} \begin{pmatrix} u_1 & b_1 & & & \\ & \ddots & \ddots & 0 & \\ & & \ddots & \ddots & \\ & 0 & & \ddots & b_{n-1} \\ & & & & u_n \end{pmatrix}$$

By equating the corresponding elements of the matrices on both sides, we see that

$a_1 = u_1$

$c_i = l_i u_{i-1}, \quad i = 2, \dots, n$

$a_i = u_i + l_i b_{i-1}, \quad i = 2, \dots, n$

from which $\{l_i\}$ and $\{u_i\}$ can be easily computed.

Computing the LU Factorization of a Tridiagonal Matrix

$$Let\ T = \begin{pmatrix} a_1 & b_1 & & \\ c_2 & \ddots & \ddots & \\ & \ddots & \ddots & b_{n-1} \\ & & c_n & a_n \end{pmatrix},$$

Set $u_1 = a_1$

For $i = 2, \dots, n$ do

$$l_i = \frac{c_i}{u_{i-1}}$$

$$u_i = a_i - l_i b_{i-1}$$

Solving a Tridiagonal System

Once we have the preceeding simple factorization of T, the solution of the tridiagonal system $Tx = b$ can be found by solving the two special bidiagonal systems:

$$Ly = b$$

and

$$Ux = y$$

Flop count.

The solutions of these two bidiagonal systems also require $2n-2$ flops. Thus, *a tridiagonal system can be solved by this procedure in only 4n-4 flops,* a very cheap procedure indeed.

Example

Triangularize

$A=\begin{pmatrix} 0.9 & 0.1 & 0 \\ 0.8 & 0.5 & 0.1 \\ 0 & 0.1 & 0.5 \end{pmatrix}$ using (a) the formula *A* = *LU* and (b) Gaussian elimination**.**

(a) From $A=LU$,

$$u_1=0.9$$

$$i=2:$$

$$l_2=\frac{c_2}{u_1}=\frac{0.8}{0.9}=\frac{8}{9}=0.8889$$

$$u_2=a_2-l_2b_1=0.5-\frac{8}{9}\times 0.1=0.4111$$

$$i=3:$$

$$l_3=\frac{c_2}{u_2}=\frac{0.1}{0.41}=0.2432$$

$$u_3=a_3-l_3b_2=0.5-0.24\times 0.1=0.4757$$

$$L=\begin{pmatrix} 1 & 0 & 0 \\ 0.8889 & 1 & 0 \\ 0 & 0.2432 & 0.1 \end{pmatrix}$$

$$U=\begin{pmatrix} 0.9 & 0.1 & 0 \\ 0 & 0.4111 & 0.1 \\ 0 & 0 & 0.4757 \end{pmatrix}$$

(b) Using Gaussian elimination with partial pivoting, we have the fol1owing

Step 1: Multiplier $m_{21}=-\dfrac{0.8}{0.9}=-0.8889$:

$$A^{(1)}=\begin{pmatrix} 0.9 & 0.1 & 0 \\ 0 & 0.4111 & 0.1 \\ 0 & 0.1 & 0.5 \end{pmatrix}$$

Step 2: Multiplier $m_{32}=-\dfrac{0.1}{0.4111}=-0.2432$

$$A^{(2)} = \begin{pmatrix} 0.9 & 0.1 & 0 \\ 0 & 0.4111 & 0.1 \\ 0 & 0 & 0.4757 \end{pmatrix} = U$$

$$L = \begin{pmatrix} 1 & 0 & 0 \\ -m_{21} & 1 & 0 \\ 0 & -m_{32} & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0.8889 & 1 & 0 \\ 0 & 0.2432 & 1 \end{pmatrix}$$