Direct Methods for Solving Systems of Linear Equations

NTNU

Tsung-Min Hwang

October 5, 2003

Direct Methods for LS

1 – Triangular Systems 5
1.1 – Forward Substitution 6
1.2 – Back Substitution 9
2 – Gaussian Elimination and LU Factorization 11
2.1 – Gaussian Elimination 12
2.2 – Gaussian Transformation and LU Factorization 21
2.3 – Existence and Uniqueness of LU Factorization 29
3 – Pivoting 34
3.1 – The Need for Pivoting 34
3.2 – Partial Pivoting and Complete Pivoting 37
4 – Some Special Linear Systems 43
4.1 – Symmetric Positive Definite System and Cholesky Factorization 43
4.2 – Diagonally Dominant Systems 49
4.3 – Tridiagonal System 53
4.4 – General Banded Systems 55
5 – Perturbation Analysis 56

Direct Methods for LS

Solve linear systems of equations

$$\left\{\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots & & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n \end{array}\right.$$

Direct Methods for LS

Solve linear systems of equations

$$\left\{\begin{array}{rcl} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n & = & b_2 \\ \vdots & & \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n & = & b_n \end{array}\right.$$

Rewrite in the matrix form

$$Ax = b, \tag{1}$$

where

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

Direct Methods for LS

☞ This equation has a unique solution $x = A^{-1}b$ when the coefficient matrix A is nonsingular.

Direct Methods for LS

☞ This equation has a unique solution $x = A^{-1}b$ when the coefficient matrix A is nonsingular.

☞ Direct methods are considered in this chapter.

Direct Methods for LS

☞ This equation has a unique solution $x = A^{-1}b$ when the coefficient matrix A is nonsingular.

☞ Direct methods are considered in this chapter.

☞ Gaussian elimination is the principal tool in the direct solution of (1).

Direct Methods for LS

☞ This equation has a unique solution $x = A^{-1}b$ when the coefficient matrix A is nonsingular.

☞ Direct methods are considered in this chapter.

☞ Gaussian elimination is the principal tool in the direct solution of (1).

☞ Use Gaussian elimination to factor the coefficient matrix into a product of matrices.

Direct Methods for LS

☞ This equation has a unique solution $x = A^{-1}b$ when the coefficient matrix A is nonsingular.

☞ Direct methods are considered in this chapter.

☞ Gaussian elimination is the principal tool in the direct solution of (1).

☞ Use Gaussian elimination to factor the coefficient matrix into a product of matrices. The factorization is called LU-factorization

Direct Methods for LS

- ☞ This equation has a unique solution $x = A^{-1}b$ when the coefficient matrix A is nonsingular.
- ☞ Direct methods are considered in this chapter.
- ☞ Gaussian elimination is the principal tool in the direct solution of (1).
- ☞ Use Gaussian elimination to factor the coefficient matrix into a product of matrices. The factorization is called LU-factorization and has the form $A = LU$,

Direct Methods for LS

☞ This equation has a unique solution $x = A^{-1}b$ when the coefficient matrix A is nonsingular.

☞ Direct methods are considered in this chapter.

☞ Gaussian elimination is the principal tool in the direct solution of (1).

☞ Use Gaussian elimination to factor the coefficient matrix into a product of matrices. The factorization is called LU-factorization and has the form $A = LU$, where L is unit lower triangular and U is upper triangular.

Direct Methods for LS

1 – Triangular Systems

1 – Triangular Systems

Let

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

1 – Triangular Systems

Let

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

☞ Provided that all $a_{ii} \neq 0$, then

$$x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T = \begin{bmatrix} b_1/a_{11} & b_2/a_{22} & \cdots & b_n/a_{nn} \end{bmatrix}^T.$$

1 – Triangular Systems

Let

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

☞ Provided that all $a_{ii} \neq 0$, then

$$x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T = \begin{bmatrix} b_1/a_{11} & b_2/a_{22} & \cdots & b_n/a_{nn} \end{bmatrix}^T.$$

☞ If $a_{ii} = 0$ and $b_i = 0$ for some index i, then x_i can be any real number.

Direct Methods for LS

1 – Triangular Systems

Let

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

☞ Provided that all $a_{ii} \neq 0$, then

$$x = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}^T = \begin{bmatrix} b_1/a_{11} & b_2/a_{22} & \cdots & b_n/a_{nn} \end{bmatrix}^T.$$

☞ If $a_{ii} = 0$ and $b_i = 0$ for some index i, then x_i can be any real number.

☞ If $a_{ii} = 0$ but $b_i \neq 0$, no solution of the system exists.

1.1 – Forward Substitution

,

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$,

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$x_1$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$x_1 =$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$x_1 = b_1/\ell_{11}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 & \end{aligned}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 &= \end{aligned}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 &= (b_2 - \ell_{21}x_1)/\ell_{22} \end{aligned}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 &= (b_2 - \ell_{21}x_1)/\ell_{22} \\ x_3 & \end{aligned}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 &= (b_2 - \ell_{21}x_1)/\ell_{22} \\ x_3 &= \end{aligned}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 &= (b_2 - \ell_{21}x_1)/\ell_{22} \\ x_3 &= (b_3 - \ell_{31}x_1 - \ell_{32}x_2)/\ell_{33} \end{aligned}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 &= (b_2 - \ell_{21}x_1)/\ell_{22} \\ x_3 &= (b_3 - \ell_{31}x_1 - \ell_{32}x_2)/\ell_{33} \\ &\vdots \end{aligned}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 &= (b_2 - \ell_{21}x_1)/\ell_{22} \\ x_3 &= (b_3 - \ell_{31}x_1 - \ell_{32}x_2)/\ell_{33} \\ &\vdots \\ x_n & \end{aligned}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 &= (b_2 - \ell_{21}x_1)/\ell_{22} \\ x_3 &= (b_3 - \ell_{31}x_1 - \ell_{32}x_2)/\ell_{33} \\ &\vdots \\ x_n &= \end{aligned}$$

1.1 – Forward Substitution

When a linear system $Lx = b$ is lower triangular of the form

$$\begin{bmatrix} \ell_{11} & 0 & \cdots & 0 \\ \ell_{21} & \ell_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \cdots & \ell_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix},$$

where all diagonals $\ell_{ii} \neq 0$, x_i can be obtained by the following procedure

$$\begin{aligned} x_1 &= b_1/\ell_{11} \\ x_2 &= (b_2 - \ell_{21}x_1)/\ell_{22} \\ x_3 &= (b_3 - \ell_{31}x_1 - \ell_{32}x_2)/\ell_{33} \\ &\vdots \\ x_n &= (b_n - \ell_{n1}x_1 - \ell_{n2}x_2 - \cdots - \ell_{n,n-1}x_{n-1})/\ell_{nn} \end{aligned}$$

Direct Methods for LS

The general formulation for computing x_i is

Direct Methods for LS

The general formulation for computing x_i is

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Big/ \ell_{ii}, \qquad i = 1, 2, \ldots, n.$$

Direct Methods for LS

The general formulation for computing x_i is

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Big/ \ell_{ii}, \qquad i = 1, 2, \ldots, n.$$

Algorithm 1 (Forward Substitution) *Suppose that $L \in \mathbb{R}^{n \times n}$ is nonsingular lower triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Lx = b$.*

Direct Methods for LS

The general formulation for computing x_i is

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Bigg/ \ell_{ii}, \qquad i = 1, 2, \dots, n.$$

Algorithm 1 (Forward Substitution) *Suppose that $L \in \mathbb{R}^{n\times n}$ is nonsingular lower triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Lx = b$.*

```
For $i = 1, \dots, n$
```

Direct Methods for LS

The general formulation for computing x_i is

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Big/ \ell_{ii}, \qquad i = 1, 2, \ldots, n.$$

Algorithm 1 (Forward Substitution) *Suppose that $L \in \mathbb{R}^{n\times n}$ is nonsingular lower triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Lx = b$.*

```
For $i = 1, \ldots, n$
    $tmp = 0$
```

Direct Methods for LS

The general formulation for computing x_i is

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Big/ \ell_{ii}, \qquad i = 1, 2, \ldots, n.$$

Algorithm 1 (Forward Substitution) *Suppose that $L \in \mathbb{R}^{n \times n}$ is nonsingular lower triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Lx = b$.*

```
For $i = 1, \ldots, n$
    $tmp = 0$
    For $j = 1, \ldots, i-1$
```

Direct Methods for LS

The general formulation for computing x_i is

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Big/ \ell_{ii}, \qquad i = 1, 2, \ldots, n.$$

Algorithm 1 (Forward Substitution) *Suppose that $L \in \mathbb{R}^{n \times n}$ is nonsingular lower triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Lx = b$.*

```
For $i = 1, \ldots, n$
    $tmp = 0$
    For $j = 1, \ldots, i-1$
        $tmp = tmp + L(i,j) * x(j)$
```

Direct Methods for LS

The general formulation for computing x_i is

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Big/ \ell_{ii}, \qquad i = 1, 2, \ldots, n.$$

Algorithm 1 (Forward Substitution) *Suppose that $L \in \mathbb{R}^{n\times n}$ is nonsingular lower triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Lx = b$.*

```
For $i = 1, \ldots, n$
    $tmp = 0$
    For $j = 1, \ldots, i-1$
        $tmp = tmp + L(i,j) * x(j)$
    End for
```

Direct Methods for LS

The general formulation for computing x_i is

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Big/ \ell_{ii}, \qquad i = 1, 2, \ldots, n.$$

Algorithm 1 (Forward Substitution) *Suppose that $L \in \mathbb{R}^{n\times n}$ is nonsingular lower triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Lx = b$.*

```
For $i = 1, \ldots, n$
    $tmp = 0$
    For $j = 1, \ldots, i-1$
        $tmp = tmp + L(i,j) * x(j)$
    End for
    $x(i) = (b(i) - tmp)/L(i,i)$
```

Direct Methods for LS

The general formulation for computing x_i is

$$x_i = \left(b_i - \sum_{j=1}^{i-1} \ell_{ij} x_j \right) \Bigg/ \ell_{ii}, \qquad i = 1, 2, \ldots, n.$$

Algorithm 1 (Forward Substitution) *Suppose that $L \in \mathbb{R}^{n\times n}$ is nonsingular lower triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Lx = b$.*

```
For $i = 1, \ldots, n$
    $tmp = 0$
    For $j = 1, \ldots, i-1$
        $tmp = tmp + L(i,j) * x(j)$
    End for
    $x(i) = (b(i) - tmp)/L(i,i)$
End for
```

Direct Methods for LS

The number of floating-point operations, flops, involved in the forward substitution are

Direct Methods for LS

The number of floating-point operations, flops, involved in the forward substitution are

$$\sum_{i=1}^{n} [2(i-1)+2] = n^2 + n.$$

Direct Methods for LS

The number of floating-point operations, flops, involved in the forward substitution are

$$\sum_{i=1}^{n} [2(i-1)+2] = n^2 + n.$$

Hence the forward substitution algorithm is an $O(n^2)$ algorithm.

1.2 – Back Substitution

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$x_n$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$x_n =$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$x_n = b_n/u_{nn}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$x_n = b_n/u_{nn}$$

$$x_{n-1}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$\begin{aligned} x_n &= b_n/u_{nn} \\ x_{n-1} &= \end{aligned}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$\begin{aligned} x_n &= b_n/u_{nn} \\ x_{n-1} &= (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} \end{aligned}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$\begin{aligned} x_n &= b_n/u_{nn} \\ x_{n-1} &= (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} \\ x_{n-2} & \end{aligned}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$\begin{aligned} x_n &= b_n/u_{nn} \\ x_{n-1} &= (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} \\ x_{n-2} &= \end{aligned}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$\begin{aligned} x_n &= b_n/u_{nn} \\ x_{n-1} &= (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} \\ x_{n-2} &= (b_{n-2} - u_{n-2,n-1}x_{n-1} - u_{n-2,n}x_n)/u_{n-2,n-2} \end{aligned}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$\begin{aligned} x_n &= b_n/u_{nn} \\ x_{n-1} &= (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} \\ x_{n-2} &= (b_{n-2} - u_{n-2,n-1}x_{n-1} - u_{n-2,n}x_n)/u_{n-2,n-2} \\ &\vdots \end{aligned}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$\begin{aligned} x_n &= b_n/u_{nn} \\ x_{n-1} &= (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} \\ x_{n-2} &= (b_{n-2} - u_{n-2,n-1}x_{n-1} - u_{n-2,n}x_n)/u_{n-2,n-2} \\ &\vdots \\ x_1 & \end{aligned}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$\begin{aligned} x_n &= b_n/u_{nn} \\ x_{n-1} &= (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} \\ x_{n-2} &= (b_{n-2} - u_{n-2,n-1}x_{n-1} - u_{n-2,n}x_n)/u_{n-2,n-2} \\ &\vdots \\ x_1 &= \end{aligned}$$

1.2 – Back Substitution

Consider the upper triangular system $Ux = b$:

$$\begin{bmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

provided that all $u_{ii} \neq 0$.The solution x_i are computed in a reversed order by

$$\begin{aligned} x_n &= b_n/u_{nn} \\ x_{n-1} &= (b_{n-1} - u_{n-1,n}x_n)/u_{n-1,n-1} \\ x_{n-2} &= (b_{n-2} - u_{n-2,n-1}x_{n-1} - u_{n-2,n}x_n)/u_{n-2,n-2} \\ &\vdots \\ x_1 &= (b_1 - u_{12}x_2 - u_{13}x_3 - \cdots - u_{1n}x_n)/u_{11} \end{aligned}$$

Direct Methods for LS

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Algorithm 2 (Back Substitution) *Suppose that $U \in \mathbb{R}^{n \times n}$ is nonsingular upper triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Ux = b$.*

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Bigg/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Algorithm 2 (Back Substitution) *Suppose that $U \in \mathbb{R}^{n \times n}$ is nonsingular upper triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Ux = b$.*

```
For $i = n, \ldots, 1$
```

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Algorithm 2 (Back Substitution) *Suppose that $U \in \mathbb{R}^{n \times n}$ is nonsingular upper triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Ux = b$.*

```
For $i = n, \ldots, 1$
    $tmp = 0$
```

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Algorithm 2 (Back Substitution) *Suppose that $U \in \mathbb{R}^{n \times n}$ is nonsingular upper triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Ux = b$.*

```
For $i = n, \ldots, 1$
    $tmp = 0$
    For $j = i + 1, \ldots, n$
```

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Algorithm 2 (Back Substitution) *Suppose that $U \in \mathbb{R}^{n \times n}$ is nonsingular upper triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Ux = b$.*

```
For $i = n, \ldots, 1$
    $tmp = 0$
    For $j = i+1, \ldots, n$
        $tmp = tmp + U(i,j) * x(j)$
```

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Algorithm 2 (Back Substitution) *Suppose that $U \in \mathbb{R}^{n\times n}$ is nonsingular upper triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Ux = b$.*

```
For $i = n, \ldots, 1$
    $tmp = 0$
    For $j = i+1, \ldots, n$
        $tmp = tmp + U(i,j) * x(j)$
    End for
```

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Algorithm 2 (Back Substitution) *Suppose that $U \in \mathbb{R}^{n\times n}$ is nonsingular upper triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Ux = b$.*

```
For $i = n, \ldots, 1$
    $tmp = 0$
    For $j = i+1, \ldots, n$
        $tmp = tmp + U(i,j) * x(j)$
    End for
    $x(i) = (b(i) - tmp)/U(i,i)$
```

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Algorithm 2 (Back Substitution) *Suppose that $U \in \mathbb{R}^{n\times n}$ is nonsingular upper triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Ux = b$.*

```
For $i = n, \ldots, 1$
    $tmp = 0$
    For $j = i+1, \ldots, n$
        $tmp = tmp + U(i,j) * x(j)$
    End for
    $x(i) = (b(i) - tmp)/U(i,i)$
End for
```

Direct Methods for LS

The general formulation is

$$x_i = \left(b_i - \sum_{j=i+1}^{n} u_{ij} x_j \right) \Big/ u_{ii}, \qquad i = n, n-1, \ldots, 1.$$

Algorithm 2 (Back Substitution) *Suppose that $U \in \mathbb{R}^{n\times n}$ is nonsingular upper triangular and $b \in \mathbb{R}^n$. This algorithm computes the solution of $Ux = b$.*

```
For $i = n, \ldots, 1$
    $tmp = 0$
    For $j = i+1, \ldots, n$
        $tmp = tmp + U(i,j) * x(j)$
    End for
    $x(i) = (b(i) - tmp)/U(i,i)$
End for
```

Back substitution requires $n^2 + O(n)$ flops.

2 – Gaussian Elimination and LU Factorization

2 – Gaussian Elimination and LU Factorization

In this section we will derive an algorithm that computes a matrix factorization called LU factorization such that $A = LU$, where L is unit lower triangular and U is upper triangular.

2 – Gaussian Elimination and LU Factorization

In this section we will derive an algorithm that computes a matrix factorization called LU factorization such that $A = LU$, where L is unit lower triangular and U is upper triangular.

The solution to the original problem $Ax = LUx = b$ is then found by a two-step triangular solve process:

$$Ly = b, \qquad Ux = y. \tag{2}$$

2.1 – Gaussian Elimination

Three types of elementary row operations for a system of linear equations:

2.1 – Gaussian Elimination

Three types of elementary row operations for a system of linear equations:

1. Interchange two equations in the system (or equivalently, interchange two rows in A):

$$\mathcal{E}_i \leftrightarrow \mathcal{E}_j;$$

   Here $\mathcal{E}_i$ denotes the i-th equation in the system.

2.1 – Gaussian Elimination

Three types of elementary row operations for a system of linear equations:

1. Interchange two equations in the system (or equivalently, interchange two rows in A):
$$\mathcal{E}_i \leftrightarrow \mathcal{E}_j;$$
Here $\mathcal{E}_i$ denotes the i-th equation in the system.
2. Multiply an equation by a non-zero constant (multiply one row of A by a non-zero constant):
$$\mathcal{E}_i \leftarrow \lambda \mathcal{E}_i.$$

2.1 – Gaussian Elimination

Three types of elementary row operations for a system of linear equations:

1. Interchange two equations in the system (or equivalently, interchange two rows in A):

$$\mathcal{E}_i \leftrightarrow \mathcal{E}_j;$$

   Here $\mathcal{E}_i$ denotes the i-th equation in the system.

2. Multiply an equation by a non-zero constant (multiply one row of A by a non-zero constant):

$$\mathcal{E}_i \leftarrow \lambda \mathcal{E}_i.$$

3. Add to an equation a multiple of some other equation (add to a row a multiple of some other row):

$$\mathcal{E}_i \leftarrow \mathcal{E}_i + \lambda \mathcal{E}_j.$$

Direct Methods for LS

Direct Methods for LS

☞ The first step in the Gaussian elimination process:

Direct Methods for LS

☞ The first step in the Gaussian elimination process: for each $i = 2, 3, \ldots, n$,

$$\mathcal{E}_i \leftarrow (\mathcal{E}_i - m_{i,1}\mathcal{E}_1), \qquad \text{where} \quad m_{i,1} = \frac{a_{i1}}{a_{11}}. \tag{3}$$

Direct Methods for LS

☞ The first step in the Gaussian elimination process: for each $i = 2, 3, \ldots, n$,

$$\mathcal{E}_i \leftarrow (\mathcal{E}_i - m_{i,1}\mathcal{E}_1), \qquad \text{where} \quad m_{i,1} = \frac{a_{i1}}{a_{11}}. \tag{3}$$

$\Rightarrow$ Transform all the entries in the first column below the diagonal are zero.

Direct Methods for LS

☞ The first step in the Gaussian elimination process: for each $i = 2, 3, \ldots, n$,

$$\mathcal{E}_i \leftarrow (\mathcal{E}_i - m_{i,1}\mathcal{E}_1), \qquad \text{where} \quad m_{i,1} = \frac{a_{i1}}{a_{11}}. \tag{3}$$

$\Rightarrow$ Transform all the entries in the first column below the diagonal are zero. For example,

$$A_1 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow A_2 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} \\ 0 & \tilde{a}_{32} & \tilde{a}_{33} \end{bmatrix}$$

Direct Methods for LS

☞ The first step in the Gaussian elimination process: for each $i = 2, 3, \ldots, n$,

$$\mathcal{E}_i \leftarrow (\mathcal{E}_i - m_{i,1}\mathcal{E}_1), \qquad \text{where} \quad m_{i,1} = \frac{a_{i1}}{a_{11}}. \tag{3}$$

$\Rightarrow$ Transform all the entries in the first column below the diagonal are zero. For example,

$$A_1 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow A_2 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} \\ 0 & \tilde{a}_{32} & \tilde{a}_{33} \end{bmatrix}$$

☞ Then the process is repeated on the resulting equations $\mathcal{E}_2, \ldots, \mathcal{E}_n$, and so on.

Direct Methods for LS

☞ The first step in the Gaussian elimination process: for each $i = 2, 3, \ldots, n$,

$$\mathcal{E}_i \leftarrow (\mathcal{E}_i - m_{i,1}\mathcal{E}_1), \qquad \text{where} \quad m_{i,1} = \frac{a_{i1}}{a_{11}}. \tag{3}$$

$\Rightarrow$ Transform all the entries in the first column below the diagonal are zero. For example,

$$A_1 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow A_2 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} \\ 0 & \tilde{a}_{32} & \tilde{a}_{33} \end{bmatrix}$$

☞ Then the process is repeated on the resulting equations $\mathcal{E}_2, \ldots, \mathcal{E}_n$, and so on.

$$A_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} \\ 0 & \tilde{a}_{32} & \tilde{a}_{33} \end{bmatrix} \Rightarrow A_3 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} \\ 0 & 0 & \hat{a}_{33} \end{bmatrix}$$

Direct Methods for LS

☞ The first step in the Gaussian elimination process: for each $i = 2, 3, \ldots, n$,

$$\mathcal{E}_i \leftarrow (\mathcal{E}_i - m_{i,1}\mathcal{E}_1), \qquad \text{where} \quad m_{i,1} = \frac{a_{i1}}{a_{11}}. \tag{3}$$

$\Rightarrow$ Transform all the entries in the first column below the diagonal are zero. For example,

$$A_1 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \Rightarrow A_2 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} \\ 0 & \tilde{a}_{32} & \tilde{a}_{33} \end{bmatrix}$$

☞ Then the process is repeated on the resulting equations $\mathcal{E}_2, \ldots, \mathcal{E}_n$, and so on.

$$A_2 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} \\ 0 & \tilde{a}_{32} & \tilde{a}_{33} \end{bmatrix} \Rightarrow A_3 \equiv \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & \tilde{a}_{22} & \tilde{a}_{23} \\ 0 & 0 & \hat{a}_{33} \end{bmatrix}$$

A_3 is upper triangular.

Direct Methods for LS

Direct Methods for LS

The process of Gaussian elimination result in a sequence of matrices as follows:

$$A = A^{(1)} \longrightarrow A^{(2)} \longrightarrow \cdots \longrightarrow A^{(n)} = \text{upper triangular matrix},$$

Direct Methods for LS

The process of Gaussian elimination result in a sequence of matrices as follows:

$$A = A^{(1)} \to A^{(2)} \to \cdots \to A^{(n)} = \text{upper triangular matrix},$$

The matrix $A^{(k)}$ has the following form:

$$A^{(k)} = \left[\begin{array}{ccc|c|cccc}
a_{11}^{(k)} & \cdots & a_{1,k-1}^{(k)} & a_{1k}^{(k)} & \cdots & a_{1j}^{(k)} & \cdots & a_{1n}^{(k)} \\
\vdots & \ddots & \vdots & \vdots & & \vdots & & \vdots \\
0 & \cdots & a_{k-1,k-1}^{(k)} & a_{k-1,k}^{(k)} & \cdots & a_{k-1,j}^{(k)} & \cdots & a_{k-1,n}^{(k)} \\ \hline
0 & \cdots & 0 & a_{kk}^{(k)} & \cdots & a_{kj}^{(k)} & \cdots & a_{kn}^{(k)} \\ \hline
\vdots & & \vdots & \vdots & & \vdots & & \vdots \\
0 & \cdots & 0 & a_{ik}^{(k)} & \cdots & a_{ij}^{(k)} & \cdots & a_{in}^{(k)} \\
\vdots & & \vdots & \vdots & & \vdots & & \vdots \\
0 & \cdots & 0 & a_{nk}^{(k)} & \cdots & a_{nj}^{(k)} & \cdots & a_{nn}^{(k)}
\end{array}\right]$$

Direct Methods for LS

In the k-th step,

Direct Methods for LS

In the k-th step,

☞ $a_{kk}^{(k)}$ is used as a pivot element

Direct Methods for LS

In the k-th step,

☞ $a_{kk}^{(k)}$ is used as a pivot element

☞ Elementary operations are applied to rows $k+1$ through n so that zeros are produced in column k below the diagonal.

Direct Methods for LS

In the k-th step,

☞ $a_{kk}^{(k)}$ is used as a pivot element

☞ Elementary operations are applied to rows $k+1$ through n so that zeros are produced in column k below the diagonal.

That is, $A^{(k+1)}$ is obtained from $A^{(k)}$ in which

Direct Methods for LS

In the k-th step,

☞ $a_{kk}^{(k)}$ is used as a pivot element

☞ Elementary operations are applied to rows $k+1$ through n so that zeros are produced in column k below the diagonal.

That is, $A^{(k+1)}$ is obtained from $A^{(k)}$ in which

☞ $a_{k+1,k}^{(k+1)}, \cdots, a_{nk}^{(k+1)}$ are zero

Direct Methods for LS

In the k-th step,

- ☞ $a_{kk}^{(k)}$ is used as a pivot element
- ☞ Elementary operations are applied to rows $k+1$ through n so that zeros are produced in column k below the diagonal.

That is, $A^{(k+1)}$ is obtained from $A^{(k)}$ in which

- ☞ $a_{k+1,k}^{(k+1)}, \cdots, a_{nk}^{(k+1)}$ are zero
- ☞ row $k+1$ through n are modified

Direct Methods for LS

In the k-th step,

☞ $a_{kk}^{(k)}$ is used as a pivot element

☞ Elementary operations are applied to rows $k+1$ through n so that zeros are produced in column k below the diagonal.

That is, $A^{(k+1)}$ is obtained from $A^{(k)}$ in which

☞ $a_{k+1,k}^{(k+1)}, \cdots, a_{nk}^{(k+1)}$ are zero

☞ row $k+1$ through n are modified

☞ row 1 through row k are unchanged.

Direct Methods for LS

More precisely, the entries of $A^{(k+1)}$ are produced by the formula

Direct Methods for LS

More precisely, the entries of $A^{(k+1)}$ are produced by the formula

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)}, & \text{for } i = 1, \ldots, k, \text{ and } j = 1, \ldots, n; \\ 0, & \text{for } i = k+1, \ldots, n, \text{ and } j = 1, \ldots, k; \\ a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \times a_{kj}^{(k)}, & \text{for } i = k+1, \ldots, n, \text{ and } j = k+1, \ldots, n. \end{cases} \tag{4}$$

Direct Methods for LS

More precisely, the entries of $A^{(k+1)}$ are produced by the formula

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)}, & \text{for } i = 1, \ldots, k, \text{ and } j = 1, \ldots, n; \\ 0, & \text{for } i = k+1, \ldots, n, \text{ and } j = 1, \ldots, k; \\ a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \times a_{kj}^{(k)}, & \text{for } i = k+1, \ldots, n, \text{ and } j = k+1, \ldots, n. \end{cases} \qquad (4)$$

Let $L = [\ell_{ik}]$ with

$$\ell_{ik} = \begin{cases} 0, & \text{if } i < k; \\ 1, & \text{if } i = k; \\ \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, & \text{if } i > k, \end{cases} \qquad (5)$$

Direct Methods for LS

More precisely, the entries of $A^{(k+1)}$ are produced by the formula

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)}, & \text{for } i = 1, \ldots, k, \text{ and } j = 1, \ldots, n; \\ 0, & \text{for } i = k+1, \ldots, n, \text{ and } j = 1, \ldots, k; \\ a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \times a_{kj}^{(k)}, & \text{for } i = k+1, \ldots, n, \text{ and } j = k+1, \ldots, n. \end{cases} \tag{4}$$

Let $L = [\ell_{ik}]$ with

$$\ell_{ik} = \begin{cases} 0, & \text{if } i < k; \\ 1, & \text{if } i = k; \\ \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, & \text{if } i > k, \end{cases} \tag{5}$$

and $U = A^{(n)}$,

Direct Methods for LS

More precisely, the entries of $A^{(k+1)}$ are produced by the formula

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)}, & \text{for } i = 1, \ldots, k, \text{ and } j = 1, \ldots, n; \\ 0, & \text{for } i = k+1, \ldots, n, \text{ and } j = 1, \ldots, k; \\ a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \times a_{kj}^{(k)}, & \text{for } i = k+1, \ldots, n, \text{ and } j = k+1, \ldots, n. \end{cases} \quad (4)$$

Let $L = [\ell_{ik}]$ with

$$\ell_{ik} = \begin{cases} 0, & \text{if } i < k; \\ 1, & \text{if } i = k; \\ \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, & \text{if } i > k, \end{cases} \quad (5)$$

and $U = A^{(n)}$, then L is unit lower triangular, U is upper triangular,

Direct Methods for LS

More precisely, the entries of $A^{(k+1)}$ are produced by the formula

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)}, & \text{for } i = 1, \ldots, k, \text{ and } j = 1, \ldots, n; \\ 0, & \text{for } i = k+1, \ldots, n, \text{ and } j = 1, \ldots, k; \\ a_{ij}^{(k)} - \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \times a_{kj}^{(k)}, & \text{for } i = k+1, \ldots, n, \text{ and } j = k+1, \ldots, n. \end{cases} \tag{4}$$

Let $L = [\ell_{ik}]$ with

$$\ell_{ik} = \begin{cases} 0, & \text{if } i < k; \\ 1, & \text{if } i = k; \\ \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, & \text{if } i > k, \end{cases} \tag{5}$$

and $U = A^{(n)}$, then L is unit lower triangular, U is upper triangular, and later we shall show that $A = LU$.

Direct Methods for LS

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \ldots, n-1$
```

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n \times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
```

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $t = A(i,k)/A(k,k)$
```

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $t = A(i,k)/A(k,k)$
        $A(i,k) = 0$
```

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $t = A(i,k)/A(k,k)$
        $A(i,k) = 0$
        $b(i) = b(i) - t \times b(k)$
```

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $t = A(i,k)/A(k,k)$
        $A(i,k) = 0$
        $b(i) = b(i) - t \times b(k)$
        For $j = k+1, \ldots, n$
```

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $t = A(i,k)/A(k,k)$
        $A(i,k) = 0$
        $b(i) = b(i) - t \times b(k)$
        For $j = k+1, \ldots, n$
            $A(i,j) = A(i,j) - t \times A(k,j)$
```

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $t = A(i,k)/A(k,k)$
        $A(i,k) = 0$
        $b(i) = b(i) - t \times b(k)$
        For $j = k+1, \ldots, n$
            $A(i,j) = A(i,j) - t \times A(k,j)$
        End for
```

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $t = A(i,k)/A(k,k)$
        $A(i,k) = 0$
        $b(i) = b(i) - t \times b(k)$
        For $j = k+1, \ldots, n$
            $A(i,j) = A(i,j) - t \times A(k,j)$
        End for
    End for
```

Direct Methods for LS

Algorithm 3 (Gaussian elimination) *Given $A \in \mathbb{R}^{n\times n}$ and $b \in \mathbb{R}^n$, this algorithm implements the Gaussian elimination procedure to reduce A to upper triangular and modify the entries of b accordingly.*

```
For $k = 1, \dots, n-1$
    For $i = k+1, \dots, n$
        $t = A(i,k)/A(k,k)$
        $A(i,k) = 0$
        $b(i) = b(i) - t \times b(k)$
        For $j = k+1, \dots, n$
            $A(i,j) = A(i,j) - t \times A(k,j)$
        End for
    End for
End for
```

Direct Methods for LS

Direct Methods for LS

Example 1 *Solve system of linear equations.*

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 34 \\ 27 \\ -38 \end{bmatrix}$$

Solution:

Direct Methods for LS

Example 1 *Solve system of linear equations.*

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 12 & -8 & 6 & 10 \\ 3 & -13 & 9 & 3 \\ -6 & 4 & 1 & -18 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 34 \\ 27 \\ -38 \end{bmatrix}$$

Solution:

1^{st} **step** Use 6 as pivot element, the first row as pivot row, and multipliers $2, \frac{1}{2}, -1$ are produced to reduce the system to

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & -12 & 8 & 1 \\ 0 & 2 & 3 & -14 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ 21 \\ -26 \end{bmatrix}$$

Direct Methods for LS

2^{nd} **step** Use -4 as pivot element, the second row as pivot row, and multipliers $3, -\frac{1}{2}$ are computed to reduce the system to

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ -9 \\ -21 \end{bmatrix}$$

Direct Methods for LS

2^{nd} **step** Use -4 as pivot element, the second row as pivot row, and multipliers $3, -\frac{1}{2}$ are computed to reduce the system to

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 4 & -13 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ -9 \\ -21 \end{bmatrix}$$

3^{rd} **step** Use 2 as pivot element, the third row as pivot row, and multipliers 2 is found to reduce the system to

$$\begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 12 \\ 10 \\ -9 \\ -3 \end{bmatrix}$$

Direct Methods for LS

Collect all the multipliers and let

$$L = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ \frac{1}{2} & 3 & 1 & 0 \\ -1 & -\frac{1}{2} & 2 & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} 6 & -2 & 2 & 4 \\ 0 & -4 & 2 & 2 \\ 0 & 0 & 2 & -5 \\ 0 & 0 & 0 & -3 \end{bmatrix},$$

then one can verify that $LU = A$. ■

Direct Methods for LS

2.2 – Gaussian Transformation and LU Factorization

For a given vector $v \in \mathbb{R}^n$ with $v_k \neq 0$ for some $1 \leq k \leq n$, let

Direct Methods for LS

2.2 – Gaussian Transformation and LU Factorization

For a given vector $v \in \mathbb{R}^n$ with $v_k \neq 0$ for some $1 \leq k \leq n$, let

$$\ell_{ik} = \frac{v_i}{v_k}, \quad i = k+1, \ldots, n,$$

2.2 – Gaussian Transformation and LU Factorization

For a given vector $v \in \mathbb{R}^n$ with $v_k \neq 0$ for some $1 \leq k \leq n$, let

$$\ell_{ik} = \frac{v_i}{v_k}, \quad i = k+1, \ldots, n,$$

$$l_k = \begin{bmatrix} 0 & \cdots & 0 & \ell_{k+1,k} & \cdots & \ell_{n,k} \end{bmatrix}^T,$$

2.2 – Gaussian Transformation and LU Factorization

For a given vector $v \in \mathbb{R}^n$ with $v_k \neq 0$ for some $1 \leq k \leq n$, let

$$\ell_{ik} = \frac{v_i}{v_k}, \quad i = k+1, \ldots, n,$$

$$l_k = \begin{bmatrix} 0 & \cdots & 0 & \ell_{k+1,k} & \cdots & \ell_{n,k} \end{bmatrix}^T,$$

and

$$M_k = I - l_k e_k^T = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & -\ell_{k+1,k} & 1 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & -\ell_{n,k} & 0 & \cdots & 1 \end{bmatrix}.$$

Direct Methods for LS

Then one can verify that

$$M_k v = \begin{bmatrix} v_1 & \cdots & v_k & 0 & \cdots & 0 \end{bmatrix}^T.$$

Direct Methods for LS

Then one can verify that

$$M_k v = \begin{bmatrix} v_1 & \cdots & v_k & 0 & \cdots & 0 \end{bmatrix}^T.$$

M_k is called a Gaussian transformation, the vector l_k a Gauss vector.

Direct Methods for LS

Then one can verify that

$$M_k v = \begin{bmatrix} v_1 & \cdots & v_k & 0 & \cdots & 0 \end{bmatrix}^T.$$

M_k is called a Gaussian transformation, the vector l_k a Gauss vector. Furthermore, one can verify that

$$M_k^{-1} = (I - l_k e_k^T)^{-1} = I + l_k e_k^T = \begin{bmatrix} 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & \cdots & \ell_{k+1,k} & 1 & \cdots & 0 \\ \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & \cdots & \ell_{n,k} & 0 & \cdots & 1 \end{bmatrix}.$$

Direct Methods for LS

Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, denote $A^{(1)} \equiv [a_{ij}^{(1)}] = A$.

Direct Methods for LS

Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, denote $A^{(1)} \equiv [a_{ij}^{(1)}] = A$. If $a_{11}^{(1)} \neq 0$,

Direct Methods for LS

Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, denote $A^{(1)} \equiv [a_{ij}^{(1)}] = A$. If $a_{11}^{(1)} \neq 0$, then

$$M_1 = I - l_1 e_1^T,$$

Direct Methods for LS

Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, denote $A^{(1)} \equiv [a_{ij}^{(1)}] = A$. If $a_{11}^{(1)} \neq 0$, then

$$M_1 = I - l_1 e_1^T,$$

where

$$l_1 = \begin{bmatrix} 0 & \ell_{21} & \cdots & \ell_{n1} \end{bmatrix}^T, \quad \ell_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \; i = 2, \ldots, n,$$

can be formed such that

Direct Methods for LS

Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, denote $A^{(1)} \equiv [a_{ij}^{(1)}] = A$. If $a_{11}^{(1)} \neq 0$, then

$$M_1 = I - l_1 e_1^T,$$

where

$$l_1 = \begin{bmatrix} 0 & \ell_{21} & \cdots & \ell_{n1} \end{bmatrix}^T, \quad \ell_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \; i = 2, \ldots, n,$$

can be formed such that

$$A^{(2)} = M_1 A^{(1)} = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix},$$

Direct Methods for LS

Given a nonsingular matrix $A \in \mathbb{R}^{n\times n}$, denote $A^{(1)} \equiv [a_{ij}^{(1)}] = A$. If $a_{11}^{(1)} \neq 0$, then

$$M_1 = I - l_1 e_1^T,$$

where

$$l_1 = \begin{bmatrix} 0 & \ell_{21} & \cdots & \ell_{n1} \end{bmatrix}^T, \quad \ell_{i1} = \frac{a_{i1}^{(1)}}{a_{11}^{(1)}}, \; i = 2, \ldots, n,$$

can be formed such that

$$A^{(2)} = M_1 A^{(1)} = \begin{bmatrix} a_{11}^{(2)} & a_{12}^{(2)} & \cdots & a_{1n}^{(2)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(2)} & \cdots & a_{nn}^{(2)} \end{bmatrix},$$

where

$$a_{ij}^{(2)} = \begin{cases} a_{ij}^{(1)}, & \text{for } i = 1 \text{ and } j = 1, \ldots, n; \\ a_{ij}^{(1)} - \ell_{i1} \times a_{1j}^{(1)}, & \text{for } i = 2, \ldots, n \text{ and } j = 2, \ldots, n. \end{cases}$$

Direct Methods for LS

In general, at the k-th step, we are confronted with a matrix

Direct Methods for LS

In general, at the k-th step, we are confronted with a matrix

$$A^{(k)}$$

Direct Methods for LS

In general, at the k-th step, we are confronted with a matrix

$$A^{(k)} \quad =$$

Direct Methods for LS

In general, at the k-th step, we are confronted with a matrix

$$A^{(k)} \quad = \quad M_{k-1} \cdots M_2 M_1 A^{(1)}$$

Direct Methods for LS

In general, at the k-th step, we are confronted with a matrix

$$\begin{aligned} A^{(k)} &= M_{k-1} \cdots M_2 M_1 A^{(1)} \\ &= \end{aligned}$$

Direct Methods for LS

In general, at the k-th step, we are confronted with a matrix

$$
\begin{aligned}
A^{(k)} &= M_{k-1}\cdots M_2 M_1 A^{(1)} \\
&= \left[\begin{array}{cccc|ccc}
a_{11}^{(k)} & a_{12}^{(k)} & \cdots & a_{1,k-1}^{(k)} & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} \\
0 & a_{22}^{(k)} & \cdots & a_{2,k-1}^{(k)} & a_{2k}^{(k)} & \cdots & a_{2n}^{(k)} \\
\vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & a_{k-1,k-1}^{(k)} & a_{k-1,k}^{(k)} & \cdots & a_{k-1,n}^{(k)} \\
\hline
0 & 0 & \cdots & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\
\vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & a_{kn}^{(k)} & \cdots & a_{nn}^{(k)}
\end{array}\right].
\end{aligned}
$$

Direct Methods for LS

In general, at the k-th step, we are confronted with a matrix

$$\begin{aligned} A^{(k)} &= M_{k-1} \cdots M_2 M_1 A^{(1)} \\ &= \left[\begin{array}{cccc|ccc} a_{11}^{(k)} & a_{12}^{(k)} & \cdots & a_{1,k-1}^{(k)} & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} \\ 0 & a_{22}^{(k)} & \cdots & a_{2,k-1}^{(k)} & a_{2k}^{(k)} & \cdots & a_{2n}^{(k)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{k-1,k-1}^{(k)} & a_{k-1,k}^{(k)} & \cdots & a_{k-1,n}^{(k)} \\ \hline 0 & 0 & \cdots & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\ \vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & a_{kn}^{(k)} & \cdots & a_{nn}^{(k)} \end{array}\right]. \end{aligned}$$

If the pivot $a_{kk}^{(k)} \neq 0$,

Direct Methods for LS

In general, at the k-th step, we are confronted with a matrix

$$
\begin{aligned}
A^{(k)} &= M_{k-1}\cdots M_2 M_1 A^{(1)} \\
&= \left[\begin{array}{cccc|ccc}
a_{11}^{(k)} & a_{12}^{(k)} & \cdots & a_{1,k-1}^{(k)} & a_{1k}^{(k)} & \cdots & a_{1n}^{(k)} \\
0 & a_{22}^{(k)} & \cdots & a_{2,k-1}^{(k)} & a_{2k}^{(k)} & \cdots & a_{2n}^{(k)} \\
\vdots & \vdots & \ddots & \vdots & \vdots & & \vdots \\
0 & 0 & \cdots & a_{k-1,k-1}^{(k)} & a_{k-1,k}^{(k)} & \cdots & a_{k-1,n}^{(k)} \\
\hline
0 & 0 & \cdots & 0 & a_{kk}^{(k)} & \cdots & a_{kn}^{(k)} \\
\vdots & \vdots & & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & 0 & a_{kn}^{(k)} & \cdots & a_{nn}^{(k)}
\end{array}\right].
\end{aligned}
$$

If the pivot $a_{kk}^{(k)} \neq 0$, then the multipliers

$$
\ell_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \quad i = k+1, \ldots, n,
$$

Direct Methods for LS

can be computed

Direct Methods for LS

can be computed and the Gaussian transformation

$$M_k = I - l_k e_k^T, \quad \text{where} \quad l_k = \begin{bmatrix} 0 & \cdots & 0 & \ell_{k+1,k} & \cdots & \ell_{nk} \end{bmatrix},$$

can be applied to the left of $A^{(k)}$ to obtain

Direct Methods for LS

can be computed and the Gaussian transformation

$$M_k = I - l_k e_k^T, \quad \text{where} \quad l_k = \begin{bmatrix} 0 & \cdots & 0 & \ell_{k+1,k} & \cdots & \ell_{nk} \end{bmatrix},$$

can be applied to the left of $A^{(k)}$ to obtain

$$A^{(k+1)} = M_k A^{(k)}$$

Direct Methods for LS

can be computed and the Gaussian transformation

$$M_k = I - l_k e_k^T, \quad \text{where} \quad l_k = \begin{bmatrix} 0 & \cdots & 0 & \ell_{k+1,k} & \cdots & \ell_{nk} \end{bmatrix},$$

can be applied to the left of $A^{(k)}$ to obtain

$$A^{(k+1)} = M_k A^{(k)}$$

$$=$$

Direct Methods for LS

can be computed and the Gaussian transformation

$$M_k = I - l_k e_k^T, \quad \text{where} \quad l_k = \begin{bmatrix} 0 & \cdots & 0 & \ell_{k+1,k} & \cdots & \ell_{nk} \end{bmatrix},$$

can be applied to the left of $A^{(k)}$ to obtain

$$A^{(k+1)} = M_k A^{(k)}$$

$$= \left[\begin{array}{cccc|cccc} a_{11}^{(k+1)} & a_{12}^{(k+1)} & \cdots & a_{1,k-1}^{(k+1)} & a_{1k}^{(k+1)} & a_{1,k+1}^{(k+1)} & \cdots & a_{1n}^{(k+1)} \\ 0 & a_{22}^{(k+1)} & \cdots & a_{2,k-1}^{(k+1)} & a_{2k}^{(k+1)} & a_{2,k+1}^{(k+1)} & \cdots & a_{2n}^{(k+1)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & a_{k-1,k-1}^{(k+1)} & a_{k-1,k}^{(k+1)} & a_{k-1,k+1}^{(k+1)} & \cdots & a_{k-1,n}^{(k+1)} \\ \hline 0 & 0 & \cdots & 0 & a_{kk}^{(k+1)} & a_{k,k+1}^{(k+1)} & \cdots & a_{kn}^{(k+1)} \\ \vdots & \vdots & & \vdots & 0 & a_{k+1,k+1}^{(k+1)} & \cdots & \vdots \\ \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 0 & 0 & a_{n,k+1}^{(k+1)} & \cdots & a_{nn}^{(k+1)} \end{array}\right],$$

Direct Methods for LS

in which

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)}, & \text{for } i = 1, \ldots, k, \, j = 1, \ldots, n; \\ 0, & \text{for } i = k+1, \ldots, n, \, j = k; \\ a_{ij}^{(k)} - \ell_{ik} a_{kj}^{(k)}, & \text{for } i = k+1, \ldots, n, \, j = k+1, \ldots, n. \end{cases}$$

Direct Methods for LS

in which

$$a_{ij}^{(k+1)} = \begin{cases} a_{ij}^{(k)}, & \text{for } i = 1, \ldots, k,\ j = 1, \ldots, n; \\ 0, & \text{for } i = k+1, \ldots, n,\ j = k; \\ a_{ij}^{(k)} - \ell_{ik} a_{kj}^{(k)}, & \text{for } i = k+1, \ldots, n,\ j = k+1, \ldots, n. \end{cases}$$

Upon the completion,

$$U \equiv A^{(n)} = M_{n-1} \cdots M_2 M_1 A$$

is upper triangular.

Direct Methods for LS

Hence

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1}$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} \quad =$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} \quad = \quad (I - l_1 e_1^T)^{-1} (I - l_2 e_2^T)^{-1} \cdots (I - l_{n-1} e_{n-1}^T)^{-1}$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$\begin{aligned} L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} &= (I - l_1 e_1^T)^{-1} (I - l_2 e_2^T)^{-1} \cdots (I - l_{n-1} e_{n-1}^T)^{-1} \\ &= \end{aligned}$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$\begin{aligned} L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} &= (I - l_1 e_1^T)^{-1} (I - l_2 e_2^T)^{-1} \cdots (I - l_{n-1} e_{n-1}^T)^{-1} \\ &= (I + l_1 e_1^T)(I + l_2 e_2^T) \cdots (I + l_{n-1} e_{n-1}^T) \end{aligned}$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$\begin{aligned} L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} &= (I - l_1 e_1^T)^{-1} (I - l_2 e_2^T)^{-1} \cdots (I - l_{n-1} e_{n-1}^T)^{-1} \\ &= (I + l_1 e_1^T)(I + l_2 e_2^T) \cdots (I + l_{n-1} e_{n-1}^T) \\ &= \end{aligned}$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$\begin{aligned} L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} &= (I - l_1 e_1^T)^{-1} (I - l_2 e_2^T)^{-1} \cdots (I - l_{n-1} e_{n-1}^T)^{-1} \\ &= (I + l_1 e_1^T)(I + l_2 e_2^T) \cdots (I + l_{n-1} e_{n-1}^T) \\ &= I + l_1 e_1^T + l_2 e_2^T + \cdots + l_{n-1} e_{n-1}^T \end{aligned}$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$\begin{aligned} L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} &= (I - l_1 e_1^T)^{-1} (I - l_2 e_2^T)^{-1} \cdots (I - l_{n-1} e_{n-1}^T)^{-1} \\ &= (I + l_1 e_1^T)(I + l_2 e_2^T) \cdots (I + l_{n-1} e_{n-1}^T) \\ &= I + l_1 e_1^T + l_2 e_2^T + \cdots + l_{n-1} e_{n-1}^T \\ \\ \\ &= \end{aligned}$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$\begin{aligned}
L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} &= (I - l_1 e_1^T)^{-1} (I - l_2 e_2^T)^{-1} \cdots (I - l_{n-1} e_{n-1}^T)^{-1} \\
&= (I + l_1 e_1^T)(I + l_2 e_2^T) \cdots (I + l_{n-1} e_{n-1}^T) \\
&= I + l_1 e_1^T + l_2 e_2^T + \cdots + l_{n-1} e_{n-1}^T \\
&= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \ell_{21} & 1 & 0 & \cdots & 0 \\ \ell_{31} & \ell_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \cdots & 1 \end{bmatrix}
\end{aligned}$$

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$\begin{aligned} L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} &= (I - l_1 e_1^T)^{-1} (I - l_2 e_2^T)^{-1} \cdots (I - l_{n-1} e_{n-1}^T)^{-1} \\ &= (I + l_1 e_1^T)(I + l_2 e_2^T) \cdots (I + l_{n-1} e_{n-1}^T) \\ &= I + l_1 e_1^T + l_2 e_2^T + \cdots + l_{n-1} e_{n-1}^T \\ &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \ell_{21} & 1 & 0 & \cdots & 0 \\ \ell_{31} & \ell_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \cdots & 1 \end{bmatrix} \end{aligned}$$

is unit lower triangular.

Direct Methods for LS

Hence

$$A = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} U \equiv LU,$$

where

$$\begin{aligned} L \equiv M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} &= (I - l_1 e_1^T)^{-1} (I - l_2 e_2^T)^{-1} \cdots (I - l_{n-1} e_{n-1}^T)^{-1} \\ &= (I + l_1 e_1^T)(I + l_2 e_2^T) \cdots (I + l_{n-1} e_{n-1}^T) \\ &= I + l_1 e_1^T + l_2 e_2^T + \cdots + l_{n-1} e_{n-1}^T \\ &= \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \ell_{21} & 1 & 0 & \cdots & 0 \\ \ell_{31} & \ell_{32} & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \ell_{n1} & \ell_{n2} & \ell_{n3} & \cdots & 1 \end{bmatrix} \end{aligned}$$

is unit lower triangular. This matrix factorization is called the LU-factorization of A.

Direct Methods for LS

Algorithm 4 (LU Factorization) *Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm computes a unit lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The matrix A is overwritten by L and U.*

Direct Methods for LS

Algorithm 4 (*LU* Factorization) *Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm computes a unit lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The matrix A is overwritten by L and U.*

```
For $k = 1, \ldots, n-1$
```

Direct Methods for LS

Algorithm 4 (LU Factorization) *Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm computes a unit lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The matrix A is overwritten by L and U.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
```

Direct Methods for LS

Algorithm 4 (LU Factorization) *Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm computes a unit lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The matrix A is overwritten by L and U.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $A(i,k) = A(i,k)/A(k,k)$
```

Direct Methods for LS

Algorithm 4 (LU Factorization) *Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm computes a unit lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The matrix A is overwritten by L and U.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $A(i,k) = A(i,k)/A(k,k)$
        For $j = k+1, \ldots, n$
```

Direct Methods for LS

Algorithm 4 (LU Factorization) *Given a nonsingular square matrix* $A \in \mathbb{R}^{n\times n}$*, this algorithm computes a unit lower triangular matrix* L *and an upper triangular matrix* U *such that* $A = LU$*. The matrix* A *is overwritten by* L *and* U*.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $A(i,k) = A(i,k)/A(k,k)$
        For $j = k+1, \ldots, n$
            $A(i,j) = A(i,j) - A(i,k) \times A(k,j)$
```

Direct Methods for LS

Algorithm 4 (LU Factorization) *Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm computes a unit lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The matrix A is overwritten by L and U.*

```
For $k = 1, \ldots, n-1$
  For $i = k+1, \ldots, n$
    $A(i,k) = A(i,k)/A(k,k)$
    For $j = k+1, \ldots, n$
      $A(i,j) = A(i,j) - A(i,k) \times A(k,j)$
    End for
```

Direct Methods for LS

Algorithm 4 (LU Factorization) *Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm computes a unit lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The matrix A is overwritten by L and U.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $A(i,k) = A(i,k)/A(k,k)$
        For $j = k+1, \ldots, n$
            $A(i,j) = A(i,j) - A(i,k) \times A(k,j)$
        End for
    End for
```

Direct Methods for LS

Algorithm 4 (LU Factorization) *Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm computes a unit lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The matrix A is overwritten by L and U.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $A(i,k) = A(i,k)/A(k,k)$
        For $j = k+1, \ldots, n$
            $A(i,j) = A(i,j) - A(i,k) \times A(k,j)$
        End for
    End for
End for
```

Direct Methods for LS

Algorithm 4 (LU Factorization) *Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm computes a unit lower triangular matrix L and an upper triangular matrix U such that $A = LU$. The matrix A is overwritten by L and U.*

```
For $k = 1, \ldots, n-1$
    For $i = k+1, \ldots, n$
        $A(i,k) = A(i,k)/A(k,k)$
        For $j = k+1, \ldots, n$
            $A(i,j) = A(i,j) - A(i,k) \times A(k,j)$
        End for
    End for
End for
```

This algorithm requires

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

2.3 – Existence and Uniqueness of LU Factorization

Definition 1 (Leading principal minor) *Let A be an $n \times n$ matrix. The upper left $k \times k$ submatrix, denoted as*

$$A_k = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1k} \\ a_{21} & a_{22} & \cdots & a_{2k} \\ \vdots & \vdots & \ddots & \vdots \\ a_{k1} & a_{k2} & \cdots & a_{kk} \end{bmatrix},$$

is called the leading $k \times k$ principal submatrix, and the determinant of A_k, $\det(A_k)$, is called the leading principal minor.

Direct Methods for LS

Theorem 1 *If all leading principal minor of $A \in \mathbb{R}^{n\times n}$ are nonzero, that is, all leading principal submatrices are nonsingular, then A has an LU-factorization.*

Direct Methods for LS

Theorem 1 *If all leading principal minor of $A \in \mathbb{R}^{n\times n}$ are nonzero, that is, all leading principal submatrices are nonsingular, then A has an LU-factorization.*

Proof: Proof by mathematical induction.

Direct Methods for LS

Theorem 1 *If all leading principal minor of $A \in \mathbb{R}^{n\times n}$ are nonzero, that is, all leading principal submatrices are nonsingular, then A has an LU-factorization.*

Proof: Proof by mathematical induction.

(i) $n = 1$, $A_1 = [a_{11}]$ is nonsingular, then $a_{11} \neq 0$. Let $L_1 = [1]$ and $U_1 = [a_{11}]$. Then $A_1 = L_1 U_1$. The theorem holds.

Direct Methods for LS

Theorem 1 *If all leading principal minor of $A \in \mathbb{R}^{n\times n}$ are nonzero, that is, all leading principal submatrices are nonsingular, then A has an LU-factorization.*

Proof: Proof by mathematical induction.

(i) $n = 1$, $A_1 = [a_{11}]$ is nonsingular, then $a_{11} \neq 0$. Let $L_1 = [1]$ and $U_1 = [a_{11}]$. Then $A_1 = L_1U_1$. The theorem holds.

(ii) Assume that the leading principal submatrices $A_1, \ldots, A_k$ are nonsingular and A_k has an LU-factorization $A_k = L_kU_k$, where L_k is unit lower triangular and U_k is upper triangular.

Direct Methods for LS

Theorem 1 *If all leading principal minor of $A \in \mathbb{R}^{n\times n}$ are nonzero, that is, all leading principal submatrices are nonsingular, then A has an LU-factorization.*

Proof: Proof by mathematical induction.

(i) $n = 1$, $A_1 = [a_{11}]$ is nonsingular, then $a_{11} \neq 0$. Let $L_1 = [1]$ and $U_1 = [a_{11}]$. Then $A_1 = L_1 U_1$. The theorem holds.

(ii) Assume that the leading principal submatrices $A_1, \ldots, A_k$ are nonsingular and A_k has an LU-factorization $A_k = L_k U_k$, where L_k is unit lower triangular and U_k is upper triangular.

(iii) Show that there exist an unit lower triangular matrix L_{k+1} and an upper triangular matrix U_{k+1} such that $A_{k+1} = L_{k+1} U_{k+1}$.

Direct Methods for LS

Write

$$A_{k+1} = \begin{bmatrix} A_k & v_k \\ w_k^T & a_{k+1,k+1} \end{bmatrix},$$

Direct Methods for LS

Write

$$A_{k+1} = \begin{bmatrix} A_k & v_k \\ w_k^T & a_{k+1,k+1} \end{bmatrix},$$

where

$$v_k = \begin{bmatrix} a_{1,k+1} \\ a_{2,k+1} \\ \vdots \\ a_{k,k+1} \end{bmatrix} \quad \text{and} \quad w_k = \begin{bmatrix} a_{k+1,1} \\ a_{k+1,2} \\ \vdots \\ a_{k+1,k} \end{bmatrix}.$$

Direct Methods for LS

Write

$$A_{k+1} = \begin{bmatrix} A_k & v_k \\ w_k^T & a_{k+1,k+1} \end{bmatrix},$$

where

$$v_k = \begin{bmatrix} a_{1,k+1} \\ a_{2,k+1} \\ \vdots \\ a_{k,k+1} \end{bmatrix} \quad \text{and} \quad w_k = \begin{bmatrix} a_{k+1,1} \\ a_{k+1,2} \\ \vdots \\ a_{k+1,k} \end{bmatrix}.$$

Since A_k is nonsingular, both L_k and U_k are nonsingular.

Direct Methods for LS

Write

$$A_{k+1} = \begin{bmatrix} A_k & v_k \\ w_k^T & a_{k+1,k+1} \end{bmatrix},$$

where

$$v_k = \begin{bmatrix} a_{1,k+1} \\ a_{2,k+1} \\ \vdots \\ a_{k,k+1} \end{bmatrix} \quad \text{and} \quad w_k = \begin{bmatrix} a_{k+1,1} \\ a_{k+1,2} \\ \vdots \\ a_{k+1,k} \end{bmatrix}.$$

Since A_k is nonsingular, both L_k and U_k are nonsingular.
$\Rightarrow L_k y_k = v_k$ has a unique solution $y_k \in \mathbb{R}^k$, and $z^t U_k = w_k^T$ has a unique solution $z_k \in \mathbb{R}^k$.

Direct Methods for LS

Write

$$A_{k+1} = \begin{bmatrix} A_k & v_k \\ w_k^T & a_{k+1,k+1} \end{bmatrix},$$

where

$$v_k = \begin{bmatrix} a_{1,k+1} \\ a_{2,k+1} \\ \vdots \\ a_{k,k+1} \end{bmatrix} \quad \text{and} \quad w_k = \begin{bmatrix} a_{k+1,1} \\ a_{k+1,2} \\ \vdots \\ a_{k+1,k} \end{bmatrix}.$$

Since A_k is nonsingular, both L_k and U_k are nonsingular.

$\Rightarrow L_k y_k = v_k$ has a unique solution $y_k \in \mathbb{R}^k$, and $z^t U_k = w_k^T$ has a unique solution $z_k \in \mathbb{R}^k$. Let

$$L_{k+1} = \begin{bmatrix} L_k & 0 \\ z_k^T & 1 \end{bmatrix} \quad \text{and} \quad U_{k+1} = \begin{bmatrix} U_k & y_k \\ 0 & a_{k+1,k+1} - z_k^T y_k \end{bmatrix}.$$

Direct Methods for LS

Then L_{k+1} is unit lower triangular, U_{k+1} is upper triangular,

Direct Methods for LS

Then L_{k+1} is unit lower triangular, U_{k+1} is upper triangular, and

$$\begin{aligned} L_{k+1}U_{k+1} &= \begin{bmatrix} L_kU_k & L_ky_k \\ z_k^TU_k & z_k^Ty_k + a_{k+1,k+1} - z_k^Ty_k \end{bmatrix} \\ &= \begin{bmatrix} A_k & v_k \\ w_k^T & a_{k+1,k+1} \end{bmatrix} = A_{k+1}. \end{aligned}$$

Direct Methods for LS

Then L_{k+1} is unit lower triangular, U_{k+1} is upper triangular, and

$$\begin{aligned} L_{k+1}U_{k+1} &= \begin{bmatrix} L_kU_k & L_ky_k \\ z_k^TU_k & z_k^Ty_k + a_{k+1,k+1} - z_k^Ty_k \end{bmatrix} \\ &= \begin{bmatrix} A_k & v_k \\ w_k^T & a_{k+1,k+1} \end{bmatrix} = A_{k+1}. \end{aligned}$$

This proves the theorem. ■

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Proof: Suppose both $A = L_1U_1$ and $A = L_2U_2$ are LU factorizations.

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Proof: Suppose both $A = L_1U_1$ and $A = L_2U_2$ are LU factorizations. Since A is nonsingular, L_1, U_1, L_2, U_2 are all nonsingular,

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Proof: Suppose both $A = L_1U_1$ and $A = L_2U_2$ are LU factorizations. Since A is nonsingular, L_1, U_1, L_2, U_2 are all nonsingular, and

$$A = L_1U_1 = L_2U_2 \Longrightarrow L_2^{-1}L_1 = U_2U_1^{-1}.$$

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Proof: Suppose both $A = L_1U_1$ and $A = L_2U_2$ are LU factorizations. Since A is nonsingular, L_1, U_1, L_2, U_2 are all nonsingular, and

$$A = L_1U_1 = L_2U_2 \Longrightarrow L_2^{-1}L_1 = U_2U_1^{-1}.$$

L_1 and L_2 are unit lower triangular

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Proof: Suppose both $A = L_1U_1$ and $A = L_2U_2$ are LU factorizations. Since A is nonsingular, L_1, U_1, L_2, U_2 are all nonsingular, and

$$A = L_1U_1 = L_2U_2 \Longrightarrow L_2^{-1}L_1 = U_2U_1^{-1}.$$

L_1 and L_2 are unit lower triangular $\Rightarrow L_2^{-1}L_1$ is unit lower triangular

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Proof: Suppose both $A = L_1U_1$ and $A = L_2U_2$ are LU factorizations. Since A is nonsingular, L_1, U_1, L_2, U_2 are all nonsingular, and

$$A = L_1U_1 = L_2U_2 \Longrightarrow L_2^{-1}L_1 = U_2U_1^{-1}.$$

L_1 and L_2 are unit lower triangular $\Rightarrow L_2^{-1}L_1$ is unit lower triangular
U_1 and U_2 are upper triangular

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Proof: Suppose both $A = L_1U_1$ and $A = L_2U_2$ are LU factorizations. Since A is nonsingular, L_1, U_1, L_2, U_2 are all nonsingular, and

$$A = L_1U_1 = L_2U_2 \Longrightarrow L_2^{-1}L_1 = U_2U_1^{-1}.$$

L_1 and L_2 are unit lower triangular $\Rightarrow L_2^{-1}L_1$ is unit lower triangular
U_1 and U_2 are upper triangular $\Rightarrow U_2U_1^{-1}$ is upper triangular

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11}\cdots u_{nn}$.*

Proof: Suppose both $A = L_1U_1$ and $A = L_2U_2$ are LU factorizations. Since A is nonsingular, L_1, U_1, L_2, U_2 are all nonsingular, and

$$A = L_1U_1 = L_2U_2 \Longrightarrow L_2^{-1}L_1 = U_2U_1^{-1}.$$

L_1 and L_2 are unit lower triangular $\Rightarrow L_2^{-1}L_1$ is unit lower triangular
U_1 and U_2 are upper triangular $\Rightarrow U_2U_1^{-1}$ is upper triangular
$\therefore L_2^{-1}L_1 = I = U_2U_1^{-1}$

Direct Methods for LS

Theorem 2 *If A is nonsingular and the LU factorization exists, then the LU factorization is unique and $\det(A) = u_{11} \cdots u_{nn}$.*

Proof: Suppose both $A = L_1 U_1$ and $A = L_2 U_2$ are LU factorizations. Since A is nonsingular, L_1, U_1, L_2, U_2 are all nonsingular, and

$$A = L_1 U_1 = L_2 U_2 \Longrightarrow L_2^{-1} L_1 = U_2 U_1^{-1}.$$

L_1 and L_2 are unit lower triangular $\Rightarrow L_2^{-1} L_1$ is unit lower triangular
U_1 and U_2 are upper triangular $\Rightarrow U_2 U_1^{-1}$ is upper triangular
$\therefore L_2^{-1} L_1 = I = U_2 U_1^{-1} \Rightarrow L_1 = L_2$ and $U_1 = U_2$ ■

Direct Methods for LS

3 – Pivoting

3.1 – The Need for Pivoting

Example. The algorithm would fail at the first step on

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

since the first pivot element is zero.

3 – Pivoting

3.1 – The Need for Pivoting

Example. The algorithm would fail at the first step on

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

since the first pivot element is zero. But if we interchange the rows, the system

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

becomes trivial to solve.

Direct Methods for LS

Example. The simple Gaussian elimination algorithm would produce relatively large error on the system

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix},$$

where $\varepsilon < \varepsilon_M$.

Direct Methods for LS

Example. The simple Gaussian elimination algorithm would produce relatively large error on the system

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix},$$

where $\varepsilon < \varepsilon_M$. Algorithm 3 would compute

$$\begin{bmatrix} \varepsilon & 1 \\ 0 & 1-\frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2-\frac{1}{\varepsilon} \end{bmatrix} \Longrightarrow \begin{bmatrix} \varepsilon & 1 \\ 0 & -\frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{1}{\varepsilon} \end{bmatrix},$$

since in the computer, if ε is small enough, $1-\frac{1}{\varepsilon}$ and $2-\frac{1}{\varepsilon}$ will be computed to be the same as $-\frac{1}{\varepsilon}$.

Direct Methods for LS

Example. The simple Gaussian elimination algorithm would produce relatively large error on the system

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix},$$

where $\varepsilon < \varepsilon_M$. Algorithm 3 would compute

$$\begin{bmatrix} \varepsilon & 1 \\ 0 & 1 - \frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 - \frac{1}{\varepsilon} \end{bmatrix} \Longrightarrow \begin{bmatrix} \varepsilon & 1 \\ 0 & -\frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{1}{\varepsilon} \end{bmatrix},$$

since in the computer, if ε is small enough, $1 - \frac{1}{\varepsilon}$ and $2 - \frac{1}{\varepsilon}$ will be computed to be the same as $-\frac{1}{\varepsilon}$. Hence,

$$x_2 = \frac{-\frac{1}{\varepsilon}}{-\frac{1}{\varepsilon}} = 1 \quad \text{and} \quad x_1 = \frac{1-1}{\varepsilon} = 0.$$

Direct Methods for LS

Example. The simple Gaussian elimination algorithm would produce relatively large error on the system

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \end{bmatrix},$$

where $\varepsilon < \varepsilon_M$. Algorithm 3 would compute

$$\begin{bmatrix} \varepsilon & 1 \\ 0 & 1-\frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 2-\frac{1}{\varepsilon} \end{bmatrix} \Longrightarrow \begin{bmatrix} \varepsilon & 1 \\ 0 & -\frac{1}{\varepsilon} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ -\frac{1}{\varepsilon} \end{bmatrix},$$

since in the computer, if ε is small enough, $1-\frac{1}{\varepsilon}$ and $2-\frac{1}{\varepsilon}$ will be computed to be the same as $-\frac{1}{\varepsilon}$. Hence,

$$x_2 = \frac{-\frac{1}{\varepsilon}}{-\frac{1}{\varepsilon}} = 1 \quad \text{and} \quad x_1 = \frac{1-1}{\varepsilon} = 0.$$

$\Rightarrow$

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \neq \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Direct Methods for LS

But actually $x_1 = x_2 = 1$ would be a much better solution since

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+\varepsilon \\ 2 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

Direct Methods for LS

But actually $x_1 = x_2 = 1$ would be a much better solution since

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+\varepsilon \\ 2 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

If we interchange the rows, then Gaussian elimination would compute

$$\begin{bmatrix} 1 & 1 \\ \varepsilon & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1-\varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1-2\varepsilon \end{bmatrix},$$

Direct Methods for LS

But actually $x_1 = x_2 = 1$ would be a much better solution since

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+\varepsilon \\ 2 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

If we interchange the rows, then Gaussian elimination would compute

$$\begin{bmatrix} 1 & 1 \\ \varepsilon & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1-\varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1-2\varepsilon \end{bmatrix},$$

and

$$x_2 = \frac{1-2\epsilon}{1-\varepsilon} \approx 1 \quad \text{and} \quad x_1 = 2 - x_2 \approx 2 - 1 = 1.$$

Direct Methods for LS

But actually $x_1 = x_2 = 1$ would be a much better solution since

$$\begin{bmatrix} \varepsilon & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1+\varepsilon \\ 2 \end{bmatrix} \approx \begin{bmatrix} 1 \\ 2 \end{bmatrix}.$$

If we interchange the rows, then Gaussian elimination would compute

$$\begin{bmatrix} 1 & 1 \\ \varepsilon & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \Longrightarrow \begin{bmatrix} 1 & 1 \\ 0 & 1-\varepsilon \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 2 \\ 1-2\varepsilon \end{bmatrix},$$

and

$$x_2 = \frac{1-2\epsilon}{1-\varepsilon} \approx 1 \quad \text{and} \quad x_1 = 2 - x_2 \approx 2 - 1 = 1.$$

The strategy of interchange rows/columns as described above is called “pivoting”.

Direct Methods for LS

3.2 – Partial Pivoting and Complete Pivoting

If $a_{kk}^{(k)}$ is small in magnitude compared to $a_{ik}^{(k)}$, $i = k+1, \ldots, n$,

3.2 – Partial Pivoting and Complete Pivoting

If $a_{kk}^{(k)}$ is small in magnitude compared to $a_{ik}^{(k)}$, $i = k+1, \ldots, n$, then the multipliers

$$\ell_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \gg 1.$$

3.2 – Partial Pivoting and Complete Pivoting

If $a_{kk}^{(k)}$ is small in magnitude compared to $a_{ik}^{(k)}$, $i = k+1, \ldots, n$, then the multipliers

$$\ell_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \gg 1.$$

Roundoff introduced in computing

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \ell_{ik} a_{kj}^{(k)}, \quad i = k+1, \ldots, n, \quad j = k+1, \ldots, n,$$

will be large.

Direct Methods for LS

3.2 – Partial Pivoting and Complete Pivoting

If $a_{kk}^{(k)}$ is small in magnitude compared to $a_{ik}^{(k)}$, $i = k+1, \ldots, n$, then the multipliers

$$\ell_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \gg 1.$$

Roundoff introduced in computing

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \ell_{ik} a_{kj}^{(k)}, \quad i = k+1, \ldots, n, \quad j = k+1, \ldots, n,$$

will be large. Also when performing the back substitution for

$$x_k = \left(\widetilde{b}_k - \sum_{j=k+1}^{n} a_{kj}^{(k)} x_j \right) \Big/ a_{kk}^{(k)},$$

any error in the numerator will be dramatically increased when dividing by a small $a_{kk}^{(k)}$.

Direct Methods for LS

3.2 – Partial Pivoting and Complete Pivoting

If $a_{kk}^{(k)}$ is small in magnitude compared to $a_{ik}^{(k)}$, $i = k+1, \ldots, n$, then the multipliers

$$\ell_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}} \gg 1.$$

Roundoff introduced in computing

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - \ell_{ik} a_{kj}^{(k)}, \quad i = k+1, \ldots, n, \quad j = k+1, \ldots, n,$$

will be large. Also when performing the back substitution for

$$x_k = \left(\tilde{b}_k - \sum_{j=k+1}^{n} a_{kj}^{(k)} x_j \right) \Big/ a_{kk}^{(k)},$$

any error in the numerator will be dramatically increased when dividing by a small $a_{kk}^{(k)}$.

To ensure that no large entries appear in the computed triangular factors, one can choose a pivot element to be the largest entry among $|a_{kk}^{(k)}|, \ldots, |a_{nk}^{(k)}|$.

Direct Methods for LS

Let $P_1, \ldots, P_{k-1}$ be the permutations chosen and $M_1, \ldots M_{k-1}$ denote the Gaussian transformations performed in the first $k-1$ steps.

Direct Methods for LS

Let $P_1, \ldots, P_{k-1}$ be the permutations chosen and $M_1, \ldots M_{k-1}$ denote the Gaussian transformations performed in the first $k-1$ steps. At the k-th step, a permutation matrix P_k is chosen so that

$$\left|(P_k M_{k-1} \cdots M_1 P_1 A)_{kk}\right| = \max_{k \le i \le n} \left|(M_{k-1} \cdots M_1 P_1 A)_{ik}\right|.$$

Direct Methods for LS

Let $P_1, \ldots, P_{k-1}$ be the permutations chosen and $M_1, \ldots M_{k-1}$ denote the Gaussian transformations performed in the first $k-1$ steps. At the k-th step, a permutation matrix P_k is chosen so that

$$\left|(P_k M_{k-1} \cdots M_1 P_1 A)_{kk}\right| = \max_{k \le i \le n} \left|(M_{k-1} \cdots M_1 P_1 A)_{ik}\right|.$$

This row interchange strategy is called partial pivoting.

Direct Methods for LS

Let $P_1, \ldots, P_{k-1}$ be the permutations chosen and $M_1, \ldots M_{k-1}$ denote the Gaussian transformations performed in the first $k-1$ steps. At the k-th step, a permutation matrix P_k is chosen so that

$$\left|(P_k M_{k-1} \cdots M_1 P_1 A)_{kk}\right| = \max_{k \le i \le n} \left|(M_{k-1} \cdots M_1 P_1 A)_{ik}\right|.$$

This row interchange strategy is called partial pivoting. As a consequence, $|\ell_{ij}| \le 1$ for $i = 1, \ldots, n, j = 1, \ldots, i.$

Direct Methods for LS

Let $P_1, \ldots, P_{k-1}$ be the permutations chosen and $M_1, \ldots M_{k-1}$ denote the Gaussian transformations performed in the first $k-1$ steps. At the k-th step, a permutation matrix P_k is chosen so that

$$|(P_k M_{k-1} \cdots M_1 P_1 A)_{kk}| = \max_{k \le i \le n} |(M_{k-1} \cdots M_1 P_1 A)_{ik}| .$$

This row interchange strategy is called partial pivoting. As a consequence, $|\ell_{ij}| \le 1$ for $i = 1, \ldots, n$, $j = 1, \ldots, i$. Upon completion, we obtain an upper triangular matrix

$$U \equiv M_{n-1} P_{n-1} \cdots M_1 P_1 A. \tag{6}$$

Direct Methods for LS

Let $P_1, \ldots, P_{k-1}$ be the permutations chosen and $M_1, \ldots M_{k-1}$ denote the Gaussian transformations performed in the first $k-1$ steps. At the k-th step, a permutation matrix P_k is chosen so that

$$\left|(P_k M_{k-1} \cdots M_1 P_1 A)_{kk}\right| = \max_{k \le i \le n} \left|(M_{k-1} \cdots M_1 P_1 A)_{ik}\right|.$$

This row interchange strategy is called partial pivoting. As a consequence, $|\ell_{ij}| \le 1$ for $i = 1, \ldots, n$, $j = 1, \ldots, i$. Upon completion, we obtain an upper triangular matrix

$$U \equiv M_{n-1} P_{n-1} \cdots M_1 P_1 A. \tag{6}$$

Since any P_k is symmetric and $P_k^T P_k = P_k^2 = I$,

Direct Methods for LS

Let $P_1, \ldots, P_{k-1}$ be the permutations chosen and $M_1, \ldots M_{k-1}$ denote the Gaussian transformations performed in the first $k-1$ steps. At the k-th step, a permutation matrix P_k is chosen so that

$$|(P_k M_{k-1} \cdots M_1 P_1 A)_{kk}| = \max_{k \le i \le n} |(M_{k-1} \cdots M_1 P_1 A)_{ik}| .$$

This row interchange strategy is called partial pivoting. As a consequence, $|\ell_{ij}| \le 1$ for $i = 1, \ldots, n, j = 1, \ldots, i$. Upon completion, we obtain an upper triangular matrix

$$U \equiv M_{n-1} P_{n-1} \cdots M_1 P_1 A. \qquad (6)$$

Since any P_k is symmetric and $P_k^T P_k = P_k^2 = I$, we have

$$M_{n-1} P_{n-1} \cdots M_2 P_2 M_1 P_2 \cdots P_{n-1} P_{n-1} \cdots P_2 P_1 A = U,$$

Direct Methods for LS

Let $P_1, \ldots, P_{k-1}$ be the permutations chosen and $M_1, \ldots M_{k-1}$ denote the Gaussian transformations performed in the first $k-1$ steps. At the k-th step, a permutation matrix P_k is chosen so that

$$\left|(P_k M_{k-1} \cdots M_1 P_1 A)_{kk}\right| = \max_{k \le i \le n} \left|(M_{k-1} \cdots M_1 P_1 A)_{ik}\right|.$$

This row interchange strategy is called partial pivoting. As a consequence, $|\ell_{ij}| \le 1$ for $i = 1, \ldots, n$, $j = 1, \ldots, i$. Upon completion, we obtain an upper triangular matrix

$$U \equiv M_{n-1} P_{n-1} \cdots M_1 P_1 A. \tag{6}$$

Since any P_k is symmetric and $P_k^T P_k = P_k^2 = I$, we have

$$M_{n-1} P_{n-1} \cdots M_2 P_2 M_1 P_2 \cdots P_{n-1} P_{n-1} \cdots P_2 P_1 A = U,$$

therefore,

$$P_{n-1} \cdots P_1 A = (M_{n-1} P_{n-1} \cdots M_2 P_2 M_1 P_2 \cdots P_{n-1})^{-1} U.$$

Direct Methods for LS

In summary, Gaussian elimination with partial pivoting leads to the LU factorization

$$PA = LU, \tag{7}$$

where

$$P = P_{n-1} \cdots P_1$$

is a permutation matrix, and

$$\begin{aligned} L &\equiv (M_{n-1}P_{n-1} \cdots M_2 P_2 M_1 P_2 \cdots P_{n-1})^{-1} \\ &= P_{n-1} \cdots P_2 M_1^{-1} P_2 M_2^{-1} \cdots P_{n-1} M_{n-1}^{-1}. \end{aligned}$$

Direct Methods for LS

In summary, Gaussian elimination with partial pivoting leads to the LU factorization

$$PA = LU, \tag{7}$$

where

$$P = P_{n-1} \cdots P_1$$

is a permutation matrix, and

$$\begin{aligned} L &\equiv (M_{n-1}P_{n-1} \cdots M_2P_2M_1P_2 \cdots P_{n-1})^{-1} \\ &= P_{n-1} \cdots P_2M_1^{-1}P_2M_2^{-1} \cdots P_{n-1}M_{n-1}^{-1}. \end{aligned}$$

Since, for $i < j$,

$$e_i^T P_j = e_i^T, \quad e_i^T \ell_j = 0,$$

$$P_j \ell_i = \begin{bmatrix} 0 & \cdots & 0 & \tilde{\ell}_{i+1,i} & \cdots & \tilde{\ell}_{n,i} \end{bmatrix}^T \equiv \tilde{\ell}_i,$$

Direct Methods for LS

In summary, Gaussian elimination with partial pivoting leads to the LU factorization

$$PA = LU, \tag{7}$$

where

$$P = P_{n-1} \cdots P_1$$

is a permutation matrix, and

$$\begin{aligned} L &\equiv (M_{n-1}P_{n-1} \cdots M_2 P_2 M_1 P_2 \cdots P_{n-1})^{-1} \\ &= P_{n-1} \cdots P_2 M_1^{-1} P_2 M_2^{-1} \cdots P_{n-1} M_{n-1}^{-1}. \end{aligned}$$

Since, for $i < j$,

$$e_i^T P_j = e_i^T, \quad e_i^T \ell_j = 0,$$

$$P_j \ell_i = \begin{bmatrix} 0 & \cdots & 0 & \tilde{\ell}_{i+1,i} & \cdots & \tilde{\ell}_{n,i} \end{bmatrix}^T \equiv \tilde{\ell}_i,$$

$\Rightarrow$

$$P_2 M_1^{-1} P_2 = P_2(I + \ell_1 e_1^T) P_2 = I + \tilde{\ell}_1 e_1^T$$

Direct Methods for LS

$\Rightarrow$

$$P_2 M_1^{-1} P_2 M_2^{-1} = (I + \tilde{\ell}_1 e_1^T)(I + \ell_2 e_2^T) = I + \tilde{\ell}_1 e_1^T + \ell_2 e_2^T,$$

Direct Methods for LS

$\Rightarrow$

$$P_2M_1^{-1}P_2M_2^{-1} = (I + \tilde{\ell}_1e_1^T)(I + \ell_2e_2^T) = I + \tilde{\ell}_1e_1^T + \ell_2e_2^T,$$

$\Rightarrow$

$$P_3\left(P_2M_1^{-1}P_2M_2^{-1}\right)P_3 = I + \hat{\ell}_1e_1^T + \tilde{\ell}_2e_2^T$$

Direct Methods for LS

$\Rightarrow$

$$P_2 M_1^{-1} P_2 M_2^{-1} = (I + \tilde{\ell}_1 e_1^T)(I + \ell_2 e_2^T) = I + \tilde{\ell}_1 e_1^T + \ell_2 e_2^T,$$

$\Rightarrow$

$$P_3 \left(P_2 M_1^{-1} P_2 M_2^{-1} \right) P_3 = I + \hat{\ell}_1 e_1^T + \tilde{\ell}_2 e_2^T$$

$\Rightarrow \cdots$

Therefore, L is unit lower triangular.

Direct Methods for LS

$\Rightarrow$

$$P_2 M_1^{-1} P_2 M_2^{-1} = (I + \tilde{\ell}_1 e_1^T)(I + \ell_2 e_2^T) = I + \tilde{\ell}_1 e_1^T + \ell_2 e_2^T,$$

$\Rightarrow$

$$P_3 \left(P_2 M_1^{-1} P_2 M_2^{-1} \right) P_3 = I + \hat{\ell}_1 e_1^T + \tilde{\ell}_2 e_2^T$$

$\Rightarrow \cdots$

Therefore, L is unit lower triangular.

Algorithm 5 *[LU-factorization with Partial Pivoting]*

Direct Methods for LS

$\Rightarrow$

$$P_2 M_1^{-1} P_2 M_2^{-1} = (I + \tilde{\ell}_1 e_1^T)(I + \ell_2 e_2^T) = I + \tilde{\ell}_1 e_1^T + \ell_2 e_2^T,$$

$\Rightarrow$

$$P_3 \left(P_2 M_1^{-1} P_2 M_2^{-1} \right) P_3 = I + \hat{\ell}_1 e_1^T + \tilde{\ell}_2 e_2^T$$

$\Rightarrow \cdots$

Therefore, L is unit lower triangular.

Algorithm 5 *[LU-factorization with Partial Pivoting] Given a nonsingular square matrix $A \in \mathbb{R}^{n \times n}$, this algorithm finds an appropriate permutation matrix P,*

Direct Methods for LS

$\Rightarrow$

$$P_2 M_1^{-1} P_2 M_2^{-1} = (I + \tilde{\ell}_1 e_1^T)(I + \ell_2 e_2^T) = I + \tilde{\ell}_1 e_1^T + \ell_2 e_2^T,$$

$\Rightarrow$

$$P_3 \left(P_2 M_1^{-1} P_2 M_2^{-1}\right) P_3 = I + \hat{\ell}_1 e_1^T + \tilde{\ell}_2 e_2^T$$

$\Rightarrow \cdots$

Therefore, L is unit lower triangular.

Algorithm 5 *[LU-factorization with Partial Pivoting] Given a nonsingular square matrix $A \in \mathbb{R}^{n \times n}$, this algorithm finds an appropriate permutation matrix P, and computes a unit lower triangular matrix L*

Direct Methods for LS

$\Rightarrow$

$$P_2 M_1^{-1} P_2 M_2^{-1} = (I + \tilde{\ell}_1 e_1^T)(I + \ell_2 e_2^T) = I + \tilde{\ell}_1 e_1^T + \ell_2 e_2^T,$$

$\Rightarrow$

$$P_3 \left(P_2 M_1^{-1} P_2 M_2^{-1} \right) P_3 = I + \hat{\ell}_1 e_1^T + \tilde{\ell}_2 e_2^T$$

$\Rightarrow \cdots$

Therefore, L is unit lower triangular.

Algorithm 5 *[LU-factorization with Partial Pivoting] Given a nonsingular square matrix $A \in \mathbb{R}^{n \times n}$, this algorithm finds an appropriate permutation matrix P, and computes a unit lower triangular matrix L and an upper triangular matrix U*

Direct Methods for LS

$\Rightarrow$

$$P_2 M_1^{-1} P_2 M_2^{-1} = (I + \tilde{\ell}_1 e_1^T)(I + \ell_2 e_2^T) = I + \tilde{\ell}_1 e_1^T + \ell_2 e_2^T,$$

$\Rightarrow$

$$P_3 \left(P_2 M_1^{-1} P_2 M_2^{-1}\right) P_3 = I + \hat{\ell}_1 e_1^T + \tilde{\ell}_2 e_2^T$$

$\Rightarrow \cdots$

Therefore, L is unit lower triangular.

Algorithm 5 *[LU-factorization with Partial Pivoting] Given a nonsingular square matrix $A \in \mathbb{R}^{n \times n}$, this algorithm finds an appropriate permutation matrix P, and computes a unit lower triangular matrix L and an upper triangular matrix U such that $PA = LU$.*

Direct Methods for LS

$\Rightarrow$

$$P_2M_1^{-1}P_2M_2^{-1} = (I+\tilde{\ell}_1e_1^T)(I+\ell_2e_2^T) = I+\tilde{\ell}_1e_1^T+\ell_2e_2^T,$$

$\Rightarrow$

$$P_3\left(P_2M_1^{-1}P_2M_2^{-1}\right)P_3 = I+\hat{\ell}_1e_1^T+\tilde{\ell}_2e_2^T$$

$\Rightarrow \cdots$

Therefore, L is unit lower triangular.

Algorithm 5 *[LU-factorization with Partial Pivoting] Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm finds an appropriate permutation matrix P, and computes a unit lower triangular matrix L and an upper triangular matrix U such that $PA = LU$. The matrix A is overwritten by L and U,*

Direct Methods for LS

$\Rightarrow$

$$P_2 M_1^{-1} P_2 M_2^{-1} = (I + \tilde{\ell}_1 e_1^T)(I + \ell_2 e_2^T) = I + \tilde{\ell}_1 e_1^T + \ell_2 e_2^T,$$

$\Rightarrow$

$$P_3 \left(P_2 M_1^{-1} P_2 M_2^{-1}\right) P_3 = I + \hat{\ell}_1 e_1^T + \tilde{\ell}_2 e_2^T$$

$\Rightarrow \cdots$

Therefore, L is unit lower triangular.

Algorithm 5 *[LU-factorization with Partial Pivoting] Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm finds an appropriate permutation matrix P, and computes a unit lower triangular matrix L and an upper triangular matrix U such that $PA = LU$. The matrix A is overwritten by L and U, and the matrix P is not formed.*

Direct Methods for LS

$\Rightarrow$

$$P_2 M_1^{-1} P_2 M_2^{-1} = (I + \tilde{\ell}_1 e_1^T)(I + \ell_2 e_2^T) = I + \tilde{\ell}_1 e_1^T + \ell_2 e_2^T,$$

$\Rightarrow$

$$P_3 \left(P_2 M_1^{-1} P_2 M_2^{-1}\right) P_3 = I + \hat{\ell}_1 e_1^T + \tilde{\ell}_2 e_2^T$$

$\Rightarrow \cdots$

Therefore, L is unit lower triangular.

Algorithm 5 *[LU-factorization with Partial Pivoting] Given a nonsingular square matrix $A \in \mathbb{R}^{n\times n}$, this algorithm finds an appropriate permutation matrix P, and computes a unit lower triangular matrix L and an upper triangular matrix U such that $PA = LU$. The matrix A is overwritten by L and U, and the matrix P is not formed. An integer array p is instead used for storing the row/column indices.*

Direct Methods for LS

$$p(1:n) = 1:n$$

Direct Methods for LS

```
$p(1:n) = 1:n$
For $k = 1, \ldots, n-1$

End For
```

Direct Methods for LS

```
$p(1:n) = 1:n$
For $k = 1, \ldots, n-1$
    $m = k$
    For $i = k+1, \ldots, n$


    End For


End For
```

Direct Methods for LS

```
$p(1:n) = 1:n$
For $k = 1, \ldots, n-1$
    $m = k$
    For $i = k+1, \ldots, n$
        If $|A(p(m),k)| < |A(p(i),k)|$, then $m = i$
    End For



End For
```

Direct Methods for LS

```
$p(1:n) = 1:n$
For $k = 1, \ldots, n-1$
    $m = k$
    For $i = k+1, \ldots, n$
        If $|A(p(m),k)| < |A(p(i),k)|$, then $m = i$
    End For
    $\ell = p(k); p(k) = p(m); p(m) = \ell$
End For
```

Direct Methods for LS

```
$p(1:n) = 1:n$
For $k = 1, \ldots, n-1$
    $m = k$
    For $i = k+1, \ldots, n$
        If $|A(p(m),k)| < |A(p(i),k)|$, then $m = i$
    End For
    $\ell = p(k); p(k) = p(m); p(m) = \ell$
    For $i = k+1, \ldots, n$



    End For
End For
```

Direct Methods for LS

```
$p(1:n) = 1:n$
For $k = 1, \ldots, n-1$
    $m = k$
    For $i = k+1, \ldots, n$
        If $|A(p(m),k)| < |A(p(i),k)|$, then $m = i$
    End For
    $\ell = p(k); p(k) = p(m); p(m) = \ell$
    For $i = k+1, \ldots, n$
        $A(p(i),k) = A(p(i),k)/A(p(k),k)$



    End For
End For
```

Direct Methods for LS

```
$p(1:n) = 1:n$
For $k = 1, \ldots, n-1$
    $m = k$
    For $i = k+1, \ldots, n$
        If $|A(p(m), k)| < |A(p(i), k)|$, then $m = i$
    End For
    $\ell = p(k); p(k) = p(m); p(m) = \ell$
    For $i = k+1, \ldots, n$
        $A(p(i), k) = A(p(i), k)/A(p(k), k)$
        For $j = k+1, \ldots, n$


        End For
    End For
End For
```

Direct Methods for LS

```
$p(1:n) = 1:n$
For $k = 1, \dots, n-1$
    $m = k$
    For $i = k+1, \dots, n$
        If $|A(p(m),k)| < |A(p(i),k)|$, then $m = i$
    End For
    $\ell = p(k); p(k) = p(m); p(m) = \ell$
    For $i = k+1, \dots, n$
        $A(p(i),k) = A(p(i),k)/A(p(k),k)$
        For $j = k+1, \dots, n$
            $A(p(i),j) = A(p(i),j) - A(p(i),k)A(p(k),j)$
        End For
    End For
End For
```

Direct Methods for LS

Since the Gaussian elimination with partial pivoting produces the factorization (7), the linear system problem should comply accordingly

$$Ax = b \Longrightarrow PAx = Pb \Longrightarrow LUx = Pb.$$

Direct Methods for LS

Since the Gaussian elimination with partial pivoting produces the factorization (7), the linear system problem should comply accordingly

$$Ax = b \Longrightarrow PAx = Pb \Longrightarrow LUx = Pb.$$

While in the partial pivoting algorithm, the k-th pivot is determined by scanning the current k-th subcolumn $A^{(k)}(k:n,k)$,

Direct Methods for LS

Since the Gaussian elimination with partial pivoting produces the factorization (7), the linear system problem should comply accordingly

$$Ax = b \Longrightarrow PAx = Pb \Longrightarrow LUx = Pb.$$

While in the partial pivoting algorithm, the k-th pivot is determined by scanning the current k-th subcolumn $A^{(k)}(k:n,k)$, another pivoting strategy called complete pivoting searches for the largest entry in magnitude in the current submatrix $A^{(k)}(k:n,k:n)$

Direct Methods for LS

Since the Gaussian elimination with partial pivoting produces the factorization (7), the linear system problem should comply accordingly

$$Ax = b \Longrightarrow PAx = Pb \Longrightarrow LUx = Pb.$$

While in the partial pivoting algorithm, the k-th pivot is determined by scanning the current k-th subcolumn $A^{(k)}(k:n,k)$, another pivoting strategy called complete pivoting searches for the largest entry in magnitude in the current submatrix $A^{(k)}(k:n,k:n)$ and permutes to the (k,k) position.

Direct Methods for LS

Since the Gaussian elimination with partial pivoting produces the factorization (7), the linear system problem should comply accordingly

$$Ax = b \Longrightarrow PAx = Pb \Longrightarrow LUx = Pb.$$

While in the partial pivoting algorithm, the k-th pivot is determined by scanning the current k-th subcolumn $A^{(k)}(k:n,k)$, another pivoting strategy called complete pivoting searches for the largest entry in magnitude in the current submatrix $A^{(k)}(k:n,k:n)$ and permutes to the (k,k) position. That is, at the k-th step two permutation matrices P_k and Q_k are determined

Direct Methods for LS

Since the Gaussian elimination with partial pivoting produces the factorization (7), the linear system problem should comply accordingly

$$Ax = b \Longrightarrow PAx = Pb \Longrightarrow LUx = Pb.$$

While in the partial pivoting algorithm, the k-th pivot is determined by scanning the current k-th subcolumn $A^{(k)}(k:n,k)$, another pivoting strategy called complete pivoting searches for the largest entry in magnitude in the current submatrix $A^{(k)}(k:n,k:n)$ and permutes to the (k,k) position. That is, at the k-th step two permutation matrices P_k and Q_k are determined so that

$$\left|(P_k A^{(k)} Q_k)_{kk}\right| = \max_{k \le i,j \le n} \left|(A^{(k)})_{ij}\right|.$$

Direct Methods for LS

Since the Gaussian elimination with partial pivoting produces the factorization (7), the linear system problem should comply accordingly

$$Ax = b \Longrightarrow PAx = Pb \Longrightarrow LUx = Pb.$$

While in the partial pivoting algorithm, the k-th pivot is determined by scanning the current k-th subcolumn $A^{(k)}(k:n,k)$, another pivoting strategy called complete pivoting searches for the largest entry in magnitude in the current submatrix $A^{(k)}(k:n,k:n)$ and permutes to the (k,k) position. That is, at the k-th step two permutation matrices P_k and Q_k are determined so that

$$\left|(P_k A^{(k)} Q_k)_{kk}\right| = \max_{k \leq i,j \leq n} \left|(A^{(k)})_{ij}\right|.$$

Gaussian elimination with complete pivoting leads to the LU factorization

$$PAQ = LU, \tag{8}$$

Direct Methods for LS

Since the Gaussian elimination with partial pivoting produces the factorization (7), the linear system problem should comply accordingly

$$Ax = b \Longrightarrow PAx = Pb \Longrightarrow LUx = Pb.$$

While in the partial pivoting algorithm, the k-th pivot is determined by scanning the current k-th subcolumn $A^{(k)}(k:n,k)$, another pivoting strategy called complete pivoting searches for the largest entry in magnitude in the current submatrix $A^{(k)}(k:n,k:n)$ and permutes to the (k,k) position. That is, at the k-th step two permutation matrices P_k and Q_k are determined so that

$$\left|(P_k A^{(k)} Q_k)_{kk}\right| = \max_{k \le i,j \le n} \left|(A^{(k)})_{ij}\right|.$$

Gaussian elimination with complete pivoting leads to the LU factorization

$$PAQ = LU, \tag{8}$$

where P, Q are permutation matrices, L is unit lower triangular, and U is upper triangular.

4 – Some Special Linear Systems

4.1 – Symmetric Positive Definite System and Cholesky Factorization

An $n \times n$ matrix A is positive definite

4 – Some Special Linear Systems

4.1 – Symmetric Positive Definite System and Cholesky Factorization

An $n \times n$ matrix A is positive definite if $x^T Ax > 0$, for all $x \in \mathbb{R}^n$, $x \neq 0$.

4 – Some Special Linear Systems

4.1 – Symmetric Positive Definite System and Cholesky Factorization

An $n \times n$ matrix A is positive definite if $x^T Ax > 0$, for all $x \in \mathbb{R}^n$, $x \neq 0$. If A is both symmetric and positive definite (*spd*),

4 – Some Special Linear Systems

4.1 – Symmetric Positive Definite System and Cholesky Factorization

An $n \times n$ matrix A is positive definite if $x^T Ax > 0$, for all $x \in \mathbb{R}^n$, $x \neq 0$. If A is both symmetric and positive definite (*spd*), then we can derive a stable LU factorization called the Choleseky factorization.

4 – Some Special Linear Systems

4.1 – Symmetric Positive Definite System and Cholesky Factorization

An $n \times n$ matrix A is positive definite if $x^T Ax > 0$, for all $x \in \mathbb{R}^n$, $x \neq 0$. If A is both symmetric and positive definite (*spd*), then we can derive a stable LU factorization called the Choleseky factorization.

Lemma 1 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then A is nonsingular and $a_{ii} > 0$ for $i = 1, \ldots, n$.*

4 – Some Special Linear Systems

4.1 – Symmetric Positive Definite System and Cholesky Factorization

An $n \times n$ matrix A is positive definite if $x^T Ax > 0$, for all $x \in \mathbb{R}^n$, $x \neq 0$. If A is both symmetric and positive definite (*spd*), then we can derive a stable LU factorization called the Choleseky factorization.

Lemma 1 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then A is nonsingular and $a_{ii} > 0$ for $i = 1, \dots, n$.*

Proof: Suppose A is singular.

4 – Some Special Linear Systems

4.1 – Symmetric Positive Definite System and Cholesky Factorization

An $n \times n$ matrix A is positive definite if $x^T Ax > 0$, for all $x \in \mathbb{R}^n$, $x \neq 0$. If A is both symmetric and positive definite (*spd*), then we can derive a stable LU factorization called the Choleseky factorization.

Lemma 1 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then A is nonsingular and $a_{ii} > 0$ for $i = 1, \ldots, n$.*

Proof: Suppose A is singular.

$\Rightarrow \exists\, x \in \mathbb{R}^n$ and $x \neq 0$ such that $Ax = 0$.

4 – Some Special Linear Systems

4.1 – Symmetric Positive Definite System and Cholesky Factorization

An $n \times n$ matrix A is positive definite if $x^T Ax > 0$, for all $x \in \mathbb{R}^n$, $x \neq 0$. If A is both symmetric and positive definite (*spd*), then we can derive a stable LU factorization called the Choleseky factorization.

Lemma 1 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then A is nonsingular and $a_{ii} > 0$ for $i = 1, \ldots, n$.*

Proof: Suppose A is singular.
$\Rightarrow \exists\, x \in \mathbb{R}^n$ and $x \neq 0$ such that $Ax = 0$.
$\Rightarrow x^T Ax = 0$, which contradicts the fact that A is positive definite.

4 – Some Special Linear Systems

4.1 – Symmetric Positive Definite System and Cholesky Factorization

An $n \times n$ matrix A is positive definite if $x^T Ax > 0$, for all $x \in \mathbb{R}^n$, $x \neq 0$. If A is both symmetric and positive definite (*spd*), then we can derive a stable LU factorization called the Choleseky factorization.

Lemma 1 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then A is nonsingular and $a_{ii} > 0$ for $i = 1, \dots, n$.*

Proof: Suppose A is singular.

$\Rightarrow \exists\, x \in \mathbb{R}^n$ and $x \neq 0$ such that $Ax = 0$.

$\Rightarrow x^T Ax = 0$, which contradicts the fact that A is positive definite.

$\Rightarrow A$ is nonsingular.

Direct Methods for LS

Since A is positive definite,

Direct Methods for LS

Since A is positive definite,

$$a_{ii} = e_i^T A e_i > 0,$$

where e_i is the i-th column of the $n \times n$ identify matrix. ■

Direct Methods for LS

Since A is positive definite,

$$a_{ii} = e_i^T A e_i > 0,$$

where e_i is the i-th column of the $n \times n$ identify matrix. ■

Lemma 2 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then all leading principal submatrices of A are nonsingular.*

Direct Methods for LS

Since A is positive definite,

$$a_{ii} = e_i^T A e_i > 0,$$

where e_i is the i-th column of the $n \times n$ identify matrix. ■

Lemma 2 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then all leading principal submatrices of A are nonsingular.*

Proof: For $1 \le k \le n$, let

$$z_k = [x_1, \ldots, x_k]^T \in \mathbb{R}^k \text{ and } x = [x_1, \ldots, x_k, 0, \ldots, 0]^T \in \mathbb{R}^n,$$

where $x_1, \ldots, x_k \in \mathbb{R}$ are not all zero.

Direct Methods for LS

Since A is positive definite,

$$a_{ii} = e_i^T A e_i > 0,$$

where e_i is the i-th column of the $n \times n$ identify matrix. ■

Lemma 2 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then all leading principal submatrices of A are nonsingular.*

Proof: For $1 \leq k \leq n$, let

$$z_k = [x_1, \ldots, x_k]^T \in \mathbb{R}^k \text{ and } x = [x_1, \ldots, x_k, 0, \ldots, 0]^T \in \mathbb{R}^n,$$

where $x_1, \ldots, x_k \in \mathbb{R}$ are not all zero. Since A is positive definite,

Direct Methods for LS

Since A is positive definite,

$$a_{ii} = e_i^T A e_i > 0,$$

where e_i is the i-th column of the $n \times n$ identify matrix. $\blacksquare$

Lemma 2 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then all leading principal submatrices of A are nonsingular.*

Proof: For $1 \leq k \leq n$, let

$$z_k = [x_1, \ldots, x_k]^T \in \mathbb{R}^k \text{ and } x = [x_1, \ldots, x_k, 0, \ldots, 0]^T \in \mathbb{R}^n,$$

where $x_1, \ldots, x_k \in \mathbb{R}$ are not all zero. Since A is positive definite,

$$z_k^T A_k z_k = x^T A x > 0,$$

where A_k is the $k \times k$ leading principal submatrix of A.

Direct Methods for LS

Since A is positive definite,

$$a_{ii} = e_i^T A e_i > 0,$$

where e_i is the i-th column of the $n \times n$ identify matrix. ■

Lemma 2 *If $A \in \mathbb{R}^{n\times n}$ is positive definite, then all leading principal submatrices of A are nonsingular.*

Proof: For $1 \le k \le n$, let

$$z_k = [x_1, \ldots, x_k]^T \in \mathbb{R}^k \text{ and } x = [x_1, \ldots, x_k, 0, \ldots, 0]^T \in \mathbb{R}^n,$$

where $x_1, \ldots, x_k \in \mathbb{R}$ are not all zero. Since A is positive definite,

$$z_k^T A_k z_k = x^T A x > 0,$$

where A_k is the $k \times k$ leading principal submatrix of A. This shows that A_k are also positive definite, hence A_k are nonsingular. ■

Direct Methods for LS

Theorem 3 *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that A has the factorization*

$$A = GG^T. \tag{9}$$

Direct Methods for LS

Theorem 3 *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that A has the factorization*

$$A = GG^T. \tag{9}$$

Proof: A is positive definite

Direct Methods for LS

Theorem 3 *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that A has the factorization*

$$A = GG^T. \tag{9}$$

Proof: A is positive definite

$\Rightarrow$ all leading principal submatrices of A are nonsingular (from Lemma 2)

Direct Methods for LS

Theorem 3 *If* $A \in \mathbb{R}^{n\times n}$ *is symmetric positive definite, then there exists a unique lower triangular matrix* $G \in \mathbb{R}^{n\times n}$ *with positive diagonal entries such that* A *has the factorization*

$$A = GG^T. \tag{9}$$

Proof: A is positive definite

$\Rightarrow$ all leading principal submatrices of A are nonsingular (from Lemma 2)

$\Rightarrow$ A has the LU factorization $A = LU$, where L is unit lower triangular and U is upper triangular.

Direct Methods for LS

Theorem 3 *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that A has the factorization*

$$A = GG^T. \tag{9}$$

Proof: A is positive definite

$\Rightarrow$ all leading principal submatrices of A are nonsingular (from Lemma 2)

$\Rightarrow$ A has the LU factorization $A = LU$, where L is unit lower triangular and U is upper triangular.

Since A is symmetric,

Direct Methods for LS

Theorem 3 *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that A has the factorization*

$$A = GG^T. \tag{9}$$

Proof: A is positive definite

$\Rightarrow$ all leading principal submatrices of A are nonsingular (from Lemma 2)

$\Rightarrow$ A has the LU factorization $A = LU$, where L is unit lower triangular and U is upper triangular.

Since A is symmetric,

$$LU = A = A^T = U^T L^T$$

Direct Methods for LS

Theorem 3 *If $A \in \mathbb{R}^{n \times n}$ is symmetric positive definite, then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n \times n}$ with positive diagonal entries such that A has the factorization*

$$A = GG^T. \tag{9}$$

Proof: A is positive definite

$\Rightarrow$ all leading principal submatrices of A are nonsingular (from Lemma 2)

$\Rightarrow$ A has the LU factorization $A = LU$, where L is unit lower triangular and U is upper triangular.

Since A is symmetric,

$$LU = A = A^T = U^T L^T \quad \Longrightarrow \quad U(L^T)^{-1} = L^{-1} U^T.$$

Direct Methods for LS

Theorem 3 *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that A has the factorization*

$$A = GG^T. \tag{9}$$

Proof: A is positive definite

$\Rightarrow$ all leading principal submatrices of A are nonsingular (from Lemma 2)

$\Rightarrow$ A has the LU factorization $A = LU$, where L is unit lower triangular and U is upper triangular.

Since A is symmetric,

$$LU = A = A^T = U^T L^T \quad \Longrightarrow \quad U(L^T)^{-1} = L^{-1}U^T.$$

$U(L^T)^{-1}$ is upper triangular and $L^{-1}U^T$ is lower triangular

Direct Methods for LS

Theorem 3 *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that A has the factorization*

$$A = GG^T. \tag{9}$$

Proof: A is positive definite

$\Rightarrow$ all leading principal submatrices of A are nonsingular (from Lemma 2)

$\Rightarrow$ A has the LU factorization $A = LU$, where L is unit lower triangular and U is upper triangular.

Since A is symmetric,

$$LU = A = A^T = U^T L^T \quad \Longrightarrow \quad U(L^T)^{-1} = L^{-1}U^T.$$

$U(L^T)^{-1}$ is upper triangular and $L^{-1}U^T$ is lower triangular

$\Rightarrow U(L^T)^{-1}$ to be a diagonal matrix, say, $U(L^T)^{-1} = D$.

Direct Methods for LS

Theorem 3 *If $A \in \mathbb{R}^{n\times n}$ is symmetric positive definite, then there exists a unique lower triangular matrix $G \in \mathbb{R}^{n\times n}$ with positive diagonal entries such that A has the factorization*

$$A = GG^T. \tag{9}$$

Proof: A is positive definite

$\Rightarrow$ all leading principal submatrices of A are nonsingular (from Lemma 2)

$\Rightarrow$ A has the LU factorization $A = LU$, where L is unit lower triangular and U is upper triangular.

Since A is symmetric,

$$LU = A = A^T = U^T L^T \quad \Longrightarrow \quad U(L^T)^{-1} = L^{-1}U^T.$$

$U(L^T)^{-1}$ is upper triangular and $L^{-1}U^T$ is lower triangular

$\Rightarrow U(L^T)^{-1}$ to be a diagonal matrix, say, $U(L^T)^{-1} = D$.

$\Rightarrow U = DL^T$. Hence

$$A = LDL^T.$$

Direct Methods for LS

Since A is positive definite,

Direct Methods for LS

Since A is positive definite,

$$x^T A x > 0$$

Direct Methods for LS

Since A is positive definite,

$$x^T A x > 0 \quad \Longrightarrow \quad x^T L D L^T x = (L^T x)^T D (L^T x) > 0.$$

Direct Methods for LS

Since A is positive definite,

$$x^T A x > 0 \quad \Longrightarrow \quad x^T L D L^T x = (L^T x)^T D (L^T x) > 0.$$

This means D is also positive definite,

Direct Methods for LS

Since A is positive definite,

$$x^T Ax > 0 \quad \Longrightarrow \quad x^T LDL^T x = (L^T x)^T D(L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$.

Direct Methods for LS

Since A is positive definite,

$$x^T Ax > 0 \quad \Longrightarrow \quad x^T LDL^T x = (L^T x)^T D(L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$. Thus $D^{1/2}$ is well-defined

Direct Methods for LS

Since A is positive definite,

$$x^T A x > 0 \quad \Longrightarrow \quad x^T L D L^T x = (L^T x)^T D (L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$. Thus $D^{1/2}$ is well-defined and we have

$$A = LDL^T = LD^{1/2}D^{1/2}L^T \equiv GG^T,$$

where $G \equiv LD^{1/2}$.

Direct Methods for LS

Since A is positive definite,

$$x^T Ax > 0 \quad \Longrightarrow \quad x^T LDL^T x = (L^T x)^T D(L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$. Thus $D^{1/2}$ is well-defined and we have

$$A = LDL^T = LD^{1/2}D^{1/2}L^T \equiv GG^T,$$

where $G \equiv LD^{1/2}$. Since the LU factorization is unique,

Direct Methods for LS

Since A is positive definite,

$$x^T Ax > 0 \quad \Longrightarrow \quad x^T LDL^T x = (L^T x)^T D(L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$. Thus $D^{1/2}$ is well-defined and we have

$$A = LDL^T = LD^{1/2}D^{1/2}L^T \equiv GG^T,$$

where $G \equiv LD^{1/2}$. Since the LU factorization is unique, G is unique. $\blacksquare$

Direct Methods for LS

Since A is positive definite,

$$x^T A x > 0 \quad \Longrightarrow \quad x^T LDL^T x = (L^T x)^T D (L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$. Thus $D^{1/2}$ is well-defined and we have

$$A = LDL^T = LD^{1/2} D^{1/2} L^T \equiv GG^T,$$

where $G \equiv LD^{1/2}$. Since the LU factorization is unique, G is unique. ■

The factorization (9) is referred to as the Cholesky factorization.

Direct Methods for LS

Since A is positive definite,

$$x^T Ax > 0 \quad \Longrightarrow \quad x^T LDL^T x = (L^T x)^T D(L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$. Thus $D^{1/2}$ is well-defined and we have

$$A = LDL^T = LD^{1/2}D^{1/2}L^T \equiv GG^T,$$

where $G \equiv LD^{1/2}$. Since the LU factorization is unique, G is unique. $\blacksquare$

The factorization (9) is referred to as the Cholesky factorization.

Derive an algorithm for computing the Cholesky factorization:

Direct Methods for LS

Since A is positive definite,

$$x^T Ax > 0 \quad \Longrightarrow \quad x^T LDL^T x = (L^T x)^T D(L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$. Thus $D^{1/2}$ is well-defined and we have

$$A = LDL^T = LD^{1/2}D^{1/2}L^T \equiv GG^T,$$

where $G \equiv LD^{1/2}$. Since the LU factorization is unique, G is unique. ■

The factorization (9) is referred to as the Cholesky factorization.

Derive an algorithm for computing the Cholesky factorization:

Assume the first $k-1$ columns of G have been determined after $k-1$ steps.

Direct Methods for LS

Since A is positive definite,

$$x^T Ax > 0 \quad \Longrightarrow \quad x^T LDL^T x = (L^T x)^T D(L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$. Thus $D^{1/2}$ is well-defined and we have

$$A = LDL^T = LD^{1/2}D^{1/2}L^T \equiv GG^T,$$

where $G \equiv LD^{1/2}$. Since the LU factorization is unique, G is unique. ■

The factorization (9) is referred to as the Cholesky factorization.

Derive an algorithm for computing the Cholesky factorization:

Assume the first $k-1$ columns of G have been determined after $k-1$ steps. By componentwise comparison with equation (9),

Direct Methods for LS

Since A is positive definite,

$$x^T A x > 0 \quad \Longrightarrow \quad x^T L D L^T x = (L^T x)^T D (L^T x) > 0.$$

This means D is also positive definite, and hence $d_{ii} > 0$. Thus $D^{1/2}$ is well-defined and we have

$$A = LDL^T = LD^{1/2}D^{1/2}L^T \equiv GG^T,$$

where $G \equiv LD^{1/2}$. Since the LU factorization is unique, G is unique. ■

The factorization (9) is referred to as the Cholesky factorization.

Derive an algorithm for computing the Cholesky factorization:

Assume the first $k-1$ columns of G have been determined after $k-1$ steps. By componentwise comparison with equation (9), one has

$$a_{kk} = \sum_{j=1}^{k} g_{kj}^2,$$

Direct Methods for LS

which gives

$$g_{kk}^2 = a_{kk} - \sum_{j=1}^{k-1} g_{kj}^2. \tag{10}$$

Direct Methods for LS

which gives

$$g_{kk}^2 = a_{kk} - \sum_{j=1}^{k-1} g_{kj}^2. \tag{10}$$

Moreover,

$$a_{ik} = \sum_{j=1}^{k} g_{ij} g_{kj}, \qquad i = k+1, \ldots, n,$$

Direct Methods for LS

which gives

$$g_{kk}^2 = a_{kk} - \sum_{j=1}^{k-1} g_{kj}^2. \tag{10}$$

Moreover,

$$a_{ik} = \sum_{j=1}^{k} g_{ij} g_{kj}, \qquad i = k+1, \ldots, n,$$

hence the k-th column of G can be computed by

$$g_{ik} = \left(a_{ik} - \sum_{j=1}^{k-1} g_{ij} g_{kj} \right) \Big/ g_{kk}, \qquad i = k+1, \ldots, n. \tag{11}$$

Direct Methods for LS

Algorithm 6 (Cholesky Factorization) *Given an $n \times n$ symmetric positive definite matrix A, this algorithm computes the Cholesky factorization $A = GG^T$.*

Direct Methods for LS

Algorithm 6 (Cholesky Factorization) *Given an $n \times n$ symmetric positive definite matrix A, this algorithm computes the Cholesky factorization $A = GG^T$.*

```
Initialize $G = 0$
```

Direct Methods for LS

Algorithm 6 (Cholesky Factorization) *Given an* $n \times n$ *symmetric positive definite matrix* A*, this algorithm computes the Cholesky factorization* $A = GG^T$*.*

```
Initialize $G = 0$
For $k = 1, \ldots, n$

End For
```

Direct Methods for LS

Algorithm 6 (Cholesky Factorization) *Given an $n \times n$ symmetric positive definite matrix A, this algorithm computes the Cholesky factorization $A = GG^T$.*

```
Initialize $G = 0$
For $k = 1, \ldots, n$
    $G(k,k) = \sqrt{A(k,k) - \sum_{j=1}^{k-1} G(k,j)G(k,j)}$
End For
```

Direct Methods for LS

Algorithm 6 (Cholesky Factorization) *Given an $n \times n$ symmetric positive definite matrix A, this algorithm computes the Cholesky factorization $A = GG^T$.*

```
Initialize $G = 0$
For $k = 1, \ldots, n$
    $G(k,k) = \sqrt{A(k,k) - \sum_{j=1}^{k-1} G(k,j)G(k,j)}$
    For $i = k+1, \ldots, n$



    End For
End For
```

Direct Methods for LS

Algorithm 6 (Cholesky Factorization) *Given an $n \times n$ symmetric positive definite matrix A, this algorithm computes the Cholesky factorization $A = GG^T$.*

```
Initialize $G = 0$
For $k = 1, \ldots, n$
    $G(k,k) = \sqrt{A(k,k) - \sum_{j=1}^{k-1} G(k,j)G(k,j)}$
    For $i = k+1, \ldots, n$
        $G(i,k) = \left(A(i,k) - \sum_{j=1}^{k-1} G(i,j)G(k,j)\right) \Big/ G(k,k)$
    End For
End For
```

Direct Methods for LS

Algorithm 6 (Cholesky Factorization) *Given an $n \times n$ symmetric positive definite matrix A, this algorithm computes the Cholesky factorization $A = GG^T$.*

```
Initialize $G = 0$
For $k = 1, \ldots, n$
    $G(k,k) = \sqrt{A(k,k) - \sum_{j=1}^{k-1} G(k,j)G(k,j)}$
    For $i = k+1, \ldots, n$
        $G(i,k) = \left(A(i,k) - \sum_{j=1}^{k-1} G(i,j)G(k,j)\right) \Big/ G(k,k)$
    End For
End For
```

In addition to n square root operations, there are approximately

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

floating-point arithmetic required by the algorithm.

4.2 – Diagonally Dominant Systems

Definition 2 *A matrix $A \in \mathbb{R}^{n\times n}$ is said to be strictly diagonally dominant*

4.2 – Diagonally Dominant Systems

Definition 2 *A matrix $A \in \mathbb{R}^{n\times n}$ is said to be strictly diagonally dominant if*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|.$$

4.2 – Diagonally Dominant Systems

Definition 2 *A matrix* $A \in \mathbb{R}^{n\times n}$ *is said to be strictly diagonally dominant if*

$$|a_{ii}| > \sum_{j=1,j\neq i}^{n} |a_{ij}|.$$

Lemma 3 *If* $A \in \mathbb{R}^{n\times n}$ *is strictly diagonally dominant, then* A *is nonsingular.*

4.2 – Diagonally Dominant Systems

Definition 2 *A matrix $A \in \mathbb{R}^{n\times n}$ is said to be strictly diagonally dominant if*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|.$$

Lemma 3 *If $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant, then A is nonsingular.*

Proof: Suppose A is singular.

4.2 – Diagonally Dominant Systems

Definition 2 *A matrix $A \in \mathbb{R}^{n\times n}$ is said to be strictly diagonally dominant if*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|.$$

Lemma 3 *If $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant, then A is nonsingular.*

Proof: Suppose A is singular. Then there exists $x \in \mathbb{R}^n$, $x \neq 0$ such that $Ax = 0$.

4.2 – Diagonally Dominant Systems

Definition 2 *A matrix $A \in \mathbb{R}^{n\times n}$ is said to be strictly diagonally dominant if*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|.$$

Lemma 3 *If $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant, then A is nonsingular.*

Proof: Suppose A is singular. Then there exists $x \in \mathbb{R}^n$, $x \neq 0$ such that $Ax = 0$. Let k be the integer index such that

$$|x_k| = \max_{1\le i\le n} |x_i| \quad \Longrightarrow \quad \frac{|x_i|}{|x_k|} < 1, \quad \forall\ |x_i| \neq |x_k|.$$

4.2 – Diagonally Dominant Systems

Definition 2 *A matrix $A \in \mathbb{R}^{n\times n}$ is said to be strictly diagonally dominant if*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|.$$

Lemma 3 *If $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant, then A is nonsingular.*

Proof: Suppose A is singular. Then there exists $x \in \mathbb{R}^n$, $x \neq 0$ such that $Ax = 0$. Let k be the integer index such that

$$|x_k| = \max_{1\le i\le n} |x_i| \quad \Longrightarrow \quad \frac{|x_i|}{|x_k|} < 1, \quad \forall\ |x_i| \neq |x_k|.$$

Since $Ax = 0$, for the fixed k,

4.2 – Diagonally Dominant Systems

Definition 2 *A matrix $A \in \mathbb{R}^{n\times n}$ is said to be strictly diagonally dominant if*

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|.$$

Lemma 3 *If $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant, then A is nonsingular.*

Proof: Suppose A is singular. Then there exists $x \in \mathbb{R}^n$, $x \neq 0$ such that $Ax = 0$. Let k be the integer index such that

$$|x_k| = \max_{1\leq i\leq n} |x_i| \quad \Longrightarrow \quad \frac{|x_i|}{|x_k|} < 1, \quad \forall \, |x_i| \neq |x_k|.$$

Since $Ax = 0$, for the fixed k, we have

$$\sum_{j=1}^{n} a_{kj}x_j = 0 \ \Rightarrow \ a_{kk}x_k = -\sum_{j=1, j\neq k}^{n} a_{kj}x_j \ \Rightarrow \ |a_{kk}||x_k| \leq \sum_{j=1, j\neq k}^{n} |a_{kj}||x_j|,$$

Direct Methods for LS

which implies

$$|a_{kk}| \leq \sum_{j=1, j\neq k}^{n} |a_{kj}| \frac{|x_j|}{|x_k|} < \sum_{j=1, j\neq k}^{n} |a_{kj}|.$$

Direct Methods for LS

which implies

$$|a_{kk}| \le \sum_{j=1, j\neq k}^{n} |a_{kj}| \frac{|x_j|}{|x_k|} < \sum_{j=1, j\neq k}^{n} |a_{kj}|.$$

But this contradicts the assumption that A is diagonally dominant.

Direct Methods for LS

which implies

$$|a_{kk}| \leq \sum_{j=1, j\neq k}^{n} |a_{kj}| \frac{|x_j|}{|x_k|} < \sum_{j=1, j\neq k}^{n} |a_{kj}|.$$

But this contradicts the assumption that A is diagonally dominant. Therefore A must be nonsingular. ■

Direct Methods for LS

which implies

$$|a_{kk}| \leq \sum_{j=1, j\neq k}^{n} |a_{kj}| \frac{|x_j|}{|x_k|} < \sum_{j=1, j\neq k}^{n} |a_{kj}|.$$

But this contradicts the assumption that A is diagonally dominant. Therefore A must be nonsingular. ■

Theorem 4 *Gaussian elimination without pivoting preserve the diagonal dominance of a matrix.*

Direct Methods for LS

which implies

$$|a_{kk}| \leq \sum_{j=1, j\neq k}^{n} |a_{kj}| \frac{|x_j|}{|x_k|} < \sum_{j=1, j\neq k}^{n} |a_{kj}|.$$

But this contradicts the assumption that A is diagonally dominant. Therefore A must be nonsingular. ∎

Theorem 4 *Gaussian elimination without pivoting preserve the diagonal dominance of a matrix.*

Proof: Let $A \in \mathbb{R}^{n\times n}$ be a diagonally dominant matrix and $A^{(2)} = [a_{ij}^{(2)}]$ is the result of applying one step of Gaussian elimination to $A^{(1)} = A$ without any pivoting strategy.

Direct Methods for LS

which implies

$$|a_{kk}| \leq \sum_{j=1, j\neq k}^{n} |a_{kj}| \frac{|x_j|}{|x_k|} < \sum_{j=1, j\neq k}^{n} |a_{kj}|.$$

But this contradicts the assumption that A is diagonally dominant. Therefore A must be nonsingular. ■

Theorem 4 *Gaussian elimination without pivoting preserve the diagonal dominance of a matrix.*

Proof: Let $A \in \mathbb{R}^{n\times n}$ be a diagonally dominant matrix and $A^{(2)} = [a_{ij}^{(2)}]$ is the result of applying one step of Gaussian elimination to $A^{(1)} = A$ without any pivoting strategy.

After one step of Gaussian elimination, $a_{i1}^{(2)} = 0$ for $i = 2, \ldots, n$, and the first row is unchanged.

Direct Methods for LS

which implies

$$|a_{kk}| \leq \sum_{j=1, j\neq k}^{n} |a_{kj}| \frac{|x_j|}{|x_k|} < \sum_{j=1, j\neq k}^{n} |a_{kj}|.$$

But this contradicts the assumption that A is diagonally dominant. Therefore A must be nonsingular. ■

Theorem 4 *Gaussian elimination without pivoting preserve the diagonal dominance of a matrix.*

Proof: Let $A \in \mathbb{R}^{n\times n}$ be a diagonally dominant matrix and $A^{(2)} = [a_{ij}^{(2)}]$ is the result of applying one step of Gaussian elimination to $A^{(1)} = A$ without any pivoting strategy.

After one step of Gaussian elimination, $a_{i1}^{(2)} = 0$ for $i = 2, \ldots, n$, and the first row is unchanged. Therefore, the property

$$a_{11}^{(2)} > \sum_{j=2}^{n} |a_{1j}^{(2)}|$$

Direct Methods for LS

is preserved,

Direct Methods for LS

is preserved, and all we need to show is that

$$a_{ii}^{(2)} > \sum_{j=2, j \neq i}^{n} |a_{ij}^{(2)}|, \quad \text{for} \quad i = 2, \ldots, n.$$

Direct Methods for LS

is preserved, and all we need to show is that

$$a_{ii}^{(2)} > \sum_{j=2, j\neq i}^{n} |a_{ij}^{(2)}|, \quad \text{for} \quad i = 2, \ldots, n.$$

Using the Gaussian elimination formula (4), we have

Direct Methods for LS

is preserved, and all we need to show is that

$$a_{ii}^{(2)} > \sum_{j=2, j\neq i}^{n} |a_{ij}^{(2)}|, \quad \text{for} \quad i = 2, \ldots, n.$$

Using the Gaussian elimination formula (4), we have

$$|a_{ii}^{(2)}| \quad =$$

Direct Methods for LS

is preserved, and all we need to show is that

$$a_{ii}^{(2)} > \sum_{j=2,j\neq i}^{n} |a_{ij}^{(2)}|, \quad \text{for} \quad i = 2, \ldots, n.$$

Using the Gaussian elimination formula (4), we have

$$|a_{ii}^{(2)}| \quad = \quad \left| a_{ii}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{1i}^{(1)} \right| = \left| a_{ii} - \frac{a_{i1}}{a_{11}} a_{1i} \right|$$

Direct Methods for LS

is preserved, and all we need to show is that

$$a_{ii}^{(2)} > \sum_{j=2, j\neq i}^{n} |a_{ij}^{(2)}|, \quad \text{for} \quad i = 2, \ldots, n.$$

Using the Gaussian elimination formula (4), we have

$$\begin{aligned} |a_{ii}^{(2)}| &= \left| a_{ii}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{1i}^{(1)} \right| = \left| a_{ii} - \frac{a_{i1}}{a_{11}} a_{1i} \right| \\ &\geq |a_{ii}| - \frac{|a_{i1}|}{|a_{11}|} |a_{1i}| \end{aligned}$$

Direct Methods for LS

is preserved, and all we need to show is that

$$a_{ii}^{(2)} > \sum_{j=2, j\neq i}^{n} |a_{ij}^{(2)}|, \quad \text{for} \quad i = 2, \ldots, n.$$

Using the Gaussian elimination formula (4), we have

$$\begin{aligned}
|a_{ii}^{(2)}| &= \left| a_{ii}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{1i}^{(1)} \right| = \left| a_{ii} - \frac{a_{i1}}{a_{11}} a_{1i} \right| \\
&\geq |a_{ii}| - \frac{|a_{i1}|}{|a_{11}|} |a_{1i}| \\
&= |a_{ii}| - |a_{i1}| + |a_{i1}| - \frac{|a_{i1}|}{|a_{11}|} |a_{1i}|
\end{aligned}$$

Direct Methods for LS

is preserved, and all we need to show is that

$$a_{ii}^{(2)} > \sum_{j=2, j\neq i}^{n} |a_{ij}^{(2)}|, \quad \text{for} \quad i = 2, \ldots, n.$$

Using the Gaussian elimination formula (4), we have

$$\begin{aligned} |a_{ii}^{(2)}| &= \left| a_{ii}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{1i}^{(1)} \right| = \left| a_{ii} - \frac{a_{i1}}{a_{11}} a_{1i} \right| \\ &\geq |a_{ii}| - \frac{|a_{i1}|}{|a_{11}|} |a_{1i}| \\ &= |a_{ii}| - |a_{i1}| + |a_{i1}| - \frac{|a_{i1}|}{|a_{11}|} |a_{1i}| \\ &= |a_{ii}| - |a_{i1}| + \frac{|a_{i1}|}{|a_{11}|} \left(|a_{11}| - |a_{1i}| \right) \end{aligned}$$

Direct Methods for LS

is preserved, and all we need to show is that

$$a_{ii}^{(2)} > \sum_{j=2,j\neq i}^{n} |a_{ij}^{(2)}|, \quad \text{for} \quad i = 2, \ldots, n.$$

Using the Gaussian elimination formula (4), we have

$$\begin{aligned}
|a_{ii}^{(2)}| &= \left| a_{ii}^{(1)} - \frac{a_{i1}^{(1)}}{a_{11}^{(1)}} a_{1i}^{(1)} \right| = \left| a_{ii} - \frac{a_{i1}}{a_{11}} a_{1i} \right| \\
&\geq |a_{ii}| - \frac{|a_{i1}|}{|a_{11}|} |a_{1i}| \\
&= |a_{ii}| - |a_{i1}| + |a_{i1}| - \frac{|a_{i1}|}{|a_{11}|} |a_{1i}| \\
&= |a_{ii}| - |a_{i1}| + \frac{|a_{i1}|}{|a_{11}|} \left(|a_{11}| - |a_{1i}| \right) \\
&> \sum_{j=2,j\neq i}^{n} |a_{ij}| + \frac{|a_{i1}|}{|a_{11}|} \sum_{j=2,j\neq i}^{n} |a_{1j}|
\end{aligned}$$

Direct Methods for LS

$$|a_{ii}^{(2)}| \quad > \quad \sum_{j=2,j\neq i}^{n} |a_{ij}| + \sum_{j=2,j\neq i}^{n} \frac{|a_{i1}|}{|a_{11}|} |a_{1j}|$$

Direct Methods for LS

$$\begin{aligned}
|a_{ii}^{(2)}| &> \sum_{j=2,j\neq i}^{n} |a_{ij}| + \sum_{j=2,j\neq i}^{n} \frac{|a_{i1}|}{|a_{11}|}|a_{1j}| \\
&\geq \sum_{j=2,j\neq i}^{n} \left| a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j} \right|
\end{aligned}$$

Direct Methods for LS

$$\begin{aligned} |a_{ii}^{(2)}| &> \sum_{j=2, j\neq i}^{n} |a_{ij}| + \sum_{j=2, j\neq i}^{n} \frac{|a_{i1}|}{|a_{11}|} |a_{1j}| \\ &\geq \sum_{j=2, j\neq i}^{n} \left| a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j} \right| \\ &= \sum_{j=2, j\neq i}^{n} |a_{ij}^{(2)}| \end{aligned}$$

Direct Methods for LS

$$
\begin{aligned}
|a_{ii}^{(2)}| &> \sum_{j=2,j\neq i}^{n} |a_{ij}| + \sum_{j=2,j\neq i}^{n} \frac{|a_{i1}|}{|a_{11}|}|a_{1j}| \\
&\geq \sum_{j=2,j\neq i}^{n} \left| a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j} \right| \\
&= \sum_{j=2,j\neq i}^{n} |a_{ij}^{(2)}|
\end{aligned}
$$

Thus $A^{(2)}$ is still diagonally dominant.

Direct Methods for LS

$$
\begin{aligned}
|a_{ii}^{(2)}| &> \sum_{j=2, j\neq i}^{n} |a_{ij}| + \sum_{j=2, j\neq i}^{n} \frac{|a_{i1}|}{|a_{11}|} |a_{1j}| \\
&\geq \sum_{j=2, j\neq i}^{n} \left| a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j} \right| \\
&= \sum_{j=2, j\neq i}^{n} |a_{ij}^{(2)}|
\end{aligned}
$$

Thus $A^{(2)}$ is still diagonally dominant. Since the subsequent steps of Gaussian elimination mimic the first, except for being applied to submatrices of smaller size, it suffices to conclude that Gaussian elimination without pivoting preserves the diagonal dominance of a matrix. $\blacksquare$

4.3 – Tridiagonal System

A square matrix $A = [a_{ij}]$ is said to be tridiagonal if

$$A = \begin{bmatrix} a_{11} & a_{12} & & \\ a_{21} & a_{22} & \ddots & \\ & \ddots & \ddots & a_{n-1,n} \\ & & a_{n,n-1} & a_{n,n} \end{bmatrix}.$$

4.3 – Tridiagonal System

A square matrix $A = [a_{ij}]$ is said to be tridiagonal if

$$A = \begin{bmatrix} a_{11} & a_{12} & & \\ a_{21} & a_{22} & \ddots & \\ & \ddots & \ddots & a_{n-1,n} \\ & & a_{n,n-1} & a_{n,n} \end{bmatrix}.$$

If Gaussian elimination can be applied safely without pivoting.

Direct Methods for LS

4.3 – Tridiagonal System

A square matrix $A = [a_{ij}]$ is said to be tridiagonal if

$$A = \begin{bmatrix} a_{11} & a_{12} & & \\ a_{21} & a_{22} & \ddots & \\ & \ddots & \ddots & a_{n-1,n} \\ & & a_{n,n-1} & a_{n,n} \end{bmatrix}.$$

If Gaussian elimination can be applied safely without pivoting. Then L and U factors would have the form

$$L = \begin{bmatrix} 1 & & & \\ \ell_{21} & 1 & & \\ & \ddots & \ddots & \\ & & \ell_{n,n-1} & 1 \end{bmatrix} \quad \text{and} \quad U = \begin{bmatrix} u_{11} & u_{12} & & \\ & u_{22} & \ddots & \\ & & \ddots & u_{n-1,n} \\ & & & u_{nn} \end{bmatrix},$$

Direct Methods for LS

and the entries are computed by the simple algorithm which only costs $3n$ flops.

Direct Methods for LS

and the entries are computed by the simple algorithm which only costs $3n$ flops.

Algorithm 7 (Tridiagonal LU Factorization) *This algorithm computes the LU factorization for a tridiagonal matrix without using pivoting strategy.*

Direct Methods for LS

and the entries are computed by the simple algorithm which only costs $3n$ flops.

Algorithm 7 (Tridiagonal LU Factorization) *This algorithm computes the LU factorization for a tridiagonal matrix without using pivoting strategy.*

$$U(1,1) = A(1,1)$$

Direct Methods for LS

and the entries are computed by the simple algorithm which only costs $3n$ flops.

Algorithm 7 (Tridiagonal LU Factorization) *This algorithm computes the LU factorization for a tridiagonal matrix without using pivoting strategy.*

```
$U(1,1) = A(1,1)$
For $i = 2, \ldots, n$



End For
```

Direct Methods for LS

and the entries are computed by the simple algorithm which only costs $3n$ flops.

Algorithm 7 (Tridiagonal LU Factorization) *This algorithm computes the LU factorization for a tridiagonal matrix without using pivoting strategy.*

```
$U(1,1) = A(1,1)$
For $i = 2, \ldots, n$
    $U(i-1,i) = A(i-1,i)$
End For
```

Direct Methods for LS

and the entries are computed by the simple algorithm which only costs $3n$ flops.

Algorithm 7 (Tridiagonal LU Factorization) *This algorithm computes the LU factorization for a tridiagonal matrix without using pivoting strategy.*

```
$U(1,1) = A(1,1)$
For $i = 2, \ldots, n$
    $U(i-1,i) = A(i-1,i)$
    $L(i,i-1) = A(i,i-1)/U(i-1,i-1)$

End For
```

Direct Methods for LS

and the entries are computed by the simple algorithm which only costs $3n$ flops.

Algorithm 7 (Tridiagonal LU Factorization) *This algorithm computes the LU factorization for a tridiagonal matrix without using pivoting strategy.*

```
$U(1,1) = A(1,1)$
For $i = 2, \ldots, n$
    $U(i-1,i) = A(i-1,i)$
    $L(i,i-1) = A(i,i-1)/U(i-1,i-1)$
    $U(i,i) = A(i,i) - L(i,i-1)U(i-1,i)$
End For
```

Direct Methods for LS

and the entries are computed by the simple algorithm which only costs $3n$ flops.

Algorithm 7 (Tridiagonal LU Factorization) *This algorithm computes the LU factorization for a tridiagonal matrix without using pivoting strategy.*

```
$U(1,1) = A(1,1)$
For $i = 2, \ldots, n$
    $U(i-1,i) = A(i-1,i)$
    $L(i,i-1) = A(i,i-1)/U(i-1,i-1)$
    $U(i,i) = A(i,i) - L(i,i-1)U(i-1,i)$
End For
```

A tridiagonal linear system arises in many applications, such as finite difference discretization to second order linear boundary-value problem and the cubic spline approximations.

4.4 – General Banded Systems

In many applications that involve linear systems, the coefficient matrix is banded. Formally, we say that $A = [a_{ij}]$ has upper bandwidth q if $a_{ij} = 0$ whenever $j > i + q$ and lower bandwidth p if $a_{ij} = 0$ whenever $i > j + p$. Substantial economies can be realized when solving banded systems because the triangular factors in the LU factorization are also banded.

Direct Methods for LS

5 – Perturbation Analysis

In this section, we develop some perturbation theory for the problem of solving linear systems $Ax = b$.

5 – Perturbation Analysis

In this section, we develop some perturbation theory for the problem of solving linear systems $Ax = b$. If we solve such a system numerically, we obtain not the exact solution x but an approximate computed solution $\widehat{x}$.

5 – Perturbation Analysis

In this section, we develop some perturbation theory for the problem of solving linear systems $Ax = b$. If we solve such a system numerically, we obtain not the exact solution x but an approximate computed solution $\widehat{x}$. The difference

$$e = x - \widehat{x}$$

is called the error vector which is, however, not known.

5 – Perturbation Analysis

In this section, we develop some perturbation theory for the problem of solving linear systems $Ax = b$. If we solve such a system numerically, we obtain not the exact solution x but an approximate computed solution $\widehat{x}$. The difference

$$e = x - \widehat{x}$$

is called the error vector which is, however, not known. Instead one can test the accuracy of $\widehat{x}$ by forming $A\widehat{x}$ to see whether it is close to b.

5 – Perturbation Analysis

In this section, we develop some perturbation theory for the problem of solving linear systems $Ax = b$. If we solve such a system numerically, we obtain not the exact solution x but an approximate computed solution $\widehat{x}$. The difference

$$e = x - \widehat{x}$$

is called the error vector which is, however, not known. Instead one can test the accuracy of $\widehat{x}$ by forming $A\widehat{x}$ to see whether it is close to b. Thus we have the definition for the residual vector.

5 – Perturbation Analysis

In this section, we develop some perturbation theory for the problem of solving linear systems $Ax = b$. If we solve such a system numerically, we obtain not the exact solution x but an approximate computed solution $\widehat{x}$. The difference

$$e = x - \widehat{x}$$

is called the error vector which is, however, not known. Instead one can test the accuracy of $\widehat{x}$ by forming $A\widehat{x}$ to see whether it is close to b. Thus we have the definition for the residual vector.

Definition 3 *Let $\widehat{x}$ be the computed solution to the linear system of equations $Ax = b$. Then the vector*

$$r = b - A\widehat{x}$$

is called the residual vector.

Direct Methods for LS

Then we can derive the residual equation

$$Ae = Ax - A\widehat{x} = b - A\widehat{x} = r \tag{12}$$

between the error vector and the residual vector.

Direct Methods for LS

Then we can derive the residual equation

$$Ae = Ax - A\widehat{x} = b - A\widehat{x} = r \tag{12}$$

between the error vector and the residual vector.

Notice that $\widehat{x}$ is the exact solution of the linear system

$$A\widehat{x} = \widehat{b},$$

which has a perturbed right-hand side

$$\widehat{b} = b - r.$$

Then

Direct Methods for LS

Then we can derive the residual equation

$$Ae = Ax - A\widehat{x} = b - A\widehat{x} = r \tag{12}$$

between the error vector and the residual vector.

Notice that $\widehat{x}$ is the exact solution of the linear system

$$A\widehat{x} = \widehat{b},$$

which has a perturbed right-hand side

$$\widehat{b} = b - r.$$

Then

$$\|x - \widehat{x}\|$$

Direct Methods for LS

Then we can derive the residual equation

$$Ae = Ax - A\widehat{x} = b - A\widehat{x} = r \tag{12}$$

between the error vector and the residual vector.

Notice that $\widehat{x}$ is the exact solution of the linear system

$$A\widehat{x} = \widehat{b},$$

which has a perturbed right-hand side

$$\widehat{b} = b - r.$$

Then

$$\|x - \widehat{x}\| \quad = \quad \|A^{-1}b - A^{-1}\widehat{b}\| = \|A^{-1}(b - \widehat{b})\|$$

Direct Methods for LS

Then we can derive the residual equation

$$Ae = Ax - A\widehat{x} = b - A\widehat{x} = r \tag{12}$$

between the error vector and the residual vector.

Notice that $\widehat{x}$ is the exact solution of the linear system

$$A\widehat{x} = \widehat{b},$$

which has a perturbed right-hand side

$$\widehat{b} = b - r.$$

Then

$$\begin{aligned} \|x - \widehat{x}\| &= \|A^{-1}b - A^{-1}\widehat{b}\| = \|A^{-1}(b - \widehat{b})\| \\ &\leq \|A^{-1}\|\|b - \widehat{b}\| \end{aligned}$$

Direct Methods for LS

Then we can derive the residual equation

$$Ae = Ax - A\widehat{x} = b - A\widehat{x} = r \tag{12}$$

between the error vector and the residual vector.

Notice that $\widehat{x}$ is the exact solution of the linear system

$$A\widehat{x} = \widehat{b},$$

which has a perturbed right-hand side

$$\widehat{b} = b - r.$$

Then

$$\begin{aligned} \|x - \widehat{x}\| &= \|A^{-1}b - A^{-1}\widehat{b}\| = \|A^{-1}(b - \widehat{b})\| \\ &\le \|A^{-1}\|\|b - \widehat{b}\| = \|A^{-1}\|\|b\|\frac{\|b - \widehat{b}\|}{\|b\|} \end{aligned}$$

Direct Methods for LS

Then we can derive the residual equation

$$Ae = Ax - A\widehat{x} = b - A\widehat{x} = r \tag{12}$$

between the error vector and the residual vector.

Notice that $\widehat{x}$ is the exact solution of the linear system

$$A\widehat{x} = \widehat{b},$$

which has a perturbed right-hand side

$$\widehat{b} = b - r.$$

Then

$$\begin{aligned} \|x - \widehat{x}\| &= \|A^{-1}b - A^{-1}\widehat{b}\| = \|A^{-1}(b - \widehat{b})\| \\ &\leq \|A^{-1}\|\|b - \widehat{b}\| = \|A^{-1}\|\|b\|\frac{\|b - \widehat{b}\|}{\|b\|} = \|A^{-1}\|\|Ax\|\frac{\|b - \widehat{b}\|}{\|b\|} \end{aligned}$$

Direct Methods for LS

Then we can derive the residual equation

$$Ae = Ax - A\widehat{x} = b - A\widehat{x} = r \tag{12}$$

between the error vector and the residual vector.

Notice that $\widehat{x}$ is the exact solution of the linear system

$$A\widehat{x} = \widehat{b},$$

which has a perturbed right-hand side

$$\widehat{b} = b - r.$$

Then

$$\begin{aligned}
\|x - \widehat{x}\| &= \|A^{-1}b - A^{-1}\widehat{b}\| = \|A^{-1}(b - \widehat{b})\| \\
&\leq \|A^{-1}\|\|b - \widehat{b}\| = \|A^{-1}\|\|b\|\frac{\|b - \widehat{b}\|}{\|b\|} = \|A^{-1}\|\|Ax\|\frac{\|b - \widehat{b}\|}{\|b\|} \\
&\leq \|A^{-1}\|\|A\|\|x\|\frac{\|b - \widehat{b}\|}{\|b\|}
\end{aligned}$$

Direct Methods for LS

Therefore

$$\frac{\|x-\widehat{x}\|}{\|x\|} \le \kappa(A)\frac{\|b-\widehat{b}\|}{\|b\|} = \kappa(A)\frac{\|r\|}{\|b\|},$$

Direct Methods for LS

Therefore

$$\frac{\|x-\widehat{x}\|}{\|x\|} \le \kappa(A)\frac{\|b-\widehat{b}\|}{\|b\|} = \kappa(A)\frac{\|r\|}{\|b\|},$$

where

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is called the condition number of A.

Direct Methods for LS

Therefore

$$\frac{\|x-\widehat{x}\|}{\|x\|} \le \kappa(A)\frac{\|b-\widehat{b}\|}{\|b\|} = \kappa(A)\frac{\|r\|}{\|b\|},$$

where

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is called the condition number of A.

On the other hand, by the residual vector, we have

Direct Methods for LS

Therefore

$$\frac{\|x-\widehat{x}\|}{\|x\|} \leq \kappa(A)\frac{\|b-\widehat{b}\|}{\|b\|} = \kappa(A)\frac{\|r\|}{\|b\|},$$

where

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is called the condition number of A.

On the other hand, by the residual vector, we have

$$\|r\|\|x\| = \|Ae\|\|A^{-1}b\| \qquad .$$

Direct Methods for LS

Therefore

$$\frac{\|x-\widehat{x}\|}{\|x\|} \leq \kappa(A)\frac{\|b-\widehat{b}\|}{\|b\|} = \kappa(A)\frac{\|r\|}{\|b\|},$$

where

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is called the condition number of A.

On the other hand, by the residual vector, we have

$$\|r\|\|x\| = \|Ae\|\|A^{-1}b\| \leq \|A\|\|A^{-1}\|\|e\|\|b\| \qquad .$$

Direct Methods for LS

Therefore

$$\frac{\|x-\widehat{x}\|}{\|x\|} \le \kappa(A)\frac{\|b-\widehat{b}\|}{\|b\|} = \kappa(A)\frac{\|r\|}{\|b\|},$$

where

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is called the condition number of A.

On the other hand, by the residual vector, we have

$$\|r\|\|x\| = \|Ae\|\|A^{-1}b\| \le \|A\|\|A^{-1}\|\|e\|\|b\| = \kappa(A)\|x-\widehat{x}\|\|b\|.$$

Direct Methods for LS

Therefore

$$\frac{\|x - \widehat{x}\|}{\|x\|} \leq \kappa(A)\frac{\|b - \widehat{b}\|}{\|b\|} = \kappa(A)\frac{\|r\|}{\|b\|},$$

where

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is called the condition number of A.

On the other hand, by the residual vector, we have

$$\|r\|\|x\| = \|Ae\|\|A^{-1}b\| \leq \|A\|\|A^{-1}\|\|e\|\|b\| = \kappa(A)\|x - \widehat{x}\|\|b\|.$$

Hence

$$\frac{1}{\kappa(A)}\frac{\|r\|}{\|b\|} \leq \frac{\|x - \widehat{x}\|}{\|x\|}. \qquad (13)$$

Direct Methods for LS

Therefore

$$\frac{\|x-\widehat{x}\|}{\|x\|} \le \kappa(A)\frac{\|b-\widehat{b}\|}{\|b\|} = \kappa(A)\frac{\|r\|}{\|b\|},$$

where

$$\kappa(A) = \|A\|\|A^{-1}\|$$

is called the condition number of A.

On the other hand, by the residual vector, we have

$$\|r\|\|x\| = \|Ae\|\|A^{-1}b\| \le \|A\|\|A^{-1}\|\|e\|\|b\| = \kappa(A)\|x-\widehat{x}\|\|b\|.$$

Hence

$$\frac{1}{\kappa(A)}\frac{\|r\|}{\|b\|} \le \frac{\|x-\widehat{x}\|}{\|x\|}. \tag{13}$$

Theorem 5

$$\frac{1}{\kappa(A)}\frac{\|r\|}{\|b\|} \le \frac{\|x-\widehat{x}\|}{\|x\|} \le \kappa(A)\frac{\|r\|}{\|b\|}.$$

Direct Methods for LS

Lemma 4 *Suppose that x and $\widetilde{x}$ satisfy*

$$Ax = b \qquad \text{and} \qquad (A + \triangle A)\widetilde{x} = b + \triangle b,$$

where $A \in \mathbb{R}^{n\times n}$, $\triangle A \in \mathbb{R}^{n\times n}$, $0 \neq b \in \mathbb{R}^{n}$, and $\triangle b \in \mathbb{R}^{n}$, with

$$\frac{\|\triangle A\|}{\|A\|} \leq \delta \qquad \text{and} \qquad \frac{\|\triangle b\|}{\|b\|} \leq \delta.$$

Direct Methods for LS

Lemma 4 *Suppose that x and $\widetilde{x}$ satisfy*

$$Ax = b \qquad \text{and} \qquad (A + \triangle A)\widetilde{x} = b + \triangle b,$$

where $A \in \mathbb{R}^{n\times n}$, $\triangle A \in \mathbb{R}^{n\times n}$, $0 \neq b \in \mathbb{R}^n$, and $\triangle b \in \mathbb{R}^n$, with

$$\frac{\|\triangle A\|}{\|A\|} \leq \delta \qquad \text{and} \qquad \frac{\|\triangle b\|}{\|b\|} \leq \delta.$$

If $\kappa(A) \cdot \delta < 1$,

Direct Methods for LS

Lemma 4 *Suppose that* x *and* $\widetilde{x}$ *satisfy*

$$Ax = b \qquad \text{and} \qquad (A + \triangle A)\widetilde{x} = b + \triangle b,$$

where $A \in \mathbb{R}^{n\times n}$, $\triangle A \in \mathbb{R}^{n\times n}$, $0 \neq b \in \mathbb{R}^n$, *and* $\triangle b \in \mathbb{R}^n$, *with*

$$\frac{\|\triangle A\|}{\|A\|} \le \delta \qquad \text{and} \qquad \frac{\|\triangle b\|}{\|b\|} \le \delta.$$

If $\kappa(A) \cdot \delta < 1$, *then* $A + \triangle A$ *is nonsingular*

Direct Methods for LS

Lemma 4 *Suppose that x and $\widetilde{x}$ satisfy*

$$Ax = b \qquad \text{and} \qquad (A + \triangle A)\widetilde{x} = b + \triangle b,$$

where $A \in \mathbb{R}^{n\times n}$, $\triangle A \in \mathbb{R}^{n\times n}$, $0 \neq b \in \mathbb{R}^n$, and $\triangle b \in \mathbb{R}^n$, with

$$\frac{\|\triangle A\|}{\|A\|} \leq \delta \qquad \text{and} \qquad \frac{\|\triangle b\|}{\|b\|} \leq \delta.$$

If $\kappa(A) \cdot \delta < 1$, then $A + \triangle A$ is nonsingular and

$$\frac{\|\widetilde{x}\|}{\|x\|} \leq \frac{1 + \kappa(A)\delta}{1 - \kappa(A)\delta}.$$

Direct Methods for LS

Lemma 4 *Suppose that x and $\widetilde{x}$ satisfy*

$$Ax = b \qquad \text{and} \qquad (A + \triangle A)\widetilde{x} = b + \triangle b,$$

where $A \in \mathbb{R}^{n\times n}$, $\triangle A \in \mathbb{R}^{n\times n}$, $0 \neq b \in \mathbb{R}^n$, and $\triangle b \in \mathbb{R}^n$, with

$$\frac{\|\triangle A\|}{\|A\|} \leq \delta \qquad \text{and} \qquad \frac{\|\triangle b\|}{\|b\|} \leq \delta.$$

If $\kappa(A)\cdot\delta < 1$, then $A + \triangle A$ is nonsingular and

$$\frac{\|\widetilde{x}\|}{\|x\|} \leq \frac{1+\kappa(A)\delta}{1-\kappa(A)\delta}.$$

Proof: Since $\|A^{-1}\triangle A\| \leq \|A^{-1}\|\|\triangle A\| \leq \delta\|A^{-1}\|\|A\| = \delta\kappa(A) < 1,$

Direct Methods for LS

Lemma 4 *Suppose that* x *and* $\widetilde{x}$ *satisfy*

$$Ax = b \qquad \text{and} \qquad (A + \triangle A)\widetilde{x} = b + \triangle b,$$

where $A \in \mathbb{R}^{n\times n}$, $\triangle A \in \mathbb{R}^{n\times n}$, $0 \neq b \in \mathbb{R}^n$, *and* $\triangle b \in \mathbb{R}^n$, *with*

$$\frac{\|\triangle A\|}{\|A\|} \leq \delta \qquad \text{and} \qquad \frac{\|\triangle b\|}{\|b\|} \leq \delta.$$

If $\kappa(A)\cdot\delta < 1$, *then* $A + \triangle A$ *is nonsingular and*

$$\frac{\|\widetilde{x}\|}{\|x\|} \leq \frac{1+\kappa(A)\delta}{1-\kappa(A)\delta}.$$

Proof: Since $\|A^{-1}\triangle A\| \leq \|A^{-1}\|\|\triangle A\| \leq \delta\|A^{-1}\|\|A\| = \delta\kappa(A) < 1$, it follows from Theorem **??** that $A + \triangle A$ is nonsingular.

Direct Methods for LS

Lemma 4 *Suppose that* x *and* $\widetilde{x}$ *satisfy*

$$Ax = b \qquad \text{and} \qquad (A + \triangle A)\widetilde{x} = b + \triangle b,$$

where $A \in \mathbb{R}^{n\times n}$*,* $\triangle A \in \mathbb{R}^{n\times n}$*,* $0 \neq b \in \mathbb{R}^n$*, and* $\triangle b \in \mathbb{R}^n$*, with*

$$\frac{\|\triangle A\|}{\|A\|} \le \delta \qquad \text{and} \qquad \frac{\|\triangle b\|}{\|b\|} \le \delta.$$

If $\kappa(A)\cdot\delta < 1$*, then* $A + \triangle A$ *is nonsingular and*

$$\frac{\|\widetilde{x}\|}{\|x\|} \le \frac{1 + \kappa(A)\delta}{1 - \kappa(A)\delta}.$$

Proof: Since $\|A^{-1}\triangle A\| \le \|A^{-1}\|\|\triangle A\| \le \delta\|A^{-1}\|\|A\| = \delta\kappa(A) < 1$, it follows from Theorem **??** that $A + \triangle A$ is nonsingular. Now $(A + \triangle A)\widetilde{x} = b + \triangle b$,

Direct Methods for LS

Lemma 4 *Suppose that x and $\widetilde{x}$ satisfy*

$$Ax = b \qquad \text{and} \qquad (A + \triangle A)\widetilde{x} = b + \triangle b,$$

where $A \in \mathbb{R}^{n\times n}$, $\triangle A \in \mathbb{R}^{n\times n}$, $0 \neq b \in \mathbb{R}^n$, and $\triangle b \in \mathbb{R}^n$, with

$$\frac{\|\triangle A\|}{\|A\|} \leq \delta \qquad \text{and} \qquad \frac{\|\triangle b\|}{\|b\|} \leq \delta.$$

If $\kappa(A) \cdot \delta < 1$, then $A + \triangle A$ is nonsingular and

$$\frac{\|\widetilde{x}\|}{\|x\|} \leq \frac{1 + \kappa(A)\delta}{1 - \kappa(A)\delta}.$$

Proof: Since $\|A^{-1}\triangle A\| \leq \|A^{-1}\|\|\triangle A\| \leq \delta\|A^{-1}\|\|A\| = \delta\kappa(A) < 1$, it follows from Theorem **??** that $A + \triangle A$ is nonsingular. Now $(A + \triangle A)\widetilde{x} = b + \triangle b$,

$$(I + A^{-1}\triangle A)\widetilde{x} = A^{-1}b + A^{-1}\triangle b$$

Direct Methods for LS

Lemma 4 *Suppose that x and $\widetilde{x}$ satisfy*

$$Ax = b \qquad \textit{and} \qquad (A + \triangle A)\widetilde{x} = b + \triangle b,$$

where $A \in \mathbb{R}^{n\times n}$, $\triangle A \in \mathbb{R}^{n\times n}$, $0 \neq b \in \mathbb{R}^n$, and $\triangle b \in \mathbb{R}^n$, with

$$\frac{\|\triangle A\|}{\|A\|} \leq \delta \qquad \textit{and} \qquad \frac{\|\triangle b\|}{\|b\|} \leq \delta.$$

If $\kappa(A)\cdot\delta < 1$, then $A + \triangle A$ is nonsingular and

$$\frac{\|\widetilde{x}\|}{\|x\|} \leq \frac{1+\kappa(A)\delta}{1-\kappa(A)\delta}.$$

Proof: Since $\|A^{-1}\triangle A\| \leq \|A^{-1}\|\|\triangle A\| \leq \delta\|A^{-1}\|\|A\| = \delta\kappa(A) < 1$, it follows from Theorem **??** that $A + \triangle A$ is nonsingular. Now $(A+\triangle A)\widetilde{x} = b + \triangle b$,

$$(I + A^{-1}\triangle A)\widetilde{x} = A^{-1}b + A^{-1}\triangle b = x + A^{-1}\triangle b,$$

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

$$\|\widetilde{x}\| \quad \leq$$

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

$$\|\widetilde{x}\| \quad \leq \quad \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \|A^{-1}\|\|\triangle b\|\right)$$

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

$$\begin{aligned}\|\widetilde{x}\| &\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \|A^{-1}\|\|\triangle b\|\right)\\ &\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \delta\|A^{-1}\|\|b\|\right)\end{aligned}$$

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

$$\begin{aligned}
\|\widetilde{x}\| &\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \|A^{-1}\|\|\triangle b\|\right) \\
&\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \|A^{-1}\triangle A\|} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right)
\end{aligned}$$

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

$$\begin{aligned}
\|\widetilde{x}\| &\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \|A^{-1}\|\|\triangle b\|\right) \\
&\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \|A^{-1}\triangle A\|} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right)
\end{aligned}$$

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

$$
\begin{aligned}
\|\widetilde{x}\| &\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \|A^{-1}\|\|\triangle b\|\right) \\
&\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \|A^{-1}\triangle A\|} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&= \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|Ax\|\right)
\end{aligned}
$$

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

$$
\begin{aligned}
\|\widetilde{x}\| &\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \|A^{-1}\|\|\triangle b\|\right) \\
&\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \|A^{-1}\triangle A\|} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&= \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|Ax\|\right) \\
&\leq \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|A\|\|x\|\right)
\end{aligned}
$$

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

$$
\begin{aligned}
\|\widetilde{x}\| &\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \|A^{-1}\|\|\triangle b\|\right) \\
&\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \|A^{-1}\triangle A\|} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&= \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|Ax\|\right) \\
&\leq \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|A\|\|x\|\right) \\
&= \frac{1}{1 - \delta\kappa(A)} (\|x\| + \delta\kappa(A)\|x\|)
\end{aligned}
$$

Direct Methods for LS

and so by taking norms and using Theorem **??** we find

$$
\begin{aligned}
\|\widetilde{x}\| &\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \|A^{-1}\|\|\triangle b\|\right) \\
&\leq \|(I + A^{-1}\triangle A)^{-1}\| \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \|A^{-1}\triangle A\|} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&\leq \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|b\|\right) \\
&= \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|Ax\|\right) \\
&\leq \frac{1}{1 - \delta\kappa(A)} \left(\|x\| + \delta\|A^{-1}\|\|A\|\|x\|\right) \\
&= \frac{1}{1 - \delta\kappa(A)} (\|x\| + \delta\kappa(A)\|x\|) \\
&= \frac{1}{1 - \delta\kappa(A)} (1 + \delta\kappa(A)) \|x\|.
\end{aligned}
$$

Direct Methods for LS

Therefore

$$\frac{\|\widetilde{x}\|}{\|x\|} \le \frac{1+\delta\kappa(A)}{1-\delta\kappa(A)}.$$

■

Direct Methods for LS

Therefore

$$\frac{\|\widetilde{x}\|}{\|x\|} \leq \frac{1+\delta\kappa(A)}{1-\delta\kappa(A)}.$$

■

Theorem 6 *If the conditions of Lemma 4 hold then*

$$\frac{\|x-\widetilde{x}\|}{\|x\|} \leq \frac{2\delta}{1-\kappa(A)\delta}\kappa(A)$$

Direct Methods for LS

Therefore

$$\frac{\|\widetilde{x}\|}{\|x\|} \le \frac{1+\delta\kappa(A)}{1-\delta\kappa(A)}.$$

■

Theorem 6 *If the conditions of Lemma 4 hold then*

$$\frac{\|x-\widetilde{x}\|}{\|x\|} \le \frac{2\delta}{1-\kappa(A)\delta}\kappa(A)$$

Proof: Since $\widetilde{x}$ satisfies $(A+\triangle A)\widetilde{x} = b+\triangle b$, $A\widetilde{x} = b+\triangle b - \triangle A\widetilde{x}$. Then we have

Direct Methods for LS

Therefore

$$\frac{\|\widetilde{x}\|}{\|x\|} \leq \frac{1+\delta\kappa(A)}{1-\delta\kappa(A)}.$$

■

Theorem 6 *If the conditions of Lemma 4 hold then*

$$\frac{\|x-\widetilde{x}\|}{\|x\|} \leq \frac{2\delta}{1-\kappa(A)\delta}\kappa(A)$$

Proof: Since $\widetilde{x}$ satisfies $(A+\triangle A)\widetilde{x} = b+\triangle b$, $A\widetilde{x} = b+\triangle b-\triangle A\widetilde{x}$. Then we have

$$A\widetilde{x} - Ax = \triangle b + \triangle A\widetilde{x}$$

Direct Methods for LS

Therefore

$$\frac{\|\widetilde{x}\|}{\|x\|} \leq \frac{1+\delta\kappa(A)}{1-\delta\kappa(A)}.$$

$\blacksquare$

Theorem 6 *If the conditions of Lemma 4 hold then*

$$\frac{\|x-\widetilde{x}\|}{\|x\|} \leq \frac{2\delta}{1-\kappa(A)\delta}\kappa(A)$$

Proof: Since $\widetilde{x}$ satisfies $(A+\triangle A)\widetilde{x} = b+\triangle b$, $A\widetilde{x} = b+\triangle b-\triangle A\widetilde{x}$. Then we have

$$A\widetilde{x} - Ax = \triangle b + \triangle A\widetilde{x}$$

and

$$\widetilde{x} - x = A^{-1}\left(\triangle b + \triangle A\widetilde{x}\right).$$

Direct Methods for LS

Hence

$$\|\widetilde{x} - x\| \quad \leq \quad \|A^{-1}\| \left(\|\triangle b\| + \|\triangle A\| \|\widetilde{x}\|\right)$$

Direct Methods for LS

Hence

$$\begin{aligned} \|\widetilde{x} - x\| &\leq \|A^{-1}\| \left(\|\triangle b\| + \|\triangle A\| \|\widetilde{x}\|\right) \\ &\leq \|A^{-1}\| \left(\delta \|b\| + \delta \|A\| \|\widetilde{x}\|\right) \end{aligned}$$

Direct Methods for LS

Hence

$$\begin{aligned}\|\widetilde{x}-x\| &\leq \|A^{-1}\|\left(\|\triangle b\| + \|\triangle A\|\|\widetilde{x}\|\right)\\ &\leq \|A^{-1}\|\left(\delta\|b\| + \delta\|A\|\|\widetilde{x}\|\right)\\ &= \delta\|A^{-1}\|\left(\|Ax\| + \|A\|\|\widetilde{x}\|\right)\end{aligned}$$

Direct Methods for LS

Hence

$$\begin{aligned}
\|\widetilde{x} - x\| &\leq \|A^{-1}\| \left(\|\triangle b\| + \|\triangle A\| \|\widetilde{x}\|\right) \\
&\leq \|A^{-1}\| \left(\delta \|b\| + \delta \|A\| \|\widetilde{x}\|\right) \\
&= \delta \|A^{-1}\| \left(\|Ax\| + \|A\| \|\widetilde{x}\|\right) \\
&\leq \delta \|A\| \|A^{-1}\| \left(\|x\| + \|\widetilde{x}\|\right),
\end{aligned}$$

Direct Methods for LS

Hence

$$\begin{aligned}
\|\widetilde{x} - x\| &\leq \|A^{-1}\| \left(\|\triangle b\| + \|\triangle A\| \|\widetilde{x}\|\right) \\
&\leq \|A^{-1}\| \left(\delta \|b\| + \delta \|A\| \|\widetilde{x}\|\right) \\
&= \delta \|A^{-1}\| \left(\|Ax\| + \|A\| \|\widetilde{x}\|\right) \\
&\leq \delta \|A\| \|A^{-1}\| \left(\|x\| + \|\widetilde{x}\|\right),
\end{aligned}$$

which gives

Direct Methods for LS

Hence

$$\begin{aligned}
\|\widetilde{x}-x\| &\le \|A^{-1}\|\left(\|\triangle b\|+\|\triangle A\|\|\widetilde{x}\|\right)\\
&\le \|A^{-1}\|\left(\delta\|b\|+\delta\|A\|\|\widetilde{x}\|\right)\\
&= \delta\|A^{-1}\|\left(\|Ax\|+\|A\|\|\widetilde{x}\|\right)\\
&\le \delta\|A\|\|A^{-1}\|\left(\|x\|+\|\widetilde{x}\|\right),
\end{aligned}$$

which gives

$$\frac{\|\widetilde{x}-x\|}{\|x\|}\le \delta\kappa(A)\left(1+\frac{\|\widetilde{x}\|}{\|x\|}\right)$$

Direct Methods for LS

Hence

$$\begin{aligned}
\|\widetilde{x}-x\| &\le \|A^{-1}\|\left(\|\triangle b\| + \|\triangle A\|\|\widetilde{x}\|\right)\\
&\le \|A^{-1}\|\left(\delta\|b\| + \delta\|A\|\|\widetilde{x}\|\right)\\
&= \delta\|A^{-1}\|\left(\|Ax\| + \|A\|\|\widetilde{x}\|\right)\\
&\le \delta\|A\|\|A^{-1}\|\left(\|x\| + \|\widetilde{x}\|\right),
\end{aligned}$$

which gives

$$\frac{\|\widetilde{x}-x\|}{\|x\|} \le \delta\kappa(A)\left(1+\frac{\|\widetilde{x}\|}{\|x\|}\right) \le \delta\kappa(A)\left(1+\frac{1+\kappa(A)\delta}{1-\kappa(A)\delta}\right)$$

Direct Methods for LS

Hence

$$\begin{aligned}
\|\widetilde{x}-x\| &\le \|A^{-1}\|\left(\|\triangle b\|+\|\triangle A\|\|\widetilde{x}\|\right)\\
&\le \|A^{-1}\|\left(\delta\|b\|+\delta\|A\|\|\widetilde{x}\|\right)\\
&= \delta\|A^{-1}\|\left(\|Ax\|+\|A\|\|\widetilde{x}\|\right)\\
&\le \delta\|A\|\|A^{-1}\|\left(\|x\|+\|\widetilde{x}\|\right),
\end{aligned}$$

which gives

$$\frac{\|\widetilde{x}-x\|}{\|x\|} \le \delta\kappa(A)\left(1+\frac{\|\widetilde{x}\|}{\|x\|}\right) \le \delta\kappa(A)\left(1+\frac{1+\kappa(A)\delta}{1-\kappa(A)\delta}\right) = \frac{2\delta\kappa(A)}{1-\delta\kappa(A)}.$$

■