Lecture 5
Matrices, Gaussian Elimination, LU Decomposition

Reminder of Matrices

Remark 1 We need to solve a system of linear equations of the form:

$$\begin{array}{rcl} a_{1,1}x_1 + a_{1,2}x_2 + ... + a_{1,n}x_n & = & b_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + ... + a_{2,n}x_n & = & b_2 \\ \vdots & & \vdots \\ a_{n,1}x_1 + an, 2x_2 + ... + a_{n,n}x_n & = & b_n \end{array}$$

or in matrix form:

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

where $A \in \mathbb{R}^{n\times n}$ is a matrix, $\mathbf{x} \in \mathbb{R}^n$ are unknowns, and $\mathbf{b} \in \mathbb{R}^n$.

Example 1

$$\begin{array}{rcl} x_1 + 2x_2 - 4x_3 + x_4 & = & 1 \\ 3x_1 - x_2 + x_3 + 4x_4 & = & 3 \\ x_1 - 2x_2 - 4x_3 + x_4 & = & -1 \\ 2x_1 - x_2 - x_3 + 3x_4 & = & 2 \end{array}$$

or in matrix form:

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

where

$$A = \begin{bmatrix} 1 & 2 & -4 & 1 \\ 3 & -1 & 1 & 4 \\ 1 & -2 & 3 & -1 \\ 2 & -1 & -1 & 3 \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 1 \\ 3 \\ -1 \\ 2 \end{bmatrix}$$

How do we solve such a system of linear equations?
Central Questions:

- What is Gaussian elimination? LU decomposition?
- How is LU decomposition related to Gaussian elimination?
- How is an LU decomposition $A = LU$ computed?
- For any square $A \in \mathbb{R}^{n \times n}$, does a decomposition $A = LU$ exist?

Definition 1 – Matrices
A Matrix $A \in \mathbb{R}^{m \times n}$ is a rectangular array of numbers:

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \dots & a_{1,n-1} & a_{1,n} \\ a_{2,1} & a_{2,2} & \dots & a_{2,n-1} & a_{2,n} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m-1,1} & a_{m-1,2} & \dots & a_{m-1,n-1} & a_{m-1,n} \\ a_{m,1} & a_{m,2} & \dots & a_{m,n-1} & a_{m,n} \end{bmatrix}$$

Numbers $a_{i,j}$ are elements of A, or entries of A.
First index (i) of element $a_{i,j}$ is the row index.
Second index (j) of element $a_{i,j}$ is the column index.

Definition 2 – Vectors An n-vector is a “skinny” matrix (dimension $n \times 1$ or $1 \times n$).

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{bmatrix} \text{ or } \mathbf{x}^T = \begin{bmatrix} x_1 & x_2 & \dots & x_{n-1} & x_n \end{bmatrix}$$

Elements x_i are components of $\mathbf{x}$.
Convention: vectors generally column vectors assume $\mathbf{x} \in \mathbb{R}^n$ means $\mathbf{x} \in \mathbb{R}^{n \times 1}$.
To `scipy`, scalars are vectors of length 1 and also matrices of dimension 1×1.

Special Matrices

Definition 3 – Zero Matrix

$$\forall A \in \mathbb{R}^{m \times n} \quad A + 0 = 0 + A = A, \text{ where } 0 = \begin{bmatrix} 0 & 0 & ... & 0 \\ 0 & 0 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 0 \end{bmatrix}$$

Definition 4 – Identity Matrix

$$A \in \mathbb{R}^{n \times n} \quad AI = IA = A, \text{ where } I = \begin{bmatrix} 1 & 0 & ... & 0 \\ 0 & 1 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & ... & 1 \end{bmatrix}$$

Special Vectors

Definition 5 – Coordinate Vector A vector with all 0s and a single 1, at position k, is the k^{th}-coordinate vector.

$$\mathbf{e}_k = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Definition 6 – Matrix Transpose If $A \in \mathbb{R}^{m\times n}$, $C = A^T \in \mathbb{R}^{n\times m}$ is

$$c_{i,j} = aj, i \quad (1 \le i \le n, 1 \le j \le m)$$

Example 2 Find the transpose:

$$\begin{bmatrix} -7 & -5 & 6 \\ -1 & -8 & 10 \end{bmatrix}^T$$

Definition 7 – Symmetric Matrix If $A \in \mathbb{R}^{n \times n}$ satisfies $A = A^T$, A is said to be symmetric.

Remark 2 If $\mu \in \mathbb{R}$ and $A \in \mathbb{R}^{m \times n}$, $C = \mu A \in \mathbb{R}^{m \times n}$ is

$$c_{i,j} = \mu a_{i,j} \quad (i = 1:m, j = 1:n)$$

Example 3 Simplify

$$3 \begin{bmatrix} 1 & -2 \\ -3 & \frac{1}{2} \end{bmatrix}$$

Remark 3 Scalar multiplication in `scipy`/`numpy` uses operator $*$.

```
A = np.array([[1,-2],[-3,0.5]])
B = 3 * A
```

Definition 8 – Matrix Addition If $A \in \mathbb{R}^{m\times n}$, $B \in \mathbb{R}^{m\times n}$, matrix sum $C = A + B \in \mathbb{R}^{m\times n}$ is

$$c_{i,j} = a_{i,j} + b_{i,j} \quad (i = 1 : m, j = 1 : n)$$

Example 4 Simplify

$$\begin{bmatrix} -2 & -3 & 3 \\ 4 & -5 & -3 \end{bmatrix} + \begin{bmatrix} 7 & 5 & 2 \\ -9 & -3 & 8 \end{bmatrix}$$

Remark 4 Matrices must be conformable (same shape) for addition.

Remark 5 Matrix addition in `scipy` uses + operator.

```
A = np.array([[-2.0,-3,3],[4,-5,-3]])
B = np.array([[7.0,5,2],[-9,-3,8]])
C = A + B
```

Definition 9 – Matrix Multiplication If $A \in \mathbb{R}^{m\times s}$, $B \in \mathbb{R}^{s\times n}$, matrix product $C = AB \in \mathbb{R}^{m\times n}$ is

$$c_{i,j} = \sum_{k=1}^{s} a_{i,k} b_{k,j} \quad (i = 1 : m, j = 1 : n)$$

Example 5 Simplify

$$\begin{bmatrix} -1 & 5 & -4 \\ -4 & 1 & 2 \end{bmatrix} \begin{bmatrix} -2 & -1 & 0 \\ 3 & 3 & 2 \\ -1 & -1 & 2 \end{bmatrix}$$

Remark 6 Note: $AB \neq BA$ in general!

Remark 7 In `scipy`: `scipy.dot(A, B)` or `scipy.matmul(A, B)`.
Requires `A` and `B` satisfies:

```
scipy.shape(A)[1] == scipy.shape(B)[0]
```

Definition 10 – Matrix Inverse Square matrix $A \in \mathbb{R}^{n\times n}$ is invertible (or regular or nonsingular) if there exists $B \in \mathbb{R}^{n\times n}$ such that

$$AB = BA = I$$

Inverse of A is unique and denoted A^{-1}; A must be square.

Example 6 Simplify

$$\begin{bmatrix} -2 & -2 & 4 \\ 1 & -3 & 0 \\ -4 & 4 & 1 \end{bmatrix}^{-1}$$

Remark 8 For any scalars $\mu \in \mathbb{R}$:

1. $A + 0 = 0 + A = A$
2. $IA = AI = A$
3. $A(B + C) = AB + AC$ for any $A \in \mathbb{R}^{m\times s}$; $B, C \in \mathbb{R}^{s\times n}$
4. $(AB)C = A(BC)$ for any $A \in \mathbb{R}^{m\times k}$, $B \in \mathbb{R}^{k\times \ell}$, $C \in \mathbb{R}^{\ell\times n}$
5. $\mu(AB) = (\mu A)B = A(\mu B)$ for any $A \in \mathbb{R}^{m\times s}$, $B \in \mathbb{R}^{s\times n}$
6. $(\mu A)^T = \mu A^T$
7. $(A + B)^T = A^T + B^T$ for any matrices $A, B \in \mathbb{R}^{m\times n}$
8. $(AB)^T = B^T A^T$ for any $A \in \mathbb{R}^{m\times s}$, $B \in \mathbb{R}^{s\times n}$
9. $(AB)^{-1} = B^{-1}A^{-1}$ for any invertible $A, B \in \mathbb{R}^{n\times n}$

Theorem 1 – Nonsingular Matrix Properties For $A \in \mathbb{R}^{n\times n}$, the following properties are equivalent:

1. The inverse of A exists; i.e., A is nonsingular.
2. $\det(A) \neq 0$.
3. For every $\mathbf{b} \in \mathbb{R}^n$, system $A\mathbf{x} = \mathbf{b}$ has unique solution $\mathbf{x} \in \mathbb{R}^n$.
4. $A\mathbf{x} = \mathbf{0} \implies \mathbf{x} = \mathbf{0}$.
5. The rows of A form a basis for $\mathbb{R}^n$.
6. The columns of A form a basis for $\mathbb{R}^n$.
7. The map $\{A : \mathbb{R}^n \text{ into } \mathbb{R}^n\}$ is one-to-one (injective).
8. The map $\{A : \mathbb{R}^n \text{ into } \mathbb{R}^n\}$ is onto (surjective).
9. 0 is not an eigenvalue of A.

Remark 9 Rule for matrix multiplication permits representation of linear systems of equations using matrices and vectors.

e.g, linear system of equations

$$
\begin{aligned}
2x_1 + x_2 + x_3 &= 4 \\
4x_1 + 3x_2 + 3x_3 + x_4 &= 11 \\
8x_1 + 7x_2 + 9x_3 + 5x_4 &= 29 \\
6x_1 + 7x_2 + 9x_3 + 8x_4 &= 30
\end{aligned}
$$

can be written as $A\mathbf{x} = \mathbf{b}$ with

$$
\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} =
\begin{bmatrix} 4 \\ 11 \\ 29 \\ 30 \end{bmatrix}
$$

Remark 10 We can solve linear systems of equations in `scipy` with the linalg module using

`scipy.linalg.solve`

scipy.linalg actually calls the LAPACK and BLAS routines, optimized for your hardware under Linux.

Simplest use:

```
import scipy
A = np.array([[7.0, 5.0, 2.0],
              [-3.0, 1.0, 0.0],
              [0.0, 12.0, -3.0]])
b = np.array([[3.0],[-4.0],[0.0]])
x = scipy.linalg.solve(A,b)
```

Remark 11 Never solve linear systems by computing A^{-1} and $\mathbf{x} = A^{-1}\mathbf{b}$! Use `scipy`'s built-in solvers that avoid inverting matrices. We will see that computing A^{-1} explicitly is slow and often leads to large numerical errors.

Definition 11 – Diagonal System

Given vector $\mathbf{b} = (b_1, ..., b_n)^T \in \mathbb{R}^n$, and diagonal matrix D, wish to solve linear system of equations $D\mathbf{x} = \mathbf{b}$, i.e.,

$$\begin{bmatrix} d_1 & & & \\ & d_2 & & \\ & & \ddots & \\ & & & d_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

Solution of $D\mathbf{x} = \mathbf{b}$ directly computable:

$$x_k = \frac{b_k}{d_k} \quad (d_k \neq 0, k = 1 : n)$$

Example 7 Solve the system of linear equations:

$$\begin{bmatrix} 2 & & \\ & 3 & \\ & & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \\ 1 \end{bmatrix}$$

Definition 12 – **Upper Triangular Systems** Given $\mathbf{b} = (b_1, ..., b_n)^T \in (R)^n$ and U upper triangular, wish to solve linear system of equations $U\mathbf{x} = \mathbf{b}$, i.e.,

$$\begin{bmatrix} U_{1,1} & U_{1,2} & ... & U_{1,n} \\ & U_{2,2} & ... & U_{2,n} \\ & & \ddots & \vdots \\ & & & U_{n,n} \end{bmatrix}$$

Remark 12 Solution of $U\mathbf{x} = \mathbf{b}$ through backward substitution:

$$x_k = \frac{1}{U_{k,k}}(b_k - \sum_{j=k+1}^{n} U_{k,j}x_k) \quad (k = 1 : n)$$

Example 8 Solve the linear system of equations:

$$\begin{bmatrix} 2 & 3 & -2 \\ & 3 & 5 \\ & & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \\ 1 \end{bmatrix}$$

Definition 13 – Lower Triangular Systems Given $\mathbf{b} = (b_1, ..., b_n)^T \in (R)^n$ and L lower triangular, wish to solve linear system of equations $L\mathbf{x} = \mathbf{b}$, i.e.,

$$\begin{bmatrix} L_{1,1} & & & \\ L_{2,1} & L_{2,2} & & \\ \vdots & \vdots & \ddots & \\ L_{n,1} & L_{n,2} & ... & L_{n,n} \end{bmatrix}$$

Remark 13 Solution of $L\mathbf{x} = \mathbf{b}$ through backward substitution:

$$x_k = \frac{1}{L_{k,k}}(b_k - \sum_{j=1}^{k-1} L_{k,j}x_j) \quad (k = 1 : n)$$

Example 9 Solve the linear system of equations:

$$\begin{bmatrix} 2 & & \\ 3 & 3 & \\ -2 & 5 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 5 \\ 9 \\ 1 \end{bmatrix}$$

Gaussian Elimination

Remark 14 Gaussian elimination transforms a general system $A\mathbf{x} = \mathbf{b}$ into an easy-to-solve system.

Elementary Row Operations

- Interchanging two equations: $R_i \leftrightarrow R_j$.
- Multiplying an equation by a nonzero scalar: $R_i \leftarrow \lambda R_j$.
- Adding a multiple of an equation to another: $R_i \leftarrow R_i + \lambda R_j$.

Central Idea

Reduce square system of linear equations to upper triangular system by sequence of elementary row operations.

Example 10 Consider solving linear system of equations:

$$\begin{aligned} 2x_1 + x_2 + x_3 &= 4 \\ 4x_1 + 3x_2 + 3x_3 + x_4 &= 11 \\ 8x_1 + 7x_2 + 9x_3 + 5x_4 &= 29 \\ 6x_1 + 7x_2 + 9x_3 + 8x_4 &= 30 \end{aligned}$$

Write the system as $A\mathbf{x} = \mathbf{b}$, form augmented system, and carry out elimination.

Definition 14 – Pivot Element

Pivot element on diagonal used to zero out entries.

$$\text{pivot} = A_{k,k} \quad (k = 1 : n-1)$$

Definition 15 – Multiplier

Multiplier for eliminating $A_{k,\ell}$ with pivot element $A_{k,k}$ is

$$m_{k,\ell} := \frac{A_{k,\ell}}{A_{k,k}} \quad (k = 1 : n-1, \ell = k+1 : n)$$

Remark 15 Multiply kth now by $-m_{k,\ell}$ and add to ℓth row.

Zeros out kth column below diagonal pivot element.

For the moment, assume no row interchanges.

Remark 16 Key Observation

Each stage of elimination amounts to multiplying A on the left by unit lower triangular matrix with negatives of multipliers in pivot column.

LU Decomposition

Remark 17 Gaussian elimination is equivalent to finding L and U such that:

- L is the lower triangular matrix (ones on diagonal),
- U is upper triangular matrix,
- $A = LU$.

Definition 16 A pair of matrices L and U with the propertiews above is an LU decomposition (or LU factorisation or Gauss factorisation) of A.

Remark 18

1. Start by writing down $n \times n$ matrix A and identity matrix.
2. Carry out steps of Gaussian elimination, transforming A to upper triangular ("row echelon") form.
3. At each stage of elimination, write multiplier $m_{k,\ell}$ in (k, ℓ) position of identity matrix $(k = 1 : n-1, \ell = k+1 : n)$.
4. At end, result is upper triangular U and unit lower triangular L.

Even if A is invertible, procedure above may not work.
Pivoting required for some matrices.

Example 11 Start from square matrix A and an identity matrix. Find the triangular factors L and U such that $LU = A$, with:

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

Remark 19 Pseudo-code for LU decomposition without pivoting:

```
Require: A ∈ ℝ^{n×n}
  U ← A
  L ← I                                  ▷ (initialize matrices)
  j ← 1
  while j ≤ n − 1 do                     ▷ (loop through pivot columns)
      i ← j + 1
      while i ≤ n do
          L_{i,j} ← U_{i,j} / U_{j,j}    ▷ (store multiplier in L matrix)
          U_{i,j:n} ← U_{i,j:n} − L_{i,j} U_{j,j:n}   ▷ (update row i of U matrix)
      end while
  end while
```

Theorem 2 For a given nonsingular matrix $A \in \mathbb{R}^{n\times n}$, the LU decomposition $A = LU$ exists and is unique if and only if all the leading principal submatrices of A are nonsingular.

Note: a leading submatrix is obtained from a matrix A by extracting its first k rows and columns: $A(1:k, 1:k)$.

- LU decomposition $A = LU$ has L lower unit triangular and U upper triangular.
- Not always possible to find $A = LU$ for A nonsingular.
- When A nonsingular, always possible to find permutation P such that $PA = LU$, i.e., so that PA has a Gauss (LU) factorisation.