2072U Computational Science I
Winter 2022

Week	Topic
1	Introduction
1–2	Solving nonlinear equations in one variable
3–4	Solving systems of (non)linear equations
5–6	Computational complexity
6–8	Interpolation and least squares
8–10	Integration & differentiation
10–12	Additional Topics

1. Newton-Raphson iteration
2. Central questions
3. Reminder: matrices and SciPy
4. Matrix operations
5. Matrix algebra
6. Systems of linear equations
7. Easy-to-solve systems
8. Gaussian elimination
9. LU decomposition

Newton iteration can be generalized to n equations with n unknowns.
Alternative derivation in 1D:

$$f(x+\delta x) \approx f(x) + f'(x)\delta x = 0 \quad \Rightarrow \quad \delta x = -\frac{f(x)}{f'(x)}$$

Now in 2D. We want to find x_1 and x_2 such that

$$\begin{aligned} f_1(x_1, x_2) &= 0 \\ f_1(x_1, x_2) &= 0 \end{aligned}$$

Note that, in general, we need the same number of equations and unknowns to find (isolated) solutions. . .

$$f_1(x_1+\delta x_1, x_2+\delta x_2) \approx f_1(x_1,x_2) + \frac{\partial f_1}{\partial x_1}(x_1,x_2)\delta x_1 + \frac{\partial f_1}{\partial x_2}(x_1,x_2)\delta x_2$$
$$f_2(x_1+\delta x_1, x_2+\delta x_2) \approx f_2(x_1,x_2) + \frac{\partial f_2}{\partial x_1}(x_1,x_2)\delta x_1 + \frac{\partial f_2}{\partial x_2}(x_1,x_2)\delta x_2$$

In matrix form:

$$\begin{pmatrix} \frac{\partial f_1}{\partial x_1}(x_1,x_2) & \frac{\partial f_1}{\partial x_2}(x_1,x_2) \\ \frac{\partial f_2}{\partial x_1}(x_1,x_2) & \frac{\partial f_2}{\partial x_2}(x_1,x_2) \end{pmatrix} \begin{pmatrix} \delta x_1 \\ \delta x_2 \end{pmatrix} = - \begin{pmatrix} f_1(x_1,x_2) \\ f_2(x_1,x_2) \end{pmatrix}$$

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

$$\begin{aligned}
a_{1,1}x_1 + a_{1,2}x_2 + \cdots + a_{1,n}x_n &= b_1 \\
a_{2,1}x_1 + a_{2,2}x_2 + \cdots + a_{2,n}x_n &= b_2 \\
\vdots \qquad\qquad &\quad \vdots \\
a_{n,1}x_1 + a_{n,2}x_2 + \cdots + a_{2,n}x_n &= b_n
\end{aligned}$$

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 the unknowns, and $\mathbf{b} \in \mathbb{R}^n$.

Example for $n = 4$ (i.e. with 4 unknowns: x_1, x_2, x_3, x_4.)

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

or in matrix form:

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

where

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

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?

In Python (use SciPy): `import scipy`, `import scipy.linalg`

Matrices

- Matrix $A \in \mathbb{R}^{m\times n}$ is rectangular array of numbers

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

- Numbers $a_{i,j}$ = elements of A = entries of A.
- First index (i) of element $a_{i,j}$ = row index.
- Second index (j) of element $a_{i,j}$ = column index.

Example: `A = np.array([[1,2],[3,4]])`

Vectors

- n-vector: "skinny" matrix (dimension $n \times 1$ or $1 \times n$)

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \\ x_n \end{pmatrix} \quad \text{or} \quad \mathbf{x}^T = \begin{pmatrix} x_1, & x_2, & \cdots, & x_{n-1}, & x_n \end{pmatrix}$$

- Elements $x_i =$ components of $\mathbf{x}$.
- Convention: vectors generically 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

Zero matrix $0 \in \mathbb{R}^{m\times n}$

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

Identity matrix $I \in \mathbb{R}^{n\times n}$

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

Examples: `scipy.zeros((3,2))`, `scipy.identity(3)`

Special vectors

- Coordinate vectors: all 0s, one 1.
- k^{th}-coordinate vector is

$$\mathbf{e}_k := I_{:,k} \in \mathbb{R}^{n\times 1},$$

  i.e., k^{th} column of $I \in \mathbb{R}^{n\times n}$.
- Convenient notation for matrix algorithms.

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

Example with $n = 4$:

```
I=scipy.identity(4)
e1=I[:,[0]], e2=I[:,[1]]
e3=I[:,[2]], e4=I[:,[3]]
```

Matrix transpose

If $A \in \mathbb{R}^{m \times n}$, $C = A^T \in \mathbb{R}^{n \times m}$ is

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

$$\text{e.g., } \begin{bmatrix} -7 & -5 & 6 \\ -1 & -8 & 10 \end{bmatrix}^T = \begin{bmatrix} -7 & -1 \\ -5 & -8 \\ 6 & 10 \end{bmatrix}$$

- Use SCIPY.TRANSPOSE or `.T` for the transpose of matrices:

```
A = np.array([[-7,-5,6],[-1,-8,10]])
C = scipy.transpose(A)
C = A.T
```

- If $A \in \mathbb{R}^{n \times n}$ satisfies $A = A^T$, A is said to be symmetric.

Scalar multiplication

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

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

$$\text{e.g.,} \qquad 3\begin{bmatrix} 1 & -2 \\ -3 & 1/2 \end{bmatrix} = \begin{bmatrix} 3 & -6 \\ -9 & 3/2 \end{bmatrix}$$

- Scalar multiplication in SciPy/NumPy uses operator $\star$

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

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

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

e.g., $$\begin{bmatrix} -2 & -3 & 3 \\ 4 & -5 & -3 \end{bmatrix} + \begin{bmatrix} 7 & 5 & 2 \\ -9 & -3 & 8 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 5 \\ -5 & -8 & 5 \end{bmatrix}$$

- 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
```

- Matrices must be conformable (same shape) for addition.

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

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

$$\text{e.g.,} \quad \begin{bmatrix} -1 & 5 & -4 \\ -4 & 1 & 2 \end{bmatrix} \begin{bmatrix} -2 & -1 & 0 \\ 3 & 3 & 2 \\ -1 & -1 & 2 \end{bmatrix} = \begin{bmatrix} 21 & 20 & 2 \\ 9 & 5 & 6 \end{bmatrix}$$

- In SciPy: `scipy.dot(A,B)` or `scipy.matmul(A,B)`
- Requires `A` and `B` satisfies `scipy.shape(A)[1]==scipy.shape(B)[0]`.
- Note: $AB \neq BA$ in general!

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,

$$\text{e.g., } \begin{bmatrix} -2 & -2 & 4 \\ 1 & -3 & 0 \\ -4 & 4 & 1 \end{bmatrix}^{-1} = \begin{bmatrix} 1/8 & -3/4 & -1/2 \\ 1/24 & -7/12 & -1/6 \\ 1/3 & -2/3 & -1/3 \end{bmatrix}$$

- Use routine `scipy.linalg.inv` for computing matrix inverse:

```
A=np.array([[-2,-3,3],[4,-5,-3]])
B=scipy.linalg.inv(A)
```

Algebra

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 $A \in \mathbb{R}^{m\times k}$, $B \in \mathbb{R}^{k\times l}$, $C \in \mathbb{R}^{l\times n}$
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 (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. $A\mathbf{x} = \mathbf{0} \Rightarrow \mathbf{x} = \mathbf{0}$
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 $\{A : \mathbb{R}^n$ into $\mathbb{R}^n\}$ is one-to-one (injective)*
8. *The map $\{A : \mathbb{R}^n$ into $\mathbb{R}^n\}$ is onto (surjective)*
9. 0 *is not an eigenvalue of A*

- 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

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

We can solve linear systems of equations in SciPy with the linalg module using `scipy.linalg.solve`.

- Simplest use:

```
In [1]: import scipy
In [2]: A = np.array([[7.0,5.0,2.0],
              [-3.0,1.0,0.0],[0.0,12.0,-3.0]])
In [3]: b = np.array([[3.0],[-4.0],[0.0]])
In [4]: x = scipy.linalg.solve(A,b)
```

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

- Present goal: to understand what `scipy.linalg.solve` does:
    - Gaussian elimination,
    - *LU* decomposition,
    - pivoting.

- Present goal: to understand what `scipy.linalg.solve` does:
  - Gaussian elimination,
  - *LU* decomposition,
  - pivoting.
- See `docs.scipy.org` reference guide.

- Present goal: to understand what `scipy.linalg.solve` does:
  - Gaussian elimination,
  - *LU* decomposition,
  - pivoting.
- See `docs.scipy.org` reference guide.

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

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 error*.

Diagonal systems:

- Given vector $\mathbf{b} = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$, 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:

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

Solve the linear system of equations

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

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

Equations are completely decoupled.

Upper triangular systems:

- Given $\mathbf{b} = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$ and U upper triangular, wish to solve linear system of equations $U\mathbf{x} = \mathbf{b}$, i.e.,

$$\begin{bmatrix} U_{1,1} & U_{1,2} & \cdots & U_{n,n} \\ & U_{2,2} & \cdots & U_{n,n} \\ & & \ddots & \\ & & & U_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

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

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

Solve the linear system of equations

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

$$\begin{aligned} -4x_3 = 1 \quad &\Rightarrow \quad x_3 = -\frac{1}{4} \\ 3x_2 + 5x_3 = 9 \quad &\Rightarrow \quad x_2 = \frac{1}{3}\left(9 - 5\left(-\frac{1}{4}\right)\right) = \frac{41}{12} \\ 2x_1 + 3x_2 - 2x_3 = 5 \quad &\Rightarrow \quad x_1 = \frac{1}{2}\left(5 - 3\left(\frac{41}{12}\right) + 2\left(-\frac{1}{4}\right)\right) = -\frac{23}{8} \end{aligned}$$

Lower triangular systems:

- Given $\mathbf{b} = (b_1, \ldots, b_n)^T \in \mathbb{R}^n$ and L lower triangular, wish to solve linear system of equations $L\mathbf{x} = \mathbf{b}$, i.e.,

$$\begin{bmatrix} L_{1,1} & & & \\ L_{2,1} & L_{2,2} & & \\ \vdots & \vdots & \ddots & \\ L_{n,1} & L_{n,2} & \cdots & L_{n,n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}$$

- Solution of $L\mathbf{x} = \mathbf{b}$ through forward substitution:

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

Solve the linear system of equations

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

$$\begin{aligned} 2x_1 = 5 \quad &\Rightarrow \quad x_1 = \frac{5}{2} \\ 3x_1 + 3x_2 = 9 \quad &\Rightarrow \quad x_2 = \frac{1}{3}\left(9 - 3\left(\frac{5}{2}\right)\right) = \frac{1}{2} \\ -2x_1 + 5x_2 - 4x_3 = 1 \quad &\Rightarrow \quad x_3 = -\frac{1}{4}\left(1 + 2\left(\frac{5}{2}\right) - 5\left(\frac{1}{2}\right)\right) = -\frac{7}{8} \end{aligned}$$

Gaussian elimination

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: $R_i \leftarrow \lambda R_i$
  - Adding a multiple of an equation to another: $R_i \leftarrow R_i + \lambda R_j$
- Applying elementary row operations to linear system of equations preserves solution of original system

Gaussian elimination

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: $R_i \leftarrow \lambda R_i$
  - Adding a multiple of an equation to another: $R_i \leftarrow R_i + \lambda R_j$
- Applying elementary row operations to linear system of equations preserves solution of original system

Central Idea

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

Example:

- Consider solving linear system of equations

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

Example:

- 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 system as $A\mathbf{x} = \mathbf{b}$ with

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

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \\ \\ \end{array}$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \quad \leftarrow R_2 - (4/2)R_1$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & & & \\ & & & & \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \\ \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & & & \\ & & & & \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \\ \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & & & & \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} \color{red}{2} & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ \color{red}{6} & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \color{red}{\leftarrow R_4 - (6/2)R_1} \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & & & & \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & 4 & 6 & 8 & 18 \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & 4 & 6 & 8 & 18 \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & 4 & 6 & 8 & 18 \end{array}\right] \begin{array}{l} \\ \\ \leftarrow R_3 - (3/1)R_2 \\ \\ \end{array}$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & 4 & 6 & 8 & 18 \end{array}\right] \begin{array}{l} \\ \\ \leftarrow R_3 - (3/1)R_2 \\ \\ \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & 2 & 2 & 4 \\ & & & & \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & {\color{red}1} & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & {\color{red}4} & 6 & 8 & 18 \end{array}\right] \begin{array}{l} \\ \\ \leftarrow R_3 - (3/1)R_2 \\ {\color{red}\leftarrow R_4 - (4/1)R_2} \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & 2 & 2 & 4 \\ & & & & \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & {\color{red}1} & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & {\color{red}4} & 6 & 8 & 18 \end{array}\right] \begin{array}{l} \\ \\ \leftarrow R_3 - (3/1)R_2 \\ {\color{red}\leftarrow R_4 - (4/1)R_2} \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & 2 & 2 & 4 \\ & & {\color{red}2} & {\color{red}4} & {\color{red}6} \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & 4 & 6 & 8 & 18 \end{array}\right] \begin{array}{l} \\ \\ \leftarrow R_3 - (3/1)R_2 \\ \leftarrow R_4 - (4/1)R_2 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & 2 & 2 & 4 \\ & & 2 & 4 & 6 \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & 4 & 6 & 8 & 18 \end{array}\right] \begin{array}{l} \\ \\ \leftarrow R_3 - (3/1)R_2 \\ \leftarrow R_4 - (4/1)R_2 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & \color{red}{2} & 2 & 4 \\ & & \color{red}{2} & 4 & 6 \end{array}\right] \begin{array}{l} \\ \\ \\ \color{red}{\leftarrow R_4 - (2/2)R_3} \end{array}$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & 4 & 6 & 8 & 18 \end{array}\right] \begin{array}{l} \\ \\ \leftarrow R_3 - (3/1)R_2 \\ \leftarrow R_4 - (4/1)R_2 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & 2 & 2 & 4 \\ & & 2 & 4 & 6 \end{array}\right] \begin{array}{l} \\ \\ \\ \leftarrow R_4 - (2/2)R_3 \end{array} \qquad \left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & 2 & 2 & 4 \\ & & & 2 & 2 \end{array}\right]$$

Form augmented system and carry out elimination

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ 4 & 3 & 3 & 1 & 11 \\ 8 & 7 & 9 & 5 & 29 \\ 6 & 7 & 9 & 8 & 30 \end{array}\right] \begin{array}{l} \\ \leftarrow R_2 - (4/2)R_1 \\ \leftarrow R_3 - (8/2)R_1 \\ \leftarrow R_4 - (6/2)R_1 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & 3 & 5 & 5 & 13 \\ & 4 & 6 & 8 & 18 \end{array}\right] \begin{array}{l} \\ \\ \leftarrow R_3 - (3/1)R_2 \\ \leftarrow R_4 - (4/1)R_2 \end{array}$$

$$\left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & 2 & 2 & 4 \\ & & 2 & 4 & 6 \end{array}\right] \begin{array}{l} \\ \\ \\ \leftarrow R_4 - (2/2)R_3 \end{array} \qquad \left[\begin{array}{cccc|c} 2 & 1 & 1 & 0 & 4 \\ & 1 & 1 & 1 & 3 \\ & & 2 & 2 & 4 \\ & & & 2 & 2 \end{array}\right]$$

We arrive at upper triangular system $U\mathbf{x} = \mathbf{c}$ to solve.

Observations:

- Pivot element on diagonal used to zero out entries

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

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

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

- Multiply kth row by $-m_{k,\ell}$ and add to ℓth row
- Zeros out kth column below diagonal pivot element.
- For the moment, assume no row interchanges.

Observations:

- Pivot element on diagonal used to zero out entries

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

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

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

- Multiply kth row by $-m_{k,\ell}$ and add to ℓth row
- Zeros out kth column below diagonal pivot element.
- For the moment, assume no row interchanges.

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.

In our example:

$$\begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 3 & 5 & 5 \\ 0 & 4 & 6 & 8 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -2 & 1 & 0 & 0 \\ -4 & 0 & 1 & 0 \\ -3 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{pmatrix} = L_1 A$$

$$\begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 4 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & -3 & 1 & 0 \\ 0 & -4 & 0 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 3 & 5 & 5 \\ 0 & 4 & 6 & 8 \end{pmatrix} = L_2 L_1 A$$

$$\begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 2 & 4 \end{pmatrix} = L_3 L_2 L_1 A$$

So that, finally,

$$L_3\,L_2\,L_1\,A = U \;\text{ or }\; A = (L_3\,L_2\,L_1)^{-1}\,U = L_1^{-1}\,L_2^{-1}\,L_3^{-1}\,U = L\,U$$

where

$$L = L_1^{-1}\,L_2^{-1}\,L_3^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 4 & 3 & 1 & 0 \\ 3 & 4 & 1 & 1 \end{pmatrix}$$

If two matrices are lower (upper) triangular, then so is their product and their inverse!

Note, because L_1, L_2, and L_3 are matrix representations of elementary row operations, their inverses are easy to find, and thus L is easy to find.

LU decomposition:

Key principle

Gaussian elimination is equivalent to finding L & U such that

LU decomposition:

Key principle

Gaussian elimination is equivalent to finding L & U such that

- L is unit lower triangular matrix (ones on diagonal),

LU decomposition:

Key principle

Gaussian elimination is equivalent to finding L & U such that

- L is unit lower triangular matrix (ones on diagonal),
- U is upper triangular matrix,

LU decomposition:

Key principle

Gaussian elimination is equivalent to finding L & U such that

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

LU decomposition:

Key principle

Gaussian elimination is equivalent to finding L & U such that

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

LU Decomposition

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

Procedure for *LU* decomposition

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 invertible, procedure above may not work.
- Pivoting required for some matrices (see Lec 6).

Example: Start from square matrix A and an identity matrix

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

$$\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Example: Start from square matrix A and an identity matrix

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} \quad \leftarrow R_2 - (4/2)R_1, \quad m_{2,1} := 2$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & & \\ & & & \end{bmatrix} \qquad \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix}$$

Example: Start from square matrix A and an identity matrix

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} \begin{array}{ll} \\ \leftarrow R_2 - (4/2)R_1, & m_{2,1} := 2 \\ \leftarrow R_3 - (8/2)R_1, & m_{3,1} := 4 \\ \\ \end{array}$$

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

Example: Start from square matrix A and an identity matrix

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} \begin{array}{ll} \\ \leftarrow R_2 - (4/2)R_1, & m_{2,1} := 2 \\ \leftarrow R_3 - (8/2)R_1, & m_{3,1} := 4 \\ \leftarrow R_4 - (6/2)R_1, & m_{4,1} := 3 \end{array}$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix} \qquad \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 4 & & 1 & \\ 3 & & & 1 \end{bmatrix}$$

Example: Start from square matrix A and an identity matrix

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} \begin{array}{ll} \\ \leftarrow R_2 - (4/2)R_1, & m_{2,1} := 2 \\ \leftarrow R_3 - (8/2)R_1, & m_{3,1} := 4 \\ \leftarrow R_4 - (6/2)R_1, & m_{4,1} := 3 \end{array}$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & {\color{red}1} & 1 & 1 \\ & {\color{red}3} & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix} \begin{array}{ll} \\ \\ \leftarrow R_3 - (3/1)R_2, & {\color{red}m_{3,2} := 3} \\ \\ \end{array} \qquad \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 4 & {\color{red}3} & 1 & \\ 3 & & & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & {\color{red}2} & {\color{red}2} \\ & & & \end{bmatrix}$$

Example: Start from square matrix A and an identity matrix

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} \begin{matrix} \\ \leftarrow R_2 - (4/2)R_1, & m_{2,1} := 2 \\ \leftarrow R_3 - (8/2)R_1, & m_{3,1} := 4 \\ \leftarrow R_4 - (6/2)R_1, & m_{4,1} := 3 \end{matrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & {\color{red}1} & 1 & 1 \\ & 3 & 5 & 5 \\ & {\color{red}4} & 6 & 8 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow R_3 - (3/1)R_2, & m_{3,2} := 3 \\ \leftarrow R_4 - (4/1)R_2, & {\color{red}m_{4,2} := 4} \end{matrix} \qquad \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 4 & 3 & 1 & \\ 3 & {\color{red}4} & & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & {\color{red}2} & {\color{red}4} \end{bmatrix}$$

Example: Start from square matrix A and an identity matrix

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} \begin{matrix} \\ \leftarrow R_2 - (4/2)R_1, \quad m_{2,1} := 2 \\ \leftarrow R_3 - (8/2)R_1, \quad m_{3,1} := 4 \\ \leftarrow R_4 - (6/2)R_1, \quad m_{4,1} := 3 \end{matrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix} \begin{matrix} \\ \\ \leftarrow R_3 - (3/1)R_2, \quad m_{3,2} := 3 \\ \leftarrow R_4 - (4/1)R_2, \quad m_{4,2} := 4 \end{matrix} \qquad \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 4 & 3 & 1 & \\ 3 & 4 & \color{red}{1} & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & \color{red}{2} & 2 \\ & & \color{red}{2} & 4 \end{bmatrix} \begin{matrix} \\ \\ \\ \leftarrow R_4 - (2/2)R_3, \quad \color{red}{m_{4,3} := 1} \end{matrix} \qquad \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & & \color{red}{2} \end{bmatrix}$$

Example: Start from square matrix A and an identity matrix

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} \quad \begin{matrix} \\ \leftarrow R_2 - (4/2)R_1, & m_{2,1} := 2 \\ \leftarrow R_3 - (8/2)R_1, & m_{3,1} := 4 \\ \leftarrow R_4 - (6/2)R_1, & m_{4,1} := 3 \end{matrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix} \quad \begin{matrix} \\ \\ \leftarrow R_3 - (3/1)R_2, & m_{3,2} := 3 \\ \leftarrow R_4 - (4/1)R_2, & m_{4,2} := 4 \end{matrix} \qquad L := \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 4 & 3 & 1 & \\ 3 & 4 & 1 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & 2 & 4 \end{bmatrix} \quad \begin{matrix} \\ \\ \\ \leftarrow R_4 - (2/2)R_3, & m_{4,3} := 1 \end{matrix} \qquad U := \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & & 2 \end{bmatrix}$$

We now have triangular factors L and U such that $LU = A$

Pseudo-code for LU decomposition:

LU decomposition without pivoting

Input: $A \in \mathbb{R}^{n\times n}$
$U \leftarrow A$, $L \leftarrow I$ (initialise matrices)
for $j = 1 : n-1$ (loop through pivot columns)
  for $i = j+1 : n$
    $L_{i,j} \leftarrow U_{i,j}/U_{j,j}$ (store multiplier in L matrix)
    $U_{i,j:n} \leftarrow U_{i,j:n} - L_{i,j}U_{j,j:n}$ (update row i of U matrix)
  end for
end for
Output: Matrices L and U

Existence of LU decomposition $A = LU$.

Proposition

For a given nonsingular matrix $A \in \mathbb{R}^{n \times n}$, the LU decomposition $A = LU$ exists and is unique iff 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 unit lower 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