UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 401 (Fall 2025)

Numerical Methods

Lecture 5: LU Decomposition & Partial Pivoting

Chong Liu
Department of Computer Science

Sep 15, 2025

Agenda

- Another direct linear system solvers:
  - LU matrix decomposition-based solver
- A method to avoid relative errors in solvers:
  - Partial pivoting

Recap: Linear systems

- An example of linear systems
  - Any linear system can always be rewritten in matrix form

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 &= b_3
\end{aligned}
\qquad
\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}
\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}
=
\begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix}
$$

- More generally, $Ax = b$
  - A is an $m \times n$ matrix
  - x is an n-dimensional vector
  - b is an m-dimensional vector
- Problem: given A and b, how can you solve x?
- New problem: given A and multiple different b, how can you solve x efficiently?

LU decomposition

- $A = LU$
  - L is a lower triangular matrix, U is a upper triangular matrix
  - Note this decomposition may **not** be unique

- In-class exercise:
  - Find an LU decomposition of $A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}$

  - Solutions:
    - $u_{11} = a_{11} = 2,\ u_{12} = a_{12} = 1,\ u_{13} = a_{13} = 1$
    - $\ell_{21} = a_{21}/u_{11} = 4/2 = 2,\ \ \ell_{31} = a_{31}/u_{11} = -2/2 = -1$
    - $u_{22} = a_{22} - \ell_{21}u_{12} = -6 - 2 \cdot 1 = -8$
    - $u_{23} = a_{23} - \ell_{21}u_{13} = 0 - 2 \cdot 1 = -2$
    - $\ell_{32} = (a_{32} - \ell_{31}u_{12})/u_{22} = (7 - (-1) \cdot 1)/(-8) = 8/(-8) = -1$
    - $u_{33} = a_{33} - \ell_{31}u_{13} - \ell_{32}u_{23} = 2 - (-1) \cdot 1 - (-1) \cdot (-2) = 2 + 1 - 2 = 1$

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -8 & -2 \\ 0 & 0 & 1 \end{bmatrix}.$$

LU decomposition for linear systems

- $A = LU$
  - L is a lower triangular matrix, U is a upper triangular matrix

- $Ax = b$ becomes $LUx = b$
- Solution:
  - Step 1: Solve y from $Ly = b$
  - Step 2: Solve x from $Ux = y$

- Discussion: what’s the runtime complexity of Step 1, 2?

Steps 1 and 2 run in $O(n)$!

- Why? Because L is a lower triangular matrix, U is a upper triangular matrix. Simply run back substitution!

- Recap of LU
  - $Ax = b$ becomes $LUx = b$
  - Solution:
    - Step 1: Solve y from $Ly = b$
    - Step 2: Solve x from $Ux = y$

In-class exercise: LU decomposition for linear systems

System: $A\mathbf{x} = \mathbf{b}$ with

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 4 & -6 & 0 \\ -2 & 7 & 2 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 5 \\ -2 \\ 9 \end{bmatrix}$$

- Step 1: Solve y from $Ly = b$
- Step 2: Solve x from $Ux = y$

- Solution: y=[5, -12, 2], x=[1, 1, 2]

Runtime of LU decomposition for linear systems

- LU decomposition: $O(n^3)$
- Back substitution: $O(n)$
- Total: $O(n^3)$

- Back to our motivation: given A and multiple different b, how can you solve x efficiently?
  - LU decomposition should be your choice!
  - Because you can save time by saving L and U for the same A.

What's the runtime of Gaussian elimination?

- Guess time! What's your quick answer?

- Let's take a closer look:
  - At step 1 we eliminate the first column: for each of the $n - 1$ rows below, we update $n - 1$ entries in that row $\Rightarrow \approx (n-1)^2$ element updates.
  - At step 2 we work on the $(n-1) \times (n-1)$ block $\Rightarrow (n-2)^2$ updates.
  - …
  - At step k we update a $(n-k) \times (n-k)$ block $\Rightarrow (n-k)^2$ updates.

  - So, $O(n^3)$ because $\sum_{m=1}^{n-1} m^2 = \frac{n(n-1)(2n-1)}{6} \sim \frac{n^3}{3}$

Both LU decomposition and Gaussian elimination run in $O(n^3)$

- But LU decomposition is usually considered to be better since it's scalable – works for multiple different b.
- And LU decomposition has been written as functions in Python and Matlab.
  - Python: scipy.linalg.lu
  - Matlab: lu()

```
A = [10 -7 0
     -3  2 6
      5 -1 5];
```

```
[L,U] = lu(A)
```

```
L = 3×3

    1.0000         0         0
   -0.3000   -0.0400    1.0000
    0.5000    1.0000         0

U = 3×3

   10.0000   -7.0000         0
         0    2.5000    5.0000
         0         0    6.2000
```

Two examples of linear systems

- Can you solve it using Gaussian elimination? $\hat{A} = \left(\begin{array}{cc|c} 0 & 1 & 1 \\ 1 & 1 & 2 \end{array}\right)$
  - Solution: $x = [1, 1]$
- How about this? $\tilde{A} = \left(\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right)$

$$\left(\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) \xrightarrow{R_2 \leftarrow R_2 - 10^{20} R_1} \left(\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 0 & -10^{20} & -10^{20} \end{array}\right)$$

  - Discussion: What's the solution?
  - Solution: $x = [0, 1]$
  - What happened? Small rounding error leads to huge relative error!

Partial pivoting prevents this issue

- How does partial pivoting work?
  - Swap rows to make pivot have the largest absolute value in its column.

$$\left(\begin{array}{cc|c} 10^{-20} & 1 & 1 \\ 1 & 1 & 2 \end{array}\right) \xrightarrow{R_1 \leftrightarrow R_2} \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 10^{-20} & 1 & 1 \end{array}\right) \xrightarrow{R_2 \leftarrow R_2 - 10^{-20} R_1} \left(\begin{array}{cc|c} 1 & 1 & 2 \\ 0 & 1 - 10^{-20} & 1 - 2 \cdot 10^{-20} \end{array}\right)$$

$$x_2 = \frac{1 - 2 \cdot 10^{-20}}{1 - 10^{-20}} \approx 1 \pm 10^{-20}$$

$$x_1 + x_2 = 2 \implies x_1 = 1 \pm 10^{-20}.$$

- Why does it work?
  - Avoid dividing by tiny numbers, reduces relative error, and makes LU numerically stable for most matrices.

Another example of partial pivoting

$$\begin{pmatrix} 2 & 1 & 3 \\ 1 & 10^{-20} & 4 \\ 7 & -20 & 5 \end{pmatrix} \xrightarrow{R_1 \leftrightarrow R_3} \begin{pmatrix} 7 & -20 & 5 \\ 1 & 10^{-20} & 4 \\ 2 & 1 & 3 \end{pmatrix} \xrightarrow{R_2 \leftarrow R_2 - (1/7)R_1, R_3 \leftarrow R_3 - (2/7)R_1} \begin{pmatrix} 7 & -20 & 5 \\ 0 & 2.857 & 3.286 \\ 0 & 6.714 & 1.5714 \end{pmatrix} \tag{11.18}$$

$$\xrightarrow{R_2 \leftrightarrow R_3} \begin{pmatrix} 7 & -20 & 5 \\ 0 & 6.714 & 1.5714 \\ 0 & 2.857 & 3.286 \end{pmatrix} \xrightarrow{R_3 \leftarrow R_3 - (2.857/6.714)R_2} \begin{pmatrix} 7 & -20 & 5 \\ 0 & 6.714 & 1.5714 \\ 0 & 0 & 2.6173 \end{pmatrix} \tag{11.19}$$