

Linear Systems and LU

CS 205A:
Mathematical Methods for Robotics, Vision, and Graphics

Doug James (and Justin Solomon)



Homework

- Homework 0:** Posted on website later today. Due next Thurs (use Gradescope)
- Late Policy:** The deadline to upload your work to Gradescope will be Thursday 11:59 pm. You have a total of 3 late periods. Using a late period means you can submit an assignment by Sunday 11:59 pm. If you exhaust your late periods, late assignments will be penalized at 50%. No work will be accepted after the late deadline.



Other Announcements

New Section Room:

- ▶ **Time:** Friday, 11:30-12:20pm
- ▶ **Place:** 260-113 (“Language Corner”)

Julia programming refs on webpage

Linear Systems

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

$$A \in \mathbb{R}^{m \times n}$$

$$\vec{x} \in \mathbb{R}^n$$

$$\vec{b} \in \mathbb{R}^m$$

Case 1: Solvable

$$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

“Completely Determined”

Case 2: No Solution

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -1 \\ 1 \end{pmatrix}$$

“Overdetermined”

Case 3: Infinitely Many Solutions

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}$$

“Underdetermined”

No Other Cases

Proposition

If $A\vec{x} = \vec{b}$ has two distinct solutions \vec{x}_0 and \vec{x}_1 , it has infinitely many solutions.

Common Misconception

Solvability *can* depend on \vec{b} !

$$\begin{pmatrix} 1 & 0 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

Dependence on Shape

Proposition

Tall matrices admit unsolvable right hand sides.

Proposition

Wide matrices admit right hand sides with infinite numbers of solutions.

For Now

All matrices will be:

- ▶ Square
- ▶ Invertible

Inverting Matrices

Do *not* compute A^{-1} if you do not need it.

- ▶ Not the same as solving $A\vec{x} = \vec{b}$
- ▶ Can be slow *and* poorly conditioned

Example

$$\begin{array}{rcl} y - z & = & -1 \\ 3x - y + z & = & 4 \\ x + y - 2z & = & -3 \end{array} \iff \left(\begin{array}{ccc|c} 0 & 1 & -1 & -1 \\ 3 & -1 & 1 & 4 \\ 1 & 1 & -2 & -3 \end{array} \right)$$

- ▶ Permute rows
- ▶ Row scaling
- ▶ Forward/back substitution

Row Operations: Permutation

$$\sigma : \{1, \dots, m\} \rightarrow \{1, \dots, m\}$$

$$P_\sigma \equiv \begin{pmatrix} - & \vec{e}_{\sigma(1)}^\top & - \\ - & \vec{e}_{\sigma(2)}^\top & - \\ & \dots & \\ - & \vec{e}_{\sigma(m)}^\top & - \end{pmatrix}$$

Row Operations: Row Scaling

$$S_a \equiv \begin{pmatrix} a_1 & 0 & 0 & \cdots \\ 0 & a_2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & a_m \end{pmatrix}$$

Row Operations: Elimination

“Scale row k by constant c and add result to row l .”

$$E \equiv I + c\vec{e}_l\vec{e}_k^T$$



Solving via Elimination Matrices

$$\left(\begin{array}{ccc|c} 0 & 1 & -1 & -1 \\ 3 & -1 & 1 & 4 \\ 1 & 1 & -2 & -3 \end{array} \right)$$

Reverse order!

Introducing Gaussian Elimination

Big idea:

General strategy to solve
linear systems via row
operations.

Elimination Matrix Interpretation

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

$$E_1 A\vec{x} = E_1 \vec{b}$$

$$E_2 E_1 A\vec{x} = E_2 E_1 \vec{b}$$

$$\vdots$$

$$\underbrace{E_k \cdots E_2 E_1 A}_{I_{n \times n}} \vec{x} = \underbrace{E_k \cdots E_2 E_1}_{A^{-1}} \vec{b}$$

Shape of Systems

$$\left(A \mid \vec{b} \right) = \left(\begin{array}{cccc|c} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{array} \right)$$

Pivot

$$\left(\begin{array}{cccc|c} \textcircled{\times} & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{array} \right)$$

Row Scaling

$$\left(\begin{array}{cccc|c} \textcircled{1} & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{array} \right)$$

Forward Substitution

$$\left(\begin{array}{cccc|c} \textcircled{1} & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{array} \right)$$

Forward Substitution

$$\left(\begin{array}{cccc|c} 1 & \times & \times & \times & \times \\ 0 & \textcircled{1} & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{array} \right)$$

Upper Triangular Form

$$\left(\begin{array}{cccc|c} 1 & \times & \times & \times & \times \\ 0 & 1 & \times & \times & \times \\ 0 & 0 & 1 & \times & \times \\ 0 & 0 & 0 & \textcircled{1} & \times \end{array} \right)$$

Back Substitution

$$\left(\begin{array}{cccc|c} 1 & \times & \times & 0 & \times \\ 0 & 1 & \times & 0 & \times \\ 0 & 0 & 1 & 0 & \times \\ 0 & 0 & 0 & \textcircled{1} & \times \end{array} \right)$$

Back Substitution

$$\left(\begin{array}{cccc|c} 1 & \times & 0 & 0 & \times \\ 0 & 1 & 0 & 0 & \times \\ 0 & 0 & 1 & 0 & \times \\ 0 & 0 & 0 & 1 & \times \end{array} \right)$$

Back Substitution

$$\left(\begin{array}{cccc|c} \textcircled{1} & 0 & 0 & 0 & \times \\ 0 & 1 & 0 & 0 & \times \\ 0 & 0 & 1 & 0 & \times \\ 0 & 0 & 0 & 1 & \times \end{array} \right)$$

Steps of Gaussian Elimination

1. Forward substitution: For each row $i = 1, 2, \dots, m$
 - ▶ Scale row to get pivot 1
 - ▶ For each $j > i$, subtract multiple of row i from row j to zero out pivot column
2. Backward substitution: For each row $i = m, m - 1, \dots, 1$
 - ▶ For each $j < i$, zero out rest of column

Total Running Time

$$O(n^3)$$

Problem

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

Even Worse

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

Pivoting

Pivoting

Permuting rows and/or columns to avoid dividing by small numbers or zero.

- ▶ *Partial* pivoting
- ▶ *Full* pivoting

$$\begin{pmatrix} 1 & 10 & -10 \\ 0 & 0.1 & 9 \\ 0 & 4 & 6.2 \end{pmatrix}$$

Reasonable Use Case

$$A\vec{x}_1 = \vec{b}_1$$

$$A\vec{x}_2 = \vec{b}_2$$

$$\vdots$$

Can we *restructure* A to make this more efficient?

Does each solve take $O(n^3)$ time?

Observation

Steps of Gaussian elimination depend only on structure of A .

Avoid repeating identical arithmetic on A ?

Another Clue:

Upper Triangular Systems

$$\left(\begin{array}{cccc|c} 1 & \times & \times & \times & \times \\ 0 & 1 & \times & \times & \times \\ 0 & 0 & 1 & \times & \times \\ 0 & 0 & 0 & \textcircled{1} & \times \end{array} \right)$$

After Back Substitution

$$\left(\begin{array}{cccc|c} 1 & \times & \times & 0 & \times \\ 0 & 1 & \times & 0 & \times \\ 0 & 0 & 1 & 0 & \times \\ \textcircled{0} & \textcircled{0} & \textcircled{0} & 1 & \times \end{array} \right)$$

No need to subtract the 0's explicitly!
 $O(n)$ time

Next Pivot: Same Observation

$$\left(\begin{array}{cccc|c} 1 & \times & 0 & 0 & \times \\ 0 & 1 & 0 & 0 & \times \\ \textcircled{0} & \textcircled{0} & 1 & \textcircled{0} & \times \\ 0 & 0 & 0 & 1 & \times \end{array} \right)$$

Observation

Triangular systems can be solved in $O(n^2)$ time.

Upper Triangular Part of A

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

$$\vdots$$

$$M_k \cdots M_1 A \vec{x} = M_k \cdots M_1 \vec{b}$$

Define:

$$U \equiv M_k \cdots M_1 A$$

Lower Triangular Part

$$U = M_k \cdots M_1 A$$

$$\Rightarrow A = (M_1^{-1} \cdots M_k^{-1}) U$$

$$\equiv LU$$

Why Is L Triangular?

$$S \equiv \text{diag}(a_1, a_2, \dots)$$

$$E \equiv I + c\vec{e}_\ell\vec{e}_k^\top$$

Proposition

The product of triangular matrices is triangular.

Solving Systems Using LU

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

$$\Rightarrow LU\vec{x} = \vec{b}$$

1. Solve $L\vec{y} = \vec{b}$ using forward substitution.
2. Solve $U\vec{x} = \vec{y}$ using backward substitution.

$O(n^2)$ (given LU factorization)

LU: Compact Storage

$$\begin{pmatrix} U & U & U & U \\ L & U & U & U \\ L & L & U & U \\ L & L & L & U \end{pmatrix}$$

Assumption: Diagonal elements of L are 1.

Warning: Do not multiply this matrix!

Computing LU Factorization

Small modification of forward substitution step to keep track of L .¹

¹See textbook for pseudocode.

Question

Does every A admit a factorization $A = LU$?

Recall: Pivoting

Pivoting

Permuting rows and/or columns to avoid dividing by small numbers or zero.

- ▶ *Partial* pivoting
- ▶ *Full* pivoting

$$\begin{pmatrix} 1 & 10 & -10 \\ 0 & 0.1 & 9 \\ 0 & 4 & 6.2 \end{pmatrix}$$

Pivoting by Swapping Columns

$$\underbrace{(E_k \cdots E_1)}_{\text{elimination}} \cdot A \cdot \underbrace{(P_1 \cdots P_\ell)}_{\text{permutations}} \cdot \underbrace{(P_\ell^\top \cdots P_1^\top)}_{\text{inv. permutations}} \vec{x}$$

$$= (E_k \cdots E_1) \vec{b}$$

$$\Downarrow$$

$$A = LUP$$