

Lecture 06 - A

§2.1, 2.2 Gaussian Elimination

T. Takahashi

Department of Mathematics and Statistics
University of New Mexico



Goals

We will answer these questions:

- How do you solve a general linear system?
- How expensive is it to perform Gaussian elimination?
- What are some problems with the "naive" algorithm?



Special Systems

- **Diagonal** - Costs $\mathcal{O}(n)$ FLOPs
$$\begin{bmatrix} \bullet & 0 & 0 & 0 \\ 0 & \bullet & 0 & 0 \\ 0 & 0 & \bullet & 0 \\ 0 & 0 & 0 & \bullet \end{bmatrix}$$

- **Banded** - Costs $\mathcal{O}(m^2n)$ FLOPs
$$\begin{bmatrix} \bullet & \bullet & 0 & 0 \\ \bullet & \bullet & \bullet & 0 \\ 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \end{bmatrix}$$

- **Triangular** - Costs $\mathcal{O}(n^2)$ FLOPs
$$\begin{bmatrix} \bullet & \bullet & \bullet & \bullet \\ 0 & \bullet & \bullet & \bullet \\ 0 & 0 & \bullet & \bullet \\ 0 & 0 & 0 & \bullet \end{bmatrix}$$

Question

- What if A is a general matrix?
- What is the cost of solving $Ax = b$?

Gaussian Elimination



Simple 2×2 Example

Example

Solve

$$x_1 + 3x_2 = 5$$

$$2x_1 + 4x_2 = 6$$

- Subtract 2 times the first equation from the second equation

$$x_1 + 3x_2 = 5$$

$$-2x_2 = -4$$



Simple 2×2 Example

Example

Solve

$$\begin{aligned}x_1 + 3x_2 &= 5 \\ 2x_1 + 4x_2 &= 6\end{aligned}$$

- Subtract 2 times the first equation from the second equation

$$\begin{aligned}x_1 + 3x_2 &= 5 \\ -2x_2 &= -4\end{aligned}$$

- The system is now in upper triangular form

$$\begin{bmatrix} 1 & 3 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ -4 \end{bmatrix}$$

We can now use **backward substitution**.



Simple 2×2 Example

- Solve the second equation for x_2

$$x_2 = \frac{-4}{-2} = 2$$

- Substitute x_2 into the first equation and solve for x_1 .

$$x_1 = 5 - (3)(2) = -1$$



Simple 2×2 Example

- Solve the second equation for x_2

$$x_2 = \frac{-4}{-2} = 2$$

- Substitute x_2 into the first equation and solve for x_1 .

$$x_1 = 5 - (3)(2) = -1$$

Key Idea

Change a system to triangular form and then do backward substitution!

$$\begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ 6 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 3 \\ 0 & -2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 5 \\ -4 \end{bmatrix} \rightarrow \text{back sub}$$



General Procedure for $n \times n$

Part 1

1. Form the **augmented system** - matrix formed by appending $[A|b]$

$$\left[\begin{array}{cccc|c} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{array} \right]$$

(The x 's represent numbers, they are generally not the same values.)

2. Choose your **pivot element** - the element which controls the multiplier. Start with the very first element A_{11} . The row containing the pivot element is called the **pivot row**.

$$\left[\begin{array}{ccccc} \boxed{x} & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{array} \right]$$



General Procedure for $n \times n$

Part 2

3. Eliminate elements under the pivot element in the first column by performing **row operations** - multiply the pivot row by a scalar and add to a row below

$$\begin{array}{c} \left[\begin{array}{ccccc} \boxed{x} & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{array} \right] \longrightarrow \left[\begin{array}{ccccc} \boxed{x} & x & x & x & x \\ 0 & x' & x' & x' & x' \\ x & x & x & x & x \\ x & x & x & x & x \end{array} \right] \\ \longrightarrow \left[\begin{array}{ccccc} \boxed{x} & x & x & x & x \\ 0 & x' & x' & x' & x' \\ 0 & x' & x' & x' & x' \\ x & x & x & x & x \end{array} \right] \\ \longrightarrow \left[\begin{array}{ccccc} \boxed{x} & x & x & x & x \\ 0 & x' & x' & x' & x' \\ 0 & x' & x' & x' & x' \\ 0 & x' & x' & x' & x' \end{array} \right] \end{array}$$

(x' indicates a value that has been changed once.)



General Procedure for $n \times n$

Part 3

4. Choose next pivot element: diagonal element in the second row.
5. Eliminate elements under the pivot element in the second column.

$$\begin{bmatrix} x & x & x & x & x \\ 0 & \boxed{x'} & x' & x' & x' \\ 0 & x' & x' & x' & x' \\ 0 & x' & x' & x' & x' \end{bmatrix} \longrightarrow \begin{bmatrix} x & x & x & x & x \\ 0 & \boxed{x'} & x' & x' & x' \\ 0 & 0 & x'' & x'' & x'' \\ 0 & x' & x' & x' & x' \end{bmatrix}$$
$$\longrightarrow \begin{bmatrix} x & x & x & x & x \\ 0 & \boxed{x'} & x' & x' & x' \\ 0 & 0 & x'' & x'' & x'' \\ 0 & 0 & x'' & x'' & x'' \end{bmatrix}$$

(x'' indicates a value that has been changed twice.)



General Procedure for $n \times n$

Part 4

6. Repeat until the system is upper triangular.

$$\begin{bmatrix} x & x & x & x & x \\ 0 & x' & x' & x' & x' \\ 0 & 0 & \boxed{x''} & x'' & x'' \\ 0 & 0 & x'' & x'' & x'' \end{bmatrix} \longrightarrow \begin{bmatrix} x & x & x & x & x \\ 0 & x' & x' & x' & x' \\ 0 & 0 & \boxed{x''} & x'' & x'' \\ 0 & 0 & 0 & x''' & x''' \end{bmatrix}$$

(x''' indicates a value that has been changed three times.)



General Procedure for $n \times n$

Part 4

6. Repeat until the system is upper triangular.

$$\begin{bmatrix} x & x & x & x & x \\ 0 & x' & x' & x' & x' \\ 0 & 0 & \boxed{x''} & x'' & x'' \\ 0 & 0 & x'' & x'' & x'' \end{bmatrix} \longrightarrow \begin{bmatrix} x & x & x & x & x \\ 0 & x' & x' & x' & x' \\ 0 & 0 & \boxed{x''} & x'' & x'' \\ 0 & 0 & 0 & x''' & x''' \end{bmatrix}$$

(x''' indicates a value that has been changed three times.)

Definition: Gaussian Elimination

The procedure of changing the augmented system to triangular form is called **Gaussian elimination** (GE).



Gaussian Elimination

Method: Gaussian Elimination

- **Given:** $n \times n$ matrix A , $n \times 1$ right hand side b
- **Output:** Upper triangular augmented system
- **Idea:** This algorithm does not assume anything about A . It is a general solver.

Getting the solution to $Ax = b$

- 1 Construct the augmented system
- 2 Perform Gaussian elimination

$$\begin{bmatrix} x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \\ x & x & x & x & x \end{bmatrix} \longrightarrow \begin{bmatrix} x & x & x & x & x \\ 0 & x' & x' & x' & x' \\ 0 & 0 & x'' & x'' & x'' \\ 0 & 0 & 0 & x''' & x''' \end{bmatrix}$$

- 3 Solve system with back sub

GE Example

Part 1

Example

Perform Gaussian elimination on the system $Ax = b$ where

$$A = \begin{bmatrix} -3 & 2 & -1 \\ 6 & -6 & 7 \\ 3 & -4 & 4 \end{bmatrix} \quad b = \begin{bmatrix} -1 \\ -7 \\ -6 \end{bmatrix}$$

1. Form the **augmented system**

$$\tilde{A} = [A \ b] = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 6 & -6 & 7 & -7 \\ 3 & -4 & 4 & -6 \end{array} \right]$$



GE Example

Part 2

2. Choose 3 as the first pivot element.

$$\tilde{A} = \left[\begin{array}{ccc|c} \boxed{-3} & 2 & -1 & -1 \\ 6 & -6 & 7 & -7 \\ 3 & -4 & 4 & -6 \end{array} \right]$$

3. How many times does $\boxed{-3}$ divide into 6? Perform row operation to zero out the element below.

- $xmult = 6 / -3 = -2$
- $R_2 \leftarrow R_2 - (xmult)R_1$

$$\tilde{A}_{(1)} = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 3 & -4 & 4 & -6 \end{array} \right]$$



GE Example

Part 3

4. How many times does $\boxed{-3}$ divide into 3? Perform row operation to zero out the element below.

- $xmult = 3 / -3 = -1$
- $R_3 \leftarrow R_3 - (xmult)R_1$

$$\tilde{A}_{(1)} = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 3 & -4 & 4 & -6 \end{array} \right]$$

$$\tilde{A}_{(1)} = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & -2 & 3 & -7 \end{array} \right]$$



GE Example

Part 4

5. Choose element -2 as the next pivot

$$\tilde{A}_{(1)} = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & \boxed{-2} & 5 & -9 \\ 0 & -2 & 3 & -7 \end{array} \right]$$

6. How many times does $\boxed{-2}$ divide into -2 ? Perform row operation to zero out the element below.

- ▶ $xmult = -2 / -2 = 1$
- ▶ $R_3 \leftarrow R_3 - (xmult)R_2$

$$\tilde{A}_{(2)} = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & 0 & -2 & 2 \end{array} \right]$$

Done.



Solving the System

Option 1: Back Sub

$$\tilde{A}_{(2)} = \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & 0 & -2 & 2 \end{array} \right]$$

Solve by back substitution to get

$$x_3 = \frac{2}{-2} = -1$$

$$x_2 = \frac{1}{-2} (-9 - 5x_3) = 2$$

$$x_1 = \frac{1}{-3} (-1 - 2x_2 + x_3) = 2$$

Option 2: Gauss-Jordan

Continue doing operations to transform matrix to an identity

$$\left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & 0 & \boxed{1} & -1 \end{array} \right]$$

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

$\vdots$

$$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & -1 \end{array} \right]$$

Gaussian Elimination Algorithm

```
1 given A, b
2
3 for k = 1...n-1
4     for i = k+1...n
5         for j = k...n
6              $a_{ij} = a_{ij} - (a_{ik}/a_{kk})a_{kj}$ 
7         end
8          $b_i = b_i - (a_{ik}/a_{kk})b_k$ 
9     end
10 end
```

Two ways to Optimize

1. Precompute the multiplier a_{ik}/a_{kk} before line 5
2. Line 6, if $j = k$ the value a_{ij} is always zero. Just skip it.



Gaussian Elimination Algorithm

```
1 given  $A, b$ 
2
3 for  $k = 1 \dots n - 1$ 
4     for  $i = k + 1 \dots n$ 
5          $xmult = a_{ik}/a_{kk}$  % precompute multiplier
6          $a_{ik} = 0$  % zero out elements below pivot
7         for  $j = k + 1 \dots n$  % start loop at  $j=k+1$ 
8              $a_{ij} = a_{ij} - (xmult)a_{kj}$ 
9         end
10         $b_i = b_i - (xmult)b_k$ 
11    end
12 end
```

Listing 1: Optimized Version



Gaussian Elimination Algorithm: Storing Multipliers

```
1 given  $A, b$ 
2
3 for  $k = 1 \dots n - 1$ 
4   for  $i = k + 1 \dots n$ 
5      $xmult = a_{ik} / a_{kk}$ 
6      $a_{ik} = xmult$  % save multiplier
7     for  $j = k + 1 \dots n$ 
8        $a_{ij} = a_{ij} - (xmult)a_{kj}$ 
9     end
10     $b_i = b_i - (xmult)b_k$ 
11  end
12 end
```

- This variant saves the multipliers by writing them into the matrix
- We will discuss why this might be desirable



Zeroing vs. Storing Multipliers

- Zeroing out version

$$\left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 6 & -6 & 7 & -7 \\ 3 & -4 & 4 & -6 \end{array} \right] \rightarrow \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & 0 & -2 & 2 \end{array} \right]$$

- Storing the multipliers *xmult* in the lower triangle

$$\left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 6 & -6 & 7 & -7 \\ 3 & -4 & 4 & -6 \end{array} \right] \rightarrow \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ -2 & -2 & 5 & -9 \\ -1 & 1 & -2 & 2 \end{array} \right]$$

Question

Why would you store multipliers?



Cost of GE



Naive Gaussian Elimination Algorithm

Question

What is the cost of Gaussian elimination?

or rather

What is the cost to solve a general system?

- Solve = Gaussian Elimination + Back Substitution
- We already know Backward Substitution is $\mathcal{O}(n^2)$
- What about Gaussian elimination?



Cost for a 3×3

Part 1

Question

What is the exact number of FLOPs for the 3×3 example?

$$\left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 6 & -6 & 7 & -7 \\ 3 & -4 & 4 & -6 \end{array} \right] \longrightarrow \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & 0 & -2 & 2 \end{array} \right]$$

- Step 1: Eliminate a_{21}

$$\left[\begin{array}{ccc|c} \boxed{-3} & 2 & -1 & -1 \\ \textcolor{red}{6} & -6 & 7 & -7 \\ 3 & -4 & 4 & -6 \end{array} \right] \longrightarrow \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 3 & -4 & 4 & -6 \end{array} \right]$$

- ▶ 1 division: $6 / -3 = -2$
- ▶ 3 multiplications: $-2R_1 = [\ 6 \ -4 \ 2 \ | \ 2]$
- ▶ 3 subtractions: $R_2 - 2R_1 = [\ 0 \ -2 \ 5 \ | \ -9]$

Note: We don't count $6 - (-3)(-2) = 0$ at A_{21} because the result it always 0



Cost for a 3×3

Part 2

- Step 2: Eliminate a_{31}

$$\left[\begin{array}{ccc|c} \boxed{-3} & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ \textcolor{red}{3} & -4 & 4 & -6 \end{array} \right] \rightarrow \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & -2 & 3 & -7 \end{array} \right]$$

- 1 division: $3 / -3 = -1$
- 3 multiplications: $-R_1 = [\ 3 \ -2 \ 1 \ | \ 1]$
- 3 subtractions: $R_3 - (-R_1) = [\ 0 \ -2 \ 3 \ | \ -7]$

- Step 3: Eliminate a_{23}

$$\left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & \boxed{-2} & 5 & -9 \\ 0 & \textcolor{red}{-2} & 3 & -7 \end{array} \right] \rightarrow \left[\begin{array}{ccc|c} -3 & 2 & -1 & -1 \\ 0 & -2 & 5 & -9 \\ 0 & 0 & -2 & 2 \end{array} \right]$$

- 1 division: $-2 / -2 = 1$
- 2 multiplications: $(1)R_1 = [\ 0 \ -2 \ 5 \ | \ -9]$
- 2 subtractions: $R_3 - R_1 = [\ 0 \ 0 \ -2 \ | \ 2]$



Cost for a 3×3

Part 3

- FLOP summary

Row	Step	Add	Multiply	Divide	Total
1	Eliminate element a_{21}	3	3	1	7
1	Eliminate element a_{31}	3	3	1	7
2	Eliminate element a_{32}	2	2	1	5

- Total cost: 19 FLOPs

What's the pattern?

Let $n = 3$.

- In row 1, there are $n - 1 = 2$ elements that need to be eliminated. Each elimination requires n additions, n multiplications, and 1 division.
- In row 2, there are $n - 2 = 1$ element(s) that need to be eliminated. Each elimination requires $n - 1$ additions, $n - 1$ multiplications, and 1 division.

What about a 4×4 ?

- We have 6 elimination targets:
$$\begin{bmatrix} x & x & x & x \\ x & x & x & x \\ x & x & x & x \\ x & x & x & x \end{bmatrix}$$
- This is the pattern for a 3×3

Row	Step	Add	Multiply	Divide	Total
1	Eliminate element a_{21}	3	3	1	7
1	Eliminate element a_{31}	3	3	1	7
2	Eliminate element a_{32}	2	2	1	5



What about a 4×4 ?

- We have 6 elimination targets:

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

- This is the pattern for a 3×3

Row	Step	Add	Multiply	Divide	Total
1	Eliminate element a_{21}	3	3	1	7
1	Eliminate element a_{31}	3	3	1	7
2	Eliminate element a_{32}	2	2	1	5

- This is the pattern for a 4×4

Row	Step	Add	Multiply	Divide	Total
1	Eliminate element a_{21}	4	4	1	9
1	Eliminate element a_{31}	4	4	1	9
1	Eliminate element a_{41}	4	4	1	9
2	Eliminate element a_{32}	3	3	1	7
2	Eliminate element a_{42}	3	3	1	7
3	Eliminate element a_{34}	2	2	1	5



The Pattern for 4×4

Row	Step	Add	Multiply
1	Eliminate element a_{21}	4	4
1	Eliminate element a_{31}	4	4
1	Eliminate element a_{41}	4	4
2	Eliminate element a_{32}	3	3
2	Eliminate element a_{42}	3	3
3	Eliminate element a_{34}	2	2

Let $n = 4$

- **Row 1:** $4 + 4 + 4 = n + n + n = (n - 1)n$ Adds
- **Row 2:** $3 + 3 = (n - 1) + (n - 1) = (n - 2)(n - 1)$ Adds
- **Row 3:** $2 = (n - 2) = (n - 3)(n - 2)$ Adds

There is a consistent pattern in the way the number of operations decrease.



How many additions?

Question

How many total additions are required for 4×4 system?

$$\underbrace{(n-1)n}_{\text{row 1}} + \underbrace{(n-2)(n-1)}_{\text{row 2}} + \underbrace{(n-3)(n-2)}_{\text{row 3}} = \sum_{j=2}^4 (j-1)j$$



How many additions?

Question

How many total additions are required for 4×4 system?

$$\underbrace{(n-1)n}_{\text{row 1}} + \underbrace{(n-2)(n-1)}_{\text{row 2}} + \underbrace{(n-3)(n-2)}_{\text{row 3}} = \sum_{j=2}^4 (j-1)j$$

Question

How many total additions are required for $n \times n$ system?

$$\sum_{j=2}^n j(j-1) \text{ additions}$$



Cost for a $n \times n$

Part 1

- Next, let's extrapolate the results for the 3×3 to determine cost for a general $n \times n$.
- Key is to count operations by rows
- Operation count:

Row j	Add	Multiply	Divide
1	$n(n-1)$	$n(n-1)$	$n-1$
2	$(n-1)(n-2)$	$(n-1)(n-2)$	$n-2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n-1$	2	2	1
Total:	$\sum_{j=2}^n j(j-1)$	$\sum_{j=2}^n j(j-1)$	$\sum_{j=1}^{n-1} j$



Cost for a $n \times n$

Part 2

- Total operations as summations

add	$\sum_{j=2}^n j(j-1) \approx \sum_{j=1}^n j^2$
multiply	$\sum_{j=2}^n j(j-1) \approx \sum_{j=1}^n j^2$
divide	$\sum_{j=1}^{n-1} j$

- Let's use the formulas $\sum_{j=1}^p j = \frac{p(p+1)}{2}$ and $\sum_{j=1}^p j^2 = \frac{p(p+1)(2p+1)}{6}$

add	$\frac{n(n-1)(2n-1)}{6}$
multiply	$\frac{n(n-1)(2n-1)}{6}$
divide	$\frac{n(n-1)}{2}$



Cost for a $n \times n$

Part 3

- Solve = Gaussian Elimination + Back Substitution
- **Gaussian elimination**

add	$\frac{n(n-1)(2n-1)}{6}$
multiply	$\frac{n(n-1)(2n-1)}{6}$
divide	$\frac{n(n-1)}{2}$

- **Back substitution**

add-subtracts	$\frac{n(n-1)}{2}$
multiply-divides	$\frac{n(n+1)}{2}$



Total Cost for Naive Gaussian Elimination

Combining the cost of forward elimination, updating b , and backward substitution gives

$$\begin{aligned}\text{add-subtracts} \quad & \frac{n(n-1)(2n-1)}{6} + \frac{n(n-1)}{2} + \frac{n(n-1)}{2} \\ &= \frac{n(n-1)(2n+5)}{6} = \frac{1}{3}n^3 + \mathcal{O}(n^2)\end{aligned}$$

$$\begin{aligned}\text{multiply-divides} \quad & \frac{n(n^2-1)}{3} + \frac{n(n-1)}{2} + \frac{n(n+1)}{2} \\ &= \frac{n(n^2+3n-1)}{3} = \frac{1}{3}n^3 + \mathcal{O}(n^2)\end{aligned}$$

Key Idea

- The cost to solve a general system $Ax = b$ is

$$\frac{2}{3}n^3 + \mathcal{O}(n^2) \quad \text{FLOPs}$$

- $\mathcal{O}(n^3)$ is expensive but it is necessary

Why “Naive”?



Question

Why is the algorithm called the “Naive” Gaussian elimination?

- It is called naive because of the simplistic **choice of pivot elements**

$$\begin{bmatrix} \boxed{x} & x & x & x & x \\ x & \boxed{x} & x & x & x \\ x & x & \boxed{x} & x & x \\ x & x & x & x & x \end{bmatrix}$$

- Naive GE choose the pivot elements to be a_{ii} for $i = 1$ to $n - 1$
- There are *drawbacks* to this method



Zero on a_{11}

- Consider performing GE on this matrix

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

- Naively choosing the pivot $a_{11} = 0$ results in division by zero

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

$$R_2 \leftarrow R_2 - \frac{4}{\textcolor{red}{0}} R_1$$

- Naive GE will not work whenever $a_{11} = 0$



Another Example

- Try Naive GE on

$$A = \begin{bmatrix} 2 & 4 & -2 \\ 1 & 2 & 4 \\ -2 & -3 & 8 \end{bmatrix} \quad b = \begin{bmatrix} -4 \\ 5 \\ 7 \end{bmatrix}$$

- Choose $a_{11} = 2$ as the pivot element

$$\left[\begin{array}{ccc|c} \boxed{2} & 4 & -2 & -4 \\ 1 & 2 & 4 & 5 \\ -2 & -3 & 8 & 7 \end{array} \right] \quad \begin{array}{l} R_2 \leftarrow -\frac{1}{2}R_1 + R_2 \\ R_3 \leftarrow R_1 + R_3 \end{array}$$

- Result:

$$\left[\begin{array}{ccc|c} 2 & 4 & -2 & -4 \\ 0 & \boxed{0} & 5 & 7 \\ 0 & 1 & 6 & -3 \end{array} \right]$$

- Problem!** The next pivot element a_{22} is a zero!



Small Pivots

- But this is **even worse**

$$A = \begin{bmatrix} 10^{-16} & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$



Small Pivots

- But this is **even worse**

$$A = \begin{bmatrix} 10^{-16} & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- After one row operation...

$$\begin{bmatrix} 10^{-16} & 2 & 3 \\ 0 & ??? & ??? \\ 7 & 8 & 9 \end{bmatrix}$$

Question

What happens to the 5 and 6 in the second row?



What's Next?

- How to deal with a zero pivot
- Three types of pivoting algorithms:
 - 1 Partial Pivoting
 - 2 Full Pivoting
 - 3 Scaled Partial Pivoting
- Cost of each pivoting algorithm
- Effect of small pivot elements in rounding

