CSE 160
Lecture 13

Numerical Linear Algebra

Announcements

- Section will be held on Friday as announced on Moodle
- Midterm Return

Today's lecture

- Gaussian Elimination (A3)

Today's lecture

- Floating Point Arithmetic
- Gaussian Elimination (A3)

Linear systems of equations

- A common task in scientific computation is to solve a system of linear equations
- Often result from of discretizing a differential equation
- Example: linear system of 2 equations in 2 unknowns

(1) $2x + 3y = 8$
(2) $3x + 2y = 7$

- Rewriting equation (1)

$x = (8\text{-}3y)/2$

- Substituting ***x*** into the LHS of equation (2)

$3(8-3y)/2 + 2y = (24-9y)/2 + 2y$
$\Rightarrow (24-9y) + 4y = 14 \Rightarrow 10 = 5y \Rightarrow y = 2$

- Back substituting the value of ***y*** into equation (1)

$x = 1$

Matrix vector equations

- Our linear system of 2 equations in 2 unknowns …
  $\boldsymbol{2x_1 + 3x_2 = 8}$
  $\boldsymbol{3x_1 + 2x_2 = 7}$
- may be conveniently expressed in matrix notation: A**x** = b

$$A = \begin{pmatrix} 2 & 3 \\ 3 & 2 \end{pmatrix}, x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}, b = \begin{pmatrix} 8 \\ 7 \end{pmatrix}$$

- When we solved for $x_1 = (8\text{-}3\ x_2)/2$ and substituted the value of x_1 into the 2nd equation, we reduced the matrix to an equivalent form

$$A = \begin{pmatrix} 2 & 3 \\ 0 & {}^{-}2.5 \end{pmatrix}, b = \begin{pmatrix} 8 \\ {}^{-}5 \end{pmatrix}$$

- We multiplied row 1 of A by 3/2 and subtracting the scaled version from row 2 of **A** and **b**
- We call this a ***rank-1 update***

Rank 1 updates

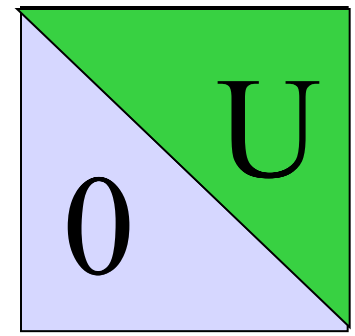
- Multiplying row 1 by 3/2: [3 9/2]
- Subtracting from row 2:
- Similarly for **b**

$$A = \begin{pmatrix} 2 & 3 \\ 3 & 2 \end{pmatrix}$$

$$A' = \begin{pmatrix} 2 & 3 \\ 0 & -2.5 \end{pmatrix}$$

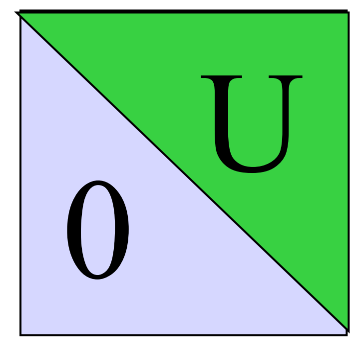
Gaussian Elimination

- The process of eliminating the non-zero values under the main diagonal is called ***Gaussian Elimination***, named after the mathematician *Johann Carl Friedrich Gauss* (1777-1855)
- Input: an n × n matrix corresponding to a linear system of n equations in n unknowns (must have non-trivial sol'n)
- Output: an n × n matrix with zeroes under the main diagonal: an ***upper triangular matrix*** **U**



Solving the system of linear equations

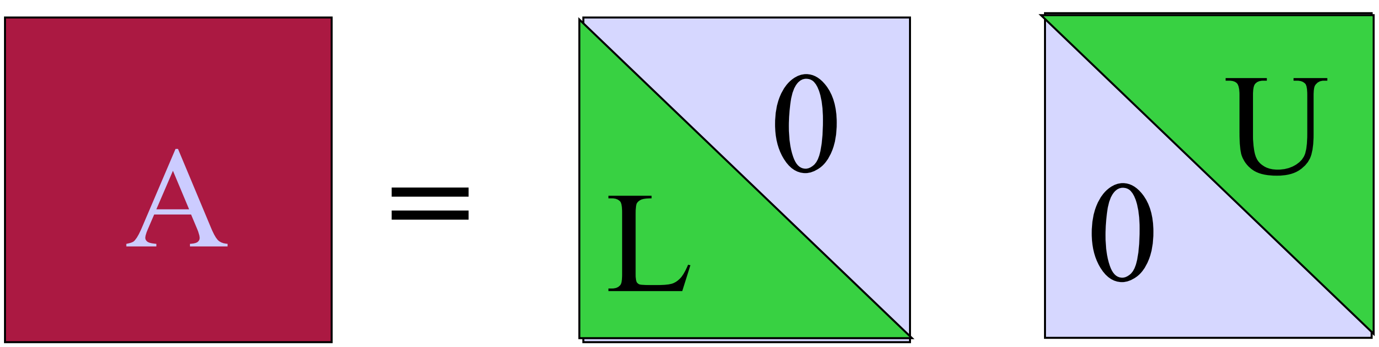
- Step 1: obtain the upper triangular matrix U …
- Step 2: solve the corresponding upper triangular system U**x** = **c** by ***back substitution***



What are we computing?

- GE computes the ***LU factorization*** A = L U, where L is a ***lower triangular matrix***
- Plugging LU into the original equation $A\mathbf{x} = \mathbf{b}$

  $A\mathbf{x} = (LU)\,\mathbf{x} = L\,(U\mathbf{x}) = L\mathbf{y} = \mathbf{b}$ where $\mathbf{y} = U\mathbf{x}$



A 3 × 3 example

- Consider the following system of equations

$$
\begin{aligned}
x_0 &+ x_1 &+ x_2 &= 3 \\
4x_0 &+ 3x_1 &+ 4x_2 &= 8 \\
9x_0 &+ 3x_1 &+ 4x_2 &= 7
\end{aligned}
$$

- We usually write the system as an ***augmented matrix***

$$
\left[\begin{array}{ccc|c}
1 & 1 & 1 & 3 \\
4 & 3 & 4 & 8 \\
9 & 3 & 4 & 7
\end{array}\right]
$$

3 × 3 example

- Multiply row 0 by 4, and subtract from row 1

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 4 & 3 & 4 & 8 \\ 9 & 3 & 4 & 7 \end{array}\right]$$

[4 3 4 8] – 4*[1 1 1 3] = [0 -1 0 -4]

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

3 × 3 example

- Multiply row 0 by 9, and subtract from row 2

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

[9 3 4 7] – 9*[1 1 1 3] = [0 -6 -5 -20]

$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 0 & -1 & 0 & -4 \\ 0 & -6 & -5 & -20 \end{array}\right]$$

3 × 3 example

- Eliminate second column
- Multiply row 1 by 6, and add to row 2

[0 -6 -5 -20] + -6*[0 -1 0 -4]
= [0 0 -5 4]

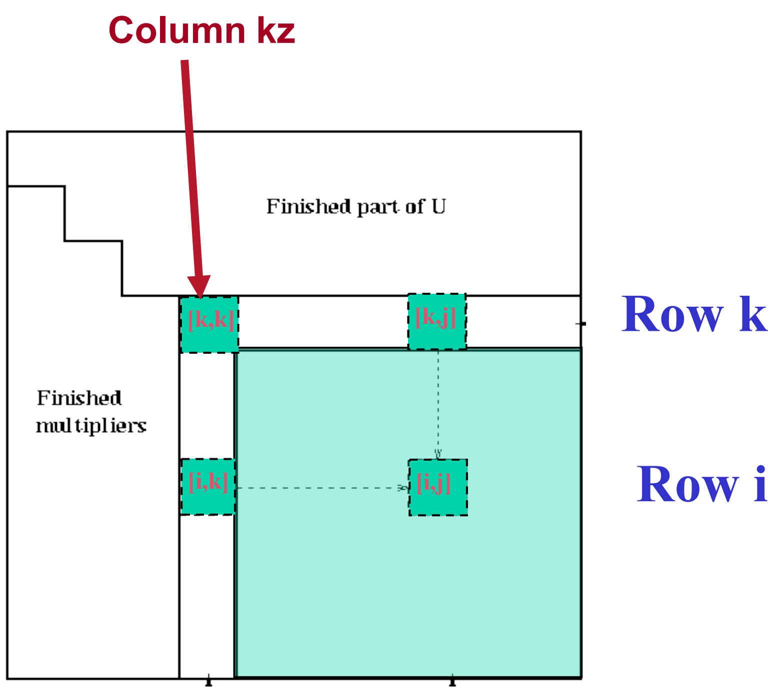
$$\left[\begin{array}{ccc|c} 1 & 1 & 1 & 3 \\ 0 & -1 & 0 & -4 \\ 0 & -6 & -5 & -20 \end{array}\right]$$

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

Visualizing the Gaussian Elimenation

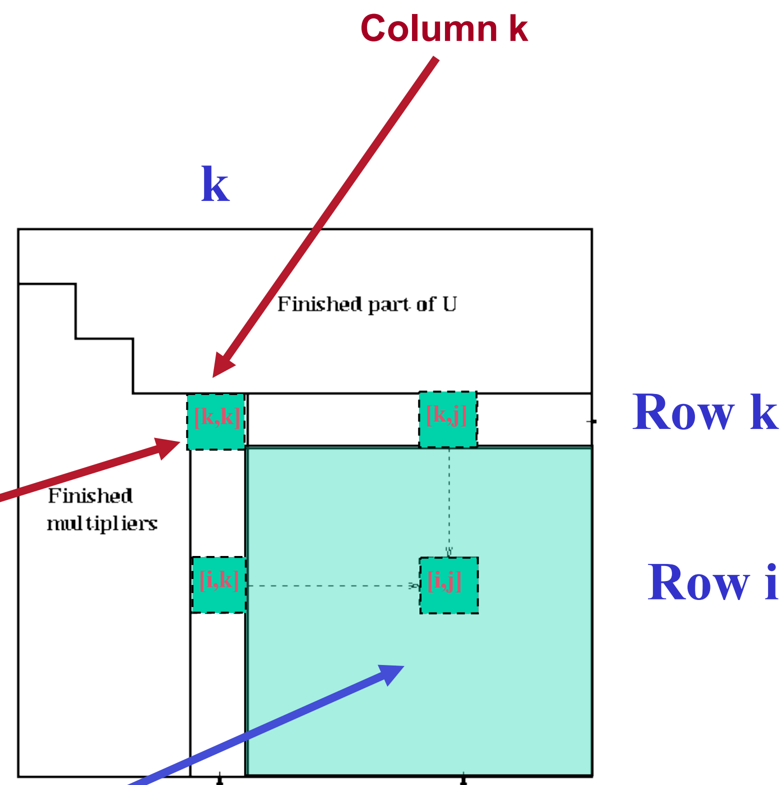
- Add multiples of each row to later rows to make A upper triangular

```
… for each column k
… zero it out below the diagonal by adding multiples of row k to later rows
for k = 0 to n-1
   … for each row i below row k
   for i = k+1 to n-1
      … add a multiple of row k to row I
      for j = k+1 to n-1
         A[i,j] -=  A[ i , k] / A[k,k] * A[k,j]
```



Eliminating the entries below the diagonal

- Add multiples of each row to lower rows to make A upper triangular
- For each column **k : 0 to n-1**
- … subtract multiples of row **k**: **A[k,k+1:n]**
  … from rows i = k+1 to n
- … to zero out column **k** below row **k**
- Multipliers **m_{ik} = A[i,k]/A[k,k]**
- … cancel the elements below the diagonal: **A[k+1:n-1,k]**
- Update only to the right of & below **A[k,k]**



for i = k+1 to n-1
    A[i,k+1:n] – = m_{ik} × A[k, k+1:n]

Cost

- To solve A**x** = **b**
  - Factorize A = LU using GE *(2/3 n^3 flops)*
  - Solve L**y** = **b** for **y** using substitution
    (n^2 flops)
  - Solve U**x** = **y** for **x** using back substitution
    (n^2 flops)
- We don't compute U explicitly unless we are solving for multiple right hand sides **b**
- Focus on factorization, which is much more expensive

Roundoff issues

- The rank-1 update step uses division …

```
A[i, k+1:n ] -= ( A[i,k]/A[k,k]) * A[k,k+1:n]
```

- We need to be able to handle vanishing denominators or ones that are very small

- Gaussian elimination will fail with this matrix

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

- But we can avoid the problem if we swap rows

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

Pivoting to avoid stability problems

- We call this process of swapping rows ***partial pivoting***
- Assume we carry 3 decimal digits of precision
- Consider the following A matrix and RHS b

$$A = \begin{bmatrix} 10^{-4} & 1 \\ 1 & 1 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

- The correct solution is

$$x = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Roundoff Error

- Eliminate zero in row 2 by subtracting $10^4 \times$ row 0

$$L|b = \left[\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 0 & 1 - 10^4 & 2 - 10^4 \end{array}\right]$$

- But $1 - 10^4$ rounds to -10^4

$$L|b = \left[\begin{array}{cc|c} 10^{-4} & 1 & 1 \\ 0 & -10^4 & -10^4 \end{array}\right]$$

- Back substituting to solve for x_2 and then x_1

  $-10^4 x_2 = -10^4 \Rightarrow \mathbf{x_2 = 1}$

- Substituting the value of x_2 into the first equation

  $10^{-4} x_1 + 1 * x_2 = 1 \Rightarrow 10^{-4} x_1 = 0 \Rightarrow \mathbf{x_1 = 0}$

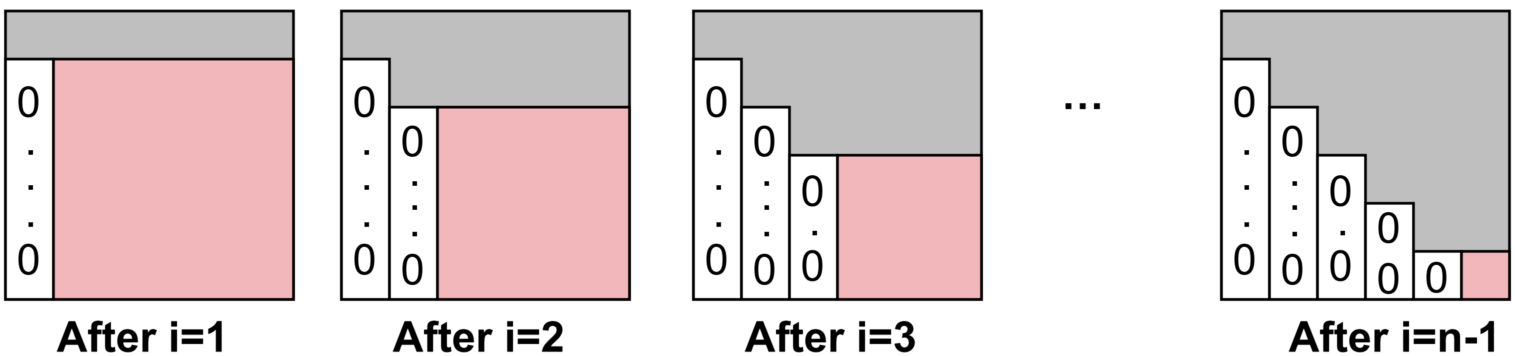
- **But the correct solution is $x_1 = x_2 = 1$**

Partial Pivoting

- Rule: pick the largest value in the column
- This is called partial pivoting, because only rows are swapped
- It can be shown that when with partial pivoting, we compute **A = P L U**, where P is a permutation matrix expressing the rows swaps
- We can also swap columns: ***full pivoting***
- But full pivoting is more expensive to implement

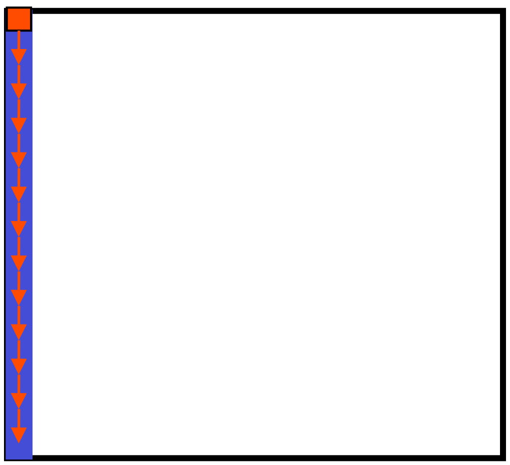
Parallelization

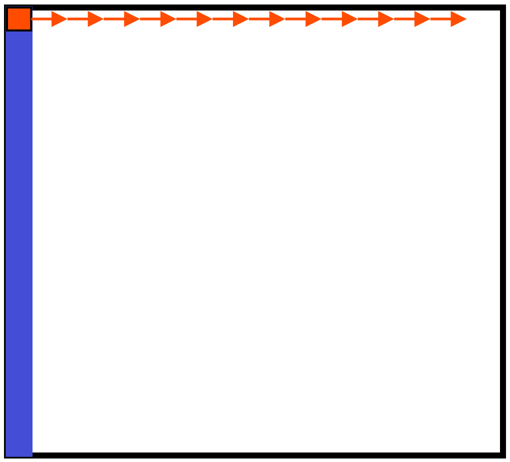
- We'll use 1D vertical strip partitioning
- The ■ represents outstanding work in succeeding k iterations
- We divide the trailing matrix + pivot column among the cores
- The pivot column is always owned by core 0



Communication and control

- Each thread in charge of eliminating N/P columns
- But as we eliminate columns, N is shrinking
- Pivot selection is serial work
- The trailing matrix multiplication parallelizes perfectly





Fin