

# Lecture 11

## Gaussian Elimination, The LU Factorization

MIT 18.335J / 6.337J  
Introduction to Numerical Methods

Per-Olof Persson  
October 17, 2006

1

## The LU Factorization

- Transform  $A \in \mathbb{C}^{m \times m}$  into upper triangular  $U$  by subtracting multiples of rows
- Each  $L_i$  introduces zeros below diagonal of column  $i$ :

$$\underbrace{L_{m-1} \cdots L_2 L_1}_{L^{-1}} A = U \implies A = LU \text{ where } L = L_1^{-1} L_2^{-1} \cdots L_{m-1}^{-1}$$

$$\begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \xrightarrow{L_1} \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \xrightarrow{L_2} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \xrightarrow{L_3} \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}$$

$A \qquad L_1 A \qquad L_2 L_1 A \qquad L_3 L_2 L_1 A$

- "Triangular triangularization"

2

## The Matrices $L_k$

- At step  $k$ , eliminate elements below  $A_{kk}$ :

$$x_k = [x_{1k} \ \cdots \ x_{kk} \ x_{k+1,k} \ \cdots \ x_{mk}]^T$$

$$L_k x_k = [x_{1k} \ \cdots \ x_{kk} \ 0 \ \cdots \ 0]^T$$

- The multipliers  $\ell_{jk} = x_{jk}/x_{kk}$  appear in  $L_k$ :

$$L_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -\ell_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -\ell_{mk} & & & 1 \end{bmatrix}$$

3

## Forming $L$

- The  $L$  matrix contains all the multipliers in one matrix (with plus signs)

$$L = L_1^{-1} L_2^{-1} \cdots L_{m-1}^{-1} = \begin{bmatrix} 1 & & & & \\ \ell_{21} & 1 & & & \\ \ell_{31} & \ell_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ \ell_{m1} & \ell_{m2} & \cdots & \ell_{m,m-1} & 1 \end{bmatrix}$$

- Define  $\ell_k = (0, \dots, 0, \ell_{k+1,k}, \dots, \ell_{m,n})$ . Then  $L_k = I - \ell_k e_k^*$ .
  - First,  $L_k^{-1} = I + \ell_k e_k^*$ , since  $e_k^* \ell_k = 0$  and  $(I - \ell_k e_k^*)(I + \ell_k e_k^*) = I - \ell_k e_k^* \ell_k e_k^* = I$
  - Also,  $L_k^{-1} L_{k+1}^{-1} = I + \ell_k e_k^* + \ell_{k+1} e_{k+1}^*$ , since  $e_k^* \ell_{k+1} = 0$  and  $(I + \ell_k e_k^*)(I + \ell_{k+1} e_{k+1}^*) = I + \ell_k e_k^* + \ell_{k+1} e_{k+1}^*$

4

## Gaussian Elimination without Pivoting

- Factorize  $A \in \mathbb{C}^{m \times m}$  into  $A = LU$ :

### Algorithm: Gaussian Elimination (no pivoting)

$$U = A, L = I$$

for  $k = 1$  to  $m - 1$

  for  $j = k + 1$  to  $m$

$$\ell_{jk} = u_{jk}/u_{kk}$$

$$u_{j,k:m} = u_{j,k:m} - \ell_{jk} u_{k,k:m}$$

- The inner loop can be written using matrix operations instead of for-loop
- Operation count  $\sim \sum_{k=1}^m 2(m-k)(m-k) \sim 2 \sum_{k=1}^m k^2 \sim 2m^3/3$

5

## Pivoting

- At step  $k$ , we used matrix element  $k, k$  as pivot and introduced zeros in entry  $k$  of remaining rows

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ & \mathbf{x_{kk}} & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & \mathbf{x_{kk}} & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix}$$

- But any other element  $i \geq k$  in column  $k$  can be used as pivot:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ & \mathbf{x_{ik}} & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ & 0 & \times & \times & \times \\ & 0 & \times & \times & \times \\ & \mathbf{x_{ik}} & \times & \times & \times \\ & 0 & \times & \times & \times \end{bmatrix}$$

6

## Pivoting

- Also, any other column  $j \geq k$  can be used:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \mathbf{\times} & \mathbf{x_{ij}} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \times & \times & \times & \times & \times \end{bmatrix} \rightarrow \begin{bmatrix} \times & \times & \times & \times & \times \\ \mathbf{\times} & \mathbf{0} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{\times} & \mathbf{0} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \times & x_{ij} & \times & \times & \times \\ \mathbf{\times} & \mathbf{0} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \end{bmatrix}$$

- Choosing different pivots means we can avoid zero or very small pivots
- Instead of using pivots at different entries, change rows or columns and use the standard triangular algorithm (*pivoting*)
- A computer code might account for the pivoting indirectly instead of actually moving the data

7

## Partial Pivoting

- Searching among all valid pivots is expensive (*complete pivoting*)
- Consider pivots in column  $k$  only and interchange rows (*partial pivoting*)

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \mathbf{x_{ik}} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{P_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ \mathbf{x_{ik}} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \times & \times & \times & \times & \times \\ \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \times & \times & \times & \times & \times \end{bmatrix} \xrightarrow{L_1} \begin{bmatrix} \times & \times & \times & \times & \times \\ x_{ik} & \times & \times & \times & \times \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \\ \mathbf{0} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} & \mathbf{\times} \end{bmatrix}$$

Pivot selection                      Row interchange                      Elimination

- In terms of matrices:

$$L_{m-1}P_{m-1} \cdots L_2P_2L_1P_1A = U$$

8

## The $PA = LU$ Factorization

- To combine all  $L_k$  and all  $P_k$  into matrices, rewrite as

$$L_{m-1}P_{m-1} \cdots L_2P_2L_1P_1A = U$$

$$(L'_{m-1} \cdots L'_2L'_1)(P_{m-1} \cdots P_2P_1)A = U$$

where

$$L'_k = P_{m-1} \cdots P_{k+1}L_kP_{k+1}^{-1} \cdots P_{m-1}^{-1}$$

- This gives the LU factorization of  $A$

$$PA = LU$$

9

## Gaussian Elimination with Partial Pivoting

- Factorize  $A \in \mathbb{C}^{m \times m}$  into  $PA = LU$ :

### Algorithm: Gaussian Elimination (partial pivoting)

$$U = A, L = I, P = I$$

for  $k = 1$  to  $m - 1$

    Select  $i \geq k$  to maximize  $|u_{ik}|$

$u_{k,k:m} \leftrightarrow u_{i,k:m}$  (interchange two rows)

$\ell_{k,1:k-1} \leftrightarrow \ell_{i,1:k-1}$

$p_{k,:} \leftrightarrow p_{i,:}$

    for  $j = k + 1$  to  $m$

$\ell_{jk} = u_{jk}/u_{kk}$

$u_{j,k:m} = u_{j,k:m} - \ell_{jk}u_{k,k:m}$

10

## Complete Pivoting

- If pivots are selected from a different column, permutation matrices  $Q_k$  for the columns are required:

$$L_{m-1}P_{m-1} \cdots L_2P_2L_1P_1AQ_1Q_2 \cdots Q_{m-1} = U$$

$$(L'_{m-1} \cdots L'_2L'_1)(P_{m-1} \cdots P_2P_1)A(Q_1Q_2 \cdots Q_{m-1}) = U$$

- Set

$$L = (L'_{m-1} \cdots L'_2L'_1)^{-1}$$

$$P = P_{m-1} \cdots P_2P_1$$

$$Q = Q_1Q_2 \cdots Q_{m-1}$$

to obtain

$$PAQ = LU$$

11