Lecture 12
Gaussian Elimination and LU Factorization

Songting Luo

Department of Mathematics
Iowa State University

MATH 562 Numerical Analysis II

Outline

Outline

Gaussian Elimination and LU Factorization

- Gaussian elimination can be viewed as "triangular triangularization" of nonsingular $\mathbf{A} \in \mathbb{C}m \times m$

$$\underbrace{\mathbf{L}_{m-1} \cdots \mathbf{L}_2 \mathbf{L}_1}_{\mathbf{L}^{-1}} \mathbf{A} = \mathbf{U}$$

analogous to Householder QR factorization of matrix $\mathbf{A} \in \mathbb{C}m \times m$

$$\underbrace{\mathbf{Q}_n \cdots \mathbf{Q}_2 \mathbf{Q}_1}_{\mathbf{Q}^*} \mathbf{A} = \mathbf{R}$$

- Example of LU factorization of 4×4 matrix $\mathbf{A}$

$$\xrightarrow{\mathbf{L}_1} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}}_{\mathbf{L}_1\mathbf{A}} \xrightarrow{\mathbf{L}_2} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}}_{\mathbf{L}_2\mathbf{L}_1\mathbf{A}} \xrightarrow{\mathbf{L}_3} \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & 0 & \times \end{bmatrix}}_{\mathbf{L}_3\mathbf{L}_2\mathbf{L}_1\mathbf{A}}$$

What is Matrices $\mathbf{L}_k$

- At step k, eliminate entries below a_{kk}: let $\mathbf{x}_k$ be kth column of $\mathbf{L}_{k-1} \cdots \mathbf{L}_1 \mathbf{A}$,
  $\mathbf{x}_k = [x_{1,k}, x_{2,k}, \ldots, x_{k,k}, x_{k+1,k}, \ldots, x_{m,k}]^T$
  $\mathbf{L}_k \mathbf{x}_k = [x_{1,k}, x_{2,k}, \ldots, x_{k,k}, 0, \cdots, 0]^T$
- The multiplier $l_{jk} = x_{jk}/x_{kk}$ appear in $\mathbf{L}_k$

$$\mathbf{L}_k = \begin{bmatrix} 1 & & & & & \\ & \ddots & & & & \\ & & 1 & & & \\ & & -l_{k+1,k} & 1 & & \\ & & \vdots & & \ddots & \\ & & -l_{m,k} & & & 1 \end{bmatrix}$$

- Let $\mathbf{l}_k = [0, \cdots, 0, l_{k+1,k}, \cdots, l_{m,k}]^T$ and $\mathbf{e}_k = [0, \cdots, 0, 1, \cdots, 0]^T$, then $\mathbf{L}_k = \mathbf{I} - \mathbf{l}_k \mathbf{e}_k^*$.

Forming $\mathbf{L}$

- Luckily, the $\mathbf{L}$ matrix contains the multipliers $l_{jk} = x_{jk}/x_{kk}$

$$\mathbf{L} = \mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\cdots\mathbf{L}_{m-1}^{-1} = \begin{bmatrix} 1 & & & & \\ l_{21} & 1 & & & \\ l_{31} & l_{32} & 1 & & \\ \vdots & \vdots & \ddots & \ddots & \\ l_{m1} & l_{m2} & \cdots & l_{m,m-1} & 1 \end{bmatrix}$$

  and is said to be a unit lower triangular matrix
- First, $\mathbf{L}_k^{-1} = \mathbf{I} + \mathbf{l}_k\mathbf{e}_k^*$, because $\mathbf{e}_k^*\mathbf{l}_k = 0$ and $(\mathbf{I} - \mathbf{l}_k\mathbf{e}_k^*)(\mathbf{I} + \mathbf{l}_k\mathbf{e}_k^*) = \mathbf{I} - \mathbf{l}_k\mathbf{e}_k^*\mathbf{l}_k\mathbf{e}_k^* = \mathbf{I}$
- Second, $\mathbf{L}_1^{-1}\mathbf{L}_2^{-1}\cdots\mathbf{L}_{k+1}^{-1} = \mathbf{I} + \sum_{j=1}^{k+1}\mathbf{l}_j\mathbf{e}_j^*$, since (prove by induction) $(\mathbf{I} + \sum_{j=1}^{k}\mathbf{l}_j\mathbf{e}_j^*)(\mathbf{I} + \mathbf{l}_{k+1}\mathbf{e}_{k+1}^*) = \mathbf{I} + \sum_{j=1}^{k+1}\mathbf{l}_j\mathbf{e}_j^* + \sum_{j=1}^{k}\mathbf{l}_j(\mathbf{e}_j^*\mathbf{l}_{k+1})\mathbf{e}_{k+1}^*$ where $\mathbf{e}_j^*\mathbf{l}_{k+1} = 0$ for $j < k+1$
- In other words, $\mathbf{L}$ is "union" of $\mathbf{L}_1^{-1}, \mathbf{L}_2^{-1}, \cdots, \mathbf{L}_{m-1}^{-1}$

Gaussian Elimination without Pivoting

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

Gaussian elimination without pivoting

$\mathbf{U} = \mathbf{A}, \mathbf{L} = \mathbf{I}$
for $k = 1$ to $m - 1$
  for $j = k + 1$ to m
    $l_{jk} = u_{jk}/u_{kk}$
    $u_{j,k:m} = u_{j,k:m} - l_{jk}u_{k,k:m}$

- Flop count $\sim \sum_{k=1}^{m} 2(m-k)(m-k) \sim 2\sum_{k=1}^{m} k^2 \sim 2m^3/3$
- In practice, $\mathbf{L}$ often overwrites lower-triangular part of $\mathbf{A}$ and $\mathbf{U}$ overwrites upper-triangular part of $\mathbf{A}$
- Question: What if u_{kk} is 0? Answer: The algorithm would break.

Partial Pivoting

- At step k, we divide by u_{kk} , which would break if u_{kk} is 0 (or close to 0), which can happen even if **A** is nonsingular

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

- However, any nonzero entry in kth column below diagonal can also be used as pivot (In general, we take nonzero entry with largest absolute value)

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

More on Partial Pivoting

- kth step of Gaussian elimination of partial pivoting

$$\underset{\text{Pivot selection}}{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & x_{ik} & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}} \xrightarrow{\mathbf{P}_k} \underset{\text{Row Interchange}}{\begin{bmatrix} \times & \times & \times & \times \\ 0 & x_{kk} & * & * \\ 0 & * & * & * \\ 0 & \times & \times & \times \end{bmatrix}} \xrightarrow{\mathbf{L}_k} \underset{\text{Elimination}}{\begin{bmatrix} \times & \times & \times & \times \\ 0 & x_{kk} & \times & \times \\ 0 & 0 & * & * \\ 0 & 0 & * & * \end{bmatrix}}$$

  and we interchange row i with row k
- In terms of matrices, it becomes $\underbrace{\mathbf{L}_{m-1}\mathbf{P}_{m-1}\cdots\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1\mathbf{A}}_{\mathbf{L}^{-1}\mathbf{P}} = \mathbf{U}$
- $\mathbf{P} = \mathbf{P}_{m-1}\cdots\mathbf{P}_2\mathbf{P}_1$ and $\mathbf{L} = (\mathbf{L}'_{m-1}\cdots\mathbf{L}'_2\mathbf{L}'_1)^{-1}$, where $\mathbf{L}'_k = \mathbf{P}_{m-1}\cdots\mathbf{P}_{k+1}\mathbf{L}_k\mathbf{P}^{-1}_{k+1}\cdots\mathbf{P}^{-1}_{m-1}$
- It is easy to verify that $\mathbf{L}_{m-1}\mathbf{P}_{m-1}\cdots\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1 = (\mathbf{L}'_{m-1}\cdots\mathbf{L}'_2\mathbf{L}'_1)(\mathbf{P}_{m-1}\cdots\mathbf{P}_2\mathbf{P}_1)$
- $\mathbf{L}'_k = \mathbf{I} - \mathbf{P}_{m-1}\cdots\mathbf{P}_{k+1}\mathbf{l}_k\mathbf{e}^*_k$ and $\mathbf{L}$ is "union" of $(\mathbf{L}'_k)^{-1} = \mathbf{I} + \mathbf{P}_{m-1}\cdots\mathbf{P}_{k+1}\mathbf{l}_k\mathbf{e}^*_k$

Algorithm of Gaussian Elimination with Partial Pivoting

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

Gaussian elimination with partial pivoting

$\mathbf{U} = \mathbf{A}, \mathbf{L} = \mathbf{I}, \mathbf{P} = \mathbf{I}$
for $k = 1$ to $m - 1$
  $i \leftarrow argmax_{i \geqslant k} |u_{ij}|$
  $\mathbf{u}_{k,k:m} \leftrightarrow \mathbf{u}_{i,k:m}$
  $\mathbf{l}_{k,1:k-1} \leftrightarrow \mathbf{l}_{i,1:k-1}$
  $\mathbf{p}_{k,:} \leftrightarrow \mathbf{p}_{i,:}$
  for $j = k + 1$ to m
    $l_{jk} = u_{jk}/u_{kk}$
    $u_{j,k:m} = u_{j,k:m} - l_{jk} u_{k,k:m}$

- Question: What if u_{kk} is 0?
- Flot count $\sim \sum_{k=1}^{m} 2(m-k)(m-k) \sim 2 \sum_{k=1}^{m} k^2 \sim 2m^3/3$, same as without pivoting.

An Alternative Implementation

- In practice, **L** and **U** overwrite **A** and **P** is represented by a vector

Gaussian elimination with partial pivoting (alternative)

$\mathbf{p} = [1, 2, \cdots, m]$;
for $k = 1$ to $m - 1$
  $i \leftarrow argmax_{i \geqslant k} |u_{ij}|$
  $\mathbf{a}_{k,1:m} \leftrightarrow \mathbf{a}_{i,1:m}$
  $p_k \leftrightarrow p_i$
  $\mathbf{a}_{k+1:m,k} \leftarrow \mathbf{a}_{k+1:m,k} / a_{k,k}$
  $\mathbf{A}_{k+1:m,k+1:m} \leftarrow \mathbf{A}_{k+1:m,k+1:m} - \mathbf{a}_{k+1:m,k} \times \mathbf{a}_{k,k+1:m}$

- Using LU factorization to solve $\mathbf{Ax} = \mathbf{b}$:
  - $\mathbf{PA} = \mathbf{LU}$ (LU factorization with partial pivoting)
  - $\mathbf{Ly} = \mathbf{Pb}$ (Forward substitution)
  - $\mathbf{Ux} = \mathbf{y}$ (Back substitution)

Complete Pivoting

- More generally, we can use any nonzero entry
- In theory, any nonzero entry $(i, j), i \geqslant k, j \geqslant k$

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

  and we then permute row i with row k, column j with column k
- In matrix operations, it can be expressed as
  $\underbrace{\mathbf{L}_{m-1}\mathbf{P}_{m-1}\cdots\mathbf{L}_2\mathbf{P}_2\mathbf{L}_1\mathbf{P}_1}_{\mathbf{L}^{-1}\mathbf{P}}\,\mathbf{A}\,\underbrace{\mathbf{Q}_1\mathbf{Q}_2\cdots\mathbf{Q}_{m-1}}_{\mathbf{Q}} = \mathbf{U}$
- Therefore, $\mathbf{PAQ} = \mathbf{LU}$ where $\mathbf{P} = \mathbf{P}_{m-1}\cdots\mathbf{P}_2\mathbf{P}_1$ and $\mathbf{L} = (\mathbf{L}'_{m-1}\cdots\mathbf{L}'_2\mathbf{L}'_1)^{-1}$
- However, complete pivoting is typically not used in practice because it increases cost in search of pivot and complexity of implementation