Parallel programming

HW1 assignment

FAKULTA
ELEKTROTECHNICKÁ
ČVUT V PRAZE

LU decomposition

- A decomposition of matrix **A** to lower and upper triangular matrices **L** and **U**, respectively.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

$$A = LU$$

- **Application:**
  - Useful to quickly resolve a linear system of equation if only the right-hand side is changed (Ax = b).

LU decomposition

- LU decomposition is basically Gaussian elimination (GE)

$$E_3 E_2 E_1 A = U$$

E_i represent one step of GE, e.g., subtraction of a row multiple from another row

- Example of E_i matrix and its inverse (subtract 2 times the first row from the third row)

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -2 & 0 & 1 \end{pmatrix}}_{E_i} \longrightarrow \underbrace{\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix}}_{E_i^{-1}}$$

- Inverting E_i matrices we get L

$$A = \underbrace{E_1^{-1} E_2^{-1} E_3^{-1}}_{L} U$$

LU decomposition

- How does L look like?

$$L = E_1^{-1} E_2^{-1} E_3^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 2 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 4 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 4 & 1 \end{pmatrix}$$

- How can we use LU decomposition to solve Ax = b?

$$A x = b$$

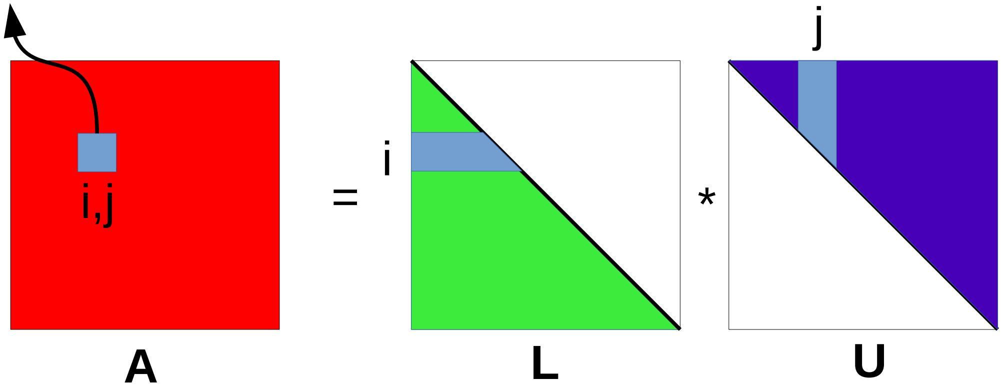
$$L \underbrace{U x}_{y} = b$$

$$L y = b$$ Solve by forward substitution

$$U x = y$$ Solve by backward substitution

LU decomposition - derivation

$$a_{ij}=\sum_{r=0}^{n-1} l_{ir} u_{rj} \qquad \forall i,j \in \{0,\dots,n-1\}$$



$$l_{ir}=\begin{cases} 0 \text{ if } r>i \\ 1 \text{ if } r=i \\ l_{ir} \text{ otherwise} \end{cases}$$

$$u_{rj}=\begin{cases} 0 \text{ if } r>j \\ u_{rj} \text{ otherwise} \end{cases}$$

$$a_{ij}=\sum_{r=0}^{n-1} l_{ir} u_{rj}=\sum_{r=0}^{i} l_{ir} u_{rj}=\sum_{r=0}^{i-1} l_{ir} u_{rj}+l_{ii} u_{ij}=\sum_{r=0}^{i-1} l_{ir} u_{rj}+1\, u_{ij}$$

$$\Rightarrow \quad u_{ij}=a_{ij}-\sum_{r=0}^{i-1} l_{ir} u_{rj}$$

$$a_{ij}=\sum_{r=0}^{n-1} l_{ir} u_{rj}=\sum_{r=0}^{j} l_{ir} u_{rj}=\sum_{r=0}^{j-1} l_{ir} u_{rj}+l_{ij} u_{jj}$$

$$\Rightarrow \quad l_{ij}=\frac{1}{u_{jj}}\left(a_{ij}-\sum_{r=0}^{j-1} l_{ir} u_{rj}\right)$$

LU decomposition - derivation

$$a_{ij}^{(k)} = a_{ij}^{(k-1)} - \sum_{r=0}^{k-1} l_{ir} u_{rj} \quad \forall k \in \{0, \ldots, n-1\}$$

iteration

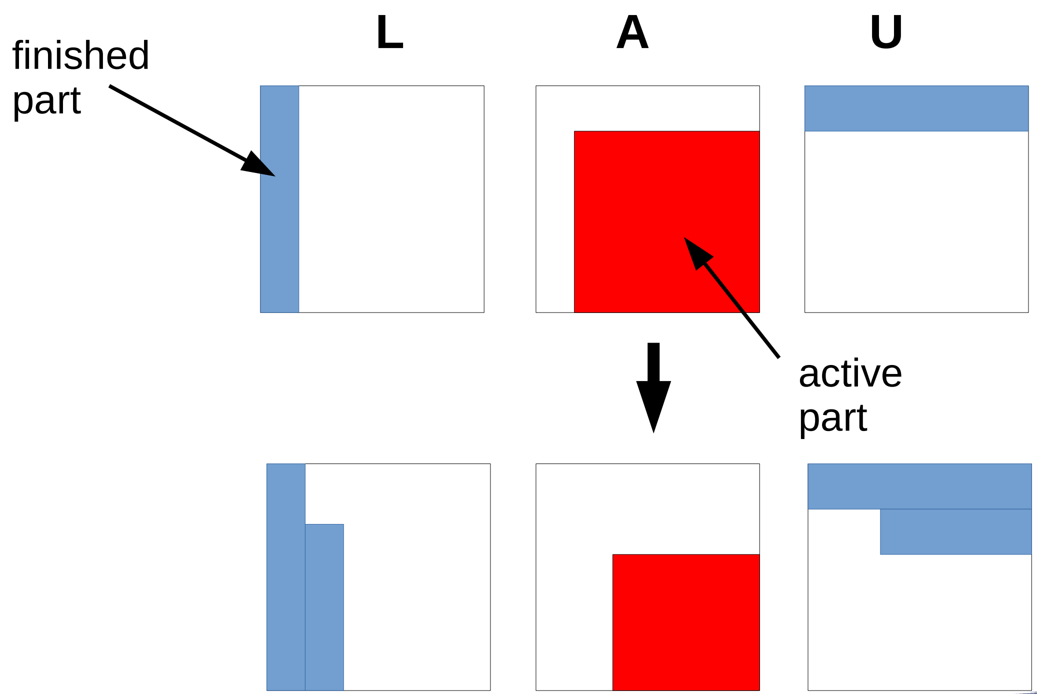
$$l_{ij} = \frac{1}{u_{jj}} \left(a_{ij} - \sum_{r=0}^{j-1} l_{ir} u_{rj} \right)$$

$$u_{ij} = a_{ij} - \sum_{r=0}^{i-1} l_{ir} u_{rj}$$

$$\Rightarrow \quad u_{ij} = a_{ij}^{(i)} \qquad l_{ij} = \frac{a_{ij}^{(j)}}{u_{jj}}$$

Pseudocode:

for k = 0 **to** n - 1 **do**
    for j = k **to** n - 1 **do**
        $u_{kj} = a_{kj}^{(k)}$
    $l_{kk} = 1$
    for i = k + 1 **to** n - 1 **do**
        $l_{ik} = a_{ik}^{(k)} / u_{kk}$
    for i = k + 1 **to** n - 1 **do**
        for j = k + 1 **to** n - 1 **do**
            $a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} u_{kj}$



HW1 assignment

- Use the provided **code skeleton**
  - reads test problems, measures runtime, prints the matrices **L**, **U**, and **A**.
- **Assignment:**
  - implement LU decomposition
  - parallelize the code using C++11 threads
  - upload your solution to UploadSystem
    - What should I upload?
- **Flags for g++ (used by UploadSystem):**
  - `-pthread -Ofast -std=c++17 -march=native`