## MAT 460 Numerical Differential Equations

Version 21.10.14

Michael Karow, Andrea Dziubek

## Lecture 2

*Topics: Linear Equation Systems, $LU$-factorization*



# Motivation:

**Physical problem:**
Prediction and/or optimization of a process.

$\rightsquigarrow$

**Mathematical model:**
Ordinary or partial differential equations
(infnitely many unknowns)

$\rightsquigarrow$

**Computational problem:**
Solution of the differential equations

$\rightsquigarrow$

approximation of the differential euqations by
equations with finitely many unknowns

$\rightsquigarrow$

**Computational problem:**
Solution of equations

$\rightsquigarrow$

**Computational problem:**
Solution of linear equation systems

| |
|---|
| *Solving linear equation systems is a fundamental task in scientific computing* |



Linear Equation System (LES):

$$\begin{aligned} 2\,x_1 + \ \ 3\,x_2 + \ \ 5\,x_3 &= 11 \\ 6\,x_1 + 12\,x_2 + 22\,x_3 &= 34 \\ 4\,x_1 + 12\,x_2 + 20\,x_3 &= 32 \end{aligned}$$

Write as matrix equation:

$$\underbrace{\begin{bmatrix} 2 & 3 & 5 \\ 6 & 12 & 22 \\ 4 & 12 & 20 \end{bmatrix}}_{A} \underbrace{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}}_{x} = \underbrace{\begin{bmatrix} 11 \\ 34 \\ 32 \end{bmatrix}}_{b}$$

Find: $x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$

**An LES in upper triangular form is solved by back substitution.**

Example:

$$\begin{bmatrix} 2 & 3 & 5 \\ 0 & 3 & 7 \\ 0 & 0 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 1 \\ 8 \end{bmatrix}$$

$\Leftrightarrow$

$$\begin{aligned} 2\,x_1 + 3\,x_2 + 5\,x_3 &= 11 \\ 3\,x_2 + 7\,x_3 &= 1 \\ -4\,x_3 &= 8 \end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned} x_3 &= 8/(-4) \\ x_2 &= (1/3)(1 - 7\,x_3) \\ x_1 &= (1/2)(11 - 5\,x_3 - 3\,x_2) \end{aligned}$$

$\Leftrightarrow$

$$x_3 = -2, \ \ x_2 = 5, \ \ x_1 = 3$$

## An LES in lower triangular form is solved by forward substitution.

Example:

$$\begin{bmatrix}1&0&0\\3&1&0\\2&2&1\end{bmatrix}\begin{bmatrix}y_1\\y_2\\y_3\end{bmatrix}=\begin{bmatrix}11\\34\\32\end{bmatrix}$$

$\Leftrightarrow$

$$\begin{aligned}y_1 &= 11\\ 3\,y_1+y_2 &= 34\\ 2\,y_1+2\,y_2+y_3 &= 32\end{aligned}$$

$\Leftrightarrow$

$$\begin{aligned}y_1 &= 11\\ y_2 &= 34-3\,y_1\\ y_3 &= 32-2\,y_1-2\,y_1\end{aligned}$$

$\Leftrightarrow$

$$y_1=11,\quad y_2=1,\quad y_3=8$$



Almost every square matrix $A\in\mathbb{R}^{n\times n}$ can be factorized as:

$$A=LU$$

Here

$L\in\mathbb{R}^{n\times n}$ is a lower triangular matrix with ones on the diagonal

$U\in\mathbb{R}^{n\times n}$ is a upper triangular matrix

Example:

$$\underbrace{\begin{bmatrix}2&3&5\\6&12&22\\4&12&20\end{bmatrix}}_{A}=\underbrace{\begin{bmatrix}1&0&0\\3&1&0\\2&2&1\end{bmatrix}}_{L}\underbrace{\begin{bmatrix}2&3&5\\0&3&7\\0&0&-4\end{bmatrix}}_{U}$$



## Solution of linear equation systems with LU-factorization.

Let $A=LU$. Then we solve the equation $Ax=b$ in two steps.

**Step 1:** Solve $Ly=b$ (by forward substitution).

**Step 2:** Solve $Ux=y$ (by backward substitution).

$\Rightarrow$

$x$ is the solution of $Ax=b$

**'Proof':**
From $A=LU$, $Ly=b$ and $Ux=y$ it follows that $Ax=LUx=Ly=b$.

**Problem:** How do we find the $LU$-factorization?

We will show how to:

Find an $LU$-factorization using the **Gauss-algorithm**.

*The basic idea of the Gauss-algorithm is:*

adding (subtracting) the multiple of an equation to one of the other equations in a linear equation system does not change the solution of the linear equation system.

We want to add (subtract) rows so that we get zeros below the diagnoal:

$$\left[\underbrace{\begin{matrix}*&*&*&*\\*&*&*&*\\*&*&*&*\\*&*&*&*\end{matrix}}_{A}\;\middle|\;\underbrace{\begin{matrix}*\\*\\*\\*\end{matrix}}_{b}\right]\rightsquigarrow\left[\begin{matrix}*&*&*&*\\0&*&*&*\\0&*&*&*\\0&*&*&*\end{matrix}\;\middle|\;\begin{matrix}*\\*\\*\\*\end{matrix}\right]\rightsquigarrow\left[\begin{matrix}*&*&*&*\\0&*&*&*\\0&0&*&*\\0&0&*&*\end{matrix}\;\middle|\;\begin{matrix}*\\*\\*\\*\end{matrix}\right]\rightsquigarrow\left[\begin{matrix}*&*&*&*\\0&*&*&*\\0&0&*&*\\0&0&0&*\end{matrix}\;\middle|\;\begin{matrix}*\\*\\*\\*\end{matrix}\right]$$

Now we can solve the LES by backward substitution.

## Gauss-algorithm (without interchanging rows)

Solve: $\begin{bmatrix} 2 & 3 & 5 \\ 6 & 12 & 22 \\ 4 & 12 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 34 \\ 32 \end{bmatrix}$

Augmented matrix:

$$\left[\begin{array}{ccc:c} 2 & 3 & 5 & 11 \\ 6 & 12 & 22 & 34 \\ 4 & 12 & 20 & 32 \end{array}\right]$$

$$\begin{array}{rcl} \text{2.row} + 3*(\text{1.row}) & \Big\uparrow\Big\downarrow & \text{2.row - } 3*(\text{1.row}) \\ \text{3.row} + 2*(\text{1.row}) & & \text{3.row - } 2*(\text{1.row}) \end{array}$$

$$\left[\begin{array}{ccc:c} 2 & 3 & 5 & 11 \\ 0 & 3 & 7 & 1 \\ 0 & 6 & 10 & 10 \end{array}\right]$$

$$\begin{array}{rcl} \text{3.row} + 2*(\text{2.row}) & \Big\uparrow\Big\downarrow & \text{3.row - } 2*(\text{2.row}) \end{array}$$

$$\left[\begin{array}{ccc:c} 2 & 3 & 5 & 11 \\ 0 & 3 & 7 & 1 \\ 0 & 0 & -4 & 8 \end{array}\right] \quad \text{triangular form} \rightarrow \text{solve by back substitution}$$



## Row operations are performed by matrix multiplication from left.

Example: The transformation

$$\begin{bmatrix} 2 & 3 & 5 \\ 6 & 12 & 22 \\ 4 & 12 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 34 \\ 32 \end{bmatrix} \xrightarrow[\text{3.row - } 2*(\text{1.row})]{\text{2.row - } 3*(\text{1.row})} \begin{bmatrix} 2 & 3 & 5 \\ 0 & 3 & 7 \\ 0 & 6 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 11 \\ 1 \\ 10 \end{bmatrix}$$

corresponds to a multiplication with the matrix

$$L_1 = \begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}$$

Since

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}}_{L_1} \begin{bmatrix} 2 & 3 & 5 \\ 6 & 12 & 22 \\ 4 & 12 & 20 \end{bmatrix} = \begin{bmatrix} 2 & 3 & 5 \\ 0 & 3 & 7 \\ 0 & 6 & 10 \end{bmatrix} \quad \text{and} \quad \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ -3 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}}_{L_1} \begin{bmatrix} 11 \\ 34 \\ 32 \end{bmatrix} = \begin{bmatrix} 11 \\ 1 \\ 10 \end{bmatrix}.$$

The reverse of the row operation above corresponds to a multiplication with the inverse of $L_1$, i. e.

$$L_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ 2 & 0 & 1 \end{bmatrix}.$$

We have: $\quad Ax = b \quad \Leftrightarrow \quad L_1 A x = L_1 b$



**Note: Row operations in the Gauss-algorithm are performed by left-multiplication of the equation $Ax = b$ with matrices of the form**

$$L_k = \begin{bmatrix} 1 & 0 & \dots & & \dots & 0 \\ 0 & \ddots & & & & \\ & & 1 & & & \\ & & -\ell_{k,k+1} & 1 & & \\ & & \vdots & & \ddots & 0 \\ 0 & & -\ell_{k,n} & & & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

$\uparrow$ $k$-th row

Such a matrix is called a **Frobenius-matrix** of type $k$.

If we multiply $A$ from left by $L_k$,
we perform the following row operations on $A$:

Subtract from the $(k+1)$-th row the $\ell_{k,k+1}$-multiple of the $k$-th row.
Subtract from the $(k+2)$-th row the $\ell_{k,k+2}$-multiple of the $k$-th row.
$\vdots$
Subtract from the $n$-th row the $\ell_{k,n}$-multiple of the $k$-th row.

**Proof:** Check

**Note: We compute the inverse of a Frobenius-matrix by changing the signs below the diagnoal:**

$$L_k = \begin{bmatrix} 1 & 0 & \dots & & \dots & 0 \\ 0 & \ddots & & & & \\ & & 1 & & & \\ & & -\ell_{k,k+1} & 1 & & \\ & & \vdots & & \ddots & 0 \\ 0 & & -\ell_{k,n} & & & 1 \end{bmatrix} \Rightarrow L_k^{-1} = \begin{bmatrix} 1 & 0 & \dots & & \dots & 0 \\ 0 & \ddots & & & & \\ & & 1 & & & \\ & & \ell_{k,k+1} & 1 & & \\ & & \vdots & & \ddots & 0 \\ 0 & & \ell_{k,n} & & & 1 \end{bmatrix}$$

**Proof:** Check that $L_k L_k^{-1} = I \quad$ (I = identity matrix).

We write the Gauss-algorithm in matrix notation as:

$$\begin{aligned} Ax = b \quad & \Leftrightarrow \quad L_1\, A\, x = L_1\, b \\ & \Leftrightarrow \quad L_2 L_1\, A\, x = L_2 L_1\, b \\ & \vdots \\ & \Leftrightarrow \quad \underbrace{L_{n-1} L_{n-2} \dots L_1\, A}_{U}\, x = L_{n-1} L_{n-2} \dots L_1\, b \end{aligned}$$

with appropriate Frobenius-matrices $L_k$.

The matrix $U = L_{n-1} L_{n-2} \dots L_1\, A$ is in upper triangular form. We have

$$\begin{aligned} \underbrace{L_1^{-1} \dots L_{n-2}^{-1} L_{n-1}^{-1}}_{=:L} U &= L_1^{-1} \dots L_{n-2}^{-1} L_{n-1}^{-1}\, L_{n-1} L_{n-2} \dots L_1 A \\ &= A \end{aligned}$$

By this we found the desired $LU$-factorization of $A$, since (check)

$$L = \begin{bmatrix} 1 & 0 & \dots & & \dots & 0 \\ \ell_{12} & \ddots & & & & \\ & \ddots & 1 & & & \\ \vdots & & \ell_{k,k+1} & \ddots & & \vdots \\ & & & \ddots & 1 & 0 \\ \ell_{1n} & & \dots & & \ell_{n,n-1} & 1 \end{bmatrix}.$$

The entry $\ell_{k,j}$ is the factor (in the Gauss-algorithm) that we multiply the $k$-th row with, before subtracting this row from the $j$-th row.



## Computing time for the LU-factorization

To replace in a $n \times n$-matrix the entries below the diagonal with zeros using the Gauss-algorithm we need $(n-1) + (n-1)^2$ multiplikations/divisions:

$$\underbrace{\begin{bmatrix} * & * & \cdots & * \\ * & * & \cdots & * \\ \vdots & \vdots & \cdots & \vdots \\ * & * & \cdots & * \end{bmatrix}}_{n} \rightsquigarrow \begin{bmatrix} * & * & \cdots & * \\ 0 & * & \cdots & * \\ \vdots & \vdots & \cdots & \vdots \\ 0 & * & \cdots & * \end{bmatrix}$$

Let $M_n$ be the number of the multiplikations/divisions we need to compute the $LU$-factorization of an $n \times n$ matrix. Then

$$\begin{aligned} M_n &= (n-1) + (n-1)^2 + M_{n-1} \\ &= (n-1) + (n-1)^2 + (n-2) + (n-2)^2 + M_{n-2} \end{aligned}$$

From this it follows:

$$\begin{aligned} M_n &= \sum_{k=1}^{n-1} k + \sum_{k=1}^{n-1} k^2 \\ &= \frac{n(n-1)}{2} + \frac{n(n-1)(2n-1)}{6} \\ &= \frac{1}{3} n^3 + \text{smaller powers of } n = \mathcal{O}(n^3). \end{aligned}$$

Examples: $M_{25} = 5200$, $M_{50} = 41650$, $M_{100} = 333300$.

**Rule:** If we double the dimension of a matrix the computation time is eight times larger.



## $PLU$-factorization:

Sometimes we need to interchange rows or columns in the Gauss-algorithm to avoid division by zero or by a very small number (division by small numbers result in numerical errors).

**Theorem:** Every intertable matrix $A$ can be written as a product,

$$A = PLU,$$

where
$L$ is a lower triangular matrix,
$U$ is a upper triangular matrix
and $P$ is a permutaion matrix.

*Explanation:*
A permutation matrix is a square matrix with exactly one 1 in each row and in each column. All other entries are 0.

Left-multiplication with a permutation matrix interchanges rows.

## Other factorization to solve linear equation systems:

- **$QR$-factorization:**
  Every (including non square) matrices $A$ can be factorized into
  $$A = QR,$$
  where $R$ is an upper triangular matrix and $Q$ is an orthogonal matrix, i. e. $Q^{-1} = Q^T$.

- **Cholesky-factorization**
  Every positive definite symmetric matrix $A$ can be factorized into
  $$A = LL^T,$$
  where $L$ is a lower triangular matrix.

$\Rightarrow$ We will learn more about these factorizations in the next lectures.

## Linear equation systems and determinants

**Cramers rule:**
Let $A = [a_1, \ldots, a_n]$ be an invertible square matrix with the columns $a_1, \ldots, a_n$. Then the solution of the equation $Ax = b$, $x = [x_1, \ldots, x_n]^T$ is as follows:

$$x_k = \frac{\det(a_1, \ldots, a_{k-1}, b, a_{k+1}, \ldots, a_n)}{\det(A)}.$$

**Note:**
Cramers rule is **not used** to solve large linear equation systems.

Reason: To compute the determinant is computationally expensive.



**Definition of a determinant:**

$$\det(A) = \sum_{\sigma \in S(n)} \text{sign}(\sigma)\, a_{1\sigma(1)}\, a_{2\sigma(2)} \ldots a_{n\sigma(n)}.$$

Here $S(n)$ is the set of permutations of the numbers $1, \ldots, n$.

Example: $n = 4$

$$\det(A) \;=\; \begin{array}{l} +a_{11}\,a_{22}\,a_{33}\,a_{44} \\ -a_{11}\,a_{22}\,a_{34}\,a_{43} \\ -a_{11}\,a_{23}\,a_{32}\,a_{44} \\ +a_{11}\,a_{23}\,a_{34}\,a_{42} \\ +a_{11}\,a_{24}\,a_{32}\,a_{43} \\ -a_{11}\,a_{24}\,a_{33}\,a_{42} \\ -a_{12}\,a_{21}\,a_{33}\,a_{44} \\ +a_{12}\,a_{21}\,a_{34}\,a_{43} \\ +a_{12}\,a_{23}\,a_{31}\,a_{44} \\ -a_{12}\,a_{23}\,a_{34}\,a_{41} \\ -a_{12}\,a_{24}\,a_{31}\,a_{43} \\ +a_{12}\,a_{24}\,a_{33}\,a_{41} \end{array} + \begin{array}{l} +a_{13}\,a_{21}\,a_{32}\,a_{44} \\ -a_{13}\,a_{21}\,a_{34}\,a_{42} \\ -a_{13}\,a_{22}\,a_{31}\,a_{44} \\ +a_{13}\,a_{22}\,a_{34}\,a_{41} \\ +a_{13}\,a_{24}\,a_{31}\,a_{42} \\ -a_{13}\,a_{24}\,a_{32}\,a_{41} \\ -a_{14}\,a_{21}\,a_{32}\,a_{43} \\ +a_{14}\,a_{21}\,a_{33}\,a_{42} \\ +a_{14}\,a_{22}\,a_{31}\,a_{43} \\ -a_{14}\,a_{22}\,a_{33}\,a_{41} \\ -a_{14}\,a_{23}\,a_{31}\,a_{42} \\ +a_{14}\,a_{23}\,a_{32}\,a_{41} \end{array}$$

To compute the determinant of $A \in \mathbb{R}^{n \times n}$ with the formula above we need $n * n!$ multiplications. For larger $n$

$$n! \approx \sqrt{2\pi\, n} \left(\frac{n}{e}\right)^n \qquad \text{(Stirlingsche Formel).}$$

Für $n = 50$: $\quad n! = 3.0414 * 10^{64}, \qquad \sqrt{2\pi\, n} \left(\frac{n}{e}\right)^n = 3.0363 * 10^{64}.$



## Computation of a determinant with the LU-factorization

Facts about determinants:

1. *Multiplication theorem for determinants:*
   The determinant of a matrix product is the product of the determinants of the factors.
2. The determinant of a *triangular matrix* is the product of its diagaonal elements.

**Conclusion:**

$$A = LU \;\Rightarrow$$

$$\begin{aligned} \det(A) &= \det(LU) = \underbrace{\det(L)}_{=1} \det(U) = u_{11} u_{22} \ldots u_{nn} \\ &= \text{Product of the diagonal elements of } U. \end{aligned}$$

## Determinant criteria for the existence of a LU-factorization

A LU-factorization $A = LU$ for

$$A = \begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & & \vdots \\ a_{n1} & \ldots & a_{nn} \end{bmatrix}$$

exist and only if all determinants of the left upper sub-matrice

$$A_k = \begin{bmatrix} a_{11} & \ldots & a_{1k} \\ \vdots & & \vdots \\ a_{k1} & \ldots & a_{kk} \end{bmatrix}, \qquad k \le n - 1$$

are not zero.

## Existence of solutions of linear equations

A linear equation system does not always have a unique solution.

**Example:**

The equation $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ does not have a solution.

The equation $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ has infinitely many solutions,

i. e. $x_1 = 1$, $x_2$ arbitrary.



## Criteria for the existence of solutions of linear equation systems

**Theorem:** Let $A$ be an $n \times n$ matrix. The following conditions are equivalent:

1. The equation $Ax = b$ has exactly one solution for every vector $b$ exactly
2. The equation $Ax = 0$ has the only solution $x = 0$.
3. The columns of $A$ are linearly independent.
4. The rows of $A$ are linearly independent.
5. $A$ is invertible, i. e. the inverse matrix $A^{-1}$ exist.
6. $\det(A) \neq 0$

**Definition:** If one and by this all conditions are not satisfied then the matrix $A$ is called **singular**.

Let $A$ be singular. Then the linear equation system $Ax = b$ has either no solution, or infinitely many solutions (depending on $b$).

If $A$ is non singular (i. e. regular, invertible), then $x = A^{-1}b$ is the unique solution to $Ax = b$.



## How do we compute the inverse of a matrix?

Let $x_1, \ldots x_n$ the columns of the inverse of $A \in \mathbb{R}^{n \times n}$, i. e.

$$A^{-1} = [x_1 \; x_2 \; \ldots \; x_n].$$

With the definition of the inverse

$$A\,[x_1 \; x_2 \; \ldots \; x_n] = \begin{bmatrix} 1 & & \\ & \ddots & \\ & & 1 \end{bmatrix} = [e_1 \; e_2 \; \ldots \; e_n] \qquad (*)$$

where $e_k$ are the standard basis vectors. The equation $(*)$ summarizes the linear equations

$$Ax_1 = e_1, \quad Ax_2 = e_2, \quad \ldots Ax_n = e_n.$$

To find the inverse of an $n \times n$-matrix $n$ linear equations need to be solved.

**Note:** We rarely need to compute the inverse of a matrix. Computing the inverse of a matrix is numerically expensive and should be avoided. When we write (in an algorithm) the equation $x = A^{-1}b$ we usually mean we have to solve the equation system $Ax = b$ **without computing the inverse**.

## LU-factorization with matlab

LU-factorization

The matlab command to perform an LU-factorization is

$$[L, U] = lu(A)$$

MATLAB often returns L multiplied by a permutation matrix.
Typing `help lu` gives more information.

The MATLAB-command to solve $Ax = b$ is

$$x = A \backslash b$$

The MATLAB-command to find the inverse of $A$ is

$$inv(A) \qquad \text{or} \qquad A^{-1}$$

## $LU$-factorization with mit column pivoting (column interchange)



**Problem:**

The Gauss-algorithm in the form we learned it until now is not always practicable.

**Elementary Example:** Consider the linear equation system

$$\begin{bmatrix} 0 & 6 & 7 \\ 1 & 1 & 1 \\ 2 & 3 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

Because the upper left entry is zero we cannot generate zeros in the rows below by subtracting from the other rows a multiple of the first row.

**Remedy:**

Transpose two rows of the matrix so that the upper left element is not zero:

$$\begin{bmatrix} 2 & 3 & 4 \\ 1 & 1 & 1 \\ 0 & 6 & 7 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \qquad (*)$$

Then proceed as usual.

**Notation:** During a Gaussian elimination-step we divide the current row by a matrix element. This matrix element is called

**pivot element**.

Example: In $(*)$ the upper left 2 is the pivot element.



If the pivot element is small, even if it is possible to perform the Gauss-algorithm, we can get large numerical errors:

**Example:** Consider the linear equation system ($\epsilon > 0$)

$$\begin{aligned} \epsilon\, x_1 + x_2 &= 1 \\ x_1 + x_2 &= 2 \end{aligned}$$

The exact solution is

$$x_1 = \frac{1}{1-\epsilon}, \qquad x_2 = \frac{1-2\epsilon}{1-\epsilon}.$$

Performing the Gauss-algorithm we subtract from the 2nd row the $1/\epsilon$-multiple of the 1st row and we get

$$\begin{aligned} \epsilon\, x_1 + x_2 &= 1 \\ (1-(1/\epsilon))\, x_2 &= 2-(1/\epsilon). \end{aligned}$$

If $\epsilon$ is very small, then $1/\epsilon$ is very large, in the worst case we get from computing with machine numbers for the second row $-1/\epsilon$ instead of $(1-(1/\epsilon))$ and $(2-(1/\epsilon))$ resp., i. e. numerically we get

$$\begin{aligned} \epsilon\, x_1 + x_2 &= 1 \\ -(1/\epsilon)\, x_2 &= -(1/\epsilon). \end{aligned}$$

If we continue from here exact we get the solution

$$x_1 = 0, \qquad x_2 = 1.$$

$\Rightarrow$ huge error in $x_1$.

We can show that:

Large pivot elements give small numerical errors.

Small pivot elements give large numerical errors.

**Conclude:** Before each elimination-step in the Gauss-algorithm find the (absolute) greatest element in the current column (below or on the diagonal),

and transpose the corresponding row with the current row so that this element becomes the pivot element.

This operation is called column pivoting.

**Schema:**

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \otimes & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \otimes & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \\ 0 & \times & \times & \times & \times \end{bmatrix} \to \begin{bmatrix} \times & \times & \times & \times & \times \\ 0 & \otimes & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \\ 0 & 0 & \times & \times & \times \end{bmatrix}$$

Start find pivot element transpose rows continue

In this example the 2nd column is the current column.

The (absolute) greatest element in the 2nd column is marked by $\otimes$.

**Remark:** (Additionally) we could perform column transpositions (row pivoting), however in practice row pivoting is usually enough.

**The fundamental theorem of LU-factorization**

With suitable row transpositions we can always perform an $LU$-factorization of an invertible matrix $A$.
In mathematical language we say:

> **Theorem:** To every invertible matrix $A \in \mathbb{R}^{n\times n}$ exist a permutation matrix $P \in \mathbb{R}^{n\times n}$, such that
> $$PA = LU.$$
> Here
> $L$ is an lower triangular matrix with 1s on the diagonaland
> $U$ is an upper triangular matrix. With suitable choice of $P$ (column pivoting) all elements of $L$ have an absolute value $\leq 1$.

**Explanation:**

A permutation matrix is a square matrix, which has in each row and in each column exactly one 1. All other entries are 0.

Left multiplication with a permutation matrix performs row transpositions.
(see next slide)

**Aim of this lecture:** Understand the theorem and its applications.



## What does a permutation matrix do?

Multiplying a permutation matrix $P \in \mathbb{R}^{m\times m}$ from left with a matrix $A \in \mathbb{R}^{m\times n}$ interchanges rows of $A$.

**Example:**

$$\underbrace{\begin{bmatrix} 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}_{PA}$$

The 1 in the **1**st row is in column 3 $\Rightarrow$ 3rd row becomes **1**st row
The 1 in the **2**nd row is in column 1 $\Rightarrow$ 1st row becomes **2**nd row
The 1 in the **3**rd row is in column 4 $\Rightarrow$ 4th row becomes **3**rd row
The 1 in the **4**th row is in column 2 $\Rightarrow$ 2nd row becomes **4**th row



## Transpositions

Transposition matrices are special permutation matrices.
They have only two 1s, which are not diagonal entries,
and they interchange only two rows.

**Example:**

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{P} \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}_{A} = \underbrace{\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}}_{PA}$$

The matrix $P$ transposes the 2nd row with the 4th row and leaves all other rows unchanged.

## Facts about permutation matrices

1. The product of permutation matrices is again a permutation matrix
2. For every permutation matrix $PP^\top = P^\top P = I$ ($I$=identity matrix).

   From that it follows that $P^\top = P^{-1}$.

   In Words: The inverse of a permutation matrix is its transposed matrix.
3. Every permutation matrix can be represented as the product of transposition matrices.
4. The identity matrix is also a permutation matrix.

## Swap a permutation matrix and a Frobenius matrix

Example:

$$\underbrace{\left[\begin{array}{cc|cccc} 1 &&&&& \\ & 1 &&&& \\ \hline &&0&0&1&0\\ &&1&0&0&0\\ &&0&0&0&1\\ &&0&1&0&0 \end{array}\right]}_{P}\underbrace{\left[\begin{array}{cc|cccc} 1&&&&&\\ &1&&&&\\ \hline &\times&1&&&\\ &\times&&1&&\\ &\times&&&1&\\ &\times&&&&1\end{array}\right]}_{L}$$

$$= \underbrace{\left[\begin{array}{cc|cccc} 1&&&&&\\ &1&&&&\\ \hline &\times&1&&&\\ &\times&&1&&\\ &\times&&&1&\\ &\times&&&&1\end{array}\right]}_{\widetilde{L}}\underbrace{\left[\begin{array}{cc|cccc} 1 &&&&& \\ & 1 &&&& \\ \hline &&0&0&1&0\\ &&1&0&0&0\\ &&0&0&0&1\\ &&0&1&0&0 \end{array}\right]}_{P}$$

If the permuation matrix $P$ does not interchange the first $k$ rows, then for every Frobenius-matrix of type $k$

$$PL = \widetilde{L}P,$$

where $\widetilde{L}$ is the Frobenius-matrix that we get, when we apply the right lower block of $P$ to the elements of $L$ in the $k$-th column.



The step

$\ldots \rightarrow$ transpose two rows $\rightarrow$ eliminate the elements below the pivot element by row operations $\rightarrow$

in matrix language:

$\ldots \rightarrow$ multiply by permutation matrix $P_k$ $\rightarrow$ multiply by Frobenius-Matrix $L_k$ $\rightarrow \ldots$

This gives: $L_{n-1}\,P_{n-1}\,L_{n-2}\,P_{n-2}\ldots L_k\,P_k\ldots L_2\,P_2\,L_1P_1\,A = U.$

Now we can swap the permutation matrices with the Frobenius-matrices (see last slide) and get

$$\begin{aligned}
& L_{n-1}\,P_{n-1}\,L_{n-2}\,P_{n-2}\ldots L_k\,P_k\ldots L_2\,P_2\,L_1P_1\,A = U\\
\Rightarrow\ & L_{n-1}\,\widetilde{L}_{n-2}\,P_{n-1}\,P_{n-2}\ldots L_k\,P_k\ldots L_2\,P_2\,L_1P_1\,A = U\\
& \vdots\\
\Rightarrow\ & L_{n-1}\,\widetilde{L}_{n-2}\ldots\widetilde{L}_k\ldots\widetilde{L}_1\,\underbrace{P_{n-1}\,P_{n-2}\ldots P_2\,P_1}_{P}\,A = U\\
\Rightarrow\ & P\,A = \underbrace{\widetilde{L}_1^{-1}\,\widetilde{L}_2^{-1}\ldots\widetilde{L}_k^{-1}\ldots\widetilde{L}_{n-2}^{-1}\,L_{n-1}^{-1}}_{L}\,U.
\end{aligned}$$

Also: $P\,A = L\,U.$



In practice the **permutation matrix** $P$ is not explicitly computed. Instead the information about row transposition is saved in a **permutation vector**.

The method is explained by example on the next slides.

**Remark:** The MATLAB-comand `[L,U,P]=lu(A)` returns also $P$.

## Example: $LU$-factorization of a $4\times 4$-matrix with column pivoting

| $L$ | $A$ | permutation vector |
|---|---|---|
| $\begin{bmatrix}1&&&\\&1&&\\&&1&\\&&&1\end{bmatrix}$ | $\begin{bmatrix}\times&\times&\times&\times\\\times&\times&\times&\times\\\times&\times&\times&\times\\\times&\times&\times&\times\end{bmatrix}$ | $p=\begin{bmatrix}1\\2\\3\\4\end{bmatrix}$ |

**Given:** An invertable matrix $A$.

**Aim:** Compute $P, L, U$ so that $PA = LU$.

**Method:** $A$ is transformed to $U$ using row operations

The factors with which we multiply the rows before subtracting we save in $L$ .

When we transpose rows we also transpose the rows in $L$ and the rows in the permutation vector $p$.

Finally, from $p$, we can compute the permutation matrix $P$.

**Step 1:** Find the pivot element (i. e. the greatest absolute value) in the 1st row of $A$.

## Example: $LU$-factorization of a $4 \times 4$-matrix with column pivoting

After the first step:

$$L = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \qquad A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \otimes & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \qquad \text{permutation vector } p = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

**Next step:** Transpose the row with the pivot element $\otimes$ with the first row.



## Example: $LU$-factorization of a $4 \times 4$-matrix with column pivoting

$$L = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \qquad A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \otimes & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \qquad \text{permutation vector } p = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

**Step:** Transpose the row with the pivot element $\otimes$ with the first row.
Also transpose the corresponding rows in the permutation vecto



## Example: $LU$-factorizaion of a $4 \times 4$-matrix with column pivoting

$$L = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & & 1 \end{bmatrix} \qquad A = \begin{bmatrix} \otimes & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \qquad \text{permutation vector } p = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix}$$

**Next step:** Eliminate the elements in the 1st column below the pivot-element using row operations.
Write the corresponding factors in $L$.

## Example: $LU$-factorization of a $4 \times 4$-matrix with column pivoting

$$L = \begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & & 1 & \\ \times & & & 1 \end{bmatrix} \qquad A \text{ modified} = \begin{bmatrix} \otimes & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix} \qquad \text{permutation vector } p = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix}$$

**Next step:** Find pivot element (greatest absolute value) below the 1st element) in the 2nd column.

**Example:** $LU$**-factorization of a** $4 \times 4$**-matrix with column pivoting**

| $L$ | $A$ modified | permutation vector |
|---|---|---|
| $\begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & & 1 & \\ \times & & & 1 \end{bmatrix}$ | $\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \otimes & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}$ | $p = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix}$ |

**Next step:** Transpose the row with the pivot element with the 2nd row. Also transpose the corresponding rows in the $L$-matrix and in the permutation vector $p$.



**Example:** $LU$**-factorization of a** $4 \times 4$**-matrix with column pivoting**

| $L$ | $A$ modified | permutation vector |
|---|---|---|
| $\begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & & 1 & \\ \times & & & 1 \end{bmatrix}$ | $\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & \otimes & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}$ | $p = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix}$ |

**Step:** Transpose the row with the pivot element with the 2nd row. Also transpose the corresponding rows in the $L$-matrix and in the permutation vector $p$.



**Example:** $LU$**-factorization of a** $4 \times 4$**-matrix with column pivoting**

| $L$ | $A$ modified | permutation vector |
|---|---|---|
| $\begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & & 1 & \\ \times & & & 1 \end{bmatrix}$ | $\begin{bmatrix} \times & \times & \times & \times \\ 0 & \otimes & \times & \times \\ 0 & \times & \times & \times \\ 0 & \times & \times & \times \end{bmatrix}$ | $p = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 4 \end{bmatrix}$ |

**Next step:** Eliminate entries below the pivot element using row operations. Write the multiplication factors in $L$.

**Example:** $LU$**-factorization of a** $4 \times 4$**-matrix with column pivoting**

| $L$ | $A$ modified | permutation vector |
|---|---|---|
| $\begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & \times & 1 & \\ \times & \times & & 1 \end{bmatrix}$ | $\begin{bmatrix} \times & \times & \times & \times \\ 0 & \otimes & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \times & \times \end{bmatrix}$ | $p = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 4 \end{bmatrix}$ |

**Next step:** Find pivot element (greatest absolute value) below the 2nd row and in the 3rd column.

## Example: $LU$-factorization of a $4 \times 4$-matrix with column pivoting

$L$ | $A$ modified | permutation vector

$$\begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & \times & 1 & \\ \times & \times & & 1 \end{bmatrix} \qquad \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \otimes & \times \end{bmatrix} \qquad p = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 4 \end{bmatrix}$$

**Next step:** Transpose the row with the pivot element with the 3rd row.
Also transpose the corresponding rows in the $L$-matrix and in the permutation vector $p$.



## Example: $LU$-factorization of a $4 \times 4$-matrix with column pivoting

$L$ | $A$ modified | permutation vector

$$\begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & \times & 1 & \\ \times & \times & & 1 \end{bmatrix} \qquad \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \times & \times \\ 0 & 0 & \otimes & \times \end{bmatrix} \qquad p = \begin{bmatrix} 3 \\ 1 \\ 2 \\ 4 \end{bmatrix}$$

**Step:** Transpose the row with the pivot element with the 3rd row.
Also transpose the corresponding rows in the $L$-matrix and in the permutation vector $p$.



## Example: $LU$-factorization of a $4 \times 4$-matrix with column pivoting

$L$ | $A$ modified | permutation vector

$$\begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & \times & 1 & \\ \times & \times & & 1 \end{bmatrix} \qquad \begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \otimes & \times \\ 0 & 0 & \times & \times \end{bmatrix} \qquad p = \begin{bmatrix} 3 \\ 1 \\ 4 \\ 2 \end{bmatrix}$$

**Next step:** Make the element below the pivot element 0 using row operations.
Write the corresponding factor in $L$.
We will see the result on the next slide.

## Example: $LU$-factorization of a $4 \times 4$-matrix with column pivoting

$L$ | $A$ modified | permutation vector

$$\begin{bmatrix} 1 & & & \\ \times & 1 & & \\ \times & \times & 1 & \\ \times & \times & \times & 1 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} \times & \times & \times & \times \\ 0 & \times & \times & \times \\ 0 & 0 & \otimes & \times \\ 0 & 0 & 0 & \times \end{bmatrix}}_{U} \qquad p = \begin{bmatrix} 3 \\ 1 \\ 4 \\ 2 \end{bmatrix}$$

The matrices $L, U$ which we just computed satisfy $PA = LU$, where

$$P = \begin{bmatrix} & & 1 & \\ 1 & & & \\ & & & 1 \\ & 1 & & \end{bmatrix}$$

$P$ moves the 3rd row to the 1st row, moves the 1st row to the 2nd row, moves the 4th row to the 3rd row, and moves the 2nd row to the 4th row.

This information is encoded in the permutation vector $p$.

**Remark 1:** We do **not need** to compute $P$ to solve linear equation systems. Instead we work with the vector $p$.

**Remark 2:** In practice we also do not perform the row transpositions, we use $p$.

**Solving a linear equation system:**

With $PA = LU$ it follows that

$$A\,x = b \quad \Rightarrow \quad PA\,x = Pb \quad \Rightarrow \quad LU\,x = Pb.$$

The equation system is solved by forward and backward substitution with the vector $Pb$.

The vector $Pb$ is obtained by interchanging the entries of $b$ corresponding to the entries of the permutation vector $p$.