Gaussian Elimination and PLU-factorization

Tom Lyche

University of Oslo

Norway

Linear Equations

Component form:

$$\begin{array}{ccccc}
a_{11}x_1 + a_{12}x_2 + & \cdots & + a_{1n}x_n & = b_1 \\
a_{21}x_1 + a_{22}x_2 + & \cdots & + a_{2n}x_n & = b_2 \\
\vdots \qquad \vdots & & \vdots & \vdots \\
a_{n1}x_1 + a_{n2}x_2 + & \cdots & + a_{nn}x_n & = b_n
\end{array}$$

n equations in n unknowns.

Matrix form

$$A\boldsymbol{x} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix} = \boldsymbol{b}.$$

- Assume $A \in \mathbb{R}^{n,n}$ or $A \in \mathbb{C}^{n,n}$ is nonsingular. The system then has a unique solution.
- Consider for simplicity $A \in \mathbb{R}^{n,n}$.

Perturbation Example

- Consider the system of two linear equations

$$\begin{aligned} x_1 \quad &+x_2 &= \quad 20 \\ x_1 \quad &+0.999x_2 &= \quad 19.99 \end{aligned}$$

- The exact solution is $x_1 = x_2 = 10$.
- Suppose we replace the second equation by

$$x_1 + 1.001x_2 = 19.99,$$

- the exact solution changes to $x_1 = 30$, $x_2 = -10$.
- A small change in one of the coefficients, from 0.999 to 1.001, changed the exact solution by a large amount.

Ill Conditioning

- A mathematical problem in which the solution is very sensitive to changes in the data is called **ill-conditioned** .
- Such problems are difficult to solve on a computer.
- If at all possible, the mathematical model should be changed to obtain a more well-conditioned or properly-posed problem.

Pertubed System

- We consider what effect a small change (perturbation) in the data A,$\boldsymbol{b}$ has on the solution $\boldsymbol{x}$ of a linear system $A\boldsymbol{x} = \boldsymbol{b}$.
- Suppose $\boldsymbol{y}$ solves $(A + E)\boldsymbol{y} = \boldsymbol{b}+\boldsymbol{e}$ where E is a (small) $n \times n$ matrix and $\boldsymbol{e}$ a (small) vector.
- How large can $\boldsymbol{y}-\boldsymbol{x}$ be?
- To measure this we use vector and matrix norms.

Conditions on the norms

- $\|\cdot\|$ will denote a vector norm on $\mathbb{C}^n$ and also a submultiplicative matrix norm on $\mathbb{C}^{n,n}$ which in addition is subordinate to the vector norm.
- Thus for any $A, B \in \mathbb{C}^{n,n}$ and any $\boldsymbol{x} \in \mathbb{C}^n$ we have

$$\|AB\| \leq \|A\|\,\|B\| \text{ and } \|A\boldsymbol{x}\| \leq \|A\|\,\|\boldsymbol{x}\|.$$

- This is satisfied if the matrix norm is the operator norm corresponding to the given vector norm.
- Another example: $\|A\boldsymbol{x}\|_2 \leq \|A\|_F \|\boldsymbol{x}\|_2$

Absolute and Relative error

- The difference $\|\boldsymbol{y}-\boldsymbol{x}\|$ measures the absolute error in $\boldsymbol{y}$ as an approximation to $\boldsymbol{x}$, while $\|\boldsymbol{y}-\boldsymbol{x}\|/\|\boldsymbol{x}\|$ or $\|\boldsymbol{y}-\boldsymbol{x}\|/\|\boldsymbol{y}\|$ is a measure for the relative error.

Right hand side

We consider first a perturbation in the right-hand side $\boldsymbol{b}$.

Theorem 1. *Suppose* $A \in \mathbb{C}^{n,n}$ *is invertible,* $\boldsymbol{b}, \boldsymbol{e} \in \mathbb{C}^n$*,* $\boldsymbol{b} \neq \boldsymbol{0}$ *and* $A\boldsymbol{x} = \boldsymbol{b}$*,* $A\boldsymbol{y} = \boldsymbol{b} + \boldsymbol{e}$*. Then*

$$\frac{1}{K(A)} \frac{\|\boldsymbol{e}\|}{\|\boldsymbol{b}\|} \leq \frac{\|\boldsymbol{y} - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq K(A) \frac{\|\boldsymbol{e}\|}{\|\boldsymbol{b}\|}, \quad K(A) = \|A\| \|A^{-1}\|. \tag{1}$$

Proof: Subtracting $A\boldsymbol{x} = \boldsymbol{b}$ from $A\boldsymbol{y} = \boldsymbol{b} + \boldsymbol{e}$ we have $A(\boldsymbol{y} - \boldsymbol{x}) = \boldsymbol{e}$ or $\boldsymbol{y} - \boldsymbol{x} = A^{-1}\boldsymbol{e}$. Thus $\|\boldsymbol{y} - \boldsymbol{x}\| = \|A^{-1}\boldsymbol{e}\| \leq \|A^{-1}\| \, \|\boldsymbol{e}\|$. Moreover, $\|\boldsymbol{b}\| = \|A\boldsymbol{x}\| \leq \|A\| \, \|\boldsymbol{x}\|$ or $1/\|\boldsymbol{x}\| \leq \|A\|/\|\boldsymbol{b}\|$. Therefore

$$\frac{\|\boldsymbol{y} - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \|A^{-1}\| \, \|A\| \frac{\|\boldsymbol{e}\|}{\|\boldsymbol{b}\|},$$

which proves the rightmost inequality in (1). Since $A(\boldsymbol{y} - \boldsymbol{x}) = \boldsymbol{e}$ and $\boldsymbol{x} = A^{-1}\boldsymbol{b}$, we have $\|\boldsymbol{e}\| \leq \|A\| \, \|\boldsymbol{y} - \boldsymbol{x}\|$ and $\|\boldsymbol{x}\| \leq \|A^{-1}\| \, \|\boldsymbol{b}\|$. This gives

$$\frac{\|\boldsymbol{y} - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \geq \frac{1}{\|A\| \, \|A^{-1}\|} \frac{\|\boldsymbol{e}\|}{\|\boldsymbol{b}\|},$$

Upper bound

$$\frac{\|\boldsymbol{y}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \le K(A)\frac{\|\boldsymbol{e}\|}{\|\boldsymbol{b}\|}, \quad K(A) = \|A\|\|A^{-1}\|.$$

- $\|\boldsymbol{e}\|/\|\boldsymbol{b}\|$ is a measure for the size of the perturbation $\boldsymbol{e}$ relative to the size of $\boldsymbol{b}$. $\|\boldsymbol{y}-\boldsymbol{x}\|/\|\boldsymbol{x}\|$ can in the worst case be

$$K(A) = \|A\|\|A^{-1}\|$$

times as large as $\|\boldsymbol{e}\|/\|\boldsymbol{b}\|$.

Condition number

- $K(A)$ is called the **condition number with respect to inversion of a matrix**, or just the condition number, if it is clear from the context that we are talking about solving linear systems or inverting a matrix.
- The condition number depends on the matrix A and on the norm used. If $K(A)$ is large, A is called **ill-conditioned** (with respect to inversion).
- If $K(A)$ is small, A is called **well-conditioned** (with respect to inversion).

Condition number 2

- Since $\|A\|\|A^{-1}\| \geq \|AA^{-1}\| = \|I\| \geq 1$ we always have $K(A) \geq 1$.
- $\|I\| \geq 1$ since subordinance implies $\|\boldsymbol{x}\| = \|I\boldsymbol{x}\| \leq \|I\|\|\boldsymbol{x}\|$ for any $\boldsymbol{x}$.
- Since all matrix norms are equivalent, the dependence of $K(A)$ on the norm chosen is less important than the dependence on A.
- Sometimes we choose the 2-norm when discussing properties of the condition number, and the $1-$ and $\infty-$ norm when we compute it or estimate it.

The 2-norm

- Suppose A has singular values $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_n > 0$. We have $K_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{\sigma_1}{\sigma_n}$ if the 2 norm is used.
- A normal matrix A can be diagonalized by a unitary similarity transformation. $U^H A U = D$, where U is unitary and D is diagonal with the eigenvalues of A on the diagonal. It follows that UDU^H is the singular value decomposition of A and $\sigma_i = |\lambda_i|$, where $|\lambda_1| \geq \cdots \geq |\lambda_n| > 0$ are the absolute values of the eigenvalues of A.
- Thus $K_2(A) = \|A\|_2 \|A^{-1}\|_2 = \frac{|\lambda_1|}{|\lambda_n|}, \quad A \text{ normal}.$
- It follows that A is ill-conditioned with respect to inversion if and only if σ_1/σ_n is large, or $|\lambda_1|/|\lambda_n|$ is large when A is normal.

Continuity of the inverse

The next result shows that small changes in A gives small changes in the inverse.

- Suppose $A \in \mathbb{C}^{n,n}$ is nonsingular and let $\|\cdot\|$ be a submultiplicative matrix norm on $\mathbb{C}^{n,n}$. If $E \in \mathbb{C}^{n,n}$ is so small that $r := \|A^{-1}E\| < 1$ then:
- $\|(A+E)^{-1}\| \leq \frac{\|A^{-1}\|}{1-r}$.

Perturbation in A

We consider next a perturbation in A.

Theorem 2. *Suppose* $A, E \in \mathbb{C}^{n,n}$, $\boldsymbol{b} \in \mathbb{C}^n$ *with* A *invertible and* $\boldsymbol{b} \neq \mathbf{0}$. *If* $r := \|A^{-1}E\| < 1$ *for some norm then* $A+E$ *is invertible. If* $A\boldsymbol{x} = \boldsymbol{b}$ *and* $(A+E)\boldsymbol{y} = \boldsymbol{b}$ *then*

$$\frac{\|\boldsymbol{y}-\boldsymbol{x}\|}{\|\boldsymbol{y}\|} \leq r \leq K(A)\frac{\|E\|}{\|A\|}. \tag{2}$$

$$\frac{\|\boldsymbol{y}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq \frac{r}{1-r} \leq \frac{K(A)}{1-r}\frac{\|E\|}{\|A\|}. \tag{3}$$

Proof: The matrix $A+E$ is invertible since $r < 1$. (2) follows easily by taking norms in the equation $\boldsymbol{x}-\boldsymbol{y} = A^{-1}E\boldsymbol{y}$ and dividing by $\|y\|$. Solving the equation $\boldsymbol{x}-\boldsymbol{y} = A^{-1}E\boldsymbol{y}$ for $\boldsymbol{y}$ we find $\boldsymbol{y} = (I+A^{-1}E)^{-1}\boldsymbol{x}$ and hence $\boldsymbol{x}-\boldsymbol{y} = A^{-1}E(I+A^{-1}E)^{-1}\boldsymbol{x}$. Taking norms and using $\|(A+E)^{-1}\| \leq \frac{\|A^{-1}\|}{1-r}$ we obtain (3).

Comments

$$\frac{\|\boldsymbol{y}-\boldsymbol{x}\|}{\|\boldsymbol{y}\|} \leq K(A)\frac{\|E\|}{\|A\|}.$$

- $\|E\|/\|A\|$ is a measure of the size of the perturbation E in A relative to the size of A.
- The condition number again plays a crucial role.
- It can be shown that the upper bound can be attained for any A and any $\boldsymbol{b}$.

The Residual

Suppose we have computed an approximate solution $\boldsymbol{y}$ to $A\boldsymbol{x} = \boldsymbol{b}$. The vector $\boldsymbol{r}(\boldsymbol{y}:) = A\boldsymbol{y} - \boldsymbol{b}$ is called the **residual vector** , or just the residual. We can bound $\boldsymbol{x} - \boldsymbol{y}$ in term of $\boldsymbol{r}(\boldsymbol{y})$.

Theorem 3. *Suppose* $A \in \mathbb{C}^{n,n}$, $\boldsymbol{b} \in \mathbb{C}^n$, A *is nonsingular and* $\boldsymbol{b} \neq \boldsymbol{0}$. *Let* $\boldsymbol{r}(\boldsymbol{y}) = A\boldsymbol{y} - \boldsymbol{b}$ *for each* $\boldsymbol{y} \in \mathbb{C}^n$. *If* $A\boldsymbol{x} = \boldsymbol{b}$ *then*

$$\frac{1}{K(A)} \frac{\|\boldsymbol{r}(\boldsymbol{y})\|}{\|\boldsymbol{b}\|} \leq \frac{\|\boldsymbol{y} - \boldsymbol{x}\|}{\|\boldsymbol{x}\|} \leq K(A) \frac{\|\boldsymbol{r}(\boldsymbol{y})\|}{\|\boldsymbol{b}\|}. \tag{4}$$

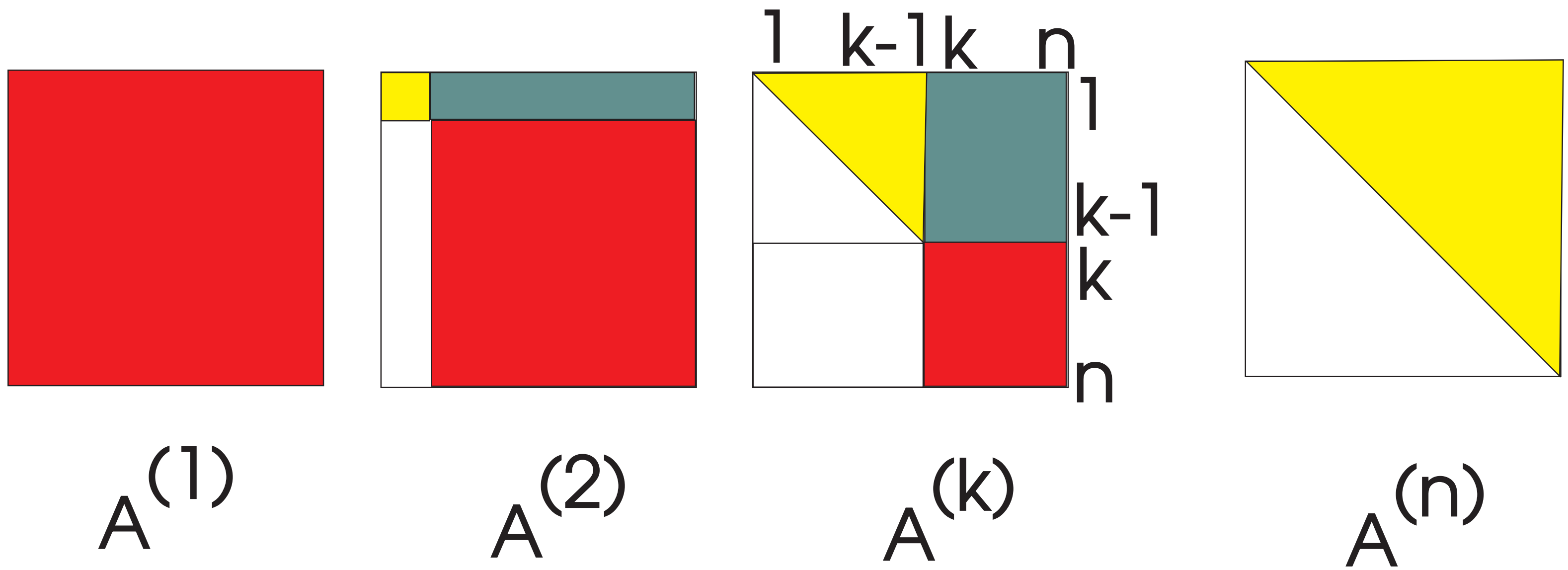
Comments

$$\frac{1}{K(A)}\frac{\|\boldsymbol{r}(\boldsymbol{y})\|}{\|\boldsymbol{b}\|} \le \frac{\|\boldsymbol{y}-\boldsymbol{x}\|}{\|\boldsymbol{x}\|} \le K(A)\frac{\|\boldsymbol{r}(\boldsymbol{y})\|}{\|\boldsymbol{b}\|}$$

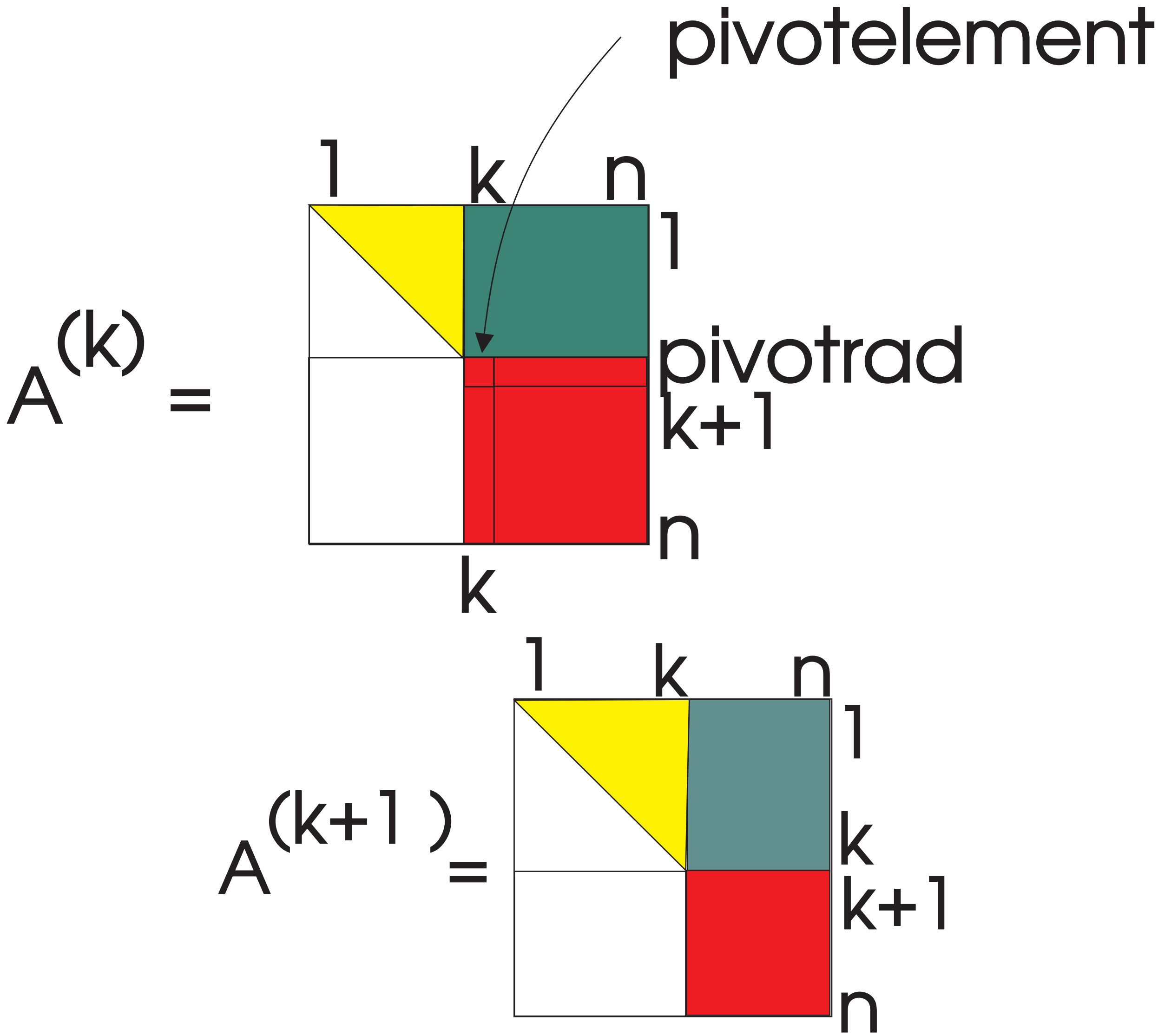
- $\|\boldsymbol{y}-\boldsymbol{x}\|/\|\boldsymbol{x}\| \approx \|\boldsymbol{r}(\boldsymbol{y})\|/\|\boldsymbol{b}\|$ if A is well-conditioned, .
- In other words, the accuracy in $\boldsymbol{y}$ is about the same order of magnitude as the residual (as long as $\|\boldsymbol{b}\| \approx 1$).
- If A is ill-conditioned, anything can happen.
- We can for example have an accurate solution even if the residual is large.

Gaussian elimination

- start with a system $Ax = b$. Set $A^{(1)} = A$ and $b^{(1)} = b$.
- LU-factorization: generate a sequence of systems $A^{(k)}x = b^{(k)}$ for $k = 2, \ldots, n$.
- solution: find x from $A^{(n)}x = b^{(n)}$ by back substitution



Pivotelement



Permutation Matrices

Definition 1. *Suppose* $\boldsymbol{p} = (i_1, \ldots, i_n)$ *is a permutation of the integers* $1, 2, \ldots, n$. *A* **permutation matrix** *is a matrix of the form*

$$P = I(:, \boldsymbol{p}) = [\boldsymbol{e}_{i_1}, \boldsymbol{e}_{i_2}, \ldots, \boldsymbol{e}_{i_n}] \in \mathbb{R}^{n,n},$$

where $\boldsymbol{e}_{i_1}, \ldots, \boldsymbol{e}_{i_n}$ *is a permutation of the unit vectors* $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n \in \mathbb{R}^n$.

- Since P has orthonormal columns it is orthogonal. Thus $P^T P = PP^T = I$, $P^{-1} = P^T$, and P^T is also a permutation matrix.
- Post-multiplying a matrix A by a permutation matrix results in a permutation of the columns, $AP = [A\boldsymbol{e}_{i_1}, \ldots, A\boldsymbol{e}_{i_n}] = A(:, p)$
- pre-multiplying by a permutation matrix gives a permutation of the rows. In symbols $P^T A = (A^T P)^T = (A^T(:, p))^T = A(p, :)$.

Exchange matrix

Definition 2. *We define a particularly simple permutation matrix called an* (j,k)-Exchange matrix I_{jk} *by exchanging column* j *and* k *of the identity matrix.*

- $I_{jk} = I_{kj}$ and an exchange matrix is symmetric.
- Since we obtain the identity by applying I_{jk} twice we see that $I_{jk}^2 = I$ and an exchange matrix is equal to its own inverse.
- Post-multiplying a matrix by an exchange matrix interchanges two columns of the matrix,
- pre-multiplication interchanges two rows.

Row Interchanges

- Consider the 3×3 system

$$A\boldsymbol{x} = \begin{bmatrix} 4 & 1 & 4 \\ 2 & -4 & 0 \\ 2 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 18 \\ -6 \\ 3 \end{bmatrix} = \boldsymbol{b}$$

- Use row interchanges. They are not necessary in this example, but are included in order to illustrate the general discussion below.

1. A Row Interchange

Interchange rows 1 and 3

$$\begin{bmatrix} 4 & 1 & 4 \\ 2 & -4 & 0 \\ 2 & -1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 18 \\ -6 \\ 3 \end{bmatrix} \to \begin{bmatrix} 2 & -1 & 1 \\ 2 & -4 & 0 \\ 4 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \\ 18 \end{bmatrix}.$$

Next we reformulate what we just did to the coefficient matrix in matrix terms . We start with $A^{(1)} := A$. We then interchange rows 1 and 3 in $A^{(1)}$ to obtain

$$B^{(1)} := I_{3,1} A^{(1)} = \begin{bmatrix} 2 & -1 & 1 \\ 2 & -4 & 0 \\ 4 & 1 & 4 \end{bmatrix}.$$

2. Elimination in Column 1

We subtract row 1 from row 2 and 2 times row 1 from row 3

$$\begin{bmatrix} 2 & -1 & 1 \\ 2 & -4 & 0 \\ 4 & 1 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -6 \\ 18 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & -1 & 1 \\ 0 & -3 & -1 \\ 0 & 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -9 \\ 12 \end{bmatrix}$$

$$A^{(2)} := M_1^{(1)} B^{(1)} = \begin{bmatrix} 2 & -1 & 1 \\ 0 & -3 & -1 \\ 0 & 3 & 2 \end{bmatrix} \text{ where } M_1^{(1)} := \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix}.$$

$$A^{(2)} := M_1^{(1)} I_{31} A$$

3. Interchange rows 2 and 3

$$\begin{bmatrix} 2 & -1 & 1 \\ 0 & -3 & -1 \\ 0 & 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ -9 \\ 12 \end{bmatrix} \to \begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 2 \\ 0 & -3 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 12 \\ -9 \end{bmatrix}.$$

$$B^{(2)} := I_{32} A^{(2)} = \begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 2 \\ 0 & -3 & -1 \end{bmatrix}.$$

$B^{(2)} := I_{32} M_1^{(1)} I_{31} A$

4. Elimination in Column 2

4) Finally, we subtract $(-1)\times$row 2 from row 3:

$$\begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 2 \\ 0 & -3 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 12 \\ -9 \end{bmatrix} \rightarrow \begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 12 \\ 3 \end{bmatrix}$$

$$A^{(3)} := M_2^{(2)} B^{(2)} = \begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 2 \\ 0 & 0 & 1 \end{bmatrix} \text{ where } M_2^{(2)} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

$A^{(3)} := M_2^{(2)} I_{32} M_1^{(1)} I_{31} A$ is upper triangular, $U := A^{(3)}$

Upper triangular system

$$\begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 3 \\ 12 \\ 3 \end{bmatrix}$$

Upper triangular system with the same solution set as the original system. Easy to solve $x = (1, 2, 3)^T$

PLU-factorization

- $U = M_2^{(2)} I_{32} M_1^{(1)} I_{31} A$
- We group the exchange matrices together by the following trick:
- Define $M_2 := M_2^{(2)}$ and $M_1 := I_{32} M_1^{(1)} I_{32}$.
- $U = M_2 I_{32} (I_{32} M_1 I_{32}) I_{31} A = M_2 M_1 I_{32} I_{31} A$
- $A = I_{31} I_{32} M_1^{-1} M_2^{-1} U$
- $A = PLU$, where $P = I_{31} I_{32}$, $L = M_1^{-1} M_2^{-1}$, and $U = A^{(3)}$.
- The matrix L is obtained easily from M_1 and M_2. We have

M's

$$M_1 = I_{32} M_1^{(1)} I_{32} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -2 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$$

$$M_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \; M_2^{-1} := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{bmatrix}, \; L = M_1^{-1} M_2^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix}$$

and

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 2 & -1 & 1 \\ 0 & 3 & 2 \\ 0 & 0 & 1 \end{bmatrix} = PLU.$$

Elementary Row Operations

The matrices M_1 and M_2 in the previous example can be written in outer product form as

$$M_1 = I - \begin{bmatrix} 0 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, \; M_2 = I - \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}$$

In general:

Elementary Row Operations II

For $1 \le k \le n-1$ and $\boldsymbol{l}_k = [l_{k+1,k}, \dots, l_{n,k}]^T \in \mathbb{R}^{n-k}$ we define the matrix

$$M_k := I - \begin{bmatrix} \mathbf{0} \\ \boldsymbol{l}_k \end{bmatrix} \boldsymbol{e}_k^T = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 & 0 & \cdots & 0 \\ \vdots & & \ddots & \vdots & \vdots & & \vdots \\ 0 & 0 & \cdots & 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & -l_{k+1,k} & 1 & \cdots & 0 \\ \vdots & & & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & -l_{n,k} & 0 & \cdots & 1 \end{bmatrix}, \tag{5}$$

where $\mathbf{0}$ is the zero vector in $\mathbb{R}^k$. We call M_k an **elementary row operation matrix.**

Elementary row operations III

If $A \in \mathbb{R}^{n,n}$ then the ith row of $M_k A$ is given by

$$\boldsymbol{e}_i^T M_k A = \boldsymbol{e}_i^T A - \boldsymbol{e}_i^T \begin{bmatrix} \mathbf{0} \\ \boldsymbol{l}_k \end{bmatrix} \boldsymbol{e}_k^T A = \begin{cases} \boldsymbol{e}_i^T A & i = 1, \ldots, k, \\ \boldsymbol{e}_i^T A - l_{ik} \boldsymbol{e}_k^T A & i = k+1, \ldots, n. \end{cases} \tag{6}$$

Thus M_k leaves the first k rows of A unchanged and row i of $M_k A$ equals row i of A minus l_{ik} times row k of A for $i = k+1, \ldots, n$. By choosing $l_{ik} = a_{ik}/a_{kk}$ the entries in column k of A under the diagonal will be zero. Indeed, for $i = k+1, \ldots, n$

$$(M_k A)_{ik} = \boldsymbol{e}_i^T M_k A \boldsymbol{e}_k = \boldsymbol{e}_i^T A \boldsymbol{e}_k - l_{ik} \boldsymbol{e}_k^T A \boldsymbol{e}_k = a_{ik} - l_{ik} a_{kk} = 0.$$

Elementary Row Operations

Lemma 3. *Suppose* M_k *is given by* (5) *for* $k = 1, \dots, n-1$*. Then*

$$L_k := M_k^{-1} = I + \begin{bmatrix} \mathbf{0} \\ \boldsymbol{l}_k \end{bmatrix} \boldsymbol{e}_k^T, \tag{7}$$

$$L_1 L_2 \cdots L_k = I + \sum_{j=1}^{k} \begin{bmatrix} \mathbf{0} \\ \boldsymbol{l}_j \end{bmatrix} \boldsymbol{e}_j^T, \tag{8}$$

Proof. (7) follows from the calculation

$$\Big(I + \begin{bmatrix} 0 \\ \boldsymbol{m}_k \end{bmatrix} \boldsymbol{e}_k^T\Big)\Big(I - \begin{bmatrix} 0 \\ \boldsymbol{m}_k \end{bmatrix} \boldsymbol{e}_k^T\Big) = I + \begin{bmatrix} 0 \\ \boldsymbol{m}_k \end{bmatrix} \boldsymbol{e}_k^T - \begin{bmatrix} 0 \\ \boldsymbol{m}_k \end{bmatrix} \boldsymbol{e}_k^T - \begin{bmatrix} 0 \\ \boldsymbol{m}_k \end{bmatrix} \boldsymbol{e}_k^T \begin{bmatrix} 0 \\ \boldsymbol{m}_k \end{bmatrix} \boldsymbol{e}_k^T =$$

using that $\boldsymbol{e}_k^T \begin{bmatrix} 0 \\ \boldsymbol{m}_k \end{bmatrix} = 0$. The proof of (8) is similar using induction on k. □

PLU-factorization

We obtain the factorization

$$U := A^{(n)} = M_{n-1}^{(n-1)} P_{n-1} \cdots M_2^{(2)} P_2 M_1^{(1)} P_1 A. \tag{9}$$

This can be converted into a PLU-factorization if we define $M_{n-1} := M_{n-1}^{(n-1)}$ and

$$M_k := P_{n-1} \cdots P_{k+1} M_k^{(k)} P_{k+1} \cdots P_{n-1},\ k = 1, \ldots, n-2. \tag{10}$$

M_k and $M_k^{(k)}$ differs only in that the multipliers under the diagonal in column k has been permuted.

PLU-factorization II

Using repeatedly that $P_k^2 = I$ for all k it follows that

$$U := M_{n-1} \cdots M_1 P_{n-1} \cdots P_1 A \tag{11}$$

$$A = P_1 \cdots P_{n-1} M_1^{-1} \cdots M_{n-1}^{-1} U$$

or $A = PLU$ where $P = P_1 \cdots P_{n-1}$ is a permutation matrix, $L = M_1^{-1} \cdots M_{n-1}^{-1}$ is unit lower triangular (and $U = A^{(n)}$ is upper triangular.

PLU-Theorem

The PLU-factorization exists if A is nonsingular

Theorem 4 (The PLU-theorem)**.** *A nonsingular matrix A has a factorization $A = PLU$, where P is a permutation matrix, L is unit lower triangular, and U is upper triangular.*

Proof. We use induction on n. $\square$

Stability Example

Without row interchanges:

$$\begin{aligned} 10^{-4}x_1 + 2x_2 &= 4 \\ x_1 + x_2 &= 3 \end{aligned} \quad \rightarrow \quad \begin{aligned} 10^{-4}x_1 + 2x_2 &= 4 \\ (1 - 2 \times 10^4)x_2 &= 3 - 4 \times 10^4 \end{aligned}$$

The exact solution is

$$x_2 = \frac{-39997}{-19999} \approx 2, \quad x_1 = \frac{4 - 2x_2}{10^{-4}} = \frac{20000}{19999} \approx 1.$$

Suppose we round the result of each arithmetic operation to three digits. The solutions fl(x_1) and fl(x_2) computed in this way is

$$\text{fl}(x_2) = 2, \quad \text{fl}(x_1) = 0.$$

The computed value 0 of x_1 is completely wrong.

Stability Example

With row interchanges:

$$\begin{aligned} x_1 + x_2 &= 3 \\ 10^{-4}x_1 + 2x_2 &= 4 \end{aligned} \quad \longrightarrow \quad \begin{aligned} x_1 + x_2 &= 3 \\ (2 - 10^{-4})x_2 &= 4 - 3 \times 10^{-4} \end{aligned}$$

Now the solution is computed as follows

$$x_2 = \frac{3.9997}{1.9999} \approx 2, \quad x_1 = 3 - x_2 \approx 1.$$

In this case rounding each calculation to three digits produces fl$(x_1) = 1$ and fl$(x_2) = 2$ which is quite satisfactory since it is the exact solution rounded to three digits.

Partial Pivoting

- In step k of Gaussian elimination we interchange row k with some row $r_k \geq k$ and then introduce zeros under the diagonal in the permuted matrix.
- The choice

$$r_k := \max\{|a_{i,k}^{(k)}| : k \leq i \leq n\}$$

with r_k the smallest such index in case of a tie is known as *partial pivoting*.
- With partial pivoting we have $|l_{ij}| \leq 1$ for all entries l_{ij} in L and this leads to an algorithm with reasonable numerical stability properties.

Solving a linear system

Solving a linear system $A\boldsymbol{x} = \boldsymbol{b}$, where $A \in \mathbb{R}^{n,n}$ and $\boldsymbol{b} \in \mathbb{R}^n$ by Gaussian elimination can be formulated as follows:

1. Find a PLU-factorization $A = PLU$ of A,
2. permute the entries of $\boldsymbol{b}$: $\boldsymbol{c} := P^T\boldsymbol{b}$,
3. solve the triangular system $L\boldsymbol{y} = \boldsymbol{c}$,
4. solve the triangular system $U\boldsymbol{x} = \boldsymbol{y}$.

Main loop, PLU-factorization

Vectorize main loop

For $i = k+1, \ldots, n$

For $j = k+1, \ldots, n$

$a_{ij} = a_{ij} - l_{ik} a_{kj}$

The right hand side can be written as an outer product

$$\begin{bmatrix} a_{k+1,k+1} & \cdots & a_{k+1,n} \\ \vdots & & \vdots \\ a_{n,k+1} & \cdots & a_{n,n} \end{bmatrix} - \begin{bmatrix} l_{k+1,k} \\ \vdots \\ l_{n,k} \end{bmatrix} \begin{bmatrix} a_{k,k+1} \cdots a_{k,n} \end{bmatrix}$$

$$A(k+1:n, k+1:n) = A(k+1:n, k+1:n) - L(k+1:n, k) * A(k, k+1:n).$$

PLU-factorization with partial pivoting

[PLU with Physical Row Interchanges]

$piv = 1 : n;$

for $k = 1, 2, \ldots, n-1$

$\quad [maxv, q] =$ max$(abs(A(k{+}1{:}n, k)));$

$\quad r = q + k - 1;$

$\quad piv([k\ r]) = piv([r\ k]);$

$\quad A([k\ r]) = A([r\ k]);$

$\quad A(k{+}1{:}n, k) = A(k{+}1{:}n, k)/A(k, k);$

$\quad A(k{+}1{:}n, k{+}1{:}n) = A(k{+}1{:}n, k{+}1{:}n) - A(k{+}1{:}n, k) * A(k, k{+}1{:}n);$

end

This algorithm requires $\frac{2}{3}n^3$ arithmetic operations.

$\boldsymbol{c} := P^T \boldsymbol{b}$

$$P = I(:, piv) = [\boldsymbol{e}_{i_1}, \boldsymbol{e}_{i_2}, \ldots, \boldsymbol{e}_{i_n}] \in \mathbb{R}^{n,n},$$

where $\boldsymbol{e}_{i_1}, \ldots, \boldsymbol{e}_{i_n}$ is a permutation of the unit vectors $\boldsymbol{e}_1, \ldots, \boldsymbol{e}_n \in \mathbb{R}^n$.

- pre-multiplying by a permutation matrix gives a permutation of the rows. In symbols $P^T A = A(piv, :)$.

In particular $\boldsymbol{c} = P^T \boldsymbol{b} = \boldsymbol{b}(piv)$

Forward Substitution

Algorithm 4 (Forward Substitution $L\boldsymbol{y} = \boldsymbol{c}$).

for $k = 1 : n$

$$y(k) = \big(c(k) - l(k, 1{:}k{-}1) * y(1{:}k{-}1)\big)/l(k, k);$$

end

This algorithm requires n^2 flops.

Backward Substitution

Algorithm 5 (Backward Substitution $U\boldsymbol{x} = \boldsymbol{y}$)**.**

for $k = n : -1 : 1$

$$x(k) = \big(y(k) - u(k, k{+}1{:}n) * x(k{+}1{:}n)\big)/u(k,k);$$

end

This algorithm requires n^2 flops.

Storage

The entries of L and U are located under and above the diagonal in A as shown here for $n = 4$

$$A = \begin{bmatrix} u_{11} & u_{12} & u_{13} & u_{14} \\ l_{21} & u_{22} & u_{23} & u_{24} \\ l_{31} & l_{32} & u_{33} & u_{34} \\ l_{41} & l_{42} & l_{43} & u_{44} \end{bmatrix}.$$

Using the Matlab functions `tril` and `triu` we recover the matrices P, L, and U such that $A = PLU$ as follows

$$P = I(:, piv),\ L = I + tril(A, -1),\ U = triu(A). \tag{12}$$

Computing Time

- This process can be time consuming if n is large.
- To quantify this we define a *flop* (floating point operation) as one of the floating point arithmetic operations, ie. multiplication, division, addition and subtraction.
- We denote by $\#$flops the total number of flops in an algorithm, i.e. the the sum of all multiplications, divisions, additions and subtractions.
- In many implementations the computing time T_A for an algorithm A applied to a large problem is proportional to $N_A := \#$flops.
- If this is true then we typically have $T_A = \alpha N_A$, where α is in the range 10^{-12} to 10^{-9} on a modern computer.

n^2 and n^3

Compare T_{PLU} and T_S Assume

$$T_{PLU} = \alpha \frac{2}{3} n^3, \quad T_S = \alpha 2n^2, \quad \alpha = \frac{3}{2} 10^{-9}$$

n	T_{PLU}	T_S
10^3	1s	0.003s
10^4	17min.	0.3s
10^6	32 years	51min