Chapter 2

Numerical Linear Algebra

2.1 Gauss Elimination

In this section it will be discussed how to solve a linear system of equations,

$$A\mathbf{x} = \mathbf{b} \tag{2.1}$$

where

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} \in \mathbb{R}^{n\times n}, \qquad \mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix} \in \mathbb{R}^n, \qquad \mathbf{b} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix} \in \mathbb{R}^n.$$

Here A and $\mathbf{b}$ are known, $\mathbf{x}$ is the unknown to be found. We will further assume that A is nonsingular, so (2.1) has a unique solution $\mathbf{x}$.

2.1.1 Notation

A matrix $U \in \mathbb{R}^{n\times n}$ is *upper triangular* if $u_{ij} = 0$ whenever $i > j$. Similar, L is *lower triangular* if $l_{ij} = 0$ whenever $i < j$, that is

$$U = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1n} \\ 0 & u_{22} & \cdots & u_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & u_{nn} \end{pmatrix}, \qquad L = \begin{pmatrix} l_{11} & 0 & \cdots & 0 \\ l_{21} & l_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & \cdots & l_{nn} \end{pmatrix}.$$

2.1.2 Naive Gauss elimination

The aim is here to transform the original system $A\mathbf{x} = \mathbf{b}$ to an upper triangular system $\tilde{A}\mathbf{x} = \tilde{\mathbf{b}}$, which is quite easy to solve. The procedure is that standard Gauss elimination, which for most of you is well known from earlier calculus classes.

Given the system of equations:

$$\begin{array}{lccccccccc} \mathrm{Eq}_1: & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ \mathrm{Eq}_2: & a_{21}x_1 & + & a_{22}x_2 & + & \cdots & + & a_{2n}x_n & = & b_2 \\ \vdots & & & & & & & & \vdots & \\ \mathrm{Eq}_n: & a_{n1}x_1 & + & a_{n2}x_2 & + & \cdots & + & a_{nn}x_n & = & b_n \end{array} \tag{2.2}$$

Keep Eq.1, and eliminate the first term of the remaining equations by

$$\mathrm{Eq}_i^{(2)} = \mathrm{Eq}_i - \frac{a_{i1}}{a_{11}} \cdot \mathrm{Eq}_1, \qquad i = 2, \ldots, n$$

that is,

$$a_{ij}^{(2)} = a_{ij} - m_{i1}a_{1j}, \qquad b_i^{(2)} = b_i - m_{i1}b_1, \text{ where } m_{i1} = a_{i1}/a_{11}, \qquad i, j = 2, \ldots, n.$$

The resulting system is of the form

$$\begin{array}{lccccccccc} \mathrm{Eq}_1: & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ \mathrm{Eq}_2^{(2)}: & & & a_{22}^{(2)}x_2 & + & \cdots & + & a_{2n}^{(2)}x_n & = & b_2^{(2)} \\ \vdots & & & & & & & \vdots & & \\ \mathrm{Eq}_n^{(2)}: & & & a_{n2}^{(2)}x_2 & + & \cdots & + & a_{nn}^{(2)}x_n & = & b_n^{(2)} \end{array}$$

Continue like this, and we end up with a triangular system of equations:

$$\begin{array}{lccccccccc} \mathrm{Eq}_1: & a_{11}x_1 & + & a_{12}x_2 & + & \cdots & + & a_{1n}x_n & = & b_1 \\ \mathrm{Eq}_2^{(2)}: & & & a_{22}^{(2)}x_2 & + & \cdots & + & a_{2n}^{(2)}x_n & = & b_2^{(2)} \\ \vdots & & & & & & & \vdots & & \\ \mathrm{Eq}_{n-1}^{(n-1)}: & & & & & a_{n-1,n-1}^{(n-1)}x_{n-1} & + & a_{n-1,n}^{(n-1)}x_n & = & b_{n-1}^{(n-1)} \\ \mathrm{Eq}_n^{(2)}: & & & & & & & a_{nn}^{(n)}x_n & = & b_n^{(n)} \end{array}$$

The transformed system can easily be solved, by starting with the last equation:

$$\begin{aligned} x_n &= b_n^{(n)}/a_{nn}^{(n)} \\ x_{n-1} &= (b_{n-1}^{(n-1)} - a_{n-1,n}^{(n-1)}x_n)/a_{n-1,n-1}^{(n-1)} \\ &\vdots \\ x_1 &= (b_1 - \sum_{j=2}^{n} a_{1j}x_j)/a_{11}. \end{aligned}$$

The solution of the triangular system is called *the back substitution.* The diagonal elements on the triangular matrix, $a_{ii}^{(i)}$ are called *pivot elements*, and the factors $m_{ik} = a_{ik}^{(i)}/a_{ii}^{(i)}$, for $i > k$ are called multipliers. They play a vital role later on.

Example 2.1.1.

$$\begin{array}{rcrcrcl} x_1 & + & 2x_2 & + & 3x_3 & = & 3, \\ 3x_1 & + & 4x_2 & + & & = & 3, \\ 2x_1 & + & 10x_2 & + & 4x_3 & = & 10, \end{array} \qquad \left| \quad \begin{array}{l} a_{11} = 1, \\ m_{21} = 3, \\ m_{31} = 2. \end{array} \right.$$

Keep the first equation, and eliminate the first term of the remaining equations.

$$\begin{array}{rcrcrcl|l} x_1 & + & 2x_2 & + & 3x_3 & = & 3, & \\ & - & 2x_2 & - & 3x_3 & = & -6, & a_{22}^{(2)} = -2, \\ & & 6x_2 & + & 2x_3 & = & 4, & m_{32} = -3, \end{array}$$

and the third step is

$$\begin{array}{rcrcrcl|l} x_1 & + & 2x_2 & + & 3x_3 & = & 3, & \\ & - & 2x_2 & - & 3x_3 & = & -6, & \\ & & & - & 7x_3 & = & -14, & a_{33}^{(3)} = -7. \end{array}$$

Back substitution gives

$$x_3 = 2, \qquad x_2 = 0, \qquad x_1 = 1$$

so the solution vector $\mathbf{x} = (1, 0, 2)^\top$.

Algorithm 1 Naive Gauss elimination with back-solution

Input: $(a_{ij})_{i,j=1}^n$, $(b_i)_{i=1}^n$.
for $k = 1, 2, \ldots, n-1$ **do**
  for $i = k+1, \ldots, n$ **do**
    $m_{ik} \leftarrow a_{ik}/a_{kk}$
    for $j = k+1, \ldots, n$ **do**
      $a_{ij} \leftarrow a_{ij} - m_{ik}a_{kj}$
    end for
    $b_i \leftarrow b_i - m_{ik}b_k$
  end for
end for
$x_n \leftarrow b_n/a_{nn}$ ▷ Back substitution
for $i = n-1, n-2, \ldots, 1$ **do**
  $x_i = \left(b_i - \sum_{j=i+1}^n a_{ij}x_j\right)/a_{ii}$
end for
Output $(x_i)_{i=1}^n$.

The algorithm will in principle works successfully as long as all the pivot elements $a_{ii} \neq 0$. This is a topic for further discussion.

The Gauss-elimination process can also be used as factorization method.

Definition 2.1.2. LU-factorization Let $A \in \mathbb{R}^{n\times n}$ be an invertible matrix. An *LU*-factorization of A is given by

$$A = LU$$

where L is a lower triangular matrix with 1's at the diagonal, and U is an upper triangular matrix with nonzero diagonal elements.

Theorem 2.1.3. *If the Gauss-elimination process is successfull, with only nonzero pivot elements, then A has an LU-factorization where U is the upper triangular matrix produced by the process, and*

$$L = \begin{pmatrix} 1 & & & & \\ m_{21} & 1 & & & \\ m_{31} & m_{32} & 1 & & \\ \vdots & & \ddots & \ddots & \\ m_{n1} & m_{n2} & \cdots & m_{n,n-1} & 1 \end{pmatrix}$$

Proof. Step k, $A^{(k+1)} = M_k A^{(k)}$ with $A^{(1)} = A$ in the algorithm is given by

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - m_{ik} a_{kj}^{(k)}.$$

where $m_{ik} = a_{ik}^{(k)} / a_{kk}^{(k)}$ are the multipliers. The Gauss-elimination process can then be considered as a sequence of matrix operations

$$U = A^{(n)} = M_{n-1} M_{n-2} \cdots M_1 A$$

thus

$$A = \overbrace{(M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1})}^{L} U,$$

so we need to prove that $L = M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1}$. Each transformation matrix M_k kan then be written as

$$M_k = \begin{pmatrix} 1 & & & & & & \\ 0 & 1 & & & & & \\ \vdots & & \ddots & & & & \\ \vdots & & & 1 & & & \\ \vdots & & & -m_{k+1,k} & 1 & & \\ \vdots & & & \vdots & & \ddots & \\ 0 & \cdots & \cdots & -m_{n,k} & 0 & \cdots & 1 \end{pmatrix} = I_n - \mathbf{m}_k \mathbf{e}_k^\top$$

where I_n is the identity matrix and

$$\mathbf{e}_k = (\overbrace{0, \ldots, 0}^{k-1}, 1, \overbrace{0, \ldots, 0}^{n-k-1})^\top, \qquad \mathbf{m}_k = (\overbrace{0, \ldots, 0}^{k}, \overbrace{m_{k+1,k}, \ldots, m_{n,k}}^{n-k})^\top.$$

Then $M_k^{-1} = I_n + \mathbf{m}_k \mathbf{e}_k^\top$ since

$$M_k^{-1} M_k = (I_n + \mathbf{m}_k \mathbf{e}_k^\top)(I_n - \mathbf{m}_k \mathbf{e}_k^\top) = I_n - \mathbf{m}_k \overbrace{\mathbf{e}_k^\top \mathbf{m}_k}^{=0} \mathbf{e}_k.$$

A similar argument shows that

$$M_k^{-1} M_{k+1}^{-1} = I_n + \mathbf{m}_k \mathbf{e}_k^\top + \mathbf{m}_{k+1} \mathbf{e}_{k+1}^\top$$

thus

$$M_1^{-1} M_2^{-1} \cdots M_{n-1}^{-1} = I_n + \sum_{k=1}^{n-1} \mathbf{m}_k \mathbf{e}_k^\top = L$$

as stated by the theorem. □

Solving the problem $A\mathbf{x} = \mathbf{b}$ is then done in three steps:

$A = LU$	The LU-factorization
Solve $L\mathbf{y} = \mathbf{b}$	Forward substitution
Solve $U\mathbf{x} = \mathbf{y}$	Back substitution

In the remaining part of this chapter, we will therefore use U for $A^{(n)}$, and similar $\mathbf{y}$ for $\mathbf{b}^{(n)}$, but not completely consistent. $\mathbf{b}^{(n)}$

Example 2.1.4. Consider Example 2.1.1 again. The LU-factorization of A is given as

$$\overbrace{\begin{pmatrix}1 & 2 & 1\\ 3 & 4 & 0\\ 2 & 10 & 4\end{pmatrix}}^{A} = \overbrace{\begin{pmatrix}1 & 0 & 0\\ 3 & 1 & 0\\ 2 & -3 & 1\end{pmatrix}}^{L} \cdot \overbrace{\begin{pmatrix}1 & 2 & 1\\ 0 & -2 & -3\\ 0 & 0 & -7\end{pmatrix}}^{U}$$

Computational complexity of the Gauss-elimination process

The complexity is measured in flop (floating point operations), that is one addition and one mulitiplication together, or one division. A matrix-vector multiplication $A\mathbf{x}$ with $A \in \mathbb{R}^{n\times n}$, $\mathbf{x} \in \mathbb{R}^n$ takes n^2 flop. The complexity of the different stages of an LU-factorization and forward and back substitution of a system of n equations are

$$\begin{aligned} A = LU \qquad & \overbrace{\sum_{k=1}^{n}(n-k)^2}^{a_{ij},(*,+)} + \overbrace{\sum_{k=1}^{n}(n-k)}^{m_{ij},(/)} = \frac{1}{3}n^3 - \frac{1}{3}n \approx \frac{1}{3}n^3 \text{ (flop)} \\ L\mathbf{y} = \mathbf{b} \qquad & \sum_{k=2}^{n}(k-1) = \frac{1}{2}n^2 - \frac{1}{2}n \approx \frac{1}{2}n^2 \text{ (flop} \\ U\mathbf{x} = \mathbf{y} \qquad & \sum_{k=2}^{n}(k-1) + n = \frac{1}{2}n^2 + \frac{1}{2}n \approx \frac{1}{2}n^2 \text{ (flop} \end{aligned}$$

Remark 2.1.5. In mathematics, it is common to write the solution of a linear system on the form $\mathbf{x} = A^{-1}\mathbf{b}$. If A^{-1} is known (which is usually not), this matrix-vector multiplication requires n^2 flop, exactly the same amount of work as the forward and back substitution if the LU-factorization is done. Thus, the LU factorized matrix works as a computational equivalent to the inverse. This is in particular useful if the same system is solved several times, with the same coefficient matrix but with different right hand sides $\mathbf{b}$.

Let $D = \text{diag}\{u_{11}, u_{22}, \ldots, u_{nn}\}$ be the diagonal matrix with the pivot elements u_{11} at the diagonal. Since, by assumption, $u_{ii} \neq 0$, D is invertible, giving rise to alternative LU-factorizations:

$$A = LU = LDD^{-1}U = \hat{L}\hat{U} \qquad \hat{L} = LD, \qquad \hat{U} = D^{-1}U,$$

where $\hat{U}$ is an upper triangular matrix with 1's at the diagonal. More important, if A is symmetric, that is $A = A^\top$ and $u_i i > 0$ for $i = 1, 2, \ldots, n$, then

$$A = CC^\top, \qquad C = L\sqrt{D},$$

where $\sqrt{D} = \text{diag}\{\sqrt{d_{11}}, \ldots, \sqrt{d_{nn}}\}$. This is called a *Cholesky factorization.*

Definition 2.1.6. A matrix $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant if

$$|a_{ii}| > \sum_{j=1, j\neq i}^{n} |a_{ij}|, \quad \text{for } i = 1, 2, \ldots, n.$$

Theorem 2.1.7. *If $A \in \mathbb{R}^{n\times n}$ is strictly diagonally dominant then it is nonsingular and it has an LU-factorization.*

Proof. We want to prove that all the pivot elements $a_{ii}^{(i)}$ created from the Gauss elimination process are nonzero. Assume A is diagonal dominant. Clearly, the first pivot element $a_{11} \neq 0$. The first step in the Gauss elimination process transforms

$$A \to \begin{pmatrix} a_{11} & \cdots \\ \mathbf{0} & \tilde{A}^{(2)} \end{pmatrix}$$

and it is sufficient to prove that also $\tilde{A}^{(2)}$ is strictly diagonal dominant, that is

$$|a_{ij}^{(2)}| > \sum_{j=2, j\neq i} |a_{ij}^{(2)}|, \quad \text{for } i = 2, 3, \ldots, n, \tag{2.3}$$

where

$$a_{ij}^{(2)} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j}. \tag{2.4}$$

In the proof, we will also use the two triangle inequalities $|x+y| \leq |x| + |y|$ and $|x-y| \geq |x| - |y|$. We then have:

$$\begin{aligned}
\sum_{j=2,j\neq i} |a_{ij}^{(2)}| &= \sum_{j=2,j\neq i} \left| a_{ij} - \frac{a_{i1}}{a_{11}} a_{1j} \right| && \text{from (2.4)} \\
&\leq \sum_{j=2,j\neq i} |a_{ij}| + \frac{|a_{i1}|}{|a_{11}|} |a_{1j}| && |x+y| \leq |x| + |y| \\
&= \sum_{j=1,j\neq i}^{n} |a_{ij}| - |a_{i1}| + \frac{|a_{i1}|}{|a_{11}|} \left(\sum_{j=1,j\neq i}^{n} |a_{1j}| - |a_{1i}| \right) \\
&\leq |a_{ii}| - |a_{i1}| + \frac{|a_{i1}|}{|a_{11}|} (|a_{11}| - |a_{1i}|) && \text{since } A \text{ is sdd} \\
&= |a_{ii}| - \frac{|a_{i1}|}{|a_{11}|} |a_{1i}| \\
&\leq \left| a_{ii} - \frac{a_{i1}}{a_{11}} a_{1i} \right| = |a_{ii}^{(2)}|, && |x| - |y| \leq |x-y| \text{ and (2.4)}
\end{aligned}$$

proving (2.3). □

Gauss elimination with pivoting

Obviously, even for nonsingular A-matrices, the Gauss elimination process may fail, if one of the pivot elements $a_{kk}^{(k)}$ becomes 0. This is not the only problem that may occur, which the following (quite standard) problem demonstrates:

Example 2.1.8.

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

where $|\varepsilon| \ll 1$. Obviously, the solution to this problem is $x_1 \approx 1$ and $x_2 \approx 1$. By the naive Gauss elimination process, the multiplier $m_{21} = 1/\varepsilon$, and the second equation is transferred into

$$(1 - 1/\varepsilon)x_2 = (2 - 1/\varepsilon) \qquad \Rightarrow \qquad x_2 = \frac{2 - 1/\varepsilon}{1 - 1/\varepsilon} \approx 1$$

as expected. The first unknown x_1 can be found from the first equation

$$x_1 = (1 - x_2)/\varepsilon \approx 0,$$

if the approximation to x_2 is used. Which is obviously wrong. The problem here is that we two almost equal numbers, 1 and x_2, are subracted, leading to large relative errors (see Section 1.6. Then this number is divided by a small number, ε, leading to large absolute errors.

Switching the two equations will solve the problem (try it yourself).

The problematic part is visible in the back substitution part of Algortithm 1,

$$x_i = \frac{1}{a_{ii}^{(i)}} \left(b_i^{(i)} - \sum_{j=i+1}^{n} a_{ij}^{(i)} x_j \right).$$

If x_i is of a reasonabel size, the pivot element $a_{ii}^{(i)}$ very small, then also $b_i^{(i)} \approx \sum_{j=i+1}^{n} a_{ij}^{(i)} x_j$, which are operations subjected to large errors in the computations. The remedy is to make the pivot elements small, which can be done by changing the order of the equations, a process called *pivoting*.

- **Partial row pivoting**: For each k in the algorithm, find the smallest q such that $|a_{qk}^{(k)}| = \max_{k \le i \le n} |a_{ik}^{(k)}|$. Switch row q and k.

- **Scaled partial row pivoting**:

  - Before the elimination process starts, find the scale $\mathbf{s} \in \mathbb{R}^n$ with the elements $s_i = \max_j |a_{ij}|$. This vector is not recomputed during the elimination process.

  - For each elimination step k, find the smalles q such that

$$\frac{a_{qk}^{(k)}}{s_q} = \max_{k \le i \le n} \frac{|a_{ik}^{(k)}|}{s_i}.$$

    Switch row k and q (including s_k and s_q).

Example 2.1.9. Let us apply these procedures on the problem $A\mathbf{x} = \mathbf{b}$, where

$$A = \begin{pmatrix} 1 & 2 & 1 \\ 3 & 4 & 0 \\ 2 & 10 & 4 \end{pmatrix}, \qquad \mathbf{b} = \begin{pmatrix} 3 \\ 3 \\ 10 \end{pmatrix}.$$

Partial row pivoting:

$$\left[\begin{array}{ccc|c} 1 & 2 & 1 & 3 \\ 3 & 4 & 0 & 3 \\ 2 & 10 & 4 & 10 \end{array}\right]$$ Choose the pivot row $q = 2$

$$\Downarrow$$ Switch row 1 and 2

$$\left[\begin{array}{ccc|c} 3 & 4 & 0 & 3 \\ 1 & 2 & 1 & 3 \\ 2 & 10 & 4 & 10 \end{array}\right]$$ $m_{21} = 1/3,\ m_{31} = 2/3$

$$\Downarrow$$ Eliminiation step

$$\left[\begin{array}{ccc|c} 3 & 4 & 0 & 3 \\ 0 & 2/3 & 1 & 2 \\ 0 & 22/3 & 4 & 8 \end{array}\right]$$ Choose the pivot row $q = 3$

$$\Downarrow$$ Switch row 2 and 3

$$\left[\begin{array}{ccc|c} 3 & 4 & 0 & 3 \\ 0 & 22/3 & 4 & 8 \\ 0 & 2/3 & 1 & 2 \end{array}\right]$$ $m_{32} = 1/11$

$$\Downarrow$$ Eliminiation step

$$\left[\begin{array}{ccc|c} 3 & 4 & 0 & 3 \\ 0 & 22/3 & 4 & 8 \\ 0 & 0 & 7/11 & 14/11 \end{array}\right]$$

The back substitution gives $\mathbf{x} = (1, 0, 2)^\top$.

Scaled partial row pivoting The scale vector is $\mathbf{s} = (2, 4, 10)^\top$. Then the first pivot element is chosen from $\max_i\{a_{i1}/s_i\} = \max\{1/2, 3/4, 2/10\}$, thus $q = 2$. Switch row 1 and 2 as above, including the relevant elements in the scale vectors. The next pivot element is chosen from $\max\{(2/3)/2, (23/3)/10\} = \max\{1/3, 23/30\}$, so $q = 3$. Switch row 2 and 3. In this particular case, the two pivoting strategies produces the same result, this is in general not the case.

In an implementation of these algorithm, the order of the rows are not changed, but the sequence in which the elimination process is done is kept track of by a pivot vector $\mathbf{p}$.

- Start with $\mathbf{p} = (1, 2, 3, \ldots, n)^\top$.
- Replace all row indices with p_i, $(a_{p_i,j},\ b_{p_i},\ s_{p_i})$.
- Before each elimination step, find the pivot row q and switch p_q and p_i.

A *permutation matrix* $P \in \mathbb{R}^{n\times n}$ is a permutation of the identity matrix I_n, thus at each row and column, there is one and only one element equal to 1, all other elements are 0.

- PA changes the order of the *rows* of A.
- AP changes the order of the *colums* of A.

Example 2.1.10.

$$\begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 1 \\ 3 & 4 & 0 \\ 2 & 10 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 4 & 0 \\ 2 & 10 & 4 \\ 1 & 2 & 1 \end{pmatrix}$$

We conclude this section with the following important result, here stated without proof:

Theorem 2.1.11. *If $A \in \mathbb{R}^{n\times n}$ is nonsingular, then there exist a permutation matrix P, a lower triangular matrix L with 1s at the diagonal, and an upper triangular matrix matrix U such that*

$$PA = LU.$$