1. Linear systems: direct methods

Reference: Trefethen & Bau, Numerical Linear Algebra, SIAM, Lectures 20 & 21

Given $n \in \mathbb{N}\setminus\{0,1\}$, $A \in \mathbb{R}^{n\times n}$, and $b \in \mathbb{R}^n$, find $x \in \mathbb{R}^n$ such that $Ax = b$.

$$A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,m} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,m} \end{bmatrix} \qquad b = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix}$$

1. Triangular systems are easy to solve

$$A = \begin{bmatrix} a_{1,1} & & \\ \vdots & \ddots & \\ a_{m,1} & \cdots & a_{m,m} \end{bmatrix} \quad \text{or} \quad A = \begin{bmatrix} a_{1,1} & \cdots & a_{1,m} \\ & \ddots & \vdots \\ & & a_{m,m} \end{bmatrix}$$

lower triangular → forward substitution

upper triangular → back substitution

2. Gaussian elimination and LU factorization

Example: $A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}$ and $b = \begin{bmatrix} 2 \\ 3 \\ 5 \\ 0 \end{bmatrix}$

Step 1:
$$\begin{bmatrix} 2 & 1 & 1 & 0 & 2 \\ -2 & 1 & 1 & 1 & -1 \\ -4 & 3 & 5 & 5 & -3 \\ -3 & 4 & 6 & 8 & -6 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} 1 & & & \\ -2 & 1 & & \\ -4 & & 1 & \\ -3 & & & 1 \end{bmatrix}}_{=: L_1} \underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}}_{= A} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix}$$

Step 2:
$$\begin{bmatrix} 2 & 1 & 1 & 0 & 2 \\ -2 & 1 & 1 & 1 & -1 \\ -4 & -3 & 2 & 2 & 0 \\ -3 & -4 & 2 & 4 & -2 \end{bmatrix} \qquad \underbrace{\begin{bmatrix} 1 & & & \\ & 1 & & \\ & -3 & 1 & \\ & -4 & & 1 \end{bmatrix}}_{=: L_2} \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & 2 & 4 \end{bmatrix}$$

Step 3:
$$\begin{bmatrix} 2 & 1 & 1 & 0 & 2 \\ +2 & 1 & 1 & 1 & -1 \\ +4 & +3 & 2 & 2 & 0 \\ +3 & +4 & +1 & 2 & -2 \end{bmatrix} \quad \text{to obtain } L \qquad \underbrace{\begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -1 & 1 \end{bmatrix}}_{=: L_3} \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & 2 & 4 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & 1 & 1 & 1 \\ & & 2 & 2 \\ & & & 2 \end{bmatrix} =: U$$

Conclusion: $x = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$; $L_3 L_2 L_1 A = U$ or $A = LU$ with $L := L_1^{-1} L_2^{-1} L_3^{-1}$

$$L_1^{-1} = \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 4 & & 1 & \\ 3 & & & 1 \end{bmatrix} \quad L_2^{-1} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & 3 & 1 & \\ & 4 & & 1 \end{bmatrix} \quad L_3^{-1} = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & 1 & 1 \end{bmatrix} \quad L = \begin{bmatrix} 1 & & & \\ 2 & 1 & & \\ 4 & 3 & 1 & \\ 3 & 4 & 1 & 1 \end{bmatrix}$$

Algorithm:

```
for i = 1, ..., m-1 do
    for j = i+1, ..., m do
        a_{j,i} ← a_{j,i} / a_{i,i}
        for k = i+1, ..., m do
            a_{j,k} ← a_{j,k} - a_{j,i} a_{i,k}
        end
        b_j ← b_j - a_{j,i} b_i
    end
end
```

```
for i = m, ..., 1 do
    x_i ← (1/a_{i,i}) (b_i - Σ_{j=i+1}^{m} a_{i,j} x_j)
end
```

Number of arithmetic operations performed

⊛ Update of b

$$\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} 2 = 2\sum_{i=1}^{m-1}(m-i) = 2\sum_{i=1}^{m-1} i = 2\,\frac{m(m-1)}{2} = m^2-m$$

⊛ Update of A, i.e., computation of L and U

$$\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\left(1+\sum_{k=i+1}^{m} 2\right) = \sum_{i=1}^{m-1}\sum_{j=i+1}^{m} 1 + 2\sum_{i=1}^{m-1}\sum_{j=i+1}^{m}\sum_{k=i+1}^{m} 1$$

$$= 2\sum_{i=1}^{m-1}(m-i)^2 + \tfrac{1}{2}m^2 - \tfrac{1}{2}m$$

$$= 2\sum_{i=1}^{m-1} i^2 + \tfrac{1}{2}m^2 - \tfrac{1}{2}m$$

$$= 2\,\frac{(m-1)m(2m-1)}{6} + \tfrac{1}{2}m^2 - \tfrac{1}{2}m$$

$$= \tfrac{2}{3}m^3 - \tfrac{1}{2}m^2 - \tfrac{1}{6}m$$

⊛ Computation of x

$$\sum_{i=1}^{m}\left(1+2(m-i)\right) = m + 2\sum_{i=1}^{m}(m-i) = m + 2\sum_{i=1}^{m-1} i = m^2$$

Total: $(m^2-m) + \left(\tfrac{2}{3}m^3 - \tfrac{1}{2}m^2 - \tfrac{1}{6}m\right) + m^2 = \boxed{\tfrac{2}{3}m^3 + \tfrac{3}{2}m^2 - \tfrac{7}{6}m}$

$\sim \tfrac{2}{3}m^3$ (i.e., $\lim_{m\to\infty} \frac{\frac{2}{3}m^3 + \frac{3}{2}m^2 - \frac{7}{6}m}{\frac{2}{3}m^3} = 1$)

Same A but with one entry modified: $A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 2 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}$

$$\begin{bmatrix} 1 & & & \\ -2 & 1 & & \\ -4 & & 1 & \\ -3 & & & 1 \end{bmatrix} \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 2 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 2 & 1 & 1 & 0 \\ & & 1 & 1 \\ & 3 & 5 & 5 \\ & 4 & 6 & 8 \end{bmatrix}$$

3. Gaussian elimination with pivoting

For numerical stability, permute two rows to maximize the absolute value of the pivot.

Same example: $A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}$ and $b = \begin{bmatrix} 2 \\ 3 \\ 5 \\ 0 \end{bmatrix}$

Step 1:

⊛ permutation of two rows

$$\begin{bmatrix} 8 & 7 & 9 & 5 & 5 \\ 4 & 3 & 3 & 1 & 3 \\ 2 & 1 & 1 & 0 & 2 \\ 6 & 7 & 9 & 8 & 0 \end{bmatrix} \quad p = \begin{bmatrix} 3 \\ 2 \\ 1 \\ 4 \end{bmatrix} \quad \underbrace{\begin{bmatrix} & & 1 & \\ & 1 & & \\ 1 & & & \\ & & & 1 \end{bmatrix}}_{=: P_1} \underbrace{\begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}}_{=A} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix}$$

⊛ Gaussian elimination

$$\begin{bmatrix} 8 & 7 & 9 & 5 & 5 \\ \frac{3}{4} & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} & -\frac{15}{4} \\ \frac{1}{4} & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} & \frac{3}{4} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \end{bmatrix} \quad \underbrace{\begin{bmatrix} 1 & & & \\ \frac{1}{2} & 1 & & \\ \frac{1}{4} & & 1 & \\ \frac{1}{4} & & & 1 \end{bmatrix}}_{=: L_1} \begin{bmatrix} 8 & 7 & 9 & 5 \\ 4 & 3 & 3 & 1 \\ 2 & 1 & 1 & 0 \\ 6 & 7 & 9 & 8 \end{bmatrix} = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{3}{4} & \frac{7}{4} & \frac{9}{4} \\ & \frac{1}{4} & -\frac{3}{4} & -\frac{5}{4} \\ & \frac{1}{2} & -\frac{1}{2} & -\frac{3}{2} \end{bmatrix}$$

Step 2:

⊛ permutation of two rows

$$\begin{bmatrix} 8 & 7 & 9 & 5 & 5 \\ \frac{3}{4} & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} & -\frac{15}{4} \\ \frac{1}{4} & -\frac{3}{4} & -\frac{5}{4} & -\frac{5}{4} & \frac{3}{4} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{3}{2} & -\frac{3}{2} & \frac{1}{2} \end{bmatrix} \quad p = \begin{bmatrix} 3 \\ 4 \\ 1 \\ 2 \end{bmatrix} \quad P_2 := \begin{bmatrix} 1 & & & \\ & & 1 & \\ & 1 & & \\ & & & 1 \end{bmatrix}$$

⊛ Gaussian elimination

$$\begin{bmatrix} 8 & 7 & 9 & 5 & 5 \\ \frac{3}{4} & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} & -\frac{15}{4} \\ \frac{1}{4} & -\frac{3}{7} & -\frac{2}{7} & \frac{4}{7} & -\frac{6}{7} \\ \frac{1}{2} & -\frac{2}{7} & -\frac{6}{7} & -\frac{2}{7} & -\frac{4}{7} \end{bmatrix} \qquad L_2 := \begin{bmatrix} 1 & & & \\ & 1 & & \\ & \frac{3}{7} & 1 & \\ & \frac{2}{7} & & 1 \end{bmatrix}$$

Step 3:

⊛ permutation of two rows

$$\begin{bmatrix} 8 & 7 & 9 & 5 & 5 \\ \frac{3}{4} & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} & -\frac{15}{4} \\ \frac{1}{2} & -\frac{2}{7} & -\frac{6}{7} & -\frac{2}{7} & -\frac{4}{7} \\ \frac{1}{4} & -\frac{3}{7} & -\frac{2}{7} & \frac{4}{7} & -\frac{6}{7} \end{bmatrix} \qquad p = \begin{bmatrix} 3 \\ 4 \\ 2 \\ 1 \end{bmatrix} \qquad P_3 := \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & & 1 \\ & & 1 & \end{bmatrix}$$

⊛ Gaussian elimination

$$\begin{bmatrix} 8 & 7 & 9 & 5 & 5 \\ \frac{3}{4} & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} & -\frac{15}{4} \\ \frac{1}{2} & -\frac{2}{7} & -\frac{6}{7} & -\frac{2}{7} & -\frac{4}{7} \\ \frac{1}{4} & -\frac{3}{7} & \frac{1}{3} & \frac{2}{3} & -\frac{2}{3} \end{bmatrix} \qquad L_3 := \begin{bmatrix} 1 & & & \\ & 1 & & \\ & & 1 & \\ & & -\frac{1}{3} & 1 \end{bmatrix}$$

Conclusion: $x = \begin{bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{bmatrix}$ and $PA = LU$ with $P := \begin{bmatrix} & & 1 & \\ & 1 & & \\ 1 & & & \\ & & & 1 \end{bmatrix}$,

$$L = \begin{bmatrix} 1 & & & \\ \frac{3}{4} & 1 & & \\ \frac{1}{2} & -\frac{2}{7} & 1 & \\ \frac{1}{4} & -\frac{3}{7} & \frac{1}{3} & 1 \end{bmatrix}, \text{ and } U = \begin{bmatrix} 8 & 7 & 9 & 5 \\ & \frac{7}{4} & \frac{9}{4} & \frac{17}{4} \\ & & -\frac{6}{7} & -\frac{2}{7} \\ & & & \frac{2}{3} \end{bmatrix}$$

Justification:

$$L_3 P_3 L_2 P_2 L_1 P_1 A = U$$

$$L_3 P_3 L_2 P_2 L_1 P_1 = L_3 \underbrace{(P_3 L_2 P_3)}_{=: L_2'} \underbrace{(P_3 P_2 L_1 P_2 P_3)}_{=: L_1'} P_3 P_2 P_1$$

$$PA = LU \text{ with } P := P_3 P_2 P_1 \text{ and } L := L_1'^{-1} L_2'^{-1} L_3^{-1}$$

Here, $L_2' = \begin{bmatrix} 1 & & & \\ & 1 & & \\ & \frac{2}{7} & 1 & \\ & \frac{3}{7} & & 1 \end{bmatrix}$

Algorithm:

$p \leftarrow \begin{bmatrix} 1 \\ \vdots \\ m \end{bmatrix}$

for $i = 1, \dots, m-1$ do

  Choose $l \in \operatorname{argmax}_{m \in \{i, \dots, m\}} |a_{m,i}|$

  if $a_{l,i} = 0$ then

    stop: A is singular

  else

    $a_{i,:} \longleftrightarrow a_{l,:}$

    $b_i \longleftrightarrow b_l$

    $p_i \longleftrightarrow p_l$

    $a_{i+1:m,i} \leftarrow \frac{a_{i+1:m,i}}{a_{i,i}}$

    for $j = i+1, \dots, n$ do

      $a_{j,i+1:m} \leftarrow a_{j,i+1:m} - a_{j,i}\, a_{i,i+1:m}$

      $b_j \leftarrow b_j - a_{j,i}\, b_i$

    end

  end

end

Number of comparisons of real numbers performed for pivoting:

$$\sum_{i=1}^{m-1} (m-i) = \sum_{i=1}^{m-1} i = \frac{1}{2} m(m-1).$$

Conclusion: in exact arithmetic, linear systems can be solved exactly within a finite number of arithmetic operations.

4. Effects of round-off errors

Computers represent real numbers in a floating-point format — typically the IEEE 754 double-precision binary floating-point format (binary64) — and use floating-point arithmetic (which is not exact arithmetic).

Backward error analysis shows that, given floating-point representations of A and b, Gaussian elimination in floating-point arithmetic returns an exact solution to a perturbed system. Therefore, the roadmap is as follows:

1. review of background material about matrix norms and condition numbers;
2. based on the reviewed material, state a perturbation theorem;
3. based on a backward error analysis of Gaussian elimination, analyze the effect of round-off errors by applying the perturbation theorem.

Additional references:

- J. W. Demmel, Applied Numerical Linear Algebra, SIAM, 1997;
- G.W. Stewart, Matrix Algorithms: Basic Decompositions, SIAM, 1998;
- N. J. Higham, Accuracy and Stability of Numerical Algorithms, second edition, SIAM, 2002;
- G. H. Golub & C. F. Van Loan, Matrix Computations, fourth edition, Johns Hopkins University Press, 2013.

4.1. Matrix norm and condition number

Proposition. If $\|\cdot\|$ is a norm on $\mathbb{R}^m$, then the function

$$\mathbb{R}^{m\times m} \to \mathbb{R} : A \mapsto \sup_{x\in\mathbb{R}^m\setminus\{0\}} \frac{\|Ax\|}{\|x\|}$$

is a norm on $\mathbb{R}^{m\times m}$ called the induced norm and denoted also by $\|\cdot\|$.

Example.

Norm on $\mathbb{R}^m$	Induced norm on $\mathbb{R}^{n\times n}$
$\|x\|_1 := \sum_{i=1}^m \lvert x_i\rvert$	$\|A\|_1 = \max_{j\in\{1,\dots,m\}} \sum_{i=1}^m \lvert a_{i,j}\rvert$
$\|x\|_2 := \sqrt{\sum_{i=1}^m x_i^2}$	$\|A\|_2 = \sigma_{max}(A) = \sqrt{\lambda_{max}(A^TA)}$
$\|x\|_\infty := \max_{i\in\{1,\dots,m\}} \lvert x_i\rvert$	$\|A\|_\infty = \max_{i\in\{1,\dots,m\}} \sum_{j=1}^m \lvert a_{i,j}\rvert$

Proposition. Every induced norm on $\mathbb{R}^{n\times n}$ is submultiplicative: for all $A, B \in \mathbb{R}^{n\times n}$,

$$\|AB\| \le \|A\|\,\|B\|.$$

Definition. The condition number of an invertible matrix $A \in \mathbb{R}^{n\times n}$ is

$$\kappa(A) := \|A\|\,\|A^{-1}\|.$$

In this chapter, only induced norms are used.

4.2. Sensitivity of linear systems

Theorem (Demmel 1997, equation (2.4))
Let $\|\cdot\|$ denote both a norm on $\mathbb{R}^n$ and the induced norm on $\mathbb{R}^{n\times n}$.

Let $A \in \mathbb{R}^{n\times n}$ be invertible, $b \in \mathbb{R}^n \setminus \{0\}$, $\tilde{A} \in \mathbb{R}^{n\times n}$, and $\tilde{b} \in \mathbb{R}^n$. If $\frac{\|A-\tilde{A}\|}{\|A\|} < \frac{1}{\kappa(A)}$, then $\tilde{A}$ is invertible. If, moreover, $x, \tilde{x} \in \mathbb{R}^n$ satisfy $Ax = b$ and $\tilde{A}\tilde{x} = \tilde{b}$, then

$$\underbrace{\frac{\|x-\tilde{x}\|}{\|x\|}}_{\text{"relative error on } x\text{"}} \leq \frac{\kappa(A)}{1-\kappa(A)\frac{\|A-\tilde{A}\|}{\|A\|}} \left(\underbrace{\frac{\|A-\tilde{A}\|}{\|A\|}}_{\text{"relative error on } A\text{"}} + \underbrace{\frac{\|b-\tilde{b}\|}{\|b\|}}_{\text{"relative error on } b\text{"}} \right).$$

4.3. Floating-point arithmetic

Computers represent real numbers in a floating-point format — typically the IEEE 754 double-precision binary floating-point format (binary64) — and use floating-point arithmetic (which is not exact arithmetic). The analysis of round-off errors is based on the existence of a positive real number $\varepsilon_{\text{machine}}$, called machine epsilon, such that:

1. for every $x \in \mathbb{R}$, there exists $\varepsilon \in [-\varepsilon_{\text{machine}}, \varepsilon_{\text{machine}}]$ such that $fl(x) = x(1+\varepsilon)$, i.e., $\frac{fl(x)-x}{x} = \varepsilon$, where $fl(x)$ is the floating-point number that is closest to x;
2. for every arithmetic operation $*$ (addition, substraction, multiplication, or division) and every pair (x, y) of floating-point numbers, there exists $\varepsilon \in [-\varepsilon_{\text{machine}}, \varepsilon_{\text{machine}}]$ such that $x \circledast y = (x * y)(1+\varepsilon)$, where $\circledast$ is $*$ in floating-point arithmetic.

Property 1 is a property of the floating-point format, while Property 2 is a property of floating-point arithmetic. Property 1 actually holds only for $x \in [-N_{\max}, N_{\max}] \setminus (-N_{\min}, N_{\min})$ but this has no practical impact. For example, with binary64, $N_{\max} = (2-2^{-52})2^{1023} \approx 10^{308}$, $N_{\min} = 2^{-1022} \approx 10^{-308}$, and $\varepsilon_{\text{machine}}$ is about $2^{-52} \approx 2.22 \cdot 10^{-16}$ or even $2^{-53} \approx 1.11 \cdot 10^{-16}$.

4.4. Gaussian elimination in floating-point arithmetic

Two sorts of errors are involved.

The first sort comes from the representation of A and b in the chosen floating-point format: A becomes $fl(A)$ and b becomes $fl(b)$, where fl is applied componentwise. Property 1 implies that $|fl(A) - A| \leq \varepsilon_{\text{machine}} |A|$ and $|fl(b) - b| \leq \varepsilon_{\text{machine}} |b|$, where the absolute value is applied componentwise.

If A is ill-conditioned, i.e., $\kappa(A)$ is large, then the perturbation of fl can be significant. If $\varepsilon_{machine}\,\kappa_\infty(A) = \frac{1}{2}$, then the upper bound given by the perturbation theorem is merely upper bounded by 2 since $\frac{\|A - fl(A)\|_\infty}{\|A\|_\infty} \leq \varepsilon_{machine}$ and $\frac{\|b - fl(b)\|_\infty}{\|b\|_\infty} \leq \varepsilon_{machine}$.

Example. Let $A = \begin{bmatrix} 1 & 2^{-54} \\ 1 & 0 \end{bmatrix}$ and $b = \begin{bmatrix} 1+2^{-54} \\ 1 \end{bmatrix}$. Then, $A^{-1}b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. With binary64, $fl(A) = \begin{bmatrix} 1 & 2^{-54} \\ 1 & 0 \end{bmatrix}$ and $fl(b) = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, and thus $fl(A)^{-1} fl(b) = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$. Since $A^{-1} = \begin{bmatrix} 0 & 1 \\ 2^{54} & -2^{54} \end{bmatrix}$, $\|A\|_\infty = 1 + 2^{-54}$, and $\|A^{-1}\|_\infty = 2^{55}$, it holds that $\kappa_\infty(A) = 2^{55} + 2$ and the upper bound given by the perturbation theorem equals 2, twice the actual relative error.

In conclusion, if A is ill-conditioned, then algorithms cannot be expected to solve the system with high accuracy.

The second sort of errors comes from floating-point arithmetic. Backward error analysis shows that the round-off errors made in the course of Gaussian elimination can be projected back on the original matrix.

Theorem (Higham 2002, Theorems 9.3 and 9.4)
Given A and b in floating-point format, Gaussian elimination yields (L, U) such that $|LU - A| \leq \frac{n\varepsilon_{machine}}{1 - n\varepsilon_{machine}} |L||U|$ (provided that $n\varepsilon_{machine} < 1$). Moreover, the computed solution $\tilde{x}$ satisfies $\tilde{A}\tilde{x} = b$ with $|A - \tilde{A}| \leq \frac{3n\varepsilon_{machine}}{1 - 3n\varepsilon_{machine}} |L||U|$ (provided that $3n\varepsilon_{machine} < 1$). Thus, $\frac{\|A - \tilde{A}\|_\infty}{\|A\|_\infty} \leq \frac{3n^3\varepsilon_{machine}}{1 - 3n\varepsilon_{machine}} \cdot \underbrace{\frac{\|U\|_{max}}{\|A\|_{max}}}_{\text{"growth factor"}}$.

$\|A\|_{max} := \max_{i,j=1,\dots,n} |a_{ij}|$

Proof of the "Thus" statement. It holds that

$$\|A - \tilde{A}\|_\infty \leq \frac{3n\varepsilon_{machine}}{1 - 3n\varepsilon_{machine}} \| |L||U| \|_\infty .$$

Moreover,

$$\| |L||U| \|_\infty \leq \| |L| \|_\infty \| |U| \|_\infty = \|L\|_\infty \|U\|_\infty \leq n \|U\|_\infty \leq n^2 \|U\|_{max} .$$

The result follows from the inequality $\|A\|_{max} \leq \|A\|_\infty$. ∎

Theorem (Stewart 1998, Theorem 4.12)
Let $PA = LU$ be an LU factorization obtained by Gaussian elimination with pivoting in exact arithmetic. Then, $\|U\|_{max} \leq 2^{n-1} \|A\|_{max}$. Moreover, equality is reached for

$$A = \begin{bmatrix} 1 & & & 1 \\ -1 & \ddots & & \vdots \\ \vdots & \ddots & \ddots & 1 \\ -1 & \cdots & -1 & 1 \end{bmatrix}.$$

In practice, the growth factor is often observed to grow as $n^{\frac{2}{3}}$. For tridiagonal matrices, it is upper bounded by 2 (Stewart 1998, p. 240).

In conclusion, backward error analysis reduces the analysis of round-off errors to the analysis of the sensitivity of the system: these errors can be estimated by combining the last two theorems with the perturbation theorem. However, the estimation is generally very pessimistic. Thus, other methods have been proposed to estimate round-off errors in practice.

4.5. Least-squares problems

Let $m, n \in \mathbb{N} \setminus \{0, 1\}$ such that $m \geq n$. Let $A \in \mathbb{R}^{m \times n}$ such that $\operatorname{rank} A = n$. Let $b \in \mathbb{R}^m$. If $b \notin \operatorname{im} A$, there is no $x \in \mathbb{R}^n$ such that $Ax = b$. However, the optimization problem $\min_{x \in \mathbb{R}^n} \|Ax - b\|$ can still be considered. It is called a least-squares problem if the norm $\|\cdot\|$ is the 2-norm $\|\cdot\|_2$. Considering the 2-norm is convenient because the 2-norm is induced by the usual inner product on $\mathbb{R}^n$. Furthermore, the choice of the 2-norm can be defended by geometric and statistical reasons. In the rest of this section, only the 2-norm is considered.

Solution: $x \in \mathbb{R}^n$ minimizes the function $\mathbb{R}^n \to \mathbb{R} : z \mapsto \|Az - b\|_2$ if and only if $Ax = y$ and $y \in \operatorname{im} A$ minimizes the function $\operatorname{im} A \to \mathbb{R} : z \mapsto \|z - b\|_2$, i.e., y is the orthogonal projection of b onto $\operatorname{im} A$. Thus, it suffices to find $x \in \mathbb{R}^n$ such that Ax is the orthogonal projection of b onto $\operatorname{im} A$, i.e., $b - Ax \perp \operatorname{im} A$ or $\langle b - Ax, Az \rangle = 0$ for all $z \in \mathbb{R}^n$ or $z^T A^T (b - Ax) = 0$ for all $z \in \mathbb{R}^n$ or $A^T(b - Ax) =$

This yields the so-called normal equations:

$$A^TA\,x = A^Tb.$$

Since $\operatorname{rank} A^TA = \operatorname{rank} A = n$, this system has a unique solution.

Example. Let $A = \begin{bmatrix} 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$ and $b = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$. Then, $b \notin \operatorname{im} A$. We compute

$$A^TA = \begin{bmatrix} 3 & 6 \\ 6 & 14 \end{bmatrix} \quad \text{and} \quad A^Tb = \begin{bmatrix} 4 \\ 8 \end{bmatrix}.$$

Thus, the least-squares solution is $\begin{bmatrix} 4/3 \\ 0 \end{bmatrix}$.