

# Computergestützte Mathematik zur Linearen Algebra

## Gleitkommaarithmetik

Achim Schädle

7. Dezember 2022

# Instabilitäten bei Gauß-Elimination

Aufgabe: Berechne LR-Zerlegung von

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

LR-Zerlegung von  $A$  ohne Zeilenvertauschung existiert nicht (Division durch 0)

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

Pivotelement  $10^{-20} \neq 0$ , LR-Zerlegung von  $A$  existiert:

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

Pythont liefert jedoch bei der Multiplikation  $L \cdot R \neq A$

# Instabilitäten bei Gauß-Elimination

Aufgabe: Berechne LR-Zerlegung von

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

LR-Zerlegung von  $A$  ohne Zeilenvertauschung existiert nicht (Division durch 0)

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

Pivotelement  $10^{-20} \neq 0$ , LR-Zerlegung von  $A$  existiert:

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

PYTHON liefert jedoch bei der Multiplikation  $L \cdot R \neq A$

# Instabilitäten bei Gauß-Elimination

Aufgabe: Berechne LR-Zerlegung von

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

LR-Zerlegung von  $A$  ohne Zeilenvertauschung existiert nicht (Division durch 0)

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

Pivotelement  $10^{-20} \neq 0$ , LR-Zerlegung von  $A$  existiert:

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

PYTHON liefert jedoch bei der Multiplikation  $L \cdot R \neq A$

# Instabilitäten bei Gauß-Elimination

Aufgabe: Berechne LR-Zerlegung von

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

LR-Zerlegung von  $A$  ohne Zeilenvertauschung existiert nicht (Division durch 0)

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

Pivotelement  $10^{-20} \neq 0$ , LR-Zerlegung von  $A$  existiert:

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

PYTHON liefert jedoch bei der Multiplikation  $L \cdot R \neq A$

# Instabilitäten bei Gauß-Elimination

Aufgabe: Berechne LR-Zerlegung von

$$A = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}$$

LR-Zerlegung von  $A$  ohne Zeilenvertauschung existiert nicht (Division durch 0)

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

Pivotelement  $10^{-20} \neq 0$ , LR-Zerlegung von  $A$  existiert:

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

PYTHON liefert jedoch bei der Multiplikation  $L \cdot R \neq A$

# Gleitkommaarithmetik

Wie lässt sich dieser Fehler erklären?

- Wie werden Zahlen im Rechner dargestellt?
- Welche Fehler können bei Grundrechenarten passieren?
- Wie kann man Instabilitäten im Gauß-Algorithmus vermeiden?

# Gleitkommaarithmetik

Wie lässt sich dieser Fehler erklären?

- Wie werden Zahlen im Rechner dargestellt?
- Welche Fehler können bei Grundrechenarten passieren?
- Wie kann man Instabilitäten im Gauß-Algorithmus vermeiden?

# Gleitkommazahlen

Eine Gleitkommazahl (engl. *floating point number*) zu einer gegebenen Basis  $b$  ist eine Zahl der Form

$$x = s \cdot m \cdot b^e,$$

wobei

- $s \in \{-1, 1\}$  das *Signum*
- $m \in [1, b)$  die *Mantisse* und
- $e \in \mathbb{Z}$  der *Exponent*

heißen.

Jede reelle Zahl lässt sich auf diese Weise eindeutig darstellen.

# Gleitkommadarstellung reeller Zahlen

Sei  $\bar{m}$  auf  $\ell$  Ziffern gerundete Mantisse von  $x = \pm m \cdot b^e$

$$\text{fl}(x) := \pm \bar{m} \cdot b^e$$

Beispiel:  $\ell = 8, b = 10, x = \pi = 3.141592653\dots$

$$\text{fl}(\pi) = 3.1415927 \cdot 10^0$$

Maschinengenauigkeit: Kleinste positive Zahl  $\text{eps}$ , so dass

$$\text{fl}(1 + \text{eps}) > 1.$$

- Dezimalsystem  $\text{eps} = 5 \cdot 10^{-\ell}$
- Binärsystem (Basis 2)  $\text{eps} = 2^{-\ell}$

# Gleitkommadarstellung reeller Zahlen

Sei  $\bar{m}$  auf  $\ell$  Ziffern gerundete Mantisse von  $x = \pm m \cdot b^e$

$$\text{fl}(x) := \pm \bar{m} \cdot b^e$$

Beispiel:  $\ell = 8, b = 10, x = \pi = 3.141592653\dots$

$$\text{fl}(\pi) = 3.1415927 \cdot 10^0$$

Maschinengenauigkeit: Kleinste positive Zahl  $\text{eps}$ , so dass

$$\text{fl}(1 + \text{eps}) > 1.$$

- **Dezimalsystem**     $\text{eps} = 5 \cdot 10^{-\ell}$
- **Binärsystem (Basis 2)**     $\text{eps} = 2^{-\ell}$

# Beispiele

## Dezimaldarstellung ( $b = 10$ )

$$x = s \cdot m \cdot 10^e, \quad m \in [1, 10)$$

$$x = 1 \quad \Rightarrow \quad s = 1, \ e = 0, \ m = 1$$

$$x = -10 \quad \Rightarrow \quad s = -1, \ e = 1, \ m = 1$$

$$x = -0,125 \quad \Rightarrow \quad s = -1, \ e = -1, \ m = 1,25$$

# Beispiele

## Dezimaldarstellung ( $b = 10$ )

$$x = s \cdot m \cdot 10^e, \quad m \in [1, 10)$$

$$x = 1 \quad \Rightarrow \quad s = 1, \ e = 0, \ m = 1$$

$$x = -10 \quad \Rightarrow \quad s = -1, \ e = 1, \ m = 1$$

$$x = -0,125 \quad \Rightarrow \quad s = -1, \ e = -1, \ m = 1,25$$

# Beispiele

## Binärdarstellung ( $b = 2$ )

$$x = s \cdot m \cdot 2^e, \quad m \in [1, 2)$$

$$x = 1 \quad \Rightarrow \quad s = 1, \ e = 0, \ m = 1$$

$$x = -10 \quad \Rightarrow \quad s = -1, \ e = 3, \ m = \frac{5}{4}$$

$$x = -0,125 \quad \Rightarrow \quad s = -1, \ e = -3, \ m = 1$$

# Beispiele

## Binärdarstellung ( $b = 2$ )

$$x = s \cdot m \cdot 2^e, \quad m \in [1, 2)$$

$$x = 1 \quad \Rightarrow \quad s = 1, \ e = 0, \ m = 1$$

$$x = -10 \quad \Rightarrow \quad s = -1, \ e = 3, \ m = \frac{5}{4}$$

$$x = -0,125 \quad \Rightarrow \quad s = -1, \ e = -3, \ m = 1$$

# Zahlendarstellung im Rechner

- Der Rechner hat nur endlich viel Speicher
- $\Rightarrow m$  und  $e$  können nur endlich viele Werte annehmen.
- Computer kann nur Nullen und Einsen speichern
- $\Rightarrow$  stelle  $m$  und  $e$  binär dar (d.h.  $b = 2$ ).
- Verwende  $p+1$  Stellen für die Mantisse,  $r$  Stellen für den Exponenten.

# Zahlendarstellung im Rechner

- Der Rechner hat nur endlich viel Speicher
- $\Rightarrow m$  und  $e$  können nur endlich viele Werte annehmen.
- Computer kann nur Nullen und Einsen speichern
- $\Rightarrow$  stelle  $m$  und  $e$  binär dar (d.h.  $b = 2$ ).
- Verwende  $p+1$  Stellen für die Mantisse,  $r$  Stellen für den Exponenten.
- Erhalte endliche Gleitkommadarstellung

$$\text{fl}(x) = s \cdot m \cdot b^e, \quad s = (-1)^S, \quad m = 1 + \sum_{k=1}^p M_{p-k} 2^{-k}, \quad e = \sum_{l=0}^{r-1} E_l 2^l - 2^{r-1} + 1,$$

wobei  $S, M_i, E_l \in \{0, 1\}$ ,  $i = 0, \dots, p-1$ ,  $l = 0, \dots, r-1$ .

# Zahlendarstellung im Rechner

- Der Rechner hat nur endlich viel Speicher
- $\Rightarrow m$  und  $e$  können nur endlich viele Werte annehmen.
- Computer kann nur Nullen und Einsen speichern
- $\Rightarrow$  stelle  $m$  und  $e$  binär dar (d.h.  $b = 2$ ).
- Verwende  $p+1$  Stellen für die Mantisse,  $r$  Stellen für den Exponenten.
- Erhalte endliche Gleitkommadarstellung

$$\text{fl}(x) = s \cdot m \cdot b^e, \quad s = (-1)^S, \quad m = 1 + \sum_{k=1}^p M_{p-k} 2^{-k}, \quad e = \sum_{l=0}^{r-1} E_l 2^l - \underbrace{2^{r-1} + 1}_{\text{Bias}},$$

wobei  $S, M_i, E_l \in \{0, 1\}$ ,  $i = 0, \dots, p-1$ ,  $l = 0, \dots, r-1$ .

# Zahlendarstellung im Rechner, Anordnung

$$\text{fl}(x) = s \cdot m \cdot b^e,$$

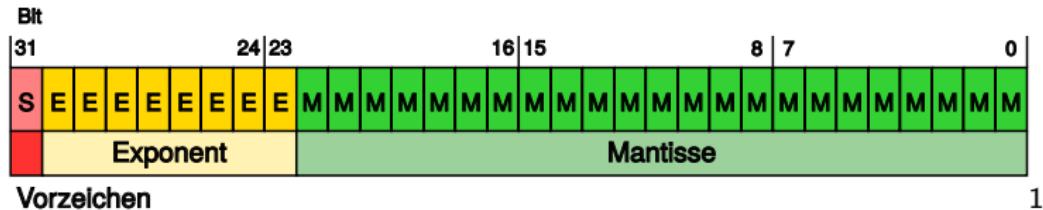
$$\text{mit } s = (-1)^S, \quad m = 1 + \sum_{k=1}^p M_{p-k} 2^{-k}, \quad e = \sum_{l=0}^{r-1} E_l 2^l - 2^{r-1} + 1,$$

# Zahlendarstellung im Rechner, Anordnung

$$\text{fl}(x) = s \cdot m \cdot b^e,$$

$$\text{mit } s = (-1)^S, \quad m = 1 + \sum_{k=1}^p M_{p-k} 2^{-k}, \quad e = \sum_{l=0}^{r-1} E_l 2^l - 2^{r-1} + 1,$$

Anordnung im Speicher bei einfacher Genauigkeit, d.h.  $p = 23$ ,  $r = 8$ :



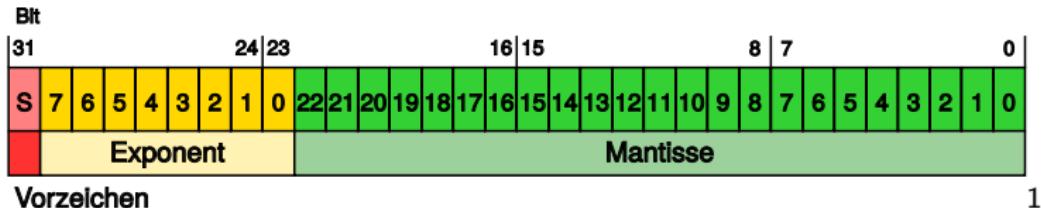
<sup>1</sup>Quelle: Wikipedia

## Zahlendarstellung im Rechner, Anordnung

$$\text{fl}(x) = s \cdot m \cdot b^e,$$

$$\text{mit } s = (-1)^S, \quad m = 1 + \sum_{k=1}^p M_{p-k} 2^{-k}, \quad e = \sum_{l=0}^{r-1} E_l 2^l - 2^{r-1} + 1,$$

Anordnung im Speicher bei einfacher Genauigkeit, d.h.  $p = 23$ ,  $r = 8$ :



- Jedes farbige Feld enthält entweder eine Null oder eine Eins.
  - Das Vorzeichenbit  $S$  steht für „+“ (0) oder „-“ (1).
  - Das vorderste Bit (7) des Exponenten bestimmt, ob er positiv (1) oder nicht-positiv (0) ist.
  - Das vorderste Bit (22) der Mantisse steht für  $\frac{1}{2}$ , das nächste (21) für  $\frac{1}{4}$ , etc.

# Zahlendarstellung im Rechner, spezielle Werte

- $\pm 0: e = -2^{r-1} + 1, m = 1$  (eigentlich nicht darstellbar)

Blt																									
31																									
S	0 0																								
Vorzeichen	Exponent Mantisse																								

- $\pm \infty: e = 2^{r-1}, m = 1$

Blt																									
31																									
S	1 1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0																								
Vorzeichen	Exponent Mantisse																								

# Beispiele

$$x = 1 = 1 \cdot 1 \cdot 2^0 \quad s = 1, \ e = 0, \ m = 1$$

$$s = 1 = (-1)^0$$

$$e = 0 = \sum_{l=0}^{r-2} \underbrace{1 \cdot 2^l}_{=2^{r-1}-1} + 0 \cdot 2^{r-1} - 2^{r-1} + 1$$

$$m = 1 = 1 + \sum_{k=1}^p 0 \cdot 2^{-k}$$

## Beispiele

$$x = 1 = 1 \cdot 1 \cdot 2^0 \quad s = 1, e = 0, m = 1$$

$$s = 1 = (-1)^0$$

$$e = 0 = \sum_{l=0}^{r-2} 1 \cdot 2^l + 0 \cdot 2^{r-1} - 2^{r-1} + 1$$

$\underbrace{\hspace{10em}}_{=2^{r-1}-1}$

$$m = 1 = 1 + \sum_{k=1}^p 0 \cdot 2^{-k}$$

# Beispiele

$$x = 1 = 1 \cdot 1 \cdot 2^0 \quad s = 1, \ e = 0, \ m = 1$$

$$s = 1 = (-1)^0$$

$$e = 0 = \underbrace{\sum_{l=0}^{r-2} 1 \cdot 2^l + 0 \cdot 2^{r-1}}_{=2^{r-1}-1} - 2^{r-1} + 1$$

$$m = 1 = 1 + \sum_{k=1}^p 0 \cdot 2^{-k}$$

# Beispiele

$$x = 1 = 1 \cdot 1 \cdot 2^0 \quad s = 1, \ e = 0, \ m = 1$$

$$s = 1 = (-1)^0$$

$$e = 0 = \sum_{l=0}^{r-2} \underbrace{1 \cdot 2^l + 0 \cdot 2^{r-1} - 2^{r-1} + 1}_{=2^{r-1}-1}$$

$$m = 1 = 1 + \sum_{k=1}^p 0 \cdot 2^{-k}$$

Blt												
31												0
0	0	1	1	1	1	1	1	0	0	0	0	0
<b>Exponent</b>											<b>Mantisse</b>	
Vorzeichen												

# Beispiele

$$x = -10 = -1 \cdot \frac{5}{4} \cdot 2^3 \quad s = -1, \ e = 3, \ m = \frac{5}{4}$$

$$s = -1 = (-1)^1$$

$$e = 3 = 0 \cdot 2^0 + 1 \cdot 2^2 + \sum_{l=0}^{r-2} 0 \cdot 2^l + 1 \cdot 2^{r-1} - 2^{r-1} + 1$$

$$m = \frac{5}{4} = 1 + 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + \sum_{k=3}^p 0 \cdot 2^{-k}$$

# Beispiele

$$x = -10 = -1 \cdot \frac{5}{4} \cdot 2^3 \quad s = -1, \ e = 3, \ m = \frac{5}{4}$$

$$s = -1 = (-1)^1$$

$$e = 3 = 0 \cdot 2^0 + 1 \cdot 2^2 + \sum_{l=0}^{r-2} 0 \cdot 2^l + 1 \cdot 2^{r-1} - 2^{r-1} + 1$$

$$m = \frac{5}{4} = 1 + 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + \sum_{k=3}^p 0 \cdot 2^{-k}$$

# Beispiele

$$x = -10 = -1 \cdot \frac{5}{4} \cdot 2^3 \quad s = -1, \ e = 3, \ m = \frac{5}{4}$$

$$s = -1 = (-1)^1$$

$$e = 3 = 0 \cdot 2^0 + 1 \cdot 2^2 + \sum_{l=0}^{r-2} 0 \cdot 2^l + 1 \cdot 2^{r-1} - 2^{r-1} + 1$$

$$m = \frac{5}{4} = 1 + 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + \sum_{k=3}^p 0 \cdot 2^{-k}$$

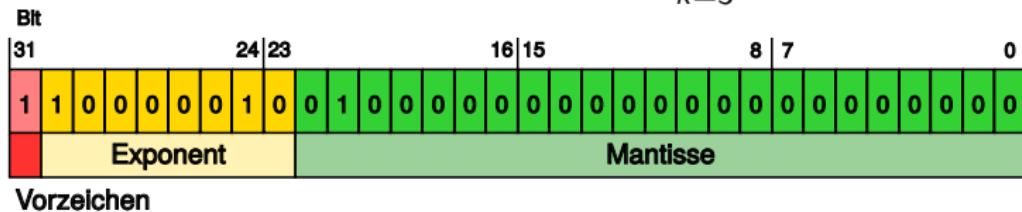
# Beispiele

$$x = -10 = -1 \cdot \frac{5}{4} \cdot 2^3 \quad s = -1, e = 3, m = \frac{5}{4}$$

$$s = -1 = (-1)^1$$

$$e = 3 = 0 \cdot 2^0 + 1 \cdot 2^2 + \sum_{l=0}^{r-2} 0 \cdot 2^l + 1 \cdot 2^{r-1} - 2^{r-1} + 1$$

$$m = \frac{5}{4} = 1 + 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} + \sum_{k=3}^p 0 \cdot 2^{-k}$$



## Beispiele

$$x = -0,125 = -\frac{1}{8} = -1 \cdot 1 \cdot 2^{-3} \quad s = -1, \ e = -3, \ m = 1$$
$$s = -1 = (-1)^1$$

$$e = -3 = 64 + 32 + 16 + 8 + 4 - 128 + 1 = \sum_{l=0}^2 0 \cdot 2^l + \sum_{l=3}^6 1 \cdot 2^l + 0 \cdot 2^7 - 2^{r-1} + 1$$
$$m = 1 = 1 + \sum_{k=1}^p 0 \cdot 2^{-k}$$

## Beispiele

$$x = -0,125 = -\frac{1}{8} = -1 \cdot 1 \cdot 2^{-3} \quad s = -1, \ e = -3, \ m = 1$$
$$s = -1 = (-1)^1$$

$$e = -3 = 64 + 32 + 16 + 8 + 4 - 128 + 1 = \sum_{l=0}^2 0 \cdot 2^l + \sum_{l=3}^6 1 \cdot 2^l + 0 \cdot 2^7 - 2^{r-1} + 1$$
$$m = 1 = 1 + \sum_{k=1}^p 0 \cdot 2^{-k}$$

## Beispiele

$$x = -0,125 = -\frac{1}{8} = -1 \cdot 1 \cdot 2^{-3} \quad s = -1, \ e = -3, \ m = 1$$
$$s = -1 = (-1)^1$$

$$e = -3 = 64 + 32 + 16 + 8 + 4 - 128 + 1 = \sum_{l=0}^2 0 \cdot 2^l + \sum_{l=3}^6 1 \cdot 2^l + 0 \cdot 2^7 - 2^{r-1} + 1$$
$$m = 1 = 1 + \sum_{k=1}^p 0 \cdot 2^{-k}$$

# Beispiele

$$x = -0,125 = -\frac{1}{8} = -1 \cdot 1 \cdot 2^{-3} \quad s = -1, \ e = -3, \ m = 1$$

$$s = -1 = (-1)^1$$

$$e = -3 = 64 + 32 + 16 + 8 + 4 - 128 + 1 = \sum_{l=0}^2 0 \cdot 2^l + \sum_{l=3}^6 1 \cdot 2^l + 0 \cdot 2^7 - 2^{r-1} + 1$$

$$m = 1 = 1 + \sum_{k=1}^p 0 \cdot 2^{-k}$$

Bit	31	24	23	16	15	8	7	0
1	0	1	1	1	1	0	0	0
Exponent				Mantisse				
Vorzeichen								

# Beispiele – Rundung!

$$x = 0,1 = 1 \cdot \frac{8}{5} \cdot 2^{-4} \quad s = 1, \ e = -4, \ m = \frac{8}{5}$$

$$s = 1 = (-1)^0$$

$$e = -4 = 64 + 32 + 16 + 8 + 2 + 1 - 128 + 1$$

$$\begin{aligned}m = \frac{8}{5} &= 1 + \frac{3}{5} \approx 1 + \frac{1}{2} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{512} + \frac{1}{4096} + \frac{1}{8192} + \frac{1}{16384} \\&\quad + \frac{1}{32768} + \frac{1}{65536} + \frac{1}{131072} + \frac{1}{262144} + \frac{1}{524288}\end{aligned}$$

# Beispiele – Rundung!

$$x = 0,1 = 1 \cdot \frac{8}{5} \cdot 2^{-4} \quad s = 1, \ e = -4, \ m = \frac{8}{5}$$

$$s = 1 = (-1)^0$$

$$e = -4 = 64 + 32 + 16 + 8 + 2 + 1 - 128 + 1$$

$$\begin{aligned}m = \frac{8}{5} &= 1 + \frac{3}{5} \approx 1 + \frac{1}{2} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{512} + \frac{1}{4096} + \frac{1}{8192} + \frac{1}{16384} \\&\quad + \frac{1}{32768} + \frac{1}{65536} + \frac{1}{131072} + \frac{1}{262144} + \frac{1}{524288}\end{aligned}$$

# Beispiele – Rundung!

$$x = 0,1 = 1 \cdot \frac{8}{5} \cdot 2^{-4} \quad s = 1, \ e = -4, \ m = \frac{8}{5}$$

$$s = 1 = (-1)^0$$

$$e = -4 = 64 + 32 + 16 + 8 + 2 + 1 - 128 + 1$$

$$\begin{aligned}m = \frac{8}{5} &= 1 + \frac{3}{5} \approx 1 + \frac{1}{2} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{512} + \frac{1}{4096} + \frac{1}{8192} + \frac{1}{65536} \\&\quad + \frac{1}{131072} + \frac{1}{1048576} + \frac{1}{2097152} + \frac{1}{8388608}\end{aligned}$$

# Beispiele – Rundung!

$$x = 0,1 = 1 \cdot \frac{8}{5} \cdot 2^{-4} \quad s = 1, \ e = -4, \ m = \frac{8}{5}$$
$$s = 1 = (-1)^0$$

$$e = -4 = 64 + 32 + 16 + 8 + 2 + 1 - 128 + 1$$

$$m = \frac{8}{5} = 1 + \frac{3}{5} \approx 1 + \frac{1}{2} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{512} + \frac{1}{4096} + \frac{1}{8192} + \frac{1}{65536}$$
$$+ \frac{1}{131072} + \frac{1}{1048576} + \frac{1}{2097152} + \frac{1}{8388608}$$

Bit	31	24	23	16	15	8	7	0	
	0	0	1	1	1	0	1	1	
Vorzeichen	Exponent				Mantisse				

# Beispiele – Rundung!

$$x = 0,1 = 1 \cdot \frac{8}{5} \cdot 2^{-4} \quad s = 1, \ e = -4, \ m = \frac{8}{5}$$

$$s = 1 = (-1)^0$$

$$e = -4 = 64 + 32 + 16 + 8 + 2 + 1 - 128 + 1$$

$$\begin{aligned}m = \frac{8}{5} &= 1 + \frac{3}{5} \approx 1 + \frac{1}{2} + \frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{512} + \frac{1}{4096} + \frac{1}{8192} + \frac{1}{65536} \\&\quad + \frac{1}{131072} + \frac{1}{1048576} + \frac{1}{2097152} + \frac{1}{8388608}\end{aligned}$$

fl(0,1)  $\neq$  0,1 ist nicht exakt darstellbar, denn

$$\frac{8}{5} = \sum_{k \geq 0} \frac{1}{2^{4k}} + \frac{1}{2^{4k+1}}$$

ist eine periodische Binärzahl.

# Beispiele – Rundung!

$$x = 1 + 10^{-8} = 1,00000001 = 1 \cdot 1,00000001 \cdot 2^0 \quad s = 1, \ e = 0, \ m = 1,000000$$

$$s = 1 = (-1)^0$$

$$e = 0 = 64 + 32 + 16 + 8 + 4 + 2 + 1 - 128 + 1$$

$$m = \frac{100000001}{100000000} = 1 + \frac{1}{100000000} \approx 1, \text{ da } 1 + 2^{-23} \approx 1,2 \cdot 10^{-7}$$

# Beispiele – Rundung!

$$x = 1 + 10^{-8} = 1,00000001 = 1 \cdot 1,00000001 \cdot 2^0 \quad s = 1, \quad e = 0, \quad m = 1,000000$$

$$s = 1 = (-1)^0$$

$$e = 0 = 64 + 32 + 16 + 8 + 4 + 2 + 1 - 128 + 1$$

$$m = \frac{100000001}{100000000} = 1 + \frac{1}{100000000} \approx 1, \quad \text{da } 1 + 2^{-23} \approx 1,2 \cdot 10^{-7}$$

# Beispiele – Rundung!

$$x = 1 + 10^{-8} = 1,00000001 = 1 \cdot 1,00000001 \cdot 2^0 \quad s = 1, \ e = 0, \ m = 1,000000$$

$$s = 1 = (-1)^0$$

$$e = 0 = 64 + 32 + 16 + 8 + 4 + 2 + 1 - 128 + 1$$

$$m = \frac{100000001}{100000000} = 1 + \frac{1}{100000000} \approx 1, \text{ da } 1 + 2^{-23} \approx 1,2 \cdot 10^{-7}$$

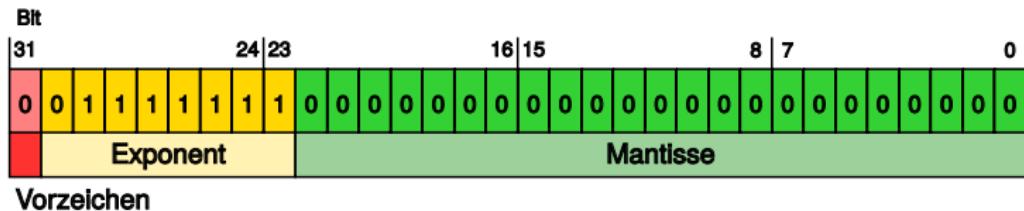
# Beispiele – Rundung!

$$x = 1 + 10^{-8} = 1,00000001 = 1 \cdot 1,00000001 \cdot 2^0 \quad s = 1, \quad e = 0, \quad m = 1,000000$$

$$s = 1 = (-1)^0$$

$$e = 0 = 64 + 32 + 16 + 8 + 4 + 2 + 1 - 128 + 1$$

$$m = \frac{100000001}{100000000} = 1 + \frac{1}{100000000} \approx 1, \quad \text{da } 1 + 2^{-23} \approx 1,2 \cdot 10^{-7}$$



Wieder  $\text{fl}(x) \neq x$ .

# Maschinengenauigkeit

Maschinengenauigkeit: Kleinste positive Zahl  $\text{eps}$ , so dass

$$\text{fl}(1 + \text{eps}) > 1.$$

Für einfache Genauigkeit:  $\text{eps} = 2^{-23} \approx 1,2 \cdot 10^{-7}$ .

# Maschinengenauigkeit

Maschinengenauigkeit: Kleinste positive Zahl  $\text{eps}$ , so dass

$$\text{fl}(1 + \text{eps}) > 1.$$

Für einfache Genauigkeit:  $\text{eps} = 2^{-23} \approx 1,2 \cdot 10^{-7}$ .

Bemerkung:  $\text{eps}$  ist der Abstand zwischen je zwei benachbarten Mantissen.

# IEEE Standard 754 - 1985/2008: Gleitkommazahlen

Genauigkeit	Bit	Byte	Vorzeichen $s$	Exponent $e$	Mantisse $m$
single	32	4	1(31)	8(30 – 23)	23(22 – 0)
double	64	8	1(63)	11(62 – 52)	52(51 – 0)
quadruple	128	16	1(127)	15(126 – 113)	113(112 – 0)
octuple	256	32	1(255)	19(254 – 237)	237(236 – 0)

Normalisierte Zahlendarstellung (s.o.):

$$s \cdot 1.M \cdot 2^{e-bias}, \quad bias = 127 \text{ bzw. } 1023 \text{ bzw. } 16383 \text{ bzw. } 262143$$

Damit ergeben sich die Maschinengenauigkeiten

- single:  $\text{eps} = 2^{-23} \approx 1,2 \cdot 10^{-7}$
- double:  $\text{eps} = 2^{-52} \approx 2,2 \cdot 10^{-16}$
- quadruple:  $\text{eps} = 2^{-113} \approx 9,6 \cdot 10^{-35}$
- octuple:  $\text{eps} = 2^{-237} \approx 4,5 \cdot 10^{-72}$

In PYTHON wird standardmäßig doppelt genau gerechnet.

# Relativer Fehler der endlichen Gleitkommadarstellung

## Satz

Für jedes  $0 \neq x \in [-2^{r-1}, 2^{r-1}] \setminus [-2^{-r+1}, 2^{-r+1}]$  ist

$$\frac{|\text{fl}(x) - x|}{|x|} \leq \text{eps},$$

das heißt der **relative Fehler** ist beschränkt durch  $\text{eps}$ .

**Beweis:**  $\mathcal{M} = \{1 + n \cdot 2^{-p} \mid n \in \{0, \dots, 2^p\}\} =$  Menge der Mantissen.

Sei  $\text{fl}(x) = \overline{m} \cdot 2^{\overline{e}}$  und  $x = m \cdot 2^{\overline{e}}$ , wobei  $m, \overline{m} \in \mathcal{M}$ . Sei  $\Delta m := m - \overline{m}$ .

Nach Bemerkung ist  $|\Delta m| \leq \text{eps}$ . Damit

$$\frac{|\text{fl}(x) - x|}{|x|} = \frac{|\overline{m} - m| \cdot 2^{\overline{e}}}{|m| \cdot 2^{\overline{e}}} = \frac{|\Delta m|}{|m|} \leq \text{eps},$$

da  $m \geq 1$ .

Schreibe daher  $\text{fl}(x) = x(1 + \varepsilon)$  mit  $|\varepsilon| \leq \text{eps}$ .



# Kondition eines Problems

Ein Problem sei durch (Auswertung einer) Abbildung

$$F : U \rightarrow V \quad x \mapsto F(x)$$

beschrieben, wobei  $U$  und  $V$  normierte Räume sind und  $F$  die Problemstellung. Die Problemstellung könnte etwas sein:

- ein Polynom auszuwerten,
- die Wurzeln einer quadratischen Gleichung zu bestimmen,
- die Lösung von  $Ax = b$  zu berechnen oder
- die Lösung eines Eigenwertproblems  $Ax = \lambda x$  ist.

**Frage:** Wie wirken sich Störungen in den Daten (Koeffizienten des Polynoms, Einträgen von  $A$  und  $b$ , ...) auf das Resultat  $F(x)$  aus?

# Kondition eines Problems

Sei  $U = \mathbb{R}^n$  und  $V = \mathbb{R}$ , dann ist:

**Definition:** Die **Kondition**  $\kappa$  von  $F$  ist die kleinste Zahl, so dass

$$\frac{|\hat{x}_i - x_i|}{|x_i|} \leq \text{eps}, \quad \forall i \implies \frac{|F(\hat{x}) - F(x)|}{|F(x)|} \leq \kappa \cdot \text{eps}.$$

Das Problem heißt **gut konditioniert**, falls  $\kappa$  nicht zu groß ist (ideal  $\kappa = 1$ ) und anderenfalls **schlecht konditioniert**.

Die Kondition kann näherungsweise durch Linearisierung mithilfe der Ableitung bestimmt werden.

# Kondition der Grundrechenarten in Gleitkommaarithmetik

## Multiplikation zweier reeller Zahlen

Sei  $F(x_1, x_2) = x_1 \cdot x_2$ . Für die gestörten Werte

$$\hat{x}_1 = x_1(1 + \varepsilon_1), \quad \hat{x}_2 = x_2(1 + \varepsilon_2), \quad |\varepsilon_i| \leq \text{eps}$$

erhalten wir für  $\frac{|F(\hat{x}) - F(x)|}{|F(x)|}$ :

$$\frac{\hat{x}_1 \hat{x}_2 - x_1 x_2}{x_1 x_2} = (1 + \varepsilon_1)(1 + \varepsilon_2) - 1 = \varepsilon_1 + \varepsilon_2 + \varepsilon_1 \varepsilon_2.$$

Da  $\text{eps} < 1$ , ist  $|\varepsilon_1 \varepsilon_2| \leq \text{eps}^2 < \text{eps}$ .

$$\left| \frac{\hat{x}_1 \hat{x}_2 - x_1 x_2}{x_1 x_2} \right| \leq 3\text{eps}.$$

Also  $\kappa(F) = 3$ , die Multiplikation ist **gut** konditioniert!

# Kondition der Grundrechenarten in Gleitkommaarithmetik

## Subtraktion zweier reeller Zahlen

Für  $F(x_1, x_2) = x_1 - x_2$  erhalten wir

$$\begin{aligned}\left| \frac{(\hat{x}_1 - \hat{x}_2) - (x_1 - x_2)}{x_1 - x_2} \right| &= \left| \frac{x_1 \varepsilon_1 - x_2 \varepsilon_2}{x_1 - x_2} \right| \\ &\leq \frac{|x_1| + |x_2|}{|x_1 - x_2|} \text{eps} =: \kappa \text{eps}.\end{aligned}$$

- Mit  $\text{sign } x_1 = -\text{sign } x_2$  (**Addition**) ist  $\kappa(F) = 1$ .  
Die Addition ist **gut** konditioniert.
- Mit  $\text{sign } x_1 = \text{sign } x_2$  (**Subtraktion**) und  $x_1 \approx x_2$  ist  $\kappa(F) \gg 1$  sehr groß.  
Die Subtraktion zweier etwa gleich großer Zahlen ist **sehr schlecht** konditioniert (**Auslöschung**).

# Gauss-Elimination

In double precision mit  $\text{eps} = 10^{-16}$

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

Pivotelement  $10^{-20} \neq 0$ , LR-Zerlegung von  $A$  existiert:

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

Jedoch wird das Ergebnis auf double precision gerundet!

$$\text{fl}(L) = \tilde{L} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad \text{fl}(R) = \tilde{R} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}$$

und damit erhalten wir numerisch (vgl. PYTHON)

$$\tilde{L}\tilde{R} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix} \neq A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

# Gauss-Elimination

In double precision mit  $\text{eps} = 10^{-16}$

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

Pivotelement  $10^{-20} \neq 0$ , LR-Zerlegung von  $A$  existiert:

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

Jedoch wird das Ergebnis auf double precision gerundet!

$$\text{fl}(L) = \tilde{L} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad \text{fl}(R) = \tilde{R} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}$$

und damit erhalten wir numerisch (vgl. PYTHON)

$$\tilde{L}\tilde{R} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix} \neq A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

# Gauss-Elimination

In double precision mit  $\text{eps} = 10^{-16}$

$$A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

Pivotelement  $10^{-20} \neq 0$ , LR-Zerlegung von  $A$  existiert:

$$L = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad R = \begin{bmatrix} 10^{-20} & 1 \\ 0 & 1 - 10^{20} \end{bmatrix}$$

Jedoch wird das Ergebnis auf double precision gerundet!

$$\text{fl}(L) = \tilde{L} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad \text{fl}(R) = \tilde{R} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}$$

und damit erhalten wir numerisch (vgl. PYTHON)

$$\tilde{L}\tilde{R} = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 0 \end{bmatrix} \neq A = \begin{bmatrix} 10^{-20} & 1 \\ 1 & 1 \end{bmatrix}$$

# Lösung eines Gleichungssystems mit $\tilde{L}\tilde{R}$ -Zerlegung

$$\text{fl}(L) = \tilde{L} = \begin{bmatrix} 1 & 0 \\ 10^{20} & 1 \end{bmatrix}, \quad \text{fl}(R) = \tilde{R} = \begin{bmatrix} 10^{-20} & 1 \\ 0 & -10^{20} \end{bmatrix}$$

Löst man mit  $\tilde{L}$  und  $\tilde{R}$  das LGS  $Ax = b$  mit  $b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ , so ergibt sich aus

$\tilde{L}\tilde{y} = b$  zunächst

$$\tilde{y} = \begin{bmatrix} 1 \\ -10^{20} \end{bmatrix} \quad (\text{richtig})$$

und dann aus  $\tilde{R}\tilde{x} = \tilde{y}$  die Lösung

$$\tilde{x} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (\text{statt richtig } x = \begin{bmatrix} -1 \\ 1 \end{bmatrix})$$

# Spaltenpivot-Suche

Problem:  $L$ -Faktor enthält ein großes Element.

Erinnerung:  $k - 1$  Schritte Gauß-Elimination liefern

$$A \rightarrow A_{k-1} = \begin{bmatrix} \times & \cdots & \times & \alpha_1 & \times & \cdots & \times \\ & \ddots & & \vdots & \vdots & & \vdots \\ & & \times & \alpha_{k-1} & \times & \cdots & \times \\ & & & \alpha_k & \times & \cdots & \times \\ & & & & \vdots & \vdots & \vdots \\ & & & & \alpha_n & \times & \cdots & \times \end{bmatrix}$$

Spaltenpivot-Suche: Wählt man  $|\alpha_j| = \max_{k \leq i \leq n} |\alpha_i|$  als Pivotelement, gilt  
 $|\ell_{ik}| \leq 1$

## Beispiel für $n = 4$

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

Python-Demo: lrdemo

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} A = \begin{bmatrix} 1 & & & \\ \frac{3}{4} & 1 & & \\ \frac{1}{2} & -\frac{2}{7} & 1 & \\ \frac{1}{4} & -\frac{3}{7} & \frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} 8 & 7 & 9 & 5 \\ \frac{7}{4} & \frac{9}{4} & \frac{17}{4} & \\ -\frac{6}{7} & -\frac{2}{7} & \frac{2}{3} \end{bmatrix}$$

# Gauß-Elimination mit Spaltenpivot-Suche

Nach  $n - 1$  Schritten Gauß-Elimination erhält man

$$L_{n-1}P_{n-1} \cdots L_2P_2L_1P_1A = R$$

mit Permutationsmatrizen  $P_1, \dots, P_{n-1}$  oder

$$L_{n-1}P_{n-1} \cdots L_2P_2L_1P_1 = (L'_{n-1} \cdots L'_2L'_1)(P_{n-1} \cdots P_2P_1)$$

mit

$$L'_k = P_{n-1} \cdots P_{k+1}L_kP_{k+1}^{-1} \cdots P_{n-1}^{-1}$$

Da  $P_j$  nur Vertauschungen Zeilen  $j$  und  $m$  mit  $m > j$  vertauscht, bleibt die Struktur von  $L'_k$  unverändert, lediglich die Elemente unterhalb der Diagonalen werden permutiert

# LR-Zerlegung mit Spaltenpivot suche

**Satz:** Jede nichtsinguläre Matrix  $A \in \mathbb{K}^{n \times n}$  hat eine Zerlerung  $PA = LR$  mit

- $P$  Permutationsmatrix
- $L$  untere Dreiecksmatrix mit  $\ell_{i,i} = 1$  und  $|\ell_{j,k}| \leq 1$
- $R$  obere Dreiecksmatrix mit  $r_{j,j} \neq 0$

Lösung linearer Gleichungssysteme  $Ax = b$  in drei Schritten

- ① Permutieren der rechten Seite  $P \cdot b$
- ② Lösen von  $Ly = Pb$
- ③ Lösen von  $Rx = y$

**Beachte:** Die Permutationsmatrix kann platzsparend mit Hilfe eines Vektors gespeichert werden