Numerische Mathematik für Elektrotechniker

Zahlendarstellung, Rundungsfehler, Stabilität eines Algorithmus

Benjamin Berkels

Karl-Heinz Brakhage, Thomas Jankuhn, Christian Löbbert

Institut für Geometrie und Praktische Mathematik
RWTH Aachen

Wintersemester 2019/2020

Zusammenfassung letzte Vorlesung

Kondition

- ist unabhängig von einem speziellen Lösungweg (Algorithmus)
- gibt an, welche Genauigkeit man bestenfalls (bei exakter Rechnung) bei gestörten Eingangsdaten erwarten kann.

Frage:

- Was ist die (relative) Kondition eines Problems?

$\Rightarrow$ Die relative Kondition eines Problems bezeichnet das Verhältnis des relativen Ausgabefehlers zum relativen Eingabefehler, d.h. die Sensitivität des Problems unter Störungen der Eingabedaten.

Zusammenfassung letzte Vorlesung

- $f : \mathbb{R}^n \to \mathbb{R}$ (Eingabe: Vektor, Ausgabe: Skalar)

$$\left|\frac{f(\tilde{x}) - f(x)}{f(x)}\right| \lesssim \kappa_{\text{rel}}(x) \sum_{j=1}^{n} \left|\frac{\tilde{x}_j - x_j}{x_j}\right|$$

mit

$$\kappa_{\text{rel}}(x) = \kappa_{\text{rel}}^{\infty}(x) := \max_j \left|\frac{\partial f(x)}{\partial x_j} \cdot \frac{x_j}{f(x)}\right|$$

- $f : \mathbb{R}^n \to \mathbb{R}^n$ (Eingabe: Vektor, Ausgabe: Vektor)
  Für $f(x) = Ax$ linear, invertierbar gilt

$$\frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|} \leq \underbrace{\|A\| \cdot \|A^{-1}\|}_{\kappa(A)} \frac{\|\tilde{x} - x\|}{\|x\|}$$

Heute in der Vorlesung

Themen: Dahmen & Reusken Kap 2.2-2.3

- Zahlendarstellung und Rundungsfehler
- Gleitpunktarithmetik
- Stabilität eines Algorithmus

Was Sie mitnehmen sollten:

- Wie werden Zahlen im Computer dargestellt
- Wichtige Eigenschaften der Gleitpunktarithmetik
- Stabilität vs. Kondition

Motivation

Warum betrachten wir Gleitpunktdarstellung?

- Aufgrund der Art und Weise, wie Zahlen im Computer dargestellt werden, können überraschende Ergebnisse auftreten

```
>> u = 0.3/0.1;
>> 3 - u

ans =   4.4409e-16
```

Ein paar schlechte Beispiele:

1. D.N. Arnold, *Some disasters attributable to bad numerical computing*, 1998. `https://www.ima.umn.edu/~arnold/disasters/`
2. T. Huckle, *Collection of Software Bugs*, 2011. `https://www5.in.tum.de/~huckle/bugse.html`
3. K. Vuik, *Some disasters caused by numerical errors*. `http://ta.twi.tudelft.nl/users/vuik/wi211/disasters.html`

Beispiel 2.31.

Wir betrachten als Beispiel die Zahl $\mathbf{123.75}$:

- Dezimalsystem (Basis 10)

$$\begin{aligned}
&\mathbf{123.75} \\
&= \mathbf{1 \cdot 10^{2} + 2 \cdot 10^{1} + 3 \cdot 10^{0} + 7 \cdot 10^{-1} + 5 \cdot 10^{-2}} \\
&= \mathbf{10^{3}\,(1 \cdot 10^{-1} + 2 \cdot 10^{-2} + 3 \cdot 10^{-3} + 7 \cdot 10^{-4} + 5 \cdot 10^{-5})}
\end{aligned}$$

- Binärsystem (Basis 2)

$$\begin{aligned}
&\mathbf{123.75} \\
&= \mathbf{1 \cdot 2^{6} + 1 \cdot 2^{5} + 1 \cdot 2^{4} + 1 \cdot 2^{3} + 0 \cdot 2^{2} + 1 \cdot 2^{1} + 1 \cdot 2^{0}} \\
&\quad \mathbf{+ 1 \cdot 2^{-1} + 1 \cdot 2^{-2}} \\
&= \mathbf{2^{7}\,(1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 0 \cdot 2^{-5} + 1 \cdot 2^{-6}} \\
&\quad \mathbf{+ 1 \cdot 2^{-7} + 1 \cdot 2^{-8} + 1 \cdot 2^{-9})}
\end{aligned}$$

Zahlendarstellung

Seien $b \in \mathbb{N}$, $b > 1$, fest gewählt. Jedes $x \in \mathbb{R}$, $x \neq 0$, lässt sich in der Form

$$x = \pm \left(\sum_{j=1}^{\infty} d_j \, b^{-j} \right) \cdot b^e$$

darstellen, mit $d_j \in \{0, 1, \ldots, b-1\}$, $d_1 \neq 0$,
und e eine ganze Zahl.

- Dezimalsystem (Basis $b = 10$)
$$123.75 \Rightarrow 0.12375 \cdot 10^3$$
- Binärsystem (Basis $b = 2$)
$$123.75 \Rightarrow 0.111101111 \cdot 2^{111}$$

Normalisierte Gleitpunktdarstellung

Gleitpunktdarstellung:

$$\begin{aligned} x &= \pm 0.d_1 d_2 \dots d_m \times b^e \\ &= \pm \left(\sum_{j=1}^{m} d_j \, b^{-j} \right) \times b^e \end{aligned}$$

wobei

- Basis $b \in \mathbb{N} \backslash \{1\}$
- Exponent $e \in \mathbb{Z}$ mit $r \leq e \leq R$
- Mantisse $f = \pm 0.d_1 d_2 \dots d_m, \; d_j \in \{0, 1, \dots, b-1\}$
- Mantissenlänge m
- Normalisierung: $d_1 \neq 0$ für $x \neq 0$

Maschinenzahlen

Nur endliche Anzahl von Zahlen darstellbar m, r, R vs. ∞:

$$x = \pm \left(\sum_{j=1}^{m} d_j \, b^{-j} \right) \times b^e, \qquad r \leq e \leq R$$

$\Rightarrow$ Maschinenzahlen $\mathbb{M}(b, m, r, R)$,
Kardinalität von $\mathbb{M}(b, m, r, R)$ ist endlich und hat

$$2 \times (b-1) \times b^{m-1} \times (R - r + 1) \qquad \text{Elemente.}$$

Definition (Rundung)

Reduktionsabbildung $\mathrm{fl} : \mathbb{R} \to \mathbb{M}(b, m, r, R)$ definiert durch

$$\mathrm{fl}(x) := \pm \begin{cases} \left(\sum_{j=1}^{m} d_j \, b^{-j} \right) \times b^e & \text{falls } d_{m+1} < \frac{b}{2}, \\ \left(\sum_{j=1}^{m} d_j \, b^{-j} + b^{-m} \right) \times b^e & \text{falls } d_{m+1} \geq \frac{b}{2}, \end{cases}$$

Bildbereich

- Betragsmäßig kleinste ($\neq 0$) Zahl:

$$d_1 = 1,\ d_2 = \cdots = d_m = 0;\ e = r: \quad x_{\text{MIN}} = b^{r-1}$$

- Betragsmäßig größte Zahl:

$$d_1 = \cdots = d_m = b-1;\ e = R: \quad x_{\text{MAX}} = (1 - b^{-m}) \times b^R$$

Erinnerung: Geomet. Reihe $(b-1)\sum_{j=1}^{m} b^{-j} = 1 - b^{-m}$

- Bildbereich $\mathbb{D} := [-x_{\text{MAX}}, -x_{\text{MIN}}] \cup [x_{\text{MIN}}, x_{\text{MAX}}]$

Achtung: Die Endlichkeit von e beschränkt den Bildbereich!

- Unterlauf, wenn $0 \neq |x| < |x_{\text{MIN}}|$;
- Überlauf, wenn $|x| > |x_{\text{MAX}}|$.

Bildbereich und Genauigkeit

Maschinenzahlen $\mathbb{M}(\boldsymbol{b}, \boldsymbol{m}, \boldsymbol{r}, \boldsymbol{R})$:

- Es gibt einen begrenzten Bereich von Zahlen, die dargestellt werden können

  Die Endlichkeit von $\boldsymbol{e}$ beschränkt den Bildbereich.

- Es gibt nur eine endliche Anzahl von Zahlen, die innerhalb des Bildbereichs dargestellt werden können

  Die Endlichkeit von $\boldsymbol{f}$ beschränkt die Genauigkeit.

Maschinengenauigkeit – Beispiel

Gleitpunktdarstellung: $b = 10$, $m = 6$

x	$\mathrm{fl}(x)$	$\left\lvert\frac{\mathrm{fl}(x)-x}{x}\right\rvert$
$\frac{1}{3} = 0.33333333\ldots$	$0.333333 * 10^{0}$	$1.0 * 10^{-6}$
$\sqrt{2} = 1.41421356\ldots$	$0.141421 * 10^{1}$	$2.5 * 10^{-6}$
$e^{-10} = 0.000045399927\ldots$	$0.453999 * 10^{-4}$	$6.6 * 10^{-7}$
$e^{10} = 22026.46579\ldots$	$0.220265 * 10^{5}$	$1.6 * 10^{-6}$
$\frac{1}{10} = 0.1$	$0.100000 * 10^{0}$	0.0

Gleitpunktdarstellung: $b = 2$, $m = 10$

x	$\mathrm{fl}(x)$	$\left\lvert\frac{\mathrm{fl}(x)-x}{x}\right\rvert$
$\frac{1}{3}$	$0.1010101011 * 2^{-1}$	$4.9 * 10^{-4}$
$\sqrt{2}$	$0.1011010100 * 2^{1}$	$1.1 * 10^{-4}$
e^{-10}	$0.1011111010 * 2^{-111}$	$3.3 * 10^{-4}$
e^{10}	$0.1010110000 * 2^{1111}$	$4.8 * 10^{-4}$
$\frac{1}{10}$	$0.1100110011 * 2^{-11}$	$2.4 * 10^{-4}$

Dahmen & Reusken

Maschinengenauigkeit

- Für den relativen Rundungsfehler erhält man für $x \in \mathbb{D}$

$$\left|\frac{\mathrm{fl}(x) - x}{x}\right| \leq \frac{\frac{b^{-m}}{2} \cdot b^e}{b^{-1} \cdot b^e} = \frac{b^{1-m}}{2}.$$

- Die (relative) Maschinengenauigkeit

$$\mathbf{eps} := \frac{b^{1-m}}{2}$$

charakterisiert das Auflösungsvermögen des Rechners, d.h.

$$\mathbf{eps} = \inf\{\delta > 0 \mid \mathrm{fl}(1+\delta) > 1\}$$

- Der Rundungsfehler ε erfüllt $|\varepsilon| \leq \mathbf{eps}$ und es gilt

$$\mathrm{fl}(x) = x\,(1+\varepsilon).$$

Maschinengenauigkeit – Beispiel

Gleitpunktdarstellung: $b = 10$, $m = 6 \rightarrow$ eps $= \frac{1}{2} \times 10^{-5}$

x	$\mathrm{fl}(x)$	$\left\lvert\frac{\mathrm{fl}(x)-x}{x}\right\rvert$
$\frac{1}{3} = 0.33333333\ldots$	$0.333333 * 10^{0}$	$1.0 * 10^{-6}$
$\sqrt{2} = 1.41421356\ldots$	$0.141421 * 10^{1}$	$2.5 * 10^{-6}$
$e^{-10} = 0.000045399927\ldots$	$0.453999 * 10^{-4}$	$6.6 * 10^{-7}$
$e^{10} = 22026.46579\ldots$	$0.220265 * 10^{5}$	$1.6 * 10^{-6}$
$\frac{1}{10} = 0.1$	$0.100000 * 10^{0}$	0.0

Gleitpunktdarstellung: $b = 2$, $m = 10 \rightarrow$ eps $= 9.8 \times 10^{-4}$

x	$\mathrm{fl}(x)$	$\left\lvert\frac{\mathrm{fl}(x)-x}{x}\right\rvert$
$\frac{1}{3}$	$0.1010101011 * 2^{-1}$	$4.9 * 10^{-4}$
$\sqrt{2}$	$0.1011010100 * 2^{1}$	$1.1 * 10^{-4}$
e^{-10}	$0.1011111010 * 2^{-111}$	$3.3 * 10^{-4}$
e^{10}	$0.1010110000 * 2^{1111}$	$4.8 * 10^{-4}$
$\frac{1}{10}$	$0.1100110011 * 2^{-11}$	$2.4 * 10^{-4}$

Dahmen & Reusken

Historie

Vor dem Jahr 1985

- Kein einheitlicher Standard
- Jeder Computer hatte seine eigene Gleitpunktdarstellung
- Manche binär (Basis 2, 8, 16), manche dezimal, sogar trinär!
- Gleitpunktarithmetik hat sich auf unterschiedlichen Computern unterschiedlich verhalten!

Im Jahr 1985

- ANSI/IEEE Standard 754-1985 for Binary Floating-Point Arithmetic
- ANSI - American National Standards Institute
- IEEE - Institute of Electrical and Electronics Engineers
- Alle Computer seit 1985 benutzen diesen Standard
- Maschinen-unabhängiges Modell, wie sich Gleitpunktarithmetik verhält

IEEE Standard

- Double-precision floating-point

  $\mathbf{64}$-bit Wort mit

  $\mathbf{52}$ bits für f

  $\mathbf{11}$ bits für e

  $\mathbf{1}$ bit für das Vorzeichen

- Der Exponent e ist eine ganze Zahl im Intervall

$$-\mathbf{1022} \leq e \leq \mathbf{1023}$$

- Effektive Mantisse "$\mathbf{1} + f$"
  ($d_1 = 1$ durch Normalisierung, muss nicht gespeichert werden)

IEEE Standard

- x_{MIN}: $f = 0$ und $e = -1022$
- x_{MAX}: $f = 1 -$ eps und $e = 1023$
- Überlauf: $e = 1024$ und $f = 0$
  - Schreibweise: **infinity** oder `Inf`
  - Erfüllt: `1/Inf = 0` und `Inf+Inf = Inf`
- Not-a-Number oder `NaN`: $e = 1024$ und $f \neq 0$
  - Undefinierte Zahl, z.B. $0/0$
- Unterlauf: $e = -1023$ (Sonderfall: denormalisierte Zahlen)
- In MATLAB:

	Binary	Decimal
eps	2^(-52)	2.2204e-16
realmin	2^(-1022)	2.2251e-308
realmax	(2-eps)*2^1023	1.7977e+308

Pseudoarithmetik

Exakte elementare arithmetische Operation von Maschinenzahlen
$\not\Rightarrow$ Maschinenzahl

Beispiel

$\boldsymbol{b = 10,\ m = 3}$:

$$\mathbf{0.346 \times 10^2 + 0.785 \times 10^2 = 0.1131 \times 10^3 \neq 0.113 \times 10^3}$$

Ähnliches passiert bei Multiplikation und Division.

Exakte Arithmetik $\rightsquigarrow$ Pseudoarithmetik (Gleitpunktarithmetik),

z.B.: $+ \rightsquigarrow \oplus$.

Pseudoarithmetik

Forderung

Für $\nabla \in \{+, -, \cdot, \div\}$ gelte

$$x \mathbin{\textcircled{\nabla}} y = \mathrm{fl}(x \nabla y) \quad \text{für } x, y \in \mathbb{M}(b, m, r, R).$$

Da $\mathrm{fl}(x) = x\,(1+\varepsilon)$, folgt somit, dass für $\nabla \in \{+, -, \cdot, \div\}$

$$x \mathbin{\textcircled{\nabla}} y = (x \nabla y)(1+\varepsilon) \quad \text{für } x, y \in \mathbb{M}(b, m, r, R)$$

und ein ε mit $|\varepsilon| \leq \mathbf{eps}$ gilt (falls $|x \nabla y| \leq x_{\mathrm{MAX}} \wedge x \textcircled{\nabla} y \neq 0$).

Vorsicht bei Pseudoarithmetik:

- Grundlegende Regeln der Algebra, die bei exakter Arithmetik gelten, sind nicht mehr gültig.
- Reihenfolge der Verknüpfung spielt eine Rolle (Assoziativität der Addition geht verloren).

Assoziativgesetz

Beispiel 2.36.

Zahlensystem mit $b = 10,\ m = 3$. Maschinenzahlen

$$\begin{aligned} x &= 6590 &&= 0.659 \times 10^4 \\ y &= 1 &&= 0.100 \times 10^1 \\ z &= 4 &&= 0.400 \times 10^1 \end{aligned}$$

Exakte Rechnung:

$$(x + y) + z = (y + z) + x = 6595.$$

Pseudoarithmetik:

$$x \oplus y = 0.659 \times 10^4 \quad \text{und} \quad (x \oplus y) \oplus z = 0.659 \times 10^4,$$

aber

$$y \oplus z = 0.500 \times 10^1 \quad \text{und} \quad (y \oplus z) \oplus x = 0.660 \times 10^4.$$

Distributivgesetz

Beispiel 2.37.

Für $b = 10,\ m = 3, x = 0.156 \cdot 10^2$ und $y = 0.157 \cdot 10^2$

$$\begin{aligned}(x - y) \cdot (x - y) &= 0.01\\ (x \ominus y) \odot (x \ominus y) &= 0.100 \times 10^{-1}\end{aligned}$$

aber

$$(x \odot x) \ominus (x \odot y) \ominus (y \odot x) \oplus (y \odot y) = -0.100 \times 10^1.$$

Auslöschung

Beispiel 2.38.

Betrachte

$$x = \mathbf{0.73563}, \quad y = \mathbf{0.73441}, \quad x - y = \mathbf{0.00122}.$$

Bei 3-stelliger Rechnung ($b = 10,\ m = 3,\ \text{eps} = \frac{1}{2} \times 10^{-2}$):

$$\begin{aligned} \tilde{x} &= \mathrm{fl}(x) = \mathbf{0.736}, & |\delta_x| &= \mathbf{0.50 \cdot 10^{-3}} \\ \tilde{y} &= \mathrm{fl}(y) = \mathbf{0.734}, & |\delta_y| &= \mathbf{0.56 \cdot 10^{-3}} \end{aligned}$$

Die relative Störung im Resultat:

$$\left| \frac{(\tilde{x} - \tilde{y}) - (x - y)}{x - y} \right| = \left| \frac{\mathbf{0.002 - 0.00122}}{\mathbf{0.00122}} \right| = \mathbf{0.64}$$

also sehr groß im Vergleich zu δ_x, δ_y.

Auslöschung

Betrachte

$$a = 1.23456 \text{ und } b = 1.23567.$$

Sind a und b Ergebnisse aus Rechnungen mit Gleitpunktarithmetik, so sind die niedrigen Stellen durch Rundungsfehler verfälscht.

Angenommen, die letzten drei Stellen sind verfälscht.

Differenz der korrekten Stellen

$$1.23 - 1.23 = 0$$

Allerdings hat

$$b - a = 0.00111 = 0.111 \cdot 10^{-2}$$

keine einzige korrekte Stelle!

Die korrekten Stellen in a und b löschen sich in $a - b$ aus!

Führende Nullen sind nicht Teil der resultierenden Maschinenzahl.

Zusammenfassung Gleitpunktarithmetik

$$\left| \frac{(x \textcircled{\nabla} y) - (x \nabla y)}{(x \nabla y)} \right| \leq \textbf{eps}, \quad x, y \in \mathbb{M}, \quad \nabla \in \{+, -, \cdot, \div\}$$

Die relativen Rundungsfehler bei den elementaren Gleitpunktoperationen sind $\leq$ **eps**, wenn die Eingangsdaten x, y Maschinenzahlen sind.

Sei $f(x, y) = x \nabla y,\ x, y \in \mathbb{R},\ \nabla \in \{+, -, \cdot, \div\}$ und κ_{rel} die relative Konditionszahl von f. Es gilt

$$\begin{aligned} \nabla \in \{\cdot, \div\} &: \quad \kappa_{\text{rel}} \leq 1 \quad \text{für alle } x, y, \\ \nabla \in \{+, -\} &: \quad \kappa_{\text{rel}} \gg 1 \quad \text{wenn } |x \nabla y| \ll \max\{|x|, |y|\} \end{aligned}$$

Sehr große Fehlerverstärkung bei $+, -$ möglich (Auslöschung).

Beispiel: Polynom 7. Grades

$$\begin{aligned} p(x) &= (x-1)^7 \\ &= x^7 - 7x^6 + 21x^5 - 35x^4 + 35x^3 - 21x^2 + 7x - 1 \end{aligned}$$

Matlab-Demo

Stabilität

Definition

Ein Algorithmus heißt gutartig oder stabil, wenn die durch ihn im Laufe der Rechnung erzeugten Fehler in der Größenordnung des durch die Kondition des Problems bedingten unvermeidbaren Fehlers bleiben.

- Kondition ist Eigenschaft des Problems
- Stabilität ist Eigenschaft des Verfahrens/Algorithmus

$\Rightarrow$ Wenn ein Problem schlecht konditioniert ist, kann man nicht erwarten, dass die Numerische Methode (ein stabiler Algorithmus) gute Ergebnisse liefert.

Ziel: Numerische Methode soll Fehlerverstärkung nicht noch weiter vergrößern

Beispiel 2.39.

Bestimmung der kleineren Lösung u^* von

$$y^2 - 2a_1 y + a_2 = 0$$

für $a_1 = 6.000227$, $a_2 = 0.01$.

Algorithmus I

$$u^* = f(a_1, a_2) = -\frac{-2 \cdot a_1}{2} - \sqrt{\left(\frac{-2 \cdot a_1}{2}\right)^2 - a_2}$$

$$= a_1 - \sqrt{a_1^2 - a_2}.$$

$$\begin{aligned} & y_1 = a_1 \, a_1 \\ \longrightarrow \quad & y_2 = y_1 - a_2 \\ \longrightarrow \quad & y_3 = \sqrt{y_2} \\ \longrightarrow \quad & u^* = a_1 - y_3 \end{aligned}$$

Beispiel 2.39.

Algorithmus I

$$u^* = f(a_1, a_2) = a_1 - \sqrt{a_1^2 - a_2}.$$

In Gleitpunktarithmetik mit $b = 10$, $m = 5$ $\left(\text{eps} = \frac{1}{2} \times 10^{-4}\right)$:

$$\tilde{u}^* = 0.90000 \times 10^{-3}$$

Exakte Lösung:

$$u^* = 0.83336 \cdot 10^{-3}$$

- Problem für diese Eingangsdaten $a_1,\ a_2$ gut konditioniert.
- Durch Algorithmus erzeugte Fehler sind sehr viel größer als der unvermeidbare Fehler.

$\Rightarrow$ Algorithmus I ist nicht stabil

Beispiel 2.39.

Bestimmung der Lösung u^* von

$$y^2 - 2a_1 y + a_2 = 0$$

für $a_1 = 6.000227$, $a_2 = 0.01$.

Algorithmus II (Alternative)

$$u^* = \frac{a_2}{a_1 + \sqrt{a_1^2 - a_2}}$$

$$\begin{aligned} & y_1 = a_1\, a_1 \\ \longrightarrow\quad & y_2 = y_1 - a_2 \\ \longrightarrow\quad & y_3 = \sqrt{y_2} \\ \longrightarrow\quad & y_4 = a_1 + y_3 \\ \longrightarrow\quad & u^* = \frac{a_2}{y_4} \end{aligned}$$

Beispiel 2.39.

Algorithmus II

$$u^* = \frac{a_2}{a_1 + \sqrt{a_1^2 - a_2}}$$

In Gleitpunktarithmetik mit $b = 10$, $m = 5$ $\left(\text{eps} = \frac{1}{2} \times 10^{-4}\right)$:

$$\tilde{u}^* = 0.83333 \times 10^{-3}$$

Exakte Lösung:

$$u^* = 0.83336 \cdot 10^{-3}$$

- Gesamtfehler bleibt im Rahmen der Maschinengenauigkeit.
- Auslöschung tritt nicht auf.

$\Rightarrow$ Algorithmus II ist stabil

Rückwärtsstabilität

Ein Verfahren zur Berechnung von $\boldsymbol{f}(\boldsymbol{x})$ liefert als Ergebnis $\tilde{\boldsymbol{f}}(\boldsymbol{x})$.

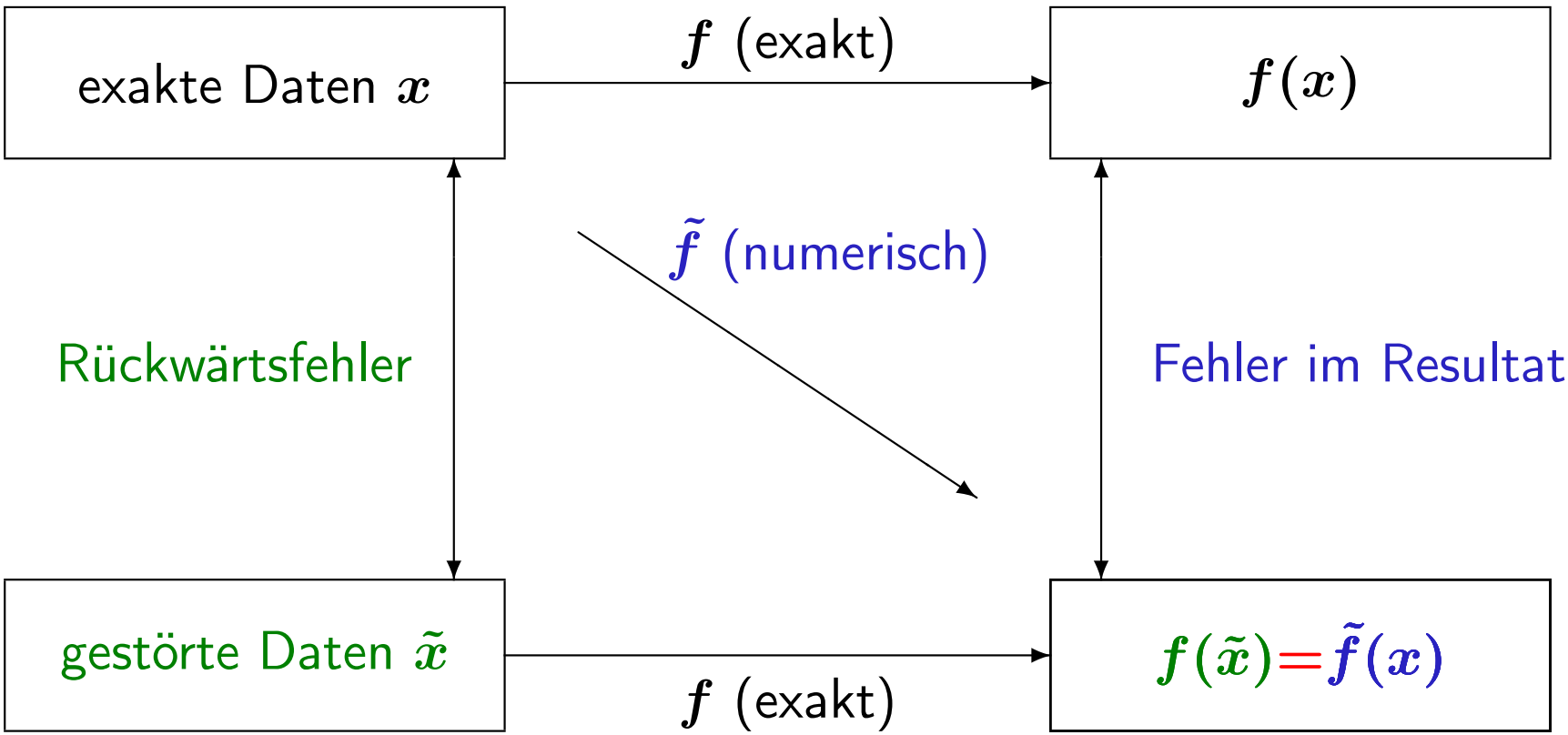
Definition

Das Verfahren heißt rückwärts stabil, wenn es für alle $\boldsymbol{x} \in \boldsymbol{X}$ ein $\tilde{\boldsymbol{x}} \in \boldsymbol{X}$ gibt, so dass

$$\tilde{\boldsymbol{f}}(\boldsymbol{x}) = \boldsymbol{f}(\tilde{\boldsymbol{x}}) \quad \text{und} \quad \frac{\|\boldsymbol{x}-\tilde{\boldsymbol{x}}\|}{\|\boldsymbol{x}\|} = \mathcal{O}(\mathbf{eps}).$$

$\Rightarrow$ Ein rückwärts stabiler Algorithmus gibt die exakte Lösung des nahezu richtigen Problems (Daten, d.h. $\boldsymbol{x} \to \tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{\Delta x}$).

Rückwärtsanalyse



Rückwärtsstabilität

Satz

Wird ein rückwärts stabiler Algorithmus zur Lösung des Problems f mit Kondition $\kappa(x)$ angewendet, so gilt

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = \mathcal{O}(\kappa(x)\,\mathbf{eps}).$$

Beweis:

$$\frac{\|\tilde{f}(x) - f(x)\|}{\|f(x)\|} = \frac{\|f(\tilde{x}) - f(x)\|}{\|f(x)\|} \lesssim \kappa(x) \underbrace{\frac{\|\tilde{x} - x\|}{\|x\|}}_{\mathcal{O}(\mathbf{eps})}.$$

Rückwärtsstabilität

Was haben wir gemacht?

Fehler im Algorithmus $\tilde{f}$ wurden

zurückgespiegelt auf Fehler in den Daten $\tilde{x}$.

$$\Downarrow$$

Vorteil: Auswertung von $\boldsymbol{f}(\tilde{\boldsymbol{x}})$ ist Frage nach Kondition von $\boldsymbol{f}$.

Beispiel 2.40.: Summation ist rückwärts stabil

Geg.: Maschinenzahlen $x_1,\ x_2,\ x_3$, Maschinengenauigkeit **eps**.
Ges.: Summe $S = (x_1 + x_2) + x_3$.

Man erhält

$$\widetilde{S} = ((x_1 + x_2)(1 + \varepsilon_2) + x_3)(1 + \varepsilon_3)$$

mit $|\varepsilon_i| \leq \text{eps},\ i = 2, 3$.

Daraus folgt

$$\begin{aligned}\widetilde{S} &= x_1(1 + \varepsilon_2)(1 + \varepsilon_3) + x_2(1 + \varepsilon_2)(1 + \varepsilon_3) + x_3(1 + \varepsilon_3) \\ &\doteq x_1(1 + \varepsilon_2 + \varepsilon_3) + x_2(1 + \varepsilon_2 + \varepsilon_3) + x_3(1 + \varepsilon_3) \\ &= x_1(1 + \delta_1) + x_2(1 + \delta_2) + x_3(1 + \delta_3)\end{aligned}$$

wobei

$$|\delta_1| = |\delta_2| = |\varepsilon_2 + \varepsilon_3| \leq 2 \cdot \text{eps}, \quad |\delta_3| = |\varepsilon_3| \leq \text{eps}$$

Beispiel 2.40.: Summation ist rückwärts stabil

Es gilt

$$\begin{aligned}\widetilde{S} &= x_1(1+\delta_1) + x_2(1+\delta_2) + x_3(1+\delta_3)\\ &=: \tilde{x}_1 + \tilde{x}_2 + \tilde{x}_3,\end{aligned}$$

wobei

$$|\delta_1| = |\delta_2| = |\varepsilon_2 + \varepsilon_3| \leq 2 \cdot \text{eps}, \quad |\delta_3| = |\varepsilon_3| \leq \text{eps}$$

$\Rightarrow$ Fehlerbehaftetes Resultat $\widetilde{S}$ als exaktes Ergebnis zu gestörten Eingabedaten $\tilde{x}_i = x_i(1+\delta_i)$.

Der durch Rechnung bedingte Fehler ist höchstens

$$\begin{aligned}\left|\frac{f(\tilde{x})-f(x)}{f(x)}\right| &\leq \kappa_{\text{rel}}(x) \cdot \sum_{j=1}^{3} \left|\frac{\tilde{x}_j - x_j}{x_j}\right| \\ &\leq \kappa_{\text{rel}}(x) \cdot \sum_{j=1}^{3} |\delta_j| \leq \kappa_{\text{rel}}(x) \cdot 5 \cdot \text{eps}.\end{aligned}$$

Beispiel 2.40.

Der für die Summation $f(x) = f(x_1, x_2, x_3) = x_1 + x_2 + x_3$ unvermeidbare Fehler ist

$$\left| \frac{f(\tilde{x}) - f(x)}{f(x)} \right| \leq \kappa_{\text{rel}}(x) \cdot \sum_{j=1}^{3} \left| \frac{\tilde{x}_j - x_j}{x_j} \right| \leq \kappa_{\text{rel}}(x) \cdot 3 \cdot \text{eps},$$

wenn Daten höchstens mit Maschinengenauigkeit gestört werden ($\tilde{x}_i = x_i(1 + \varepsilon),\ |\varepsilon| \leq \text{eps}$).

Die Größenordnung der Fehler ist identisch

$\Rightarrow$ Berechnung von S ist ein stabiler Algorithmus.

Summenbildung

Summenbildung tritt in vielen Problemen z. B. Skalarprodukte, Matrix/Vektor-Multiplikation, . . . auf.

Wir betrachten: $\boldsymbol{S_n = \sum_{j=1}^{n} x_j}$

Analog zum Fall $\boldsymbol{S = (x_1 + x_2) + x_3}$ kann man zeigen, dass

$$\begin{aligned}(x_1 \oplus x_2 \oplus \cdots \oplus x_n) - (x_1 + x_2 + \cdots + x_n) \\ \doteq x_1(\varepsilon_1 + \varepsilon_2 + \cdots + \varepsilon_n) \\ + x_2(\varepsilon_2 + \cdots + \varepsilon_n) + \cdots + x_n\varepsilon_n\end{aligned}$$

mit $|\varepsilon_i| \leq \textbf{eps},\ i = 1, \ldots, n$.

- Der erste Summand wird mit größtem Fehler multipliziert.
- Reihenfolge bei der Summation wichtig

$\Rightarrow$ der relative Fehler wird am kleinsten, wenn die betragsgrößten Summanden zuletzt aufsummiert werden (vgl. Beispiel 2.36.).

Zusammenfassung

Was Sie mitnehmen sollten:

Wie werden Zahlen im Computer dargestellt

- Maschinenzahlen $\mathbb{M}(\boldsymbol{b}, \boldsymbol{m}, \boldsymbol{r}, \boldsymbol{R})$
  $\Rightarrow$ $\boldsymbol{x}_{\mathrm{MIN}}$, $\boldsymbol{x}_{\mathrm{MAX}}$, **eps**, $|\varepsilon| \leq$ **eps**

Welche Probleme können dabei/deswegen auftreten?

- Assoziativ- und Distributivgesetz nicht mehr gültig
- Gefahr der Auslöschung bei $\boldsymbol{\nabla} \in \{+, -\}$

Zusammenfassung

Stabilität vs. Kondition

- Bei einem stabilen Lösungsverfahren bleiben die im Laufe der Rechnung erzeugten Rundungsfehler in der Größenordnung der durch die Kondition des Problems bedingten unvermeidbaren Fehler.
- Kenntnisse über die Kondition eines Problems sind oft für die Interpretation oder Bewertung der Ergebnisse von entscheidender Bedeutung
  - "Schlechtes Ergebnis" bedeutet nicht unbedingt gleich "instabiler Algorithmus", sondern deutet evtl. auf eine schlechte Kondition des Problems hin.
- In einem Algorithmus sollen (wegen Stabilität) Auslöschungseffekte vermieden werden.

Verständnisfragen

Es seien x_{MIN} bzw. x_{MAX} die kleinste bzw. größte (strikt) positive Zahl sowie $\mathbf{eps}$ die relative Maschinengenauigkeit in der Menge $\mathbb{M}(b, m, r, R)$ der Maschinenzahlen und $\mathbb{D} := [-x_{\mathrm{MAX}}, -x_{\mathrm{MIN}}] \cup [x_{\mathrm{MIN}}, x_{\mathrm{MAX}}]$. Ferner beschreibe $\mathbf{fl} : \mathbb{D} \to \mathbb{M}(b, m, r, R)$ die Standardrundung. Alle Zahlen sind im Dezimalsystem angegeben.

Berechnen Sie x_{MAX} für $\mathbb{M}(3, 2, -1, 3)$ **24**

w Es gilt $\left|\frac{\mathbf{fl}(x-y)-(x-y)}{(x-y)}\right| \leq \mathbf{eps}$ für alle $x, y \in \mathbb{M}(b, m, r, R)$ mit $x \neq y$ und $|x - y| \leq x_{\mathrm{MAX}}$.

f Es gilt $\left|\frac{\mathbf{fl}(x-y)-(x-y)}{(x-y)}\right| \leq \mathbf{eps}$ für alle $x, y \in \mathbb{D}$ mit $x \neq y$ und $|x - y| \leq x_{\mathrm{MAX}}$.

f Bei einem stabilen Algorithmus ist der Ausgabefehler nicht viel größer als der Eingabefehler.