Numerisches Rechnen

Gleitpunktdarstellung, Stabilität

M. Grepl

J. Berger & R. O'Connor

Institut für Geometrie und Praktische Mathematik
RWTH Aachen

Wintersemester 2015/16

Vorlesungsinhalt

1. **Fehleranalyse: Kondition, Rundungsfehler, Stabilität**
   a) $\boldsymbol{y = f(x)}$, Eingabefehler $\boldsymbol{\Delta x \rightarrow}$ Ausgabefehler $\boldsymbol{\Delta y}$
   b) Fehler aufgrund Gleitpunktdarstellung
   c) Fehler (durch Algorithmus) $\approx$ Fehler (durch Kondition)
2. Lineare Gleichungssysteme, direkte Lösungsverfahren
   geg.: $\boldsymbol{A \in \mathbb{R}^{n \times n}}$, $\boldsymbol{b \in \mathbb{R}^n}$;
   ges.: $\boldsymbol{x \in \mathbb{R}^n}$, so dass $\boldsymbol{Ax = b}$
3. Lineare Ausgleichsrechnung
   geg.: $\boldsymbol{A \in \mathbb{R}^{m \times n}}$, $\boldsymbol{b \in \mathbb{R}^m}$, $\boldsymbol{m > n}$;
   ges.: $\boldsymbol{x^* \in \mathbb{R}^n}$, so dass $\boldsymbol{x^* = \arg\min_{x \in \mathbb{R}^n} \|Ax - b\|_2}$

Übersicht

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
- Welche Probleme können dabei/deswegen auftreten?
- 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 = ?
```

Ein paar schlechte Beispiele:

1. D.N. Arnold, *Some disasters attributable to bad numerical computing*, 1998. `http://www.ima.umn.edu/~arnold/disasters/`
2. T. Huckle, *Collection of Software Bugs*, 2011. `http://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)

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

- Binärsystem (Basis 2)

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

Zahlendarstellung

Man kann zeigen, dass für jedes feste $b \in \mathbb{N}$, $b > 1$, sich jede beliebige reelle Zahl $x \neq 0$ in der Form

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

darstellen lässt, wobei der ganzzahlige Exponent e so gewählt werden kann, dass $d_1 \neq 0$.

- Dezimalsystem (Basis $b = 10$)

$$123.75 \Rightarrow 0.12375 \times 10^3$$

- Binärsystem (Basis $b = 2$)

$$123.75 \Rightarrow 0.111101111 \times 2^{111}$$

Normalisierte Gleitpunktdarstellung

Floating Point Representation:

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

wobei

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

N2.4

Historie

- Vor dem Jahr 1985
  - Keinen einheitlichen 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.

Maschinenzahlen

Nur endliche Anzahl von Zahlen darstellbar (m vs. ∞):

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

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

Definition

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

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

d.h. die letzte Stelle der Mantisse wird um eins erhöht bzw. beibehalten, falls die Ziffer in der nächsten Stelle $\geq \frac{b}{2}$ bzw. $< \frac{b}{2}$ ist.

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**.

Bildbereich

Die Endlichkeit von e beschränkt den **Bildbereich**:

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

$$x_{\mathrm{MIN}} = b^{r-1}$$

- Betragsmäßig größte Zahl:

$$x_{\mathrm{MAX}} = (1 - b^{-m})\, b^{R}$$

Achtung:

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

Maschinengenauigkeit – Beispiel

Gleitpunktdarstellung mit $b = 10$ und $m = 6$

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 mit $b = 2$ und $m = 10$

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

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

- Die (relative) Maschinengenauigkeit N2.5

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

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

N2.6 D:ML

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

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

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

IEEE Standard

N2.7

- 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

$$-1022 \leq e \leq 1023$$

- Der Wert von $2^{52} f$ ist eine natürliche Zahl im Intervall

$$0 \leq 2^{52} f < 2^{52}$$

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: $e = -1023$
- 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

Die Verknüpfung von Maschinenzahlen durch eine exakte elementare arithmetische Operation liefert nicht unbedingt eine Maschinenzahl

Beispiel

$b = 10,\ m = 3$:

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

Ähnliches passiert bei Multiplikation und Division.

Die üblichen arithmetischen Operationen müssen also durch geeignete Gleitpunktoperationen $\textcircled{\triangledown}$ ersetzt werden (Pseudo-arithmetik).

Pseudoarithmetik

Forderung

Für $\nabla \in \{+, -, \times, \div\}$ 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 $\nabla \in \{+, -, \times, \div\}$

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

und ein ε mit $|\varepsilon| \leq \mathrm{eps}$ gilt.

Vorsicht bei Pseudoarithmetik:

- Grundlegende Regeln der Algebra, die bei exakter Arithmetik gelten, sind nicht mehr gültig.
- Reihenfolge der Verküpfung 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 $b = 10,\ m = 3, x = 0.156 \times 10^2$ und $y = 0.157 \times 10^2$

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

aber

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

Auslöschung

Beispiel 2.38:

Betrachte

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

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

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

Die relative Störung im Resultat bei Subtraktion ist hier

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

also sehr groß im Vergleich zu δ_x, δ_y.

Zusammenfassung

N2.8

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

Die relativen Rundungsfehler bei den elementaren Gleitpunktoperationen sind betragsmäßig kleiner als die Maschinengenauigkeit, wenn die Eingangsdaten x, y **Maschinenzahlen** sind.

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

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

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

Beispiele

- `t = 0.1`
- `u = 0.3/0.1`
  - Das Ergebnis ist nicht identisch gleich 3, da Zähler etwas kleiner als 0.3 und Nenner etwas größer als 0.1.
- `a = 2^100; b = a + 2^47; b == a`
  - Der relative Fehler zwischen `a` und `b` is kleiner als $\mathbf{eps}$
  - Es gibt keine "double-precision floating point" Zahl zwischen $\mathbf{2^{100}}$ und $\mathbf{2^{100} + 2^{48}}$
- `eps/2 + 1 - eps/2`
  - Beim Summieren betragsmäßig kleinste Zahlen zuerst aufsummieren.

Beispiele

- Auswerten der Funktion $f(x) = 1 - x\left(\frac{x+1}{x} - 1\right)$
  - Exakte Auswertung

$$\begin{aligned} f(x) &= 1 - x\left(\frac{x+1}{x} - 1\right) \\ &= 1 - x\,\frac{x+1-x}{x} = 0 \quad \text{für alle } x > 0 \end{aligned}$$

- Lösung des (singulären) Gleichungssystems

$$\begin{aligned} 17x_1 + 5x_2 &= 22 \\ 1.7x_1 + 0.5x_2 &= 2.2 \end{aligned}$$

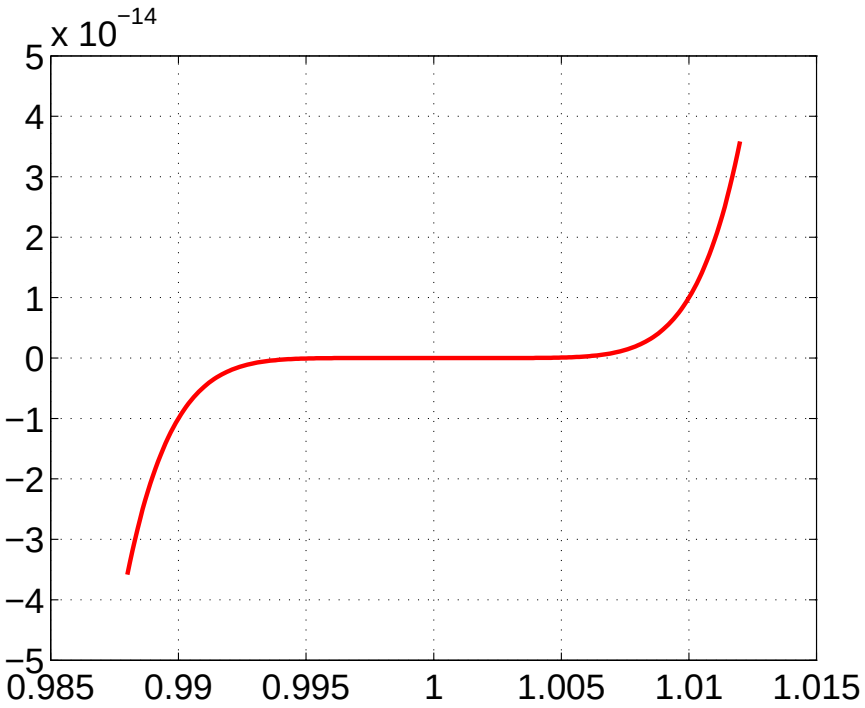
ergibt unendlich viele Lösungen

$$\{(x_1, x_2) \,|\, 17x_1 + 5x_2 = 22\}.$$

Beispiel: Polynom 7. Grades

Matlab Plot

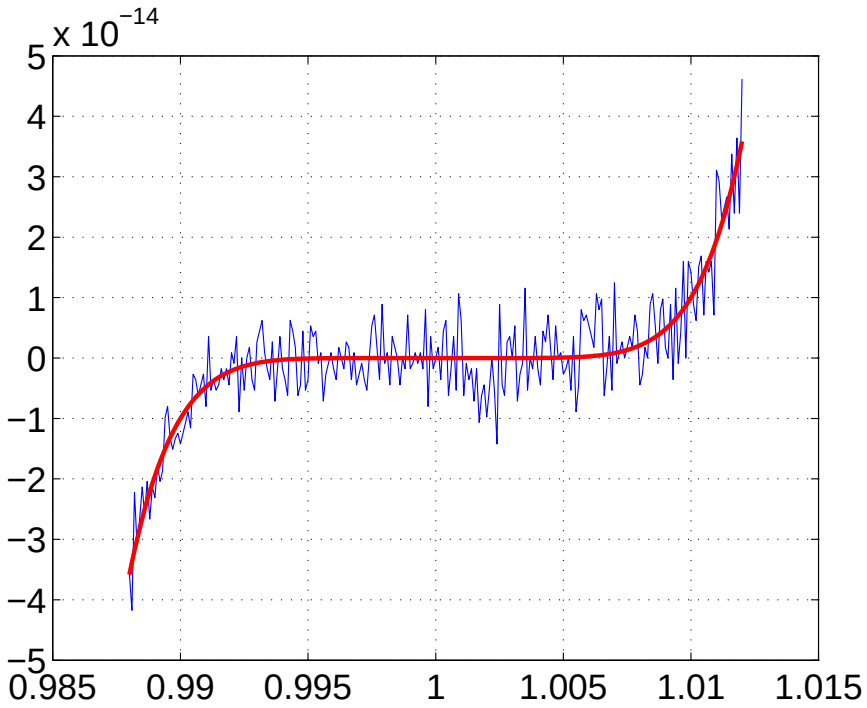
```
x = 0.988:0.0001:1.012;
y = (x-1).^7;
plot(x,y)
```



Beispiel: Polynom 7. Grades

Matlab Plot

```
x = 0.988:0.0001:1.012;
y = x.^7-7*x.^6+21*x.^5-35*x.^4+35*x.^3-21*x.^2+7*x-1;
plot(x,y)
```



Beispiel: Differenzenquotient

Aus der Taylorreihe 2. Ordnung von $f(x)$,

$$f(x+\Delta x) = f(x) + f'(x)\,\Delta x + \frac{f''(x)}{2}\Delta x^2,$$

erhält man eine Approximation der 1. Ableitung durch

$$f'(x) \approx \frac{f(x+\Delta x) - f(x)}{\Delta x} \; (+\mathcal{O}(\Delta x)).$$

ML-Demo:

- Berechne Ableitung von $f(x) = \sin(x)$ für $x = \pi/4$.
- Variiere Δx und plotte den relativen Fehler

D:MV

Vorbemerkung

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 Problem schlecht konditioniert, kann man nicht erwarten, dass die Numerische Methode, d.h. ein stabiler Algorithmus, gute Ergebnisse liefert.

Ziel: Numerische Methode soll Fehlerverstärkung nicht noch weiter vergrößern $\Rightarrow$ Gute Kondition soll erhalten bleiben.

Beispiel 2.39: $y^2 - 2a_1 y + a_2 = 0$

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

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

Für $a_1 = 6.000227$, $a_2 = 0.01$ in einem Gleitpunkt-Zahlensystem mit $b = 10$, $m = 5$ bekommt man das Ergebnis

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

Exakte Lösung: $u^* = 0.83336 \times 10^{-3}$.

- Problem für diese Eingangsaten $a_1,\ a_2$ gut konditioniert.
- Durch Algorithmus erzeugte Fehler sehr viel größer als unvermeidbarer Fehler.

$\Rightarrow$ Algorithmus I ist nicht stabil

Beispiel 2.39: $y^2 - 2a_1 y + a_2 = 0$

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

Algorithmus II

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

Mit $b = 10$, $m = 5$ bekommt man das Ergebnis

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

Exakte Lösung: $u^* = 0.83336 \times 10^{-3}$

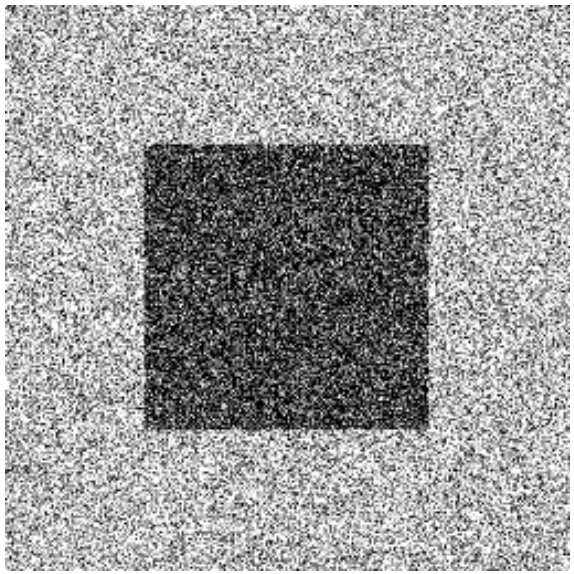
- Gesamtfehler bleibt im Rahmen der Maschinengenauigkeit.
- Auslöschung tritt nicht auf.

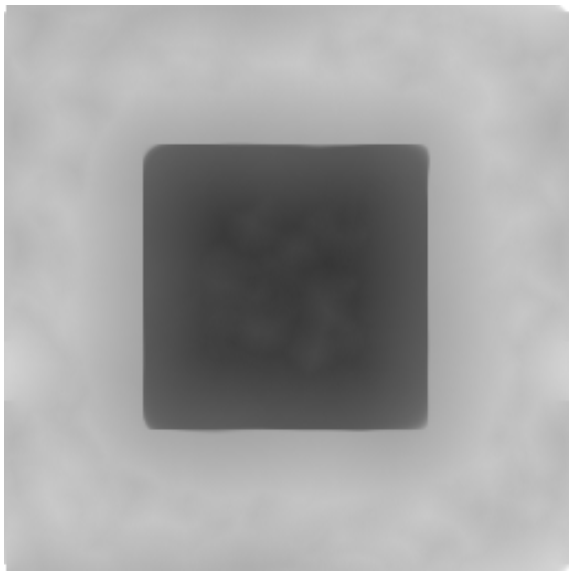
$\Rightarrow$ Algorithmus II ist somit stabil

Beispiel: Filtern verrauschter Bilder

- Ergebnis "Total Variation Minimization"
- Zeitabhängige nichtlineare Diffusionsgleichung
- Matlab demo "Instabilität"

D:ML





Exaktheit eines Algorithmus

Wunsch: Auswertung von $f : X \to Y$
Wirklichkeit: Auswertung von $\tilde{f} : X \to Y$
wobei $f \neq \tilde{f}$ aufgrund von

- Rundungsfehlern (Maschinengenauigkeit),
- Gleitpunktarithmetik.

Definition

Ein Algorithmus heißt exakt, wenn

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

$\Rightarrow$ Ziel "exakter Algorithmus" ist zu ehrgeizig
Grund: Wenn Problem f schlecht konditioniert ist, werden Rundungsfehler um Kondition κ des Problems verstärkt.

Rückwärtsstabilität

Definition

Ein Verfahren heißt rückwärts stabil, wenn für alle $x \in X$,

$$\tilde{f}(x) = f(\tilde{x})$$

für ein $\tilde{x}$ mit $\frac{\|x-\tilde{x}\|}{\|x\|} = \mathcal{O}(\mathbf{eps})$. N2.9

$\Rightarrow$ Ein rückwärst stabiler Algorithmus gibt die exakte Antwort auf die nahezu richtige Frage (Daten, d.h. $x \longrightarrow \tilde{x} = x + \Delta x$).

Erinnerung: Gleitpunkarithmetik erfüllt die Forderung

$$\begin{aligned} \mathbf{fl}(x) &= x\,(1+\varepsilon) \\ \mathbf{fl}(x \nabla y) &= (x \nabla y)\,(1+\varepsilon), \quad \nabla \in \{+,-,\times,\div\} \end{aligned}$$

mit $\varepsilon \leq \mathbf{eps}$.

Rückwärtsstabilität

Satz

Wird ein rückwärst stabiler Algorithmus zur Lösung eines Problems $f : X \to Y$ *mit Kondition* $\kappa(x)$ *angewendet, so gilt*

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

Beweis: $\frac{\|\tilde{f}(x)-f(x)\|}{\|f(x)\|} = \frac{\|f(\tilde{x})-f(x)\|}{\|f(x)\|} \approx \kappa(x)\,\frac{\|\tilde{x}-x\|}{\|x\|}$.

Was haben wir gemacht?

- **Fehler im Algorithmus** wurden zurückgespiegelt auf **Fehler in den Daten**.

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

Beispiel 2.40

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

Man erhält

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

mit $|\varepsilon_i| \leq \mathbf{eps},\ i = 2, 3$. Daraus folgt

$$\begin{aligned}
\tilde{S} &= x_1(1 + \varepsilon_2)(1 + \varepsilon_3) + x_2(1 + \varepsilon_2)(1 + \varepsilon_3) + x_3(1 + \varepsilon_3) \\
&\doteq x_1(1 + \varepsilon_2 + \varepsilon_3) + x_2(1 + \varepsilon_2 + \varepsilon_3) + x_3(1 + \varepsilon_3) \\
&= x_1(1 + \delta_1) + x_2(1 + \delta_2) + x_3(1 + \delta_3) \\
&=: \hat{x}_1 + \hat{x}_2 + \hat{x}_3,
\end{aligned}$$

wobei

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

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

Beispiel 2.40

Der durch Rechnung bedingte Fehler ist höchstens

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

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

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

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

Größenordnung der Fehler identisch $\Rightarrow$ Berechnung von S ist ein stabiler Algorithmus.

Summenbildung

Summenbildung tritt in vielen Problemen auf (Skalarprodukte, Matrix/Vektor-Multiplikation, ...).

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

Man kann zeigen, dass

$$\begin{aligned}\mathbf{fl}(x_1 + x_2 + \ldots + x_n) - (x_1 + x_2 + \ldots + x_n) \\ \approx x_1(\varepsilon_1 + \varepsilon_2 + \ldots + \varepsilon_n) \\ + x_2(\varepsilon_2 + \ldots + \varepsilon_n) + \ldots + x_n\varepsilon_n\end{aligned}$$

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

- 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).

Eigenwertbestimmung

Bestimmung der Eigenwerte einer Matrix mit Hilfe des charakteristischen Polynoms:

$$A = \begin{bmatrix} 1 + 10^{-14} & 0 \\ 0 & 1 \end{bmatrix}$$

Eigenwerte (exakt): $\lambda_1 = 1 + 10^{-14}$ und $\lambda_2 = 1$.

Berechnung über charakteristischen Polynom ergibt:

$$\tilde{\lambda}_1 = 1 + \sqrt{\text{eps}} \quad \Rightarrow \quad \left| \frac{\tilde{\lambda}_1 - \lambda_1}{\lambda_1} \right| = \sqrt{\text{eps}}$$

$$\tilde{\lambda}_2 = 1 - \sqrt{\text{eps}} \quad \Rightarrow \quad \left| \frac{\tilde{\lambda}_2 - \lambda_2}{\lambda_2} \right| = \sqrt{\text{eps}}$$

Relativer Fehler in den Daten (Koeffizienten des charakteristischen Polynoms) ist $\mathcal{O}(\text{eps}) = \mathcal{O}(10^{-16})$, aber Fehler im Ergebnis ist $\mathcal{O}(\sqrt{\text{eps}}) = \mathcal{O}(10^{-8})$.

Zusammenfassung

Was Sie mitnehmen sollten:

Wie werden Zahlen im Computer dargestellt

- Maschinenzahlen $\mathbb{M}(b, m, r, R)$
  $\Rightarrow \; x_{\mathrm{MIN}}, \; x_{\mathrm{MAX}}, \; \mathrm{eps}, \; |\varepsilon| \leq \mathrm{eps}$
- IEEE Standard "double precision floating point"
  Maschinengenauigkeit $\mathrm{eps} \approx 1.11 \times 10^{-16}$

Welche Probleme können dabei/deswegen auftreten?

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

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.