# 1. Grundprobleme des numerischen Rechnens

Viele physikalische Probleme sind nicht mathematisch exakt lösbar, aber nur mit sehr großem Aufwand.
Beispiel: QCD, Naviar-Stokes-Gleichung, usw.

Numerische Verfahren bieten sich oft als Alternative an.
Im Allgemeinen vollziehen wir folgende Schritte vom physikalischen Problem zu dessen Lösung.

## a) mathematische Beschreibung/Modellierung

Unter Umständen werden hier schon Annahmen gemacht, die bereits Näherungen darstellen.

Beispiel:

Ein Teilchen der Masse $m$ bewegt sich entlang der x-Achse unter einer Kraft $F(x)$.

$$F = m \cdot a = m \frac{dv}{dt} \qquad (1)$$

## b) Realisierung

Für das Modell muss ein Lösungsverfahren gefunden werden.

Beispiel:

Teile Zeit in kleine Intervalle $\tau = t_{i+1} - t_i$

$$\Rightarrow v_i = \frac{x_{i+1} - x_i}{t_{i+1} - t_i} = \frac{x_{i+1} - x_i}{\tau} \qquad (2)$$

$$a_i = \frac{v_{i+1} - v_i}{\tau} \qquad (3)$$

Kombiniere 1, 2 und 3

$$x_{i+1} = x_i + \tau v_i$$

$$v_{i+1} = v_i + \frac{\tau}{m} F_i \qquad \text{mit} \quad F_i = F(x)$$

Lösung rekursiv von gegebenen Anfangsbedingungen $(v_0, x_0)$

## c) Validierung

Das Modell und das Lösungsverfahren müssen auf Zuverlässigkeit getestet, Fehlerquellen analysiert und die Stabilität des Verfahrens untersucht werden.
Ggf. müssen mehrere Verfahren verglichen werden.

Beispiel

- Variation von $\tau$
- andere Diskretisierung
- ...

## Definition Algorithmus

Ein Algorithmus ist eine eindeutige Vorschrift, eine mathematische Aufgabenstellung für jeden möglichen Satz von Eingabedaten durch eine endliche Folge algebraischer Grundrechenoperationen mit vorgegebener Genauigkeit zu realisieren.

## Beispiel: Bisektion für $y = \sqrt{x}$

| | |
|---|---|
| Input: $x > 0$, $\varepsilon > 0$ | $x = 5$ |
| if($x \geq 1$) $l = 0, r = x$ | $l_0 = 0$, $r_0 = 5$ $\Rightarrow y_1 = 2{,}5$ |
| else $l = x, r = 1$ | $l_1 = 0$, $r_1 = 2{,}5$ $\Rightarrow y_2 = 1{,}25$ |
| while ($r - l > \varepsilon$) | $l_2 = 1{,}25$, $r_2 = 2{,}5$ $\Rightarrow y_3 = 1{,}875$ |
| $\quad y = \frac{r+l}{2}$ | $\ldots \Rightarrow y_4 = 2{,}1875$ |
| $\quad$ if ($y^2 < x$) $l = y$ | $\Rightarrow y_5 = 2{,}34375$ |
| $\quad$ else $r = y$ | $\Rightarrow y_6 = 2{,}265625$ |
| return $y$ | $\sqrt{5} = 2{,}23606\ldots$ |

Liefert dieser Algorithmus für alle Eingaben $x$ ein Ergebnis?
Nein, für $x < 1$ nicht, da $\sqrt{x} > x \;\forall\; 0 < x < 1$

## Klassifikation von Fehlern

a) <u>Diskretisierungsfehler:</u> Fehler, die durch die Diskretisierung eines Problems kommen

z.B. $\int \rightarrow \Sigma$

b) <u>Datenfehler:</u> Fehler in den Eingangsdaten, Messfehler

c) <u>Rundungsfehler:</u> Fehler bedingt durch die endliche Anzahl von Stellen in der Zahlendarstellung auf einem Rechner

Ist $\tilde{x}$ eine Näherung für $x$, so nennt man

$\Delta_x = \tilde{x} - x$ den <u>absoluten Fehler</u> und für $x \neq 0$

$\delta_x = \frac{\tilde{x} - x}{x}$ den <u>relativen Fehler</u>

## 1.1 Maschinenzahlen

Auf dem Rechner ist nur eine Teilmenge, $M$, der reellen Zahlen darstellbar. Nach IEEE Standard gilt:

$$x = \mathrm{sign}(x) \cdot a \cdot E^{e}$$

- Basis $E \in \mathbb{N}$, $E > 1$ (meist $E = 2$)
- Exponente $e$, im Bereich $e_{min} < e < e_{max}$ wobei $e_{min}, e_{max} \in \mathbb{Z}$
- Mantisse $a \in \mathbb{N}_0$, $a = a_1 E^{k-1} + a_2 E^{k-2} + \dots + a_n E^{0}$ wobei $k$ die Mantissenlänge ist und $a$ die Ziffern im Zahlensystem mit Basis $E = 2$

Bspl: „double" 64bit

| Sign(x) | Exponet | Mantisse |
|---|---|---|
| 1 | 11 | 52 |

$\Rightarrow 10^{-308} < x < 10^{308}$ mit 15-16 Stellen

$\Rightarrow$ Bei der Abbildung der reellen Zahlen auf die Maschinenzahlen muss fast immer eine Rundungsoperation vorgenommen werden. Dabei geht Information verloren und eine Rücktransformation ist nicht möglich.

Beispiel:

a) $0{,}1_{10} = 0{,}0001100110011\dots_2$

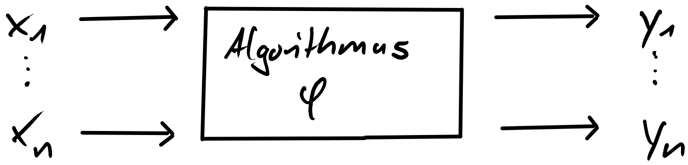
$\rightarrow$ unendlich periodischer Dualbruch

b) Beim addieren zweier $k$-stelliger Zahlen ensteht i.A. ein $(k+1)$-stelliger Ausdruck.

Als Maschinengenauigkeit bezeichnet man die kleinste positive Maschinenzahl $\varepsilon_m \in M$ für die
$1 + \varepsilon_m > 1$ gilt.

## 1.2 Fehlerfortpflanzung, Konditionisierung

Betrachte einen Algorithmus in folgender Weise:




also $y_i = f_i(x_1, \dots, x_n)$ mit $i = 1, \dots, m$

Für $n = m = 1$: $\quad y = f(x)$

Sei $\Delta x$ der Absolute Datenfehler von $x$, dann gilt mit exakter Arithmetik in erster Näherung (Taylor)

$$\Delta y = \frac{\partial f}{\partial x} \Delta x$$

und damit

$$\delta y = \frac{\Delta y}{y} = \frac{x}{y} \frac{\partial f}{\partial x} \frac{\Delta x}{x} = \frac{x}{y} \frac{\partial f}{\partial x} \delta x$$

wobei $K = \frac{x}{y} \frac{\partial f}{\partial x} = \frac{x}{f(x)} \frac{\partial f}{\partial x}$ Konditionszahl genannt wird.

Für $n, m > 1$ ist $K$ offensichtlich eine Matrix:

$$K_{ij} = \frac{x_j}{y_i} \frac{\partial f_i}{\partial x_j} \qquad \text{wobei} \qquad \delta y_i = \sum_{j=1}^{m} K_{ij} \, \delta x_j$$

Die Konditionszahlen sind die Verstärkungsfaktoren des relativen Eingabefehlers und hängen offensichtlich stark vom Algorithmus $f$ ab.

## typische Beispiele

Berechne Nullstellen von $x^2 - 60x + 1 = 0$ mit 4-stelliger Arithmetik. Exakt $x_1 = 0{,}0166713$

$x_2 = 59{,}9833$

$$\alpha = -\frac{p}{2} = 30.00$$

$$\beta = \alpha^2 = 900.0$$

$$\gamma = \beta - q = 899.0$$

$$\delta = \sqrt{\gamma} = 29{,}98$$

$\Rightarrow \tilde{x}_1 = \alpha - \delta = 0{,}02000$ $\qquad$ $\tilde{x}_2 = \alpha + \delta = 59{,}98$

$\delta_{x_1} = -0{,}1997$ $\qquad$ $\delta_{x_2} = 0{,}00005502$

$$x_{1/2} = -\frac{p}{2} \pm \sqrt{\left(\frac{p}{2}\right)^2 - q}$$

Problem: subtrahieren zweier fast gleich großer Zahlen.