# Computergestüzte Methoden der exakten Naturwissenschaften

Prof. Dr. Roland Netz

18. Oktober 2013

## Inhaltsverzeichnis

# 1 Fehler

Ein Ziel der Naturwissenschaften ist die Beschreibung der Natur mit Hilfe von mathematischen Gleichungen und deren Lösungen, daraus ergibt sich allerdings ein Problem.

**Problem:** Die Gleichungen der naturwissenschaftlichen Beschreibungen können nicht immer mit Bleistift und Papier zu gelöst werden.

**Lösung 1:** Vereinfachung der Gleichungen $\hat{=}$ Näherung/Approximation

**Lösung 2:** Numerische Lösung der Gleichungen.

Diese Vorlesung möchte sich mit der zweiten Lösungsmethode befassen, hierbei ist es allerdings wichtig die Genauigkeit der numerisch ermittelten Ergebnisse (die Fehler) mit zu berücksichtigen.

Allgemein gibt es für es verschiedene Quellen für Fehler:

**Eingabefehler:** Diese entstehen durch Ungenauigkeiten innerhalb der Eingabedaten.

**Näherungsfehler:** Solche entstehen aus der Verwendung vereinfachter mathematischer Ausdrücke anstelle der exakten.

**Modellfehler:** Diese entstehen aus der Nutzung vereinfachter physikalischer Modelle.

**Rundungsfehler:** Solche entstehen aus der numerischen Darstellung von Zahlen und der damit verbundenen endlichen Genauigkeit.

## 1.1 Beispiele für Näherungsfehler

Viele mathematische Gleichungen der Physik sind in ihren exakten Formulierungen nicht oder nur sehr aufwendig lösbar. Ein Ausweg stellen Approximationen dar aus welchen allerdings zusätzliche Näherungsfehler resultieren. Beispiele hierfür sind über unendliche Reihen definierte Funktionen aber auch Differentialgleichungen im Kontinuum.

**Exponentialfunktion:** Die Exponentialfunktion ist definiert durch:

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}.$$

Eine solche Funktion kann durch eine endliche Reihe genähert werden:

$$e^x = \sum_{n=0}^{N} \frac{x^n}{n!}$$

**Differentialgleichung im Kontinuum:** Eine Differentialgleichung im Kontinuum kann durch die Lösung der zugehörigen diskretisierten Gleichung genähert werden. Sei die Differentialgleichung gegeben durch:

$$\frac{d}{dx} f(x) = a\ f(x),$$

so ergibt sich die dikretisierte Gleichung aus der Diskretisierung auf bestimmte Gitterpunkte $x_i$ mit dem Abstand $\Delta x = x_{i+1} - x_i$:

$$\frac{f(x_{i+1}) - f(x_i)}{x_{i+1} - x_i} = a\ \frac{f(x_{i+1}) + f(x_i)}{2}.$$

Zur Verbesserung der Diskretisierung kann dann $\Delta x$ immer weiter gegen 0 gesetzt werden. Ein *Nachteil* ist hierbei die Erhöhung der Rechenoperationen und der damit verbundenen Rechenzeit. Außerdem vergrößern sich hiermit die Rundungsfehler.

Das Ziel der Numerik besteht nun im optimalen Kompromiss zwischen Fehler und Rechenzeit.

## 1.2 Beispiel für Modellfehler

Als Beispiel für einen aus einem Modell resultierenden Fehler wird die Planetenbewegung betrachtet. Nach dem ersten newtonschen Gesetz gilt:

$$\mathbf{F} = m\mathbf{a} = m\ddot{\boldsymbol{r}} = -\frac{M\ \mathbf{r}}{|\mathbf{r}|^3}$$

Hierbei gehen allerdings eine reihe von Näherungen ein:

- Die Sonnenmasse $M$ wird relativ zur Planetenmasse als sehr groß angenommen
- Eine geschwindigkeitsabhängige Reibungskraft $\boldsymbol{F_R} = \gamma\ \dot{\boldsymbol{r}}$ wird vernachlässigt, diese ist allerdings für kleinere Objekte wichtig.
- Auch relativistische Effekte werden vernachlässigt, solche erklären allerdings Phänomene wie die Periheldrehung des Merkurs.
- Eigentlich handelt es sich um ein Mehrkörperproblem der Form:

$$m_1\ddot{\boldsymbol{r}}_{\boldsymbol{i}} = \sum \boldsymbol{F}_i = -\sum_{j \neq i} G m_i m_j \frac{\boldsymbol{r_i} - \boldsymbol{r_j}}{|\boldsymbol{r_i} - \boldsymbol{r_j}|^3}.$$

Eine Gleichung für mehr als 3 Objekte kann so also leicht geschrieben werden. Allerdings ist das Problem bereits ab einer Beteiligung von 3 Objekten nur noch unter Annahme bestimmter Bedingungen und ab 4 Objekten überhaupt nicht mehr exakt lösbar.

## 1.3 Rundungsfehler

Beim durchführen von Rechenoperationen mit reellen Zahlen am Computer muss gerundet werden, die daraus entstehenden Fehler heißen Rundungsfehler.

### 1.3.1 Gleitpunktarithmetik

Reelle Zahlen werden am Computer in das Gleitpunktformat umgewandelt. Der Vorteil gegenüber dem Festpunktformat liegt im geringeren Speicherbedarf. Hierzu werden die eingegebenen Zahlen in der Form:

$$x = \pm \sum_{i=1}^{n} z_i \; B^{E-i} := \pm \underbrace{(0, z_1 z_2 ... z_n)_B}_{\text{Mantisse}} \; B^E$$

dargestellt. Dabei gilt des weiteren für den Exponenten $E \in \mathbb{Z}$: $m \leq E \leq M$. Außerdem gilt $z_i \in \{0, 1...B-1\}$.
**Beispiel**:

$$1234,567 = (0,1234567)_{10} \cdot 10^4$$

Die Werte für $n$, $B$, $m$ und $M$ sind hierbei maschinenabhängig, werden also durch den Rechner und den Compiler bestimmt.
Übliche Basen sind:

$B = 2$: Dualzahlen

$B = 8$: Oktalzahlen

$B = 10$: Dezimalzahlen

$B = 16$: Hexadezimalzahlen

**Standartformate für B=2:**
**Single:** Dieses Format besteht aus 32 Bits bzw. 4 Bytes. Diese ergeben sich aus:

Vorzeichen: 1 Bit

Exponent: 8 Bits

Mantisse: 23 Bits

Genauigkeit: 6 Ziffern unterscheidbar

**Double:** Dieses Format besteht hingegen aus 64 Bits:

Vorzeichen: 1 Bit

Exponent: 11 Bits

Mantisse: 52 Bits

Genauigkeit: 15 Ziffern unterscheidbar

**Beispiel** Binäre Darstellung von $(5,0625)_{10}$:

$$(5,0625)_{10} = (0,50625)_{10} \cdot 10^1 = 2^2 + 2^0 + 2^{-3} + 2^{-4}$$
$$\Rightarrow (5,0625)_{10} = (101,0001)_2 = (0,1010001)_2 \cdot 2^{(11)_2}$$

Manche Zahlen wie $(0,3)_{10}$ lassen sich allerdings nur schwer als duale Zahlen darstellen.

Die **größte darstellbare Zahl** ergibt sich zu:

$$x_{max} = (0, \underbrace{[B-1][B-1]...[B-1]}_{\text{n Ziffern}})_B \; B^M = B^M[B-1]\frac{B^{-n}(B^n-1)}{B-1} = B^M(1-B^{-n}).$$

Dagegen ergibt sich die **kleinstmögliche Zahl** zu:

$$x_{min} = B^{m-1}$$

Folglich ist die Menge der darstellbaren Maschinenzahlen endlich. Ergibt sich während der Rechnung eine Zahl $x > x_{max}$ folgt ein overflow und die Zahl wird auf $\infty$ gesetzt. In gleicher weise ergibt sich für $x < x_{min}$ der underflow und die Zahl wird auf 0 gesetzt.
**Beispiele:**

$$x_{max} + x_{max} = \infty \tag{1.1}$$
$$x_{min} \; B^{-1} = 0 \tag{1.2}$$

Jede reelle Zahl die keine Maschinenzahl ist muss in eine solche umgewandelt werden. Idealerweise wählt man Maschinenzahl dabei möglichst nahe der reellen Zahl $\hat{=}$ Rundung.

### 1.3.2 Rundung

Beim Runden wird für eine Zahl $x$ eine Näherung $rd(x)$ unter den Maschinenzahlen geliefert, so dass der absolute Fehler $|x - rd(x)|$ minimal ist. Der dabei unvermeidbare Fehler heißt Rundungsfehler. Eine n-stellige Dezimalzahl im Gleitpunktformat $\tilde{x} = \pm(0, z_1...z_n)_{10} = rd(x)$ hat einen maximalen Fehler von:

$$|x - rd(x)| \leq 0, \underbrace{00...00}_{\text{n Ziffern}} 5 \cdot 10^E = 0,5 \cdot 10^{E-n}.$$

Für eine allgemeine Basis B ergibt sich:

$$|x - rd(x)| \leq \frac{B}{2} \; \frac{1}{B} \; B^{E-n} = \frac{1}{2} B^{E-n}.$$

Rundungsfehler werden durch die gesamte Rechnung getragen.

Bei einer **n-stellige Gleitpunktarithmetik** wird jede einzelne Rechenoperation auf $n+1$ Stellen genau berechnet und dann auf $n$ Stellen gerundet. Es wird also nicht nur das Endergebnis gerundet.

**Beispiel:** $2590 + 4 + 4$ in *3-stelliger dezimaler Gleitpunktarithmetik*
Von links nach rechts:

$$2590 + 4 = 2594 \underbrace{\Rightarrow}_{\text{Rundung}} 2590$$
$$2590 + 4 = 2594 \underbrace{\Rightarrow}_{\text{Rundung}} 2590$$

Von rechts nach links:

$$4 + 4 = 8 \underbrace{\Rightarrow}_{\text{Rundung}} 8$$
$$2590 + 8 -> 2598 \underbrace{\Rightarrow}_{\text{Rundung}} 2600$$

Das exakte Ergebnis wäre 2598. Die Reihenfolge der Ausführungen der Rechenoperationen verändert also das Ergebnis. Daraus folgt die **Regel**, dass beim **Addieren** die Summanden in der Reihenfolge ihrer aufsteigender Beträge addiert werden. So erhält man bei gleicher Rechenzeit bessere Ergebnisse.

*Einschub:Maß für die Rechenzeit eines Computers:*
*flops$\hat{=}$floating point operations per second, dabei sind Multiplikation und Division typische Operationen. Eine Rangliste schnellsten Computer wird auf www.top500.org geführt.*

Der **relative Fehler** ist meist relevanter als der absolute Fehler. Die Näherung $\tilde{x}$ zu dem exaktem Wert $x$ ergibt einen relativer Fehler: $\epsilon = |\frac{\tilde{x}-x}{x}| \approx |\frac{\tilde{x}-x}{\tilde{x}}|$.
Daraus ergibt sich der maximaler Rundungsfehler zu:

$$\epsilon_{max} = \frac{\frac{1}{2}\, B^{E-n}}{B^{E-1}} = \frac{1}{2}\, B^{1-n}$$

Für duale Rechnungen im Computer gilt also $B = 2\epsilon_{max} \cdot 2^{-n}$.
$\epsilon_{max}$ wird auch **Maschinengenauigkeit** genannt und gibt die kleinste positive Zahl an für die gilt $1 \cdot \epsilon_{max} \neq 1$.
$\epsilon_{max}$ kann aus Rechenoperationen rekonstruiert werden.