CDS404 - Numerische Methoden

Silvan Wiedmer, Daniel Schafhäutle

April 27, 2023

Contents

1 Einführung

1.1 Begriffe

1.1.1 Numerik

Numerik (auch Numerische Mathematik genannt) bezeichnet den Bereich der Mathematik, der sich mit der Entwicklung und Analyse von Verfahren zur numerischen Lösung von mathematischen Problemen beschäftigt.
Dabei geht es in erster Linie darum, rechnergestützt mathematische Berechnungen durchzuführen und Ergebnisse zu erhalten, die in der Praxis anwendbar sind.
Die Numerik liefert nur Näherungsweise Ergebnisse.

1.1.2 Gleitkommazahl

Gleitkommazahlen bestehen aus einer festen Anzahl von Ziffern, um den Wert der Zahl anzugeben, sowie der Angabe, um wie viele Stellen das Komma nach links oder rechts verschoben werden muss, um die Zahl in der Dezimalschreibweise zu erhalten.
Der Vorteil einer solchen Zahlendarstellung ist, dass mit einer festen Anzahl von ausgeschriebenen Ziffern ein großer Zahlenraum abgedeckt wird und die Zahlen trotzdem eine gute Genauigkeit behalten.

$$12000 = 1.2 \cdot 10^4$$
$$12000 = 1.2e4$$

1.1.3 Gleitkommaarithmetik

Gleitkommaarithmetik ist eine Methode zur Darstellung und Berechnung von Zahlen in der Computerarithmetik, die auf der Approximation von reellen Zahlen durch Mantisse und Exponenten basiert.
Sie bezeichnet das Rechnen mit Gleitkommazahlen und deren Darstellung.

1.1.4 Basis, Mantisse & Exponent

$$1.2 \cdot 10^4$$

- Mantisse: 1.2
  Die Ziffernstellen einer Gleitkommazahl vor der Potenz.
- Basis: 10
- Exponent: 4

1.1.5 Rundungsfehler

Die meisten Gleitkommazahlen können nicht exakt dargestellt werden, was zu Rundungsfehlern führt, wenn solche Zahlen berechnet oder manipuliert werden. Diese Fehler entstehen durch die Notwendigkeit, die dargestellte Zahl auf die nächstgelegene darstellbare Zahl zu runden.
Dies kann zu einer Abweichung vom tatsächlichen Wert führen und zu inkorrekten Ergebnissen führen.

1.2 Unterschied Arithmetik & Gleitkommaarithmetik

Der Hauptunterschied zwischen Arithmetik und Gleitkommaarithmetik besteht darin, dass Arithmetik sich auf die grundlegenden mathematischen Operationen bezieht, während Gleitkommaarithmetik eine Methode zur Darstellung und Berechnung von Zahlen ist, die auf den Anforderungen der Computerarithmetik basiert.
Arithmetik kann auf verschiedene Arten durchgeführt werden, z. B. mit ganzen Zahlen, rationalen Zahlen oder reellen Zahlen. Gleitkommaarithmetik dagegen bezieht sich speziell auf die Darstellung von reellen Zahlen und erfordert eine bestimmte Konvention für die Repräsentation von Zahlen mit einer begrenzten Anzahl von Bits.

1.3 Probleme der Gleitkommaarithmetik

Die wichtigsten Probleme/Effekte, welche zu falschen Ergebnissen führen.

1.3.1 Auslöschung

Die Auslöschung tritt auf, wenn zwei zahlen subtrahiert werden, welche in ihren vorderen Dezimalstellen identisch sind. Die hinteren Dezimalstellen haben sehr wahrscheinlich Rundungsfehler, nun werden diese "nach vorne gezogen" und haben eine viel grössere Bedeutung.

Example 1.1.

$$2.345678 - 2.346789 = 0.001111$$

Angenommen die ersten 4 Ziffern sind genau, die restlichen haben Rundungsfehler. Dann wäre das Ergebnis eigentlich:

$$2.346 - 2.345 = 0.001$$

Das ergibt einen Relativen Fehler von:

$$\begin{aligned} 0.001111 - 0.001 &= 0.000111 \\ \frac{0.000111}{0.001} &= 0.111 \\ &= 11.1\% \end{aligned}$$

Vor der Subtraktion war der Fehler vernachlässigbar, aber jetzt ist er durch die Auslöschung sehr gross geworden.

1.3.2 Absorption

Absorption geschieht, wenn sehr kleine Zahlen mit Grossen addiert/subtrahiert werden und sich der Betrag des Ergebnis nicht von dem der Grösseren unterscheidet.
Was dabei geschieht ist, dass die Veränderung der Zahl so weit hinten geschieht, so dass sie nach dem Umwandeln in eine Gleitkommazahl gar nicht mehr in der Zahl "vorhanden ist".

Example 1.2. Angenommen, es werden 4 Ziffern für die Zahl gespeichert.

$$100 + 0.001 = 100.001 = 1.000'01 \cdot 10^2 \approx 1.000 * 10^2$$

So fällt die eins einfach aus der Zahl heraus und geht verloren.

1.3.3 Unterlauf

Liegt vor, wenn das Ergebnis einer Rechenoperation zwischen der kleinsten darstellbaren Zahl und null liegt, dann wird zu null abgerundet.
Z.B. Wenn 8 Stellen gespeichert werden:

$$1 - 0.999'999'99 = 0.000'000'001$$

Die Eins liegt ausserhalb des gespeicherten Bereiches und fällt deswegen einfach weg. Als Ergebnis gäbe es also 0.

1.4 Fixpunkt Iteration

$$f(x) = x$$

Heron-Verfahren

- Gegeben: $g > 0$
- Gesucht: $\sqrt{}(q)$
- $a_0 = 1$

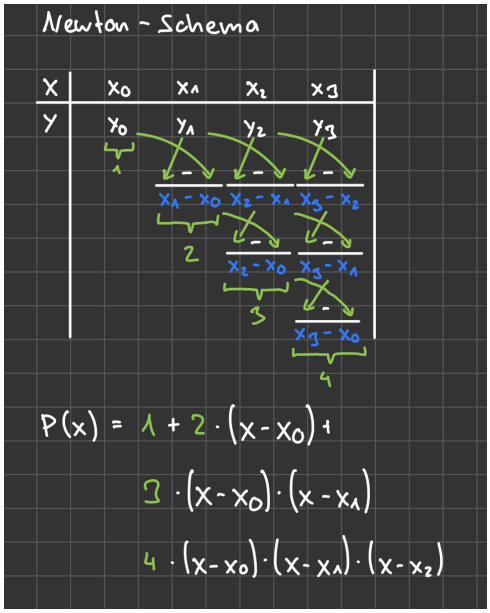
$$a_{k+1} = \frac{1}{2} \cdot \left(a_k + \frac{q}{a_K} \right)$$

Banach-Fixpunkt-Satz

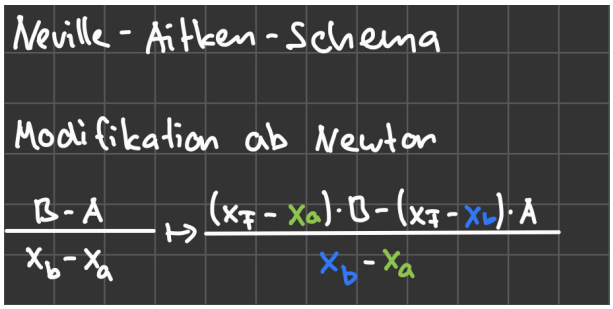
$$|f(x_2) - f(x_1)| \leq m \cdot |x_2 - x_1|$$

2 Interpolation & Extrapolation

2.1 Polynom-Interpolation



2.2 Polynom-Extrapolation



2.3 Regression

2.3.1 Lineare Regression

LGLS in Matrix-Form

$$\vec{u} = \begin{bmatrix} m \\ q \end{bmatrix}, \vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_N \end{bmatrix}, \vec{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_N \end{bmatrix}, A = \begin{bmatrix} x_1 & 1 \\ \vdots & \vdots \\ x_N & 1 \end{bmatrix}$$

$$A^T \cdot A \cdot \vec{u} = A^T \cdot \vec{y}$$

2.3.2 Exponentielle Regression

Um die Regression zu finden, werden die Daten mittels logarithmierung auf einen lineare Zusammenhang zurückgeführt.

3 Numerische Integration

Ist die Stammfunktion nicht bekannt, so kann das Integral durch eine einfachere funktion approximiert werden.

Dabei wird die Annäherung durch die folgenden Funktionen realisiert.

- konstante Funktion
- lineare Funktion
- quadratische Funktion

3.1 Konstante Funktion

Links

$$I = h \sum_{k=0}^{n-1} f(x_k)$$

Rechts

$$I = h \sum_{k=1}^{n} f(x_k)$$

Mittelpunkt

$$I = h \sum_{k=0}^{n-1} f\left(\frac{x_k + x_{k+1}}{2}\right)$$

3.2 Lineare Funktion (Trapez Regel)

Allgemein

$$I = \frac{1}{2} \sum_{k=0}^{n-1} (x_{k+1} - x_k) \cdot (f(x_{k+1}) + f(x_k))$$

Äquidistante X-Werte

$$I = \frac{h}{2} \sum_{k=0}^{n-1} (f(x_{k+1}) + f(x_k))$$

3.3 Quadratische Funktion (Simpson Regel)

$$A_k = \frac{h}{3}(f(x_{k-1}) + 4f(x_k) + f(x_{k-1}))$$

4 Code Snippets

4.1 numpy

import

```
import numpy as np
```

4.1.1 linspace

```
np.linspace(start, stop, num=50)
```

4.1.2 polyfit

```
np.polyfit(x_data, y_data, grad)
```

4.1.3 polyval

```
np.polyval(polynom, x_data)
```

4.1.4 trapz

```
np.trapz(y_data, x_data)
```

4.2 matplotlib

4.2.1 import

```
import matplotlib.pyplot as plt
```

4.2.2 rcParams

Mit den **rcParams** kann die visualisierung beliebig konfiguriert werden.

```
plt.rcParams['figure.figsize'] = (7.03, 15)
plt.rcParams['font.size'] = 9
plt.rcParams['font.family'] = 'serif'
plt.rcParams['text.usetex'] = True
```

4.2.3 plot

```
plt.plot(x_data, y_data, '--', linewidth=3, label=r'$g$')
plt.xlabel(r'$x$')
plt.ylabel(r'$y$')
plt.legend()
plt.grid(visible=True)
plt.axis('image')
```

4.3 scipy

4.3.1 import

```
import scipy.interpolate as ip
import scipy.integrate as ig
```

4.3.2 Polynom interpolation

```
po = ip.BarycentricInterpolator(x_data, y_data)
```

4.3.3 Cubic Spline

```
cs = ip.CubicSpline(x_data, y_data)
```

4.3.4 Simpson Regel

```
I = ig.simps(y_data, x_data)
```