Übung 02: Verfahrens- und Maschinenfehler

Tobias Blesgen und Leonardo Thome

19.05.2021

Im folgenden wollen wir die 2-Punktformel und 3-Punktformel auf ihre Verfahrensfehler untersuchen und durch Verwendung verschiedener Datentypen den Maschienfehler abschätzen.

Die 2-Punktformel beschreibt die Ableitung einer Funktion $f(x)$ an der Stelle x_i mit:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} + \mathcal{O}(h) \tag{1}$$

Die 3-Punktformel leistet die Berechnung hingegen mit einem kleineren Verfahrensfehler der Ordnung $\mathcal{O}(h^2)$

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} + \mathcal{O}(h^2) \tag{2}$$

Verfahrens- und Maschinenfehler

Verfahrensfehler

Verfahrensfehler der 2-Punktformel

Nach der Taylor Entwicklung einer Funktion ergibt sich

$$f(x_{i+1}) = f(x_i) + hf'(x_i) + \frac{h^2}{2}f''(x_i) + \mathcal{O}(h^3). \tag{3}$$

Stellt man die Gleichung nach der Ableitung um, ergibt sich die gesuchte Form für die 2-Punktformel mit Fehlerordnungen:

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_i)}{h} - \frac{h}{2}f''(x_i) + \mathcal{O}(h^2) \tag{4}$$

Somit wird die höchste Fehlerordnung, und somit der Verfahrensfehler der 2-Punktformel durch $\frac{h}{2}f''(x_i)$ beschrieben.

Verfahrensfehler der 3-Punktformel

Nach einem ähnlichen Ansatz wie zuvor betrachten wir in der 3-Punktformel die Taylorformel vor und nach x_i:

$$f(x_{i\pm1}) = f(x_i) \pm hf'(x_i) + \frac{h^2}{2}f''(x_i) \pm \frac{h^3}{6}f'''(x_i) + \mathcal{O}(h^4). \tag{5}$$

Subtrahieren wir die positive und negative Gleichung ergibt sich

$$f'(x_i) = \frac{f(x_{i+1}) - f(x_{i-1})}{2h} - \frac{h^2}{6}f'''(x_i) + \mathcal{O}(h^3) \tag{6}$$

Der Verfahrensfehler liegt also bei $\frac{h^2}{6}f'''(x_i)$.

Maschinenfehler

Da wir Zahlen am Computer nur endliche Stellen haben, geschiehen Rundungsfehler bei Berechnung von Zahlen die mehr Stellen besitzen als der benutzte Typ. Diese Genauigkeit nennen wir Maschiengenauigkeit δ_M und ist definiert über die größte reelle Zahl δ_M, so dass der Rechner noch $1 + \delta_M = 1$ berechnet (Seite 12 /cite{DeltaM}).

Maschinenfehler der 2-Punktformel

Mit

$$\Delta M(f(x+h) - f(x)) = \sqrt{f(x+h)^2\sigma^2 + f(x)^2\sigma^2} \tag{7}$$

$$\approx \sqrt{2\sigma^2}|f(x)| \tag{8}$$

$$= \sqrt{2}\sigma|f(x)| = \sqrt{\frac{2}{3}}\delta_M|f(x)| \tag{9}$$

ergibt sich für den Maschienenfehler F_M der 2-Punktformel:

$$F_M = \sqrt{\frac{2}{3}}\frac{\delta_M}{h}|f(x)| \tag{10}$$

Maschinenfehler der 3-Punktformel

Analog zur 2-Punktformel ergibt sich mit,

$$\Delta M(f(x+h) - f(x-h)) = \sqrt{\frac{2}{3}}\delta_M|f(x)| \tag{11}$$

der Maschienenfehler F_M durch:

$$F_M = \sqrt{\frac{1}{6}}\frac{\delta_M}{h}|f(x)| \tag{12}$$

Gesammtfehler

Somit ergibt sich der Gesamtfehler der 2-Punktformel als:

$$\delta_{Ges} = \sqrt{\frac{2}{3}}\frac{\delta_M}{h}|f(x)| + \frac{h}{2}f''(x_i) \tag{13}$$

bzw. in unserem Fall:

$$\delta_{Ges} = (\sqrt{\frac{1}{6}}\frac{\delta_M}{h} + \frac{h}{2})e^x \tag{14}$$

Ähnilch ergibt sich für den Gesamtfehler der 3-Punktformel:

$$\delta_{Ges} = \sqrt{\frac{1}{6}}\frac{\delta_M}{h}|f(x)| + \frac{h^2}{6}f'''(x_i) \tag{15}$$

bzw. in unserem Fall:

$$\delta_{Ges} = (\sqrt{\frac{1}{6}}\frac{\delta_M}{h} + \frac{h^2}{6})e^x \tag{16}$$

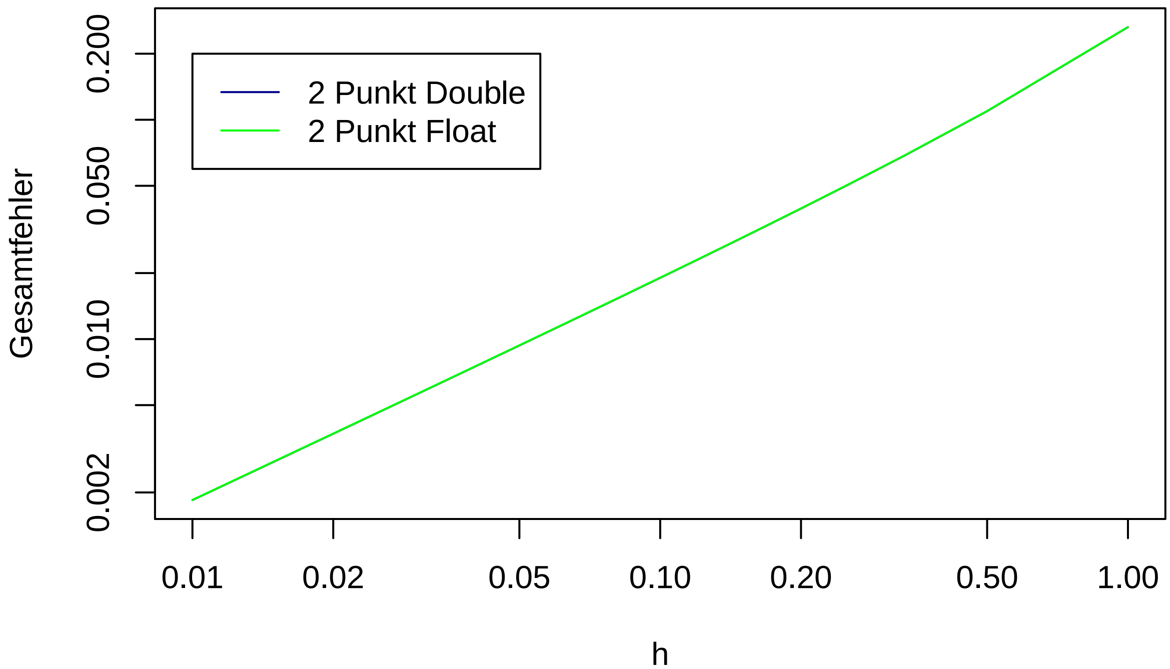
Um dieses Verhalten zu bestätigen werden die Gesamtfehler im späteren Verflauf an die aufgetragenen Fehlerverläufe gefittet. Dabei wird für float (single precision) eine Maschienengenauigkeit von $\delta_{Mf} \approx 6*10^{-8}$ und für double (double precision) eine von $\delta_{Md} \approx 1,1*10^{-16}$ angenommen [2].

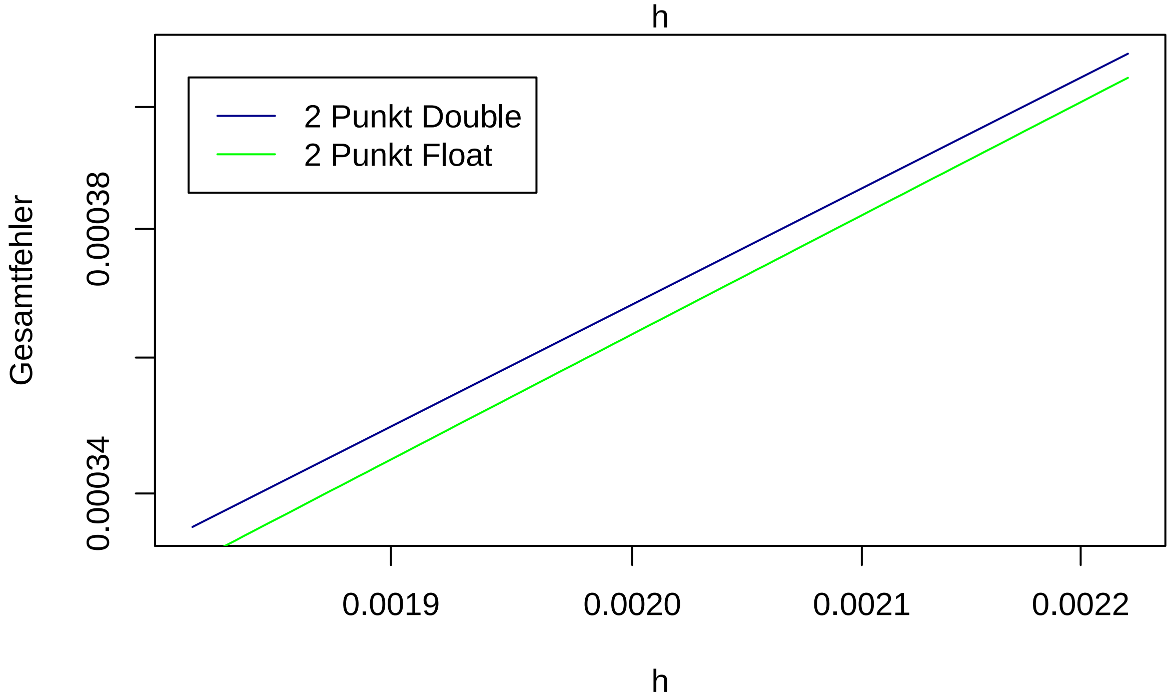
Beispiel Anhand der Differentation

Der 2-Punkt Ansatz

Am Beispiel der Differentation der Funktion $f(x) = e^x$ an der Stelle x = -1 sollen die Fehlertypen aufgezeigt werden. Wir betrachten zuerst die 2-Punktformel und den Unterschied zwischen einem Ansatz, der `Double` und einem, der `Floats` verwendet. Da der Code bis auf den Typwechsel identisch ist, liegt hier nur der Double-Code vor:

```
#include <Rcpp.h>
#include <math.h>

using namespace Rcpp;

//[[Rcpp::export]]
Rcpp::List diff2PunktDouble(const double x, const int r){
  // Array der ersten 100 Werte:
    Rcpp::NumericVector xValue(100);
    Rcpp::NumericVector yValue(100);
  // Quelltext
    for (int i = r; i<=99+r; i++){
      xValue[i-r] = 1./i;
      yValue[i-r] = (exp(x+1./i) - exp(x))/(1./i);
    }
  // Rückgabe für eine grafische Wiedergabe
    return List::create(Named("x") = xValue, Named("y") = yValue);
}
```





Während für h in der Größenortnung 1 bis 0,01 keine sichtbaren Differenzen zwischen den Gesammtfehlern hervorruft, kann in der nächst kleineren Größerenordnung eine Divergenz der beiden Fehlergrößen beobachtet werden.

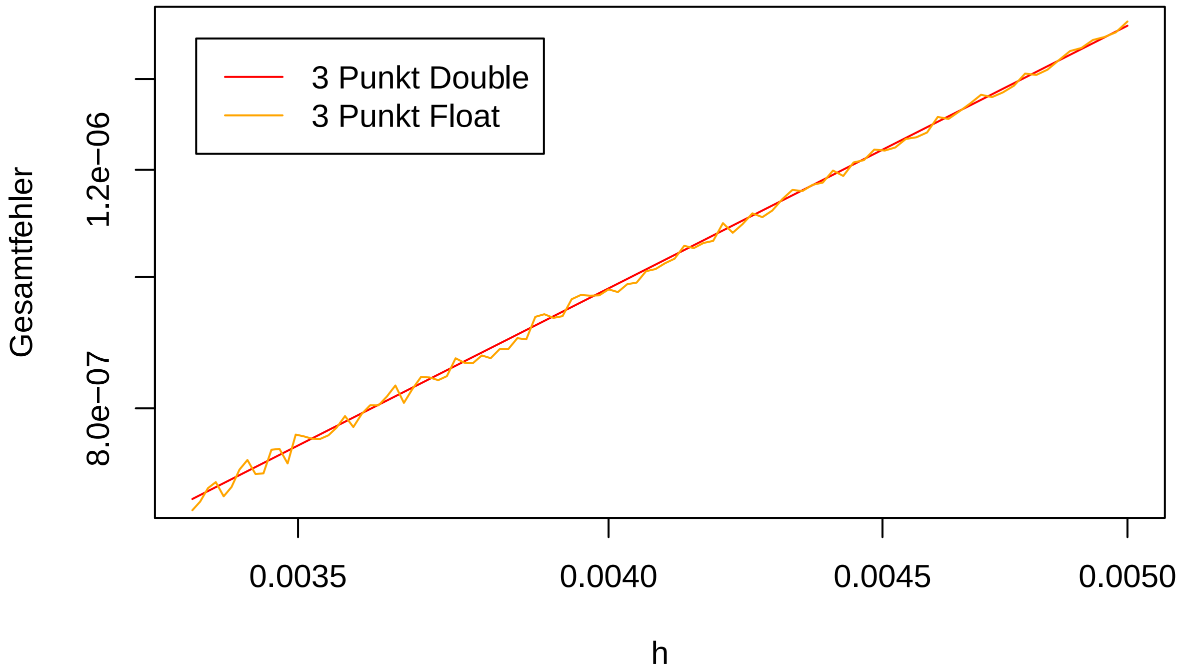
Der 3-Punkt Ansatz

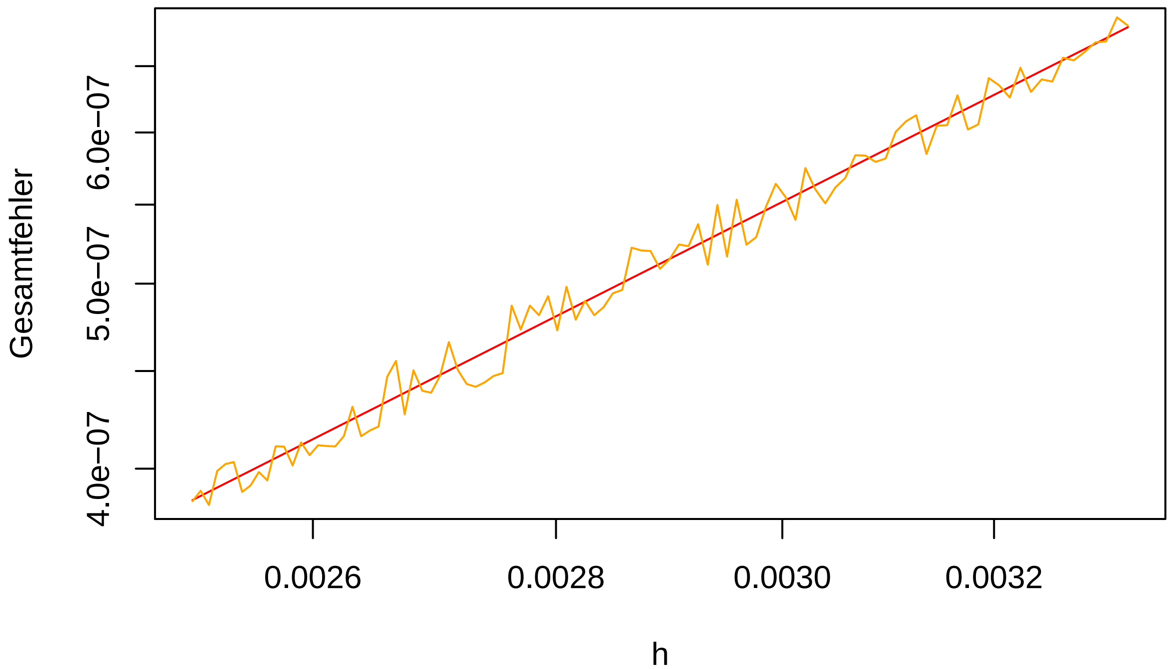
Um das 2-Punktverfahren auch mit einem anderen Verfahren vergleichen zu können, verwenden wir das 3-Punktverfahren. Erneut implementieren wir es einmal mit `Double` und einmal mit `Float`-Precision.

```
#include <Rcpp.h>
#include <math.h>
```

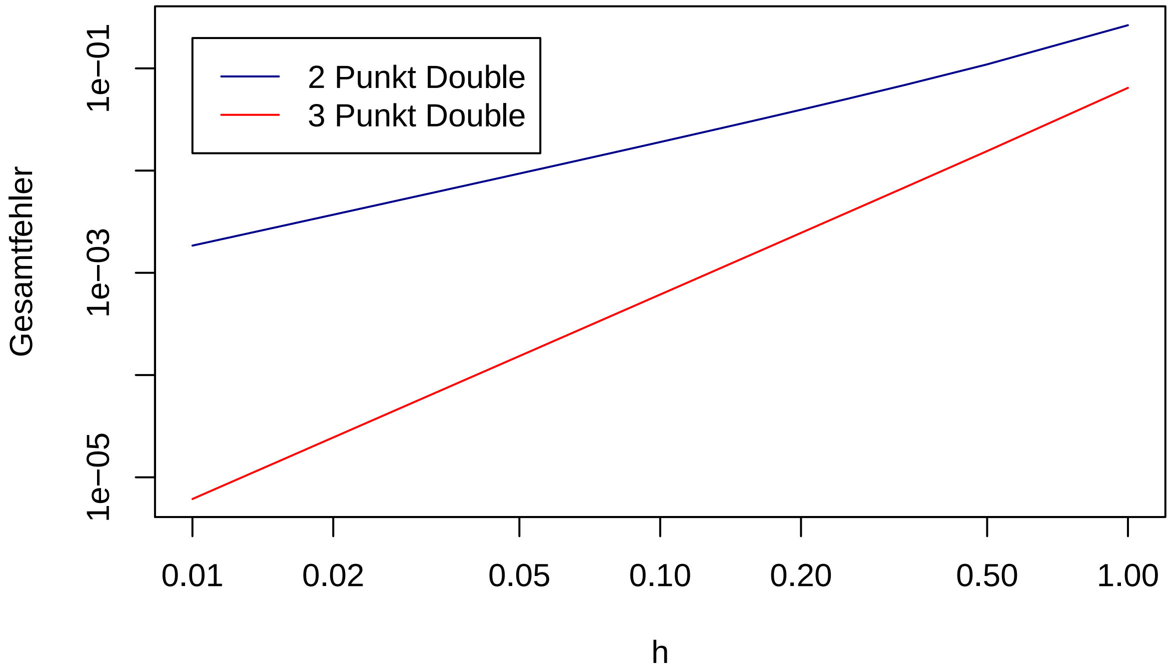
```
using namespace Rcpp;

//[[Rcpp::export]]
Rcpp::List diff3PunktDouble(const double x, const int r){
  // Array der ersten 100 Werte:
    Rcpp::NumericVector xValue(100);
    Rcpp::NumericVector yValue(100);
  // Quelltext
    for (int i = r; i<=99+r; i++){
      xValue[i-r] = 1./i;
      yValue[i-r] = (exp(x+1./i) - exp(x-1./i))/(2./i);
    }
  // Rückgabe für eine grafische Wiedergabe
    return List::create(Named("x") = xValue, Named("y") = yValue);
}
```

Trägt man nun die beiden Graphen grafisch auf so findet man für h's bei um die 0,005 die ersten Auffälligen Unterschiede:





Während der `Double`-Graph eine glatte Gerade beschreibt, springt der `Float`-Graph wahllos wirkend über und unter diese Gerade. Dieses Zittern wird mit sinkenden h immer extremer und verdeutlicht, dass die Unterschiede der Näherungswerte die Größenordnung des Maschinenfehlers erreichen. Um diese Zitterbewegung auch am `Float`-Graphen zu entdecken, muss das h in der Größenordnung von 10^{-4} betrachtet werden.



Literatur

[1] C. Urbach, *Vorlesungsskript Computerphysik*,Seite 12,2021.

[2] Wikipedia, Artikel: https://de.wikipedia.org/wiki/Maschinengenauigkeit, 15.05.2021.