Abgabe 2 Verfahrens- und Maschinenfehler

Wayne Ströbel, Silas Müller

5/5/2021

Verfahrens- und Maschinenfehler

Wir wollen anhand des Beispiels der numerischen Differentiation die bei praktischen Anwendungen häufig gegenläufig auftretenden Einflüsse von Verfahrens- und Maschinenfehler empirisch untersuchen. Wir betrachte die Ableitung der folgenden Testfunktion:

$$f(x) = e^x \tag{1}$$

an der Stelle $x = -1$.

Zwei-Punkt-Formel

Zuerst wollen wir die Ableitung mit der Zwei-Punkt-Formel bestimmen:

$$f'(x_i) = \frac{f(x_i + h) - f(x_i)}{h} \tag{2}$$

mit $x_i = -1$ und h wählbar.

Diese Formel können wir nun wie folgt in C implementieren:

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

//[[Rcpp::export]]
float PPAbleitung_expf(float x, float g){
  float z;
  float f;
  for (float h=1; h>g;h=h-g){
    z=(exp(x+h)-exp(x))/h;
    f=z-exp(x);
  }
  return z;
}

//[[Rcpp::export]]
double PPAbleitung_expd(double x, double g){
  double z;
```

```
  double f;
  for (double h=1; h>g;h=h-g){
    z=(exp(x+h)-exp(x))/h;
    f=z-exp(x);
  }
  return z;
}

//[[Rcpp::export]]
double PPPAbleitung_expd(double x, double g){
  double z;
  double f;
  for (double h=1; h>g;h=h-g){
    z=1/(2*h)*(exp(x+h)-exp(x-h));
    f=z-exp(x);
    }
 return z;
}

//[[Rcpp::export]]
float PPPAbleitung_expf(float x, float g){
  float z;
  float f;
  for (float h=1; h>g;h=h-g){
    z=1/(2*h)*(exp(x+h)-exp(x-h));
    f=z-exp(x);
  }
  return z;
}
```

```
x_i = -1
h = 0.0001

ff = PPAbleitung_expf(x_i, h)
dd = PPAbleitung_expd(x_i, h)
fff= PPPAbleitung_expf(x_i,h)
ddd= PPPAbleitung_expd(x_i,h)
ff
```

```
## [1] 0.3678177
```

```
dd
```

```
## [1] 0.3678978
```

```
fff
```

```
## [1] 0.3678177
```

```
ddd
```

```
## [1] 0.3678794
```