

Differentialrechnung

September 26, 2022

0.1 Einige Hinweise zur Differentialrechnung am Computer

Notebook erstellt am 03.09.2022 von C. Rockstuhl, überarbeitet von Y. Augenstein

In der Erarbeitung habe ich mich an den Beispielen orientiert, die im Skript *Einführung in die mathematische Behandlung der Naturwissenschaften* von Prof. Dr. H. Ebert enthalten sind.

Dieses Notebook soll in aller Kürze zeigen, wie einige Differenzialrechnungen effektiv am Computer durchgeführt werden können.

Eine klassische Bibliothek, welche das symbolische Rechnen in Python erlaubt, ist **sympy**. Diese stellt viele Routinen bereit. Eine Übersicht über die Funktionalität finden Sie z.B. hier [Overview over Sympy](#). Beachten Sie bitte, für symbolisches Rechnen können Sie auch Programmpakete wie Mathematica oder Maple verwenden.

Wir beginnen als Erstes mit dem Importieren von Funktionen und wir generieren uns einige Symbole, mit denen wir im Folgenden symbolisches Rechnen durchführen:

```
[1]: from sympy import limit, sin, cos, tan, exp, log
      # Wir importieren hier wieder als Erstes aus
      ↳ der erwähnten Bibliothek sympy eine
      # Reihe von Elementarfunktionen, mit denen
      ↳ wir dann später rechnen können.

from sympy.abc import x, d, n # Wir müssen auch einige Variablen
      ↳ definieren, mit denen wir dann symbolisch
      # arbeiten wollen.
      # Im Speziellen benötigen wir x, weil unsere
      ↳ Funktionen üblicherweise von x
      # abhängen, d als eine kleine Größe die wir
      ↳ dann für die Grenzwertbildung
      # benötigen und abschließend n, welches wir
      ↳ als eine Potenz benötigen.
```

Anschließend berechnen wir die Ableitung einer Funktion als Grenzwert eines Differenzenquotienten. Diesen haben wir in der Vorlesung kennengelernt und würden den also gerade definieren als

$$\frac{df}{dx}(x) = \lim_{\Delta x \rightarrow 0} = \frac{f(x + \Delta x) - f(x)}{\Delta x}$$

und genau diese Größe würden wir berechnen lassen. Zunächst einmal nur am einfachen Beispiel von $f(x) = \sin(x)$.

```
[2]: limit((sin(x + d) - sin(x)) / d, d, 0) # Die Funktion limit berechnet uns
      ↪ einfach den Grenzwert. Das erste Argument
      # ist die Funktion, dessen Grenzwert
      ↪ evaluiert werden soll. Und das zweite
      # und dritte Argument geben dann
      ↪ Information über die Variabel, deren
      # Grenzwert betrachtet werden soll.
```

[2]: $\cos(x)$

Das Ergebnis entspricht gerade genau unserer Erwartung, d.h., $f'(x) = \cos(x)$. Wir können das auch mit anderen Funktionen erproben.

```
[3]: limit((exp(x + d) - exp(x)) / d, d, 0)
```

[3]: e^x

```
[4]: limit(((x + d) ** n - x**n) / d, d, 0)
```

[4]: nx^{n-1}

Die folgenden Beispiele sind etwas komplexer, so dass wir das Programm forcieren müssen, eine einfache Formulierung für die entsprechenden Ausdrücke zu finden. Daher würden wir aus der Bibliothek sympy die Funktion simplify importieren, die die entsprechenden Ausdrücke vereinfachen soll.

```
[5]: from sympy import simplify

      simplify(limit((tan(x + d) - tan(x)) / d, d, 0))
```

[5]: $\frac{1}{\cos^2(x)}$

```
[6]: simplify(limit((sin(x + d) / cos(x + d) - sin(x) / cos(x)) / d, d, 0))
```

[6]: $\frac{1}{\cos^2(x)}$

Wem diese Grenzwertbildung zu anstrengend ist, kann Funktionen mittels sympy auch direkt differenzieren. Wir müssen die entsprechende Funktion aus der Bibliothek einbinden.

```
[7]: from sympy import diff
```

und können dann direkt einige sehr ähnliche Beispiele lösen.

```
[8]: diff(sin(x), x)
```

[8]: $\cos(x)$

```
[9]: simplify(diff(x**n, x))
```

[9]: nx^{n-1}

Wir können auch mit Funktionen rechnen, ohne diese bereits explizit spezifizieren zu müssen. Damit können wir einfache Rechenregeln der Differenziation überprüfen.

```
[10]: from sympy import Function
      from sympy.abc import x

      f = Function("f")(x)          # Hier definieren wir uns einfach eine
      ↪ allgemeine Funktion f die von x abhängt.
      g = Function("g")(x)          # Hier definieren wir uns einfach eine
      ↪ allgemeine Funktion f die von x abhängt.
```

Hier hätten wir die Summenregel:

```
[11]: diff(f + g, x)
```

[11]: $\frac{d}{dx}f(x) + \frac{d}{dx}g(x)$

Hier hätten wir die Produktregel:

```
[12]: diff(f * g, x)
```

[12]: $f(x)\frac{d}{dx}g(x) + g(x)\frac{d}{dx}f(x)$

Die Quotientenregel:

```
[13]: simplify(diff((f / g), x))
```

[13]: $\frac{-f(x)\frac{d}{dx}g(x) + g(x)\frac{d}{dx}f(x)}{g^2(x)}$

Oder auch die Kettenregel:

```
[14]: f = Function("f")(g)
      diff(f, x)
```

[14]: $\frac{d}{dg(x)}f(g(x))\frac{d}{dx}g(x)$

Wir können auch einfach Ableitungen höherer Ordnung berechnen.

```
[15]: diff(sin(x), x)
```

```
[15]: cos(x)
```

```
[16]: diff(diff(sin(x), x), x)
```

```
[16]: -sin(x)
```

```
[17]: diff(sin(x), x, 2)
```

```
[17]: -sin(x)
```