

MATLAB/ Simulink | Numerisches Differenzieren

Beschreibung

In dieser Hausaufgabe werden Sie den in der Vorlesung aufgesetzten Newton-Algorithmus erweitern. Abzugebende Dateien sind folgende:

- (a) `runMyNewton.m`
- (b) `myNewton.m`
- (c) `myPoly.m`
- (d) `dmyPoly.m`
- (e) `numDiff.m`

Die Dateien a-d sollen interaktiv während der Vorlesung mitprogrammiert werden. Neben der Erstellung von Datei e müssen zur Realisierung der Hausaufgabe zusätzlich die Dateien a und b nochmals bearbeitet werden.

Theorie: Numerisches Differenzieren

Die numerische Approximation der Ableitung einer Funktion $f : \mathbb{R} \rightarrow \mathbb{R}$ an der Stelle x_0 geht auf die Definition des Differenzenquotienten zurück:

$$\left. \frac{df(x)}{dx} \right|_{x_0} := \lim_{h \rightarrow 0} \frac{f(x_0 + h) - f(x_0)}{h} \quad (1)$$

Der Grenzübergang von Sekanten- zu Tangentensteigung $h \rightarrow 0$ kann aufgrund des damit einhergehenden Exponentenüberlaufs der Gleitkommaarithmetik* (auch: overflow) im IEEE-754-Format† numerisch nicht durchgeführt werden. Deshalb wird die Ableitung der Funktion f im Punkt x_0 über eine Sekantensteigung nach Gleichung 2 mit einem endlichen Wert für h approximiert.

$$D^+ f(x_0) \approx \frac{f(x_0 + h) - f(x_0)}{h} \quad (2)$$

*Der Leser entnehme eine Einführung in die Gleitkommaarstellung aus [2].

†IEEE-754 wurde 1985 erstmalig als technischer Standard für binäre Gleitkommazahlen vom Institute of Electrical and Electronics Engineers (IEEE) festgelegt.

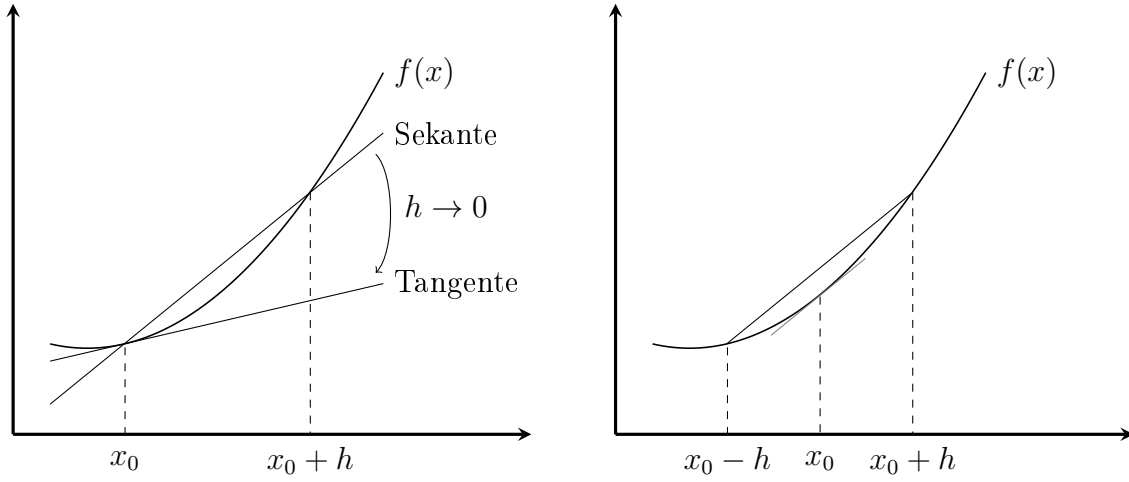


Abbildung 1: Links: Visualisierung des Grenzübergangs $h \rightarrow 0$, rechts: Exemplarische Darstellung der Zentraldifferenz.

$$D^- f(x_0) \approx \frac{f(x_0) - f(x_0 - h)}{h} \quad (3)$$

Für $h > 0$ ($h < 0$) ergibt sich die Vorwärts- (Rückwärts-) Differenzenformel. Es lässt sich zeigen, dass lineare Polynome mit Gleichung 2 bzw. 3 exakt differenziert werden [3]. Mit Hilfe der Taylor-Entwicklung ergibt sich für den Diskretisierungsfehler ein proportionales Verhalten zur verwendeten Schrittweite h [1].

$$f'(x) = \frac{f(x_0 + h) - f(x_0)}{h} - \frac{h}{2} f''(\zeta) \quad (4)$$

Wie in Abbildung 1 illustriert, wird die Genauigkeit der Sekanten-Approximation für ein festes h direkt durch die etwaige nicht-Linearität der Funktion determiniert. Durch Bilden des arithmetischen Mittels von $D^- f(x)$ und $D^+ f(x)$ ergibt sich die Zentraldifferenz.

$$Df(x) \approx \frac{f(x_0 + h) - f(x_0 - h)}{2h} \quad (5)$$

Augenscheinlich werden mit Gleichung 5 Polynome bis zum Grad zwei exakt differenziert [3]. Der Diskretisierungsfehler skaliert quadratisch mit h [1].

$$f'(x) = \frac{f(x_0 + h) - f(x_0 - h)}{2h} - \frac{h^2}{6} f'''(\zeta) \quad (6)$$

Dem Diskretisierungsfehler, welcher für $h \rightarrow 0$ minimiert wird, wirken aufgrund von Rundungsfehlern Auslöschungseffekte, welche für $h \rightarrow 0$ maximal werden, entgegen. Für eine mit Rundungsfehlern ϵ behaftete Funktionsauswertung $\tilde{f}(x)$ gilt $|\tilde{f}(x) - f(x)| \leq \epsilon$.

Dabei ergeben sich folgende Abschätzungen für das Verhalten der Rundungsfehler für Vorwärtsdifferenzen:

$$\begin{aligned}\delta &= \frac{f(x_0 + h) - f(x_0)}{h} \\ \tilde{\delta} &= \frac{\tilde{f}(x_0 + h) - \tilde{f}(x_0)}{h} \\ |\delta - \tilde{\delta}| &= \frac{1}{h} \left[\left(f(x_0 + h) - \tilde{f}(x_0 + h) \right) - \left(f(x_0) - \tilde{f}(x_0) \right) \right] \leq \frac{2}{h} \epsilon\end{aligned}\quad (7)$$

und für Zentralkdifferenzen:

$$\begin{aligned}\Delta &= \frac{f(x_0 + h) - f(x_0 - h)}{2h} \\ \tilde{\Delta} &= \frac{\tilde{f}(x_0 + h) - \tilde{f}(x_0 - h)}{2h} \\ |\Delta - \tilde{\Delta}| &= \frac{1}{2h} \left[\left(f(x_0 + h) - \tilde{f}(x_0 + h) \right) - \left(f(x_0 - h) - \tilde{f}(x_0 - h) \right) \right] \\ &\leq \frac{1}{h} \epsilon\end{aligned}\quad (8)$$

Damit ergibt sich für Vorwärtsdifferenzen ein Gesamtfehler nach Gleichung 9a mit einem Minimum nach Gleichung 9b.

$$F_{\text{Vor}}(h) \leq \frac{2}{h} \epsilon + c_1 \cdot h \quad (9a)$$

$$h_{\text{opt,Vor}} = \sqrt{\frac{2\epsilon}{c_1}} \quad (9b)$$

Für Zentralkdifferenzen ergibt sich ein Gesamtfehler nach Gleichung 10a mit dem Minimum nach Gleichung 10b.

$$F_{\text{Zen}}(h) \leq \frac{1}{h} \epsilon + c_2 \cdot h^2 \quad (10a)$$

$$h_{\text{opt,Zen}} = \sqrt[3]{\frac{\epsilon}{2c_2}} \quad (10b)$$

Dabei sind c_1, c_2 in der Praxis nicht ohne Weiteres bestimmbar und skalieren bei Vorwärtsdifferenzen mit der zweiten Ableitung (bei Zentralkdifferenzen mit der dritten Ableitung) an der Stelle x_0 . Der Einfluss der Konstanten auf h_{opt} wird jeweils mit der zweiten und dritten Wurzel abgeschwächt. Für eine erste Abschätzung der optimalen Schrittweite sei $c_1 \approx c_2 \approx 10^0$ gewählt. Bei einer vorliegenden Maschinengenauigkeit von $\epsilon \approx 10^{-16}$ ergeben sich dann für die optimalen Schrittweiten $h_{\text{opt,Vor}} \approx 10^{-8}$ und $h_{\text{opt,Zen}} \approx 10^{-6}$.

Aufgabe 1: Vorwärtsdifferenzen

Ziel ist es den Newton-Aufruf mit lediglich dem Übergabeparameter '*Function*' zu ermöglichen. Gehen Sie dabei wie folgt vor:

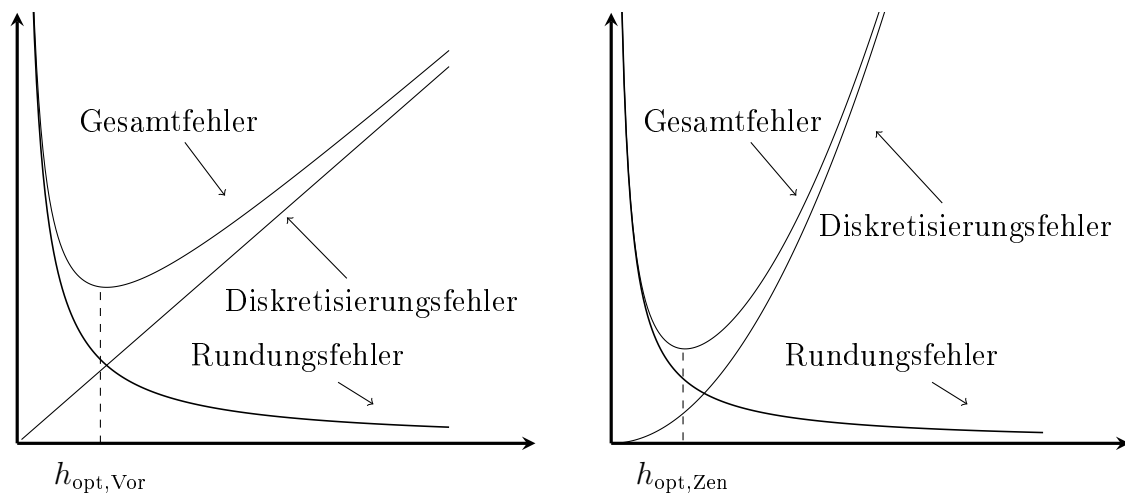


Abbildung 2: Visualisierung von Diskretisierungs-, Rundungs- und Gesamtfehler für Vorwärtsdifferenzen (links) und Zentraldifferenzen (rechts).

- Schreiben Sie eine Funktion `numDiff(func,x)`, welche von einer übergebenen Funktion `func` die numerische Ableitung an einer Stelle `x` mittels Vorwärtsdifferenzen approximiert.
- Implementieren Sie in Ihrer Newton-Funktion eine Routine, welche erkennt, wenn beim Aufruf **keine** Funktion `'Derivative'` als analytische Ableitung der zu untersuchenden Funktion `'Function'` übergeben wurde.
- Verwenden Sie dann die geschriebene Funktion `numDiff(func,x)` zur Berechnung der Ableitung in Ihrem Newton-Algorithmus.

Aufgabe 2: Weitere Differenzierungsverfahren & User Input

Ziel ist es dem user eine Auswahl von verschiedenen numerischen Verfahren für die Approximation der Ableitung zur Verfügung zu stellen. Gehen Sie dabei wie folgt vor:

- Erweitern Sie Ihre `numDiff(func,x,Method)`-Funktion um einen zusätzlichen Parameter `Method`. In Abhängigkeit dieses Parameters soll intern die Berechnung der Ableitung mittels:
 1. Vorwärtsdifferenzen ($h = 10^{-8}$)
 2. Rückwärtsdifferenzen ($h = 10^{-8}$)
 3. Zentraldifferenzen ($h = 10^{-6}$)
 erfolgen.
- Implementieren Sie in der Newton-Funktion eine user Abfrage, welche das zu verwendende Differenzierungsverfahren determiniert, um die Ableitung zu approximieren, wenn die Newton-Funktion ohne Parameter `'Derivative'` aufgerufen wird. Schlagen Sie dazu den Befehl `questdlg` nach und verwenden Sie diesen.

Bewertungskriterien

Sie können mit dieser Abgabe 5 Punkte erreichen. Sie erhalten zunächst für die Umsetzung, welche die Funktionalität des Programms sicherstellt 60% der maximal erreichbaren Punkte. Darüber hinaus, entfallen 40% auf Stil und performance-orientierte Implementierung. Halten Sie sich dazu, wie in der Vorlesung besprochen, an ein Minimum an Clean-Code Regeln. Verwenden Sie für Ihre Dateien ein m-File Header Template wie sie etwa [hier](#) zu finden sind.

Literatur

- [1] W. Dahmen und A. Reusken. *Numerik für Ingenieure und Naturwissenschaftler*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2008. ISBN: 9783540764939. URL: <https://books.google.de/books?id=d8MfBAAQBAJ>.
- [2] Thomas Richter und Thomas Wick. “Einleitung”. In: *Einführung in die Numerische Mathematik: Begriffe, Konzepte und zahlreiche Anwendungsbeispiele*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, S. 1–32. ISBN: 978-3-662-54178-4. DOI: [10.1007/978-3-662-54178-4_1](https://doi.org/10.1007/978-3-662-54178-4_1). URL: https://doi.org/10.1007/978-3-662-54178-4_1.
- [3] T. Westermann. *Mathematik für Ingenieure: Ein anwendungsorientiertes Lehrbuch*. Springer-Lehrbuch. Springer Berlin Heidelberg, 2011. ISBN: 9783642127601. URL: <https://books.google.de/books?id=3bYoBAAQBAJ>.