Übungen zu Physik auf dem Computer SS 2012

Übungsblatt 7: Nullstellen, numerische Differentiation und Integration

1. Juni 2012

Allgemeine Hinweise

- Abgabetermin ist **Montag, 11.6.2012, 13:00**
- Zur Abgabe schickst Du die Lösungsdatei(en) im Anhang einer Email an Deinen Tutor:
  - Florian (`floh@icp.uni-stuttgart.de`; Dienstag, 15:45–17:15)
  - Dominic (`dominic@icp.uni-stuttgart.de`; Dienstag, 15:45–17:15)
  - Olaf (`olenz@icp.uni-stuttgart.de`; Mittwoch, 15:45–17:15)
- Die Übungen werden in Gruppen von jeweils zwei oder drei Leuten bearbeitet. Diese dürfen sich gerne von Blatt zu Blatt unterscheiden. Aus formalen Gründen muss allerdings jeder von Euch eine eigene Lösung abgeben. Schreibt bitte auf die Lösungen, mit wem Ihr zusammengearbeitet habt, um uns das Korrigieren zu erleichtern.
- Die Übungen finden statt im CIP-Pool des Instituts für Computerphysik (ICP) im Pfaffenwaldring 27.

Aufgabe 7.1 (4 Punkte): Newtonverfahren höherer Ordnung

Wie in der Vorlesung für höhere Dimensionalität gezeigt, kann man das Newtonsche Näherungsverfahren zur Nullstellensuche durch eine Taylorentwicklung herleiten. Wenn man die Entwicklung nach dem linearen Glied abschneidet, dann handelt es sich dabei um die im Newtonverfahren gesuchte Tangente. Durch Nullsetzen dieser Tangente erhält man den nächsten Näherungswert.

Um eine Näherung höherer Ordnung zu erhalten, kann man die Taylorentwicklung später abschneiden. Verwendet man ein weiteres Glied und nähert so eine Parabel an die Funktion an, erhält man das sogenannte parabolische Halleyverfahren.

- 7.1.1 (2 Punkte) Leite die Formel für das parabolische Halleyverfahren aus der Taylorentwicklung her. Setze die Herleitung entweder in LaTeX (gut), oder scanne Deine handschriftliche Herleitung (weniger gut) und gib sie als Lösung ab.

  Hinweis Eine Parabel hat zwei Nullstellen. Gesucht ist die mit der kleineren Korrektur.

- 7.1.2 (1 Punkt) Das Pythonskript `/share/Courses/PC2012/07/find_roots.py` ist die Musterlösung vom letzten Übungsblatt zur Nullstellensuche. Erweitere das Skript um eine Funktion, die die Nullstelle mit Hilfe des parabolischen Halleyverfahrens annähert und vergleiche die Konvergenzgeschwindigkeit des Verfahrens mit den anderen Verfahren. Gib das Skript als Lösung ab.
- 7.1.3 (1 Punkt) In welchen Fällen konvergiert diese Methode *nicht* wesentlich schneller als das einfache Newtonverfahren? Schreibe die Lösung in die Lösungsemail.

Aufgabe 7.2 (2 Punkte): Numerisches Differenzieren

Ein Nachteil des Newtonverfahrens ist die Notwendigkeit der analytischen Ableitung. Allerdings habt Ihr in der Vorlesung gelernt, dass man eine Funktion auch numerisch annähern kann.

Ein weiteres Verfahren zum Suchen von Nullstellen kann man also erzeugen, indem man im Newtonverfahren die analytische Ableitung der Funktion durch den Differenzenquotienten ersetzt. Dabei können die beiden letzten Näherungen zur Annäherung der Ableitung dienen.

- 7.2.1 (1 Punkt) Erweitere das Pythonskript `/share/Courses/PC2012/07/find_roots.py` um das neue Verfahren und vergleiche die Konvergenzgeschwindigkeit des Verfahrens mit den anderen Verfahren. Gib das Skript als Lösung ab.
- 7.2.2 (1 Punkt) Das Verfahren ist unter einem anderen Namen bekannt. Unter welchem? Schreibe die Antwort in die Lösungsemail.

Aufgabe 7.3 (4 Punkte): Numerische Integration eines Pendels

Das mathematische Pendel der Länge l mit der Anfangsauslenkung θ_0 ist hinreichend bekannt. Es folgt der Differentialgleichung

$$\ddot{\theta} = -\frac{g}{l}\sin(\theta) \tag{1}$$

wobei g die Gravitationsbeschleunigung ist. Die Gleichung ist analytisch lösbar, wenn man annimmt, dass der Sinus linear ist. Diese Näherung stimmt aber nur für sehr kleine Winkel.

Durch einige Umformungen, die in `/share/Courses/PC2012/07/pendelintegration.pdf` genauer beschrieben sind, können wir trotzdem eine Formel angeben, die aus der Differentialgleichung die Schwingungsdauer bestimmt, ohne sie direkt zu lösen. Wir erhalten

$$\begin{aligned} T &= 4\sqrt{\frac{l}{g}}K\left(\sin\frac{\theta_0}{2}\right) \quad \text{mit} && (2)\\ K(k) &= \int_0^{\pi/2} \frac{d\xi}{\sqrt{1-k^2\sin^2\xi}} && (3)\end{aligned}$$

- 7.3.1 (1 Punkt) Schreibe eine Pythonfunktion `integrate(f, a, b, N)`, die eine numerische Integration mit Hilfe der Trapezregel durchführt. Zerlege das übergebene Intervall $[a, b]$ dafür gleichmäßig in N Teilintervalle. Gib ein Pythonskript mit der Funktion als Lösung ab.
- 7.3.2 (2 Punkt) Verwende die Funktion aus der vorigen Aufgabe, um die Schwingungsdauer T für 50 linear verteilte Anfangsauslenkungen zwischen 0 und 90 Grad zu berechnen ($l = 1.0$, $g = 9.81$) und plotte die Schwingungsdauer über dem Winkel. Gib das fertige Skript als Lösung ab.

  Hinweis Um `integrate` eine Funktion mit Parameter θ_0 zu übergeben, ist die anonyme `lambda`-Funktion von Nutzen.
- 7.3.3 (1 Punkt) Ab welchem der in 7.3.2 bestimmten Winkel weicht die analytische lineare Näherung um mehr als 5% von der numerischen Lösung ab? Wie groß ist der relative Fehler bei 30 Grad Auslenkung?