Georg-August-Universität Göttingen
Institut für Theoretische Physik
Prof. Dr. M. Müller
Priv.-Doz. Dr. S.R. Manmana
SoSe 2018

Übungen zur Vorlesung
Computergestütztes Wissenschaftliches Rechnen

Blatt 4

Lernziele dieses Übungsblattes

- Numerische Lösung von Geschwindigkeitsabhängigen linearen Differentialgleichungen.
- Algorithmen: Runge-Kutta 2. Ordnung und Leapfrog-Verfahren.

Aufgabe 1 *Prozesse und Systeminformationen*

Viele der Hilfsprogramme, die Ihnen unter dem Terminal zur Verfügung stehen, verfügen über eine Option um die Ausgabe benutzerfreundlicher darzustellen. Die übliche Methode ist der Aufruf dieser Routinen mit der zusätzlichen Option $-\mathrm{h}$ (human readable). Vergleichen Sie die Ausgaben der folgenden Befehle mit und ohne die $-\mathrm{h}$-Option und finden Sie heraus, wozu Sie diese Programme benutzen können.

a) df

b) du

c) ls

In der folgenden Aufgabe werden Sie ein Programm ausführen, das unter Umständen viel Zeit benötigt. Eine typische Herangehensweise ist das Starten eines solchen Programms mit niedriger Priorität, um den Computer für weitere Programmierarbeiten ohne Geschwindigkeitseinbußen benutzen zu können. Die folgenden Schritte können Sie erst durchführen, wenn Sie das Programm aus Aufgabe 2) geschrieben haben.

a) Führen Sie in einem Terminal den Befehl top (oder top $-$u username, ersetzen Sie username mit Ihrem Benutzernamen) aus, während Ihr Programm läuft. Versuchen Sie, die Ausgabe dieses Programms zu verstehen und zu sehen, wie stark CPU und Arbeitsspeicher ausgelastet sind. Es ermoeglich Ihnen auch, die Priorität von laufenden Programmen zu ändern oder sie zu beenden.

b) Drücken Sie die k-Taste (k wie "kill"), und beenden Sie Ihr Programm, indem Sie den Anfragen der top-Eingabemaske folgen.

c) Sie können die Priorität der Prozesse mittels r (r wie "renice") verändern. Alternative dazu können Sie ein Programm mit der niedrigsten Priorität auf der Konsole mittels nice -19 ./... starten.

Hinweis: Sie können alternativ auch das Programm htop benutzen.

Aufgabe 2 *Schallmauer*

Ein raketengetriebenes Auto soll die Schallmauer durchbrechen. Dazu werden wir ein Modell aufsetzen, das die Bewegung des Raketenwagens auf einer Ebene berechnet. Wir gehen von einer konstanten Antriebskraft F und einem geschwindigkeitsabhängigen Windwiderstand $c(v)$ aus, um folgende partielle Differentialgleichung zu lösen

$$M\frac{\mathrm{d}^2}{\mathrm{d}t^2}r = F - F_{\mathrm{Reibung}}\left(\frac{\mathrm{d}}{\mathrm{d}t}r\right) . \tag{1}$$

Dabei ist $r(t)$ die Position und $M = 2.000\,\mathrm{kg}$ die Masse des Autos. Die Reibungskraft des Autos folgt aus

$$F_{\mathrm{Reibung}}(v) = c_w(v)v^2 . \tag{2}$$

Der Strömungswiderstandkoeffizient wurde in einem Windkanal vermessen und wird als Funktion in Abhängigkeit der dimensionslosen Machzahl $m = \frac{v}{v_{\mathrm{Schallgeschwindigkeit}}} = \frac{v}{343\,m/s}$ angegeben

$$c_w(m) = \frac{\mathrm{kg}}{\mathrm{m}}\left(0.227 + 0.120\frac{m^2 - 1.18m + 0.205}{m^3 - 3m + 2.05}\right) . \tag{3}$$

Implementieren Sie zwei alternative Integratoren, die den Leapfrog- und Runge-Kutta-Algorithmus zweiter Ordnung realisieren. Prüfen Sie für verschiedene Zeitschritte $dt = [0.01s, 1s, 100s, 10000s]$ die Qualität der Integratoren, indem Sie die Trajektorien grafisch darstellen. Welchen Vorteil bietet der Leapfrog-Algorithmus?

Hinweise:
i) Interpolieren Sie falls benötigt die Geschwindigkeit linear.
ii) Der Leapfrog-Algorithmus angewendet auf die DGL dieser Aufgabe wird durch folgende Schritte implementiert:

$$r(t + \Delta t) = r(t) + v\left(t + \frac{1}{2}\Delta t\right)\Delta t \tag{4}$$

$$v\left(t + \frac{1}{2}\Delta t\right) = v(t - \frac{1}{2}\Delta t) + a(t)\Delta t. \tag{5}$$

In dieser Umsetzung wird die Geschwindigkeit erst zur Zeit $t + \Delta t/2$ berechnet, diese wird dann für die Position $r(t + \Delta t)$ verwendet. In dieser Form überspringen ("leap") die Geschwindigkeiten die Positionen. Der Nachteil ist, dass die Geschwindigkeiten nicht zum gleichen Zeitpunkt wie die Positionen bekannt sind, allerdings lassen sich diese über

$$v(t) = \frac{1}{2}\left[v\left(t - \frac{1}{2}\Delta t\right) + v\left(t + \frac{1}{2}\Delta t\right)\right] \tag{6}$$

ausrechnen.

a) Bestimmen Sie die minimale Antriebskraft, die nötig ist, um mit diesem Auto die Schallmauer zu durchbrechen, indem sie die Kraft variieren und die Endgeschwindigkeit des Wagens durch das Programm bestimmen lassen. Geben Sie auch die zurückgelegte Strecke an, die zum Erreichen der Schallgeschwindigkeit nötig war.

b) Erstellen Sie eine Tabelle der Endgeschwindigkeit als Funktion der Antriebskraft und stellen Sie diese mittels gnuplot dar.

Aufgabe 3 *Wiederholung*

Für das selbstständige Erarbeiten des Projektes am Ende des Semesters ist es wichtig, dass Sie mit den Grundkonzepten des Erstellens eines Programms vertraut sind. Bitte schauen Sie sich die Übungsaufgaben der letzten Zettel noch einmal an und fragen Sie Ihre Tutoren, wenn Sie bei der Lösung und Verständnis der Aufgaben und/oder beim Verständnis der Lösungswege Hilfe benötigen.

Selbsttest

- Welche numerischen Verfahren bzw. Algorithmen kennen Sie, um gewöhnliche Differentialgleichungen zu lösen?
- Welche Vorteile besitzen Runge-Kutta und Leapfrog-Verfahren gegenüber dem Euler-Cauchy-Algorithmus?
- Für welche Situationen würden Sie jeweils Runge-Kutta, Leapfrog oder Euler-Cauchy nutzen? Welche Möglichkeiten hat man jeweils um den numerischen Fehler kleiner zu machen (bzw. zu kontrollieren)?
- Bis zu welcher Ordnung geht man typischerweise bei Runge-Kutta-Verfahren? Ggf. im Internet recherchieren.