Rechnernutzung in der Physik

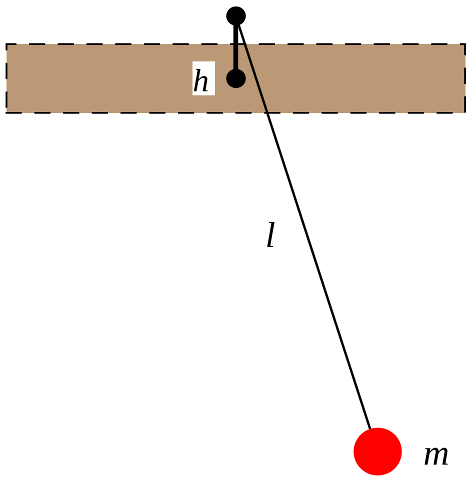
Institut für Experimentelle Teilchenphysik
Institut für Theoretische Teilchenphysik
Interfakultatives Institut für Anwendungen der Informatik

Dr. Th. Kuhr, Prof. Dr. M. Steinhauser, Prof. Dr. U. Husemann
Mildenberger / Hoff / Hermann / Heck

WS2012/13 – Blatt 08
`http://comp.physik.kit.edu`
Prog: Di, 11.12.2012 / Ausarb: Fr, 14.12.2012

Aufgabe 18: „Upside-down“ Pendel **Programmtestat und Ausarbeitung**

Betrachten Sie ein mathematisches Pendel der Länge l im Schwerefeld der Erde, das in der x-y-Ebene schwingt, wobei die Ruhelage der schwingenden Masse bei $x = 0$ und $y = -l$ sein soll. Es sollen kleine Auslenkungen des Aufhängepunkts in vertikaler Richtung zugelassen werden, so dass sich für die Koordinaten des Massenpunkts folgende Parametrisierung anbietet.



$$\begin{aligned} x(t) &= l \sin\theta(t), \\ y(t) &= h(t) + l \cos\theta(t), \end{aligned}$$

wobei für die Auslenkung des Aufhängepunkts $h(t) = a \sin(\omega t)$ gelten soll. a und ω sind freie Parameter. (Für $t = 0$ und $\theta(0) = 0$ befindet sich der Massenpunkt bei $x = 0$ und $y = +l$.)

(a) Stellen Sie die Lagrange-Funktion $L = T - U$ auf, wobei T die kinetische und U die potentielle Energie ist. Leiten Sie daraus (unter Verwendung von `Mathematica`) mit Hilfe der Euler-Lagrange-Gleichungen die Bewegungsgleichungen ab.

(b) Schreiben Sie ein Modul, das die Länge l, Amplitude a, Frequenz ω, sowie die Anfangswerte $\theta(0)$ und $\dot{\theta}(0)$ als Parameter hat, die Bewegungsgleichungen numerisch löst und mit Hilfe von `Animate` (oder `Manipulate`) die Lösung graphisch animiert darstellt. Verwenden Sie für die Erdbeschleunigung $g = 9.81\,\mathrm{m/s^2}$.

(c) Rufen Sie Ihr Modul mit $l = 10\,\mathrm{m}$, $a = 0.5\,\mathrm{m}$, $\omega = 2\pi\,\mathrm{s}^{-1}$, $\theta(0) = 3.0$ und $\dot{\theta}(0) = 0.01$ auf. Erhalten Sie das erwartete Resultat?

(d) Wählen Sie nun $\omega = 2\pi n\,\mathrm{s}^{-1}$, $\theta(0) = 0$, $\dot{\theta}(0) = 0.01$ mit $n = 1, 2, 3, \ldots, 10$. Für welche Werte von n schwingt das Pendel „auf dem Kopf“, d.h. es führt stabile Schwingungen um $x = 0$ und $y = l$ aus?

Ausarbeitung: Die Ausarbeitung soll den *handschriftlich* kommentierten Quelltext des Moduls aus Teilaufgabe (b) enthalten. Erstellen Sie für die Konfiguration aus Teilaufgabe (d) mit $n = 3$ alle 5 Sekunden eine Momentaufnahme im Zeitraum von 0 bis 30 Sekunden. Exportieren Sie deren gemeinsame Darstellung als **maximal eine** `.pdf`-Seite und drucken Sie diese aus.

Hinweis: Folgende `Mathematica`-Befehle erweisen sich in dieser Aufgabe als nützlich: `D`, `NDSolve`, `Evaluate`, `Graphics`, `Animate`, `GraphicsGrid` und `Export`.

Aufgabe 19: „Game of Life“ **freiwillig**

Zelluläre Automaten (erstmals 1940 von John von Neumann und Stanislav Ulam beschrieben) modellieren Systeme vieler Einzelelemente (den Zellen), die jeweils diskrete Zustände einnehmen können. Die zeitliche Änderung des Zustands einer Zelle wird durch einen Satz von Regeln beschrieben, der diese Zelle zu ihren nächsten Nachbarn in Beziehung setzt. Ein Iterationsschritt (auch Generation genannt) besteht aus der Anwendung der Regeln auf alle Zellen.

Der britische Mathematiker John Conway formulierte 1970 das „Game of Life“ als einen Satz von Regeln für einen zweidimensionalen zellulären Automaten, dessen Zellen nur zwischen den beiden Zuständen lebend (`1`) oder tot (`0`) wechseln können. Diese (recht simplen) Regeln lauten für eine Zelle:

1. Eine lebende Zelle mit weniger als zwei lebenden Nachbarn stirbt (Unterbevölkerung).
2. Eine lebende Zelle mit zwei oder drei lebenden Nachbarn lebt weiter.
3. Eine lebende Zelle mit mehr als drei lebenden Nachbarn stirbt (Überbevölkerung).
4. Eine tote Zelle mit genau drei lebenden Nachbarn wird lebendig (Reproduktion).

In zwei Dimensionen hat eine Zelle insgesamt acht nächste Nachbarn, die diagonal liegenden zählen also dazu. (Das „Game of Life“ ist eine sog. universelle Turing-Maschine oder Turing-vollständig, es können also beliebige Rechenoperationen durch bestimmte Konfigurationen des Zellgitters nachgebildet werden.)

Implementieren Sie die Regeln des „Game of Life“ in Mathematica. Erzeugen Sie eine zufällige Anfangskonfiguration auf einem Raster dessen Größe zu Anfang frei wählbar sein soll. Stellen Sie nun die iterierte Anwendung der Regeln auf das Zellgitter als Animation graphisch dar. Schreiben Sie hierfür zunächst eine Funktion, die die Anzahl lebender Nachbarn einer Zelle ermittelt. Beobachten Sie Gruppierungen von Zellen mit zeitlich statischer oder periodischer Konfiguration? Sie können Ihre Implementierung mit der in `Mathematica` vorhandenen vergleichen. Dazu benötigen Sie den Befehl:

```
CellularAutomaton[{224, {2, {{2, 2, 2}, {2, 1, 2}, {2, 2, 2}}}, {1, 1}},
                  <Anfangskonfiguration>, <Generation>];
```

Es gibt im Internet zahlreiche weiterführende Quellen zu zellulären Automaten und „Game of Life“ sowie ganze Lexika zu dessen „Lebensformen“ mit bestimmten Eigenschaften (Gleiter, Raumschiffe, Kanonen, etc.):

`http://en.wikipedia.org/wiki/Conway's_Game_of_Life`
`http://mathworld.wolfram.com/GameofLife.html`
`http://www.argentum.freeserve.co.uk/lex.htm`

Hinweis: Folgende zusätzliche `Mathematica`-Befehle erweisen sich in dieser Aufgabe als nützlich: `RandomInteger`, `Table`, `Join`, `Total`, `Which`, `Function`, `Module`, `ArrayPlot`, `Dynamic`, `Pause` und `Animator`.

Hinweis: Mit dem Rechnernamen `fphctssh.physik.uni-karlsruhe.de` können Sie von überall aus mittels ssh/scp Programm auf einen Poolrechner zugreifen.