

Praktikumsversuch 3: Weltraum-Simulation

Hardwarenahe Programmierung · Wintersemester 2019/20 · Prof. Dr. Peter Gerwinski

Aufgabe: Schreiben Sie ein C-Programm, das die Bahnen von mindestens 3 Massenpunkten unter Einfluß der Gravitation simuliert und in bewegter Grafik darstellt.

- Zwei Massenpunkte („Himmelskörper“) mit den Massen m_i und m_j an den Orten \vec{r}_i und \vec{r}_j ziehen einander an. Diese Kraft heißt Gravitation. Sie hat den Betrag:

$$|\vec{F}_{ij}| = \frac{m_j \cdot m_i \cdot G}{|\vec{r}_j - \vec{r}_i|^2} \quad (1)$$

Hierbei ist G eine Konstante (Gravitationskonstante).

- Die auf einen Himmelskörper wirkende Gesamtkraft \vec{F}_i ergibt sich als Summe der von allen anderen Himmelskörpern herrührenden Gravitationskräfte:

$$\vec{F}_i = \sum_{j=0, j \neq i}^{N-1} \vec{F}_{ij} \quad (2)$$

- Die Gravitationskraft beschleunigt jeden Himmelskörper gemäß:

$$\vec{F}_i = m_i \cdot \vec{a}_i \quad (3)$$

- Beispiel: Wir betrachten zwei Himmelskörper. Einer davon („Zentralgestirn“) ruht im Zentrum ($\vec{r}_0 = 0$, $\vec{v}_0 = 0$) und hat eine wesentlich größere Masse als der andere („Satellit“, $m_1 \ll m_0$). Mit geeignetem Anfangsort \vec{r}_1 und geeigneter Anfangsgeschwindigkeit \vec{v}_1 beschreibt dann der Satellit eine elliptische Umlaufbahn um das Zentralgestirn.
- Wir rechnen in zwei Dimensionen x und y .
- Für die Zerlegung einer Kraft \vec{F}_{ij} in x - und y -Komponenten benötigen Sie nur die Grundrechenarten und die Wurzelfunktion, jedoch insbesondere *keine* trigonometrischen Funktionen:

$$\vec{F}_{ij} = |\vec{F}_{ij}| \cdot \frac{\vec{r}_j - \vec{r}_i}{|\vec{r}_j - \vec{r}_i|} \quad (4)$$

- Die Wurzelfunktion `sqrt()` finden Sie in der Mathematik-Bibliothek. Um diese zu nutzen, verwenden Sie `#include <math.h>` im Quelltext, und geben Sie beim `gcc`-Aufruf `-lm` mit an.

- Für die Simulation betrachten wir das System in kurzen Zeitintervallen dt und berechnen die Änderungen des Ortes $\vec{r}_i = (x_i, y_i)$ und der Geschwindigkeit $\vec{v}_i = (v_{xi}, v_{yi})$ jedes Himmelskörpers mit Hilfe des expliziten Eulerschen Polygonzugverfahrens.

(Wer möchte, darf natürlich auch andere Verfahren anwenden, beispielsweise das klassische Runge-Kutta-Verfahren 4. Ordnung.)

- Für eine derartige Simulation einschließlich ihrer Darstellung als bewegte Grafik können Sie sich von dem Beispiel-Programm `gtk-16.c` inspirieren lassen. (Compilieren mit: `gcc -Wall -O $(pkg-config --cflags --libs gtk+-3.0) gtk-16.c -o gtk-16`)

- In einer `GTK+-drawing_area` liegt der Nullpunkt der Zeichnung oben links, eine Längeneinheit entspricht einem Pixel, und die y -Koordinate wächst nach unten. Es empfiehlt sich, die Koordinaten so umzurechnen, daß der Nullpunkt in der Mitte der Zeichnung liegt, die Längeneinheit Ihrem persönlichen Geschmack entspricht und die y -Koordinate nach oben wächst.

- Beispiel-Szenarien für 3 oder mehr Körper:
 - Planet mit Mond umkreist Sonne
 - Sonne mit mehreren Planeten, die sich gegenseitig beeinflussen
 - zwei Sonnen umkreisen sich gegenseitig, Planet kreist drumherum
 - Raumsonde besucht nacheinander mehrere Planeten

Viel Erfolg!

Stand: 10. Dezember 2019

Copyright © 2014, 2015, 2018, 2019 Peter Gerwinski

Lizenz: CC-by-sa (Version 3.0) oder GNU GPL (Version 3 oder höher)

Sie können diese Praktikumsunterlagen einschließlich \LaTeX -Quelltext herunterladen unter:

<https://gitlab.cvh-server.de/pgerwinski/hp>