

Übungsblatt 5

Ausgabe: 26.5.

Abgabe: 9.6.

In diesem Aufgabenblatt wollen wir nun die Parameter des in Aufgabe 4 modellierten N -Körper-Problems anpassen. Dieses Problem ist in ähnlicher Art immer dann gegeben, wenn viele unabhängige Einheiten Wirkungen aufeinander ausüben. Es können etwa N Rechner sich gegenseitig beeinflussen, oder N Neuronen, oder N Gene, usw.

Es ist bekannt, dass bei N Körpern bereits für $n \geq 3$ die zugehörigen Differentialgleichungen im Allgemeinen keine geschlossene analytische Lösung mehr besitzen, mit denen sich die Bewegungsbahnen der beteiligten Körper beschreiben lassen. Bis auf Spezialfälle können sie in der Regel nur noch numerisch bestimmt werden.

Im vorliegenden Fall wollen wir einmal die Veränderung der Position der einzelnen Körper aufgrund ihrer gravitativen Wechselwirkung modellieren und uns dabei das Problem der Parameterschätzung des resultierenden Modells etwas genauer ansehen. Es wird sich herausstellen, dass wir dabei auch gleichzeitig ein recht interessantes Problem lösen können.

Zur Erinnerung: Es gibt mehrere Himmelskörper K_i unterschiedlicher Masse im Universum, weitab jeder Galaxis. Sie haben einen momentanen Zustand (Masse m_i mit Position \mathbf{p}_i im Raum, sind in Bewegung mit der Geschwindigkeit \mathbf{v}_i sowie der Beschleunigung \mathbf{a}_i) und üben dabei Kräfte \mathbf{F}_i auf einander aus.

Dabei erhielten wir mit

$$\vec{F}_i = \sum_{j=1, j \neq i}^n \vec{F}_{ij} = \sum_{j=1, j \neq i}^n G \cdot \frac{m_i \cdot m_j}{\|\vec{p}_j - \vec{p}_i\|_2^3} \cdot (\vec{p}_j - \vec{p}_i)$$

$$\vec{a}_i = \vec{F}_i / m_i = \sum_{j=1, j \neq i}^n G \cdot \frac{m_j}{\|\vec{p}_j - \vec{p}_i\|_2^3} \cdot (\vec{p}_j - \vec{p}_i)$$

folgende Differentialgleichungen:

$$\boxed{\frac{d}{dt} \vec{p}_i = \vec{v}_i} \quad \boxed{\frac{d}{dt} \vec{v}_i = \sum_{j=1, j \neq i}^n G \cdot \frac{m_j}{\|\vec{p}_j - \vec{p}_i\|_2^3} \cdot (\vec{p}_j - \vec{p}_i)}$$

Obwohl wir die komplette Dynamik des Systems mit den beiden obigen Differentialgleichungen eingefangen haben, haben wir nun dennoch ein Problem. Wenn wir uns die obigen Differentialgleichungen ansehen, dann stellen wir fest, dass unser Modell eine Reihe von freien Parametern besitzt, nämlich gerade die Massen m_1, \dots, m_n . Sie beschreiben gerade, wie stark die Positionen \mathbf{p}_i der Punkte gegenseitig aufeinander Einfluss nehmen. Kennen wir diese Massen nicht, so ist unser Modell leider absolut wertlos, da wir keine korrekten Vorhersagen darüber machen können, wie genau sich die Position der Körper unter Einfluss der Gravitation mit der Zeit verändert – und das, obwohl wir das System korrekt modelliert haben!

Wir benötigen also ein Verfahren, mit dem wir diese Parameter bestimmen können, um ein Modell zu erhalten, das auch wirklich der Realität entspricht.

Eine gute Vorgehensweise bei der Modellierung eines Systems könnte beispielsweise aus den folgenden Punkten bestehen:

1. Aufstellen eines Modells für das System
2. Messen realer Daten des Systems
3. Anpassen der Modellparameter für das System anhand der gemessenen Daten

Punkt 1 haben wir bereits in der Einleitung in Form von Differentialgleichungen erledigt, die das Verhalten des Systems beschreiben.

Punkt 2 ermöglicht es einerseits, das Modell mit der Realität zu vergleichen (und damit zu verifizieren) und andererseits eventuelle Modellparameter anzupassen. Dies sollte man stets tun, da ein Modell, das nicht der Realität entspricht, in der Regel nicht sonderlich sinnvoll ist. Die notwendigen Daten haben wir einmal in Form von Messungen der Positionen von 7 einzelnen Körpern zu verschiedenen Zeitpunkten gesammelt. Die Messungen befinden sich in der Datei 7-body-system.dt und enthalten in der ersten Spalte den Zeitindex für jede Messung. In jeder weiteren Spalte befindet sich eine Abfolge der x-, y- und z-Koordinaten eines bestimmten Körpers (hintereinander angeordnet), wobei die i-te Spalte eine zeitliche Abfolge der Punktpositionen des i-ten Körpers darstellt. Das Format der Datei `7-body-system.dt` ist in der nachstehenden Tabelle noch einmal zur Verdeutlichung dargestellt:

Zeitindex 1	x-Koordinate von Punkt 1	x-Koordinate von Punkt 2	...
Zeitindex 1	y-Koordinate von Punkt 1	y-Koordinate von Punkt 2	...
Zeitindex 1	z-Koordinate von Punkt 1	z-Koordinate von Punkt 2	...
Zeitindex 2	x-Koordinate von Punkt 1	x-Koordinate von Punkt 2	...
Zeitindex 2	y-Koordinate von Punkt 1	y-Koordinate von Punkt 2	...
Zeitindex 2	z-Koordinate von Punkt 1	z-Koordinate von Punkt 2	...
...

Der letzte Punkt 3 hat den Sinn, das Modell gewissermaßen mit den gemessenen Daten zu „eichen“, um ein Modell zu erhalten, das auch wirklich der Realität entspricht.

Die Optimierung einer Zielfunktion mit Hilfe eines Gradientenabstiegs war bisher eine sehr nützliche Methode, mit der sich die unbekannt Parameter des black-box-Modells optimieren und der Realität anpassen lassen. Aus diesem Grund wollen wir dies auch in dieser Aufgabe anhand des Modells für das N -Körper-System ausprobieren.

Aufgabe 5.1 Die Zielfunktion (4 Punkte)

Als erstes benötigen wir eine geeignete Zielfunktion, deren Parameter wir mit Hilfe des Gradientenabstiegs optimieren wollen. Zeigen Sie, warum die Funktion

$$R(m_1, \dots, m_n) = \frac{1}{2} \cdot \frac{1}{T} \cdot \sum_{\tau=1}^T \frac{1}{n} \cdot \sum_{i=1}^n \left(\vec{\ell}_i(\tau) - \vec{y}_i(\tau) \right)^T \cdot \left(\vec{\ell}_i(\tau) - \vec{y}_i(\tau) \right)$$

mit

$$\vec{\ell}_i(\tau) = \left(\frac{\vec{p}_i(t_{\tau+2}) - \vec{p}_i(t_{\tau+1})}{t_{\tau+2} - t_{\tau+1}} - \frac{\vec{p}_i(t_{\tau+1}) - \vec{p}_i(t_{\tau})}{t_{\tau+1} - t_{\tau}} \right) / (t_{\tau+1} - t_{\tau})$$

$$\vec{y}_i(\tau) = \sum_{j=1, j \neq i}^n G \cdot m_j \cdot \frac{\vec{p}_j(t_{\tau}) - \vec{p}_i(t_{\tau})}{\|\vec{p}_j(t_{\tau}) - \vec{p}_i(t_{\tau})\|_2^3}$$

bei Kenntnis der Punktpositionen $\mathbf{p}_1(t), \dots, \mathbf{p}_n(t)$ zu T auf einander folgenden Zeitpunkten $t = t_1, \dots, t_T$ für kleine Zeitunterschiede $(t_{\tau+1} - t_{\tau})$, $\tau = 1, \dots, T-1$ ein guter Kandidat für eine geeignete Zielfunktion sein könnte.

Aufgabe 5.2 Bestimmung des Gradienten (8 BonusPunkte)

Um das Gradientenverfahren zur Optimierung der Modellparameter anzuwenden, benötigen wir noch die partiellen Ableitungen

$$\frac{\partial}{\partial m_k} R(m_1, \dots, m_n)$$

der Zielfunktion, da diese den Gradienten (Richtungsvektor des steilsten Anstiegs der Zielfunktion) bilden.

Zeigen Sie, dass diese partiellen Ableitungen durch

$$\frac{\partial}{\partial m_k} R(m_1, \dots, m_n) = -\frac{1}{T} \cdot \sum_{\tau=1}^T \frac{1}{n} \cdot \sum_{i=1, i \neq k}^n \left(\vec{\ell}_i(\tau) - \vec{y}_i(\tau) \right)^T \cdot G \cdot \frac{\vec{p}_k(t_\tau) - \vec{p}_i(t_\tau)}{\|\vec{p}_k(t_\tau) - \vec{p}_i(t_\tau)\|_2^3}$$

gegeben sind.

Diese Aufgabe ist evtl. etwas knifflig, da sie eine recht unschöne Rechnung besitzt. Es kann aber nicht schaden, die Ableitung einer Zielfunktion einmal wirklich selbst ausgerechnet zu haben. Aus diesem Grund werden die Punkte dieser Aufgabe als Bonuspunkte gutgeschrieben.

Aufgabe 5.3 Anpassung der Parameter (8 Punkte)

Verwenden Sie nun die oben angegebenen partiellen Ableitungen, um mit Hilfe eines Gradientenabstiegs die Massen der einzelnen Körper (und damit die Parameter unseres Modells) allein anhand der Positionen der Körper zu bestimmen, die in der Datei `7-body-system.dt` aufgezeichnet wurden.

Hinweise:

- Zur Erinnerung: Bei einem Gradientenabstieg werden die Parameter a_1, \dots, a_n einer Zielfunktion R für alle $i = 1, \dots, n$ gleichzeitig gemäß der Iteration

$$a_i \leftarrow a_i - \gamma \cdot \frac{\partial}{\partial a_i} R$$

angepasst, mit einer zu wählenden Lernrate (Schrittweite) $\gamma > 0$.

- Das Gradientenverfahren wird unter Umständen etwas schneller konvergieren, wenn Sie statt des eigentlichen Gradienten, den sogenannten stochastischen Gradienten verwenden, der auf die Mittelwertbildung über alle Trainingsbeispiele verzichtet und stattdessen die partiellen Ableitungen

$$\frac{\partial}{\partial m_k} R_\tau(m_1, \dots, m_n) = -\frac{1}{n} \cdot \sum_{i=1, i \neq k}^n \left(\vec{\ell}_i(\tau) - \vec{y}_i(\tau) \right)^T \cdot G \cdot \frac{\vec{p}_k(t_\tau) - \vec{p}_i(t_\tau)}{\|\vec{p}_k(t_\tau) - \vec{p}_i(t_\tau)\|_2^3}$$

der Funktion

$$R_\tau(m_1, \dots, m_n) = \frac{1}{2} \cdot \frac{1}{n} \cdot \sum_{i=1}^n \left(\vec{\ell}_i(\tau) - \vec{y}_i(\tau) \right)^T \cdot \left(\vec{\ell}_i(\tau) - \vec{y}_i(\tau) \right)$$

zum Anpassen der Parameter verwendet. Die Anpassung der Parameter erfolgt dann nacheinander für jedes $\tau = 1, \dots, T$.

Aufgabe 5.4 Simulation des Systems (9 Punkte)

Zur Simulation eines Dynamischen Modells möchten wir die *Euler-Cauchy-Integration* kennen lernen. Damit lassen sich die (evtl. vektorwertigen) Funktionswerte $\mathbf{z}(t)$ einer Differentialgleichung der Form $d/dt \mathbf{z}(t) = \mathbf{f}(\mathbf{z}(t))$, ausgehend von einem Startwert $\mathbf{z}(0)$, zu verschiedenen Zeitpunkten t approximieren. Sie basiert auf der Taylorreihenentwicklung von $\mathbf{z}(t)$ zum Entwicklungspunkt t'

$$\vec{z}(t) = \vec{z}(t') + \frac{d}{dt} \vec{z}(t') \cdot (t - t') + \frac{1}{2!} \cdot \frac{d^2}{dt^2} \vec{z}(t') \cdot (t - t')^2 + \dots$$

Einsetzen von $t = t' + \Delta t$ und Abbrechen nach dem ersten Glied liefert

$$\vec{z}(t' + \Delta t) \approx \vec{z}(t') + \frac{d}{dt}z(t') \cdot \Delta t$$

Umbenennen von t' in t und Einsetzen von $d/dt z(t)$ liefert dann die Formel für die **Euler-Cauchy-Integration**

$$\vec{z}(t + \Delta t) \approx \vec{z}(t) + f(\vec{z}(t)) \cdot \Delta t$$

Ein Problem mit dieser Formel besteht jedoch darin, dass sie alle Terme höherer Ordnung in der Taylorreihenentwicklung vernachlässigt, was zwangsläufig große Fehler zur Folge hat. Die Approximation von \mathbf{z} , verglichen mit anderen Methoden, die auch die höheren Terme beachten, ist daher nach Ausführen sehr vieler Zeitschritte Δt nicht sonderlich gut.

Ein besseres Verfahren ist zum Beispiel die sogenannte **Runge-Kutta-Methode**, die auch den Fehler der nächsthöheren Terme in der Taylorreihenentwicklung zu minimieren versucht. Sie wird beschrieben durch die Gleichungen:

$$\begin{aligned} \vec{k}_1 &= \Delta t \cdot f(\vec{z}(t)) \\ \vec{k}_2 &= \Delta t \cdot f(\vec{z}(t) + \frac{1}{2} \cdot \vec{k}_1) \\ \vec{k}_3 &= \Delta t \cdot f(\vec{z}(t) + \frac{1}{2} \cdot \vec{k}_2) \\ \vec{k}_4 &= \Delta t \cdot f(\vec{z}(t) + \vec{k}_3) \\ \vec{z}(t + \Delta t) &\approx \vec{z}(t) + \frac{1}{6} \cdot (\vec{k}_1 + 2\vec{k}_2 + 2\vec{k}_3 + \vec{k}_4) \end{aligned}$$

Im Folgenden wollen wir uns nun einmal das Fehlerverhalten der beiden Methoden genauer ansehen, um ein Gefühl von der Qualität der beiden Verfahren zu bekommen.

- Simulieren Sie das Dynamische Modell aus Aufgabe 5.3 einmal mit Hilfe der Euler-Cauchy-Integration und einmal mit Hilfe der Runge-Kutta-Methode. Als Startwerte für die Geschwindigkeiten und Positionen der Punkte können Sie die Werte aus den Dateien `v0.dt` und `p0.dt` verwenden. Die Datei `v0.dt` enthält in der i -ten Spalte den Geschwindigkeitsvektor und die Datei `p0.dt` die Koordinaten des i -ten Körpers zum Zeitpunkt $t = 1$. Erstellen Sie dann einen Plot, in dem der durchschnittliche Fehlerbetrag der berechneten Positionen zu den tatsächlichen Positionen der Körper für beide Methoden zu erkennen ist. (8 Pkte)
- Was stellen Sie fest, wenn Sie den Fehler der Euler-Cauchy-Integration mit dem Fehler der Runge-Kutta-Methode vergleichen? (1 Pkt)

Hinweise:

- Zur Berechnung des Fehlers können Sie die realen Punktpositionen aus der Datei `7-body-system.dt` verwenden. Beachten Sie dabei auch den Zeitindex in dieser Datei.
- Für die Massen m_1, \dots, m_n können Sie diejenigen Werte verwenden, die Sie in Aufgabe 5.3 berechnet haben. Zum Vergleich können Sie diese aber auch aus der Datei `m0.dt` auslesen, die in der i -ten Spalte den Wert der Masse m_i enthält.
- Als Schrittweite beider Simulationen empfiehlt sich $\Delta t = 0,01$.