Dokumentation PHIT Projekt

Simulation eines Zweikörperproblems

Die Überlegungen, Methoden und Arbeitsschritte zur Simulation eines Zweikörpersystems bis hin zur Entwicklung eines Simulationstools in der Programmiersprache Java.

- Studiengang: Informatik
- Modul: Physik für Informatik (PHIT)
- Semester: HS 2019 (3. Semester)
- Abgabetermin: Freitag, 15. November 2019

Autoren

Benjamin Bertalan   Fabian Flütsch   Etienne Gubler

Aus der Klasse IT18a_WIN der Zürcher Hochschule für angewandte Wissenschaften.

- Dozent: Rudolf Füchslin

Inhaltsverzeichnis

1. Einleitung

Im vorliegenden Dokument wird das von uns gewählte Vorgehen zur Lösung des Zweikörperproblems beschrieben. Dies umfasst sämtliche Überlegungen und Arbeitsschritte, die zur Erarbeitung der Simulation notwendig waren. Ausserdem werden die erreichten Resultate und insbesondere das Verhalten der Simulation erklärt.

Die Resultate der Simulation wurden graphisch ansprechend gestaltet und sind ebenfalls Teil dieser Dokumentation.

Dieser Bericht ist gemäss den Punkten A – D in der Aufgabenstellung [1] gegliedert.

2. Ausgangslage

Gegeben seien die zwei folgenden Körper:

Körper 1

Als erster Körper wird die Erde gewählt. Die Erde hat eine Masse $\mathbf{m_1}$ von $\mathbf{5.972 * 10^{24}\, \mathit{kg}}$. [2]

Körper 2

Als zweiter Körper wird der Mond gewählt. Der Mond hat eine Masse $\mathbf{m_2}$ von $\mathbf{7.349 * 10^{22}\, \mathit{kg}}$. [3]

Der Mond rotiert in einer elliptischen Bahn um die Erde (wenn die Erde als Ursprung des Koordinatensystems gewählt wird). Die folgende Abbildung veranschaulicht diesen Sachverhalt.

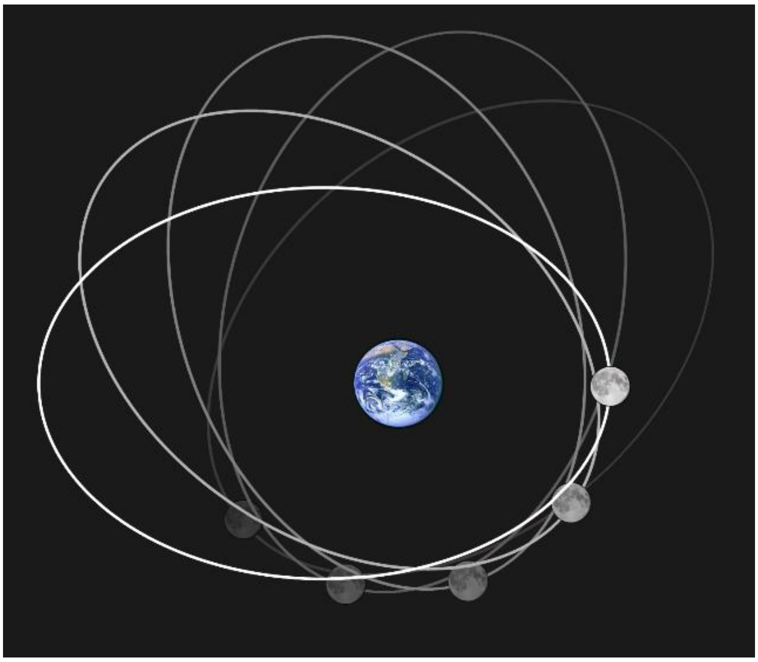

Abbildung 1: Umlaufbahn des Mondes um die Erde (Quelle: Wikipedia)

Gesucht sind die Bahnen, auf welchen sich die zwei Körper (Erde und Mond) bewegen. Also die Positionen der Körper in Abhängigkeit der Zeit.

3. Modellierung (Punkt A)

Dieses Kapitel beschreibt den Prozess zur Erarbeitung der Modellgleichungen. Die Modellgleichungen müssen das Bewegungsverhalten der zwei Körper (Erde und Mond) beschreiben.

3.1. Vorüberlegungen

Um ein Modell für die Bewegung zu erhalten, müssen wir als erstes feststellen, welche Grössen die Bewegung der betrachteten Körper beeinflussen. In den folgenden Kapiteln werden diese Grössen und deren Einfluss auf die Bewegung der Körper genauer betrachtet.

3.1.1. Kraft

Aufgrund der Massen $\mathbf{m_1}$ und $\mathbf{m_2}$ der beiden Körper, wirkt zwischen den Körpern eine Kraft. Diese Kraft verläuft entlang der Verbindungslinie zwischen den beiden Körpern. Die Kraft auf $\mathbf{m_1}$ und auf $\mathbf{m_2}$ sind betragsmässig identisch und verlaufen in entgegengesetzte Richtungen.

Die wirkenden Kräfte sind abhängig von der Distanz zwischen den Körpern (und von deren Massen). Das folgende Wechselwirkungsgesetz beschreibt die Kraft, welche auf die beiden Körper wirkt:

$$\vec{F}_{12} = \gamma \frac{m_1 * m_2}{|\vec{r}_{12}|^3} \vec{r}_{12}$$

Hierbei bezeichnet $\boldsymbol{\gamma}$ die Gravitationskonstante: $\boldsymbol{\gamma} \approx \mathbf{6.6743 * 10^{-11}} \, \frac{\boldsymbol{m^3}}{\boldsymbol{kg*s^2}}$ [4]

Ausserdem bezeichnet $\vec{\boldsymbol{r}}_{\mathbf{12}}$ den Vektor von $\mathbf{m_1}$ zu $\mathbf{m_2}$ (also von der Erde zum Mond): $\vec{\boldsymbol{r}}_{\mathbf{12}} = \boldsymbol{r_2} - \boldsymbol{r_1}$

In der folgenden Abbildung sind die auf $\mathbf{m_1}$ und $\mathbf{m_2}$ wirkenden Kräfte dargestellt.

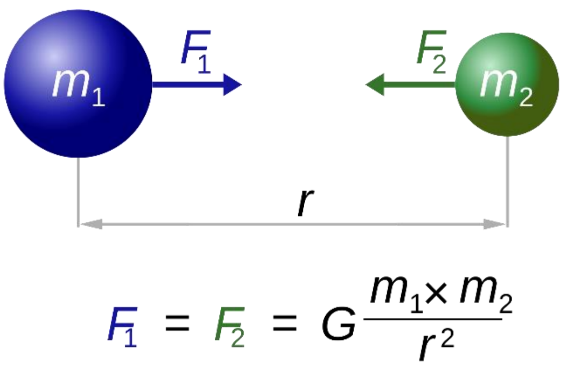

Abbildung 2: Anziehungskräfte zwischen zwei Massen (Quelle: Wikipedia)

Der Betrag und die Richtung von $\vec{F}_{12}$ (resp. $\vec{F}_{21} = -\vec{F}_{12}$) verändern sich, wenn sich die Position der Körper verändert. Dies führt dazu, dass sich beide Körper auf einer elliptischen Bahn bewegen.

Abgesehen von der Kraft $\vec{F}_{12}$ (resp. $\vec{F}_{21}$) wirken keine weiteren Kräfte auf die betrachteten Körper. Dies ist eine Vereinfachung, welche es uns erlaubt, die Bewegungen der Erde und des Mondes in einem zweidimensionalen System (in der Ebene) zu beschreiben.

Wie oben beschrieben, wirken die zwei betrachteten Kräfte entlang der Verbindungslinie zwischen den zwei Körpern. Diese Linie liegt zu jedem Zeitpunkt in derselben Ebene. Dies ist dadurch gegeben, dass die aus der Kraft resultierende Bewegung (resp. deren Veränderung durch die wirkende Beschleunigung) in Richtung der Kraft verläuft.

Obige Annahme gilt nur dann, wenn auch die Anfangsgeschwindigkeiten in der betrachteten Ebene liegen. Die Anfangsgeschwindigkeiten sind im folgenden Abschnitt genauer beschrieben.

3.1.2. Anfangsgeschwindigkeiten

Die oben beschriebenen Kräfte wirken auf der Verbindungslinie zwischen den zwei Körpern. Wenn also die Anfangsgeschwindigkeiten beider Körper null wären, so würde die Gravitationskraft dazu führen, dass sich die Körper annähern und schliesslich kollidieren.

Dies entspricht nicht den Erwartungen an das Bewegungsverhalten von Erde und Mond. Die Anfangsgeschwindigkeiten können somit nicht null sein.

Der Betrag der Anfangsgeschwindigkeit kann anhand der folgenden Überlegungen ermittelt werden:

Wir wissen, dass sich der Mond in einer nahezu kreisförmigen Umlaufbahn um die Erde bewegt. Daher ist der Abstand zwischen den Körpern Erde und Mond auch zu jedem Zeitpunkt annähernd gleich gross. Er beträgt etwa 384'400 km [3]. Anhand dieses Wertes können wir die Länge der Mond Umlaufbahn berechnen.

$$l_{moon\ orbit} = 2 * |\vec{r}_{12}| * \pi \approx 2.415 * 10^9\ m$$

Wir wissen ausserdem, dass der Mond ungefähr einen Monat benötigt, um die Erde einmal zu umkreisen. Ein Monat entspricht einer Zeit von ungefähr $2.628 * 10^6\ s$. Es lässt sich nun die Durchschnittsgeschwindigkeit des Mondes wie folgt berechnen:

$$v_{moon_{avg}} = \frac{l_{moon\ orbit}}{t_{moon\ revolution}} \approx 9.189 * 10^2\ m/s$$

Da der Mond stets ungefähr gleich weit von der Erde entfernt ist, können wir davon ausgehen, dass die Geschwindigkeit des Mondes zu jedem Zeitpunkt annähernd der oben berechneten Durchschnittsgeschwindigkeit entspricht.

Wenn die Geschwindigkeit des Mondes zu jedem Zeitpunkt annähernd gleich gross (also konstant) ist, dann entspricht auch die Anfangsgeschwindigkeit diesem konstanten Wert. Es gilt also:

$$|\vec{v}_{moon}(0)| = 9.189 * 10^2\ m/s$$

Es gilt nun noch die Richtung dieser Anfangsgeschwindigkeit zu bestimmen. Bei kreisförmigen Bewegungen verläuft die Geschwindigkeit stets tangential zur Umlaufbahn. Wir bestimmen nun als nächstes die Anfangspositionen beider Körper und legen dann die Richtungen deren Anfangsgeschwindigkeiten entsprechend fest.

Auch die Erde hat eine Anfangsgeschwindigkeit. Deren Berechnung wird im Kapitel «Modellgleichungen» erläutert. Wir stellen aber bereits jetzt folgendes über die Anfangsgeschwindigkeit der Erde fest:

Die Masse der Erde ist wesentlich grösser als jene des Mondes. Daher bewegt sich der Mond um die Erde. Die Erde bewegt sich dabei nur minimal und mit viel kleinerer Geschwindigkeit als der Mond. Die Anfangsgeschwindigkeit der Erde (sowie deren Geschwindigkeit zu jedem anderen Zeitpunkt) wird ausserdem immer in entgegengesetzter Richtung zur (Anfangs-) Geschwindigkeit des Mondes verlaufen. Den Grund dafür werden wir ebenfalls im Kapitel «Modellgleichungen» untersuchen.

3.1.3. Anfangspositionen

Um den Verlauf der Position der Erde und des Mondes (deren Trajektorien) zu berechnen, müssen wir für beide Körper die Anfangspositionen festlegen. Die einzige Bedingung an die Anfangspositionen ist, dass deren Abstand der Distanz zwischen der Erde und dem Mond entsprechen muss.

Die Position dieser zwei Anfangspunkte in der Ebene ist nicht weiter eingeschränkt. Wir wählen daher für die Erde die folgende Anfangsposition:

$$\vec{r}_{earth}(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

Ausgehend davon müssen wir für die Anfangsposition des Mondes gemäss der oben beschriebenen Bedingung einen Punkt wählen, der genau $|\vec{r}_{12}|$ vom Anfangspunkt der Erde entfernt ist. Wir wählen den folgenden Punkt:

$$\vec{r}_{moon}(0) = \begin{pmatrix} |\vec{r}_{12}| \\ 0 \end{pmatrix}$$

Auf der folgenden Abbildung sind die gewählten Anfangspositionen und das zugrundeliegende Koordinatensystem dargestellt:

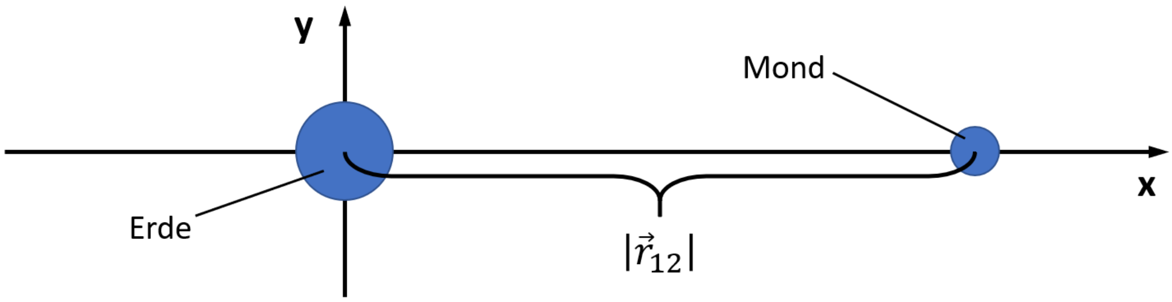

Abbildung 3: Koordinatensystem mit Anfangspositionen der beiden Körper

Wenn wir nun die Umlaufbahn des Mondes als kreisförmige Bewegung um das Zentrum der Erde einzeichnen, können wir die Richtungen der Anfangsgeschwindigkeit des Mondes bestimmen.

Die Geschwindigkeit muss bei einer kreisförmigen Bewegung, wie bereits oben festgehalten, stets tangential zur Umlaufbahn verlaufen. Es stehen uns somit nur zwei Möglichkeiten für die Wahl der Richtung zur Verfügung. Entweder zeigt die Anfangsgeschwindigkeit des Mondes in Richtung der positiven oder in Richtung der negativen Y-Achse. Wir wählen die Anfangsgeschwindigkeit wie folgt:

$$\vec{v}_{moon}(0) = \begin{pmatrix} 0\,\frac{m}{s} \\ -9.189 * 10^2\,\frac{m}{s} \end{pmatrix}$$

Die oben getroffene Annahme, dass die Umlaufbahn des Mondes ein exakter Kreis um das Zentrum der Erde ist, stellt eine starke Vereinfachung dar. Diese Vereinfachung haben wir nur benötigt, um die Richtung der Anfangsgeschwindigkeit des Mondes zu bestimmen. Im Weiteren gehen wir wieder davon aus, dass die Umlaufbahn des Mondes eine beliebige (nahezu kreisförmige) Ellipse sein wird.

Wir haben insbesondere auch die Bewegung der Erde vernachlässigt. Diese werden wir aber später berechnen.

Nachdem wir die Richtung der Anfangsgeschwindigkeit des Mondes kennen, können wir nun auch die Richtung der Anfangsgeschwindigkeit der Erde bestimmen. Wir haben oben festgehalten, dass die Geschwindigkeiten von Mond und Erde zu jedem Zeitpunkt in entgegengesetzte Richtungen zeigen müssen. Folglich verläuft die Anfangsgeschwindigkeit der Erde in Richtung der positiven Y-Achse. Wir werden feststellen, dass dies tatsächlich stimmt, wenn wir im Kapitel «Modellgleichungen» den exakten Wert von $\vec{v}_{earth}(0)$ berechnen.

In der untenstehenden Abbildung ist die vollständige Anfangssituation mit dem gewählten Koordinatensystem dargestellt.

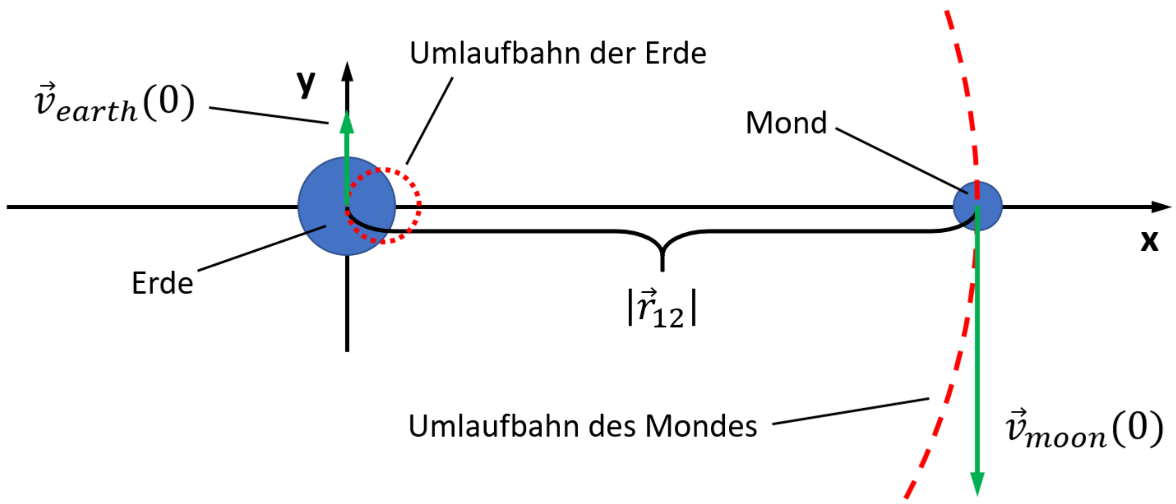

Abbildung 4: Koordinatensystem mit Anfangsgeschwindigkeiten und Umlaufbahnen

Aus obiger Abbildung geht hervor, dass wir für den Betrag der Anfangsgeschwindigkeit der Erde einen Wert erwarten, welcher (wesentlich) kleiner ist als die bereits berechnete Anfangsgeschwindigkeit des Mondes. Die dargestellten Umlaufbahnen sind Vermutungen. Wir erwarten insbesondere, dass die Umlaufbahn der Erde sehr klein sein wird. Es wird also hauptsächlich der Mond um die Erde kreisen.

3.2. Umlaufbahnen

Zwei Körper können sich auf mehrere Arten gegenseitig Umkreisen. Im Folgenden sind die verschiedenen Fälle beschrieben. Anschliessend wird die spezifische Situation für die Körper Mond und Erde genauer betrachtet.

3.2.1. Allgemeine Betrachtung

Die erste Möglichkeit ist, dass die Umlaufbahnen der Körper ineinander liegen. Somit wird also der innere Körper vom äusseren Körper umkreist. Der äussere Körper hat eine grössere Umlaufbahn und bewegt sich schneller als der innere. Dies ist dann der Fall, wenn die zwei Massen (stark) verschieden sind. Der innere Körper ist jener mit der grösseren Masse, er bewegt sich weniger schnell.

Die zweite Möglichkeit ist, dass sich die Umlaufbahnen der zwei betrachteten Körper schneiden. Dies ist dann der Fall, wenn die Massen beider Körper ähnlich gross sind. Die Grössen der sich schneidenden Umlaufbahnen können in diesem Fall trotzdem unterschiedlich sein. Gleich grosse Umlaufbahnen entstehen nur dann, wenn die zwei Massen identisch sind.

Die folgenden Bilder veranschaulichen die zwei oben beschriebenen Fälle.

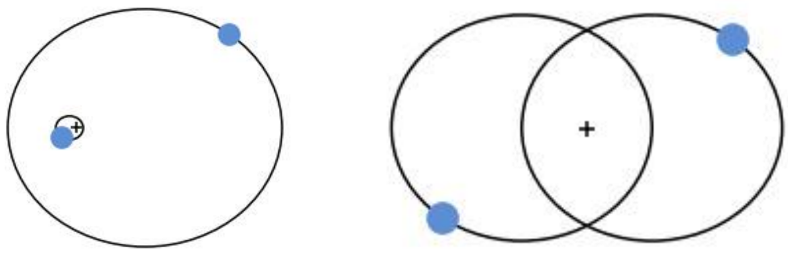

Abbildung 5: Umlaufbahnen bei unterschiedlich grossen Massen (links) resp. ähnlich grossen Massen (rechts) (Quelle: Wikipedia)

Für zwei Körper mit Massen $\mathbf{m_1}$ und $\mathbf{m_2}$ kann der Massenschwerpunkt bestimmt werden. Dieser ist für beide Situationen in der obenstehenden Abbildung mit einem **+** Symbol gekennzeichnet. Der Massenschwerpunkt liegt immer im inneren von beiden Umlaufbahnen. Hierbei muss noch folgender Spezialfall betrachtet werden:

Wenn man davon ausgeht, dass die zwei Massepunkte $\mathbf{m_1}$ und $\mathbf{m_2}$ eine gewisse Ausdehnung haben, so kann sich der Massenschwerpunkt im inneren des grösseren (resp. schwereren) Körper befinden.

3.2.2. Situation für die Körper Erde und Mond

Die Massen der zwei Körper Erde und Mond unterscheiden sich um zwei Grössenordnungen. Dies führt dazu, dass deren Umlaufbahnen ineinander liegen. Die Situation entspricht genau dem oben geschilderten Fall für zwei Körper mit unterschiedlich grossen Massen. Ausserdem trifft für die Körper Erde und Mond der beschriebene Spezialfall zu. Der Massenschwerpunkt befindet sich also im Inneren der Erde.

Der Massenschwerpunkt (insbesondere dessen Bewegung) ist für die im folgenden Kapitel beschriebenen Modellgleichungen wichtig. Die Tatsache, dass der Massenschwerpunkt im Inneren der Erde liegt ist im Weiteren jedoch nicht von Bedeutung. Wir betrachten die Erde und den Mond als Punktmassen $\mathbf{m_1}$ und $\mathbf{m_2}$, welche keine Ausdehnung haben.

3.3. Modellgleichungen

Im folgenden Kapitel werden wir die Gleichungen erarbeiten, die das Bewegungsverhalten der Erde und des Mondes beschreiben. Dabei werden wir auf Anfangswertprobleme treffen. Diese können wir dann mithilfe der oben berechneten Anfangsbedingungen lösen. Es ist also von grosser Bedeutung, dass wir die Anfangsbedingungen richtig bestimmt haben.

3.3.1. Schwerpunktsgeschwindigkeit

Bevor wir die Modellgleichungen für das Zweikörperproblem aufstellen, müssen wir noch die bereits bestimmten Anfangsbedingungen vervollständigen. Es fehlt uns noch die genaue Anfangsgeschwindigkeit der Erde. Für deren Richtung haben wir bereits eine Vermutung getroffen. Ausserdem gingen wir bis anhin davon aus, dass die (Anfangs-) Geschwindigkeit der Erde wesentlich kleiner sein wird als jene des Mondes. Beide Vermutungen werden wir nun anhand der folgenden Überlegungen und mithilfe der anschliessenden exakten Berechnung bestätigen.

Die Erde und der Mond sollen beide (auf unterschiedlichen elliptischen Bahnen) um einen festen Punkt kreisen. Dieser feste Punkt entspricht genau dem bereits oben erwähnten Massenschwerpunkt zwischen der Erde und dem Mond. Wir müssen also die Anfangsgeschwindigkeit der Erde so wählen, dass für die Geschwindigkeit des Massenschwerpunktes $v_{barycenter} = 0$ gilt.

Diese Bedingung stellt sicher, dass die zwei zu simulierenden Körper nicht in eine bestimmte Richtung abdriften, sondern stets im gleichen Bereich des Koordinatensystems bleiben. Dort kreisen sie dann auf elliptischen Bahnen um ihren gemeinsamen Massenschwerpunkt.

Wir berechnen nun die Anfangsgeschwindigkeit der Erde $\vec{v}_{earth}(0)$ anhand der zwei Massen **m_1** und **m_2** sowie mithilfe der folgenden bereits bestimmten Werte:

$$\vec{v}_{moon}(0) = \begin{pmatrix} 0\,\frac{m}{s} \\ -9.189 * 10^2\,\frac{m}{s} \end{pmatrix}, \qquad v_{barycenter} = 0$$

Für die Geschwindigkeit des Massenschwerpunktes gilt:

$$\vec{v}_{barycenter} = (m_1 * \vec{v}_1 + m_2 * \vec{v}_2) * \frac{1}{m_1 + m_2}$$

$$\Leftrightarrow$$

$$\vec{v}_1 = \frac{\vec{v}_{barycenter} * (m_1 + m_2) - m_2 * \vec{v}_2}{m_1}$$

$$\vec{v}_{barycenter} = 0$$

$$\Leftrightarrow$$

$$\vec{v}_1 = \frac{0 * (m_1 + m_2) - m_2 * \vec{v}_2}{m_1}$$

$$\Leftrightarrow$$

$$\vec{v}_1 = \frac{-m_2 * \vec{v}_2}{m_1}$$

Wir wählen also für die Anfangsgeschwindigkeit der Erde den folgenden Wert:

$$\vec{v}_{earth}(0) = \frac{-7.349 * 10^{22}\,kg}{5.972 * 10^{24}\,kg} * \begin{pmatrix} 0\,\frac{m}{s} \\ -9.189 * 10^2\,\frac{m}{s} \end{pmatrix} = \begin{pmatrix} 0\,\frac{m}{s} \\ 1.131 * 10^1\,\frac{m}{s} \end{pmatrix}$$

Dieser Wert stimmt mit unseren zuvor aufgestellten Vermutungen überein. Insbesondere stellen wir fest, dass die zwei Anfangsgeschwindigkeiten der Erde und des Mondes tatsächlich in entgegengesetzte Richtungen zeigen. Die Anfangsgeschwindigkeit der Erde ist auch wie vermutet wesentlich kleiner als jene des Mondes.

Wenn wir für die Anfangsgeschwindigkeit der Erde einen anderen Wert wählen würden, dann ergäbe sich daraus eine konstante Geschwindigkeit für den Massenpunkt (also $v_{barycenter} = const \neq 0$).

Das würde bedeuten, dass die Erde und der Mond mit der Zeit in eine feste Richtung abdriften. Der Massenschwerpunkt würde sich dann entlang einer geraden Bahn bewegen.

Da wir aber nur an den Umlaufbahnen der Erde und des Mondes interessiert sind, ist es sinnvoll, für den Massenschwerpunkt eine Geschwindigkeit von $v_{barycenter} = 0$ festzulegen. Wenn wir auch externe Einflüsse (zum Beispiel die Gravitationskraft der Sonne) betrachten würden, so würde sich der gemeinsame Massenschwerpunkt von Erde und Mond natürlich bewegen. Diese Bewegung des Massenschwerpunktes wäre dann allerdings nicht mehr geradlinig.

Die folgende Abbildung veranschaulicht den oben beschriebenen Sachverhalt:

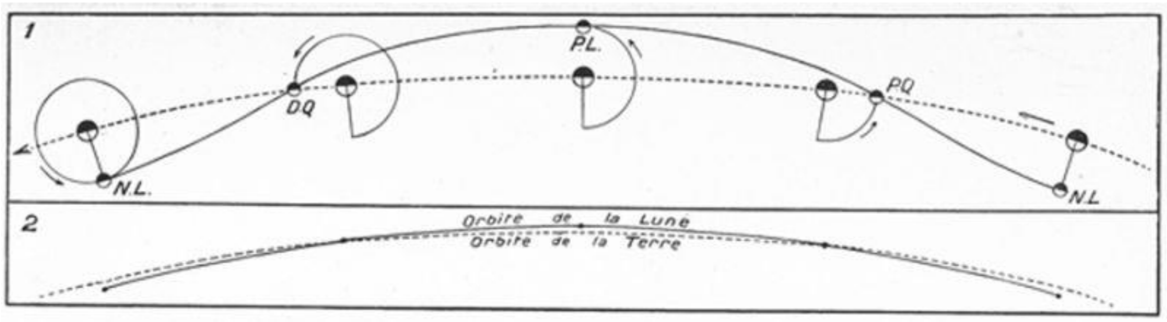

Abbildung 6: Umlaufbahn des Mondes um die Erde unter dem Einfluss der Sonne (Quelle: ilovetheuniverse.com)

3.3.2. Umlaufbahnen und Positionsgleichungen

Unser Ziel besteht darin, die Umlaufbahnen der Erde und des Mondes zu finden. Wir betrachten dabei die Umlaufbahnen, welche durch die gegenseitig wirkenden Gravitationskräfte dieser beiden Körper entstehen. Um das Bewegungsverhalten der Erde und des Mondes (also deren Umlaufbahnen) zu beschreiben, suchen wir nach einer Funktion für die Positionen dieser beiden Körper in Abhängigkeit der Zeit.

Wir haben im Kapitel «Vorüberlegungen» festgestellt, dass die Positionen der beiden Körper durch die auf sie wirkenden Kräfte beeinflusst werden. Wir beginnen unsere Suche nach den Bewegungsgleichungen also bei der wirkenden Kraft. Im Folgenden betrachten wir nur einen einzelnen Körper. Die Überlegungen zur Bewegungsgleichung gelten aber in gleichem Masse für beide Körper.

Um von der Kraft auf die Bewegung eines Körpers schliessen zu können, muss in einem ersten Schritt die wirkende Beschleunigung berechnet werden. Diese resultiert direkt aus der Kraft gemäss folgender Formel:

$$\vec{F} = m * \vec{a}$$

$$\Leftrightarrow$$

$$\vec{a} = \frac{\vec{F}}{m}$$

Die Richtung der Beschleunigung verläuft stets parallel zur Richtung der wirkenden Kraft.

Wenn die Anfangsgeschwindigkeit eines Körpers (zum Zeitpunkt t_0) und die zu jedem Zeitpunkt t auf ihn wirkende Beschleunigung bekannt sind, so lässt sich für jeden Zeitpunkt t die Geschwindigkeit des Körpers berechnen.

Dies ist ein Anfangswertproblem, zu dessen Lösung eine Differentialgleichung 1. Ordnung gelöst werden muss. Die folgende Gleichung beschreibt dieses Anfangswertproblem. Daraus geht hervor, dass die Ableitung der Geschwindigkeit eines Körpers gerade dessen Beschleunigung entspricht.

$$\dot{\vec{v}} = \vec{a}$$

Analog zu den eben beschriebenen Schritten kann nun auch von der Geschwindigkeit eines Körpers auf dessen Position geschlossen werden. Die folgende Gleichung zeigt, dass die Ableitung der Position eines Körpers gerade dessen Geschwindigkeit entspricht.

$$\dot{\vec{s}} = \vec{v}$$

Damit haben wir eine Beschreibung für die Position des Körpers gefunden. Die folgende Gleichung stellt die obigen Schritte zusammengefasst dar. Daraus geht direkt hervor, dass wir die Position des Körpers ausgehend von der auf ihn wirkenden Kraft beschreiben können.

$$\ddot{\vec{s}} = \frac{\vec{F}}{m}$$

Diese Gleichung gilt für beide Körper. Wir setzen nun noch die Formel für die Kraft aus dem Kapitel «Vorüberlegungen» ein und erhalten so je eine separate Beschreibung für die Position der Erde und für die Position des Mondes.

$$\ddot{\vec{s}}_{earth} = \frac{\gamma \frac{m_1 * m_2}{|\vec{r}_{12}|^3} \vec{r}_{12}}{m_1}, \qquad \ddot{\vec{s}}_{moon} = \frac{-\gamma \frac{m_1 * m_2}{|\vec{r}_{12}|^3} \vec{r}_{12}}{m_2}$$

Man beachte, dass sich die wirkende Kraft in den beiden obenstehenden Gleichungen lediglich im Vorzeichen unterscheidet.

Mit diesen Gleichungen haben wir eine Beschreibung für die Position beider Körper in Abhängigkeit der auf sie wirkenden Kräfte gefunden. Damit ist unser Modell vollständig und wir können die gesuchte Umlaufbahn sowohl für die Erde als auch für den Mond bestimmen.

Die gefundenen Gleichungen sind Differentialgleichungen 2. Ordnung. Sie bilden zusammen mit den zuvor bestimmten Anfangsbedingungen je ein Anfangswertproblem für die Position der Erde und für die Position des Mondes.

Um die Umlaufbahnen der beiden Körper darzustellen benötigen wir deren Position zu jedem Zeitpunkt t. Wir könnten also versuchen für deren Positionen explizite Funktionen von der folgenden Form zu finden:

$$\vec{s}(t) = ?$$

Eine solche analytische Lösung wäre aber nicht zielführend und sehr schwierig zu finden. Zur Simulation des Zweikörperproblems ist eine nummerische Lösung der Positionsgleichungen wesentlich besser geeignet. Dabei stehen uns verschiedene nummerische Verfahren zur Auswahl. Den durch die nummerischen Verfahren entstehenden Einfluss auf die resultierenden Umlaufbahnen werden wir im Kapitel «Einfluss von nummerischen Verfahren» betrachten.

Im nächsten Kapitel werden wir nun die Simulation des Zweikörperproblems und die daraus entstandenen Resultate untersuchen.

4. Simulation (Punkt B)

Wir möchten die oben erarbeiteten Modellgleichungen nun verwenden, um eine grafische Darstellung der Umlaufbahnen von Erde und Mond zu erhalten. Dazu könnten wir einen grafischen Modelleditor, wie zum Beispiel Berkeley-Madonna, verwenden. Diesen Modelleditor müssten wir dann so konfigurieren, dass er in der Lage ist unser Zweikörperproblem zu simulieren.

Wir haben uns jedoch dafür entschieden, ein eigenes Simulationstool in der Programmiersprache Java zu entwickeln. Die Eigenschaften dieses Programms werden im folgenden Kapitel erläutert.

4.1. Simulationstool

Das von uns entwickelte Programm ist in der Lage, das Zweikörperproblem mit den Körpern Erde und Mond zu simulieren. Gegenüber Berkeley-Madonna bietet unser Programm den Vorteil, dass die Ausgabe der Resultate spezifisch an unser Zweikörperproblem angepasst ist.

4.1.1. Grafische Darstellung

In unserem Simulationsprogramm wird die Erde **rot** und der Mond **blau** dargestellt. Das Programm zeigt jeweils die aktuelle Position dieser beiden Körper, sowie den Verlauf der bislang zurückgelegten Strecke auf deren Umlaufbahn. Die folgende Abbildung zeigt einen Screenshot des Programms, nachdem es gestartet wurde.

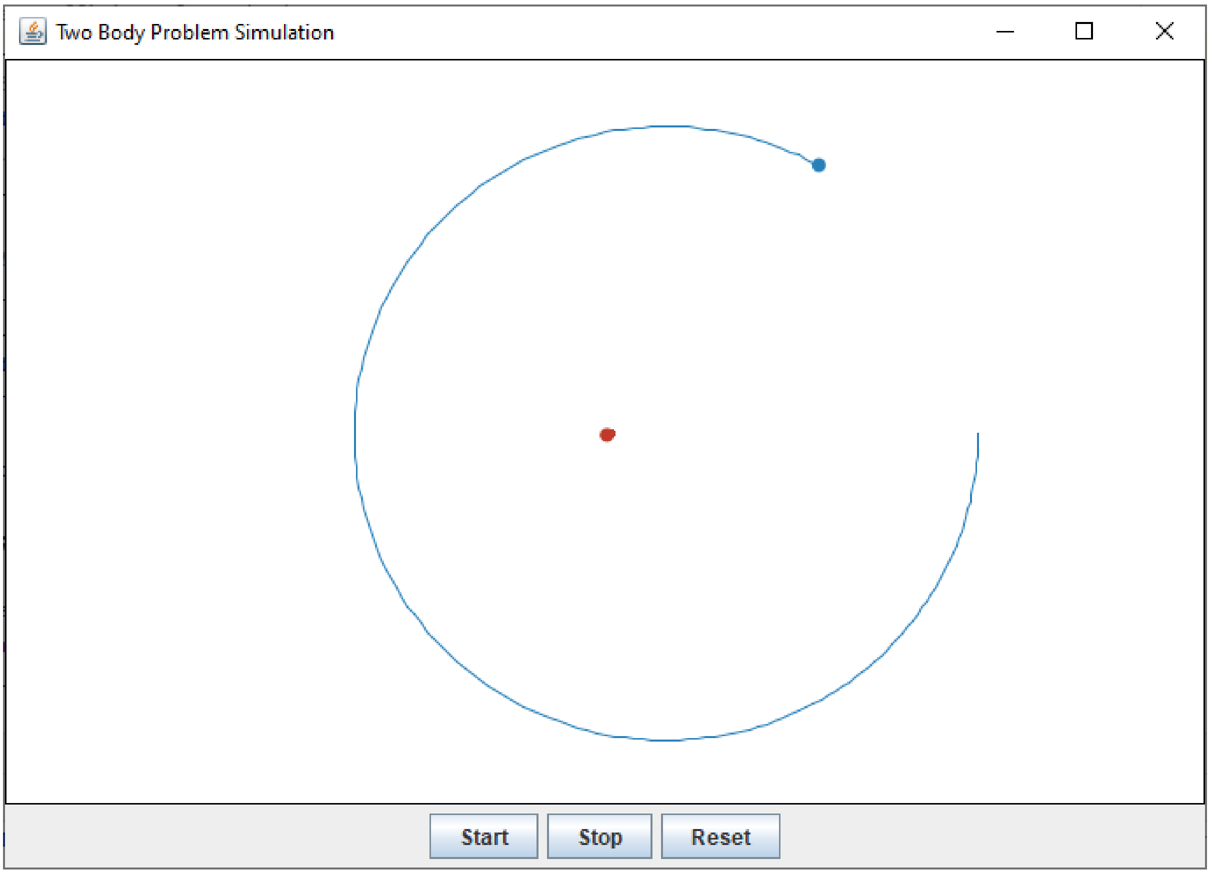

Abbildung 7: Screenshot des zur Simulation entwickelten Java Programms

4.1.2. Unterstützte nummerische Verfahren

Um die Umlaufbahnen der beiden Körper darstellen zu können, müssen die oben erarbeiteten Anfangswertprobleme gelöst werden. Unsere Anwendung macht dies mit einem nummerischen Verfahren. Es existieren verschiedene numerische Verfahren zur Lösung von Anfangswertproblemen. Diese unterscheiden sich bezüglich Genauigkeit und Rechenaufwand.

Unsere Anwendung unterstützt zwei solche Verfahren zur Lösung der Anfangswertprobleme. Wir haben die folgenden Algorithmen implementiert:

- Runge Kutta 4. Ordnung
- Euler Verfahren

Der Einfluss dieser verschiedenen Algorithmen auf die berechneten Umlaufbahnen wird im Kapitel «Einfluss von nummerischen Verfahren» betrachtet.

4.1.3. Konfigurierbare Parameter

Das Simulationsprogramm bietet ausserdem verschiedene Parameter, um das Verhalten der Simulation zu konfigurieren. Es stehen folgende Parameter zur Auswahl:

- Zeitschrittgrösse
- Potenz des Wechselwirkungsgesetzes $1 / r^{\alpha}$

Auch der Einfluss dieser Parameter wird in den zwei folgenden Kapiteln untersucht.

4.2. Simulation des Zweikörperproblems Erde und Mond

In diesem Kapitel wird untersucht, wie sich die Umlaufbahnen der Erde und des Mondes verhalten, wenn die oben erarbeiteten Anfangsbedingungen und Modellgleichungen für die Simulation verwendet werden.

Das Ziel ist es, das tatsächliche Verhalten der Erde und des Mondes möglichst genau abzubilden. Dazu wird das Runge Kutta Verfahren der 4. Ordnung angewendet, um die Anfangswertprobleme zu lösen. Dieses Verfahren ist im Allgemeinen wesentlich genauer als das einfachere Euler Verfahren.

Um die Genauigkeit weiter zu erhöhen wird eine kleine Zeitschrittgrösse verwendet. Ausserdem wenden wir für diese Simulation das normale Wechselwirkungsgesetz an.

Die für die Simulation verwendete Konfiguration ist also wie folgt:

Algorithmus	Runge Kutta 4. Ordnung
Zeitschrittgrösse	$8.64 * 10^3\ s\ \hat{=}\ 0.1\ Tage$
Wechselwirkungsgesetz	$1 / r^{\alpha}$ mit $\alpha = 2.0$

Zugehörige Abbildung siehe nächste Seite.

Der simulierte Zeitraum entspricht ungefähr 2 Jahren. Dies ergibt sich daraus, dass wir den Mond ca. 24 mal um die Erde rotieren liessen, bevor wir die Simulation beendet haben.

Die folgende Abbildung zeigt die Ausgabe des Simulationstools.

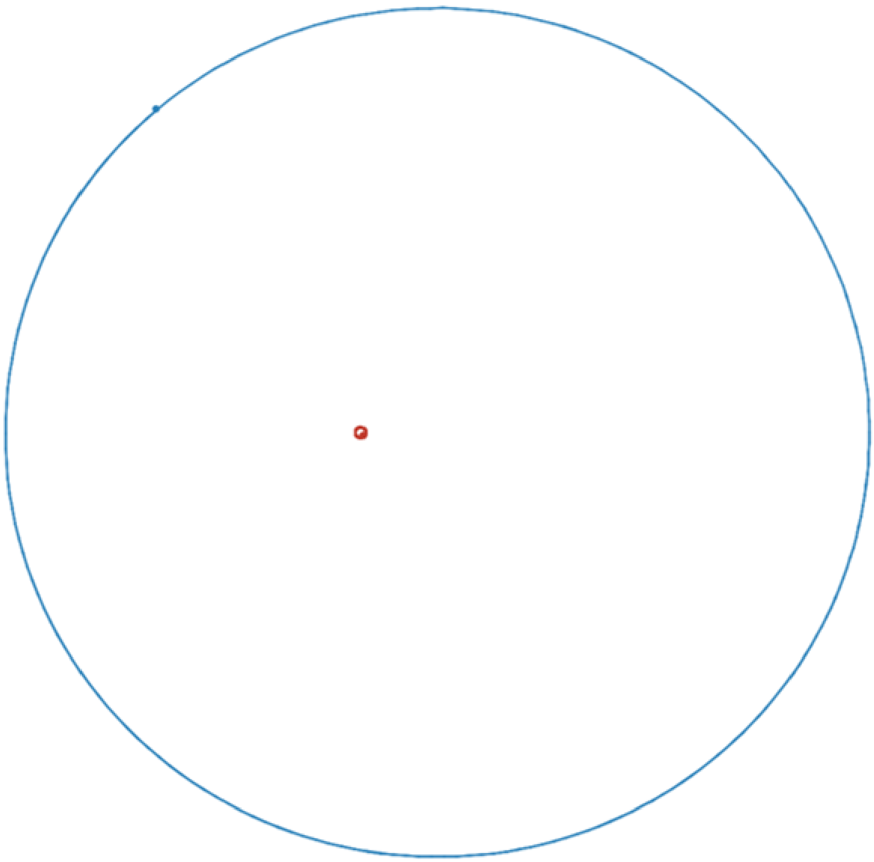

Abbildung 8: Simulation unter Standardbedingungen

Als erstes stellen wir fest, dass sich der Mond tatsächlich auf einer nahezu kreisförmigen Umlaufbahn um die Erde bewegt. Auch die Erde bewegt sich dabei auf einer kreisförmigen Umlaufbahn, diese ist jedoch wesentlich kleiner als jene des Mondes.

Es bestätigen sich also die im Kapitel «Modellierung» aufgestellten Vermutungen.

Wir machen ausserdem die folgenden Feststellungen:

- Der Mond (und die Erde) bewegen sich mit nahezu konstanter Geschwindigkeit
- Der Mond bewegt sich bei jeder Umrundung der Erde entlang derselben Bahn

Wir schliessen daraus, dass sich das Zweikörpersystem Erde und Mond in einem sehr stabilen Zustand befindet. Der Mond wird also noch für eine lange Zeit auf seiner heutigen Umlaufbahn um die Erde kreisen.

5. Einfluss von nummerischen Verfahren (Punkt C)

In diesem Kapitel untersuchen wir, wie die Wahl des nummerischen Verfahrens das Resultat der Simulation beeinflusst. Wir werden also das Runge Kutta Verfahren der 4. Ordnung mit dem Euler Verfahren vergleichen.

Allgemein gilt, dass das Runge Kutta Verfahren wesentlich genauer ist. Dies erlaubt es, einen signifikant grösseren Wert für den Zeitschritt zu wählen, ohne dass sich die Qualität der Simulation verschlechtert

5.1. Runge Kutta Verfahren 4. Ordnung

Wir haben bereits gesehen, dass unser Simulationstool mit dem Runge Kutta Verfahren unter Verwendung von kleinen Zeitschritten (0.1 Tage) sehr gute Resultate liefert. Nun führen wir erneut eine Simulation mit dem Runge Kutta Verfahren durch, jedoch mit stark vergrösserter Zeitschrittgrösse. Die Parameter sind wie folgt eingestellt:

Algorithmus	Runge Kutta 4. Ordnung
Zeitschrittgrösse	$\mathbf{1.728 * 10^5\ s \ \widehat{=}\ 2\ Tage}$
Wechselwirkungsgesetz	$1 / r^{\alpha}$ mit α = 2.0

Zugehörige Abbildung siehe unten.

Dies macht die Berechnung schneller, da weniger Punkte der Umlaufbahnen berechnet werden müssen. Es reduziert jedoch auch die Genauigkeit, da die Ungenauigkeiten des nummerischen Verfahrens dann stärker zu tragen kommen.

Die folgende Abbildung zeigt das Resultat der Simulation.

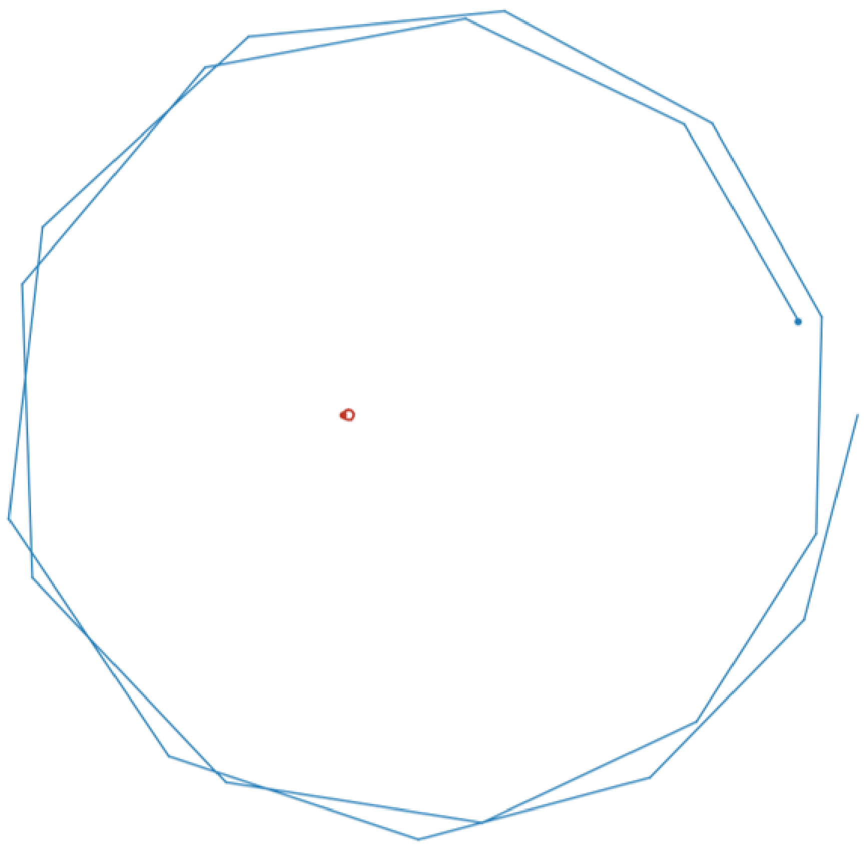

Abbildung 9: Simulation mit dem Runge Kutta Verfahren mit grossen Zeitschritten

Dieser Versuch zeigt, dass das Runge Kutta Verfahren der 4. Ordnung wirklich äusserts gute Näherungslösungen für die Anfangswertprobleme liefert. Die Umlaufbahn des Mondes (und der Erde) sind noch eindeutig zu erkennen, obwohl wir gerade mal 12 Punkte pro Erdumrundung berechnet haben. Wir haben die Position des Mondes also nur ca. alle zwei Tage berechnet und erhalten trotzdem noch ein brauchbares Resultat.

Wir haben diesen Versuch noch ein drittes Mal durchgeführt, und dabei eine Zeitschrittgrösse von 1 Tag gewählt. Dann ist die Ausgabe des Simulationstools bereits so exakt, dass man das Bild nicht mehr von Abbildung 8 (mit einer Zeitschrittgrösse von 0.1 Tagen) unterscheiden kann. Dies zeigt, dass unter Verwendung des Runge Kutta Verfahrens eine Zeitschrittgrösse von 1 Tag absolut ausreichend ist.

5.2. Euler Verfahren

Wir vergleichen die obigen Versuche nun mit dem Euler Verfahren. Dieses Verfahren wird wesentlich ungenauere Resultate liefern.

In einem ersten Versuch wählen wir als Zeitschrittgrösse den Wert von 0.1 Tagen. Dies hat mit dem Runge Kutta verfahren zu hervorragenden Resultaten geführt. Die Parameter sind wie folgt:

Algorithmus	**Euler Verfahren**
Zeitschrittgrösse	$\mathbf{8.64 * 10^3\ s\ \widehat{=}\ 0.1\ Tage}$
Wechselwirkungsgesetz	$1\ /\ r^{\alpha}$ mit α = 2.0

Zugehörige Abbildung siehe unten.

Die folgende Abbildung zeigt das Resultat dieses Versuchs.

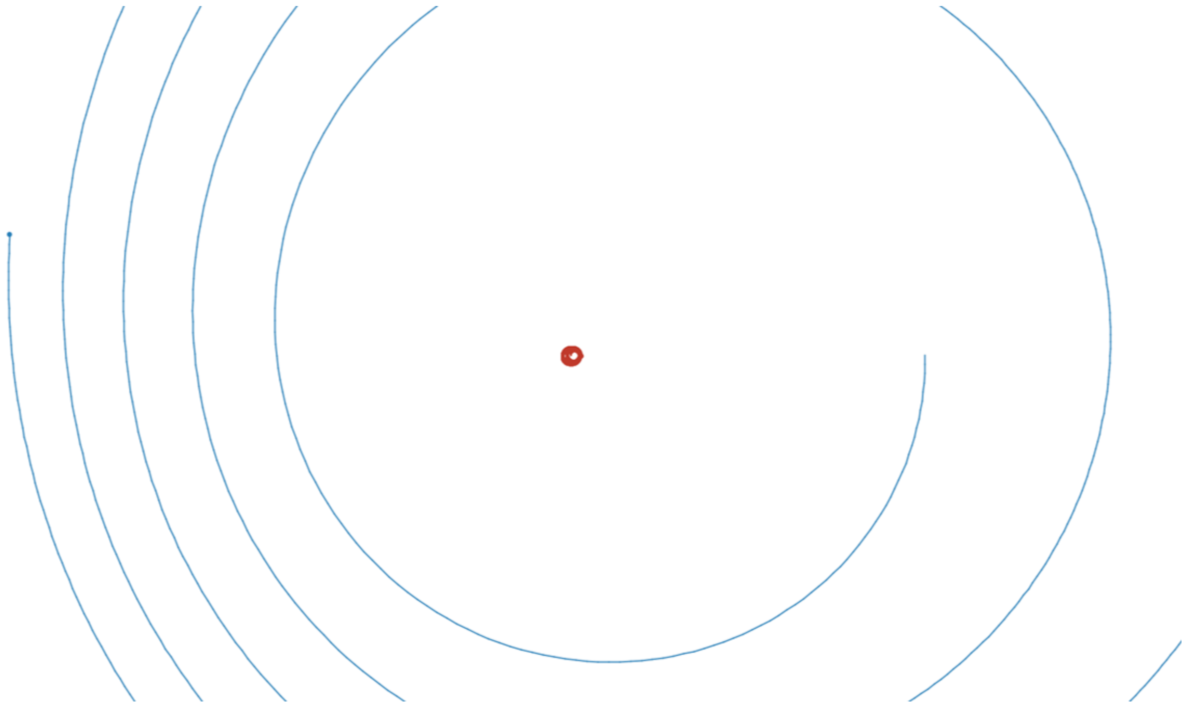

Abbildung 10: Simulation mit dem Euler Verfahren mit kleinen Zeitschritten

Die Abbildung zeigt, dass der Versuch mit den gewählten Einstellungen keine brauchbaren Resultate liefert. Wenn das Euler Verfahren verwendet wird, müsste eine noch viel kleinere Zeitschrittgrösse verwendet werden, um das Verhalten der Erde und des Mondes ähnlich genau wie mit dem Runge Kutta Verfahren zu simulieren.

Weitere Versuche haben gezeigt, dass für die Zeitschrittgrösse ein Wert von 0.0001 Tagen gewählt werden muss, um ähnliche Simulationsresultate wie unter Anwendung des Runge Kutta Verfahrens (bei einer Zeitschrittgrösse von 1 Tag) zu erhalten.

Der Wert von 0.0001 Tagen entspricht ca. 9 Sekunden. Beim Euler Verfahren müssen also äusserst viele Punkte der Umlaufbahnen berechnet werden (die Positionen beider Körper alle 9 Sekunden) um eine korrekte Simulation zu erhalten.

	Gute Simulationsresultate ab einer Zeitschrittgrösse von:
Runge Kutta Verfahren 4. Ordnung	1 Tag
Euler Verfahren	0.0001 Tage

Es lässt sich also sagen, dass das Runge Kutta Verfahren der 4. Ordnung um den Faktor 10^4 genauer ist als das Euler Verfahren. Man beachte jedoch, dass diese spezifischen Werte nur für das von uns modellierte Zweikörperproblem gelten.

6. Einfluss des Wechselwirkungsgesetzes (Punkt D)

In diesem Kapitel möchten wir noch untersuchen, wie verschiedene Wechselwirkungsgesetze das Verhalten der Simulation beeinflussen. Allgemein gilt folgendes Wechselwirkungsgesetz:

$$F = \gamma \frac{m_1 * m_2}{r^\alpha} \quad mit\ \alpha = 2.0$$

Der Parameter $\alpha = 2.0$ kann jedoch verändert werden, um ein anderes Verhalten der gegenseitig wirkenden Gravitationskräfte zu erzielen.

In einem ersten Versuch verwenden wir die folgenden Parameter:

Algorithmus	Runge Kutta 4. Ordnung
Zeitschrittgrösse	$8.64 * 10^3\ s\ \widehat{=}\ 0.1\ Tage$
Wechselwirkungsgesetz	**$1 / r^\alpha$ mit $\alpha = 2.03$**

Zugehörige Abbildung siehe nächste Seite.

Wir arbeiten von nun an nur noch mit dem Runge Kutta Verfahren, da dieses wesentlich genauer ist als das Euler Verfahren. Die Zeitschrittgrösse erhöhen wir nach Bedarf, falls die Simulationsresultate zu wenig exakt sind.

Wir haben in diesem Versuch einen Wert für α gewählt, welcher leicht grösser ist als 2.0. Wir erwarten, dass sich der Mond nun weiter von der Erde entfernen kann. Dies schliessen wir daraus, dass die

Potenz von r (also von der Distanz zwischen den zwei Körpern) erhöht wurde. Die dämpfende Wirkung einer grossen Distanz sollte also stärker ins Gewicht fallen und somit die Anziehungskraft zwischen der Erde und dem Mond reduzieren.

Die folgende Abbildung zeigt das Resultat des Versuchs.

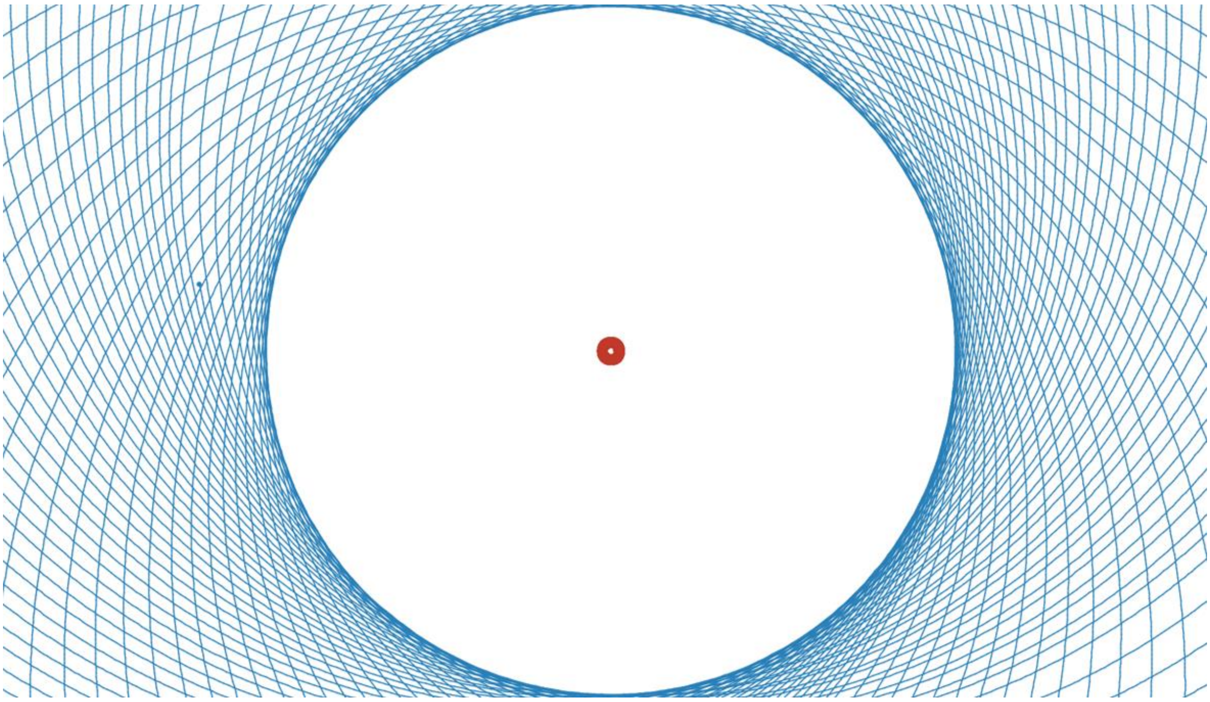

Abbildung 11: Simulation mit einem α Wert von grösser als 2.0

Wir stellen fest, dass sich der Mond tatsächlich weiter von der Erde entfernen kann. Das System scheint allerdings immer noch einen stabilen Zustand zu haben. Der Mond entfernt sich nach jeder vollständigen Umrundung wieder gleich weit von der Erde (und nicht weiter als zuvor). Es ist also offensichtlich nicht der Fall, dass der Mond zu einem gewissen Zeitpunkt seine Umlaufbahn um die Erde verlassen kann.

Wir machen einen zweiten Versuch mit einem noch etwas grösseren Wert für α:

Algorithmus	Runge Kutta 4. Ordnung
Zeitschrittgrösse	$8.64 * 10^3\ s\ \hat{=}\ 0.1\ Tage$
Wechselwirkungsgesetz	**$1\ /\ r^{\alpha}$ mit α = 2.042**

Zugehörige Abbildung siehe nächste Seite.

Auch hier zeigt sich wieder ein ähnliches Bild wie zuvor. Wir haben auch für diesen Versuch zusätzlich verifiziert, dass sich der Mond nach jeder Umrundung nicht weiter von der Erde entfernt als in der vorhergehenden Umrundung.

Es fällt bei diesem Versuch auf, dass sich die Erde auch bewegt. Durch das veränderte Wechselwirkungsgesetz wurde also nicht nur die Bewegung des Mondes, sondern auch jene der Erde verstärkt.

Das folgende Bild zeigt die Simulierten Umlaufbahnen.

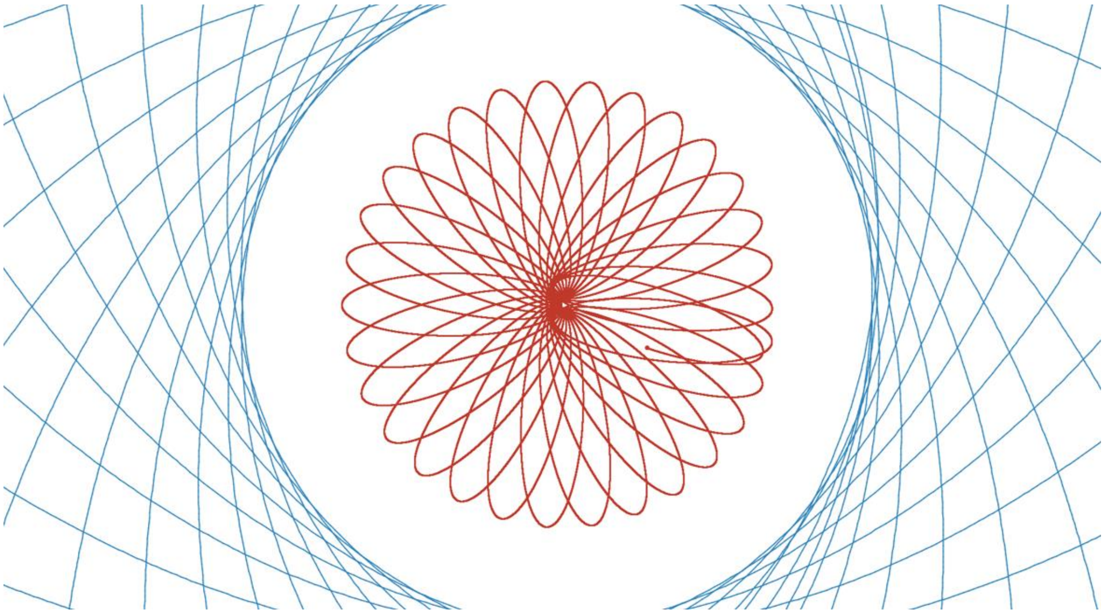

Abbildung 12: Simulation mit einem noch grösseren α Wert (α = 2.042)

Nun möchten wir noch herausfinden, was geschieht, wenn wir den Wert von α kleiner als 2.0 wählen. Wir führen eine Simulation mit den folgenden Parametern durch:

Algorithmus	Runge Kutta 4. Ordnung
Zeitschrittgrösse	$8.64 * 10^3\ s\ \widehat{=}\ 0.1\ Tage$
Wechselwirkungsgesetz	**$1\ /\ r^{\alpha}$ mit α = 1.99**

Zugehörige Abbildung siehe unten.

Wir erwarten analog zu den zwei vorherigen Versuchen, dass sich der Mond nun stärker an die Erde annähert. Die folgende Ausgabe des Simulationstools bestätigt diese Vermutung.

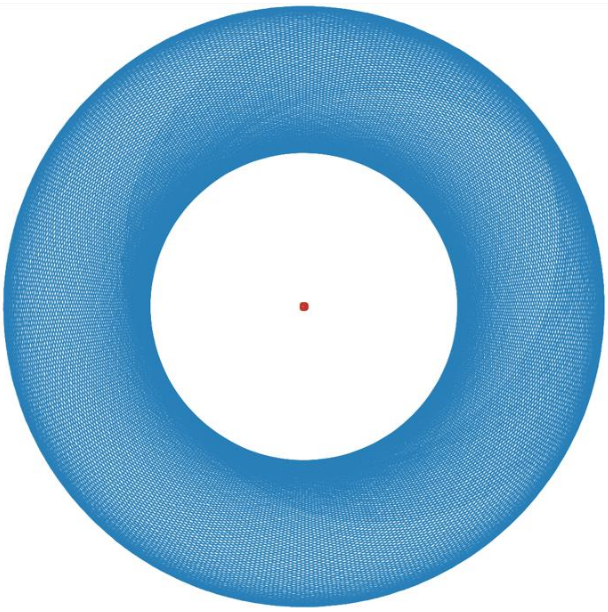

Abbildung 13: Simulation mit einem α Wert von kleiner als 2.0

Auch hier stellen wir fest, dass das Zweikörpersystem mit dem gewählten α Wert stabil zu sein scheint. Der Mond nähert sich zwar stärker an die Erde an als gewöhnlich, er entfernt sich dann aber auch wieder in gleichem Masse.

Wir machen ein weiteres Experiment mit noch kleinerem α Wert:

Algorithmus	Runge Kutta 4. Ordnung
Zeitschrittgrösse	$8.64 * 10^3\ s\ \widehat{=}\ 0.1\ Tage$
Wechselwirkungsgesetz	**$1\ /\ r^{\alpha}$ mit α = 1.945**

Zugehörige Abbildung siehe unten.

Die folgende Abbildung stellt die Resultate der Simulation dar.

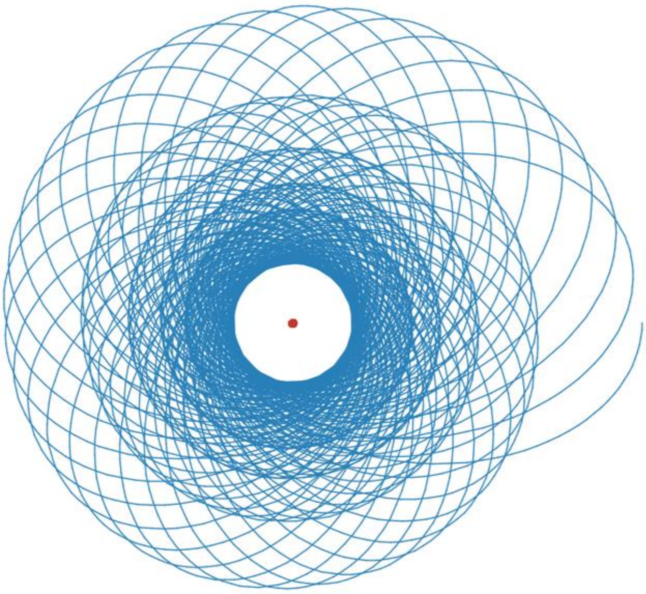

Abbildung 14: Simulation mit einem noch kleineren α Wert (α = 1.945)

In diesem Versuch fällt sofort auf, dass sich der Mond nun kontinuierlich der Erde annähert. Dies ist ein neues Verhalten, welches wir bis anhin noch in keinem Versuch beobachten konnten. Dieses System ist also nicht stabil. Wenn die Simulation lange genug läuft kollidiert der Mond mit der Erde.

Nach der Betrachtung aller Versuche zum Einfluss des Wechselwirkungsgesetzes können wir folgende Vermutung aufstellen:

- Wir gehen davon aus, dass ein $\alpha > 2.0$ die Anziehungskräfte zwischen den Körpern reduziert. Der Mond kann sich somit weiter von der Erde entfernen. Er kreist jedoch immer um die Erde und kann diese Umlaufbahn nicht verlassen.

- Ausserdem vermuten wir, dass ein $\alpha < 2.0$ die Anziehungskräfte zwischen den Körpern verstärkt. Der Mond bewegt sich in diesem Fall näher an die Erde heran. Dieses System scheint nicht immer stabil zu sein. Wenn ein genügend kleiner α Wert gewählt wird, so kollidiert der Mond nach einer gewissen Zeit mit der Erde.

7. Schlussfolgerungen

Abschliessend werden in diesem Kapitel die wichtigsten Erkenntnisse aus dieser Untersuchung zusammengefasst.

7.1. Modellierung

Wir haben festgestellt, dass sich Zweikörperprobleme mit erstaunlich wenigen und einfachen Modellgleichungen beschreiben lassen.

Ausschlaggebend für ein korrektes Modell sind insbesondere die Anfangsbedingungen. Diese müssen sinnvoll gewählt werden, da sie das Verhalten des Systems massgeblich beeinflussen.

7.2. Simulation

Zweikörperprobleme lassen sich dank des einfachen Modells auch mit sehr wenig Aufwand simulieren. Bereits eine rudimentäre Implementation kann zu sehr spektakulären Resultaten führen. Für genauere und aufschlussreichere Versuche ist es dann entscheidend, die numerischen Verfahren für die Simulation korrekt zu konfigurieren.

Eine exakte Simulation ist nur dann möglich, wenn ein mehrstufiges Verfahren zur nummerischen Lösung der Anfangswertprobleme verwendet wird. Zum Beispiel das Runge Kutta Verfahren der 4. Ordnung.

Unsere Versuche haben die im Kapitel «Modellierung» aufgestellten Vermutungen bestätigt. Der Mond kreist also tatsächlich auf einer nahezu kreisförmigen Bahn um die Erde. Die Erde bewegt sich dabei auch, aufgrund ihrer grösseren Masse jedoch nur minimal.

Die Simulationen haben ausserdem gezeigt, dass das Wechselwirkungsgesetz einen grossen Einfluss auf die Umlaufbahn des Mondes hat. Nur unter Anwendung des $1 / r^2$ - Gesetzes bewegt sich der Mond auf seiner tatsächlichen Bahn um die Erde. Wird für α ein Wert ungleich 2.0 eingesetzt, so kann sich der Mond entweder näher an die Erde heran oder weiter von ihr wegbewegen. Wird ein gewisser α Wert unterschritten, so ist das System nicht mehr stabil und der Mond wird nach einer gewissen Zeit mit der Erde kollidieren.

7.3. Numerik

Wir haben für unser spezifisches Zweikörpersystem festgestellt, dass der Runge Kutta Algorithmus der 4. Ordnung wesentlich genauer ist als das Euler Verfahren. Diese Feststellung gilt auch im Allgemeinen.

Durch unsere Versuche haben wir ausserdem erkannt, dass die Ungenauigkeit des Euler Verfahrens teilweise durch eine (starke) Verkleinerung der Zeitschrittgrösse kompensiert werden kann. Dies ist aber ineffizient und geht zulasten der Performance.

Ein effizientes nummerisches Verfahren zur Lösung von Anfangswertproblemen, wie zum Beispiel das Runge Kutta Verfahren, ist für die computergestützte Simulation von grosser Bedeutung.

8. Quellenverzeichnis

[1] „Auftrag: Simulation eines Zweikörperproblems“.

[2] Wikipedia, „Erde“. URL: https://de.wikipedia.org/wiki/Erde

[3] Wikipedia, „Mond“. URL: https://de.wikipedia.org/wiki/Mond

[4] Wikipedia, „Gravitationskonstante“.
URL: https://de.wikipedia.org/wiki/Gravitationskonstante

9. Abbildungsverzeichnis

10. Anhang

10.1. Quellcode

Unser Simulationstool ist vollständig in Java programmiert und basiert auf keinerlei externem Code. Alle Funktionen (inklusive der nummerischen Verfahren zur Lösung der Anfangswertprobleme) sowie die grafische Benutzeroberfläche wurden von uns selbst entwickelt.

Auf den folgenden Seiten befindet sich der komplette Quellcode unserer Anwendung.

```java
package ch.gubleet.phit;

import javax.swing.*;
import java.awt.*;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Line2D;
import java.awt.geom.Point2D;
import java.awt.image.BufferedImage;
import java.util.LinkedList;
import java.util.List;
import java.util.concurrent.Executors;
import java.util.concurrent.ScheduledExecutorService;
import java.util.concurrent.ScheduledFuture;
import java.util.concurrent.TimeUnit;

public class TwoBodyProblem {

    // used for simulation
    private static final double GRAVITATIONAL_CONSTANT = 6.6743e-11;
    private static final double EARTH_WEIGHT = 5.972e24;
    private static final double MOON_WEIGHT = 7.349e22;
    private static final double G_M1 = GRAVITATIONAL_CONSTANT * EARTH_WEIGHT;
    private static final double G_M2 = GRAVITATIONAL_CONSTANT * MOON_WEIGHT;

    // algorithms
    private static final int RUNGE_KUTTA_4 = 0;
    private static final int EULER = 1;

    //initial values
    private static final double earth_x_init = 0;
    private static final double earth_y_init = 0;
    private static final double earth_vx_init = 0;
    private static final double earth_vy_init = 1.131e1;
    private static final double moon_x_init = 3.844e8;
    private static final double moon_y_init = 0;
    private static final double moon_vx_init = 0;
    private static final double moon_vy_init = -9.189e2;

    // ------------------------------------------------------------
    // CONFIGURE VALUES TO ADJUST SIMULATION BEHAVIOR
    // ------------------------------------------------------------
    // 8.64e4 = 1 Day (sufficient for runge kutta 4th order)
    // 3.6e3 = 1 Hour (almost sufficient for euler)
    private static final double TIME_STEP = 8.64e4;
    // RUNGE_KUTTA_4 or EULER
    private static final int ALGORITHM = RUNGE_KUTTA_4;
    // alpha for law of gravitation 1/r^alpha
    private static final double ALPHA = 2.0;

    // only used to calculate reasonable paint refresh rate
    private static final double AVG_MOON_EARTH_DISTANCE = 3.844e8;
    private static final double AVG_MOON_SPEED = 9.189e2;
    private static final int SECONDS_PER_REVOLUTION = 3;
    private static final long STEP_DELAY_MICROSECONDS = Math.round(
SECONDS_PER_REVOLUTION * 1e6 / (2.628e6 / TIME_STEP));
```

```java

    private final Object lock = new Object();
    private ScheduledFuture future;
    private boolean running;

    private List<Position> earthPath;
    private List<Position> moonPath;
    private BufferedImage buffer;

    // earth state
    private double earth_x;
    private double earth_y;
    private double earth_vx;
    private double earth_vy;

    private double earth_dx;
    private double earth_dy;
    private double earth_dvx;
    private double earth_dvy;

    // moon state
    private double moon_x;
    private double moon_y;
    private double moon_vx;
    private double moon_vy;

    private double moon_dx;
    private double moon_dy;
    private double moon_dvx;
    private double moon_dvy;

    private double scale;
    private double origin_x;
    private double origin_y;

    private int paint_d;
    private int unpainted;

    private JPanel cp;
    private SimulationPanel sim;
    private JPanel glassPanel;

    private Color c_earth = new Color(192, 57, 43);
    private Color c_moon = new Color(41, 128, 185);

    private TwoBodyProblem() {
        Dimension screen = Toolkit.getDefaultToolkit().getScreenSize();
        double scale_max = Math.max(screen.width, screen.height) / ( AVG_MOON_EARTH_DISTANCE * 2);
        paint_d = (int) Math.ceil(10 / (AVG_MOON_SPEED * TIME_STEP * scale_max ));
        initialize();
        JFrame frame = new JFrame();
        frame.setTitle("Two Body Problem Simulation");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

```
        frame.setResizable(true);
        frame.setMinimumSize(new Dimension(700, 500));
        frame.setExtendedState(JFrame.MAXIMIZED_BOTH);
        createUI();
        frame.setContentPane(cp);
        frame.pack();
        frame.setLocationRelativeTo(null);
        frame.setVisible(true);
    }

    public static void main(String[] args) {
        new TwoBodyProblem();
    }

    private void initialize() {
        earthPath = new LinkedList<>();
        moonPath = new LinkedList<>();
        // earth
        earth_x = earth_x_init;
        earth_y = earth_y_init;
        earth_vx = earth_vx_init;
        earth_vy = earth_vy_init;
        earthPath.add(new Position(earth_x, earth_y));
        // moon
        moon_x = moon_x_init;
        moon_y = moon_y_init;
        moon_vx = moon_vx_init;
        moon_vy = moon_vy_init;
        moonPath.add(new Position(moon_x, moon_y));
        // paint counter
        unpainted = 0;
    }

    @SuppressWarnings("ConstantConditions")
    private void step() {
        // method to solve the differential equations
        if (ALGORITHM == RUNGE_KUTTA_4) {
            rungeKutta4();
        }
        if (ALGORITHM == EULER) {
            euler();
        }
        unpainted++;
        if (unpainted == paint_d) {
            earthPath.add(new Position(earth_x, earth_y));
            moonPath.add(new Position(moon_x, moon_y));
            synchronized (lock) {
                if (buffer != null) {
                    drawLast();
                }
            }
            unpainted = 0;
        }
    }

```

```java
    private void rungeKutta4() {
        double t = TIME_STEP;
        double[] dt = new double[]{t / 2.0, t / 2.0, t, 0.0};
        double[] c = new double[]{t / 6.0, t / 3.0, t / 3.0, t / 6.0};
        double[] u0 = new double[]{earth_x, earth_y, earth_vx, earth_vy, moon_x, moon_y, moon_vx, moon_vy};
        double[] ut = new double[]{0, 0, 0, 0, 0, 0, 0, 0};
        double[] u = varToU();
        for (int j = 0; j < 4; j++) {
            uToVar(u);
            derivative();
            double[] du = new double[]{earth_dx, earth_dy, earth_dvx, earth_dvy, moon_dx, moon_dy, moon_dvx, moon_dvy};
            for (int i = 0; i < 4; i++) {
                u[i] = u0[i] + dt[j] * du[i];
                ut[i] = ut[i] + c[j] * du[i];
                u[i + 4] = u0[i + 4] + dt[j] * du[i + 4];
                ut[i + 4] = ut[i + 4] + c[j] * du[i + 4];
            }
        }
        for (int i = 0; i < 8; i++) {
            u[i] = u0[i] + ut[i];
        }
        uToVar(u);
    }

    private void euler() {
        double t = TIME_STEP;
        derivative();
        // earth
        earth_x += earth_dx * t;
        earth_y += earth_dy * t;
        earth_vx += earth_dvx * t;
        earth_vy += earth_dvy * t;
        // moon
        moon_x += moon_dx * t;
        moon_y += moon_dy * t;
        moon_vx += moon_dvx * t;
        moon_vy += moon_dvy * t;
    }

    private void derivative() {
        // vector from earth to moon
        double dx = moon_x - earth_x;
        double dy = moon_y - earth_y;
        double l = Math.sqrt(dx * dx + dy * dy);
        // earth
        earth_dx = earth_vx;
        earth_dy = earth_vy;
        double p = ALPHA + 1;
        earth_dvx = (G_M2 / Math.pow(l, p)) * dx;
        earth_dvy = (G_M2 / Math.pow(l, p)) * dy;
        // moon
        moon_dx = moon_vx;
        moon_dy = moon_vy;
```

```
        moon_dvx = (-G_M1 / Math.pow(l, p)) * dx;
        moon_dvy = (-G_M1 / Math.pow(l, p)) * dy;
    }

    private void uToVar(double[] u) {
        earth_x = u[0];
        earth_y = u[1];
        earth_vx = u[2];
        earth_vy = u[3];
        moon_x = u[4];
        moon_y = u[5];
        moon_vx = u[6];
        moon_vy = u[7];
    }

    private double[] varToU() {
        double[] u = new double[8];
        u[0] = earth_x;
        u[1] = earth_y;
        u[2] = earth_vx;
        u[3] = earth_vy;
        u[4] = moon_x;
        u[5] = moon_y;
        u[6] = moon_vx;
        u[7] = moon_vy;
        return u;
    }

    private Point2D posToPoint(Position position) {
        double x = origin_x + position.x * scale;
        double y = origin_y + position.y * scale * -1;
        return new Point2D.Double(x, y);
    }

    private void start() {
        if (!running) {
            ScheduledExecutorService scheduler = Executors.newSingleThreadScheduledExecutor();
            future = scheduler.scheduleAtFixedRate(this::step,
                    0, STEP_DELAY_MICROSECONDS, TimeUnit.MICROSECONDS);
            running = true;
        }
    }

    private void stop() {
        if (running) {
            future.cancel(false);
            running = false;
        }
    }

    private void reset() {
        if (running) {
            stop();
        }
```

```java
        initialize();
        synchronized (lock) {
            buffer = null;
        }
        glassPanel.repaint();
        sim.repaint();
    }

    private void drawLast() {
        drawLast(earthPath);
        drawLast(moonPath);
        glassPanel.repaint();
        sim.repaint();
    }

    private void drawLast(List<Position> path) {
        Graphics2D g = buffer.createGraphics();
        g.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
        if (path == earthPath) {
            g.setColor(c_earth);
        }
        if (path == moonPath) {
            g.setColor(c_moon);
        }
        Point2D from = posToPoint(path.get(path.size() - 1));
        Point2D to = posToPoint(path.get(path.size() - 2));
        g.draw(new Line2D.Double(from, to));
        g.dispose();
    }

    private void drawAll() {
        drawAll(earthPath);
        drawAll(moonPath);
        glassPanel.repaint();
        sim.repaint();
    }

    private void drawAll(List<Position> path) {
        Graphics2D g = buffer.createGraphics();
        g.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
        if (path == earthPath) {
            g.setColor(c_earth);
        }
        if (path == moonPath) {
            g.setColor(c_moon);
        }
        Point2D prev = null;
        for (int i = 0; i < path.size(); i++) {
            Position position = path.get(i);
            Point2D curr = posToPoint(position);
            if (prev == null) {
                prev = curr;
                continue;
```

```java
            }
            g.draw(new Line2D.Double(prev, curr));
            prev = curr;
        }
        g.dispose();
    }

    private void createUI() {
        cp = new JPanel();
        cp.setLayout(new BorderLayout());
        // simulation
        sim = new SimulationPanel();
        // glass panel
        glassPanel = new JPanel() {
            @Override
            protected void paintComponent(Graphics g) {
                super.paintComponent(g);
                Graphics2D g2d = (Graphics2D) g;
                g2d.setRenderingHint(RenderingHints.KEY_ANTIALIASING, RenderingHints.VALUE_ANTIALIAS_ON);
                // earth indicator
                Point2D p = posToPoint(earthPath.get(earthPath.size() - 1));
                Ellipse2D e = new Ellipse2D.Double(p.getX() - 4, p.getY() - 4, 8, 8);
                g.setColor(c_earth);
                g2d.fill(e);
                // moon indicator
                p = posToPoint(moonPath.get(moonPath.size() - 1));
                e = new Ellipse2D.Double(p.getX() - 4, p.getY() - 4, 8, 8);
                g.setColor(c_moon);
                g2d.fill(e);
            }
        };
        glassPanel.setOpaque(false);
        // buttons
        JPanel buttons = new JPanel();
        JButton start = new JButton("Start");
        start.addActionListener(e -> start());
        start.setFocusable(false);
        JButton stop = new JButton("Stop");
        stop.addActionListener(e -> stop());
        stop.setFocusable(false);
        JButton reset = new JButton("Reset");
        reset.addActionListener(e -> reset());
        reset.setFocusable(false);
        buttons.add(start);
        buttons.add(stop);
        buttons.add(reset);
        // content pane
        JLayeredPane layers = new Layers();
        layers.add(glassPanel, Integer.valueOf(1));
        layers.add(sim, Integer.valueOf(0));
        cp.add(layers, BorderLayout.CENTER);
        cp.add(buttons, BorderLayout.SOUTH);
    }
```

```java

    private static class Position {

        double x;
        double y;

        Position(double x, double y) {
            this.x = x;
            this.y = y;
        }
    }

    private static class Layers extends JLayeredPane {

        Layers() {
            setLayout(new LayoutManager() {
                @Override
                public void addLayoutComponent(String name, Component comp) {
                    // nothing
                }

                @Override
                public void removeLayoutComponent(Component comp) {
                    // nothing
                }

                @Override
                public Dimension preferredLayoutSize(Container parent) {
                    return new Dimension(Integer.MAX_VALUE, Integer.MAX_VALUE);
                }

                @Override
                public Dimension minimumLayoutSize(Container parent) {
                    return new Dimension(0, 0);
                }

                @Override
                public void layoutContainer(Container parent) {
                    for (int i = 0; i < parent.getComponentCount(); i++) {
                        parent.getComponent(i).setBounds(parent.getBounds());
                    }
                }
            });
        }
    }

    private class SimulationPanel extends JPanel {

        SimulationPanel() {
            setBorder(BorderFactory.createLineBorder(Color.BLACK));
            setBackground(Color.WHITE);
        }

        private BufferedImage getBuffer() {
```

```
            if (buffer == null) {
                Insets i = getInsets();
                int w = Math.max(1, getWidth() - i.left - i.right);
                int h = Math.max(1, getHeight() - i.top - i.bottom);
                buffer = new BufferedImage(w, h, BufferedImage.TYPE_INT_ARGB);
                origin_x = i.left + (w / 2.0);
                origin_y = i.top + (h / 2.0);
                double scale_w = w / (AVG_MOON_EARTH_DISTANCE * 2);
                double scale_h = h / (AVG_MOON_EARTH_DISTANCE * 2);
                scale = Math.min(scale_w, scale_h);
                drawAll();
            }
            return buffer;
        }

        @Override
        public void invalidate() {
            synchronized (lock) {
                buffer = null;
                super.invalidate();
            }
        }

        @Override
        protected void paintComponent(Graphics g) {
            synchronized (lock) {
                super.paintComponent(g);
                Insets i = getInsets();
                g.drawImage(getBuffer(), i.left, i.top, this);
            }
        }
    }
}
```