

Sonnensystem mit Bewegung der Sonne

Projekt F

Elisabeth Zeyen und Michael Nowak

Tutor: Laura Martin

13. Januar 2017

Inhaltsverzeichnis

1. Theorie	1
1.1. Sonnensystem	1
1.2. Newtonsches Gravitationsgesetz	1
1.3. Baryzentrum	2
1.4. Runge-Kutta-Methode	2
2. Simulation	3
2.1. Wahl der Einheiten	3
2.2. Näherungen	3
2.3. Code Snippets mit Kommentaren	4
2.3.1. class Planet	4
2.3.2. Funktionen	5
2.3.3. main-Funktion	6
3. Auswertung	8
3.1. Beeinflusst die Bewegung der Planeten die Position der Sonne?	8
3.2. Energieerhaltung	9
3.3. Stepsize	10
3.4. Statistische Auswertung der gewonnen Größen	10
3.5. Ausblick	11
A. Statische Auswertung	13

1. Theorie

1.1. Sonnensystem

Das Sonnensystem setzt sich zusammen aus der Sonne, die von acht Planeten umkreist wird, sowie weiteren kleineren Körpern, dazu zählen Zwergplaneten, Satelliten, Kometen und Asteroiden.

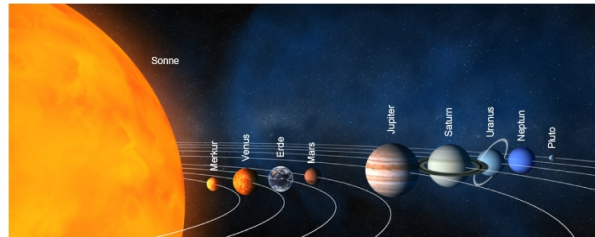


Abbildung 1: Schematische Abbildung des Sonnensystems. Abstände nicht maßstabgetreu.⁴

Die **Sonne** ist mit Abstand der massereichste Körper des Sonnensystems und umfasst mit ihrem Gewicht etwa 99,86 % der Gesamtmasse des Sonnensystems. In ihr verschmelzen unter hohem Druck und hohen Temperaturen hauptsächlich Wasserstoffkerne zu Helium und die dabei freiwerdende Energie wird abgestrahlt.

Der Planet mit der kleinsten Umlaufbahn ist Merkur, gefolgt von Venus, Erde und Mars. Diese Planeten zählen zum **inneren Sonnensystem**, sie bestehen im Wesentlichen aus Metallen und Gestein. Demnach besitzen sie eine relativ hohe Dichte zwischen $3,95 \text{ g/cm}^3$ bis $5,52 \text{ g/cm}^3$ und haben eine feste Oberfläche und einen kleineren Durchmesser als die äußeren Planeten (zwischen etwa 4800 km bis 12800 km).¹

Getrennt durch einen Asteroidengürtel folgen die **äußeren Gasplaneten** Jupiter, Saturn, Uranus und Neptun.

Früher wurde ebenfalls Pluto zu den äußeren Gasplaneten gezählt, mittlerweile wird er als Zwergplanet klassifiziert.

Diese Planeten bestehen in erster Linie aus Helium und Wasserstoff, sie haben eine hohe Rotationsgeschwindigkeit und eine niedrige Dichte (zwischen $0,7 \text{ g/cm}^3$ bis $1,66 \text{ g/cm}^3$),¹ zudem werden sie von Monden umkreist und besitzen Ringe.

1.2. Newtonsches Gravitationsgesetz

Das **Gravitationsgesetz** wurde erstmals 1686 von Isaac Newton formuliert und beschreibt die Kraft, welche zwischen zwei Massenpunkten entsteht.

Es lautet in vektorieller Schreibweise

$$\vec{F}_1 = -G \cdot m_1 \cdot m_2 \cdot \frac{\vec{r}_2 - \vec{r}_1}{|\vec{r}_2 - \vec{r}_1|^3} \quad (1)$$

Hierbei ist F_1 die Kraft, die auf den Körper 1 wirkt, $G = 6,674 \cdot 10^{-11} \text{ m}^3/(\text{kg} \cdot \text{s}^2)$ die Gravitationskonstante, m_1 die Masse des ersten Körpers, m_2 die Masse des zweiten Körpers, r_1 und r_2 die jeweiligen Ortsvektoren der Körper 1 bzw. 2.

Hat man nun ein System aus **mehreren Körpern** k , so lässt sich die obige Gleichung (1) umschreiben in folgende Form:

$$\vec{F}_1 = -G \cdot m_1 \cdot \sum_{i=2}^k m_i \cdot \frac{\vec{r}_i - \vec{r}_1}{|\vec{r}_i - \vec{r}_1|^3} \quad (2)$$

1.3. Baryzentrum

Die Aussage, dass die Planeten die Sonne umkreisen, ist eigentlich nicht ganz korrekt, denn Sonne und Planeten umkreisen deren gemeinsamen Massenschwerpunkt, welcher auch **Baryzentrum** genannt wird.

Da die Sonne der mit Abstand schwerste Körper des Sonnensystems ist, liegt das Baryzentrum in der Nähe des Massenzentrums der Sonne.

Durch die Bewegung der Planeten verändert sich ständig die Position des Baryzentrums, welches mal tief im Inneren der Sonne liegt, mal gerade außerhalb der Sonne in der Nähe ihrer Oberfläche.⁷

Dabei haben die massenreichsten Planeten Jupiter (0,1 % der Sonnenmasse) und Saturn (0,03 % der Sonnenmasse) den größten Einfluss auf die Position des Baryzentrums.

Der Ortsvektor des Baryzentrums (Massenmittelpunkt) mehrerer Körper berechnet sich wie folgt:

$$\vec{r}_s = \frac{1}{M} \sum_i m_i \vec{r}_i \quad (3)$$

Hierbei ist \vec{r}_i der Ortsvektor des i -ten Körpers und $M = \sum_i m_i$ die Gesamtmasse.

1.4. Runge-Kutta-Methode

Mit Hilfe der **Runge-Kutta-Methode** lassen sich Anfangswertprobleme näherungsweise lösen. Nimmt man ein allgemeines Anfangswertproblem an:

$$y'(t) = f(t, y(t)) \quad (4)$$

mit gegebenen Anfangsbedingungen $y(t_0) = y_0$

So lässt sich die exakte Lösung $y(t)$ mit y_n mit folgender Formel annähern:

$$y_{n+1} = y_n + h \cdot \sum_i b_i k_i \quad (5)$$

Dabei bezeichnet h die Schrittweite, b_i die gewichteten Vorfaktoren und k_i die Zwischenschritte.

In unserer Simulation haben wir das **4 stufige Runge-Kutta-Verfahren** angewendet, bei diesem berechnen sich die Zwischenschritte wie folgt:

$$k_1 = h \cdot f(y(x), x) \quad (6)$$

$$k_2 = h \cdot f(y(x) + 0.5 \cdot k_1, x + 0.5 \cdot h) \quad (7)$$

$$k_3 = h \cdot f(y(x) + 0.5 \cdot k_2, x + 0.5 \cdot h) \quad (8)$$

$$k_4 = h \cdot f(y(x) + k_3, x + h) \quad (9)$$

Hierbei entspricht k_1 dem Differential am Beginn des Intervalls, k_2 dem Differential an dem Punkt $y(x) + 0.5 \cdot k_1$ innerhalb des Intervalls, k_3 dem Differential an dem Punkt $y(x) + 0.5 \cdot k_2$ innerhalb des Intervalls und k_4 dem Differential an dem Punkt $y(x) + k_3$ am Ende des Intervalls.

Und daraus ergibt sich mit den gewichteten Vorfaktoren $y(x + h)$:

$$y(x + h) = y(x) + \frac{k_1 + 2 \cdot k_2 + 2 \cdot k_3 + k_4}{6} + \mathcal{O}(h^5) \quad (10)$$

2. Simulation

2.1. Wahl der Einheiten

Um nicht mit sehr großen Zahlen rechnen zu müssen werden die physikalischen Größen sinnvoll skaliert.

Massen werden in Erdmassen angegeben, Längen in Astronomischer Einheit (A.U., entspricht dem mittleren Abstand zwischen Sonne und Erde), die Zeit in Erdjahren.

2.2. Näherungen

Für die Simulation werden einige Näherungen angenommen. Zum einen werden nur die acht Planeten und die Sonne betrachtet, weitere Himmelskörper fließen nicht mit in die Simulation ein.

Die einzelnen Körper werden als Massenpunkte und nicht als ausgedehnte Körper betrachtet. Es wird als Startbedingung angenommen, dass sich die Planeten auf Kreisbahnen bewegen und sich alle in einer Ebene befinden.

Auch die entnommenen Daten der Planeten (Distanz zur Sonne, Geschwindigkeit) sind Mittelwerte, da die Bahnen eigentlich elliptisch sind.

2.3. Code Snippets mit Kommentaren

Das Makefile kompiliert das Program (main.cpp, add.cpp und planet.cpp), führt es aus, erstellt alle Plots, sowie das GIF-Video und löscht daraufhin, die nicht mehr benötigten Dateien.

2.3.1. class Planet

Die Klasse Planet enthält alle Daten eines Planeten, die Daten werden aus der Tabelle ausgelesen und in einem Vektor namens *“myData”* gespeichert

```
1  class Planet {
2  public: [...]
3      Planet( std::string val ) {
4          this->myData = new std::string[ dataNr ];
5          std::string value;
6          std::stringstream data_stream( val );
7          int i=0;
8          while ( getline( data_stream, value, ',' ) and i<dataNr ) {
9              this->myData[ i ] = value;
10             i++;
11         } [...]
```

Für jede Größe (Masse, Distanz, ...) enthält die Klasse eine *“get_quantity”* Funktion, welche als Ausgabewert die gewünschte Größe enthält. Zudem wird die Winkelgeschwindigkeit in der Funktion *“angular_velocity”* und die Distanz zum Massenschwerpunkt in der Funktion *“calculated_distance”* berechnet.

```
1  [...]
2      double get_orbit_velocity(void) {return orbit_velocity;}
3      double angular_velocity(void) {return 1./ orbit_period;}
4  [...]
```

Die Klasse Planet enthält die Funktion findParam, welche die Daten eines bestimmten Parameters (zum Beispiel die Masse) als Return-Wert ausgibt, wird der Parameter nicht gefunden, wird ein leerer String ausgegeben:

```
1  std::string Planet::findParam( std::string param, int* index ) {
2      for ( int i=0; i<dataNr; i++ ) {
3          if ( dataName[ i ]==param ) {
4              if ( index!=NULL ) { *index = i; }
5              return this->myData[ i ];
6          }
7      }
8      if ( index!=NULL ) { *index = -1; }
```

```

9     return "";
10 }

```

Mit der Funktion `initialize()` werden die einzelnen Parameter initialisiert

```

1 void Planet::initialize () {
2     std::string::size_type sz;
3     name = planetName;
4     symbol = findParam(dataName[5]).c_str();
5     orbit_velocity = std::stod (findParam(dataName[4]).c_str(), &sz);
6     orbit_period = std::stod (findParam(dataName[3]).c_str(), &sz);
7     distance = std::stod (findParam(dataName[1]).c_str(), &sz);
8     mass = std::stod (findParam(dataName[2]).c_str(), &sz);
9     GtotalMass = 39.4836;
10 }

```

2.3.2. Funktionen

Im Folgenden werden nur einige Funktionen des Codes vorgestellt.

evaluate Die Funktion `evaluate` berechnet die Beschleunigung durch die Gravitationskraft (2) zu einer bestimmten Zeit t und gibt je nach Vorgabe (Parameter i) entweder die Geschwindigkeit ($i=0,1$) eines Planeten j bzw. seine Beschleunigung ($i=2,3$) aus (hier als Beispiel nur die x-Richtung, y-Richtung analog):

```

1 for (int i=0; i<8; i++) {
2     if (i == j)
3         continue;
4     ax += -G * (solar_system[i]->get_mass()) * (x[0][j] - x[0][i]) /
           / pow(sqrt(pow(x[0][j]-x[0][i],2)+pow(x[1][j]-x[1][i],2)),3);
5     [...]
6 }
7 ax += -G * sunMass * (x[0][j] - 0) / pow(sqrt(pow(x[0][j] - 0,2)+
           pow(x[1][j] - 0,2)),3);
8 [...]
9 switch (i) {
10     case 0: return x[2][j];
11     case 1: return x[3][j];
12     case 2: return ax;
13     case 3: return ay;

```

solve Die Funktion solve berechnet nach dem Runge-Kutta-Verfahren die neue Position und Geschwindigkeit eines Planeten j. Dabei werden zunächst die 4 Zwischenschritte k1-k4 berechnet (6), als Beispiel hier die Berechnung von k2:

```
1  for (i=0; i<nVar; i++)
2  inp[i][j] = vars[i][j]+k1[i]*h/2;
3  for (i=0; i<nVar; i++)
4  k2[i] = evaluate(i,t+h/2,inp, solar_system, j);
```

Diese werden dann mit den gewichteten Vorfaktoren multipliziert:

```
1  for (i=0; i<nVar; i++) {
2  vars[i][j] = vars[i][j]+(k1[i]+2*k2[i]+2*k3[i]+k4[i])*h/6;
3  }
```

solveSun Diese Funktion berechnet die Position des Baryzentrums, da die Sonne um das Baryzentrum kreist, werden diese Daten später verwendet, um die Bewegung der Sonne zu ermitteln.

energy_conservation Diese Funktion berechnet die gesamte kinetische und potentielle Energie des Systems, sowie daraus die Gesamtenergie, zudem werden Impuls und Drehimpuls des Systems berechnet und die Ergebnisse in einem 5dim-Vektor gespeichert. Zunächst wird in einer for-Schleife die kinetische $E_{kin} = 0.5 \cdot m \cdot v^2$ und die potentielle Energie $E_{pot} = G \cdot \sum_j m_j \frac{1}{r_j}$ der einzelnen Planeten berechnet und addiert, schließlich wird die kinetische und potentielle Energie der Sonne noch hinzugefügt. Die Geschwindigkeit der Sonne berechnet sich dabei aus der Distanz, die sie in der Zeit Δt zurück gelegt hat (diese wird der Funktion als Parameter dx übergeben), geteilt durch die Zeit Δt

```
1  E_kin += 0.5*sunMass*pow((dx/stepsize),2);
```

Impuls ($p = m \cdot v$) und Drehimpuls ($L = r \cdot m \cdot v$) werden analog berechnet.

2.3.3. main-Funktion

Zu Beginn werden die Daten aus der Tabelle der Planeteneigenschaften ausgelesen, für jeden Planeten wird ein Objekt der Klasse Planet initialisiert mit den dazu passenden Daten (Orbitgeschwindigkeit, Periode, Distanz zur Sonne, Masse). Es wird ein Vektor namens "solar_system" angelegt, jedes Element dieses Vektors entspricht einer Klasse des jeweiligen Planeten.


```

1  i=0;
2  while( getline( data , line ) ) {
3      solar_system[i] = new Planet( line );
4      i++;
5  }

```

Dann werden mit der Funktion initialize() die einzelnen Klassen initialisiert 2.3.1.

Zum Speichern der Position und der Geschwindigkeit der einzelnen Planeten wird eine **4x8 Matrix namens vars[4][8]** initialisiert, dabei steht die jeweilige Zeile für die x- bzw. y-Position (vars [0] [i] bzw. vars [1] [i]) und die Geschwindigkeit in x- bzw. y-Richtung (vars [2] [i] bzw. vars [3] [i]), jede Spalte steht für einen Planeten.

Es wird zunächst die **totale Masse** des gesamten Systems berechnet:

```

1  double totalMass = sunMass;
2  for ( int j=0; j<8; j++) {
3      totalMass += solar_system[j]->get_mass();
4  }

```

Und dann die Position des **Massenschwerpunktes (Baryzentrums)** nach Gleichung (3) bestimmt, dabei wird mit Hilfe einer for-Schleife jeder Planet durchgegangen, seine Masse mit seiner Position multipliziert und durch die Gesamtmasse geteilt.

```

1  double xBarycenter = 0.0;
2  double yBarycenter = 0.0;
3  for ( int j=0; j<8; j++) {
4      xBarycenter += ( solar_system[j]->get_mass() ) * ( solar_system[j]->↵
        calculated_distance() ) * cos(2*pi* ( solar_system[j]->↵
        angular_velocity() ) * startTime) / totalMass;
5      yBarycenter += ( solar_system[j]->get_mass() ) * ( solar_system[j]->↵
        calculated_distance() ) * sin(2*pi* ( solar_system[j]->↵
        angular_velocity() ) * startTime) / totalMass;
6  }

```

Schließlich wird der **Abstand des Baryzentrums** zum Ursprung berechnet, zu Beginn der Simulation soll das Baryzentrum möglichst nahe am Ursprung liegen und es wird eine dementsprechend möglichst günstige Startzeit gewählt.

Für jeden Planeten wird nun der Abstand (distBarycenter) zwischen Baryzentrum und der eigenen Position ermittelt, sowie der Winkel (phiDistBarycenter).

Um die **Geschwindigkeit der einzelnen Planeten** zu ermitteln werden Zentripetalkraft und Graviationskraft gleichgesetzt und nach der Geschwindigkeit v aufgelöst, die berechneten Werte werden in der 4x8 Matrix vars an der entsprechenden Stelle gespeichert und der Betrag der Geschwindigkeit ermittelt:

```

1  vars[2][j] = sqrt(G * totalMass / distBarycenter) * cos(↵
    phiDistBarycenter + pi/2);
2  vars[3][j] = sqrt(G * totalMass / distBarycenter) * sin(↵
    phiDistBarycenter + pi/2);
3  orbitVelocity = sqrt(pow(vars[2][j],2) + pow(vars[3][j], 2));

```

Für jeden Planeten und die Sonne wird eine **Output-Datei** angelegt, in welche der Winkel und der Abstand zum Ursprung, sowie die Position und die Geschwindigkeit gespeichert werden.

Nun wird in einer **for-Schleife** für jeden Zeitpunkt $t + \Delta t$ das **Runge-Kutta-Verfahren** angewendet (Funktion solve (2.3.2)), die Position der Sonne (Funktion solveSun) und die Gesamtenergie des Systems ermittelt (Funktion energy-conservation (2.3.2)). Die gewonnenen Daten werden in den entsprechenden Output-Dateien gespeichert.

```

1  for (int iTimes=0;iTimes<nTimeSteps;iTimes++) {
2  double t = h*iTimes;
3  for (int j=0; j<8; j++) {
4  solar_system[j]->initialize();
5  solve(t, h, solar_system, vars, nVar, j);
6  }
7  [...]
8  for (int i=0; i<2; i++) {
9      sunCoord_old[i]=sunCoord[i];
10     }
11  solveSun(sunCoord, vars, solar_system);
12  dx=sqrt(pow(sunCoord_old[0]-sunCoord[0],2)+pow(sunCoord_old[1]-↵
    sunCoord[1],2));
13  energy_conservation (solar_system,vars, h, dx, energy);
14  }

```

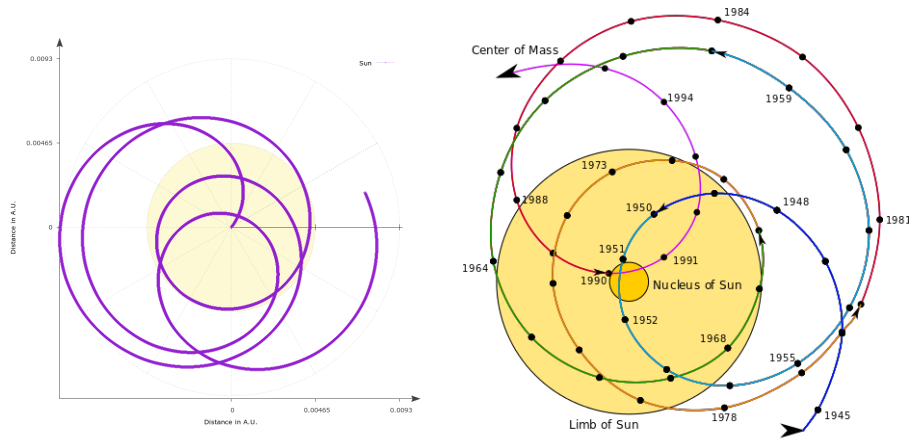
Am Ende der main-Funktion werden für jeden Planeten die mittlere Orbitgeschwindigkeit, die mittlere Periode und die mittlere Distanz zur Sonne und mit den Anfangsbedingungen bzw. den Daten von NASA¹² verglichen. Schließlich werden alle Vektoren, welche mit Hilfe des Befehls **new** erschaffen wurden mit Hilfe von **delete** gelöscht.

3. Auswertung

3.1. Beeinflusst die Bewegung der Planeten die Position der Sonne?

Die Bewegung der Planeten beeinflusst die Position der Sonne, die sich um den gemeinsamen Massenschwerpunkt (Baryzentrum) dreht. Den größten Einfluss haben die massenreichsten Planeten Jupiter (0,1 % der Sonnenmasse) und Saturn (0,03 % der Sonnenmasse). Die Ergebnisse unserer Simulation stimmen dabei mit der tatsächlichen

Bewegung der Sonne sehr gut überein.

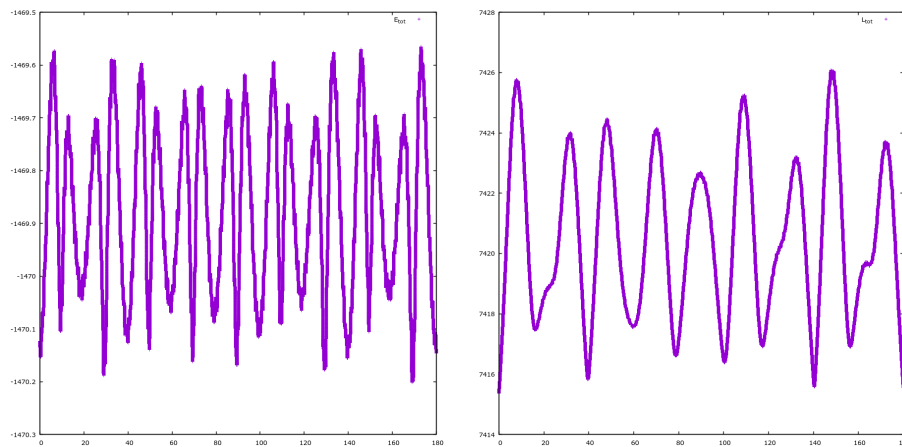


(a) Bewegung der Sonne in 50 Jahren aus der Simulation (b) tatsächliche Bewegung der Sonne im Laufe der Zeit¹¹

Abbildung 2: Vergleich der Bewegung der Sonne

3.2. Energieerhaltung

Die Energieerhaltung ist eines der grundlegendsten Gesetze der Physik. Bei der Anwendung von numerischen Integrationen, kann es jedoch dazu kommen, dass die Energie mit der Zeit fluktuiert bzw. bei einer Simulation, die über einen langen Zeitraum läuft, dass die Energie exponentiell wächst oder fällt. Dies kommt dadurch zu Stande, dass in jedem Schritt der berechnete Wert der Geschwindigkeit um einen kleinen Betrag von dem tatsächlichen Wert der Geschwindigkeit abweicht.

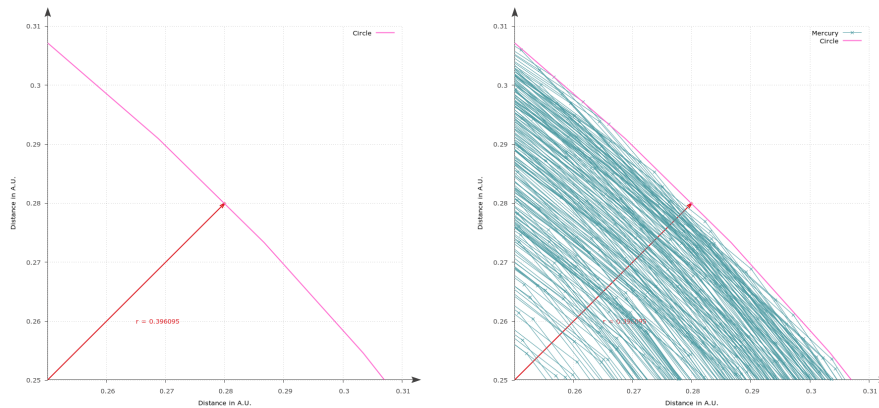


(a) Gesamtenergie des Sonnensystems gegen die Zeit (b) Gesamtdrehimpuls des Sonnensystems gegen die Zeit

Aus der Fluktuation der Energie lässt sich die Güte der Simulation abschätzen.

3.3. Stepsize

In der Simulation wurde eine Schrittgröße von $h=0,001$ gewählt. Würde man eine noch kleinere Schrittgröße wählen, benötigt man eine größere Rechenzeit, bei einer größeren Schrittgröße bricht die Simulation nach relativ schnell aus (beispielsweise bei einer Schrittgröße von $h=0,01$ nach etwa 150 Jahren).



(a) Ausschnitt der Planetenbahn des Merkurs bei Schrittweite $h = 0.001$ (b) Ausschnitt der Planetenbahn des Merkurs bei Schrittweite $h = 0.01$

Abbildung 4: Vergleich der Planetenbahn des Merkurs bei verschiedenen Schrittweiten

3.4. Statistische Auswertung der gewonnen Größen

Um eine statistische Auswertung der simulierten Größen zu erhalten, haben wir die Simulation über einen Zeitraum von 4950 Erdjahren mit einer Schrittweite von $h = 0,001$ laufen lassen.

Exemplarisch werden im Folgenden die Werte der statistischen Auswertung für die **mittlere Orbitgeschwindigkeit**, die **mittlere Orbitperiode** und den **mittleren Abstand zur Sonne** von Merkur, welcher der Sonne am nächsten liegt und Neptun, dessen Umlaufbahn am weitesten entfernt ist von der Sonne, vorgestellt und mit den Daten der NASA verglichen¹² (die kompletten Daten der statistischen Auswertung von allen Planeten befinden sich im AnhangA).

Für **Merkur** liegt die mittlere Bahngeschwindigkeit der Simulation bei $v_{m_orbit} \approx 9,984 A.U./year$, der Wert weicht nur um etwa 0,019% von den Daten der NASA ab. Die mittlere Orbitperiode hat nach unserer Simulation eine Dauer von 0,249 Erdjahren, was etwa 3,9% über dem NASA-Wert (0,24 Erdjahre) liegt. Der mittlere Abstand zur Sonne beträgt in der Simulation 0,396 A.U. und liegt damit um circa 2,3% höher als der von der NASA

angegebene Wert.

Die mittlere Bahngeschwindigkeit bei unserer Simulation des **Neptuns** liegt bei $v_{m_orbit} \approx 1,1456 \text{ A.U./year}$, dieser Wert liegt nur um 0,005% unterhalb des NASA-Wertes. Die mittlere Orbitperiode dauert nach Simulation 164,98 Erdjahre, 0,1% niedriger als die mittlere Orbitperiode laut NASA und auch der mittlere Abstand zur Sonne, welcher in unserer Simulation bei circa 30,08 A.U. liegt, weicht nur um etwa 0,03% vom tatsächlichen Wert ab.

In der folgenden Tabelle ist die Varianz der Radien und die Exzentrizität der elliptischen Umlaufbahn aller Planeten und der Sonne angegeben:

Name	Varianz (Radius)	MAX r	MIN r	max. Exzentrizität	Exzentrizität NASA
Merkur	4.021E-09	0.3962988	0.396	0,0371	0.02056
Venus	2.77179E-08	0.724934923	0.724308847	0.041551352	0.00677323
Erde	5.01872E-08	1.002373548	1.00E+00	0.042283991	0.01671022
Mars	4.77821E-08	1.533545128	1.53E+00	0.035604708	0.09341233
Jupiter	1.04568E-05	5.222988905	5.21223492	0.06413816	0.04839266
Saturn	5.16041E-06	9.568689055	9.56E+00	0.043853064	0.05386179
Uranus	3.57905E-05	19.23390134	19.2064495	0.053408709	0.04716771
Neptun	6.34744E-05	30.09856444	30.06503686	0.047186989	0.00858587
Sonne	4.69983E-06	0.010089836	1.34783E-05	–	–

Die Sonne hat in der Simulation einen mittleren Abstand 0,0055 A.U. von dem Ursprung.

Allgemein kann man sagen, dass die Werte unserer Simulation sehr gut mit den von der NASA angegebenen Werten übereinstimmen.

3.5. Ausblick

Die Rechenzeit des Codes lässt sich noch verkürzen, in dem man die mathematischen Funktionen, wie beispielsweise `pow()` ausschreiben würde. Da man mit diesem Code noch vernünftige Werte in einer annehmbaren Rechenzeit erhalten hat und der Code mit Hilfe dieser Funktionen deutlich übersichtlicher und lesbarer ist, haben wir darauf verzichtet.

Die Simulation würde noch genauer werden, wenn man anstatt kreisförmigen Bahnen als Startbedingung, elliptische annimmt und noch weitere Himmelskörper mit einbezieht.

A. Statische Auswertung

nTimeSteps: 4950000 h: 0.0010000000

After simulation via Runge Kutta: **Mercury**

Mean orbit velocity from NASA: 9.9841000000 A.U/yr
Calculated orbit velocity initial condition: 9.9840582797 A.U/yr
Simulated mean orbit velocity: 9.9821693931 A.U/yr
Absolute deviation of orbit velocity with NASA: -0.0019306069 A.U/yr
Relative deviation of orbit velocity with NASA: 0.0001933681
Absolute deviation of orbit velocity with initial condition: -0.0018888866 A.U/yr
Relative deviation of orbit velocity with initial condition: 0.0001891903

Mean orbit period from NASA: 0.2400000000 yr
Calculated orbit period initial condition: 0.2492709449 yr
Simulated mean orbit period: 0.2493586388 yr
Absolute deviation of orbit period with NASA: 0.0093586388 yr
Relative deviation of orbit period with NASA: -0.0389943283
Absolute deviation of orbit period with initial condition: 0.0000876939 yr
Relative deviation of orbit period with initial condition: -0.0003518014

Mean distance to Sun from NASA: 0.3871000000 A.U
Calculated distance initial condition: 0.3960945794 A.U
Simulated mean distance to Sun: 0.3961589624 A.U
Absolute deviation of simulated mean distance with NASA: 0.0090589624 A.U
Relative deviation of distance with NASA: -0.0234021246
Absolute deviation of distance with initial condition: 0.0000643830 A.U
Relative deviation of distance with initial condition: -0.0001625446

After simulation via Runge Kutta: **Venus**

Mean orbit velocity from NASA: 7.3825000000 A.U/yr
Calculated orbit velocity initial condition: 7.3825608555 A.U/yr
Simulated mean orbit velocity: 7.3807989270 A.U/yr
Absolute deviation of orbit velocity with NASA: -0.0017010730 A.U/yr
Relative deviation of orbit velocity with NASA: 0.0002304196
Absolute deviation of orbit velocity with initial condition: -0.0017619285 A.U/yr
Relative deviation of orbit velocity with initial condition: 0.0002386609

Mean orbit period from NASA: 0.6200000000 yr
Calculated orbit period initial condition: 0.6165703908 yr

Simulated mean orbit period: 0.6168621817 yr
Absolute deviation of orbit period with NASA: -0.0031378183 yr
Relative deviation of orbit period with NASA: 0.0050609973
Absolute deviation of orbit period with initial condition: 0.0002917909 yr
Relative deviation of orbit period with initial condition: -0.0004732484

Mean distance to Sun from NASA: 0.7233000000 A.U
Calculated distance initial condition: 0.7244523612 A.U
Simulated mean distance to Sun: 0.7246222268 A.U
Absolute deviation of simulated mean distance with NASA: 0.0013222268 A.U
Relative deviation of distance with NASA: -0.0018280476
Absolute deviation of distance with initial condition: 0.0001698656 A.U
Relative deviation of distance with initial condition: -0.0002344745

After simulation via Runge Kutta: **Earth**

Mean orbit velocity from NASA: 6.2783000000 A.U/yr
Calculated orbit velocity initial condition: 6.2783281346 A.U/yr
Simulated mean orbit velocity: 6.2768537115 A.U/yr
Absolute deviation of orbit velocity with NASA: -0.0014462885 A.U/yr
Relative deviation of orbit velocity with NASA: 0.0002303631
Absolute deviation of orbit velocity with initial condition: -0.0014744231 A.U/yr
Relative deviation of orbit velocity with initial condition: 0.0002348433

Mean orbit period from NASA: 1.0000000000 yr
Calculated orbit period initial condition: 1.0024632803 yr
Simulated mean orbit period: 1.0029344844 yr
Absolute deviation of orbit period with NASA: 0.0029344844 yr
Relative deviation of orbit period with NASA: -0.0029344844
Absolute deviation of orbit period with initial condition: 0.0004712041 yr
Relative deviation of orbit period with initial condition: -0.0004700462

Mean distance to Sun from NASA: 1.0000000000 A.U
Calculated distance initial condition: 1.0016883331 A.U
Simulated mean distance to Sun: 1.0019238225 A.U
Absolute deviation of simulated mean distance with NASA: 0.0019238225 A.U
Relative deviation of distance with NASA: -0.0019238225
Absolute deviation of distance with initial condition: 0.0002354895 A.U
Relative deviation of distance with initial condition: -0.0002350926

After simulation via Runge Kutta: **Mars**

Mean orbit velocity from NASA: 5.0755000000 A.U/yr
Calculated orbit velocity initial condition: 5.0754748717 A.U/yr
Simulated mean orbit velocity: 5.0743234921 A.U/yr
Absolute deviation of orbit velocity with NASA: -0.0011765079 A.U/yr
Relative deviation of orbit velocity with NASA: 0.0002318014
Absolute deviation of orbit velocity with initial condition: -0.0011513796 A.U/yr
Relative deviation of orbit velocity with initial condition: 0.0002268516

Mean orbit period from NASA: 1.8800000000 yr
Calculated orbit period initial condition: 1.8974147209 yr
Simulated mean orbit period: 1.8982979987 yr
Absolute deviation of orbit period with NASA: 0.0182979987 yr
Relative deviation of orbit period with NASA: -0.0097329780
Absolute deviation of orbit period with initial condition: 0.0008832778 yr
Relative deviation of orbit period with initial condition: -0.0004655164

Mean distance to Sun from NASA: 1.5237000000 A.U
Calculated distance initial condition: 1.5327067827 A.U
Simulated mean distance to Sun: 1.5330724241 A.U
Absolute deviation of simulated mean distance with NASA: 0.0093724241 A.U
Relative deviation of distance with NASA: -0.0061510954
Absolute deviation of distance with initial condition: 0.0003656414 A.U
Relative deviation of distance with initial condition: -0.0002385592

After simulation via Runge Kutta: **Jupiter**

Mean orbit velocity from NASA: 2.7523000000 A.U/yr
Calculated orbit velocity initial condition: 2.7523019817 A.U/yr
Simulated mean orbit velocity: 2.7494212668 A.U/yr
Absolute deviation of orbit velocity with NASA: -0.0028787332 A.U/yr
Relative deviation of orbit velocity with NASA: 0.0010459373
Absolute deviation of orbit velocity with initial condition: -0.0028807149 A.U/yr
Relative deviation of orbit velocity with initial condition: 0.0010466565

Mean orbit period from NASA: 11.8600000000 yr
Calculated orbit period initial condition: 11.8989595668 yr
Simulated mean orbit period: 11.9241587887 yr
Absolute deviation of orbit period with NASA: 0.0641587887 yr
Relative deviation of orbit period with NASA: -0.0054096786
Absolute deviation of orbit period with initial condition: 0.0251992219 yr

Relative deviation of orbit period with initial condition: -0.0021177668

Mean distance to Sun from NASA: 5.2029000000 A.U

Calculated distance initial condition: 5.2122495828 A.U

Simulated mean distance to Sun: 5.2178209236 A.U

Absolute deviation of simulated mean distance with NASA: 0.0149209236 A.U

Relative deviation of distance with NASA: -0.0028678090

Absolute deviation of distance with initial condition: 0.0055713408 A.U

Relative deviation of distance with initial condition: -0.0010688937

After simulation via Runge Kutta: **Saturn**

Mean orbit velocity from NASA: 2.0320000000 A.U/yr

Calculated orbit velocity initial condition: 2.0319974421 A.U/yr

Simulated mean orbit velocity: 2.0315910511 A.U/yr

Absolute deviation of orbit velocity with NASA: -0.0004089489 A.U/yr

Relative deviation of orbit velocity with NASA: 0.0002012544

Absolute deviation of orbit velocity with initial condition: -0.0004063910 A.U/yr

Relative deviation of orbit velocity with initial condition: 0.0001999958

Mean orbit period from NASA: 29.4500000000 yr

Calculated orbit period initial condition: 29.5682790174 yr

Simulated mean orbit period: 29.5786032974 yr

Absolute deviation of orbit period with NASA: 0.1286032974 yr

Relative deviation of orbit period with NASA: -0.0043668352

Absolute deviation of orbit period with initial condition: 0.0103242800 yr

Relative deviation of orbit period with initial condition: -0.0003491674

Mean distance to Sun from NASA: 9.5367000000 A.U

Calculated distance initial condition: 9.5624534999 A.U

Simulated mean distance to Sun: 9.5638792787 A.U

Absolute deviation of simulated mean distance with NASA: 0.0271792787 A.U

Relative deviation of distance with NASA: -0.0028499668

Absolute deviation of distance with initial condition: 0.0014257788 A.U

Relative deviation of distance with initial condition: -0.0001491018

After simulation via Runge Kutta: **Uranus**

Mean orbit velocity from NASA: 1.4333000000 A.U/yr

Calculated orbit velocity initial condition: 1.4332987351 A.U/yr

Simulated mean orbit velocity: 1.4332623283 A.U/yr
Absolute deviation of orbit velocity with NASA: -0.0000376717 A.U/yr
Relative deviation of orbit velocity with NASA: 0.0000262832
Absolute deviation of orbit velocity with initial condition: -0.0000364068 A.U/yr
Relative deviation of orbit velocity with initial condition: 0.0000254007

Mean orbit period from NASA: 84.0200000000 yr
Calculated orbit period initial condition: 84.2530618401 yr
Simulated mean orbit period: 84.2567794717 yr
Absolute deviation of orbit period with NASA: 0.2367794717 yr
Relative deviation of orbit period with NASA: -0.0028181323
Absolute deviation of orbit period with initial condition: 0.0037176315 yr
Relative deviation of orbit period with initial condition: -0.0000441246

Mean distance to Sun from NASA: 19.1891000000 A.U
Calculated distance initial condition: 19.2195202053 A.U
Simulated mean distance to Sun: 19.2198800477 A.U
Absolute deviation of simulated mean distance with NASA: 0.0307800477 A.U
Relative deviation of distance with NASA: -0.0016040381
Absolute deviation of distance with initial condition: 0.0003598424 A.U
Relative deviation of distance with initial condition: -0.0000187228

After simulation via Runge Kutta: **Neptune**

Mean orbit velocity from NASA: 1.1457000000 A.U/yr
Calculated orbit velocity initial condition: 1.1456990938 A.U/yr
Simulated mean orbit velocity: 1.1456396518 A.U/yr
Absolute deviation of orbit velocity with NASA: -0.0000603482 A.U/yr
Relative deviation of orbit velocity with NASA: 0.0000526736
Absolute deviation of orbit velocity with initial condition: -0.0000594420 A.U/yr
Relative deviation of orbit velocity with initial condition: 0.0000518827

Mean orbit period from NASA: 164.7900000000 yr
Calculated orbit period initial condition: 164.9621149960 yr
Simulated mean orbit period: 164.9810292782 yr
Absolute deviation of orbit period with NASA: 0.1910292782 yr
Relative deviation of orbit period with NASA: -0.0011592286
Absolute deviation of orbit period with initial condition: 0.0189142821 yr
Relative deviation of orbit period with initial condition: -0.0001146583

Mean distance to Sun from NASA: 30.0698000000 A.U
Calculated distance initial condition: 30.0797981308 A.U

Simulated mean distance to Sun: 30.0816862308 A.U
Absolute deviation of simulated mean distance with NASA: 0.0118862308 A.U
Relative deviation of distance with NASA: -0.0003952880
Absolute deviation of distance with initial condition: 0.0018880999 A.U
Relative deviation of distance with initial condition: -0.0000627697

Literatur

- [1] Wikipedia, *Sonnensystem*, <https://de.wikipedia.org/wiki/Sonnensystem>, 7. Januar 2017
- [2] Michael Wapp, *Die neun Planeten*, <http://www.neunplaneten.de/nineplanets/overview.html>, 7. Januar 2017
- [3] Matthias Meier, *Planeten acht und mehr*, <http://www.planeten.ch/sonnensystem/>, 7. Januar 2017
- [4] Zahnplanet, *Abbildung des Sonnensystems*, http://www.zahnplanet.de/assets/background-bilder/_resampled/mainbackgroundimage-sonnensystem.jpg, 7. Januar 2017
- [5] Wikipedia, *Newtonsche Gravitationsgesetz*, https://de.wikipedia.org/wiki/Newtonsches_Gravitationsgesetz, 7. Januar 2017
- [6] Leifi, *Gravitationsgesetz*, <http://www.leifiphysik.de/mechanik/gravitationsgesetz-und-feld>, 7. Januar 2017
- [7] spaceplace NASA, *NASA Barycenter*, <http://spaceplace.nasa.gov/barycenter/en/>, 7. Januar 2017
- [8] Wikipedia, *Wiki Baryzentrum*, <https://de.wikipedia.org/wiki/Baryzentrum>, 7. Januar 2017
- [9] Wikipedia, *Wiki Runge-Kutta*, https://en.wikipedia.org/wiki/Runge%E2%80%93Kutta_methods, 7. Januar 2017
- [10] Physics Buffalo, *Runge Kutta Method*, <http://www.physics.buffalo.edu/phy410-505/2011/topic2/app1/index.html>, 7. Januar 2017
- [11] Real clear science Abbildung, *Sun Movement*, http://www.realclearscience.com/blog/2014/08/technically_the_earth_does_not_orbit_the_sun.html, 9. Januar 2017
- [12] NASA, *dates of the eighth planets*, <http://solarsystem.nasa.gov/planetinfo/chart/chart.cfm>, 9. Januar 2017