Heinrich-Heine-Universität, Düsseldorf WS 06/07

Projektseminar

zu

Numerik gewöhnlicher Differentialgleichungen

Thema: Simulation des äusseren Sonnensystems

Bearbeitet von: Tugba Arslan und Abdullah Demirel

Einleitung

Es ist sehr interessant, die in der Natur auftretenden Phänomene zu verstehen. Auch ist die mathematische Betrachtung und Beschreibung dieser Phänomene sehr interessant. Da die mathematische Erklärung dieser Phänomene oft zu einer faszinierenden Erkenntnis über die Funktionsweise der Systeme in der Natur führen und die Lösung der Problematik zur Beschreibung dieser Phänomene als eine weitere Anwendung der Mathematik angesehen werden kann.
Unsere Aufgabe bestand darin, das äußere Sonnensystem zu simulieren. In dieser Zusammenfassung zu unserem Projektseminar werden wir sowohl die Problematik und die erarbeiteten Ergebnisse als auch die Schwierigkeiten verfassen, die während des Projektseminares aufgetreten sind. Wir hoffen, dass wir durch unsere folgende Arbeit diese Problemstellung angemessen und anschaulich darstellen und lösen konnten.

Simulation des äußeren Sonnensystems

Für die Simulation des äußeren Sonnensystems ist es erforderlich, die Planetenbewegungen mathmatisch zu beschreiben. Die Bewegungen der Planeten hängen sowohl vom Impuls als auch von den Anziehungskräften zwischen den Planeten ab.
Für die Anziehungskraft zwischen zwei Massepunkten gilt das Newtonsche Gravitationsgesetz:

$$F = -G\frac{m_1 m_2}{d^2}$$

F: Anziehungskraft zwischen den Massen
G: Gravitationskonstante
m_1: Masse vom einen Massepunkt
m_2: Masse vom anderen Massepunkt
d: Abstand zwischen den Massenpunkten

Der Impuls eines Massenpunktes beschreibt deren Translationsbewegung und wird folgendermaßen berechnet:

$$p = mv$$

p: Impuls des Massenpunktes
m: Masse des Massenpunktes
v: Geschwindigkeit des Massenpunktes

Weiterhin gilt der Energieerhaltungssatz für die Gesamtenergie:

$$E = K + P$$

E: Gesamtenergie
K: kinetische Energie
P: potentielle Energie

Für die Simulation ist es für uns wichtig, die Bewegungen der Planeten durch Gleichungen zu charakterisieren. In der Physik benutzt man hierzu Bewegungsgleichungen, diese sind mathematisch gesehen Differentialgleichungen.
Es stellte sich heraus, dass das System der Bewegungsgleichungen der Planetenbewegungen einem Hamilton-System entspricht und die zugehörige Hamilton-Funktion für das äußere Sonnensystem ist von der Form:

$$H(p,q) = \frac{1}{2}\sum_{i=0}^{5}\frac{1}{m_i}p_i^T p_i - G\sum_{i=1}^{5}\sum_{j=0}^{i-1}\frac{m_i m_j}{||q_i - q_j||}$$

p: 3x6 Impulsmatrix der Planeten
q: 3x6 Positionsmatrix der Planeten
G: Gravitationskonstante m_i: Masse des i-ten Planeten

Aus der Physik ist es bekannt, dass man für das Aufstellen der Bewegungsgleichungen den Hamilton-Formalismus verwenden kann. Nach diesem Formalismus erhält man die Bewegungsgleichungen zu der oben genannten Hamilton-Funktion. Die Gleichungen sind von folgender Form:

$$\dot{q_j} = \frac{\partial H}{\partial p_j}$$

$$\dot{p_j} = -\frac{\partial H}{\partial q_j}$$

Wir berechnen diese Differentialgleichungen explizit aus:

$$\dot{q_j} = \frac{\partial H}{\partial p_j} = \begin{pmatrix} \frac{p_{j_1}}{m_j} \\ \frac{p_{j_2}}{m_j} \\ \frac{p_{j_3}}{m_j} \end{pmatrix}_{j=0...5}$$

Die Ableitung nach q_j ist etwas aufwendiger:

$$\sum_{i=1}^{5}\sum_{j=0}^{i-1}\frac{m_i m_j}{||q_i - q_j||} = \sum_{j=0}^{0}\frac{m_1 m_j}{||q_1 - q_j||} + \sum_{j=0}^{1}\frac{m_2 m_j}{||q_2 - q_j||} + \sum_{j=0}^{2}\frac{m_3 m_j}{||q_3 - q_j||} + \sum_{j=0}^{3}\frac{m_4 m_j}{||q_4 - q_j||} + \sum_{j=0}^{4}\frac{m_5 m_j}{||q_5 - q_j||}$$

Man kann erkennen, dass die gesamte Summe jeweils 5 Summanden besitzt, die abhängig von q_j mit $j = 0...5$ sind. Also folgt insgesamt für die Bewegungsgleichung:

$$\dot{p_j} = -\frac{\partial H}{\partial q_j} = \begin{pmatrix} G\sum_{k=0,k\neq j}^{5}\frac{m_j m_k}{||q_j - q_k||^3}(q_{j_1} - q_{k_1}) \\ G\sum_{k=0,k\neq j}^{5}\frac{m_j m_k}{||q_j - q_k||^3}(q_{j_2} - q_{k_2}) \\ G\sum_{k=0,k\neq j}^{5}\frac{m_j m_k}{||q_j - q_k||^3}(q_{j_3} - q_{k_3}) \end{pmatrix}_{j=0...5}$$

Sobald wir nun also geeignete Anfangswerte zu Impulsen und Positionen der Planeten wählen kommt es nur noch auf die Wahl des numerischen Verfahrens an, welches diese Problemstellung anständig löst.
Als Anfangswerte haben wir folgende Daten benutzt:

Planet	Masse relativ zur Sonne	Position in $AE = 149.597.870km$	Geschwindigkeit in AE/Tag
Sonne	1.00000597682	0 0 0	0 0 0
Jupiter	0.000954786104043	-3.5023653 -3.8169847 -1.5507963	0.00565429 -0.00412490 -0.00190589
Saturn	0.000285583733151	9.0755314 -3.0458353 -1.6483708	0.00168318 0.00483525 0.00192462
Uranus	0.0000437273164546	8.3101420 -16.2901086 -72521278	0.00354178 0.00137102 0.00055029
Neptun	0.0000517759138449	11.4707666 -25.7294829 -10.8169456	0.00288930 0.00114527 0.00039677
Pluto	$\frac{1}{1.3 \cdot 10^8}$	-15.5387357 -25.2225594 -3.1902382	0.00276725 -0.00170702 -0.00136504

Die Gravitationskonstante entspricht $2.95912208286 \cdot 10^{-4}$. Somit benötigen wir nur noch ein numerisches Verfahren, welches unsere Problemstellung löst.
Das einfachste Verfahren zur Lösung gewöhnlicher Differentialgleichungen ist das explizite Euler Verfahren und sieht wie folgt aus:

$$x_{i+1} = x_i + h\dot{x}_i$$

Für unser Problem erhalten wir demnach:

$$p_{i+1} = p_i + h\dot{p}_i$$

$$q_{i+1} = q_i + h\dot{q}_i$$

Wir benutzen Matlab für die Implementierung. Das Programm sieht dann gekürzt wie folgt aus:

```
%Eingabe der Anfangswerte
m=Massenvektor der Planeten;
G=Gravitationskonstante;
q=3x6 Positionsmatrix der Planeten;
v=3x6 Geschwindigkeitsmatrix der Planeten;

%Berechne 3x6 Impulsmatrix der Planeten
for k=1:6
      p(1:3,k)=m(k).*v(1:3,k);
end

%Berechne Bewegungsgleichungen
qt=dHdp(p,m);
pt=dHdq(q,m,G);

%Expliziter Euler:
y=[p;q];
f=[pt;qt];
h=Schrittweite in Tagen;

for k=1:20000
      hold on
      plot3(y(4,1),y(5,1),y(6,1),'r',y(4,2),y(5,2),y(6,2),'b',...
            y(4,3),y(5,3),y(6,3),'g',y(4,4),y(5,4),y(6,4),'c',...
            y(4,5),y(5,5),y(6,5),'m',y(4,6),y(5,6),y(6,6),'k')
      hold off
*     y=y+h*f-h*repmat([[0;0;0];qt(1:3,1)],1,6);
      p2=y(1:3,1:6);
      q2=y(4:6,1:6);
      qt2=dHdp(p2,m);
      pt2=dHdq(q2,m,G);
      f=[pt2;qt2];
end
```

Die Differenz bei * ist nur deswegen enthalten, da der Impuls der Sonne nicht immer der Nullvektor ist. Da die Sonne im Sonnensystem jedoch dem Zentrum entspricht, erreichen wir durch diese Differenz, dass die Sonne stets im Zentrum befindet. Alle anderen Planeten

müssen dann natürlich ebenfalls in der gleichen Richtung und Länge verschoben werden.
Eine weitere Möglichkeit wäre es, dass man die Werte der Bewegungsgleichungen der Sonne stets als Nullvektor speichert. Insbesondere sollte die Bewegungsgleichung für die Sonne den Nullvektor stets annehmen, welche für die Verschiebung der Sonne verantwortlich ist.
Wir möchten nun auch die Implementierung der Matlab-Funktionen für die Berechnung der Bewegungsgleichungen vorstellen:

```
%Die Funktion dHdp(p,m) zur Berechnung von q̇
%Eingabeparameter sind die Impulsmatrix und der Massenvektor der Planeten
function x=dHdp(p,m)
for k=1:6
       q(:,k)=1/m(k).*p(:,k);
end
x=q;
```

```
%Die Funktion dHdq(q,m,G) zur Berechnung von ṗ
%Eingabeparameter sind die Gravitationskonstante sowie die Positionsmatrix und
der Massenvektor der Planeten
function x=dHdq(q,m,G)
for j=1:6
      for k=1:6
           if k==j
                pt(:,k)=[0;0;0];
           else
                pt(:,k)=(m(k)*m(j)/(norm(q(:,k)-q(:,j))^3).*(q(:,k)-q(:,j));
           end
      end
      p(:,j)=G.*sum(pt,2);
end
x=p;
```

Das Programm liefert uns leider ein nicht zufriedenstellendes Ergebnis, wie man es auch aus Abbildung 1 bzw. der Animation explsolar.avi sehen kann. Das explizite Eulerverfahren ist zwar einfach zu implementieren liefert in unserer Problemstellung jedoch kein ausreichendes Ergebnis.
Bevor wir jedoch ein weiteres Verfahren implementieren möchten wir vorerst den Grund für dieses Ergebnis verstehen. Anscheinend bleibt die Energie nicht erhalten. Hierzu betrachten

wir die Hamilton-Funktion und wie man aus Abbildung 2 sehen kann, wächst diese Funktion streng monoton. Daher kann man von einer Energieerhaltung nicht sprechen.

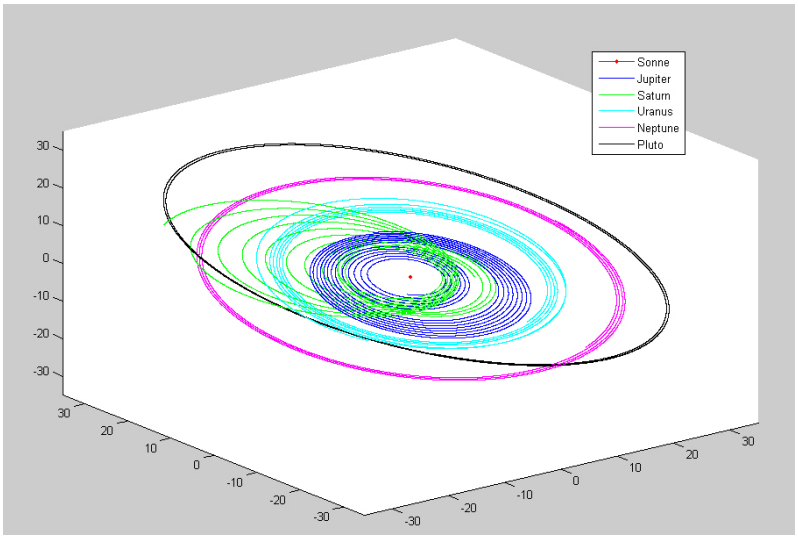

Abbildung 1: Expliziter Eulerverfahren mit Schrittweite h=10

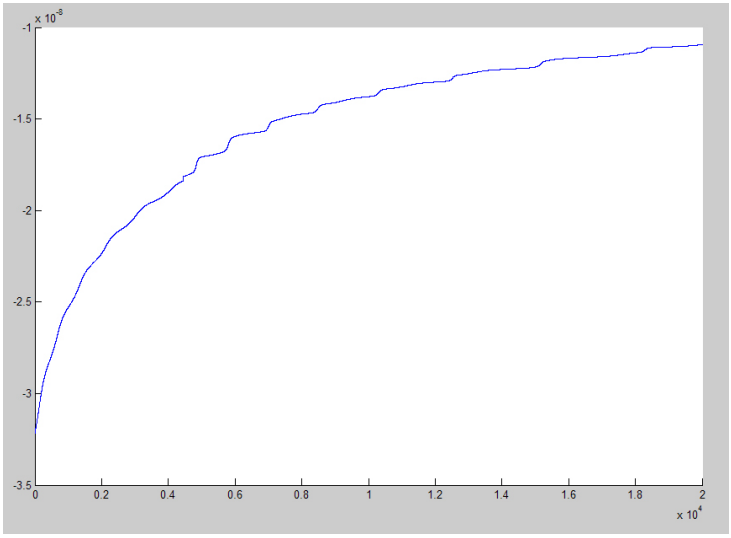

Abbildung 2: logarithmische Darstellung der Hamilton-Funktion(Gesamtenergie)

Das nächste Verfahren, welches wir für die Lösung unserer Problematik verwenden, ist das symplektische Eulerverfahren:

$$p_{n+1} = p_n - h\frac{\partial H}{\partial q}(p_{n+1}, q_n)(1)$$

$$q_{n+1} = q_n + h\frac{\partial H}{\partial p}(p_{n+1}, q_n)(2)$$

Im Allgemeinen müsste man die erste mit dem impliziten Eulerverfahren und die zweite mit dem expliziten Eulerverfahren lösen. Wir haben jedoch die Ableitungen berechnet und wissen, dass $\frac{\partial H}{\partial q}$ nur von q und $\frac{\partial H}{\partial p}$ nur von p abhängig ist. Daher können wir zuerst die erste und anschliessend die zweite
Gleichung jeweils explizit lösen. Die Implementierung in Matlab sieht dann folgendermaßen aus:

```
%Eingabe der Anfangswerte
m=Massenvektor der Planeten;
G=Gravitationskonstante;
q=3x6 Positionsmatrix der Planeten;
v=3x6 Geschwindigkeitsmatrix der Planeten;

%Berechne 3x6 Impulsmatrix der Planeten
for k=1:6
      p(1:3,k)=m(k).*v(1:3,k);
end

%Berechne Bewegungsgleichungen
qt=dHdp(p,m);
pt=dHdq(q,m,G);

%Symplektischer Euler:
h=Schrittweite in Tagen;

for k=1:20000
      hold on
      plot3(y(4,1),y(5,1),y(6,1),'r',y(4,2),y(5,2),y(6,2),'b',...
            y(4,3),y(5,3),y(6,3),'g',y(4,4),y(5,4),y(6,4),'c',...
            y(4,5),y(5,5),y(6,5),'m',y(4,6),y(5,6),y(6,6),'k')
      hold off
      p=p+h*pt;
      qt=dHdp2(p,m);
      q2=y(4:6,1:6);
*     q=q+h*qt-h*repmat(qt(1:3,1),1,6);
      pt2=dHdq(q2,m,G);
      pt=dHdq2(q,m,G);
end
```

Der Grund für die Differenz bei * ist genau dieselbe wie beim expliziten Eulerverfahren. Das Ergebnis dieser Implementierung spiegelt das richtige Verhalten im Sonnensystem wider, was man gut an der Abbildung 3 bzw. der Animation symplsol.avi sehen kann.

Die Abbildung 4 zeigt uns, dass die Energieerhaltung beim symplektischem Eulerverfahren ziemlich gut funktioniert. Wir haben nun also die Planeten und deren Umlaufbahnen

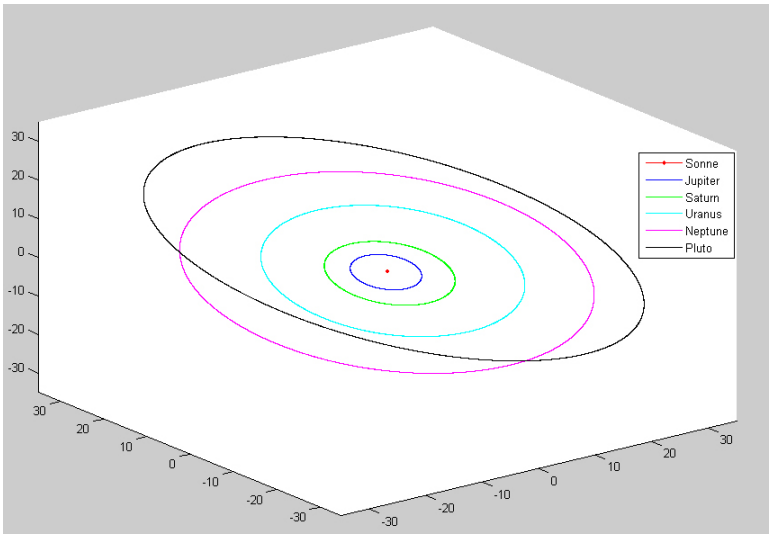

Abbildung 3: Symplektische Eulerverfahren mit Schrittweite h=10

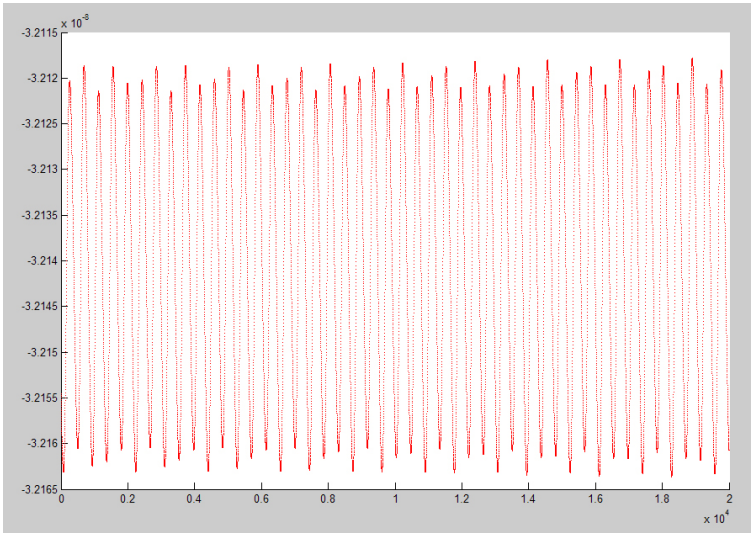

Abbildung 4: logarithmische Darstellung der Hamilton-Funktion(Gesamtenergie)

erfolgreich in unsere Simulation einbezogen. Allerdings sind auch kleinere Körper im äußeren Sonnensystem enthalten wie z.B. Asteroiden, Kometen, Monde usw. und daher fragten wir uns zunächst wie sich beliebig positionierte kleine Körper in unserer Simulation verhalten. Dafür haben wir zufällig gewählte Massen in den Massenvektor eingefügt sowie zufällige Positionen und Anfangsgeschwindigkeiten für die kleinen Körper definiert.
Zunächst haben wir nur einen Körper genommen, dann versuchten wir unsere Simulation mit vier kleinen Körpern zu erweitern (s.Abbildung 5 und 6). Mit steigender Anzahl der Körper

gewann die Laufzeit unserer Simulation immer mehr an Bedeutung. Wir mussten erstaunlicherweise feststellen, dass nicht die Berechnungen sondern die Achsenskalierung sowie die Zuteilung einer Legende auf den Graphen innerhalb der for-Schleife die Laufzeit stark beeinflusste. Durch die Herausnahme der Achsenskalierung und der Legende aus der for-Schleife erreichten wir eine Laufzeitverbesserung um ca. 95%.
Weiterhin war es uns möglich einige Berechnungen zu sparen, da diese kaum Wirkungen auf das Ergebnis hatten und zwar ist es so, dass die kleinen Körper kaum eine Anziehungskraft auf die Planeten haben und daher haben wir auf deren Berechnungen verzichtet. Das anschliessende Ergebnis bestätigte unsere Überlegung (s. Abbildung 7).
Nach dieser Laufzeitverbesserungen konnten wir sogar unsere Simulation durch 100 zufällig positionierten kleinen Körpern ergänzen (s. Abbildung 8).

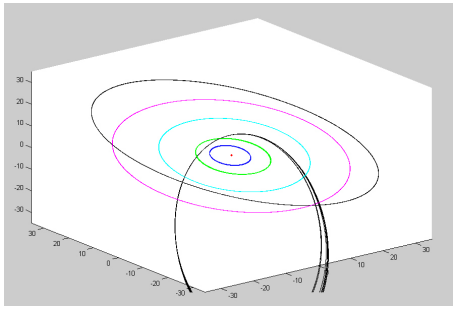

Abbildung 5: Erweiterung um 1 kleinen Körper

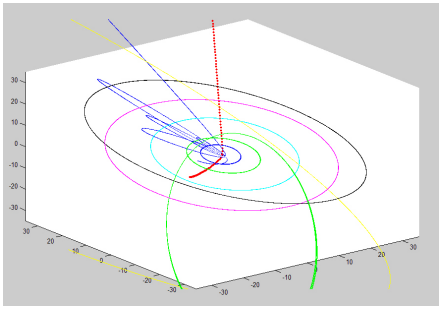

Abbildung 6: Erweiterung um 4 kleine Körper

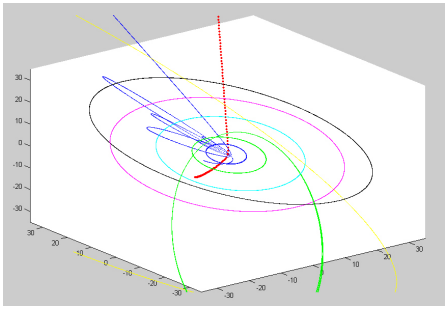

Abbildung 7: 4 kl. Körper nach Laufzeitersparnis

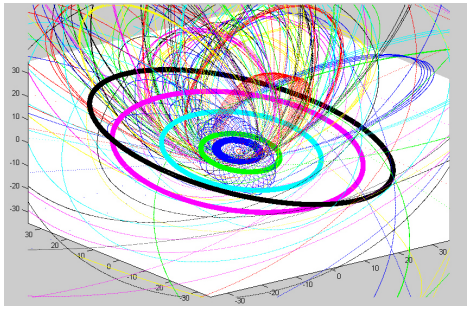

Abbildung 8: Erweiterung um 100 kleine Körper

Das Ergebnis führte uns, wegen der Unübersichtlichkeit der entstehenden Simulation, zu dem Schluss, dass wir statt vielen zufälligen kleinen Körpern die Planetenmonde der Planeten und deren Umlaufbahnen im äußeren Sonnensystem in unserer Simulation darzustellen und diese anschliessend auch zu animieren.
Die Problematik war es, dass wir für die Planetenmonde keine Anfangswerte hatten. Daher haben wir Behörden wie die NASA sowie auch das Deutsche Zentrum für Luft- und

Raumfahrt aus diesem Grund angeschrieben. Wir selbst haben im Internet nach evtl. nützlichen Daten gesucht.
Da unsere E-Mails leider unbeantwortet blieben mussten wir uns mit den Daten begnügen, die wir auf http://www.wikipedia.de gefunden haben. Wir entschieden uns dazu sechs der Planetenmonde in unsere Simulation einzubeziehen.
Wir mussten jedoch einige Umrechnungen vornehmen. Hierzu haben wir uns eine weitere Matlab-Funktion implementiert, welches uns die umgerechneten Daten liefert. Allerdings möchten wir hier nur die Berechnungen kurz erläutern.

1. Berechnung der Massen:
   Die Massen m_{j_q} mit $j_q = 1, ..., 6$ der Planetenmonde in kg wurden durch die Masse der Sonne M_{s_q} in kg aus derselben Quelle dividiert und anschliessend mit der Masse der Sonne aus obiger Tabelle M_{s_t} multipliziert.

$$m_{j_t} = \frac{M_{s_t} m_{j_q}}{M_{s_q}}$$

2. Berechnung der Positionen in AE:
   Zunächst werden die Einheiten der Abstände der Planetenmonde zum zugehörigen Planeten d_{km} von km auf AE umgewandelt. Hierzu rechnet man:

$$d_{ae} = \frac{d_{km}}{149597870}$$

   Nun normiert man jeweils die Ortsvektoren der Planeten p_j und verlängert den ursprünglichen Ortsvektor um den Abstand des Planetenmondes zum jeweiligen Planeten d_{ae}, wodurch man die Position des Planetenmondes erhält.

$$p_{mj} = p_j + \frac{d_{ae}}{||p_j||} p_j$$

3. Berechnung der Geschwindigkeit in AE/Tag:
   Es schlägt fehl, wenn man für die Anfangsgeschwindigkeiten der Planetenmonde die Anfangsgeschwindigkeiten der Planeten verwendet, es kommt stets zu einer Kollision. Da wir wissen, dass die Planetenmonde die Planeten umrunden müssen diese also betragsmäßig eine grössere Geschwindigkeit aufweisen.
   Wir haben zwar später Daten zur mittleren Geschwindigkeit v_{j_m} in km/s gefunden, deren Einheiten wir dann nur umrechnen brauchten (1). Vorher hatten wir allerdings nur einen Wert für die Umlaufszeit t_m der Planetenmonde um den zugehörigen Planeten, was ebenfalls ein zufriedenstellendes Ergebnis lieferte.

Hierzu gingen wir davon aus, dass die Planetenmonde die Planeten umkreisen. Den Umfang vom Kreis können wir berechnen, da der Abstand d_{ae} als Radius dient. Umfang/Umlaufzeit ergibt folglich die Umlaufsgeschwindigkeit v_{j_u}. Also normierten wir den Geschwindigkeitsvektor v_j des Planeten und ergänzten diese um die Umlaufsgeschwindigkeit (2).

$$v_{mj} = v_{j_m} \frac{3600 \cdot 24}{149597870} \qquad (1)$$

$$v_{mj} = v_j + \frac{v_{j_u}}{||v_j||} v_j \text{ mit } v_{j_u} = \frac{2\pi d_{ae}}{t_m} \qquad (2)$$

Unsere Datentabelle für die Anfangswerte, die wir aus den verfügbaren Daten ermittelt haben, sieht dann wie folgt aus:

Planetenmond	Masse relativ zur Sonne	Position in $AE = 149.597.870km$	Geschwindigkeit in AE/Tag
Kallisto	$10^{-7} \cdot 0.54114183819067$	-3.5103013 -3.8256336 -1.5543103	0.00934587 -0.00681797 -0.00315021
Ganymed	$10^{-7} \cdot 0.74532732732209$	-3.4977301 -3.8119331 -1.5487439	0.01054958 -0.00769610 -0.00355595
Rhea	$10^{-7} \cdot 0.01165064295867$	9.0788229 -3.0469400 -1.6489686	0.00319033 0.00916482 0.00364796
Titania	$10^{-7} \cdot 0.01773295651915$	8.3114502 -16.2926731 -7.2532694	0.00560463 0.00216955 0.00087080
Triton	$10^{-7} \cdot 0.10797690767615$	11.4716680 -25.7315049 -10.8177957	0.00522735 0.00207203 0.00071784
Charon	$10^{-7} \cdot 0.00814730276830$	-15.5388040 -25.2226703 -3.1902522	0.00286832 -0.00176937 -0.00141490

Nun setzen wir unsere Werte ein und erhalten unsere Simulation zum äußeren Sonnensystem. Man kann zwar bei Betrachtung der Gesamtdarstellung nicht erkennen, dass die Planeten von

den Planetenmonden umlaufen werden (s. Abbildung bzw. sim1.avi). Doch durch Betrachtung der einzelnen Planeten z.B. von Jupiter (s. Abbildung) kann man erkennen, dass die Planetenmonde den Planeten umlaufen. Wegen der kurzen Umlaufzeiten der Planetenmonde sind natürlich dementsprechend kleine Schrittweiten zu wählen
Für die Erstellung der Animationen nutzten wir eine in Matlab integrierten Erase-Modus, sodass auch der zur Laufzeit erstellte Graph kaum Auswirkungen auf die Laufzeit des Programms hatte.

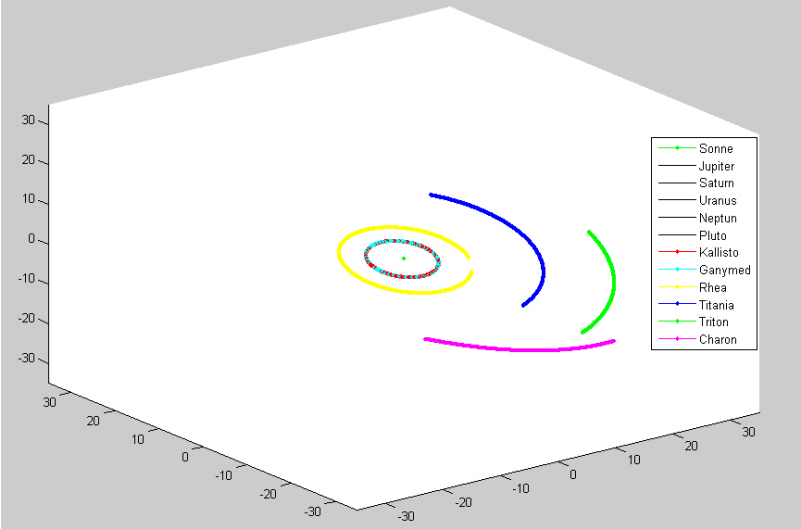

Abbildung 9: Erweiterung um 6 Planetenmonde

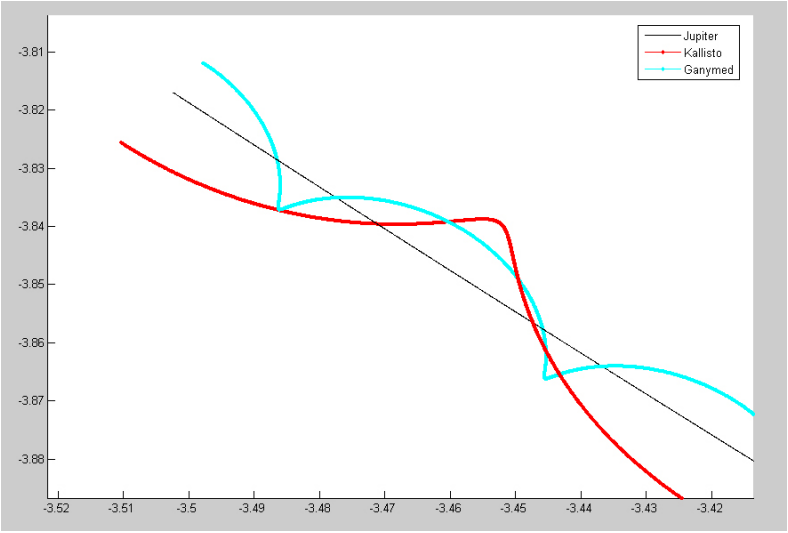

Abbildung 10: Jupiter, Kallisto und Ganymed

Die oben genannte Animation symplsol.avi haben wir dann zum Abschluss etwas anschaulicher gestaltet (s. solar.avi).