Bismarckschule Hannover
Seminarfach: Astronomie und Astrophysik
Computergestützte Himmelsmechanik

Vergleich numerischer Verfahren zur Modellierung des Dreikörperproblems

Tobias Steinbrecher

Lehrkraft: M. Frank
Bearbeitungszeitraum: 25.01.2022 - 22.03.2022
Schuljahr: 2021/2022, Klassenstufe: 12.2
Maximale Seitenanzahl: 15

FORMATIERUNG ÜBERARBEITET

Inhaltsverzeichnis

1 Vorwort

Zur Erkundung der Numerik angewandt auf das Dreikörperproblem, wurde ich vor allem durch die Publikation *On the reliability of N-body simulations* [9] inspiriert. Der Verfasser *Dr. Tjarda Boekholt* riet mir auszuprobieren, inwiefern ich einige seiner bzw. anderer Forschungsergebnisse im Bereich des numerischen Lösens von Dreikörpersystemen, mit verschiedenen Messungen reproduzieren kann. In der angesprochenen Publikation, wurde dafür ein Integrator gemäß dem *Bulirsch-Stoer-Verfahren* auf Basis des *Verlet-Leapfrog-Verfahrens* entwickelt und umfangreich getestet. Das Interesse besteht somit vor allem darin, ein ähnliches Verfahren zu implementieren und im Vergleich zu weiteren Verfahren in Bezug auf das Dreikörperproblem zu testen. Anregungen brachten auch einige Aufgabenstellungen in Bezug auf numerische Simulationen des Dreikörperproblems aus *Practical Course in Astronomy* der *Universität Tübingen* [10]. Darunter befindet sich eine Implementierung des *Euler-*, *Euler-Cromer-* und *Verlet-Verfahrens*, welche auch zur Wahl der Integratoren in dieser Facharbeit gehören. Als langfristige Motivation und Ziel, jedoch nicht als Bestandteil dieser Facharbeit, steht die Anwendung von *Machine-Learning*, welche auch die aktuelle Forschung beschäftigt [11].

2 Einleitung

Ein System mit drei Punktmassen[1] (im Folgenden auch Körper), zwischen welchen die Gravitationskraft wirkt, bezeichnet man als Dreikörperproblem [13]. Das gewöhnliche, nicht relativistische, Dreikörperproblem zeichnet sich durch ein deterministisch chaotisches Verhalten, sowie eine allgemeine analytische Unlösbarkeit aus [14]. Aus diesem Grund eignen sich computergestützte numerische Berechnungen für eine zeitspezifische Geschwindigkeits- und Trajektoriebestimmung der Körper. Die numerischen Verfahren basieren auf wiederholten Berechnungen im Abstand von sehr kleinen Zeitschritten. Die Bewegungsgrößen der drei Körper, werden auf Basis der wirkenden Kräfte für den jeweils nächsten Zeitschritt aktualisiert [1]. Eine Herausforderung bei solch einer numerischen Simulation, besteht in der Instabilität der Dreikörpersysteme. Kleine Abweichungen in den Startbedingungen oder Fehler während der Simulation eines Systems, sorgen über Zeit für eine exponentielle Divergenz von den *echten* Werten [15]. Dadurch sind numerische Simulationen auf einer großen Zeitskala womöglich nicht mehr aussagekräftig [14]. Die Fragestellung der Präzision und Genauigkeit von numerischen Simulationen in Bezug auf Dreikörperprobleme ist relevant für das Verständnis von Stabilität planetarer Systeme, sowie Raumfahrt-Missionen und Dynamik in Galaxien [14]. In dieser Facharbeit soll der Leitfrage nachgegangen werden, inwiefern sich qualitative Unterscheidungen in der Präzision von verschiedenen numerischen Verfahren in Anwendung auf das Dreikörperproblem treffen lassen. Zunächst werden dafür einige physikalische Grundlagen herausgestellt und die implementierten Verfahren umfangreich analysiert. Dabei muss gezielt auf verschiedene Fehlerquellen geachtet werden, welche bei der computergestützten Numerik eine Rolle spielen [9]. Für den Vergleich der Verfahren und zur Beantwortung der Leitfrage, lassen sich Erhaltungsgrößen sowie eine Analyse der Divergenz von perturbierten[2] Systemen und Zeit-Reversibilität heranziehen.

3 Physikalische Grundlagen

3.1 Newtonsches Gravitationsgesetz

Die physikalische Grundlage für das gewöhnliche Dreikörperproblem besteht im Newtonschen Gravitationsgesetz, welches die Kraftwirkung zwischen zwei Körpern (i und j) mit den Massen m_i, m_j und den Positionen $\vec{r}_i, \vec{r}_j$ beschreibt:

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

wobei γ die Gravitationskonstante ist und der normierte Vektor $\frac{(\vec{r}_j - \vec{r}_i)}{|\vec{r}_j - \vec{r}_i|}$ für den Erhalt der Richtung sorgt, jedoch keine Veränderung des Betrages der Kraft verursacht [2]. Bei Anwendung auf drei Körper, wirken auf i, durch die beiden anderen Körper j und k, jeweils zwei Kräfte.

[1] Gesamtmasse ist in einem Punkt vereinigt (dimensionslos) [12]

[2] *gestörten*

Nach Einsetzen von $\vec{F}_{ij} = m_i \cdot \ddot{\vec{r}}_i$ in die Summe aus zweimal Gl. 1 und Division durch m_i, folgt:

$$\begin{aligned} \ddot{\vec{r}}_i &= \gamma \left(\frac{m_j}{|\vec{r}_j - \vec{r}_i|^3}(\vec{r}_j - \vec{r}_i) + \frac{m_k}{|\vec{r}_k - \vec{r}_i|^3}(\vec{r}_k - \vec{r}_i) \right) \\ \ddot{\vec{r}}_i &= \gamma \left(\sum_{j=1}^{3} \frac{m_j}{|\vec{r}_j - \vec{r}_i|^3}(\vec{r}_j - \vec{r}_i) \right) \quad i \neq j \end{aligned} \tag{2}$$

Mit diesen gewöhnlichen Differentialgleichungen (DGLs), lassen sich somit die Beschleunigungen $\ddot{\vec{r}}_i$ zu einem besimmten Zeitpunkt bei gegebenen Positionen, für alle drei Körper ($i = 1, 2, 3$) eines Systems berechnen [2]. Eine DGL ist eine Gleichung, in welcher mindestens eine Ableitung einer Funktion vorkommt. Dabei kann auch die Funktion selbst im Zusammenhang stehen [3]. Hierbei handelt es sich um die Position $\vec{r}$ und um die zweite zeitliche Ableitung $\ddot{\vec{r}}$ (Beschleunigung), welche beide in einer Gleichung vorkommen und somit eine DGL bilden [16]. Eine Visualisierung der wirkenden Beschleunigung $\ddot{\vec{r}}_i$ über ein sogenanntes *Stromfeld* bietet sich an:

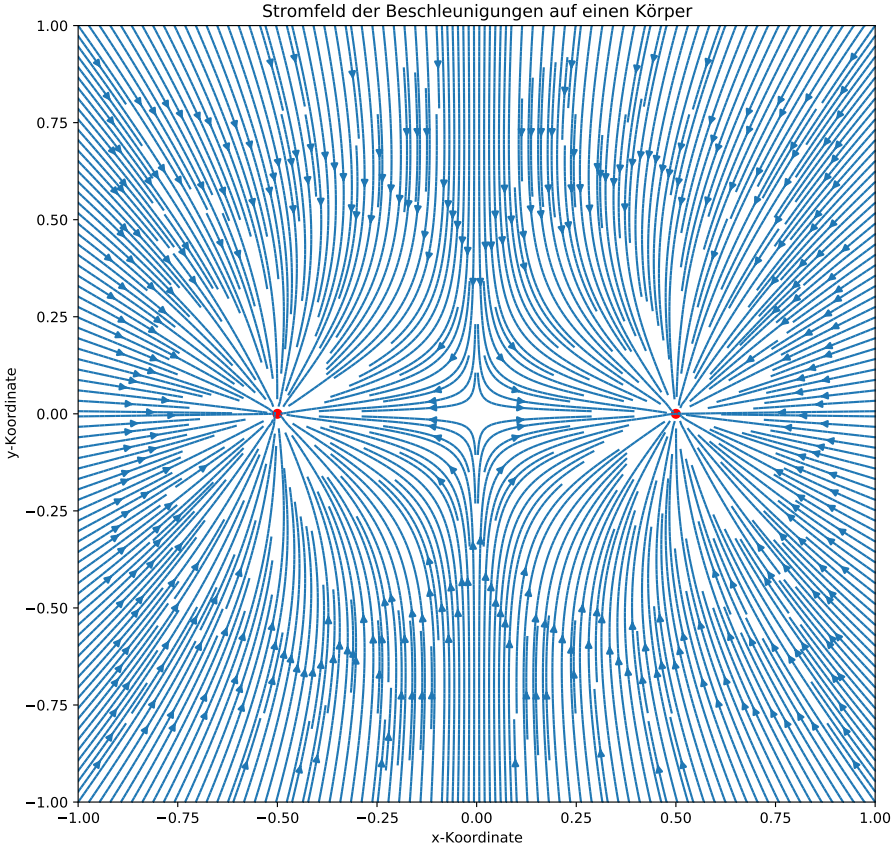


Abbildung 1: Darstellung der Beschleunigung, welche durch die Gravitationskraft auf m_3 ($\gamma = 1$) wirkt. Dabei gilt $m_1 = m_2 = m_3 = 1$ und m_1, m_2 auf $(-0.5|0)$ und $(0.5|0)$ fixiert.

3.2 Hamiltonsche Mechanik

Die Hamilton-Mechanik bietet mit der Differenzierung der Hamilton-Funktion ein weiteres Verfahren, um die Bewegungsgleichungen eines Systems herzuleiten [4]. Die Formulierung in der Hamilton-Mechanik ist auch von Nutzen für spätere Messungen der Integratoren. Die Hamilton-Funktion $H(q,p)$ beschreibt die Summe der potentiellen und kinetischen Energie eines Systems, wobei q die verallgemeinerten Ortskoordinaten und p die verallgemeinerten Impulskoordinaten sind [10]. Für verallgemeinerte Koordinaten gilt, dass diese zwangsläufig voneinander unabhängig und essentiell für die Beschreibung eines Systems sind [17]. Da das System des Dreikörperproblems über keine Einschränkungen (Zwangsbedingungen) verfügt, entsprechen die verallgemeinerten Ortskoordinaten den Freiheitsgraden[3] des Systems [18]. Jeder der drei Körper kann sich unabhängig in alle drei Raumrichtungen bewegen, somit werden $3 \cdot 3 = 9$ verallgemeinerte Ortskoordinaten für das Beschreiben einer Phase des dreidimensionalen Dreikörperproblems benötigt [16]. Diese verallgemeinerten Ortskoordinaten werden folgend mit q_{ik} gekennzeichnet, wobei i zur Identifikation des Körpers und k zur Identifikation der Koordinate des Körpers dient. Das gleiche Vorgehen kann für die verallgemeinerten Impulskoordinaten p_{ik} angewendet werden. Jeder Körper verfügt über einen unabhängigen Impuls in alle drei Raumrichtungen $\Rightarrow$ 9 verallgemeinerte Impulskoordinaten [17]. Die bereits beschriebenen Hamiltonfunktion lässt sich somit als $H(q,p) = E_{kin} + E_{pot}$ darstellen. Die kinetische Energie des gesamten Systems lässt sich über $E_{i,kin} = \frac{1}{2} m_i \cdot \dot{r}_i^2 = \frac{1}{2}\frac{p_i^2}{m_i}$ mit verallgemeinerten Impulskoordinaten als

$$E_{kin} = \sum_{i=1}^{3} \sum_{k=1}^{3} \frac{p_{ik}^2}{2m_i} \tag{3}$$

formulieren [14]. Die kinetische Energie in jeder der drei Raumrichtungen ($k = 1,2,3$) jedes Körpers ($i = 1,2,3$) ist für die gesamte kinetische Energie zu summieren. Die potentielle Energie folgt hierbei nach Energie = Kraft $\cdot$ Weg, aus der resultierenden Kraft (Gl. 1 ohne die Richtung $\frac{(\vec{r}_i - \vec{r}_j)}{|\vec{r}_i - \vec{r}_j|}$) und der Distanz zwischen zwei Körpern $|q_{jk} - q_{ik}|$, wobei jedoch fraglich bleibt, inwiefern sich die potentielle Energie in eine unabhängige Raumrichtung k vorzustellen ist:

$$E_{pot} = -\gamma \sum_{i=1}^{3} \sum_{j=i+1}^{3} \sum_{k=1}^{3} \frac{m_i \cdot m_j}{|q_{jk} - q_{ik}|^2} \cdot |q_{jk} - q_{ik}| \tag{4}$$

Hierbei erfolgt eine Summierung über alle Körper i in Kombination mit Körper j und in alle Raumrichtungen k [10]. Der Startwert der zweiten Summe ($j = i+1$), verhindert dabei, dass die potentielle Energie zwischen zwei Körpern zweimal in die gesamte potentielle Energie des Systems addiert wird. Das negative Vorzeichen folgt hierbei daraus, dass Arbeit verrichtet werden muss, um die beiden Körper weiter auseinander zu bringen und somit die potentielle Energie, bei größerer Entfernung, größer wird [5]. Da die Entfernung im Nenner steht, kann der geschilderte Zusammenhang über ein negatives Vorzeichen erreicht werden. Für die Hamiltonfunktion

[3] Bewegungsmöglichkeiten [18]

folgt somit:

$$H(q,p) = \sum_{i=1}^{3}\sum_{k=1}^{3}\frac{p_{ik}^2}{2m_i} - \gamma\sum_{i=1}^{3}\sum_{j=i+1}^{3}\sum_{k=1}^{3}\frac{m_i \cdot m_j}{|q_{ik} - q_{jk}|} \tag{5}$$

Um nun die hamiltonschen Bewegungsgleichungen herzuleiten, lässt sich die Hamiltonfunktion nach p_{ki} und nach q_{ki} partiell differenzieren [14].

$$\frac{\partial H(p,q)}{\partial p_{ik}} = \frac{p_{ik}}{m_i} = \frac{m_i \cdot \dot{q}_{ik}}{m_i} = \dot{q}_{ik} \qquad \frac{\partial H(p,q)}{\partial q_{ik}} = -\dot{p}_{ik} = F_{ik} \tag{6}$$

Dadurch, dass das System über jeweils 9 verschiedene q_{ik} und p_{ik} verfügt, ergeben sich 18 DGLs erster Ordnung [19]. Die Formulierung in der Hamilton-Mechanik und die durchgeführte Herleitung bildet die Grundlage einer Argumentation für die allgemeine Unlösbarkeit des Problems[2]. Die Existenz von maximal 12 Bewegungsintegralen, woraus die Reduktion auf 6 DGLs resultiert, welche nicht allgemein analytisch zu lösen sind, kann nachgewiesen [14], soll jedoch hierbei nicht weiter konkretisiert werden. Durch diese allgemeine Unlösbarkeit des Dreikörperproblems, ist es notwendig auf numerische Verfahren (Abschnitt 4) zurückzugreifen (es gibt einige analytische Lösungen für eingschränkte Fälle [16]). Für verschiedene spätere Messungen der Simulationen (z.B. Unterabschnitt 5.6), wird eine quantitative Größe für den Unterschied zwischen zwei Systemen A und B benötigt. Dafür lässt sich ein Zusammenhang der bereits herausgestellten generalisierten Koordinaten formulieren [9]:

$$\delta_{A,B} = \frac{1}{18}\left(\sum_{i=1}^{3}\sum_{k=1}^{3}|q_{A,i,k} - q_{B,i,k}| + \sum_{i=1}^{3}\sum_{k=1}^{3}|p_{A,i,k} - p_{B,i,k}|\right) \tag{7}$$

Es werden die Differenzen zwischen den Phasen A und B, in den generalisierten Orts- und Impulskoordinaten jedes Körpers summiert. Nach [9] stellt der Unterschied zwischen generalisierten Orts- und Impulskoordinaten dabei kein Problem dar. Dazu wird durch die Anzahl der summierten Koordinaten geteilt und somit der arithmetische Mittelwert aller Unterschiede in den generalisierten Orts- und Impulskoordinaten der beiden Systeme berechnet [9]. Da das Problem in den Simulationen auf den zweidimensionalen Raum reduziert wird, folgt $k \in \{1,2\}$, weswegen für das zweidimensionale Dreikörperproblem nur durch 12 dividiert werden muss.

4 Numerische Simulationen

Es stellt sich heraus, dass das Dreikörperproblem ein Anfangswertproblem (AWP) ist [1]. Ein AWP besteht darin, dass eine DGL der Form $y' = f(t,y)$ und ein Anfangswertepaar (t_0, y_0) gegeben ist, während jedoch y unbekannt ist [1]. Das Problem besteht darin, eine Annäherung für einen bestimmten Funktionswert $y(t_{n-1})$ bzw. Funktionswerte der Funktion $y(t_i)$ zu generieren [6]. Verschiedene numerische Verfahren ermöglichen solch eine Annäherung von y, generiert über ein bestimmtes Intervall $[t_0, t_{n-1}]$ [6]. Dabei wird mit einer gewählten Anzahl an Schritten n und der Schrittgröße $h = \frac{t_{n-1}-t_0}{n}$, bei y_0 gestartet und mit $t_{i+1} = t_i + h$, eine Annäherung für $y_1, y_2, ..., y_{n-1}$ getroffen. Hierbei gilt jedoch lediglich $y_i \approx y(t_i)$, was auf einige Fehlerquel-

len zurückzuführen ist (Unterabschnitt 4.1). Eine Unterscheidung zwischen den numerischen Verfahren findet sich in den Feinheiten der Berechnung der einzelnen y_i wieder. Die in dieser Facharbeit thematisierten Verfahren, werden parallel in einem *Python-Jupyter-Notebook* implementiert. Zunächst finden sie Anwendung auf eine analytisch lösbare Differentialgleichung [27]. Des Weiteren, werden sie für Integration des Dreikörperproblems eingesetzt [28]. Im Folgenden soll jedoch lediglich auf den mathematischen Hintergrund eingegangen werden, während nähere Details zur Implementierung in den Notebooks [27, 28] zu finden sind. Um die Visualisierung zu vereinfachen und den Rechenaufwand zu minimieren, wird das Dreikörperproblem lediglich in 2 Dimensionen simuliert. Für die Anwendung der numerischen Verfahren und für das Dreikörperproblem macht dies keinen Unterschied [9, 20].

4.1 Fehlerquellen

Eine Ungenauigkeit bei numerischen Berechnungen mit dem Computer, entsteht durch die endliche Darstellung und Präzision von Zahlen [6]. Durch die Limitierung an Nachkommastellen, kommt es zu Situationen in denen die Genauigkeit, speziell bei sehr kleinen Zahlen, durch den Rundungsfehler problematisch wird [9]. Um diesen Fehler zu reduzieren, eignen sich Bibliotheken für arithmetische Präzision (z.B. [29]), wodurch die Menge an Speicherplatz (standardmäßig: $\psi = 15\text{Bits}$ [29]) bzw. die Anzahl an Nachkommastellen von Gleitkommazahlen kontrolliert werden kann. Zudem kann der Rundungsfehler durch das Reduzieren von Berechnungen verringert werden [4], dieser ist somit auch vom numerischen Verfahren abhängig.
Eine weitere Fehlerquelle bei der numerischen Integration ist der Diskretisierungsfehler, welcher dadurch entsteht, dass die Zeitschritte h, mit welchen angenähert wird, nicht infinitesimal klein werden können[6]. Um eine Ordnung $\mathcal{O}(...)$ und damit eine Abschätzung dieses Fehlers zu konkretisieren, kann das jeweilige numerische Verfahren mit der unendlichen *Taylorreihe*

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x) + ... \tag{8}$$

in Zusammenhang gebracht werden [1]. Die *Taylorreihe* ermöglicht es, unendlich differenzierbare Funktionen mit einem Polynom zu darzustellen, indem jeder Summand $\frac{h^n}{n!}f^{(n)}(x)$ den Funktionswert der n-ten Ableitung der approximierten Funktion kontrolliert [7]. Je mehr Terme dabei hinzugefügt werden ($\sum_{k=0}^{n} \frac{h^k}{k!}f^{(k)}(x)$ für $n \to \infty$), desto genauer kann $f(x+h)$ angenähert werden [6]. Die Konvergenz, Funktionalität und Anwendbarkeit dieser unendlichen Reihe in Bezug auf das Dreikörperproblem, soll hierbei nicht weiter thematisiert werden [4]. Die *Taylorreihe* stellt hierbei lediglich ein Werkzeug für das Abschätzen des Diskretisierungsfehlers dar. Zudem ergeben sich aus einer bestimmten Anzahl von Termen der Taylor-Reihe bereits viele verschiedene numerische Verfahren, der sogenannte Runge-Kutta-Familie [10]. Für den Diskretisierungsfehler, lässt sich zwischen einem lokalen Fehler E ($\Rightarrow$ Fehler bei einem Schritt der Größe h) und einem globalen Fehler E_{ges} ($\Rightarrow$ Angesammelter Fehler nach n-Schritten) unterscheiden [1, 21]. Man bestimme die Ordnung von E und E_{ges} für die folgenden Verfahren mithilfe der *Taylorreihe*.

4.2 Euler-Verfahren

Das Vorgehen beim *Euler-Verfahren* basiert auf den ersten Termen der *Taylorreihe* (Gl. 8), welche sich auch als Differenzenquotient schreiben lassen ($f'(x) \approx \frac{f(x+h)-f(x)}{h}$) [3]. Auch grafisch, als einfaches Steigungsdreieck, lässt sich das *Euler-Verfahren* ableiten (Abb. 2) und ist somit als $y(t+h) = y(t) + hy'(t)$ zu formulieren, wobei h einer gewählten Schrittgröße entspricht.

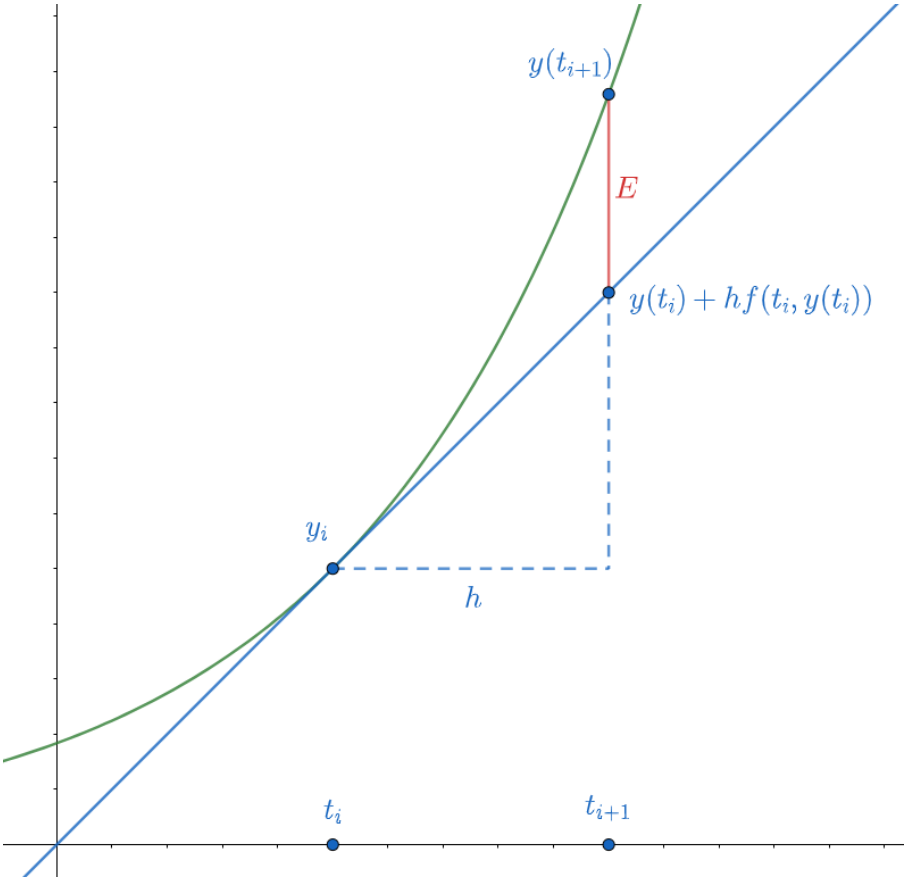


Abbildung 2: Euler-Verfahren

Aus einem AWP: $y' = f(t, y)$ und $y_0 = y(t_0)$, lassen sich somit über $y_{i+1} = y_i + hf(t_i, y_i)$, verschiedene Punkte (t_i, y_i) von $y(t)$ mit dem Abstand h approximieren [1, 21]. Für den Sachzusammenhang werden im Folgenden die Bewegungsgrößen der drei Körper zusammengefasst definiert. Für das numerische Verfahren macht dies keinen Unterschied, da das Vorgehen für jeden der drei Körper gleich ist und somit lediglich in jeder Zeitinstanz t_i dreimal wiederholt werden muss, dabei indiziert i den Zeitschritt. Bei jeder Wiederholung der numerischen Integration i $(0 \leq i < n)$, wird die DGL (Gl. 2) in zwei verschiedene Approximationvorgänge zerlegt und entspricht somit der Form des AWP

$$\begin{aligned} \vec{r}_{i+1} &= \vec{r}_i + h\dot{\vec{r}}_i \\ \dot{\vec{r}}_{i+1} &= \dot{\vec{r}}_i + h\ddot{\vec{r}}(\vec{r}_i) \end{aligned} \tag{9}$$

numerisch gelöst [21]. Das *Euler-Verfahren* wird jeweils auf die Geschwindigkeit und Position gleichzeitig angewendet. Der Diskretisierungsfehler des *Euler-Verfahrens* lässt sich als Trunkie-

rung[4] der unendlichen *Taylorreihe* (Gl. 8) darstellen [7]:

$$f(x+h) = f(x) + hf'(x) + \mathcal{O}(h^2) \tag{10}$$

Dadurch, dass h sehr klein wird, ist der größte abgeschnittene, von h abhängige Summand in der *Taylorreihe*, der abgeschnittene Summand mit dem kleinsten Exponenten: $\frac{h^2}{2!}f''(x)$. Da es sich bei der Fehlerabschätzung lediglich um eine asymptotische Ordnung handelt, werden die restlichen kleineren Terme und konstante Faktoren ignoriert [10]. Der lokale Fehler E entspricht somit der Ordnung $\mathcal{O}(h^2)$ [1]. Bei der Annahme, dass E über die gesamte Integration konstant ist, lässt sich eine Ordnung des gesamten Fehlers nach n Integrationsschritten feststellen (E_{ges}):

$$E_{ges} = \frac{t_{n-1} - t_0}{h} \cdot E = (t_{n-1} - t_0) \cdot \frac{\mathcal{O}(h^2)}{h} = \mathcal{O}(h) \tag{11}$$

Dadurch, dass sich $n = \frac{t_{n-1}-t_0}{h}$ mal der lokale Fehler ergibt, folgt die Ordnung $\mathcal{O}(h)$, wobei der Konstante Faktor $t_{n-1} - t_0$ vernachlässigt wird [1]. Das Konzept des Algorithmus, kann durch eine Formulierung in „Pseudocode“ konkretisiert werden (hierbei sind die Größen nach den Körpern indiziert; z.B. $\vec{r}_1$: Position von Körper 1):

Algorithmus 1 Euler-Verfahren

$pos \leftarrow \{\{\vec{r}_1\}, \{\vec{r}_2\}, \{\vec{r}_3\}\}$ ▷ zweidimensionale Liste an Positionen
$vel \leftarrow \{\{\dot{\vec{r}}_1\}, \{\dot{\vec{r}}_2\}, \{\dot{\vec{r}}_3\}\}$ ▷ zweidimensionale Liste an Geschwindigkeiten
for Zeitschritt $0 \leq i < n$ **do** ▷ Iteration über die Zeitschritte
  for Körper $0 \leq j < 3$ **do** ▷ Iteration über die Körper (0-basiert)
    $pos[j] \overset{+}{\leftarrow} pos[j][i] + vel[j][i] \cdot h$ ▷ Berechnen der Position bei Zeitschritt $i+1$
    $vel[j] \overset{+}{\leftarrow} vel[j][i] + \ddot{\vec{r}}_j(pos[:][i]) \cdot h$ ▷ Berechnen der Geschw. bei Zeitschritt $i+1$
  end for
end for

[4] Verkürzung

4.3 Euler-Cromer-Verfahren

Im Gegensatz zum *Euler-Verfahren* wird im *Euler-Cromer-Verfahren*, zuerst die Geschwindigkeit $\dot{\vec{r}}_{i+1}$ und darauffolgend die Position r_{i+1} der Körper berechnet, wobei für die Berechnung der Position, bereits die neu berechnete Geschwindigkeit $\dot{\vec{r}}_{i+1}$ verwendet wird [10]:

$$\begin{aligned} \dot{\vec{r}}_{i+1} &= \dot{\vec{r}}_i + h\ddot{\vec{r}}(\vec{r}_i) \\ \vec{r}_{i+1} &= \vec{r}_i + h\dot{\vec{r}}_{i+1} \end{aligned} \tag{12}$$

Auch dieses Verfahren lässt sich visualisieren (Abb. 3):

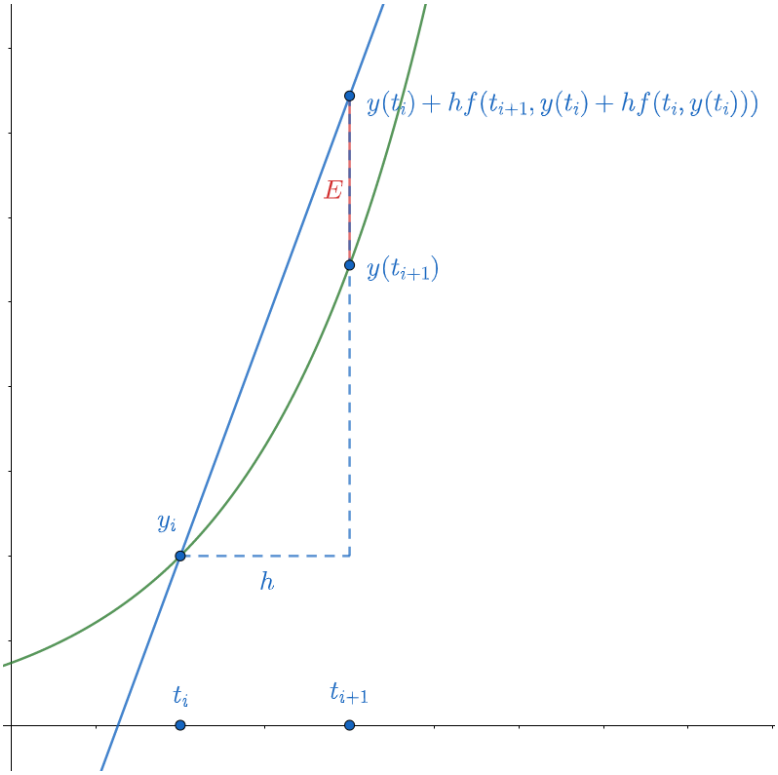


Abbildung 3: Euler-Cromer-Verfahren

Die Steigung des nächsten Zeitschrittes (hier $\dot{\vec{r}}_{i+1}$) wird zur Berechnung des Funktionswertes des nächsten Zeitschrittes (hier $\vec{r}_{i+1}$) verwendet [10]. Der Algorithmus, kann wiederum durch eine Formulierung in „Pseudocode“ verdeutlicht werden:

Algorithmus 2 Euler-Cromer-Verfahren

$pos \leftarrow \{\{\vec{r}_1\}, \{\vec{r}_2\}, \{\vec{r}_3\}\}$
$vel \leftarrow \{\{\dot{\vec{r}}_1\}, \{\dot{\vec{r}}_2\}, \{\dot{\vec{r}}_3\}\}$
for Zeitschritt $0 \leq i < n$ **do**
  for Körper $0 \leq j < 3$ **do**
    $vel[j] \xleftarrow{+} vel[j][i] + \ddot{\vec{r}}_j(pos[:][i]) \cdot h$
    $pos[j] \xleftarrow{+} pos[j][i] + vel[j][i+1] \cdot h$
  end for
end for

Aus der Kombination von *Euler-Cromer* und *Euler*, resultiert das *Heun-Verfahren* [10]. Hierbei wird für die Berechnung der Position des nächsten Zeitschritte $\vec{r}_{i+1}$, der arithmetische Mittelwert der Geschwindigkeiten zum Zeitpunkt i und $i+1$ berechnet [10]:

$$\begin{aligned} \dot{\vec{r}}_{i+1} &= \dot{\vec{r}}_i + h\ddot{\vec{r}}(\vec{r}_i) \\ \vec{r}_{i+1} &= \vec{r}_i + \frac{h}{2}(\dot{\vec{r}}_i + \dot{\vec{r}}_{i+1}) \end{aligned} \tag{13}$$

Algorithmus 3 Heun-Verfahren

$pos \leftarrow \{\{\vec{r}_1\},\{\vec{r}_2\},\{\vec{r}_3\}\}$
$vel \leftarrow \{\{\dot{\vec{r}}_1\},\{\dot{\vec{r}}_2\},\{\dot{\vec{r}}_3\}\}$
for Zeitschritt $0 \leq i < n$ **do**
  for Körper $0 \leq j < 3$ **do**
    $vel[j] \overset{+}{\leftarrow} vel[j][i] + \ddot{\vec{r}}_j(pos[:][i]) \cdot h$
    $pos[j] \overset{+}{\leftarrow} pos[j][i] + (vel[j][i] + vel[j][i+1]) \cdot \frac{h}{2}$
  end for
end for

4.4 Runge-Kutta-Verfahren 4er Ordnung

Die Idee von einem Mittelwert mehrerer Steigungen, lässt sich auf noch mehr Terme erweitern. So gibt es beispielsweise das *Runge-Kutta-Verfahren 4er Ordnung* (*RK4*), in welchem diese Idee umgesetzt wird [1]. Das *RK4*, wird in mehreren Termen und einem gewichteten Mittelwert definiert [22]:

$$\begin{aligned} K_1 &= hf(t,y) \\ K_2 &= hf(t+\frac{h}{2}, y+\frac{K_1}{2}) \\ K_3 &= hf(t+\frac{h}{2}, y+\frac{K_2}{2}) \\ K_4 &= hf(t+h, y+K_3) \\ y(t+h) &= y(t) + \frac{1}{6}(K_1 + 2K_2 + 2K_3 + K_4) \end{aligned} \tag{14}$$

Es kann nachgewiesen werden, dass das *RK4*, die ersten 5 Terme der *Taylorreihe* einbindet und somit die Ordnung des lokalen Fehlers für dieses Verfahren bei $E = \mathcal{O}(h^5)$ liegt [8]. Der Gesamtfehler leitet sich über

$$E_{ges} = n \cdot E = \frac{t_{n-1} - t_0}{h} \cdot E = (t_{n-1} - t_0) \cdot \frac{\mathcal{O}(h^5)}{h} = \mathcal{O}(h^4) \tag{15}$$

her. Wie der Name schon sagt, hat das genannte Verfahren, einen Fehler vierter Ordnung [8]. Der Bezug auf das Dreikörperproblem, wird in [22] herausgestellt, wobei die Terme, wie auch schon bei den vorherigen Verfahren, jeweils auf die Änderung der Geschwindigkeit und die

Änderung der Position angewendet werden. Insgesamt ist *RK4* deutlich rechenintensiver, was womöglich Einfluss auf den Rundungsfehler nimmt [14]. Für den Algorithmus folgt:

Algorithmus 4 RK4-Verfahren

$pos \leftarrow \{\{\vec{r}_1\}, \{\vec{r}_2\}, \{\vec{r}_3\}\}$
$vel \leftarrow \{\{\dot{\vec{r}}_1\}, \{\dot{\vec{r}}_2\}, \{\dot{\vec{r}}_3\}\}$
for Zeitschritt $0 \leq i < n$ **do**
  for Körper $0 \leq j < 3$ **do**
    $K_{v1} \overset{+}{\leftarrow} h \cdot \ddot{\vec{r}}_j(pos[:][i])$
    $K_{r1} \overset{+}{\leftarrow} h \cdot vel[j][i]$
  end for
  for Körper $0 \leq j < 3$ **do**
    $K_{v2} \overset{+}{\leftarrow} h \cdot \ddot{\vec{r}}_j(pos[:][i] + 0.5K_{r1}[:])$
    $K_{r2} \overset{+}{\leftarrow} h \cdot (vel[j][i] + 0.5K_{v1}[j])$
  end for
  for Körper $0 \leq j < 3$ **do**
    $K_{v3} \overset{+}{\leftarrow} h \cdot \ddot{\vec{r}}_j(pos[:][i] + 0.5K_{r2}[:])$
    $K_{r3} \overset{+}{\leftarrow} h \cdot (vel[j][i] + 0.5K_{v2}[j])$
  end for
  for Körper $0 \leq j < 3$ **do**
    $K_{v4} \overset{+}{\leftarrow} h \cdot \ddot{\vec{r}}_j(pos[:][i] + K_{r3}[:])$
    $K_{r4} \overset{+}{\leftarrow} h \cdot (vel[j][i] + K_{v3}[j])$
  end for
  for Körper $0 \leq j < 3$ **do**
    $vel[j] \overset{+}{\leftarrow} vel[j][i] + \frac{1}{6}(K_{v1} + 2K_{v2} + 2K_{v3} + K_{v4})$
    $pos[j] \overset{+}{\leftarrow} pos[j][i] + \frac{1}{6}(K_{r1} + 2K_{r2} + 2K_{r3} + K_{r4})$
  end for
end for

4.5 Verlet-Leapfrog-Verfahren

Die *Verlet-Leapfrog-Verfahren* (*VLV*) eignen sich für DGLs der Form $y'' = f(t, y)$ und können somit auf das Dreikörperproblem gemäß Gl. 2 angewendet werden [14]. Die Idee ist, die Positionen des nächsten Zeitschrittes $(i+1)$, mit den Geschwindigkeiten bei $i+\frac{1}{2}$ anstatt bei i oder $i+1$ zu berechnen [10]. Dabei wird jedoch nicht mehr der Mittelwert der Geschwindigkeiten bei den Intervallgrenzen $\dot{\vec{r}}_i$ und $\dot{\vec{r}}_{i+1}$ zur Berechnung der nächsten Position $\vec{r}_{i+1}$ benutzt, sondern versetzt über eine halbe Schrittgröße, Geschwindigkeit $\dot{\vec{r}}$ und Position $\vec{r}$ berechnet [21].

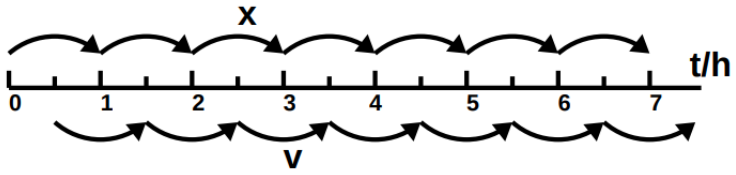


Abbildung 4: (Aus [21]) Leapfrog-Prinzip ($x := \vec{r}$ und $v := \dot{\vec{r}}_i$)

Dieser Versatz wird über den gesamten Integrationsprozess beibehalten.

$$\begin{aligned} \vec{r}_{i+1} &= \vec{r}_i + h\dot{\vec{r}}_{i+\frac{1}{2}} \\ \dot{\vec{r}}_{i+\frac{3}{2}} &= \dot{\vec{r}}_{i+\frac{1}{2}} + h\ddot{\vec{r}}(\vec{r}_{i+1}) \end{aligned} \tag{16}$$

Da für das Dreikörperproblem AWP, die Geschwindigkeit und Position jedes Körpers für $i = 0$ gegeben sind, muss zunächst der Versatz von $\frac{1}{2}$ eingefügt werden [21]. Dafür kann ein Schritt gemäß des *Euler-Verfahrens* genutzt werden: $\dot{\vec{r}}_{\frac{1}{2}} = \dot{\vec{r}}_0 + \frac{h}{2}\ddot{\vec{r}}(\vec{r}_0)$ [21]. Die algorithmische Umsetzung folgt:

Algorithmus 5 Verlet-Leapfrog-Verfahren

$pos \leftarrow \{\{\vec{r}_1\}, \{\vec{r}_2\}, \{\vec{r}_3\}\}$
$vel \leftarrow \{\{\dot{\vec{r}}_1\}, \{\dot{\vec{r}}_2\}, \{\dot{\vec{r}}_3\}\}$
for Körper $0 \leq j \leq 3$ **do**
  $vel[j] \overset{+}{\leftarrow} vel[j][0] + \ddot{\vec{r}}_j(pos[:][0]) \cdot \frac{1}{2}$
end for
for Zeitschritt $0 \leq i < n-1$ **do**
  for Körper $0 \leq j \leq 3$ **do**
    $pos[j] \overset{+}{\leftarrow} pos[j][i] + vel[j][i] \cdot h$
  end for
  for Körper $0 \leq j < 3$ **do**
    $vel[j] \overset{+}{\leftarrow} vel[j][i] + \ddot{\vec{r}}_j(pos[:][i+1]) \cdot h$
  end for
end for

Für spätere Messungen des Impulses und Berechnungen der Energie, ist es wichtig, über die Geschwindigkeit bei Zeitschritt i zu verfügen. Deswegen lässt sich eine Abwandlung von Gl. 16

treffen, in der zwei Halbschritte durchgeführt werden ($\dot{\vec{r}}_i \rightarrow \dot{\vec{r}}_{i+\frac{1}{2}} \rightarrow \dot{\vec{r}}_{i+1}$) [21]:

$$\begin{aligned} \dot{\vec{r}}_i &= \dot{\vec{r}}_{i-\frac{1}{2}} + \frac{h}{2}\ddot{\vec{r}}(\vec{r}_i) \\ \dot{\vec{r}}_{i+\frac{1}{2}} &= \dot{\vec{r}}_i + \frac{h}{2}\ddot{\vec{r}}(\vec{r}_i) \\ \vec{r}_{i+1} &= \vec{r}_i + h\dot{\vec{r}}_{i+\frac{1}{2}} \end{aligned} \tag{17}$$

Die Berechnung der Beschleunigung $\ddot{\vec{r}}$ muss auch hierbei nur einmal erfolgen, da in beiden Berechnungen der gleiche Wert $\ddot{\vec{r}}(\vec{r}_i)$ genutzt wird [21]. Somit wird keine weitere Berechnung, welche möglicherweise den Rundungsfehler vergrößert, benötigt. Bei diesem Algorithmus (Gl. 17) lassen sich die Geschwindigkeit $\dot{\vec{r}}_i$ und die Position $\vec{r}_i$, bei Anpassung der entsprechenden Änderungsrate, auch vertauschen. Dabei sind für die Positionen jeweils zwei Halbschritte $i \rightarrow i+\frac{1}{2} \rightarrow i+1$ und für die Geschwindigkeiten jeweils ein Ganzschritt $i \rightarrow i+1$ durchzuführen [21]. Es lässt sich zwischen dem *Geschwindigkeit-Verlet-* und dem *Position-Verlet-Verfahren* unterscheiden. In der Original-Publikation von *Verlet 1967* [23], wird ein Vorgehen des Types:

$$\vec{r}_{i+1} = -\vec{r}_{i-1} + 2\vec{r}_i + h^2\ddot{\vec{r}}(\vec{r}_i) \tag{18}$$

beschrieben. Es stellt sich heraus, das dieses Vorgehen aus Gl. 16 hervorgeht und somit äquivalent ist [21]:

$$\begin{aligned} &\vec{r}_{i+1} = \vec{r}_i + h\dot{\vec{r}}_{i+\frac{1}{2}} \quad | \quad \dot{\vec{r}}_{i+\frac{1}{2}} = \dot{\vec{r}}_{i-\frac{1}{2}} + h\ddot{\vec{r}}(\vec{r}_i) \\ &\vec{r}_{i+1} = \vec{r}_i + h(\dot{\vec{r}}_{i-\frac{1}{2}} + h\ddot{\vec{r}}(\vec{r}_i)) \quad | \quad h\dot{\vec{r}}_{i-\frac{1}{2}} = \vec{r}_i - \vec{r}_{i-1} \text{ (Gl. 16 umgestellt)} \\ &\vec{r}_{i+1} = 2\vec{r}_i - \vec{r}_{i-1} + h^2\ddot{\vec{r}}(\vec{r}_i) \end{aligned} \tag{19}$$

Bei diesem Zusammenhang wird jedoch nicht mehr die Geschwindigkeit berechnet, welche für verschiedene Messungen von Bedeutung ist. In der Originalpublikation, berechnete *Verlet* die Geschwindigkeit mit der Änderung der Position über zwei Schrittlängen ($\dot{\vec{r}}_i = \frac{\vec{r}_{i+1}-\vec{r}_{i-1}}{2h}$) (*Störmer-Verfahren*) [23], was sich im Vergleich zu anderen Verfahren jedoch als unpräzise herausstellt (Abschnitt 5). Der Fehler gemäß der *Taylorreihe* (Gl. 8), folgt aus der herausgestellten Berechnung für die Geschwindigkeit über zwei Schrittlängen:

$$\begin{aligned} \dot{\vec{r}}_i &= \frac{\vec{r}_{i+1} - \vec{r}_{i-1}}{2h} \quad | \quad \cdot(-2h) \\ -2h \cdot \dot{\vec{r}}_i &= -\vec{r}_{i+1} + \vec{r}_{i-1} \quad | \quad +\vec{r}_{i+1} \\ \vec{r}_{i+1} - 2h \cdot \dot{\vec{r}}_i &= \vec{r}_{i-1} \end{aligned} \tag{20}$$

Nach dem Einsetzen in Gl. 19, lässt sich eine Trunkierung der *Taylorreihe* (Gl. 8) feststellen:

$$\begin{aligned} \vec{r}_{i+1} &= 2\vec{r}_i - (\vec{r}_{i+1} - 2h \cdot \dot{\vec{r}}_i) + h^2 \ddot{\vec{r}}(\vec{r}_i) \quad | \quad + \vec{r}_{i+1} \\ 2\vec{r}_{i+1} &= 2\vec{r}_i + 2h \cdot \dot{\vec{r}}_i + h^2 \ddot{\vec{r}}(\vec{r}_i) \quad | \quad \cdot \frac{1}{2} \\ \vec{r}_{i+1} &= \vec{r}_i + h \cdot \dot{\vec{r}}_i + \frac{h^2 \ddot{\vec{r}}(\vec{r}_i)}{2} \end{aligned} \tag{21}$$

Der größte fehlende Summand der *Taylorreihe*, ist $\frac{h^3}{3!} f'''(x)$, weswegen sich auf einen lokalen Fehler der Ordnung $E = \mathcal{O}(h^3)$ schließen lässt [8]. Der erste Schritt mit dem *Euler-Verfahren*, sorgt für einen Fehler der Ordnung $\mathcal{O}(h^2)$, da dieser Fehler jedoch nur einmal auftritt, ist jener als konstanter Faktor des Gesamtfehlers zu betrachten, und kann vernachlässtigt werden [21]. Der Gesamtfehler leitet sich ähnlich wie bei vorherigen Verfahren über

$$E_{ges} = n \cdot E = \frac{t_{n-1} - t_0}{h} \cdot E = (t_{n-1} - t_0) \cdot \frac{\mathcal{O}(h^3)}{h} = \mathcal{O}(h^2) \tag{22}$$

her. Die *VLV*, sind somit numerische Verfahren zweiter Ordnung.

4.6 Bulirsch-Stoer-Verfahren

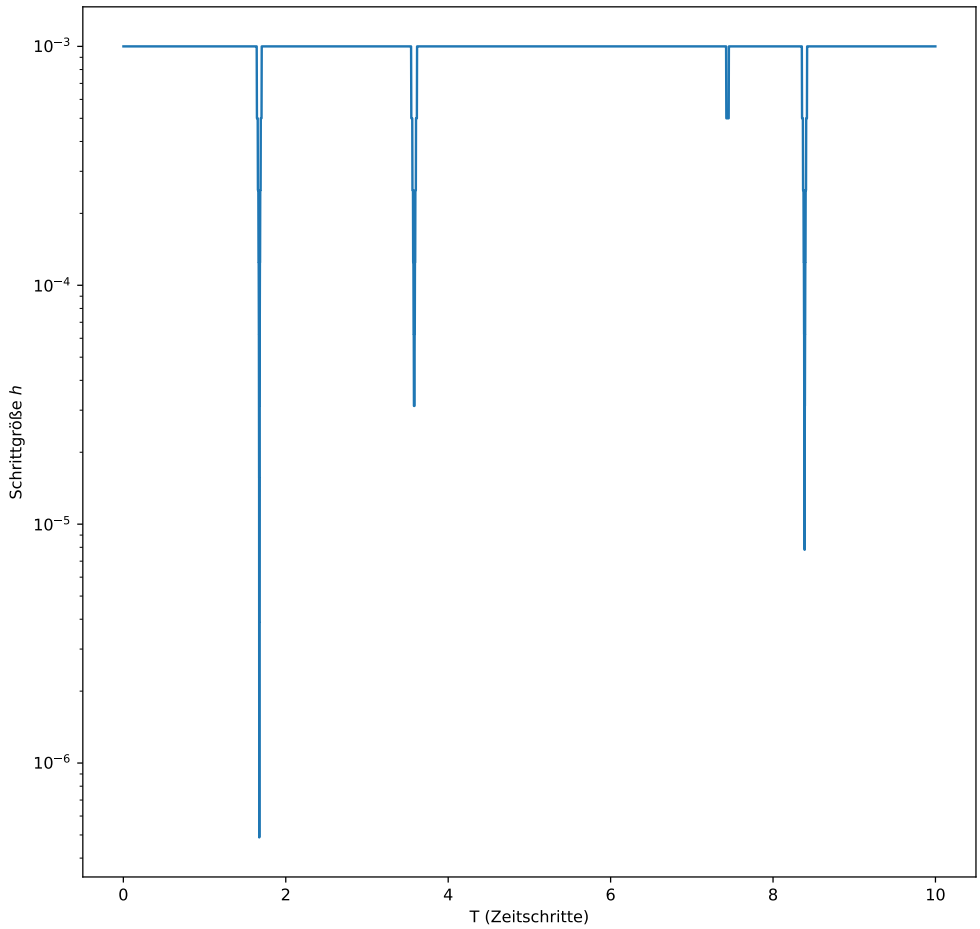
Das *Bulirsch-Stoer-Verfahren* (*BSV*) ist ein Optimierungsverfahren mit adaptiver Schrittgröße, welches ermöglicht den Fehler systematisch unter einer gewissen Toleranz ϵ zu halten [7]. Dabei wird ein Integrationsverfahren in Kombination mit einem Extrapolationsverfahren benutzt [9]. Extrapolation beschreibt den Prozess, vom Berechnen einer Funktion, welche durch gegebene Wertepaare verläuft, mit dem Ziel einen unbekannten Funktionswert, außerhalb der Reichweite der bekannten Wertepaare, herauszufinden [7]. Im Rahmen des Extrapolationsverfahrens, lässt sich eine Fehlerabschätzung *err* für den extrapolierten Wert treffen [1, 7]. Ein Integrationsschritt mit dem BSV von $t_i \to t_{i+1}$, besteht darin, dass das Integrationsintervall $[t_i, t_i + h]$, in n kleinere Schritte eingeteilt wird [1]. Das Intervall $[t_i, t_i + h]$ wird nun mit einem bekannten Integrationsverfahren (z.B. *VLV*) integriert [1]. Die Schrittgröße für das gewählte Integrationsverfahren wird dabei durch $\eta = \frac{h}{n}$ bestimmt [7].

Nachdem das Intervall mit $n = 2$ und $n = 4$ mit einem bekannten Verfahren integriert wurde, wird eine Extrapolation durchgeführt, welche die extrapolierten Werte und eine Fehlerabschätzung *err* ermöglicht (Präzisierung folgt) [7]. Liegt dieser Fehler unter einer bestimmten Toleranz ($err \leq \epsilon$), wird dieser Integrationsschritt ($t_i \to t_{i+1}$) akzeptiert und der nächste Schritt folgt [1]. Sollte $err > \epsilon$, folgt eine Vergrößerung von n, bzw. eine Verkleinerung von η [7]. Nach jedem Wert für n, wird eine Extrapolation durchgeführt, und die bereits herausgestellte Toleranzüberprüfung findet statt. Die Schrittanzahl n kann dabei nach der Reihe: $n = 2, 4, 6, 8, 10, 12, 14, 16, ...$ vergrößert werden [8]. Insgesamt, wird jeder Integrationsschritt des *BSV*, somit mit n Schritten in einem anderen Verfahren integriert, wobei n solange erhöht wird, bis eine gewünschte Toleranz ϵ erreicht ist.

Es ist jedoch wichtig, dass es einen Maximalwert für n gibt, falls der Fehler beispielsweise

einer zu geringen Genauigkeit beim Berechnen (Rundungsfehler) in Kombination mit einem zu großen Integrationsintervall h unterliegt und über die Anzahl an Schritten n nicht weiter optimiert werden kann [7]. Es werden verschiedene Maximalwerte empfohlen, im Folgenden soll jedoch $n_{max} = 16$ gelten [7]. Kann der Fehler mit $n = 16$ nicht unter der Toleranz gehalten werden, folgt eine Halbierung der Schrittgröße ($h_{neu} = \frac{h_{alt}}{2}$) und es wird erneut bei $n = 2$ begonnen den neuen Schritt $[t_i, t_i + \frac{h_{alt}}{2}]$ zu integrieren. Bei erfolgreicher Integration folgt eine Verdopplung der Schrittgröße. Es gibt andere empirische Zusammenhänge für die Ermittlung der Schrittgröße h in Abhängigkeit der Bewegungsgrößen der Körper [9]. Aus eigener Erfahrung folgt jedoch das Prinzip der Halbierung und Verdoppelung. Eine Visualisierung dieses Vorgehens kann über ein Darstellung der Schrittgröße h des *BSV* in Abhängigkeit vom Zeitschritt getroffen werden:

Abbildung 5: ($h = 10^{-3}$, $\epsilon = 10^{-5}$,$\psi = 136$Bits entspricht 40Dez., $T = 10$, $\gamma = 1$)
Entwicklung der Schrittgröße h des Bulirsch-Stoer-Verfahren bei Simulation des *Pythagoreischen Dreikörperproblems*



Der Graph gehört zu einer Simulation des *Pythagoreischen Dreikörperproblems* bis $T = 10$ (Un-

terabschnitt 5.3). Die vier Einschnitte sind dabei mit den sehr nahen Begegnungen der Körper (Abb. 13, Reihe 3, Spalte 3) zu erklären. Somit adaptiert die Halbierung und Verdopplung, die Schrittgröße in Abhängigkeit der Situation des Systems. Nach [14] bieten Verfahren mit adaptiver Schrittgröße insgesamt mehr Potential für genauere Ergebnisse. Die Extrapolation stellt dabei einen essentiellen Teil des Verfahrens dar, und soll daher im Folgenden konkretisiert werden [7].

Der Grundgedanke bei der Extrapolation ist es, eine Funktion $g(\eta)$ zu erzeugen [1]. Diese Funktion soll dabei einen Zusammenhang zwischen den verschiedenen Integrationsergebnissen für einen Zeitschritt, und der dabei genutzten Schrittgröße $\eta = \frac{h}{n}$ herstellen. Das Ergebnis der Simulation mit $n = 2$, wird somit als z.B. $g(\frac{h}{2})$ angesehen [7]. Das Ziel hierbei ist es, den Wert $g(0)$ zu extrapolieren (potentielles Ergebnis mit einer Schrittgröße von $\eta = 0$), welcher insgesamt eine deutlich bessere Näherung für das gewünschte Integrationsergebnis bei $t+h$ darstellt, als die Ergebnisse mit $n = 2, 4, 6, ...$ Schritten.

Hierbei unterscheiden sich die Verfahren. Nach [7] und [9] liefert eine standardmäßige Polynomial-Extrapolation (z.B. mit *Neville's Algorithmus*) die effizientesten Ergebnisse und eine einfache Quantifizierung des Fehlers. Während nach [1], das Verfahren der *Richardson-Extrapolation* genutzt werden sollte.

Die *Richardson-Extrapolation* funktioniert nach der Annahme, dass für jede Schrittgröße $\eta = \frac{h}{n}$, ein Fehler existiert, welcher durch $g(0) = g(\eta) + c\eta^p$ beschrieben werden kann [1]. Wenn diese Fehlergleichung von zwei verschiedene Schrittgrößen η_1 und η_2 nach c umgestellt und gleichgesetzt wird, folgt [7]:

$$
\begin{aligned}
\frac{g(0) - g(\eta_1)}{\eta_1^p} &= \frac{g(0) - g(\eta_2)}{\eta_2^p} && | \quad \cdot \eta_2^p \\
\frac{\eta_2^p}{\eta_1^p} g(0) - \frac{\eta_2^p}{\eta_1^p} g(\eta_1) &= g(0) - g(\eta_2) && | \quad + \left(\frac{\eta_2}{\eta_1}\right)^p g(\eta_1) \\
\frac{\eta_2^p}{\eta_1^p} g(0) &= g(0) - g(\eta_2) + \frac{\eta_2^p}{\eta_1^p} g(\eta_1) && | \quad - g(0) \\
g(0) \left(\frac{\eta_2^p}{\eta_1^p} - 1\right) &= -g(\eta_2) + \frac{\eta_2^p}{\eta_1^p} g(\eta_1) && | \quad : \left(\frac{\eta_2^p}{\eta_1^p} - 1\right) \\
g(0) &= \frac{\left(\frac{\eta_2}{\eta_1}\right)^p g(\eta_1) - g(\eta_2)}{\left(\frac{\eta_2}{\eta_1}\right)^p - 1}
\end{aligned}
\tag{23}
$$

Hiermit lässt sich somit eine verbesserte Annäherung vom gesuchten Funktionswert treffen. Der Exponent p entspricht dabei der Fehlerordnung $\mathcal{O}$ des numerischen Verfahrens [1]. Die *Richardson-Extrapolation* wurde für das Lösen einer analytisch lösbaren DGL implementiert [27].

285 Zur Polynomial-Extrapolation lassen sich verschiedenen Algorithmen benutzen. Zum Einen, gibt es verschiedenen *Python*-Bibliotheken für Polynomial-Extrapolation (z.B. [30] & [31]), welche jedoch nicht komfortabel auf mehrere Dimensionen bzw. auf mehrere Größen gleichzeitig ausgeweitet werden können, was hierbei essentiell für das Dreikörperproblem ist. Zum Anderen

kann *Neville's Algorithmus* selbst in Python implementiert werden, was sich hierbei als beste Lösung herausstellte. Der Algorithmus basiert auf einer Tableau-Struktur, welche mithilfe von [7] (3.2.2) nachvollzogen werden kann. Das Tableau wird dabei nach jedem Wert für n, um eine Reihe erweitert, wobei der erste Wert der Reihe k ($P_{k,0}$), dem Ergebnis der Integration mit n_k-Schritten entspricht [7]:

$$P_{k,j} = P_{k,j-1} + \frac{P_{k,j-1} - P_{k-1,j-1}}{(\frac{n_k}{n_{k-j}})^2 - 1}, \qquad \text{für } j = 1, ..., k \tag{24}$$

wobei j der Spalte im Tableau entspricht [7]. Mithilfe von $err = |P_{k,k} - P_{k,k-1}|$ wird der Fehler berechnet, nachdem die Integration mit n_k-Schritten durchgeführt und Reihe k hinzugefügt wurde [7]. Folgend kann der Wert $P_{k,k}$ als verbessertes Ergebnis des Integrationsschrittes benutzt werden. Dieses Vorgehen funktioniert für mehrere Dimensionen (Alle Körper und Bewegungsgrößen gleichzeitig), weswegen jenes hier in Bezug auf das Dreikörperproblem genutzt wird [28]. Bei der Implementierung dieses Verfahrens, half [1] als Referenz, wobei *Neville's Algorithmus* und der Bezug auf das Dreikörperproblem vollständig selbst umgesetzt wurden.

5 Messung/Vergleich der Simulationen

Die Messung der Genauigkeit von numerischen Verfahren, erfolgt meist über einen Vergleich mit der analytischen Lösung. Da für das Dreikörperproblem keine analytische Lösung bekannt ist, muss auf andere physikalische Konzepte zurückgegriffen werden [4]. Um jedoch einen groben Eindruck der Genauigkeit der Schrittintegratoren zu bekommen, soll die Integration einer analytisch lösbaren DGL zum Vergleich herangezogen werden. Insgesamt wurden alle in Abschnitt 4 benannten Integratoren in der Programmiersprache *Python* implementiert und ausprobiert. In Bezug auf eine analytisch lösbaren DGL, wurden die folgenden Integratoren funktionstüchtig umgesetzt: *Euler-Verfahren*, *Euler-Cromer-Verfahren*, *Heun-Verfahren*, *RK2*, *RK4* und das *BSV* in Kombination mit dem *Modifizierten-Heun-Verfahren*. In Bezug auf das Dreikörperproblem, wurden die folgenden Integratoren funktionstüchtig umgesetzt: *Euler-(Cromer)-Verfahren*, *Heun-Verfahren*, *Leapfrog-Verfahren*, *Original-Verlet-Verfahren*, *Geschw.-Verlet-Verfahren*, sowie *RK4* und das *BSV*, welches aufgrund der Zeit-Effizienz, jedoch nur eingeschränkt nutzbar ist. Der Quelltext zu allen Integratoren wird in [27, 28] aufgeführt. Außerdem sind zu Integratoren, welche in dieser Facharbeit nicht thematisiert wurden, Anmerkungen beigefügt. Zur Darstellung des Messungen, wurde die *Python*-Bibliothek *Matplotlib* verwendet [32]. Eine animierte und interaktive Simulation mit dem Euler-Verfahren, umgesetzt mit *C#* und der „Game-Engine“ *Unity* [33], ist unter [34] und der dazugehörige Quelltext unter [35] zu finden.

5.1 Vergleich numerischer Verfahren

Die genannten Verfahren sollen eingesetzt werden um die DGL $y' = cos(x)$ numerisch zu lösen. Die analytischen Lösung dieser DGL ergibt sich nach Integrationsregeln als $y = sin(x)$. Der absolute Fehler der verschiedenen Verfahren lässt sich begutachten:

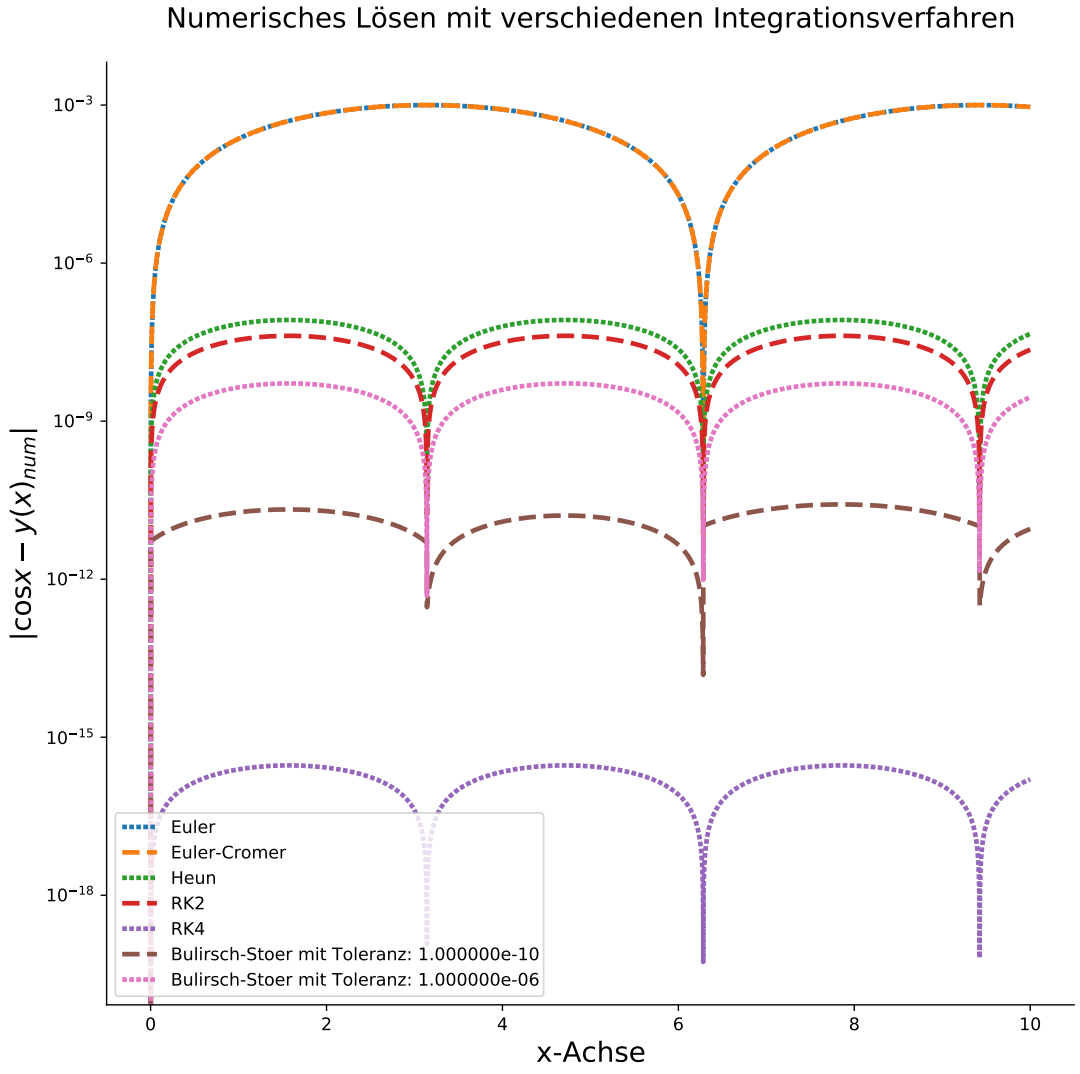


Abbildung 6: ($h = 10^{-3}$ und $\psi = 169$Bits entspricht 50Dez.)
Absoluter Fehler beim numerischen Lösen von $y' = cos(x) \Rightarrow y = sin(x)$

Dabei wird deutlich, dass der Fehler über ein periodisches Verhalten verfügt und somit auch vom Verhalten der approximierten Funktion abhängig ist. Insgesamt lässt sich den Erwartungen entsprechend, ein kleinerer Fehler bei Verfahren mit einer höheren Fehler Ordnung $\mathcal{O}$ ablesen. Das *RK2*, auf welches in dieser Facharbeit nicht eingegangen wurde, ist etwas präziser als das *Heun-Verfahren*, was sich auf die Ungenauigkeit des Mittelwertes zurückführen lässt. Eine genauere Beschreibung dazu, befindet sich im Jupyter-Notebook [27]. Zudem lässt sich erkennen,

wie die Verringerung der Toleranz ϵ des *BSV*, die absolute Abweichung, wie erwartet, reduziert.
330 Die Beobachtungen bestätigen sich auch bei Integration einer weiteren DGL:.

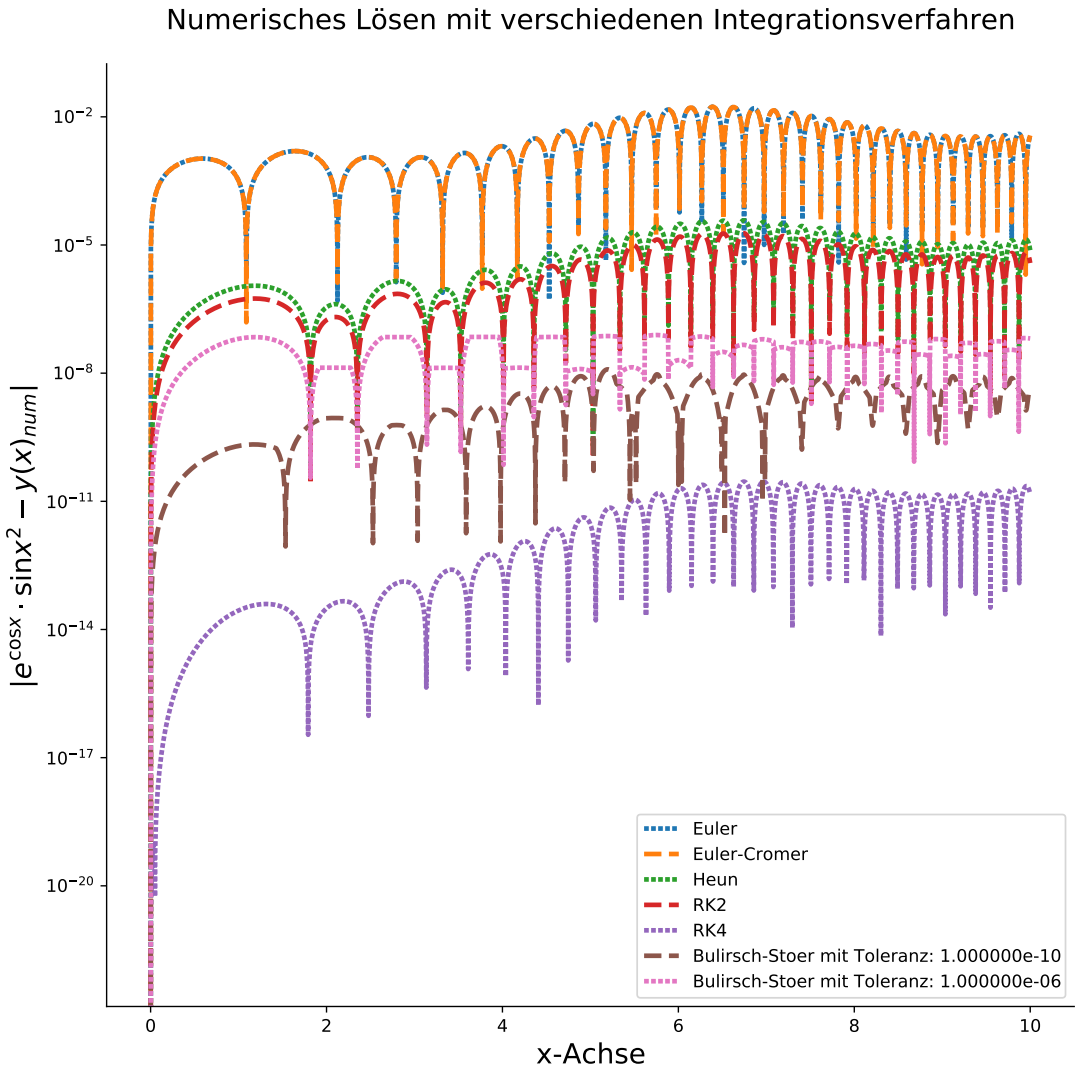


Abbildung 7: ($h = 10^{-3}$ und $\psi = 169$Bits entspricht 50Dez.)
Absoluter Fehler beim numerischen Lösen von $y' = e^{cos(x)} \cdot (2 \cdot x \cdot cos(x) - sin(x) \cdot sin(x^2)) \Rightarrow y = e^{cos(x)} \cdot sin(x^2)$

5.2 Fehlerentwicklung für verschiedene Schrittgrößen

Eine weitere Untersuchung des Fehlers, lässt sich mithilfe einer Variation der Schrittgröße für verschiedene Verfahren treffen. Hierbei soll der Mittelwert des Fehlers in Abhängigkeit der Schrittgröße für verschiedene numerische Verfahren gemessen werden. Abb. 8 zeigt das Verhalten des Mittelwertes der absoluten Fehler, errechnet über $\overline{err}_{abs} = \frac{\sum_{i=0}^{n} err_{i,abs}}{n}$, für verschiedene numerische Verfahren.

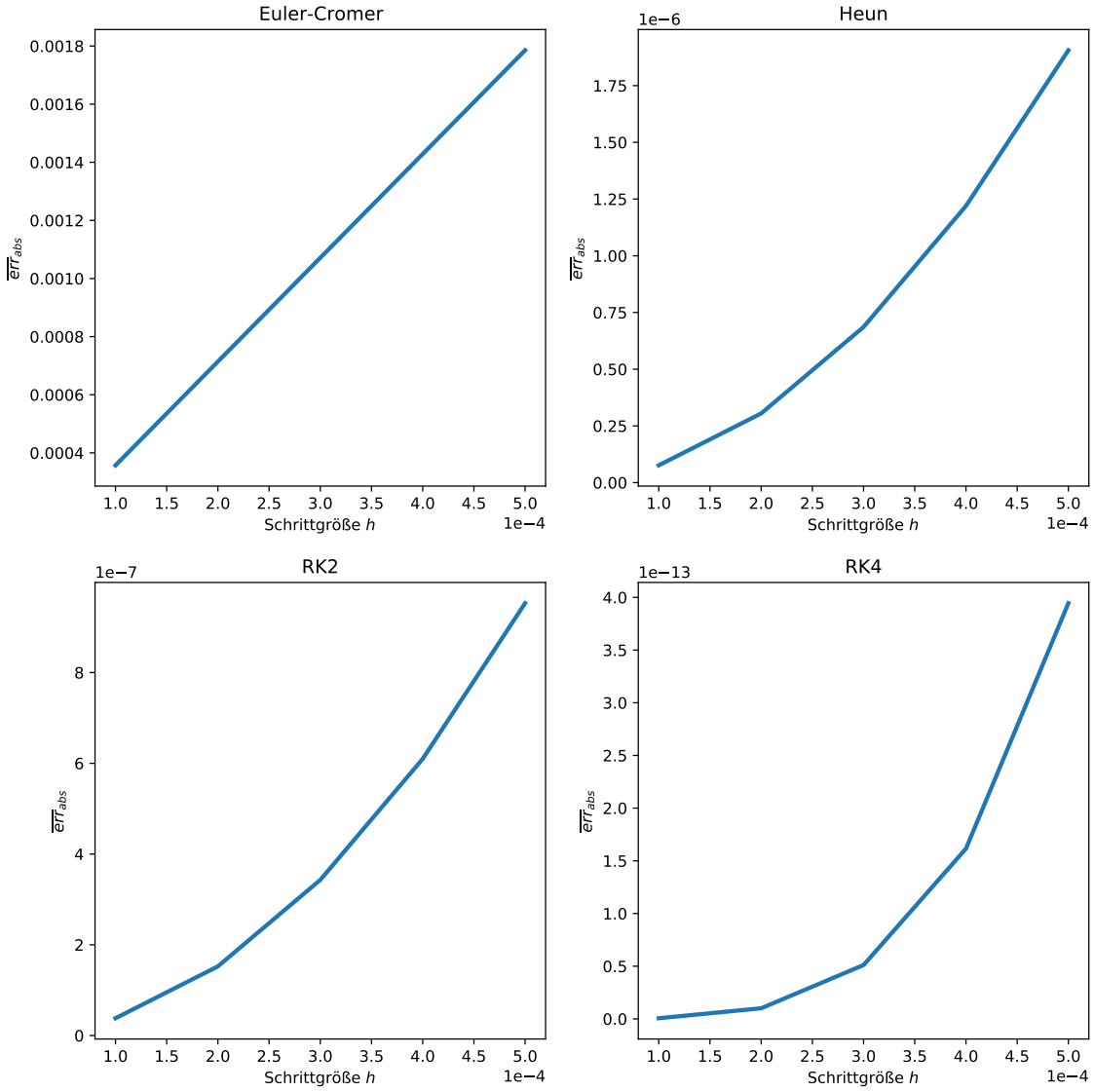


Abbildung 8: ($\psi = 169$Bits entspricht 50Dez.)
Mittelwert des absoluten Fehler beim numerischen Lösen von $y' = e^{cos(x)} \cdot (2 \cdot x \cdot cos(x) - sin(x) \cdot sin(x^2))$ in Abhängigkeit der Schrittgröße h

Das Globalverhalten der Zusammenhänge entspricht den Erwartungen:

$$lim_{h \to \infty} \overline{err}_{abs} \to \infty \tag{25}$$

Je größer die Schrittgröße h wird, desto größer wird $\overline{err}_{abs}$. Es lassen sich jedoch Unterschiede

in den Proportionalitäten erkennen $\overline{err}_{abs} \propto h^p$. Schätzungsweise lässt sich $p_{Euler-Cromer} = 1$, $p_{Heun} = 2, p_{RK2} = 2, p_{RK4} = 4$ ablesen. Es fällt auf, dass die Fehler der verschiedenen Verfahren mit der jeweiligen Ordnung $\mathcal{O}(h^p)$ übereinstimmen, welcher zuvor mathematisch herausgestellt wurde. Die Messungen sind somit eine Bestätigung der theoretischen Fehlerordnung der einzelnen Verfahren.

5.3 Pythagoreisches Dreikörperproblem

Als Startbedingungen für die Integration des Dreikörperproblems, werden hierbei die Bedingungen des *Pythagoreischen Dreikörperproblems* [24] (*PDKP*) verwendet, wobei die Körper in einem Masseverhältnis von 3 : 4 : 5 stehen und sich in der Konstellation eines Dreiecks befinden. Dabei entsprechen die Seitenlängen, den gegenüberliegenden Massen

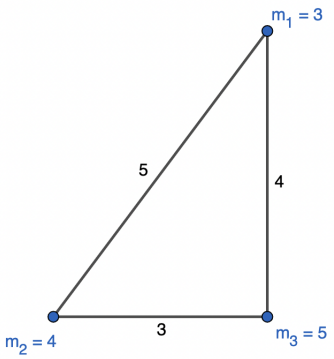


Abbildung 9: Startbedingungen des *Pythagoreischen Dreikörperproblems*

Die Startgeschwindigkeiten aller Körper sind 0, wodurch das System einen Gesamtimpuls von 0 besitzt [24]. Abb. 10 zeigt die Lösung nach [25] für das *PDKP*:

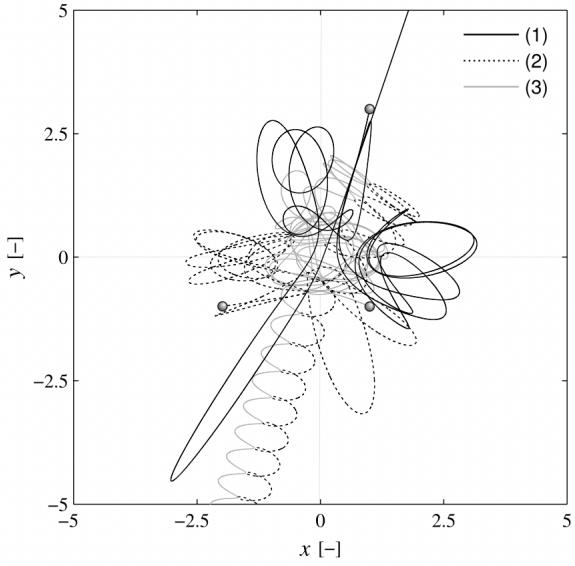


Abbildung 10: (Aus: [25]) Lösung des *Pythagoreischen Dreikörperproblems* (Punkte symbolisieren Startpunkte)

Eine Eigenschaft des *PDKP* ist das stark chaotische Verhalten und die mehreren sehr engen Begegnungen der Körper, was insgesamt eine Herausforderung für die numerischen Verfahren darstellt [20] Mit dem BSV ($\epsilon = 10^{-5}$ und $h = 10^{-3}$) soll das *PDKP* bis $T = 20$ integriert werden und somit die Ergebnisse aus [25] (Abb. 10) reproduziert werden:

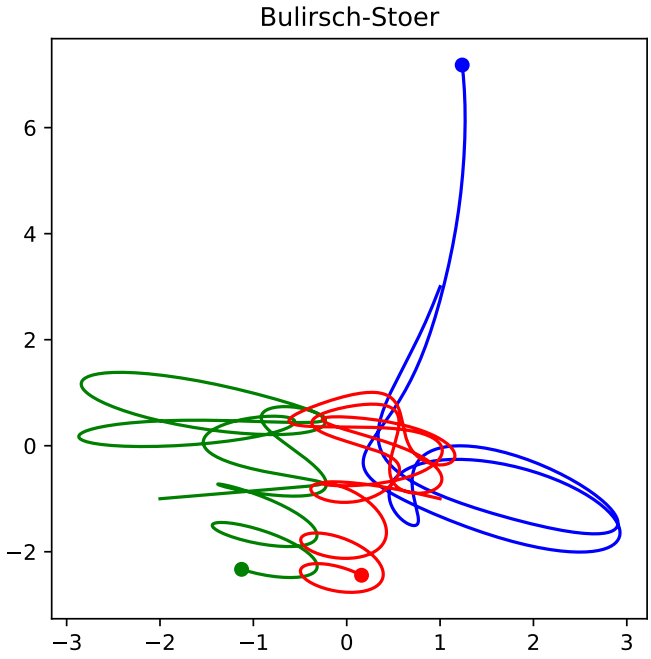


Abbildung 11: ($h = 10^{-3}$, $\epsilon = 10^{-5}$, $\psi = 136$Bits entspricht 40Dez., $T = 20$, $\gamma = 1$)
Simulation des *Pythagoreischen Dreikörperproblems* mit dem *Bulirsch-Stoer-Verfahren* (Daten: [36])

Die Ergebnisse (Abb. 11) zeigen ebenfalls wie Abb. 10, eine Spaltung in eine *binäre Struktur* und einen *Ausreißer* [24]. In Abb. 10 sind jedoch auffällig mehr *Windungen* und elliptische Bewegungen zu verzeichnen, bevor die Spaltung vorliegt. Beim Vergleich muss auf die Verzerrung durch die Skalierung geachtet werden. Leider konnten keine Rohdaten gefunden werden, um einen quantitativen Vergleich durchzuführen. Abb. 12 zeigt die Trajektorien bei Integration des *PDKP* mit allen genutzten Verfahren bis $T = 5$ und Abb. 13 bis $T = 10$. Bei Vergleich mit der Lösung bis $T = 10$ nach [24] (Abb. 5.3), wird ersichtlich, dass alle Integratoren, ausgenommen das *Heun-*, *RK4* und *Euler-Verfahren*, bei welchen womöglich aufgrund der engen Begegnungen $\ddot{\vec{r}}_i \to \infty$ (Kollision), eine ähnliche Flugbahn generieren und somit im Vergleich zu Abb. 5.3 plausible Ergebnisse erzeugen. Dabei sind jedoch leichte Abweichung des *BSV* zu erkennen.

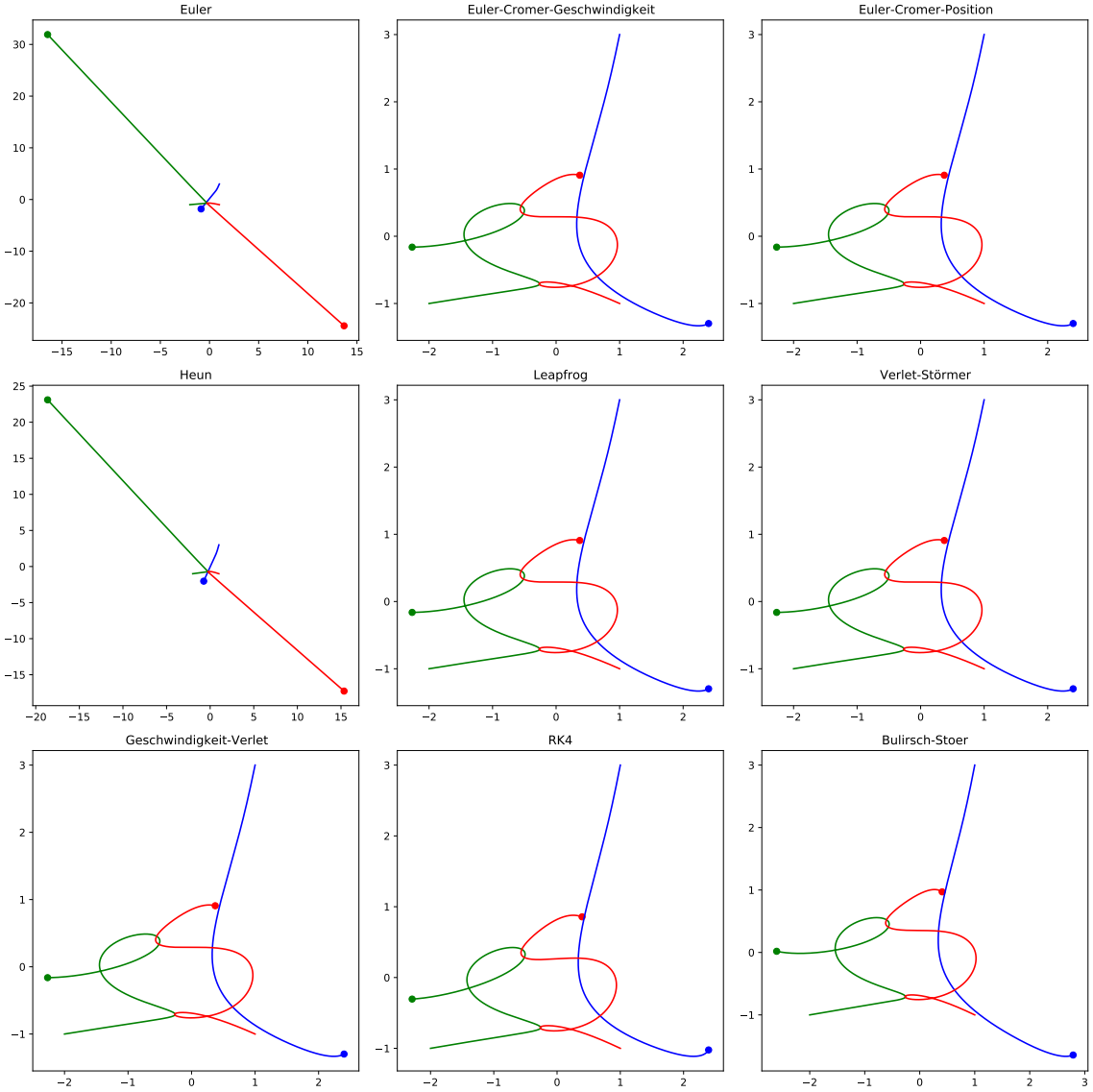


Abbildung 12: ($h = 10^{-4}, \psi =$ 136Bits entspricht 40Dez., $T = 5, \gamma = 1$) *Pythagoreisches Dreikörperproblem*

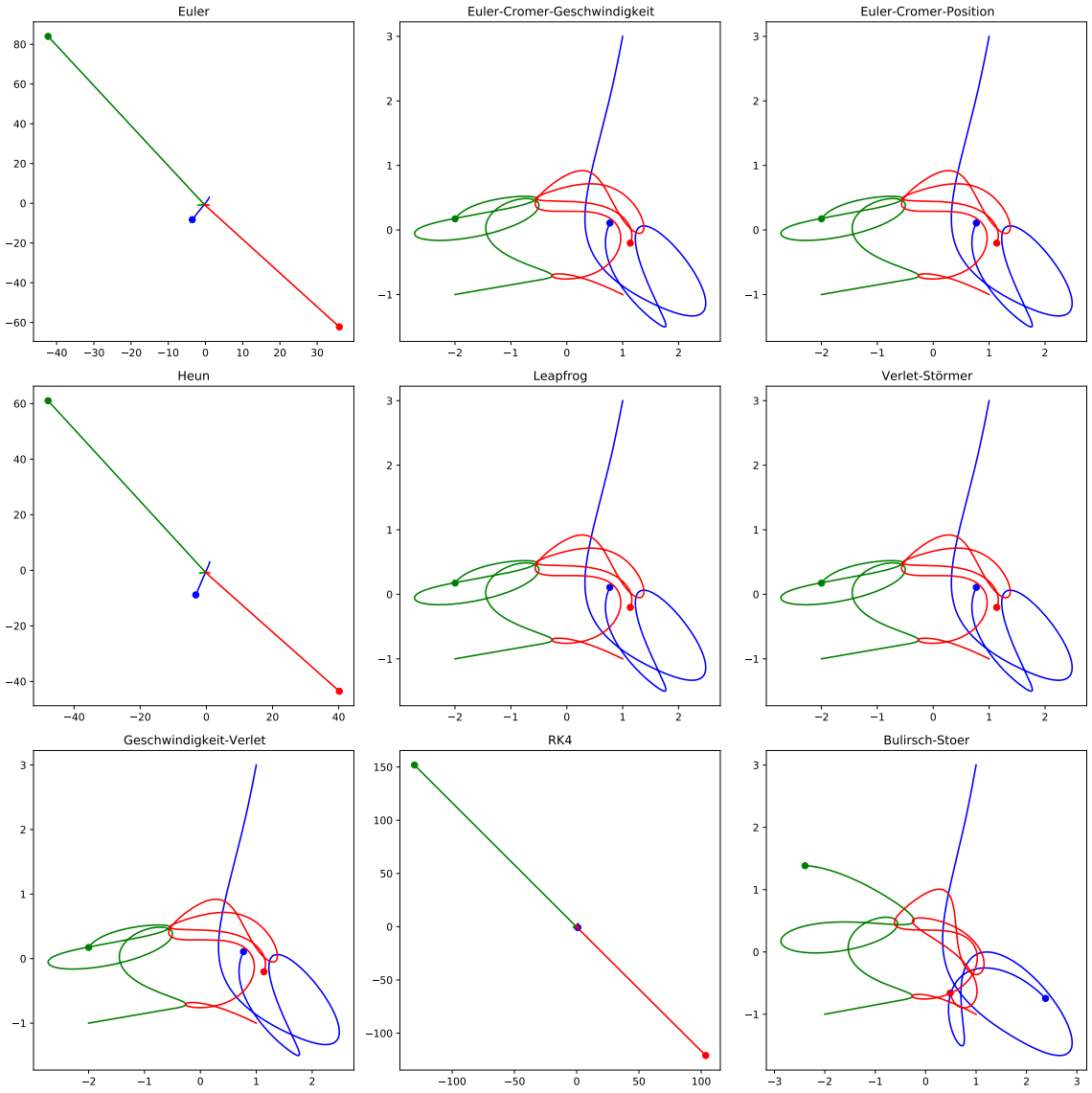


Abbildung 13: ($h = 10^{-4}, \psi = 136$Bits entspricht 40Dez.$, T = 10, \gamma = 1$) *Pythagoreisches Dreikörperproblem*

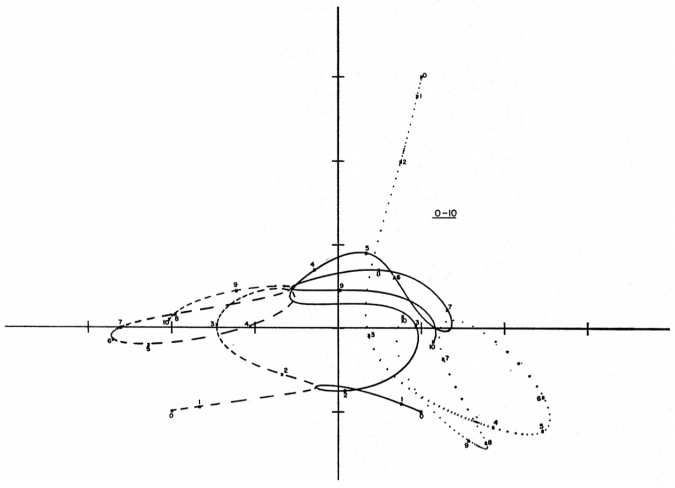


Abbildung 14: (Aus: [24]) Lösung des *Pythagoreischen Dreikörperproblems* bis $T = 10$

5.4 Impulserhaltung

Alle Kräfte ($\vec{F} = m \cdot \ddot{\vec{r}}$) in einem Dreikörpersystem gleichen sich gegenseitig aus [2, 14]. Nach Integration folgt daraus die Impulserhaltung des Systems:

$$\sum_{i=1}^{3} m_i \cdot \ddot{\vec{r}}_i = 0 \qquad \Rightarrow \qquad \sum_{i=1}^{3} m_i \cdot \dot{\vec{r}}_i = \vec{C} \tag{26}$$

Somit ist die Summe aller Impulse konstant. Diese Eigenschaft, lässt sich als Parameter zum Vergleich der numerischen Verfahren in Bezug auf das Dreikörperproblem verwenden [4]. Bei jedem Zeitschritt des Integrationsprozesses aller Verfahren bei Integration des $PDKP$ (Abb. 12), lässt sich der Gesamtimpuls nach Gl. 26 berechnen. Außerdem wird zu jedem Zeitschritt i die Differenz zum Anfangsimpulsvektor $\vec{\Delta}_i = \vec{C}_0 - \vec{C}_i$ berechnet. Die Länge dieses Vektors, sei die absolute Gesamtimpulsabweichung: $err_{abs,i} = |\vec{\Delta}_i|$. Abb. 15 zeigt $err_{abs,i}$ aller Verfahren für das $PDKP$ bis $T = 10$.

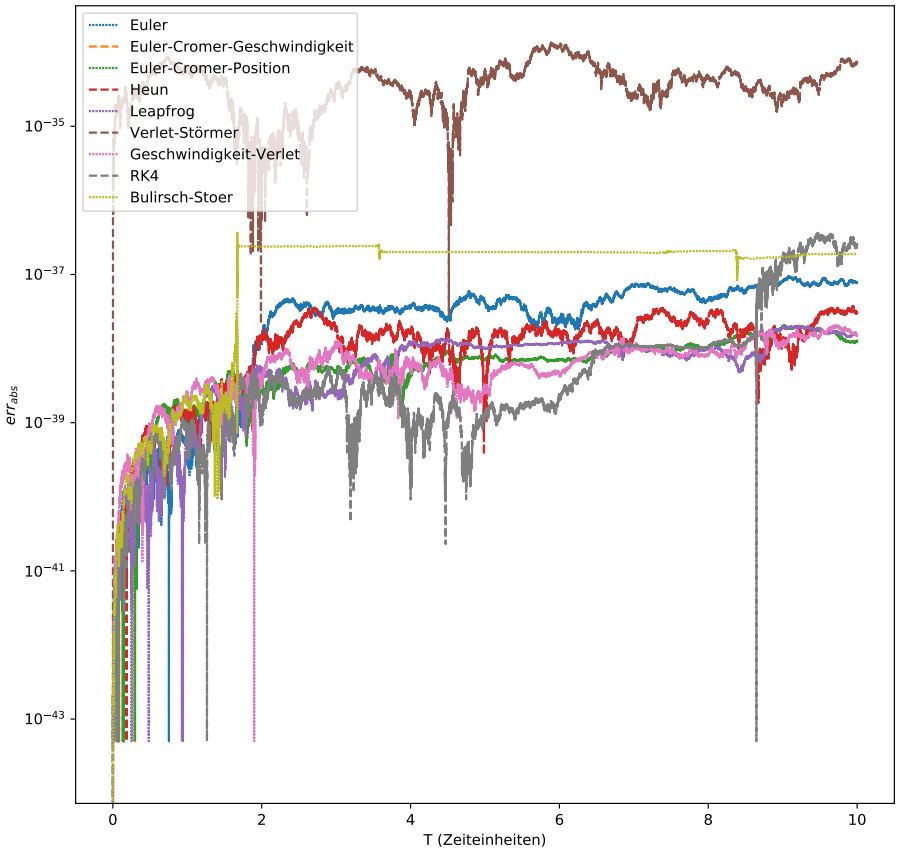


Abbildung 15: ($h = 10^{-4}, \psi = 136$Bits entspricht 40Dez.$, T = 10, \gamma = 1$) Absolute Impulsabweichung err_{abs} im *Pythagoerischen Dreikörperproblem* mit verschiedenen Integratoren

Zunächst lässt sich eine deutliche Absetzung des *Verlet-Störmer-Verfahrens* erkennen. Vermutlich hängt dies mit der Berechnung der Geschwindigkeit zusammen, welche hierbei im Gegensatz zu allen anderen Verfahren, über einen Differenzenquotienten mit der Position berechnet wird (Gl. 20). Insgesamt kann man sagen, dass die Impulserhaltung im betrachteten System, keine aussagekräftigen Ergebnisse liefert, da die kollidierten Systeme (*Euler-, Heun-Verfahern, RK4*), dennoch in einem akzeptablen Bereich bleiben. Eine wichtige Beobachtung, ist jedoch am Verhalten des *BSV*-Systems zu erkennen. Der Fehler err_{abs} bleibt aufgrund der adaptiven Schrittgröße und der Fehlertoleranz, über lange Zeiträume nahezu konstant, hierbei könnten weiterführende Untersuchungen der Toleranzänderung angestellt werden.

5.5 Energieerhaltung

Auch die Gesamtenergie, welche zu Anfang im System des Dreikörperproblems steckt, muss nach der *Energieerhaltung* vollständig erhalten bleiben. Es wird die relative Abweichung der Gesamtenergie bei jedem Zeitschritt berechnet [9]. Die relative Abweichung wird über den Zusammenhang $\Delta H_{rel} = \frac{|H_i - H_0|}{H_0}$, wobei H_i der Gesamtenergie bei Zeitschritt i entspricht. Dabei kann auf Gl. 5 zurückgegriffen werden. Intressant bei den Ergebnissen der Energieerhaltung (Abb. 16) ist, dass die Verfahren alle Systeme, abgesehen vom *BSV* und den kollidierten Systemen, über eine nahezu identische Energieabweichung verfügen. Dies ist mit der Eigenschaft eines *symplektischen*[5] Integrators zu erklären, was weiterführend über [21] nachvollzogen werden kann. Diese Eigenschaft markiert dabei auch den entscheidenden Unterschied zwischen den *Euler-Cromer-Verfahren* und dem *Euler-Verfahren* [21]. Eine weitere Beobachtung betrifft Ausschläge im Graphen, welche im Zusammenhang mit engen Begegnungen der Körper stehen.

[5] strukturerhaltenden (in Bezug auf die *Hamilton-Funktion*)

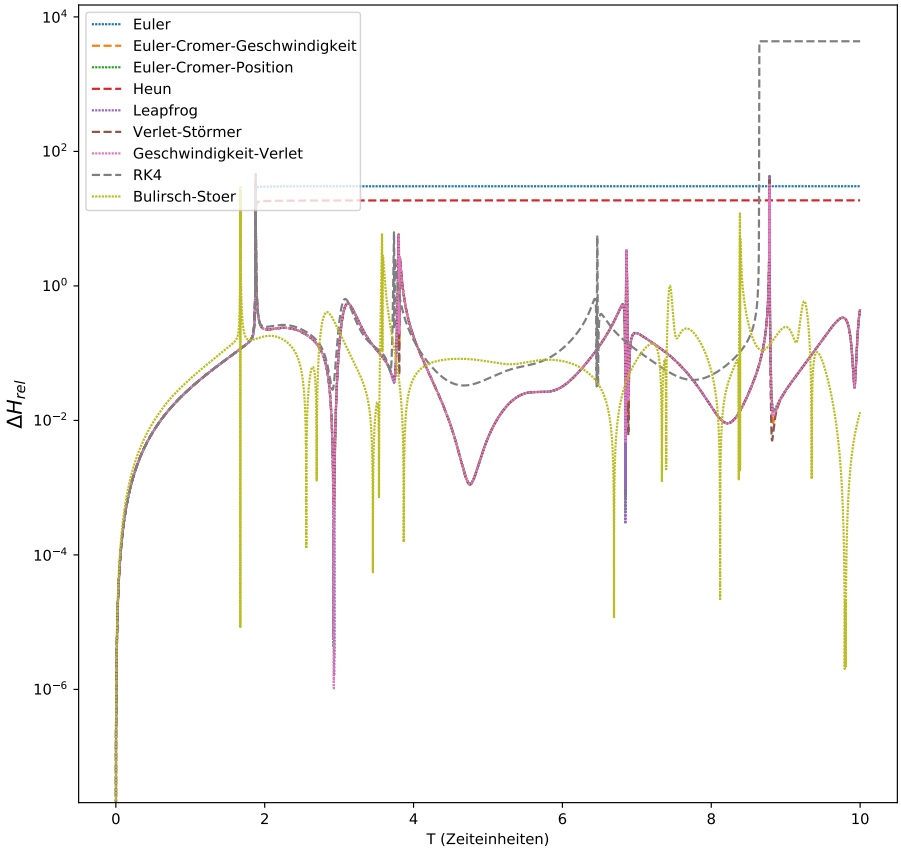


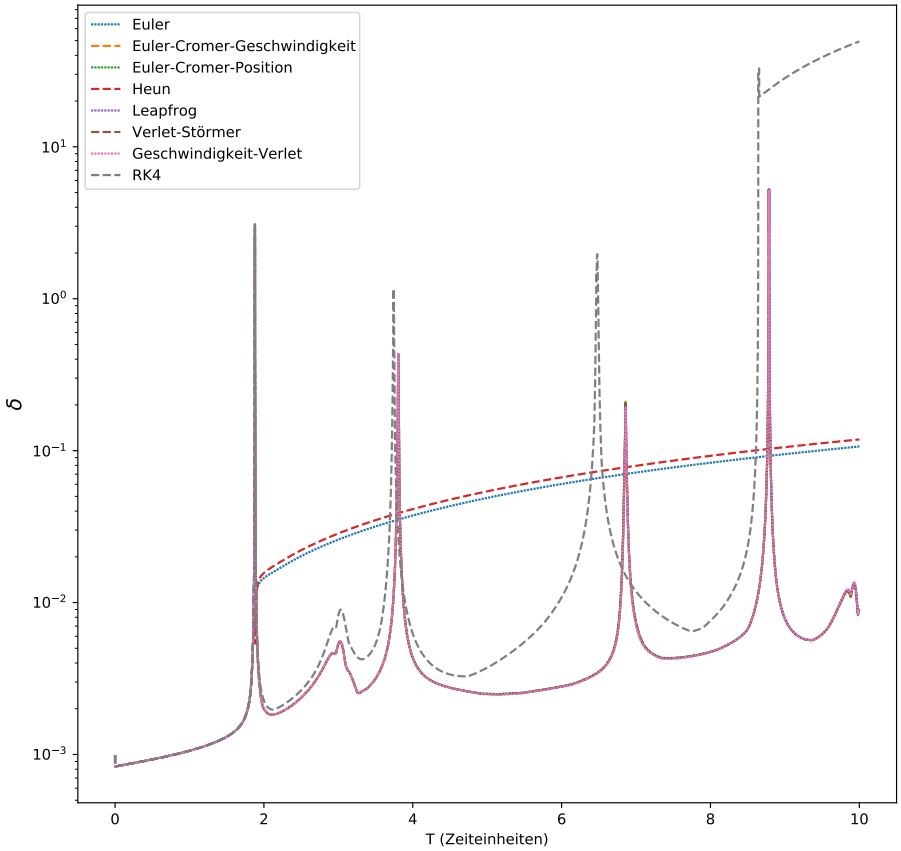
Abbildung 16: ($h = 10^{-4}, \psi = 136$Bits entspricht 40Dez.$, T = 10, \gamma = 1$) Relative Energieabweichung ΔH_{rel} im *Pythagoreischen Dreikörperproblem* mit verschiedenen Integratoren

5.6 Exponentielle Instabilität

Durch leichte Veränderung der Startbedingungen (verschieben von $m_1 = 3$ um 0.01 in x-Richtung), kann ein *gestörtes* System erzeugt werden [9]. Abb. 17 zeigt die Phasenraumdifferenz[6] (Gl. 7) zwischen dem *PDKP* und dem *gestörten PDKP* zu jedem Zeitpunkt. Die kollidierten Systeme, sowie die Kollisionszeitpunkte, lassen sich durch diese Auswertung klar ausfindig machen. Insgesamt lässt sich die Exponentielle Instabilität/Divergenz der beiden Systeme zeigen (*log*-Skala) [15], welche eine wichtige Eigenschaft des Dreikörperproblems ist. Hierbei wurde das *BSV* vernachlässigt, da sich in zwei verschiedenen Systemen auch die Schrittgrößen unterscheiden würden, was die Auswertung auf diesem Wege erschwert.

[6]Differenz der Phasen (generalisierten Koordinaten)

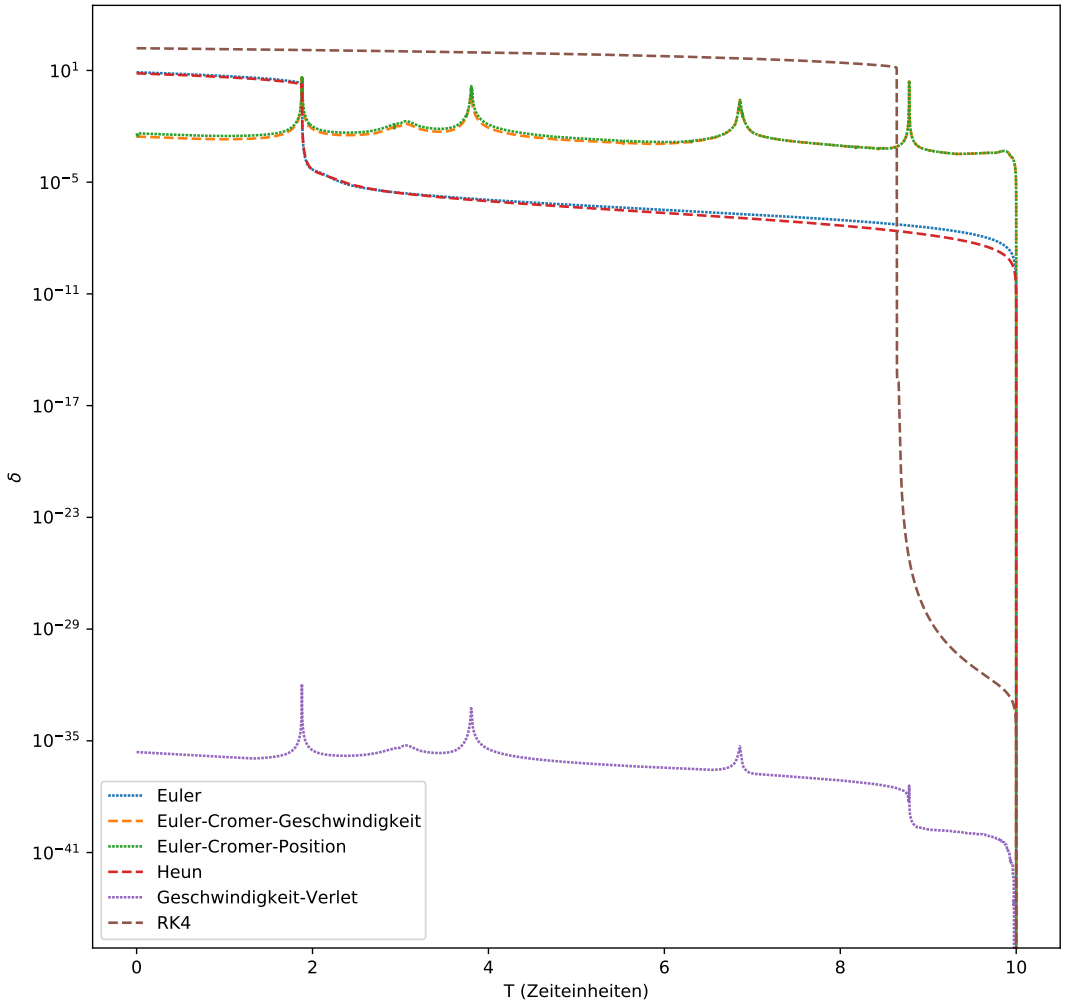
Abbildung 17: ($h = 10^{-4}, \psi = 136$Bits entspricht 40Dez., $T = 5, \gamma = 1$) Entwicklung der Differenz δ zwischen dem *Pythagoreischen Dreikörperproblem* und einem leicht perturbierten System (x-Koordinate von $m_1 = 3$ um 0.01 verschoben)



5.7 Zeit-Reversibilität

Wenn an einem bestimmten Zeitpunkt, alle Geschwindigkeiten negiert werden ($\dot{\vec{r}}_{n-1,B} = -\dot{\vec{r}}_{n-1,A}$), müssten die 3 Körper ihre Flugbahn genau zurückverfolgen [4]. Durch die bereits herausgestellten Fehlerquellen (Unterabschnitt 4.1), kommt es jedoch zu Abweichungen. Diese Messung wurde aufgrund der Zeit-Effizienz nicht für alle Verfahren durchgeführt. Abb. 18 zeigt δ zwischen dem simulierten System bis $T = 10$ und dem umgekehrten System, gemäß Gl. 7. Dabei wird mit dem Betrag der Impulse gerechnet, da alle Geschwindigkeiten negiert wurden. Hierbei zeigt sich, die Eigenschaft der Zeit-Reversibilität der *VLV*, welche mithilfe von [21] weiter nachvollzogen werden kann. Die kollidierten Verfahren zeigen erwartungsgemäß, bis zu ihren Kollisionszeitpunkten eine akzeptable Differenz δ, was vermutlich mit den stabilen Flugbahnen nach der Kollision zusammenhängt.

Abbildung 18: ($h = 10^{-4}, \psi = 136$Bits entspricht 40Dez.$, T = 10, \gamma = 1$) Zeit-Reversibilität (Phasenraumdistanz δ der Hin- und Rücksimulation)

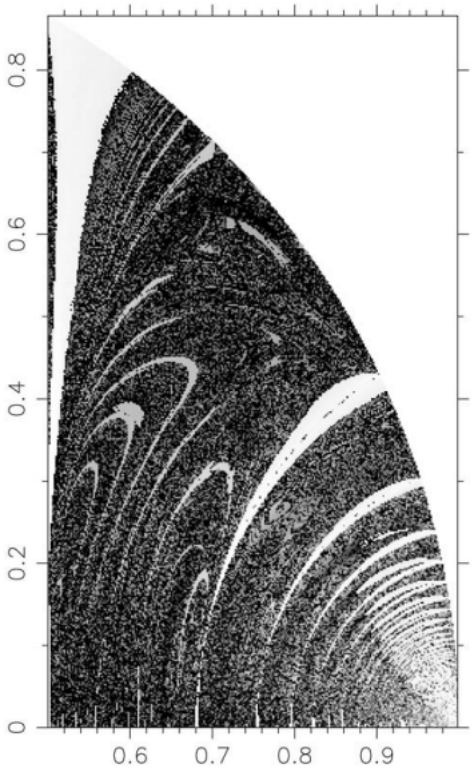


5.8 Chaos im Dreikörperproblem

Ein ursprünglicher Anreiz zur Erkundung des Dreikörperproblems bestand auch in der Faszination eines chaotischen Systems. Um dieses Prinzip des Chaos in Bezug auf das Dreikörperproblem zu reproduzieren und zu visualisieren, eignet sich eine *Agekyan-Anosova Karte* (AA) [4]. Dabei wird die *Lebenszeit* eines Systems (T) in Abhängigkeit der Startposition eines Körpers dargestellt [4]. Die Startbedingungen der beiden anderen Körper bleiben dabei in allen simulierten Systemen gleich. Für alle Massen und Startgeschwindigkeiten gilt $m = 1$ und $\dot{\vec{r}}_0 = \vec{0}$. Für den dritten Körper wird nun wiederholt die Startposition angepasst und eine Simulation durchgeführt. T soll nach [4] genau die Zeit sein, welche vergeht, bis sich das System in mindestens zwei Teile aufspaltet oder eine Kollision stattfindet. Abb. 19 zeigt die Ergebnisse nach [4], wo-

bei die fixierten Startpositionen $(-0.5, 0)$ und $(0, 0.5)$ gewählt wurden. Die Position des dritten Körpers variiert in einer speziellen Form im Bereich $(0 \leq x \leq 0.5, 0 \leq y \leq 1)$ [4].

Abbildung 19: (Aus: [4]) *Agekyan–Anosova Karte*: Die *Lebenszeit* des Tripelsystems in Abhängigkeit der Startposition eines Körpers. Die *Lebenszeit* ist in hellen Bereichen der Karte kurz, in den dunklen Bereichen lang.



Dabei ist die Beschriftung $0.5 \rightarrow 1$ etwas irreführend. Für die Reproduktion, wurde als Maß für T die Zeit gewählt, die verstrichen ist, bis sich zwei verschiedene Körper mindestens weiter als 10 Längeneinheiten entfernt haben, oder eine Kollision eingetreten ist. Bei diesem Vorgehen, mit den Startpositionen: $(-1, 0), (0, 1), (-0.5 \leq x < 0.5, 0 \leq y < 1)$ und der Simulation mithilfe des *Leapfrog-Integrators* (Gl. 16) mit Schrittgröße $h = 10^{-3}$ und einer Präzision von $\psi = 169$ Bits bzw. 50 Nachkommastellen, konnten folgende Ergebnisse erzielt werden (Abb. 20).

Darstellung **1** zeigt hierbei für 10000 Simulationen ($\approx$ 8h Simulationsdauer), die verschiedenen T des Dreikörpersystems in Zeiteinheiten gemäß der Farbleiste. Es stellt sich heraus, das aufgrund der Symmetrie der Startbedingungen, auch eine Symmetrie in T vorhanden ist. **3** zeigt dabei den Raum $-0.5 \rightarrow 0$ für die Startbedingungen des dritten Körpers (Linke Hälfte von **1**). Ein weiterer Schluss, den man aus diesen Simulationbedingungen ziehen kann, ist die Differenz Δ, wenn beide Seiten der Symmetrie aufeinander projiziert werden. In der Theorie müssten beide Seiten aufgrund der achsensymmetrischen Startbedingungen identische T aufweisen. Durch mögliche numerische Ungenauigkeiten und Rundungsfehler kommt es jedoch zu einigen Abweichungen, welche in **2** dargestellt werden. Diese Ergebnisse sind aufgrund des Determinismus schlecht nachzuvollziehen. Um eine vergleichbare Darstellung mit den Ergebnissen aus [4] zu haben (Abb. 19), zeigt Abb. 21, 5000 Simulationen mit den Startbedingungen $(-0.5, 0), (0, 0.5), (0 \leq x < 0.5, 0 \leq y < 1)$ und dem *Leapfrog-Integrator* mit der Schrittgröße

$h = 10^{-3}$. Die Ergebnisse nach [4] lassen sich aufgrund des unbekannten Vorgehens und der unvergleichbaren Anzahlen an Simulationen, schlecht reproduzieren. Dennoch sind einige Folgerungen übertragbar. Aus Abb. 20 und Abb. 21 lässt sich erkennen, dass T eines simulierten Systems, sich ungefähr an seinen Nachbarn ableiten lässt, jedoch insgesamt schlecht vorhersehbar ist [4]. Durch einige Strukturen, wie beispielsweise im Bereich $y > 0.9$ (Abb. 21) oder die weißen Bereiche aus Abb. 19, lässt sich auf kein *vollständiges Chaos*, sondern ein sogenanntes *schwaches Chaos* schließen [4].

Abbildung 20: ($h = 10^{-3}, \gamma = 1, \psi = 136$Bits entspricht 40Dez. $[-0.5 \leq x < 0.5 \mid 0 \leq y < 1]$) 10000 Simulationen mit Leapfrog-Verfahren (Abhängigkeit T von Startposition, Daten: [37])

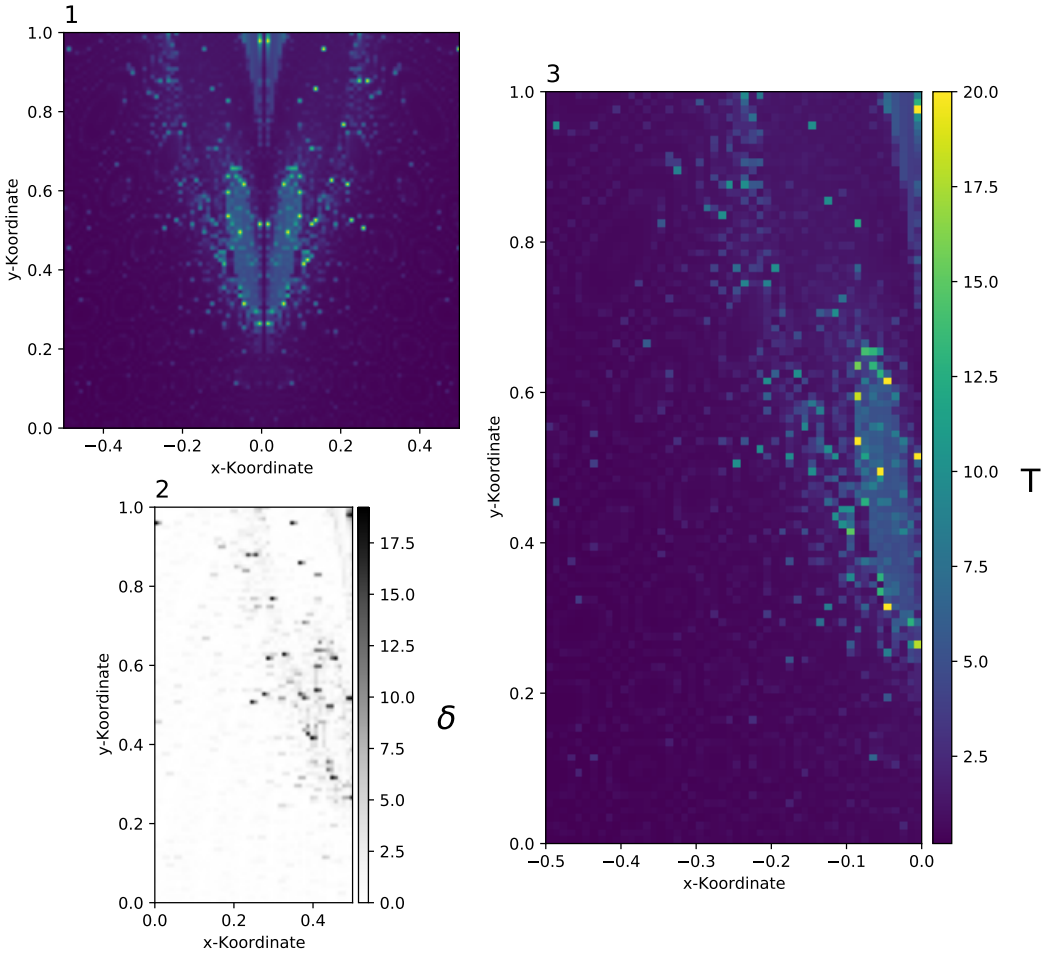
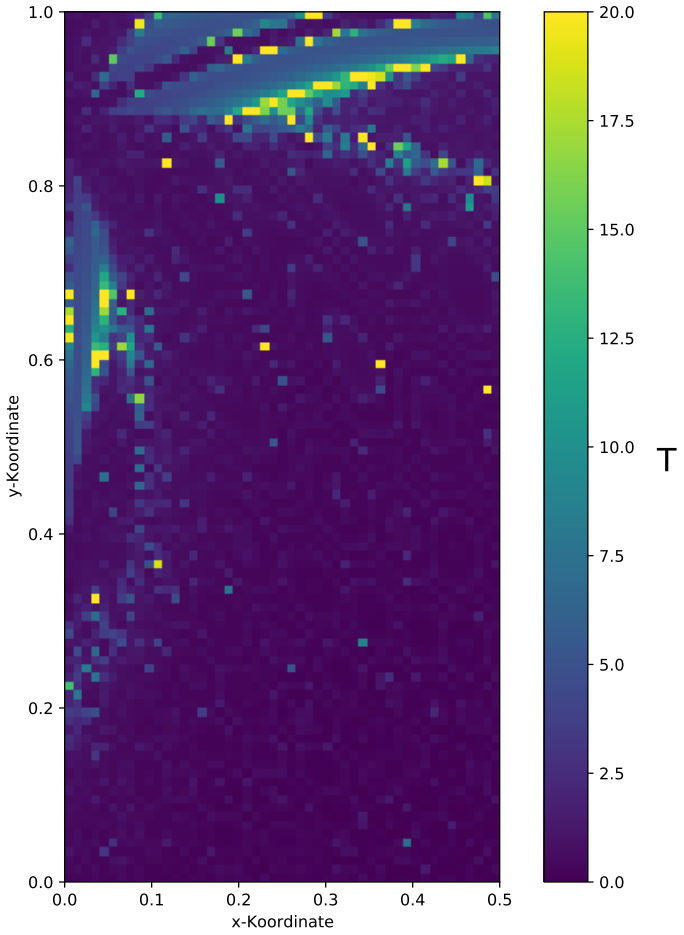


Abbildung 21: ($h = 10^{-3}, \gamma = 1, \psi = 136$Bits entspricht 40Dez. $[0 \leq x < 0.5 \mid 0 \leq y < 1]$)
5000 Simulationen mit Leapfrog-Verfahren (Abhängigkeit T von Startposition, Daten: [38])



6 Fazit

Rückblickend lässt sich sagen, dass der Themenrahmen der Numerik in Anwendung auf das Dreikörperproblem, viele Konzepte der theoretischen Physik, Mathematik und Informatik anbietet. In dieser Facharbeit sollten dabei die physikalischen Grundlagen und die mathematischen Konzepte von numerischer Integration zielführend für die Messung von Präzision am Dreikörperproblem eingesetzt werden. Der Erkenntnisgewinn besteht dabei vor allem in der theoretischen Physik (z.B. *Hamilton-Mechanik*) und dem Verständnis von numerischen bzw. mathematischen Konzepten (z.B. *Neville-Algorithmus*). Das Nachvollziehen von Messverfahren sowie die gewonnene Kreativität zum Ausprobieren und Entwerfen von eigenen Fragestellungen führt zu vielen Ideen, welche fortlaufend beantwortet werden sollen. So könnte die Bewegung des Massezentrum in Abhängigkeit der Zeit untersucht werden [4]. Auch die Korrelation und Entwicklung der Zeit-Effizienz in Abhängigkeit von Schrittgröße oder Toleranz kann dargestellt werden (auch Zeit-Komplexität der Algorithmen bestimmen [theoretische Informatik]).

Der Zusammenhang zwischen dem Rundungsfehler (bzw. Präzision ψ) und der Genauigkeit der Ergebnisse sollte ermittelt werden. [9] stellt verschiedene empirische Zusammenhänge zwischen h, ψ, ϵ auf, welche ausprobiert werden sollten. Insgesamt lassen sich Optimierungen des Quelltextes vornehmen und die Verfahren auf eine effizientere Programmiersprache (z.B. C++) übertragen. Adaptive Schrittgröße in Bezug auf Runge-Kutta-Verfahren stellt ein weitere Optimierungsmöglichkeit dar [7]. Denkbar wäre auch ein Vergleich mit einem bestehenden Integrator oder das eigenständige Ermitteln von periodischen Orbits [14]. Offene Fragestellungen auf theoretischer Ebene, betreffen die Eigenschaft eines *symplektischen Integrator* und die *Regularisierung* der Bewegungsgleichungen [14]. Die Integrationsergebnisse sollen in Zukunft auch dazu verwendet werden, ein *Neuronales-Netz* zu trainieren, und damit den Bezug zu *Machine-Learning* herzustellen, was vor allem durch privates Interesse motiviert ist. Zu diesem Themenbereich gibt es bereits einige interessante Publikationen: [11, 26].

Referenzen

Literaturverzeichnis

[1] Jaan Kiusalaas. *Numerical Methods in Engineering with Python 3*. New York, NY, USA: Cambridge University Press, 2013. ISBN: 9781107033856.

[2] Volker Maiwald, Dominik Quantius und Benny Rievers. *Grundlagen der Orbitmechanik*. Bremen, Deutschland: Carl Hanser Verlag München, 2020. ISBN: 9783446462793.

[3] Karin Benecke und Lutz Breidert u.a. *Elemente der Mathematik, Qualifikationsphase Niedersachsen: erhöhtes Anforderungsniveau*. Braunschweig, Deutschland: Westermann Gruppe, 2018. ISBN: 9783507891135.

[4] Mauri Valtonen und Hannu Karttunen. *The Three-Body Problem*. New York, NY, USA: Cambridge University Press, 2006. ISBN: 9780511132896.

[5] K. A. Tsokos. *Physics for IB Dimploma (Standard and Higher Level)*. Cambridge University Press, 2010. ISBN: 9780521138215.

[6] A. Kharab und Ronald B. Guenther. *An Introduction to Numercial Methods*. Boca Raton, FL, USA: CRC Press, 2019. ISBN: 9781138093072.

[7] William H. Press u. a. *Numerical Recipes (Third Edition)*. New York, NY, USA: Cambridge University Press, 2007. ISBN: 9780511335556.

[8] Philipp O.J. Scherer. *Computational Physics: Simulation of Classical and Quantum Systems*. Heidelberg, Deutschland: Springer, 2010. ISBN: 9783642139895.

Quellenverzeichnis

[9] Tjarda Boekholt und Simon Portegies Zwart. „On the Reliability of N-body Simulations“. In: *Computational Astrophysics and Cosmology* (2014). URL: https://arxiv.org/abs/1411.6671.

[10] Wilhelm Kley und Christoph Schäfer. *Chaos in planetary systems*. 2021. URL: https://www.tat.physik.uni-tuebingen.de/~schaefer/teach/f/chaos_english.pdf.

[11] Jing Li, Xiaoming Li und Shijun Liao. *Can machine learning really solve the three-body problem?* 2020. URL: https://www.researchgate.net/publication/343563677_Can_machine_learning_really_solve_the_three-body_problem.

[12] Prof. Dr. W. Bauhofer u.a. *Lexikon der Physik: Massenpunkt*. 1998. URL: https://www.spektrum.de/lexikon/physik/massenpunkt/9474.

[13] David Michael Hernandez. „Solving the N-body Problem in Astrophysics“. Doctor of Philosophy. Massachusetts Institute of Technology. Department of Physics, 2018. URL: https://dspace.mit.edu/handle/1721.1/119107.

[14] Z.E. Musielak und B. Quarles. *The three-body problem*. 2015. URL: https://arxiv.org/abs/1508.02312.

[15] Jeremy Goodman, Douglas C. Heggie und Piet Hut. „On the Exponential Instability of N-Body Systems“. In: *The Astrophysical Journal* (1993). URL: https://adsabs.harvard.edu/full/1993ApJ...415..715G.

[16] Matthias Kemper und Guido Willems. *Das 3-Körper-Problem*. 2012. URL: https://www.uni-muenster.de/imperia/md/content/physik_ft/pdf/ws1112/seminar/111918/willems-kemper.pdf.

[17] Prof. Dr. W. Bauhofer u.a. *Lexikon der Physik: verallgemeinerte Koordinaten*. 1998. URL: https://www.spektrum.de/lexikon/physik/verallgemeinerte-koordinaten/15122.

[18] Prof. Dr. W. Bauhofer u.a. *Lexikon der Physik: Freiheitsgrad*. 1998. URL: https://www.spektrum.de/lexikon/physik/freiheitsgrad/5308.

[19] Prof. Dr. W. Bauhofer u.a. *Lexikon der Physik: Analytische Mechanik*. 1998. URL: https://www.spektrum.de/lexikon/physik/analytische-mechanik/502.

[20] S. Aarseth u. a. „Global chaoticity in the Pythagorean three-body problem“. In: *Celestial Mechanics and Dynamical Astronomy* (1994). URL: https://www.researchgate.net/publication/226415727_Global_chaoticity_in_the_Pythagorean_three-body_problem.

[21] Peter Young. *Physics 115/242: The leapfrog method and other "symplectic" algorithms for integrating Newton's laws of motion*. 2014. URL: https://young.physics.ucsc.edu/115/leapfrog.pdf.

[22] Helge Todt. *Computational Astrophysics I: Introduction and basic concepts*. 2021. URL: https://www.astro.physik.uni-potsdam.de/~htodt/ca/three-n-body.pdf.

[23] Loup Verlet. „Computer Experiments on Classical Fluids. Thermodynamical Properties of Lennard-Jones Molecules“. In: *Physical Review* (1967). URL: https://journals.aps.org/pr/abstract/10.1103/PhysRev.159.98.

[24] Victor Szebehely und C. Frederick Peters. „Complete solution of a general problem of three bodies“. In: *The Astronomical Journal* (1967). URL: https://adsabs.harvard.edu/full/1967AJ.....72..876S.

[25] Javier Roa, Hodei Urrutxua und Jesus Pelaez. „Stability and chaos in Kustaanheimo-Stiefel space induced by the Hopf fibration“. In: *Monthly Notices of the Royal Astronomical Society* (2016). URL: https://www.researchgate.net/publication/299776019_Stability_and_chaos_in_Kustaanheimo-Stiefel_space_induced_by_the_Hopf_fibration.

[26] Philip G. Breen u. a. „Newton vs the machine: solving the chaotic three-body problem using deep neural networks“. In: *Monthly Notices of the Royal Astronomical Society* (2019). URL: https://arxiv.org/abs/1910.07291.

Weiteres

[27] Tobias Steinbrecher. *Quelltext: Vergleich numerischer Verfahren für das Lösen einer Differentialgleichung.* URL: https://github.com/Tobotis/Three-Body-Problem/blob/main/Vergleich_Numerische_Integration.ipynb.

[28] Tobias Steinbrecher. *Quelltext: Vergleich numerischer Verfahren für das Lösen des Dreikörperproblems.* URL: https://github.com/Tobotis/Three-Body-Problem/blob/main/Vergleich_Numerische_Integration.ipynb.

[29] *mpmath: python arbitrary precision library.* URL: https://mpmath.org/.

[30] *numpy: np.polyfit (python library for polynomial-extrapolation).* URL: https://numpy.org/doc/stable/reference/generated/numpy.polyfit.html.

[31] *scipy: lagrange interpolation (library for polynomial-extrapolation).* URL: https://docs.scipy.org/doc/scipy/reference/generated/scipy.interpolate.lagrange.html.

[32] *Matplotlib: Visualization with Python.* URL: https://matplotlib.org/stable/.

[33] Unity Technologies. *Unity Real-Time Development Platform — 3D, 2D VR & AR Engine.* URL: https://unity.com/.

[34] Tobias Steinbrecher. *Simulation (Euler-Verfahren) des Dreikörperproblems (C-Sharp/Unity).* URL: https://tobotis.github.io/NBodySimWebGL/.

[35] Tobias Steinbrecher. *Quelltext: Simulation (Euler-Verfahren) des Dreikörperproblems (C-Sharp/Unity).* URL: https://github.com/Tobotis/NBodySim/tree/main/NBodySimulation/Assets/Scripts.

[36] Tobias Steinbrecher. *Daten: Simulation des Pyth. Dreikörperproblems mit dem Bulirsch-Stoer-Verfahren bis $T = 20$.* URL: https://github.com/Tobotis/Three-Body-Problem/blob/main/data_bs_20.txt.

[37] Tobias Steinbrecher. *Daten: Variation der Positionen (Erkundung chaotisches Verhalten 1).* URL: https://github.com/Tobotis/Three-Body-Problem/blob/main/chaos_data1.txt.

[38] Tobias Steinbrecher. *Daten: Variation der Positionen (Erkundung chaotisches Verhalten 2).* URL: https://github.com/Tobotis/Three-Body-Problem/blob/main/chaos_data2.txt.

7 Versicherung der selbstständigen Anfertigung

Hiermit versichere ich, dass ich die Arbeit selbstständig angefertigt, keine anderen als die angegebenen Hilfsmittel benutzt und die Stellen der Facharbeit, die im Wortlaut oder im wesentlichen Inhalt aus anderen Werken entnommen wurden, mit genauer Quellenangabe kenntlich gemacht habe.

Hannover, 14.01.2022
Tobias Steinbrecher

8 Einverständniserklärung zur schulinternen Veröffentlichung

Ich bin mit der schulinternen Veröffentlichung dieser von mir angefertigten Seminarfacharbeit einverstanden.

Hannover, 14.01.2022
Tobias Steinbrecher