# 1 Das klassische $N$-Körper-Problem

(Roland Speith und Christoph Schäfer, Stand 17. Oktober 2021)

## 1.1 Das Anfangswertproblem

Als Anfangswertproblem wird eine in der Physik wichtige Klasse von Differentialgleichungen bezeichnet. Die Lösung des Anfangswertproblems ist die Lösung der Differentialgleichung unter zusätzlicher Berücksichtigung gegebener Anfangswerte. Mathematisch ist es folgendermaßen definiert.

**Definition 1.** *Gegeben ist eine auf der Menge $D$ der $(t, y)$-Ebene erklärte Funktion $f(t, y)$ und ein fester Punkt $(\xi, \eta) \in D$. Gesucht ist eine in einem Intervall $J$ (mit $\xi \in J$) differenzierbare Funktion $y(t)$, für welche*

$$\frac{\mathrm{d}y}{\mathrm{d}t} = y'(t) = f(t, y(t)) \quad in \quad J, \tag{1.1}$$

$$y(\xi) = \eta. \tag{1.2}$$

*gilt.*

Die Gleichung (1.2) wird Anfangsbedingung genannt. Eine weitere wichtige Klasse von Differentialgleichungen sind Randwertprobleme, bei denen anstatt Anfangsbdingungen zusätzliche Bedingungen an die Lösung auf dem Rand des Definitionsbereichs der Funktion gegeben sind. In diesem Projekt befassen wir uns aber ausschließlich mit der numerischen Lösung von Anfangswertproblemen.

## 1.2 Das klassische $N$-Körper-Problem

Ein typisches Anfangswertproblem ist das $N$-Körper-Problem. Das klassische $N$-Körper-Problem ist in der Astrophysik von großer Bedeutung für unser Verständnis der Entwicklung und Stabilität von Planetensystemen, Sternhaufen und Galaxien. Die Dynamik dieser Vielteilchensysteme wird dominiert von der paarweisen gravitativen Wechselwirkung zwischen den einzelnen Teilchen. In der numerischen Behandlung kommt es deshalb darauf an, jede dieser Zwei-Körper-Wechselwirkungen mit hoher Genauigkeit zu berücksichtigen.

Typische Teilchenzahlen der oben genannten Systeme sind $\sim 10$ für Planetensysteme, $10^4$ bis $10^6$ für Sternhaufen und bis über $10^8$ bei Galaxien. Für Teilchenzahlen dieser Größenordnung lassen sich die Methoden der statistischen Mechanik nur bedingt anwenden, so dass man auf die direkte Integration aller Teilchenbahnen unter dem Einfluss ihrer wechselseitigen Gravitationskräfte angewiesen ist.

**Abbildung 1.1:** Kugelsternhaufen M80 (credits: Hubble Space Telescope, NASA, 1999), 33 000 Lichtjahre von der Erde entfernt. Der Kugelsternhaufen enthält mehr als hunderttausend Sterne, sein maximaler Durchmesser beträgt 90 Lichtjahre. Er ist einer der dichtesten Kugelsternhaufen.

In dieser Übung sollen verschiedene numerische Verfahren zur Lösung gewöhnlicher Differentialgleichungen untersucht werden auf ihre Eignung zur Durchführung dieser Bahnintegration. Ein wesentliches Testbeispiel wird der Vergleich mit den bekannten Lösungen des Zwei-Körper-Problems darstellen, weitere Simulationen beschränken sich aus Rechenzeitgründen auf Teilchenzahlen von $N = 3$ bis maximal $N = 1000$.

Beim klassischen $N$-Körper-Problem werden die Bewegungen von $N$ Punktmassen in ihrem gemeinsamen Gravitationsfeld bestimmt. Jedes Teilchen $i$ mit der Masse $m_i$ habe zur Zeit $t$ die Position $\mathbf{r}_i$ und die Geschwindigkeit $\mathbf{v}_i$.

Die Hamilton-Funktion dieses Systems lautet

$$H = \sum_{i=1}^{N} \frac{\mathbf{p}_i{}^2}{2m_i} - \sum_{i=1}^{N} \sum_{j=i+1}^{N} \frac{G m_i m_j}{|\mathbf{q}_i - \mathbf{q}_j|} \tag{1.3}$$

mit den kanonisch konjugierten Variablen Impuls $\mathbf{p}_i = m_i \mathbf{v}_i$ und Ort $\mathbf{q}_i = \mathbf{r}_i$ für alle $i = 1 \ldots N$ Punktmassen. Das Hamiltonsche Prinzip liefert die Bewegungsgleichung für das Teilchen $i$ mit $k = 1, 2, 3$ für jede Raumrichtung,

$$\dot{q}_{i,k}(t) = \frac{\partial H}{\partial p_{i,k}} \quad \Longrightarrow \quad \frac{\mathrm{d}\mathbf{r}_i}{\mathrm{d}t} = \mathbf{v}_i, \tag{1.4}$$

$$\dot{p}_{i,k}(t) = -\frac{\partial H}{\partial q_{i,k}} \quad \Longrightarrow \quad \frac{\mathrm{d}\mathbf{v}_i}{\mathrm{d}t} = \mathbf{a}_i \tag{1.5}$$

mit der Beschleunigung

$$\mathbf{a}_i(t) = \sum_{j \neq i}^{N} G m_j \frac{\mathbf{r}_{ij}}{r_{ij}^3}. \tag{1.6}$$

Es wird später auch die Zeitableitung der Beschleunigung benötigt,

$$\dot{\mathbf{a}}_i(t) = \sum_{j \neq i}^{N} G m_j \left( \frac{\mathbf{v}_{ij}}{r_{ij}^3} - \frac{3(\mathbf{v}_{ij} \cdot \mathbf{r}_{ij})}{r_{ij}^5} \mathbf{r}_{ij} \right) \tag{1.7}$$

wobei $\mathbf{r}_{ij} := \mathbf{r}_j(t) - \mathbf{r}_i(t)$, $r_{ij} := |\mathbf{r}_{ij}|$, $\mathbf{v}_{ij} := \mathbf{v}_j(t) - \mathbf{v}_i(t)$, $v_{ij} := |\mathbf{v}_{ij}|$.

Zur Berechnung aller Beschleunigungen zu einem gegebenen Zeitpunkt sind also (unter Ausnutzung der Symmetrie der Kräfte) $N\,(N-1)/2$ Summanden auszuwerten, d.h. der Rechenaufwand für große $N$ verhält sich asymptotisch wie $N^2$.

### 1.2.1 Das Zwei-Körper-Problem

Die Hamilton-Funktion (1.3) erlaubt für das Zwei-Körper-Problem eine Separierung in Schwerpunkts- und Relativbewegung, d.h.

$$H = H_{\mathrm{cm}} + H_{\mathrm{rel}} = \frac{\mathbf{p}_{\mathrm{cm}}^2}{2M} + \frac{\mathbf{p}^2}{2\mu} - \frac{G m_1 m_2}{r} \tag{1.8}$$

mit der Gesamtmasse $M := m_1 + m_2$, der reduzierten Masse $\mu := m_1 m_2 / M$, dem Abstandsvektor $\mathbf{r} := \mathbf{r}_1 - \mathbf{r}_2$, dem Abstand $r := |\mathbf{r}|$, dem Relativimpuls $\mathbf{p} := \mathbf{p}_1 - \mathbf{p}_2$ und dem Schwerpunktsimpuls $\mathbf{p}_{\mathrm{cm}}$.

Die Hamilton-Funktion der Schwerpunktsbewegung $H_{\mathrm{cm}}$, d.h. der erste Term auf der rechten Seite von Gleichung (1.8), besitzt triviale Lösungen der zugehörigen kanonischen Gleichungen. Dies folgt aus der Tatsache, dass $H_{\mathrm{cm}}$ nur vom (Schwerpunkt-) Impuls $\mathbf{p}_{\mathrm{cm}}$ abhängt und nicht von der dazu konjugierten Variable $\mathbf{r}_{\mathrm{cm}}$, der (Schwerpunkt-) Koordinate. Es folgt eine gleichförmige Schwerpunktsbewegung.

Für die Hamiltonfunktion der Relativbewegung $H_{\mathrm{rel}}$ ergibt sich die Bewegungsgleichung des klassischen Zwei-Körper-Problems

$$\dot{\mathbf{p}} = \mu \dot{\mathbf{v}} = -\frac{G m_1 m_2}{r^3} \mathbf{r} \tag{1.9}$$

mit $\mathbf{v} = \mathbf{v}_1 - \mathbf{v}_2 = \mathbf{p}/\mu$, oder schließlich in der gewohnten Notation

$$\ddot{\mathbf{r}} = \dot{\mathbf{v}} = -\frac{GM}{r^3} \mathbf{r}. \tag{1.10}$$

Die Lösung des Kepler-Problems stellen Kegelschnitte mit der Bahnebene dar, die Bewegung von $\mathbf{r}$ beschreibt also eine Ellipse, eine Parabel oder eine Hyperbel. Neben der Energie $E$ sind der spezifische Drehimpuls

$$\mathbf{j} = \mathbf{r} \times \mathbf{v} \tag{1.11}$$

und der Runge-Lenz-Vektor

$$\mathbf{e} = \frac{\mathbf{v} \times \mathbf{j}}{GM} - \frac{\mathbf{r}}{r} \tag{1.12}$$

Konstanten der Zwei-Körper-Bewegung. Mit (1.10) gilt für die Zeitableitung des Drehimpulses

$$\frac{\mathrm{d}\mathbf{j}}{\mathrm{d}t} = \mathbf{v} \times \mathbf{v} + \mathbf{r} \times \dot{\mathbf{v}} = -\frac{GM}{r^3}(\mathbf{r} \times \mathbf{r}) = 0. \tag{1.13}$$

Für den Beweis der Konstanz von $\mathbf{e}$ ist eine kurze Zwischenrechnung unter Verwendung der Vektoridentität $(\mathbf{a} \times \mathbf{b}) \times \mathbf{c} = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{a}(\mathbf{b} \cdot \mathbf{c})$ hilfreich:

$$\frac{\mathbf{j} \times \mathbf{r}}{r^3} = \frac{(\mathbf{r} \times \mathbf{v}) \times \mathbf{r}}{r^3} = \frac{\mathbf{v}}{r} - \mathbf{r}\frac{(\mathbf{r} \cdot \mathbf{v})}{r^3} = \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathbf{r}}{r}\right). \tag{1.14}$$

Damit, mit (1.10) und mit (1.13) erhält man

$$\frac{\mathrm{d}\mathbf{e}}{\mathrm{d}t} = -\frac{GM}{r^3}\frac{\mathbf{r} \times \mathbf{j}}{GM} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathbf{r}}{r}\right) = \frac{\mathbf{j} \times \mathbf{r}}{r^3} - \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{\mathbf{r}}{r}\right) = 0. \tag{1.15}$$

Wir betrachten noch den Ausdruck

$$\mathbf{r} \cdot \mathbf{e} + r = \frac{\mathbf{r} \cdot (\mathbf{v} \times \mathbf{j})}{GM} = \frac{(\mathbf{r} \times \mathbf{v}) \cdot \mathbf{j}}{GM} = \frac{j^2}{GM}. \tag{1.16}$$

Das Skalarprodukt $\mathbf{r} \cdot \mathbf{e}$ wird ausgedrückt durch den Winkel $\phi - \phi_0$ zwischen $\mathbf{e}$ und dem Positionsvektor $\mathbf{r}$, so dass $re\cos(\phi - \phi_0) + r = j^2/(GM)$ und somit

$$r(\phi) = \frac{j^2/(GM)}{1 + e\cos(\phi - \phi_0)} \tag{1.17}$$

gilt, was die bekannte Gleichung eines Kegelschnittes darstellt. Für $0 \leq e < 1$ liegt eine gebundene Bewegung in Form einer Ellipse vor und der Betrag $e = |\mathbf{e}|$ des Runge-Lenz-Vektors entspricht der Exzentrizität der Ellipsenbahn an. Dann betragen die Maximal- und Minimaldistanz[1] der beiden Punktmassen

$$r_{\max\,/\,\min} = \frac{j^2/(GM)}{1 \pm e}, \tag{1.18}$$

und die große Halbachse der Bahnellipse ergibt sich zu

$$a = \frac{1}{2}(r_{\min} + r_{\max}) = \frac{j^2/(GM)}{1 - e^2}. \tag{1.19}$$

Eine schematische Darstellung der Bewegung im Falle einer gebundenen Bahn ist in Abbildung 1.2 zu sehen. Gemäß des dritten Keplerschen Gesetzes („Die Quadrate der Umlaufzeiten zweier Planeten verhalten sich wie die Kuben der großen Halbachsen ihrer Ellipsenbahnen“) kann damit die Umlauffrequenz

$$\omega = \sqrt{\frac{GM}{a^3}} \tag{1.20}$$

berechnet werden.

[1] Der Abstand zur Apoapsis und zum Periapsis.

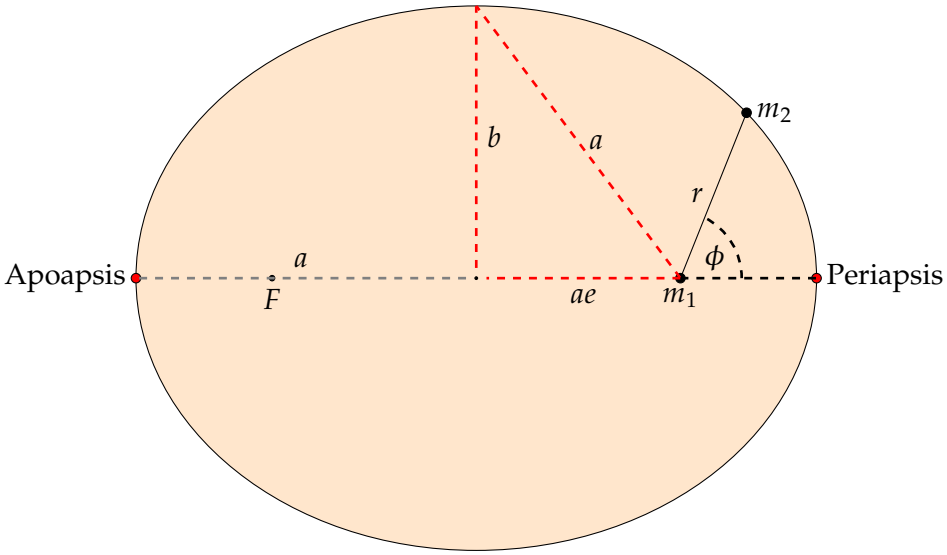



**Abbildung 1.2:** Gebundene Bahn im Zwei-Körperproblem. Die Masse $m_1$ befindet sich im Brennpunkt der Ellipse mit den Halbachsen $a$ und $b$ und der Exzentrizität $e$. Der Abstand von $m_1$ zur Apoapsis ist $r_{\text{max}}$ und zur Periapsis ist $r_{\text{min}}$. Die Bahngleichung ist gegeben durch Gleichung 1.17, wobei das Koordinatensystem so gewählt wurde, dass $\phi_0 = 0$ ist.

## 1.3 Numerische Lösung der Bewegungsgleichung: Zeitintegratoren

Um die Bahnen der Punktmassen zu berechnen, muss für jedes Teilchen $i$ die zugehörige Bewegungsgleichung, also das System gewöhnlicher Differentialgleichungen, das von (1.4) und (1.5) gebildet wird, mit entsprechenden Anfangswerten für Orte und Geschwindigkeiten gelöst werden. Dies ist in der Regel nur numerisch möglich, mit der Folge, dass man die Teilchenverteilung nur zu diskreten Zeiten bestimmen kann. Ausgehend von dem aktuellen Zeitpunkt $t = t_n$ wird der Ort und die Geschwindigkeit zu einem späteren Zeitpunkt $t_{n+1} = t_n + \Delta t$ berechnet, der dann neue aktuelle Zeit wird, von der aus der nächste Zeitschritt ausgeführt wird, und so fort.

Im Folgenden werden verschiedene numerische Verfahren dafür vorgestellt, die hier aus nahe liegenden Gründen Zeitintegratoren genannt werden, die sich aber allgemein zur Lösung eines Systems gewöhnlicher Differentialgleichungen der Form $\mathrm{d}y/\mathrm{d}x = f(x, y)$ mit Anfangswertbedingung eignen. Die im Weiteren auftretenden Indizes beziehen sich in der Regel auf den zugehörigen Zeitpunkt, z.B. bedeutet $r_n = r(t_n)$ etc. Zur Vereinfachung der Darstellung werden im folgenden nur Skalare betrachtet und der Teilchen-Index $i$ weggelassen.

### 1.3.1 Einschrittverfahren

Grundlegende Idee: Betrachte die Differentiale $\mathrm{d}y$ und $\mathrm{d}x$ als endliche Intervalle $\Delta y$ und $\Delta x$

$$\frac{\mathrm{d}y}{\mathrm{d}x} \Rightarrow \frac{\Delta y}{\Delta x} = f(x, y)$$

und diskretisiere mit $\Delta x = h$

$$\Delta y = y_{n+1} - y_n = \Delta x f = hf.$$

Allgemein können Einschrittverfahren in folgender Form geschrieben werden

$$y_{n+1} = y_n + h\Phi(x_n, y_n, y_{n+1}, h),$$

hierbei nennt man $\Phi$ die Inkrementfunktion. Das Verfahren ist explizit falls $\Phi = \Phi(x_n, y_n, h)$ und implizit falls $\Phi = \Phi(x_n, y_n, y_{n+1}, h)$. Man schreitet somit von $x_n$ nach $x_{n+1}$ und von $y_n$ nach $y_{n+1}$.

Einschrittverfahren können durch die Taylorentwicklung konstruiert werden. Als erstes, einfaches Einschrittverfahren betrachten wir das explizite Eulerverfahren mit $\Phi(x_n, y_n, h) = f(x_n, y_n)$.

### 1.3.2 Einfache Zeitintegratoren: Das Euler-Verfahren und die Euler-Cromer-Methode

Die einfachste Möglichkeit, eine Zeitintegration durchzuführen, ergibt sich aus der Taylor-Entwicklung von Geschwindigkeit und Ort gemäß

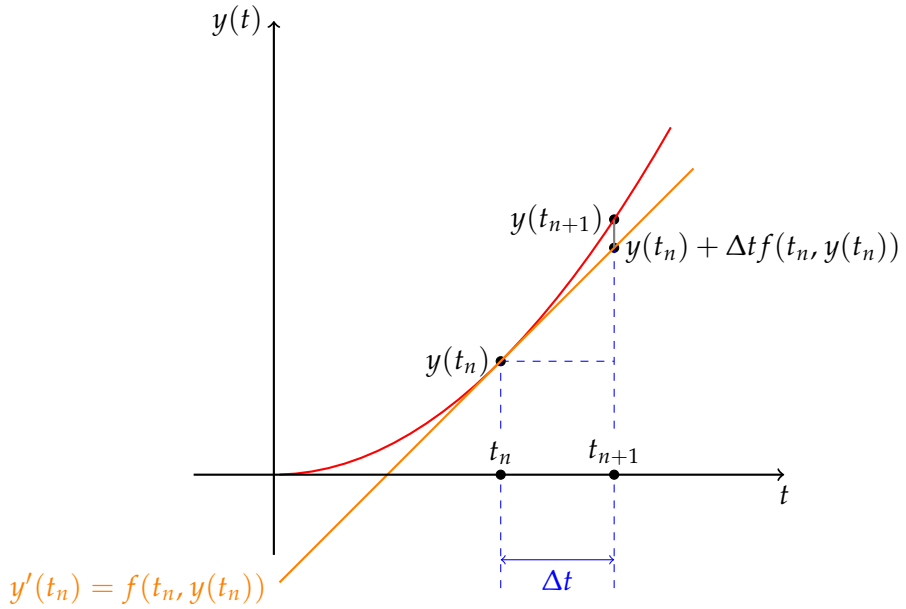
$$\begin{aligned} v(t_n + \Delta t) &= v(t_n) + \frac{\mathrm{d}v}{\mathrm{d}t}(t_n)\Delta t + \mathcal{O}(\Delta t^2), && (1.21)\\ &= v(t_n) + a(t_n)\Delta t + \mathcal{O}(\Delta t^2), && (1.22)\\ r(t_n + \Delta t) &= r(t_n) + \frac{\mathrm{d}r}{\mathrm{d}t}(t_n)\Delta t + \mathcal{O}(\Delta t^2), && (1.23)\\ &= r(t_n) + v(t_n)\Delta t + \mathcal{O}(\Delta t^2), && (1.24) \end{aligned}$$

wobei die Beschleunigung $a$ gemäß (1.6) bestimmt wird. Dies lässt sich unmittelbar für einen Algorithmus verwenden, dem so genannten Euler-Verfahren, wobei in jedem Zeitschritt die Berechnungen

$$\begin{aligned} v_{n+1} &= v_n + a_n\Delta t, && (1.25)\\ r_{n+1} &= r_n + v_n\Delta t && (1.26) \end{aligned}$$

durchgeführt werden.

Anschaulich wird beim Euler-Verfahren der Vorwärtsschritt mit Hilfe das einfachen Steigungsdreiecks durchgeführt:




D.h. zum Zeitpunkt $t_n$ wird die Änderung der Funktion $y(t)$ von $t_n$ nach $t_{n+1} = t_n + \Delta t$ durch $\Delta t y'(t_n) = \Delta t f(t_n, y(t_n))$ angenähert. Der Diskretisierungsfehler ist durch die Differenz $y(t_{n+1}) - y(t_n)$ gegeben.

Da beim Euler-Verfahren zuerst die Geschwindigkeit für den neuen Zeitpunkt berechnet wird, ergibt sich als Alternative die Euler-Cromer-Methode, bei der zur Berechnung des neuen Ortes nicht die Geschwindigkeit zum alten Zeitpunkt $t_n$ sondern zum neuen Zeitpunkt $t_{n+1}$ verwendet wird, also

$$v_{n+1} = v_n + a_n \Delta t, \tag{1.27}$$
$$r_{n+1} = r_n + v_{n+1} \Delta t. \tag{1.28}$$

Mitteln dieser beiden Verfahren führt zur Mittelungs-Methode

$$v_{n+1} = v_n + a_n \Delta t, \tag{1.29}$$
$$r_{n+1} = r_n + \frac{1}{2}(v_n + v_{n+1}) \Delta t. \tag{1.30}$$

#### Einschub zur Verfahrensgenauigkeit

Der Nachteil dieser Verfahren besteht darin, dass ihre Genauigkeit verhältnismäßig gering ist. Sollen etwa die Teilchenbahnen für einen Zeitraum $t_{\max}$ bestimmt werden, wobei jeder Zeitschritt die Länge $\Delta t$ haben soll, so sind $N_t := t_{\max}/\Delta t$ Zeitschritte notwendig. Die Genauigkeit pro Zeitschritt (lokaler Fehler) ist von der Ordnung $\mathcal{O}(\Delta t^2)$ (siehe die Taylor-Entwicklung), so dass der Gesamtfehler (globaler Fehler) der Simulation von der Ordnung $\mathcal{O}(N_t \Delta t^2) = \mathcal{O}(\Delta t)$ ist. Hierbei sind Rundungsfehler durch begrenzte Rechengenauigkeit der Computer noch nicht berücksichtigt.

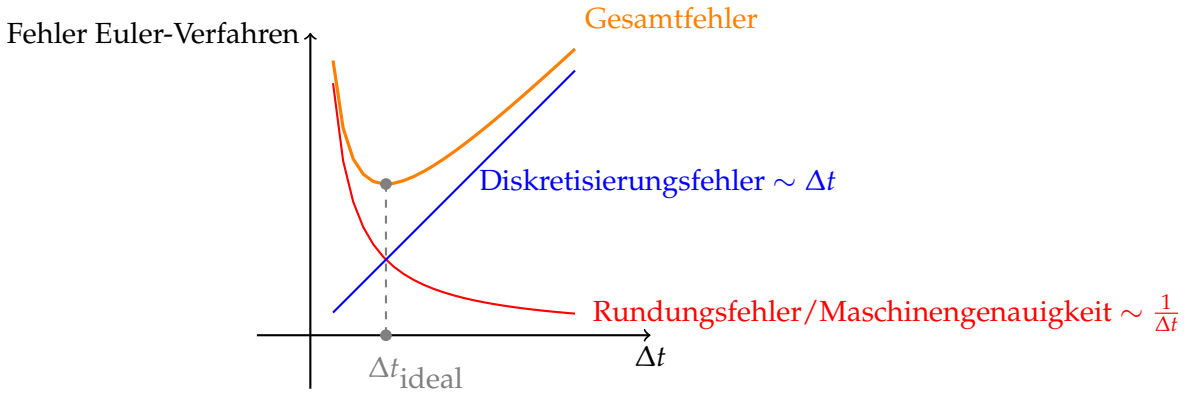
#### Einschub zur Rechengenauigkeit

Wenn $\epsilon$ die Maschinengenauigkeit des Computers ist, so wächst der Fehler aufgrund der Rechenungenauigkeit mit $\mathcal{O}(N_t \epsilon)$. Eine Verbesserung der Verfahrensgenauigkeit durch kleinere

Zeitschritte $\Delta t$ wird also mit einem größeren Rundungsfehler aufgrund der größeren Anzahl $N_t$ an Zeitschritten erkauft. Bei einer $N$-Körper-Simulation besitzen globale Werte der Simulationsresultate wie z.B. die Gesamtenergie $E$ also einen Fehler von $\mathcal{O}(N_t N\epsilon)$.

Ein Zahlenbeispiel zur Verdeutlichung: Beim Zwei-Körper-Problem sind je nach numerischem Verfahren ca. 100 bis 1000 Schritte pro Orbit notwendig, um pro Zeitschritt einen Fehler in der Energie in der Größenordnung der Maschinengenauigkeit einhalten zu können, also $\Delta E/E \sim 10^{-13}$. Damit summiert sich der Fehler in der Energie für einen gesamten Umlauf auf $\Delta E/E \sim 10^{-11} - 10^{-10}$. Zur Simulation der Entwicklung eines Kugelsternhaufens mit $10^6$ Sternen und einem Alter von $10^4$ typischen Umlaufszeiten müssten entsprechend $10^{10}$ Orbits mit je 1000 Zeitschritten, insgesamt also $10^{13}$ Zeitschritte durchgeführt werden. Damit würde der Fehler der Gesamtenergie am Ende $\Delta E \sim 100\,\%$ betragen.

Schematisch ergibt sich folgendes Bild




### 1.3.3 Das Leap-Frog-Verfahren

Eine Verbesserung des Euler-Verfahrens besteht darin, die Geschwindigkeit für die Berechnung des nächsten Ortes nicht zu den Zeiten $t_n$ oder $t_{n+1}$ zu nehmen, sondern zum Zeitpunkt $t_{n+1/2} = t_n + \Delta t/2$. Das Schema dieses Verfahrens lautet

Anfangszeitschritt:

$$\begin{aligned} r_{1/2} &= r_0 + v_0 \frac{\Delta t}{2}, \\ & \text{Berechne } a_{1/2} = a(t_{1/2}, r_{1/2}), \end{aligned}$$

Reguläre Zeitschritte:

$$\begin{aligned} v_{n+1} &= v_n + a_{n+1/2}\,\Delta t, \\ r_{n+3/2} &= r_{n+1/2} + v_{n+1}\,\Delta t, \end{aligned}$$

Letzter Zeitschritt:

$$r_{n+1} = r_{n+1/2} + v_{n+1}\frac{\Delta t}{2}.$$

Der Name "Leap Frog" für diesen Zeitintegrator rührt von der um jeweils eine halbe Zeitschritteinheit $\Delta t$ voneinander entfernten Berechnung des Ortes und der Geschwindigkeit her, die Berechnungszeitpunkte „springen" auf der Zeitachse jeweils übereinander hinweg. Ort und Geschwindigkeit sind also während der Rechnung nicht gleichzeitig bekannt. Falls die

Geschwindigkeit zum selben Zeitpunkt bestimmt werden soll, kann diese nachträglich über $v_{n+1/2} = (v_n + v_{n+1})/2$ berechnet werden.

Im Grenzübergang $\Delta t \to 0$ erhält man die üblichen kanonischen Gleichungen, und für endliches $\Delta t$ ist dieser Integrator symplektisch, d.h. es gibt eine Hamilton-Funktion $\hat{H} = H + \Delta t^2 H_2 + \Delta t^4 H_4 + \ldots$, für welche die "Leap Frog"-Lösung korrekt ist. Das wirkt sich positiv auf die Konstanz der Erhaltungsgrößen aus.

### 1.3.4 Der Verlet- und Velocity-Verlet-Algorithmus

Ein Lösungsverfahren höherer Genauigkeit ergibt sich durch Taylor-Entwicklung höherer Ordnung. Ausgehend von

$$r(t_n + \Delta t) = r(t_n) + v(t_n)\Delta t + \frac{a(t_n)}{2}\Delta t^2 + \mathcal{O}(\Delta t^3) \quad \text{und} \tag{1.31}$$

$$r(t_n - \Delta t) = r(t_n) - v(t_n)\Delta t + \frac{a(t_n)}{2}\Delta t^2 + \mathcal{O}(\Delta t^3) \tag{1.32}$$

folgt durch Addition und Subtraktion und geeignete Umformung das Schema

$$r_{n+1} = 2r_n - r_{n-1} + a_n \Delta t^2 \quad ( +\mathcal{O}(\Delta t^4) ), \tag{1.33}$$

$$v_n = \frac{r_{n+1} - r_{n-1}}{2\Delta t} \quad ( +\mathcal{O}(\Delta t^2) ). \tag{1.34}$$

Der Vorteil dieser so genannten Verlet-Methode besteht in der sehr großen Genauigkeit bei der Ortsberechnung. Diese hängt nicht von der Geschwindigkeit ab, allerdings darf in dem angegebenen Schema auch die Beschleunigung nicht von der Geschwindigkeit abhängen, da letztere immer erst nachträglich für den jeweils vorausgehenden Zeitpunkt bestimmt wird. Außerdem ist die Geschwindigkeitsberechnung zwei Größenordnungen ungenauer als die Ortsberechnung, so dass dieses Verfahren insgesamt ideal ist für Anwendungen, bei denen im Wesentlichen nur die Teilchenbahnen von Interesse sind.

Ein weiterer Nachteil besteht darin, dass zu Beginn der Rechnung die Anfangswerte der Teilchenpositionen zu zwei Zeitpunkten gegeben sein müssen, das sind $r_0$ und $r_{-1}$, wovon in der Regel nur der erste Wert vorliegt. Nach (1.32) kann ein Wert für $r_{-1}$ aber gemäß

$$r_{-1} = r_0 - v_0 \Delta t + \frac{a_0}{2}\Delta t^2 \tag{1.35}$$

gefunden werden.

Am Ende der Simulation muss entsprechend über das eigentliche Zeitintervall $t_{\max}$ hinaus bis zur Zeit $t_{\max} + \Delta t$ gerechnet werden, um auch die Geschwindigkeit $v(t_{\max})$ bestimmen zu können.

In der Praxis wird meistens eine davon abgeleitete Methode verwendet, der sogenannte Velocity-Verlet Algorithmus, der sehr ähnlich zum Leap-Frog ist mit dem ausgesprochenen Vorteil, dass Geschwindigkeit und Orte zum selben Zeitpunkt bestimmt werden. Das Schema lautet

$$r_{n+1} = r_n + v_n \Delta t + \frac{1}{2} a_n \Delta t^2, \tag{1.36}$$

$$v_{n+1} = v_n + \frac{1}{2}\left(a_n + a_{n+1}\right)\Delta t. \tag{1.37}$$

Aus Stabilitätsgründen muss hierbei ein konstanter Zeitschritt $\Delta t$ gewählt werden. Für den Fall variabler Zeitschritte hat sich die *kick-drift-kick* Variante etabliert, die folgendermaßen lautet

$$v_{n+1/2} = v_n + \frac{1}{2} a_n \Delta t, \tag{1.38}$$

$$r_{n+1} = r_n + v_{n+1/2} \Delta t, \tag{1.39}$$

$$v_{n+1} = v_{n+1/2} + \frac{1}{2} a_{n+1} \Delta t. \tag{1.40}$$

### 1.3.5 Das Hermite-Schema
### Predictor-Corrector-Integratoren

Ein weiteres Verfahren höherer Genauigkeit erhält man mit dem so genannten Hermite-Schema. Im Gegensatz zu Einschrittverfahren wie das o.g. Euler-Verfahren oder die später betrachteten Runge-Kutta-Verfahren, nutzen Mehrschrittverfahren die Information aus den zuvor bereits errechneten Stützpunkten (predicted values) und korrigieren dann in einem Korrekturschritt (corrected values). In einem ersten Schritt, der sog. "Prediction", berechnet man Orte und Geschwindigkeiten (wieder aus einer Taylor-Entwicklung) zu

$$v^p_{n+1} = v_n + a_n \Delta t + \frac{1}{2} \dot{a}_n \Delta t^2, \tag{1.41}$$

$$r^p_{n+1} = r_n + v_n \Delta t + \frac{1}{2} a_n \Delta t^2 + \frac{1}{6} \dot{a}_n \Delta t^3, \tag{1.42}$$

($p$ steht für "predicted"), wobei die Beschleunigung $a_n$ und ihre Zeitableitung $\dot{a}_n$ gemäß (1.6) und (1.7) aus den Orten $r_n$ und den Geschwindigkeiten $v_n$ zur Zeit $t_n$ berechnet werden.

Mit Hilfe der so gewonnenen Positionen $r^p_{n+1}$ und Geschwindigkeiten $v^p_{n+1}$ können nun für alle Teilchen über (1.6) und (1.7) die Beschleunigung und ihre ersten Ableitung zum Zeitpunkt $t_{n+1}$ bestimmt werden:

$$a^p_{n+1} = a(t_{n+1}, r^p_{n+1}) \qquad \text{und} \qquad \dot{a}^p_{n+1} = \dot{a}(t_{n+1}, r^p_{n+1}, v^p_{n+1}). \tag{1.43}$$

Ebenso könnte man $a_{n+1}$ und $\dot{a}_{n+1}$ aus einer Taylor-Reihe berechnen:

$$a_{n+1} = a_n + \dot{a}_n \Delta t + \frac{1}{2} a^{(2)}_n \Delta t^2 + \frac{1}{6} a^{(3)}_n \Delta t^3, \tag{1.44}$$

$$\dot{a}_{n+1} = \dot{a}_n + a^{(2)}_n \Delta t + \frac{1}{2} a^{(3)}_n \Delta t^2. \tag{1.45}$$

Durch Gleichsetzen der beiden Ansätze und Auflösen erhält man die sog. Hermite-Interpolation für die zweite und dritte Ableitung der Beschleunigung:

$$\frac{1}{2} a^{(2)}_n = -3 \frac{a_n - a^p_{n+1}}{\Delta t^2} - \frac{2\dot{a}_n + \dot{a}^p_{n+1}}{\Delta t}, \tag{1.46}$$

$$\frac{1}{6} a^{(3)}_n = 2 \frac{a_n - a^p_{n+1}}{\Delta t^3} + \frac{\dot{a}_n + \dot{a}^p_{n+1}}{\Delta t^2}. \tag{1.47}$$

Im abschließenden "Correction"-Schritt können damit die im "Prediction"-Schritt gewonnenen Werte für die Orte und Geschwindigkeiten auf eine höhere Ordnung korrigiert werden zu

$$v^c_{n+1} = v^p_{n+1} + \frac{1}{6} a^{(2)}_n \Delta t^3 + \frac{1}{24} a^{(3)}_n \Delta t^4, \tag{1.48}$$

$$r^c_{n+1} = r^p_{n+1} + \frac{1}{24} a^{(2)}_n \Delta t^4 + \frac{1}{120} a^{(3)}_n \Delta t^5 \tag{1.49}$$

(mit $c$ für "corrected").

Zur Vermeidung von Asymmetrien der paarweisen Kräfte muss darauf geachtet werden, zunächst $r^c_{n+1}$ und $v^c_{n+1}$ für alle Teilchen zu berechnen, bevor die Positionen und Geschwindigkeiten gemäß $r_{n+1} = r^c_{n+1}$ und $v_{n+1} = v^c_{n+1}$ für den nächsten Zeitschritt aktualisiert werden. Nachdem der Vorgang für alle Teilchen durchgeführt wurde, kann ein neuer Zeitschritt mit der Berechnung der Beschleunigungen und ihrer Zeitableitungen beginnen.

Innerhalb dieses Verfahrens ist es leicht möglich, individuelle Zeitschritte einzuführen, d.h. die Korrektur an einem gegebenen Zeitpunkt nur für eine Teilmenge der Teilchen durchzuführen. Das Verfahren ist jedoch nicht symplektisch, d.h. seine Lösungen sind nicht darstellbar als Lösungen der kanonischen Gleichungen einer Hamilton-Funktion, die sich nur leicht von der des gegebenen Problems unterscheidet. Dadurch weist dieses Verfahren im Allgemeinen immer eine leichte Drift der Erhaltungsgrößen wie der Energie auf.

Gegenüber Methoden wie dem Leap Frog muss die Summe über alle Teilchen hier zweimal durchgeführt werden, da sowohl $a$ als auch $\dot{a}$ berechnet werden müssen, was besonders bei größeren Teilchenzahlen schnell wachsenden Rechenzeitbedarf bedeutet.

#### Das iterierte zeitumkehrbare Hermite-Verfahren

Speziell für die Anwendung bei $N$-Körper-Simulationen hat Makino (1997) erkannt, dass das Hermite-Verfahren durch den Verzicht auf den höchsten Term der Korrektur von $r^c_{n+1}$ und durch eine iterierbare, zeitumkehrbare Form der Korrektur verbessert werden kann. Nach Weglassen des Terms höchster Ordnung für $r^c_{n+1}$ in Gleichung (1.49) erhält man durch Einsetzen von (1.41), (1.42), (1.46) und (1.47) in (1.48) und (1.49) und durch geeignetes Umformen

$$v^c_{n+1} = v_n + \frac{1}{2}\left(a^p_{n+1} + a_n\right)\Delta t + \frac{1}{12}\left(\dot{a}^p_{n+1} - \dot{a}_n\right)\Delta t^2, \tag{1.50}$$

$$r^c_{n+1} = r_n + \frac{1}{2}\left(v^c_{n+1} + v_n\right)\Delta t + \frac{1}{12}\left(a^p_{n+1} - a_n\right)\Delta t^2. \tag{1.51}$$

Trotz der Übereinstimmung der Formeln läuft dieses Verfahren algorithmisch aber völlig anders ab als das normale Hermite-Verfahren (selbst wenn man dieses iterieren würde): Zuerst wird $v^c_{n+1}$ berechnet und dann noch im gleichen Schritt in die Berechnung von $r^c_{n+1}$ eingesetzt. Dadurch wird die Konvergenz beschleunigt, aber auch eine Zeitsymmetrie des Algorithmus erreicht (wie sie z.B. auch beim Leap Frog vorhanden ist, nicht aber beim normalen Hermite-Verfahren).

Die Iteration findet dadurch statt, dass mit den neuen Werten für $r^c_{n+1}$ und $v^c_{n+1}$ auch die Größen $a^p_{n+1}$ und $\dot{a}^p_{n+1}$ erneut berechnet werden können und damit wiederum neue Werte für $r^c_{n+1}$ und $v^c_{n+1}$ bestimmt werden. Schon nach zwei Iterationen zeigt sich eine erhebliche Verbesserung der Eigenschaften des Integrators, vor allem für kleine Teilchenzahlen. Der numerische Aufwand steigt allerdings durch jede Iteration, da jeweils eine zusätzliche Berechnung von $a^p_{n+1}$ und $\dot{a}^p_{n+1}$ für alle Teilchen notwendig ist. Im Allgemeinen genügen jedoch zwei Iterationen.

Die Berechnung von $a^{(2)}_n$ und $a^{(3)}_n$ gemäß der Hermite-Interpolation (1.46) und (1.47) ist streng genommen nicht mehr nötig. Da jedoch bei hohen Teilchenzahlen der Aufwand dafür gering ist, werden diese gerne zusätzlich bestimmt, um eine bessere Anpassungsformel für den Zeitschritt (siehe Aufgaben, Gleichung (1.78)) verwenden zu können.

### 1.3.6 Runge-Kutta Integratoren

Für eine Diskussion der populären Runge-Kutta Integratoren wird auf die Literatur verwiesen, z.B. auf Kapitel 16.1 der alten Dokumentation zur Programm-Bibliothek Numerical Recipes (Kapitel 17.1 der aktuellen Version). Die alte Dokumentation ist online frei verfügbar auf `http://numerical.recipes/` unter dem Punkt **Obsolete Versions**. Ein guter Einstieg bietet auch die englischsprachige Wikipediaseite[2]. Runge-Kutta-Integratoren sind in der Regel fester Bestandteil von Programmbibliotheken zur numerischen Lösungen von Differentialgleichungen, und finden sich in `scipy` oder `MATLAB`. Die GSL (GNU Scientific Library)[3] zum Beispiel bietet acht verschiedene Runge-Kutta Integratoren, explizit und implizit, von zweiter bis 8. Ordnung.

Runge-Kutta-Integratoren eignen sich zur Berechnung beliebiger gewöhnlicher Differentialgleichungen (sofern diese nicht zu „steif" sind, d.h. alle auftretenden Zeitskalen von ähnlicher Größenordnung sind), also zur numerischen Lösung einer Gleichung der Form

$$\frac{\mathrm{d}y}{\mathrm{d}x} = f(x, y), \tag{1.52}$$

wobei $y$ der Vektor der Größen darstellt, die bezüglich der Variable $x$ integriert werden sollen, und $f$ als „rechte Seite" der Gleichung eine beliebige Funktion von $x$ und $y$ sein kann. Für den Fall, dass Gleichung (1.52) eine Bewegungsgleichung ist, gilt entsprechend

$$x = t, \quad y = \begin{pmatrix} \mathbf{r}_i \\ \mathbf{v}_i \end{pmatrix}, \quad f = \begin{pmatrix} \mathbf{v}_i \\ \mathbf{a}_i \end{pmatrix}. \tag{1.53}$$

Im Prinzip kann das in Abschnitt 1.3.2 vorgestellte Euler-Verfahren (1.25) auch als ein (sehr einfaches) Runge-Kutta-Verfahren aufgefasst werden:

$$y_{n+1} = y_n + \Delta t\, f(t_n, y_n). \tag{1.54}$$

Im Allgemeinen sind Runge-Kutta-Verfahren aber Einschrittverfahren, bei denen ein Zeitschritt in Zwischenschritte unterteilt wird, was die Berechnung der Funktion $f$ an zusätzlichen Zeitpunkten zwischen $t_n$ und $t_{n+1}$ erfordert. Im Folgenden werden zwei Beispiele vorgestellt.

#### Das Halbschritt- bzw. Mittelpunktsverfahren (Runge-Kutta 2. Ordnung)

Schema:

$$\begin{aligned} k_1 &= \Delta t\, f(t_n, y_n), && (1.55)\\ k_2 &= \Delta t\, f(t_n + \tfrac{1}{2}\Delta t, y_n + \tfrac{1}{2}k_1), && (1.56)\\ y_{n+1} &= y_n + k_2 \quad ( +\mathcal{O}(\Delta t^3) ). && (1.57) \end{aligned}$$

Hierbei sind $k_1$ und $k_2$ temporäre Zwischengrößen.

[2] `https://en.wikipedia.org/wiki/Runge-Kutta_methods`
[3] `https://www.gnu.org/software/gsl/`

### Das Heun-Verfahren (Runge-Kutta 2. Ordnung)

Schema:

$$
\begin{aligned}
k_1 &= \Delta t\, f(t_n, y_n), && (1.58)\\
k_2 &= \Delta t\, f(t_n + \Delta t, y_n + k_1), && (1.59)\\
y_{n+1} &= y_n + \frac{1}{2}(k_1 + k_2) \quad (+\mathcal{O}(\Delta t^3)). && (1.60)
\end{aligned}
$$

Hierbei sind $k_1$ und $k_2$ temporäre Zwischengrößen.

### Das klassische Runge-Kutta-Verfahren 4. Ordnung

Schema:

$$
\begin{aligned}
k_1 &= \Delta t\, f(t_n, y_n), && (1.61)\\
k_2 &= \Delta t\, f(t_n + \tfrac{1}{2}\Delta t, y_n + \tfrac{1}{2}k_1), && (1.62)\\
k_3 &= \Delta t\, f(t_n + \tfrac{1}{2}\Delta t, y_n + \tfrac{1}{2}k_2), && (1.63)\\
k_4 &= \Delta t\, f(t_n + \Delta t, y_n + k_3), && (1.64)\\
y_{n+1} &= y_n + \frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \quad (+\mathcal{O}(\Delta t^5)). && (1.65)
\end{aligned}
$$

Für den Spezialfall einer Bewegungsgleichung ergibt sich daraus

$$
\begin{aligned}
\tilde{v}_1 &= a_n \Delta t, && (1.66)\\
\tilde{r}_1 &= v_n \Delta t, && (1.67)\\
\tilde{v}_2 &= a(t_n + \tfrac{1}{2}\Delta t, r_n + \tfrac{1}{2}\tilde{r}_1)\Delta t, && (1.68)\\
\tilde{r}_2 &= (v_n + \tfrac{1}{2}\tilde{v}_1)\Delta t, && (1.69)\\
\tilde{v}_3 &= a(t_n + \tfrac{1}{2}\Delta t, r_n + \tfrac{1}{2}\tilde{r}_2)\Delta t, && (1.70)\\
\tilde{r}_3 &= (v_n + \tfrac{1}{2}\tilde{v}_2)\Delta t, && (1.71)\\
\tilde{v}_4 &= a(t_n + \Delta t, r_n + \tilde{r}_3)\Delta t, && (1.72)\\
\tilde{r}_4 &= (v_n + \tilde{v}_3)\Delta t, && (1.73)\\
v_{n+1} &= v_n + \frac{1}{6}\tilde{v}_1 + \frac{1}{3}\tilde{v}_2 + \frac{1}{3}\tilde{v}_3 + \frac{1}{6}\tilde{v}_4 \quad (+\mathcal{O}(\Delta t^5)). && (1.74)\\
r_{n+1} &= r_n + \frac{1}{6}\tilde{r}_1 + \frac{1}{3}\tilde{r}_2 + \frac{1}{3}\tilde{r}_3 + \frac{1}{6}\tilde{r}_4 \quad (+\mathcal{O}(\Delta t^5)). && (1.75)
\end{aligned}
$$

## 1.4 Literatur

**C.L. Siegel** Himmelsmechanik
**K. Stumpff** Himmelsmechanik I-III
**T. Pang** An Introduction to Computational Physics
**Press, Teukolsky, Vetterling, Flannery** Numerical Recipes
**Makino, J. et al.** Astrophysical Journal Vol. 480, p. 432 (1997)
**Makino, J., Aarseth, S.J.** Proc. Astron. Soc. Japan Vol. 44, p. 141 (1992)
**Aarseth, S.J.** Celest. Mechanics and Dyn. Astron. Vol. 73, p. 127 (1999)

## Aufgaben

Für die folgenden Aufgaben sollte $G = 1$ gesetzt und die Gesamtmasse des Systems jeweils immer auf $M = 1$ normiert werden. Dadurch hat eine typische System-Zeit (z.B. die Kreisbahn-Umlaufzeit an einem bestimmten Radius) immer den gleichen Wert unabhängig von der Teilchenzahl $N$. Außerdem sollen alle Positionen und Geschwindigkeiten nach der Einleseroutine zuerst in das Schwerpunktssystem transfomiert werden, bevor die Simulation gestartet wird, um die Drift des Schwerpunktes zu vermeiden.

#### Anfangsverteilungen der Punktmassen

Sie finden unter `http://www.tat.physik.uni-tuebingen.de/~schaefer/teach/cp.html` die relevanten Eingabedateien in zwei unterschiedlichen Formaten. Einerseits im folgenden Format

`in2, pla3, pl.100` und `pl.1k`

mit Muster-Inputdaten für Systeme mit 2, 3 sowie 100 und 1000 Teilchen (die letzten beiden wurden so erzeugt, dass die Dichteverteilung einem sog. Plummer-Modell entspricht, welches einen homogenen Kern und eine stark abfallende Dichte nach außen besitzt und Kugelsternhaufen beschreibt).

In der ersten Zeile jeder Datei stehen die Größen

$$N, t_{\max}, \eta$$

($N$: Teilchenzahl, $t_{\max}$: maximale Integrationszeit, $\eta$: Zeitschrittparameter). In den folgenden $3N$ Zeilen finden sich die Massen, Positionen, und Geschwindigkeiten der Teilchen.

Alternativ dazu finden Sie auch die Eingabefiles `2body, 3body, 100body, 1kbody` im einfachen Zeilenformat, wobei jede Zeile folgende Spalten enthält

$$x \quad y \quad z \quad v_x \quad v_y \quad v_z \quad m$$

Orte, Geschwindigkeiten, Masse. Die Teilchenzahl ist durch die Anzahl der Zeilen gegeben, die maximale Integrationszeit und der Zeitschrittparameter $\eta$ entnehmen Sie den o.g. Dateien.

### Teilaufgabe 1: Das Programm

Schreiben Sie ein $N$-Körper-Simulationsprogramm, das die obigen Inputdaten in einem der vorhandenen Formate einliest und nach direkter Kraftberechnung vorwärts integriert. Legen Sie das Programm am besten so an, dass sich die verschiedenen Zeitintegratoren austauschen lassen.

Bereiten Sie zur Kontrolle des jeweils verwendeten Zeitintegrators eine Berechnung der Erhaltungsgrößen Gesamtenergie, Gesamtimpuls, und Gesamtdrehimpuls zu jedem Zeitschritt vor.

Programmieren Sie die in der Vorlesung vorgestellten Zeitintegratoren (Euler, Euler-Cromer, Velocity-Verlet, Hermite, iterierter Hermite, Heun, RK4) in das Programm[4].

[4] Für das Runge-Kutta-Verfahren kann alternativ zur Implementierung des klassischen Runge-Kutta-Verfahrens 4. Ordnung auch versucht werden, keine eigene Unterroutine zu schreiben, sondern die in den Numerical Recipes vorgestellten Routinen `odeint, rkqs, rkck` zu verwenden.

### Bestimmung der Zeitschrittweite $\Delta t$

Es gibt immer die Möglichkeit, für die gesamte Simulation jeweils eine feste Zeitschrittweite zu verwenden, z.B.

$$\Delta t = \eta \qquad \text{oder, wenn das nicht ausreicht,} \qquad \Delta t = \eta^2. \tag{1.76}$$

Damit sollten auch die Tests von Teilaufgabe 2 durchgeführt werden.

Oft ist es jedoch besser, den Zeitschritt jeweils an die aktuellen Gegebenheiten während der Rechnung anzupassen. So kann die Länge des Zeitschrittes z.B. von der aktuellen Krümmung der Teilchenbahn abhängig gemacht werden. Dann wird der Zeitschritt gemäß

$$\Delta t = \eta \min_{i=1\ldots N} \left( \frac{|\mathbf{a}_i(t_n)|}{|\dot{\mathbf{a}}_i(t_n)|} \right) \tag{1.77}$$

aus den Beschleunigungen und ihren Zeitableitungen berechnet.

Auf diese Weise sollten auch die Zeitschritte im Programm bestimmt werden. Sofern das nicht möglich ist, kann die Bedingung (1.76) verwendet werden, z.B. für den Anfangs-Zeitschritt falls zu Beginn alle Geschwindigkeiten $\mathbf{v}_i = 0$ sind.

Für die Hermite-Schemata kann zusätzlich noch auf die höheren Ableitungen der Beschleunigungen aus dem vorhergehenden Zeitschritt zurückgegriffen werden. Hier lautet eine entsprechende Formel für die Zeitschrittweite

$$\Delta t = \eta \min_{i=1\ldots N} \left( \sqrt{\frac{|\mathbf{a}||\mathbf{a}^{(2)}| + |\dot{\mathbf{a}}|^2}{|\dot{\mathbf{a}}||\mathbf{a}^{(3)}| + |\mathbf{a}^{(2)}|^2}} \right). \tag{1.78}$$

Bei dem Runge-Kutta-Integrator aus den Numerical Recipes sollte statt eigener Zeitschrittweitenberechnung die in den Routinen vorgesehene automatische Schrittweitensteuerung genutzt werden.

## Teilaufgabe 2: Das Zwei-Körper-Problem

Testen Sie die Eigenschaften der verschiedenen Zeitintegratoren anhand einer Simulation des Zwei-Körper-Problems. Berechnen Sie zusätzlich für jeden Zeitschritt den spezifischen Drehimpuls $\mathbf{j}$, Gleichung (1.11), den Runge-Lenz-Vektor $\mathbf{e}$, Gleichung (1.12), und die große Halbachse $a_\mathrm{e}$, Gleichung (1.19).

Die Qualität des verwendeten Zeitintegrators kann durch verschiedene Kriterien geprüft werden (dabei ist es am praktischsten, etwa 100-1000 Perioden zu integrieren). Überprüfbar sind

- die Konstanz von $E$, $|\mathbf{j}|$, $|\mathbf{e}\,|$, $a_\mathrm{e}$ .

Plotten Sie $\log|(E - E^{\mathrm{start}})/E^{\mathrm{start}}|$, $\log||(\mathbf{e}\,| - |\mathbf{e}^{\,\mathrm{start}}|)/|\mathbf{e}^{\,\mathrm{start}}||$, $\log|(a_\mathrm{e} - a_\mathrm{e}^{\mathrm{start}})/a_\mathrm{e}^{\mathrm{start}}|$, als Funktion der Zeit für verschiedene Zeitschrittweitenparameter $\eta$ (z.B. 0.5, 0.1, 0.05, 0.01, etc.). Statt der großen Halbachse kann auch $r_\mathrm{max}$ oder $r_\mathrm{min}$ überprüft werden.[5]

Die Verwendung des Logarithmus ist sinnvoll, weil es hier, besonders in der Anfangsphase, um die Diskussion sehr kleiner Abweichungen (und deren Unterschiede für die verschiedenen Zeitintegratoren) geht.

[5] Als Plot-Routine kann `gnuplot` aufgerufen werden, für eine online Dokumentation siehe `http://www.gnuplot.info/documentation.html`,

Variieren Sie die Geschwindigkeit des zweiten Teilchens in der Startdatei, so dass Bahnen verschiedener Elliptizität auftreten. Beobachten Sie das Verhalten der Qualität des jeweiligen Integrators (Genauigkeit bei gleichem $\eta$ und $t_{\max}$) bei zunehmender Exzentrizität.

### Teilaufgabe 3: Das $N$-Körper-Problem

Führen Sie die Untersuchung von $\log|(E - E^{\text{start}})/E^{\text{start}}|$ als Funktion von $\eta$ für 100 bzw. 1000 Teilchen durch. Was beobachten Sie? Wie verändert sich der numerische Aufwand (CPU-Zeit für eine Zeiteinheit) mit der Teilchenzahl $N$?

Anmerkung: Für sinnvolle Messwerte kann es notwendig werden, nur ein wesentlich kürzeres Zeitintervall $t_{\max}$ als die oben angegebenen zwei Zeiteinheiten zu berechnen. Auch kann die Rechenzeit für 1000 Punktmassen auf langsameren Computern so groß werden, dass diese Simulation praktisch nicht durchgeführt werden kann.

In vielen Fällen kann man bei den obigen Experimenten erleben, dass eine explosionsartige Energieveränderung auftritt und die Teilchen plötzlich auseinander laufen. Dies ist ein Problem aller gewöhnlichen $N$-Körper-Zeitintegratoren, wenn zwei Teilchen sich sehr nahe kommen. Es wird durch numerische Abschneidefehler (des $1/r$-Potentials) hervorgerufen. Versucht man die Genauigkeit zu halten, tritt ein Einfrieren des Zeitschrittes auf.

Happy Coding!