

Computerphysik

Hausarbeit 4

Friedrich Hübner 2897111

Fiona Paulus 2909625

7. Juli 2017

Der Differentialgleichungssolver

Für den DGL-Solver wird das Runge-Kutta-Verfahren mit den Fehlbergkoeffizienten und automatischer Schrittweitensteuerung aus der Vorlesung verwendet. Die Berechnung aller Werte erfolgt genau nach diesem Verfahren.

Programm (abgabe4_runge_kutta.cpp)

In dieser Datei befindet sich der DGL-Solver, der von allen anderen Programmen eingebunden wird.

Mit der Funktion 'runge_kutta_init(f , t_0 , y_0)' wird der DGL-Solver initialisiert. Dabei ist $f : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ die Differentialgleichung ($y' = f(t, y)$), $t_0 \in \mathbb{R}$ der Anfangszeitpunkt und $y_0 \in \mathbb{R}^n$ der Anfangswert.

Mit der Funktion 'runge_kutta_iterate()' wird ein Schritt ausgeführt und die neue optimale Schrittweite berechnet.

Außerdem werden in der Datei Operatoren für Vektorarithmetik (+, +=, -, *) und für Ausgabe von Vektoren (<<) überladen. Zusätzlich gibt es noch Funktionen für den Betrag von Vektoren und das Skalar- und Kreuzprodukt.

Aufgabe 3

Allgemeine Hinweise

Das Programm wurde unter Windows 10 mit "g++ -o abgabe4_3.exe -Wall -Wextra -std=c++0x -O2 -static abgabe4_3.cpp" kompiliert.

a)

Die Bewegungsgleichung lautet einfach:

$$m\ddot{\vec{x}} = e\dot{\vec{x}} \times \vec{B} \quad (0.1)$$

$$= e\dot{\vec{x}} \times \left(\frac{\mu_0}{4\pi} \frac{3\vec{x}(\vec{x} \cdot \vec{m}) - \vec{m} \cdot \vec{r}^2}{|\vec{r}|^5} \right) \quad (0.2)$$

Der Dipol wurde Richtung z-Achse gelegt: $\vec{m} = (0, 0, m)^T$.

Für $\vec{y} = (\vec{x}, \dot{\vec{x}})$ gilt nun:

$$\dot{\vec{y}} = \left(\dot{\vec{x}}, \frac{e}{m} \dot{\vec{x}} \times \left(\frac{\mu_0}{4\pi} \frac{3\vec{x}(\vec{x} \cdot \vec{m}) - \vec{m} \cdot \vec{r}^2}{|\vec{r}|^5} \right) \right) \quad (0.3)$$

b) Programm (abgabe4_3.cpp)

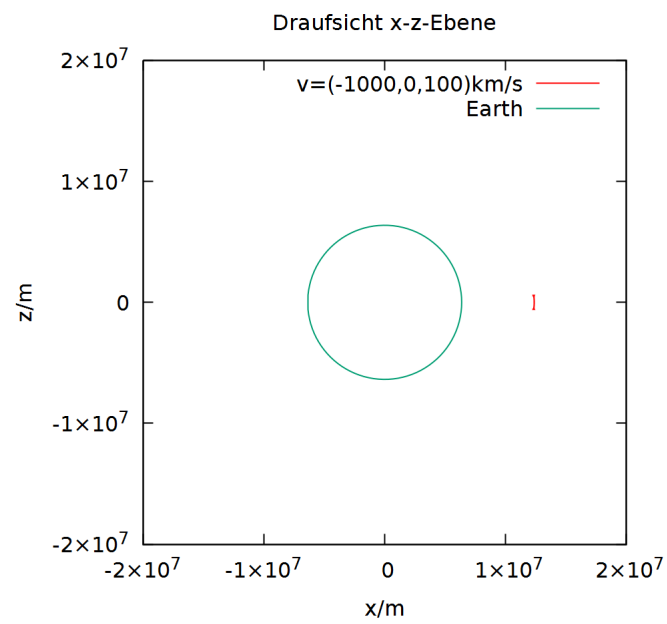
Das Programm benutzt den Fehlberg-DGL-Solver. Die verwendete Funktion f ist dabei in a) gegeben. Die Anfangsbedingungen muss man in das Programm eintragen. Das Programm simuliert 100s und gibt alle 0.1s Simulationszeit die aktuelle Zeit, Position und Geschwindigkeit aus.

c)

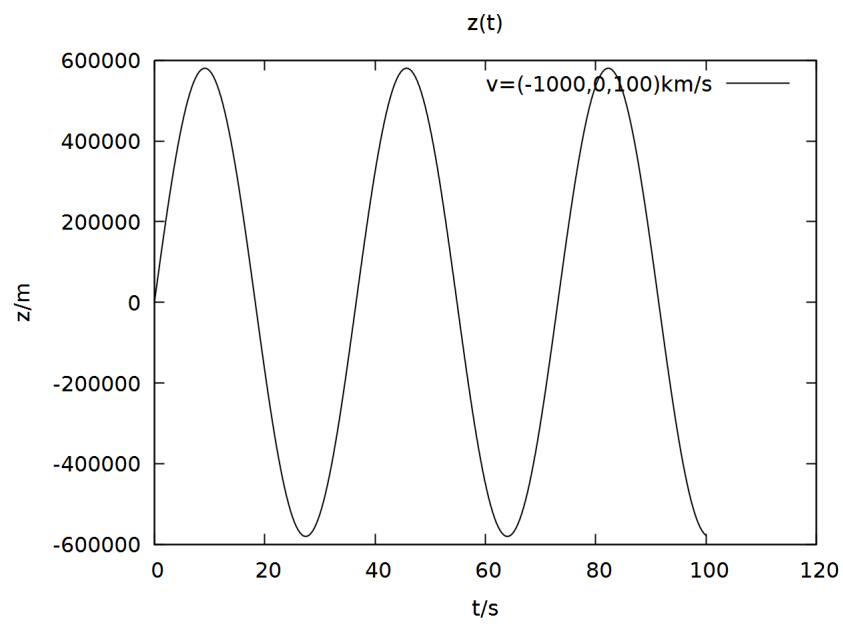
Da der Dipol in z-Richtung angeordnet ist, sind die Feldlinien am Äquator parallel zur z-Achse. Also startet das Proton mit $\vec{v} = (-1000, 0, v_z)^T$.

1. $v_z = 100\text{km/s}$

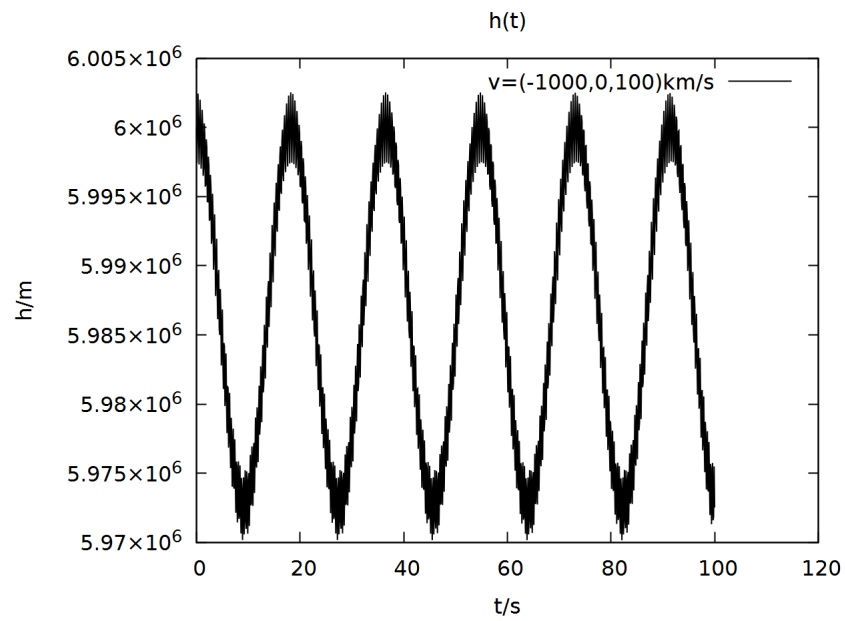
i)



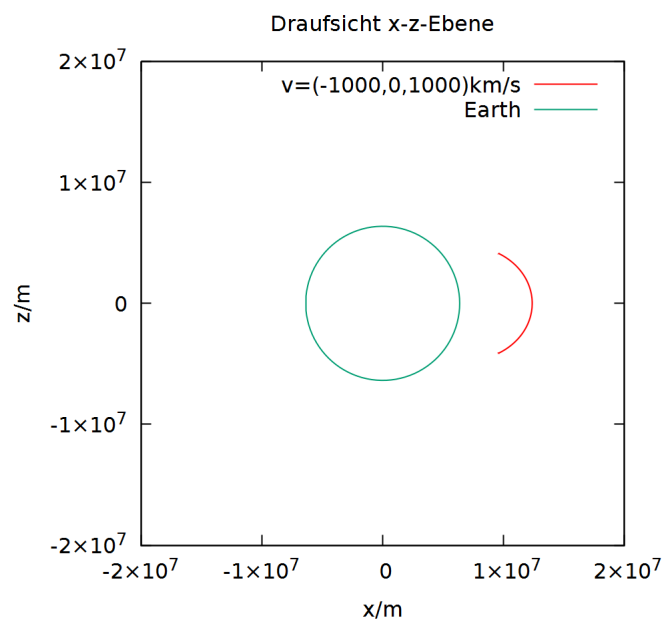
ii)



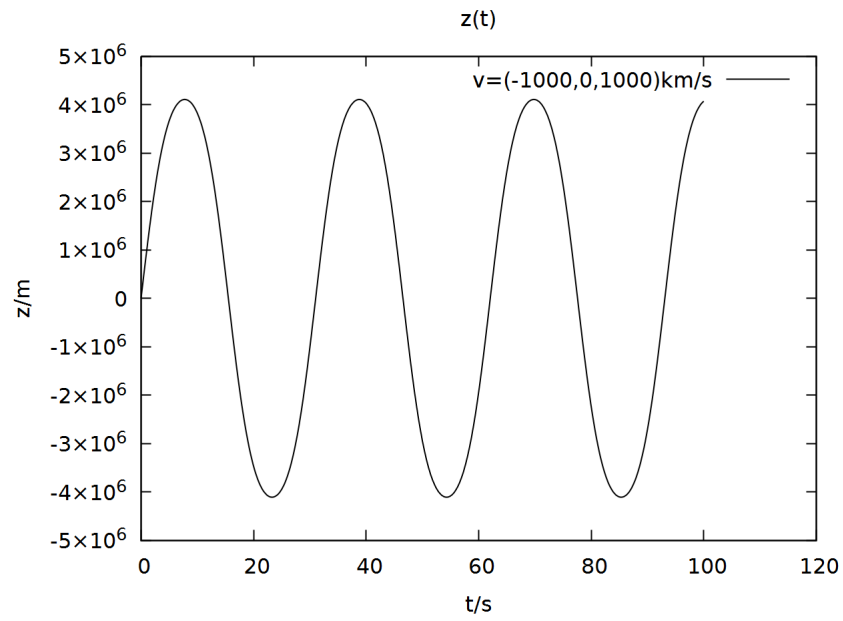
iii)

2. $v_z = 1000 \text{ km/s}$

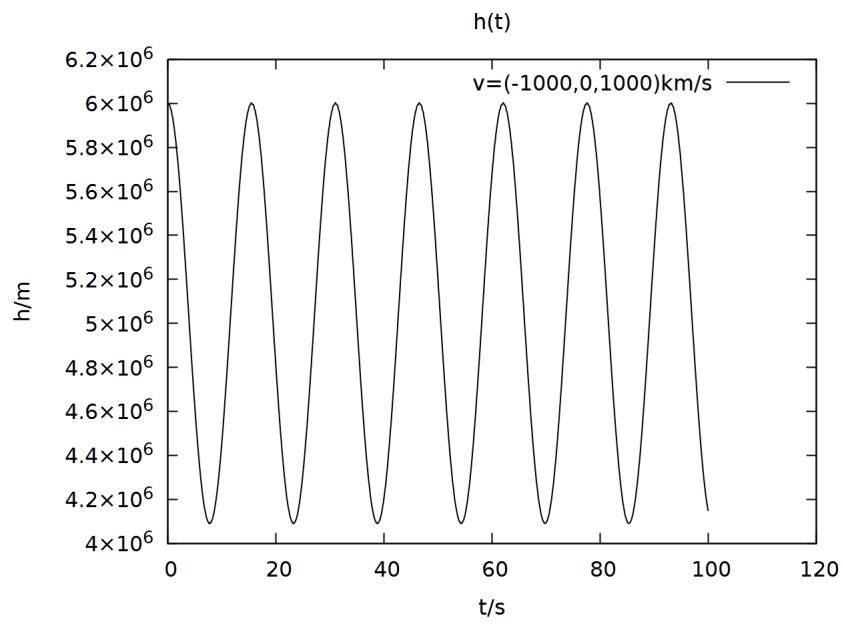
i)



ii)

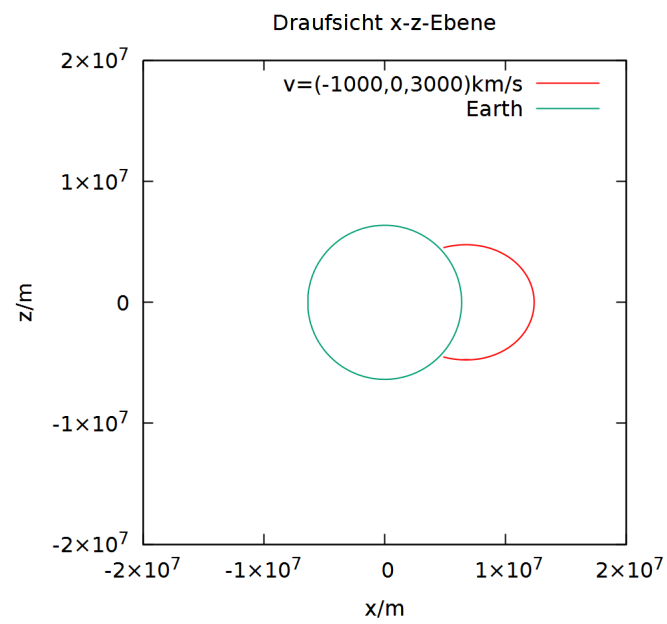


iii)

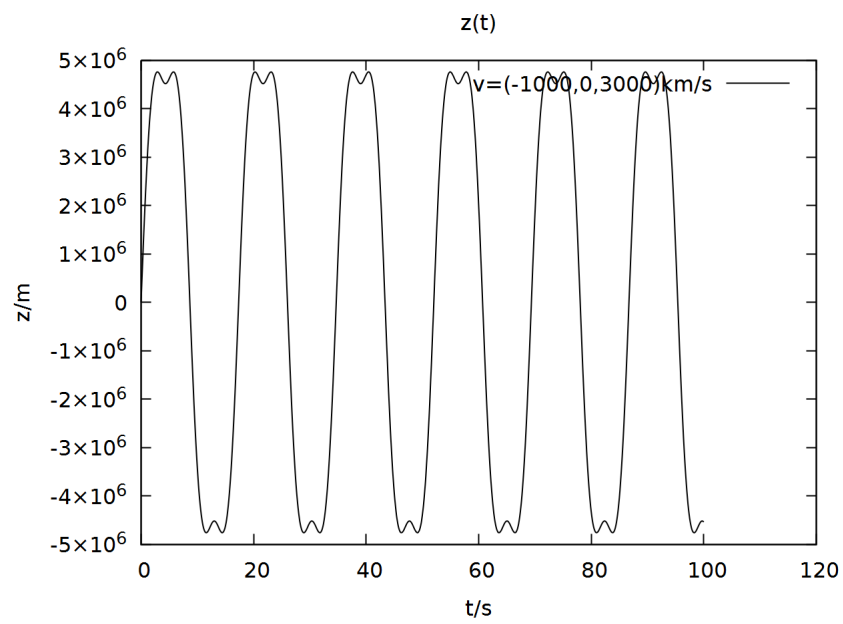


3. $v_z = 3000 \text{ km/s}$

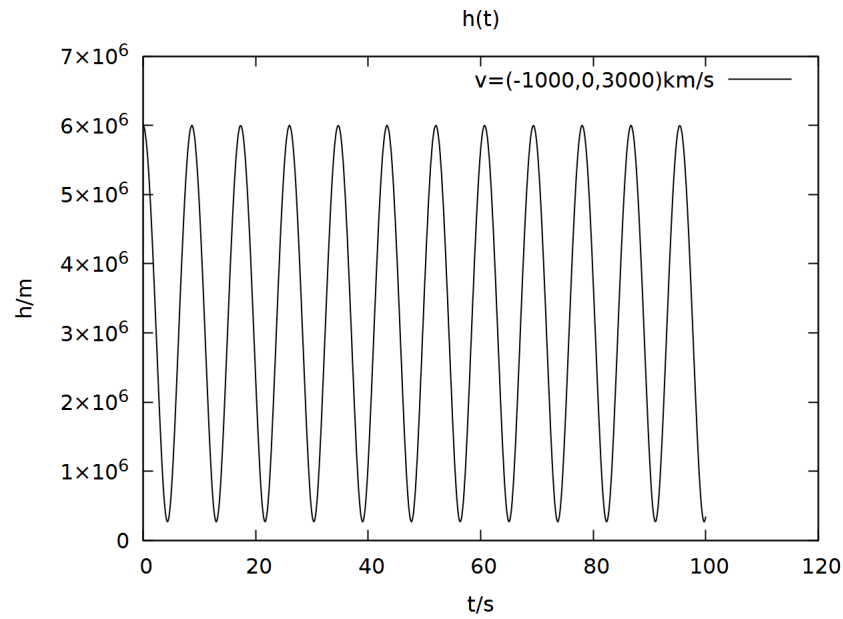
i)



ii)



iii)



Sonstige abgegebene Dateien

plot_i.plt, plot_ii.plt, plot_iii.plt

Die Plot-Dateien für die Diagramme für jeweils Fall i), ii), iii)

1.txt, 2.txt, 3.txt

Die Ausgabedateien für die einzelnen Anfangswerte