TALLINNA TEHNIKAÜLIKOOL
FÜÜSIKAINSTITUUT

Laetud osakese liikumine kolmemõõtmelise ruumi magnetväljas

Projekt

Ver 2.0

Üliõpilane: Toomas Tahves

Üliõpilase kood: 164107

Õppejõud: Mihhail Klopov

Esitamise kuupäev: 06.12.2018

Parandatud: 07.12.2018

Tallinn 2018

Sissejuhatus

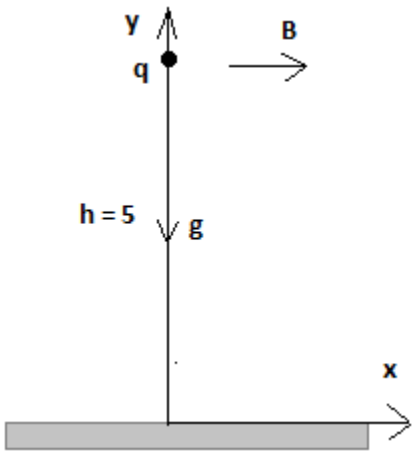
Töö eesmärk on kirjeldada laetud osakese liikumist kolmemõõtmelise ruumi magnetväljas.

Töövahenditeks on programmeerimiskeel Python ja raamistikud Numpy, Scipy ja Matplotlib.

Töö tulemuseks on tuletatud valemid ja graafikud, mis kirjeldavad osakese liikumise trajektoori, kiirust ning energiat (potentsiaalne, kineetiline, koguenergia).

Teoreetilised alused

Algtingimused: asukoht $\vec{r} = (0;\ 5;\ 0)$, kiirus $\vec{v} = (0; 0; 0)$, magnetvälja induktsioon $\vec{B} = (0.1; 0; 0)$, mass m = 1 mg, laeng q = 1 mC ning raskuskiirendus g = 9.814 $\frac{m}{s^2}$. Osake asub vaakumis ja algseisu kirjeldab Joonis 1.



Joonis 1 Osakese algseisu kirjeldus.

Osakesele mõjub kaks jõudu: raskusjõud ja Lorentzi jõud.

Magnetväljas liikuvale osakese mõjub Lorentzi jõud (1).

$$\vec{F} = q(\vec{v} \times \vec{B}) \qquad (1)$$

Selleks, et arvutada Lorentzi jõu projektsioone telgedele, on vaja arvutada maatriksi (2) determinant ja saadud tulemused grupeerida ühikvektorite $\vec{\imath}, \vec{\jmath}$ ja $\vec{k}$ järgi.

$$\begin{vmatrix} \vec{\imath} & \vec{\jmath} & \vec{k} \\ v_x & v_y & v_z \\ B_x & B_y & B_z \end{vmatrix} = \left(v_y B_z - v_z B_y\right)i + (v_z B_x - v_x B_z)\vec{\jmath} + (v_x B_y - v_y B_x)\,\vec{k} \qquad (2)$$

Lorentzi jõu projektsioone x, y, z telgedele kirjeldab võrrandite süsteem (3).

$$\begin{cases} F_x = q(v_y B_z - v_z B_y) \\ F_y = q(v_z B_x - v_x B_z) \\ F_z = q(v_x B_y - v_y B_x) \end{cases} \qquad (3)$$

Newtoni teisese seaduse $F = ma \Rightarrow a = \frac{F}{m}$ rakendamisel saab võrrandid (3) kirjutada kujule (4).

$$\begin{cases} \ddot{x} = a_x = \frac{q}{m}(v_y B_z - v_z B_y) \\ \ddot{y} = a_y = \frac{q}{m}(v_z B_x - v_x B_z) \\ \ddot{z} = a_z = \frac{q}{m}(v_x B_y - v_y B_x) \end{cases} \quad (4)$$

Kiiruse projektsioone kirjeldab võrrandisüsteem (5).

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \end{cases} \quad (5)$$

Selleks, et osakese liikumist modeleerida, tuleb raskusjõu ja Lorentzi jõu projektsioonid kokku panna. Programmeeritud lahenduses on kasutatud võrrandeid (6).

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \ddot{x} = \frac{q}{m}(v_y B_z - v_z B_y) \\ \ddot{y} = \frac{q}{m}(v_z B_x - v_x B_z) - g \\ \ddot{z} = \frac{q}{m}(v_x B_y - v_y B_x) \end{cases} \quad (6)$$

Kineetilise, potentsiaalse ja koguenergia arvutamist kirjeldavad valemid (7).

$$\begin{cases} E_{kin} = \frac{mv^2}{2} = \frac{m(v_x^2 + v_y^2 + v_z^2)}{2} \\ E_{pot} = mgh = mgy \\ E_{tot} = E_{kin} + E_{pot} \end{cases} \quad (7)$$

Töö lahendus

Ülesanne on lahendatud Pythonis. Kood on esitatud terviklikult ja seda saab sobivas keskkonnas käivitada. Edaspidi järgneb lahendus ja selle selgitus.

Kood	Selgitus

`import matplotlib.pyplot as plt`
`import scipy.integrate as ode`
`import numpy as np`
	Lahendamisel kasutatud teekide laadimine.

`g = 9.814`
`q = 10**-3`
`m = 10**-6`
`h = 5`
`B = [0.1, 0, 0]`
`y0 = [0, h, 0, 0, 0, 0]`
	Lähteandmete defineerimine.
	g - raskuskiirendus [m/s^2]
	q - osakese laeng [C]
	m - osakese mass [kg]
	h - osakese esialgne y-asukoht [m]
	B – magnetilise induktsiooni vektori komponendid [T]
	y0 – esialgsete andmete massiiv kujul $[x_0, y_0, z_0, v_x, v_y, v_z]$

`def func(y, t):`
`  yp = np.zeros_like(y)`
`  yp[0] = y[3]`
`  yp[1] = y[4]`
`  yp[2] = y[5]`
`  yp[3] = q / m * (y[4]*B[2] - y[5]*B[1])`
`  yp[4] = q / m * (y[5]*B[0] - y[3]*B[2]) - g`
`  yp[5] = q / m * (y[3]*B[1] - y[4]*B[0])`
`  return yp`
	Funktsiooni integreerimiseks ettevalmistamine vastavalt valemitele (6).

`time = np.arange(0, 1, 0.001)`	Tulemused arvutatakse ajavahemikus $t \in (0; 1)$ sammuga dt = 0.001 [s].
`res = ode.odeint(func, y0, time)`	Diferentsiaalvõrrandite lahendamisel kasutatakse odeint funktsiooni. Tulemused tagastatakse massiivi kujul $[x, y, z, v_x, v_y, v_z]$
`x, y, z, vx, vy, vz = np.transpose(res)`	Transponeerin massiivi, et edaspidi oleks väärtuseid mugavam kasutada.

`ekin = 0.5 * m * (vx**2 + vy**2 + vz**2)`
`epot = m * g * y`
`etot = np.around(ekin + epot, 10)`
	Kin, pot ja koguenergia arvutamine vastavalt valemitele (7). Koguenergia ümardatud nulli, sest arvutamisel jääb sisse väike jääk.

`plt.figure(figsize=(8,5))`
`p1 = plt.subplot()`
`p1.set_title('Trajektoor YZ-tasandil')`
`p1.set_xlabel('z (m)')`
	Graafikute joonistamine.
	Esimene graafik kirjeldab YZ-tasandil osakese trajektoori.

p1.set_ylabel('y (m)') p1.plot(z, y, color='tab:blue') p1.grid() fig, ax1 = plt.subplots() ax1.set_xlabel('Aeg (s)') lns1 = ax1.plot(time, ekin*10**8, color='tab:red', label='E-kin (10^-8 J)') ax1.tick_params(axis='y', labelcolor='tab:red') ax2 = ax1.twinx() lns2 = ax2.plot(time, epot*10**8, color='tab:blue', label='E-pot (10^-8 J)') ax2.tick_params(axis='y', labelcolor='tab:blue') ax3 = ax1.twinx() lns3 = ax3.plot(time, etot, color='tab:green', label='E-tot (J)') ax3.tick_params(axis='y', right=False, labelright=False) lns = lns1 + lns2 + lns3 labs = [l.get_label() for l in lns] ax1.legend(lns, labs, loc=1) ax1.grid() fig.tight_layout() plt.show()	Teisel graafikul on kolm alagraafikut, millel on E_{kin}, E_{pot}, E_{tot} väärtuste sõltuvus ajast. twinx() funktsiooni abil saab ühise x-teljega, aga erineva y-telje skaalaga jooni samale graafikule joonistada Legendide kuvamiseks sai alagraafikute legendide nimed kokku kogutud ja esimesel alagraafikul kuvatud. tight_layout() funktsioon, et graafiku äärtes oleks rohkem ruumi numbrite ilusamalt kuvamiseks.

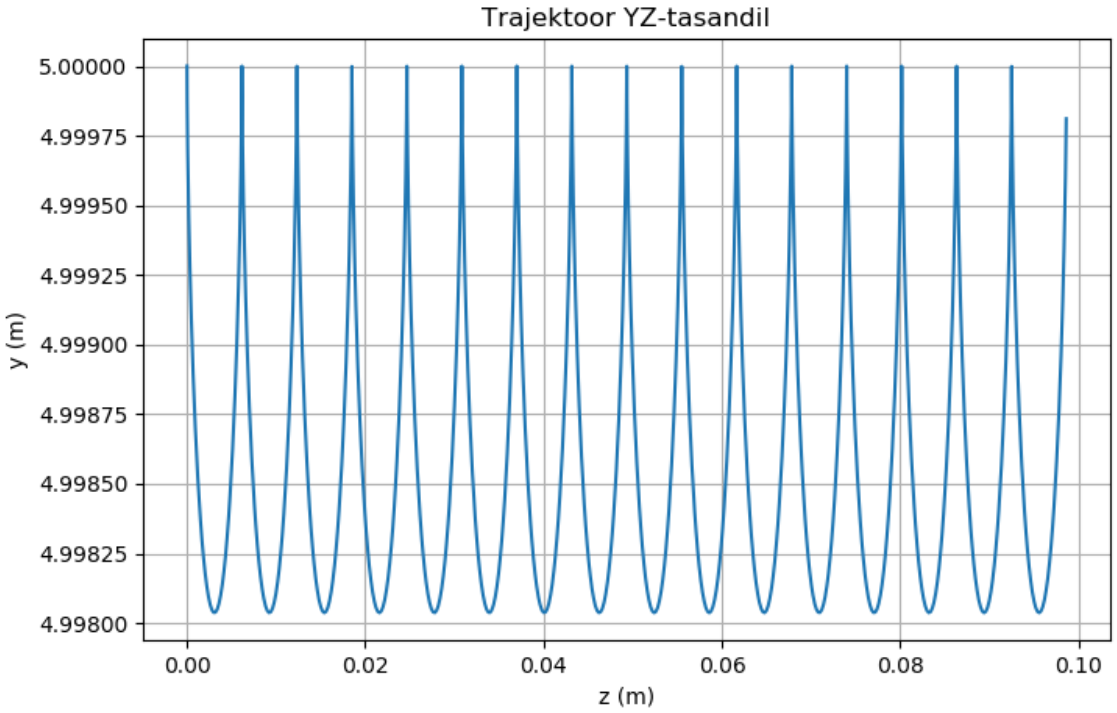
Kokkuõte

Töö käigus sai tuletatud vajalikud valemid, mis kirjeldavad laetud osakese liikumist magnetväljas. Modeleerimisel sai kasutatud programmeerimiskeelt Python ja selle raamistikke.

Töö tulemusena sai loodud graafikud, mis kirjeldavad osakese trajektoori, kiirust ja energia muutumist ajas.

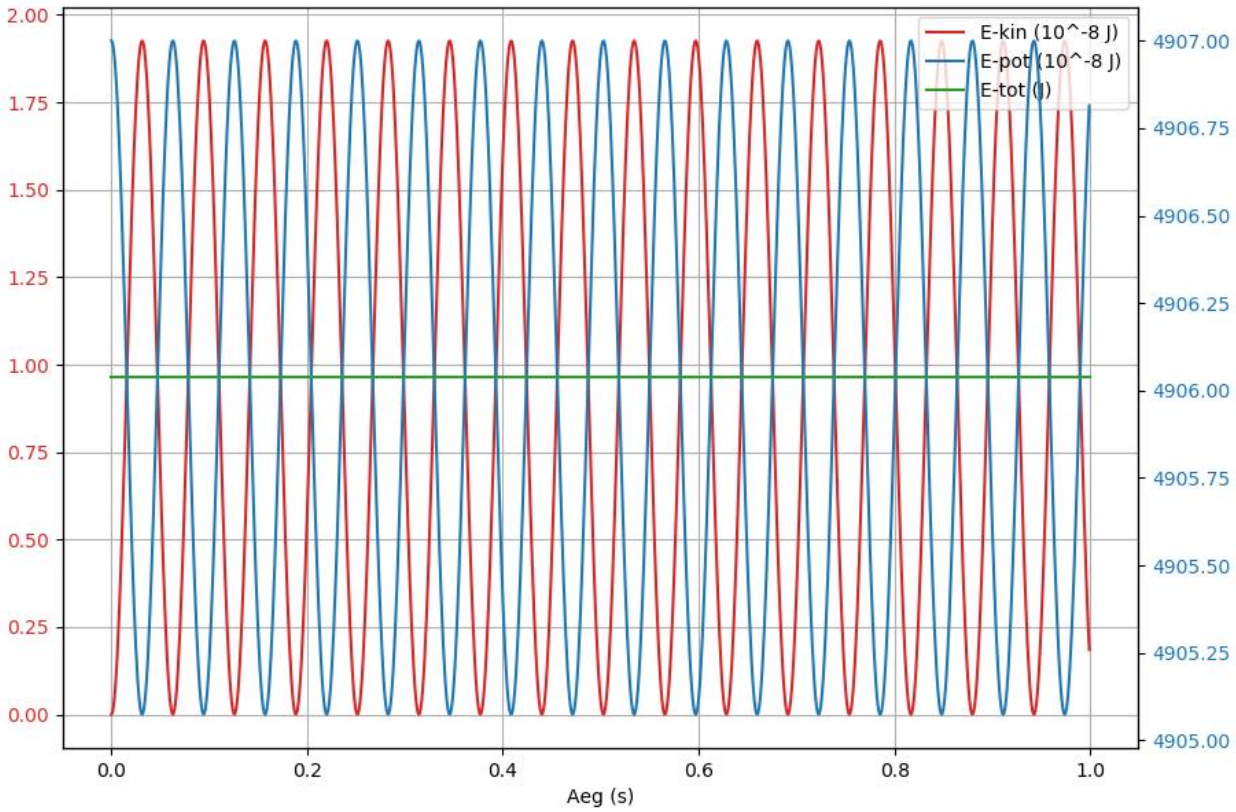
Osakese X koordinaat ei muutu ja osakese trajektoori YZ-tasandil kirjeldab Joonis 2, mis asub Lisa 1 all. Osakese potentsiaalse, kineetilise ja koguenergia sõltuvust ajast kirjeldab Joonis 3, mis asub Lisa 2 all.

Lisa 1



Joonis 2 Osakese trajektoor YZ-tasandil.

Lisa 2



Joonis 3 Osakese kineetilise, potentsiaalse ja koguenergia sõltuvus ajast.