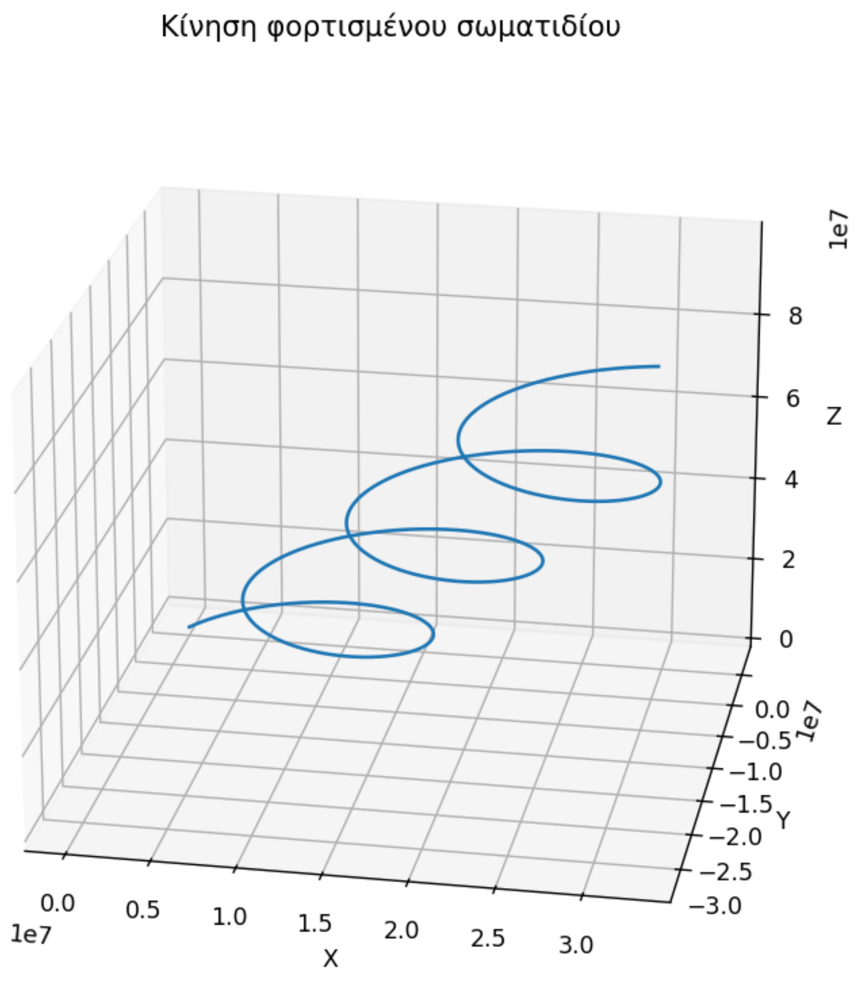
ΔΗΜΟΚΡΙΤΕΙΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΡΑΚΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ & ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

Μάθημα: Φυσική Πλάσματος

Θέμα Εργασίας : Ανάπτυξη κώδικα για την προσομοίωση κίνησης φορτισμένου σωματιδίου σε Ομογενές Μαγνητικό Πεδίο B(0,0,B_z) και Ηλεκτρικό πεδίο E(E_x,E_y,0) κάθετα μεταξύ τους .



Ονοματεπώνυμο: Ηρακλής Γούσης

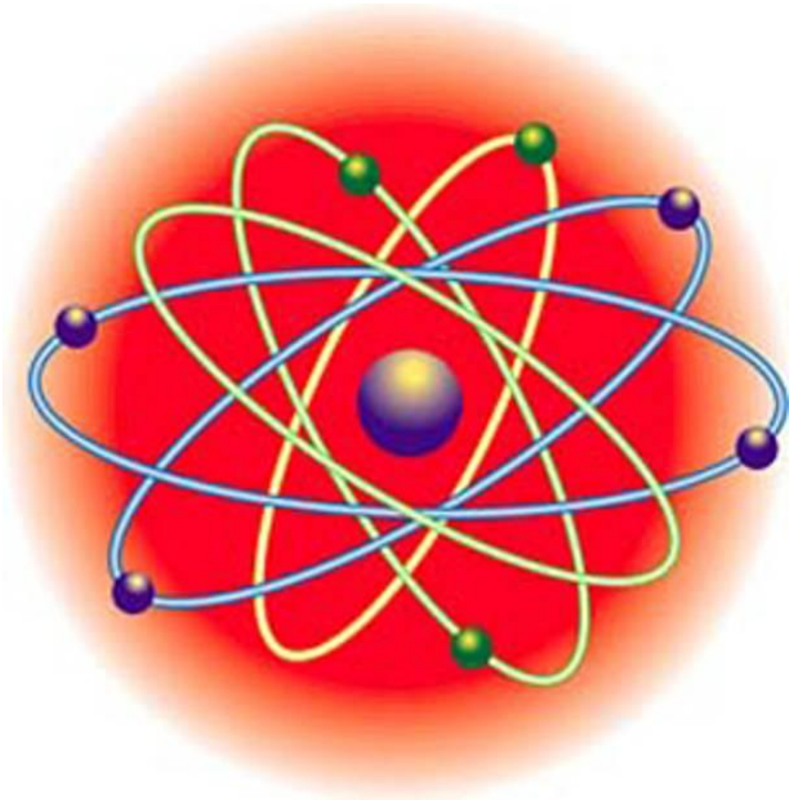
ΑΜ: 58650

Εξάμηνο: 6^{O}

Περιεχόμενα

Εισαγωγή

Στο μάθημα Φυσική πλάσματος γίνεται αναφορά στην κίνηση ενός φορτισμένου σωματιδίου σε διαφορές καταστάσεις μέσα σε Ομογενές Μαγνητικό Πεδίο και Ομογενές Ηλεκτρικό Πεδίο. Συγκεκριμένα αντικείμενο της παρούσας εργασίας είναι η προσομοίωση κίνησης φορτισμένου σωματιδίου σε Ομογενές Μαγνητικό Πεδίο Β(0,0,$\mathrm{B_z}$) και Ηλεκτρικό πεδίο Ε(E_x,$\mathrm{E_y}$,0) κάθετα μεταξύ τους. Στην εργασία επίσης περιλαμβάνεται η θεωρητική ανάλυση της κίνησης και η επεξήγηση του κώδικα. Τέλος για την προσομοίωση έγινε χρήση της γλώσσας προγραμματισμού Python στον Διερμηνευτή PyCharm.



1. Θεωρητική Ανάλυση

Η κίνηση ενός σωματιδίου περιγράφεται από την εξίσωση κίνησης :

$$m * \frac{dv}{dt} = q(E + v \; x \; B) + F_{non-EM}$$

οπού m: μάζα του σωματιδίου

$\frac{dv}{dt}$: η επιτάχυνση του σωματιδίου

q: φορτίο του σωματιδίου

v: ταχύτητα του σωματιδίου

E: το Ηλεκτρικό πεδίο

B: το Μαγνητικό πεδίο

$F_{\mathrm{non-EM}}$: Δύναμη προερχομένη από μη-Ηλεκτρομαγνητικά Πεδία

Στην συγκεκριμένη περίπτωση έχουμε $F_{\mathrm{non-EM}}$=0 , E(E_x,E_y,0) , B(0,0,B_z)

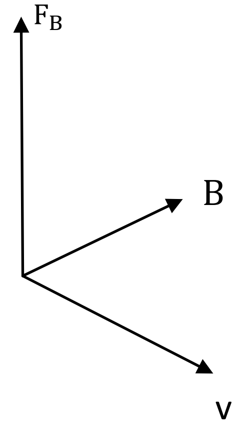
Άρα η εξίσωση κίνησης γίνεται ως έξης :

$$m * \frac{dv}{dt} = q(E + v \; x \; B) = qE + q(v \; x \; B) \qquad \textit{εξίσωση (1)}$$

- το γινόμενο qE είναι η ηλεκτρική δύναμη που δρα στο σωματίδιο λόγω του ηλεκτρικού πεδίο E.
- το γινόμενο q(v x B) είναι η μαγνητική δύναμη που δρα στο σωματίδιο λόγω του μαγνητικού πεδίου B .

<u>Σχετικά με τις δυνάμεις :</u>

Η δύναμη που ασκεί το μαγνητικό πεδίο στο φορτισμένο σωματίδιο , έχει διεύθυνση κάθετη στο επίπεδο που ορίζουν η ταχύτητα και το μαγνητικό πεδίο και η φορά της είναι ανάλογη του πρόσημου του φορτίου ($\vec{F} = q * (\vec{v}\ x\ \vec{B})$)



Σε αυτό το σημείο είναι καλό να αναφερθεί ότι τα πεδία που προκαλούνται από την κίνηση των σωματιδίων είναι αμελητέα σε σύγκριση με τα εφαρμοσμένα πεδία.

v = (v_x, v_y, v_z) E(E_x,E_y,0) B(0,0,B_z)

Από την εξίσωση (1) γίνεται :

$$m * \frac{d}{dt}(v_x, v_y, v_z) = q(E_x, E_y, 0) + q(v_x, v_y, v_z) \times (0,0,B_z)$$

άρα η επιτάχυνση είναι

$$\frac{d}{dt}(v_x, v_y, v_z) = \frac{q}{m}(E_x, E_y, 0) + \frac{q}{m}(v_x, v_y, v_z) \times (0,0,B_z) \qquad \text{εξίσωση (2)}$$

οπού το ’’ (v_x, v_y, v_z) x (0,0,B_z) ‘’ είναι εξωτερικό γινόμενο

<u>Υπολογισμός του εξωτερικού γινομένου ‘’ v x B ‘’:</u>

$$v \times B = \begin{vmatrix} i & j & k \\ v_x & v_y & v_z \\ 0 & 0 & B_z \end{vmatrix} = \begin{vmatrix} v_y & v_z \\ 0 & B_z \end{vmatrix} * \vec{i} - \begin{vmatrix} v_x & v_z \\ 0 & B_z \end{vmatrix} * \vec{j} + \begin{vmatrix} v_x & v_y \\ 0 & 0 \end{vmatrix} * \vec{k} \Leftrightarrow$$

$$\mathrm{v \times B} = \mathrm{v_y} * \mathrm{B_z} * \vec{\imath} - v_x * B_z * \vec{\jmath} + 0 * \vec{k}$$

άρα η εξίσωση (2) γίνεται

$$\frac{d}{dt}\left(\mathrm{v_x}, \mathrm{v_y}, \mathrm{v_z}\right) = \frac{\mathrm{q}}{m}(\mathrm{E}_x, \mathrm{E_y}, 0) + \frac{q}{m}(\mathrm{v_y} * \mathrm{B_z}, -\mathrm{v_x} * \mathrm{B_z}, 0)$$

άρα για την επιτάχυνση στον κάθε άξονα θα ισχύει :

$$\mathrm{a_x} = \frac{\mathrm{q}}{\mathrm{m}} * (\mathrm{E_x} + \mathrm{v_y} * \mathrm{B_z})$$

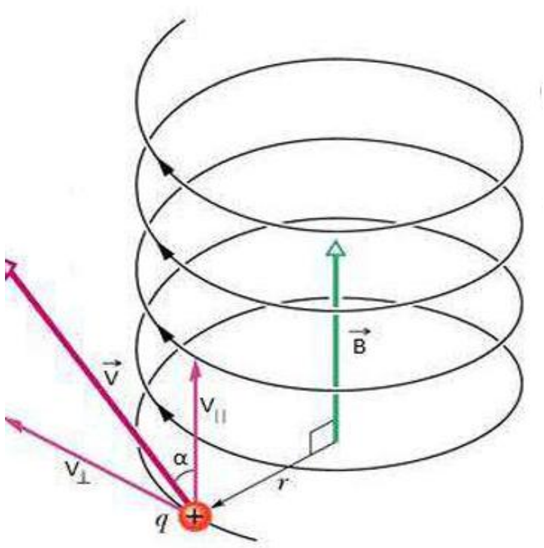
$$\mathrm{a_y} = \frac{\mathrm{q}}{\mathrm{m}} * (\mathrm{E_y} + \mathrm{v_x} * \mathrm{B_z})$$

$$\mathrm{a_z} = \frac{\mathrm{q}}{\mathrm{m}} * (0) = 0$$

<u>Σημείωση:</u>

Οι παραπάνω επιταχύνσεις δημιουργούνται λόγω των πεδίων

Επομένως λόγω των δυνάμεων που δρουν πάνω στο φορτισμένο σωματίδιο από τα πεδία ,το σωματίδιο αποκτά μια συνιστώσα ταχύτητα που αν την αναλύσουμε στους άξονες X/Y θα διαπιστώσουμε ότι εξαιτίας της ταχύτητας (του άξονα Y που είναι παράλληλη με το $\vec{B}$) το φορτισμένο σωματίδιο θα εκτελέσει Ευθύγραμμη Ομαλή Κίνηση και εξαιτίας της ταχύτητας (του άξονα X που είναι κάθετη με το $\vec{B}$) θα εκτελέσει Ομαλή Κυκλική Κίνηση. Δηλαδή το σωματίδιο θα εκτελέσει δυο κινήσεις και ταυτόχρονα.



2. Επεξήγηση του Κώδικα

Για την προσομοίωση έγινε χρήση της γλώσσας προγραμματισμού Python στον Διερμηνευτή PyCharm.

Αρχικά γίνεται εισαγωγή των απαραιτήτων βιβλιοθηκών

```
import numpy as np
import matplotlib.pyplot as plt
```

Το NumPy είναι μια βιβλιοθήκη Python που χρησιμοποιείται για εργασία με πίνακες . Έχει επίσης συναρτήσεις για εργασία στον τομέα της γραμμικής άλγεβρας, του μετασχηματισμού Fourier και των πινάκων. Αυτή η βιβλιοθήκη μας είναι χρήσιμη έτσι ώστε να ορίσουμε το χρονικό διάστημα που θα γίνει η προσομοίωση .

Το Matplotlib είναι μια ολοκληρωμένη βιβλιοθήκη για τη δημιουργία στατικών, κινούμενων και διαδραστικών απεικονίσεων στην Python.

Στην συνεχεία αρχικοποιώ την μάζα του σωματιδίου και το φορτίο του και τιμές του Ομογενές Μαγνητικού Πεδίου B(0,0,$\mathrm{B_z}$) και του Ηλεκτρικού πεδίο E(E_x,$\mathrm{E_y}$,0)

```
q = 1
m = 1
Bz = 4*pow(10,-2)
Ex = 2*pow(10,3)
Ey = 2*pow(10,3)
```

Το x^y στην γλώσσα προγραμματισμού Python ορίζεται ως pow(x,y)

Επομένως ορίζω στο πρόγραμμα B(0,0,4*10^{-2}) και E(2*10^3,2*10^3,0)

Έπειτα εντάσσω τις αρχικές συντεταγμένες του φορτισμένου σωματιδίου και την αρχική ταχύτητα του.

```
x0 = 0.0
y0 = 0.0
z0 = 0.0
vx0 = 2.0*pow(10,5)
vy0 = 2.0*pow(10,5)
vz0 = 2.0*pow(10,5)
```

άρα είναι στην θέση O(0,0,0) και έχει αρχική ταχύτητα $v_0 = (2 * 10^5, 2 * 10^5, 2 * 10^5)$ m/s

Για το χρονικό διάστημα που θα γίνει η προσομοίωση μπορούμε να το κωδικοποιήσουμε με τις έξης εντολές

```
dt = 0.01
timesteps = 50000
t = np.linspace(0, timesteps * dt, timesteps)
```

- Με την μεταβλητή dt ορίζεται το δευτερόλεπτο (1s)
- Η μεταβλητή timesteps είναι το χρονικό βήμα που θα γίνει η προσομοίωση
- η συνάρτηση np.linspace() ορίζει τον συνολικό χρονικό διάστημα που γίνει η προσομοίωση. Ουσιαστικά είναι ένας πινάκας με διανεμημένο εύρος τιμών .

  Η συνάρτηση αυτή συντάσσετε με τον έξης τρόπο

  np.linspace (start , stop , num)

όπου

- το ''start'' είναι η αρχική τιμή του εύρους
- το ''stop'' η τελική τιμή του εύρους
- το ''num'' ο αριθμός των σημείων που θα δημιουργηθούν

Για την καταγραφή θέσεων και ταχύτητας του σωματιδίου προσθέτουμε στον κώδικα τις απαραίτητες λίστες

```
x_list = []
y_list = []
z_list = []
vx_list = []
vy_list = []
vz_list = []
```

Μέθοδος Euler

Για την πραγματοποίηση της προσομοίωσης θα χρειαστεί η μέθοδος Euler. Η συγκεκριμένη μέθοδος επιτρέπει την επίλυση διαφορικών εξισώσεων πρώτης τάξης με μια δεδομένη αρχική τιμή.

Συγκεκριμένα για διαφορική εξίσωση $\frac{dS(t)}{dt} = f(t, S(t))$ οπού η F είναι μια συνάρτηση που επιστρέφει την παράγωγο , η την αλλαγή , μιας κατάστασης με δεδομένη τιμή χρόνου και κατάστασης. Έπειτα το t είναι αριθμητικό πλέγμα του διαστήματος $[t_0, t_f]$ με απόσταση h. χωρίς απώλεια γενικότητας , υποθέτουμε ότι $t_0 = 0$ και το t_f=N*h για κάποιο θετικό ακέραιο N.

Η γραμμική προσέγγιση του S(t) μεταξύ του t_j και t_{j+1} είναι $S(t_{j+1}) = S(t_j) + h * f(t_j, S(t_j))$ και ο τύπος Euler μας επιτρέπει να υπολογιστεί μια προσέγγιση για την κατάσταση $S(t_{j+1})$ γνωρίζοντας το $S(t_j)$. Φυσικά ξεκινώντας από μια δεδομένη αρχική τιμή του $S_0 = S(t_0)$ μέχρι την κατάσταση $S(t_f)$.

Για όλα τα παραπάνω λοιπόν θα χρειαστεί ένας βρόχος επανάληψης και μέσα σε αυτόν εφαρμογή της μεθόδου Euler .

```
for i in range(timesteps):
    x_list.append(x0)
    y_list.append(y0)
    z_list.append(z0)
    vx_list.append(vx0)
    vy_list.append(vy0)
    vz_list.append(vz0)


    ax = (q / m) * (Ex + vy0 * Bz)
    ay = (q / m) * (Ey-vx0*Bz)
    az =0


    vx0 = vx0 + ax * dt
    vy0 = vy0 + ay * dt
    vz0 = vz0 + az * dt

    x0 = x0 +  vx0 * dt
    y0 = y0 +  vy0 * dt
    z0 = z0 +  vz0 * dt
```

Ο βρόχος επανάληψης είναι το ‘’for’’ που στην Python συντάσσετε με τον έξης τρόπο:

‘’for i in range (timesteps):’’

Η συνάρτηση range() αντιπροσωπεύει μια αμετάβλητη ακολουθία αριθμών και χρησιμοποιείται συνήθως για τη δημιουργία βρόχου συγκεκριμένου αριθμού φορών σε for βρόχους.

Η μέθοδος append() προσθέτει στοιχεία στο τέλος της λίστας και με αυτό τον τρόπο και με την επανάληψη του βρόχου θα έχουμε τις συντεταγμένες και τις ταχύτητες του κάθε άξονα.

```
x_list.append(x0)
y_list.append(y0)
z_list.append(z0)
vx_list.append(vx0)
vy_list.append(vy0)
vz_list.append(vz0)
```

Τέλος για την απεικόνιση της πορείας του φορτισμένου σωματιδίου στο χώρο θα χρειαστούν οι εξής εντολές :

```
figure = plt.figure()
ax = figure.add_subplot(111, projection='3d')
ax.plot(x_list, y_list, z_list)
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Κίνηση Φορτισμένου Σωματιδίου')
plt.show()
```

```
figure = plt.figure()
ax = figure.add_subplot(111, projection='3d')
```

Οι δυο πρώτες εντολές μας επιτρέπουν την δημιουργία τρισδιάστατων αξόνων.

```
ax.plot(x_list, y_list, z_list)
```

Με αυτή την εντολή απεικονίζονται όλα τα σημεία του σωματιδίου που έχουν αποθηκευτεί κατά την διάρκεια της προσομοίωσης.

```
ax.set_xlabel('X')
ax.set_ylabel('Y')
ax.set_zlabel('Z')
ax.set_title('Κίνηση Φορτισμένου Σωματιδίου')
```

Η εντολή ax.set_ label(‘ ’) ονοματίζει τους άξονες και η εντολή ax.set_title(‘ ’) δίνει Τίτλο στο διάγραμμα .

```
plt.show()
```

Τέλος η παραπάνω εντολή εμφανίζει το διάγραμμα σε νέο παράθυρο.

3. Πηγές

1. https://www.youtube.com/watch?v=PCeBV_lhCsQ
2. https://pythonnumericalmethods.studentorg.berkeley.edu/notebooks/chapter22.03-The-Euler-Method.html
3. https://matplotlib.org/
4. https://www.w3schools.com/python/numpy/numpy_intro.asp#:~:text=NumPy%20is%20a%20Python%20library,in%202005%20by%20Travis%20Oliphant.
5. https://www.youtube.com/watch?v=FYKCnSH-RuI&list=PLLMmbOLFy25Eohpgb_V3GWKdf8sL0Upvt&index=33
6. https://www.freecodecamp.org/news/python-for-loop-for-i-in-range-example/
7. https://docs.python.org/3/library/stdtypes.html#range
8. https://stackoverflow.com/questions/19431730/python-append0
9. https://stackoverflow.com/questions/53103168/add-subplot111-projection-3d-vs-axes3dfig
10. Αρχείο .PDF Μάθημα 4^{o} & 5^{o} (Κίνηση Φορτισμένων Σωματιδίων – Μαγνητικός καθρέφτης – Αδιαβατικές Σταθερές)
11. Αρχείο .PDF plasma_physics_KanarisTsigganos