ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΑ ΠΕΔΙΑ Β

(2020-2021)

3η Σειρά Ασκήσεων

Ονοματεπώνυμο:

- Χρήστος Τσούφης

Αριθμός Μητρώου:

- 03117176

Στοιχεία Επικοινωνίας:

- el17176@mail.ntua.gr

Η/Μ Πεδία Β

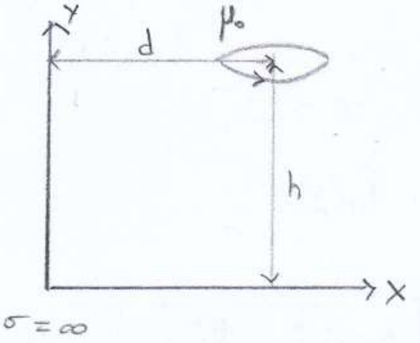
Σειρά Ασκήσεων 3η

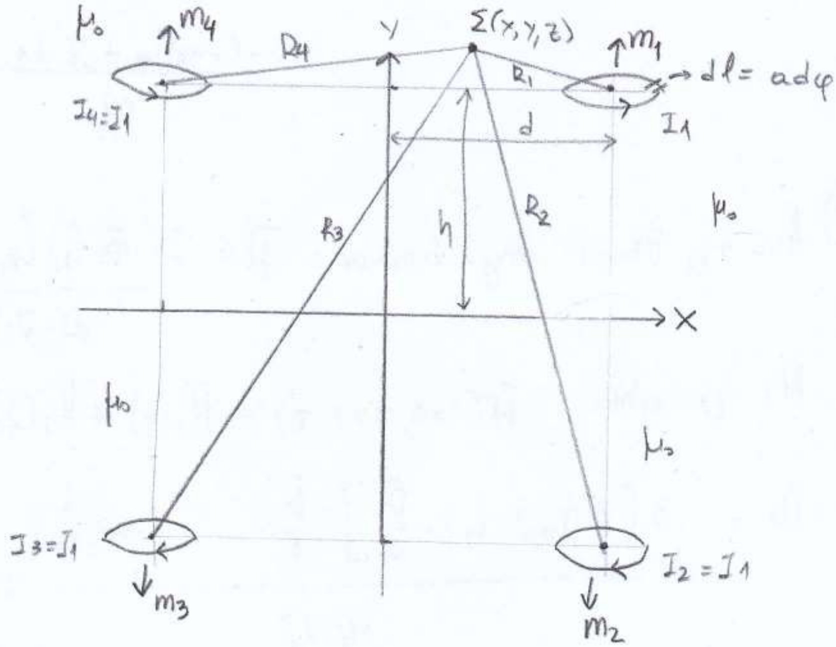
2020 - 2021

Άσκηση 8

Χρήστος Τσούφης

03117176





a) Εύρεση Διανυσματικού Δυναμικού:

- για $x<0$ ή $y<0$: $\bar{A}(\Sigma)=0$
- για $x>0$, $y>0$: $\bar{A}(\Sigma) = \bar{A}_1(\Sigma) + \bar{A}_2(\Sigma) + \bar{A}_3(\Sigma) + \bar{A}_4(\Sigma)$

$A_i = \frac{\mu}{4\pi}\int_{C_i} I(\bar{r})\, d\bar{l}_i$, $\hat{i}_\varphi = -\sin\varphi\cdot\hat{i}_x + \cos\varphi\cdot\hat{i}_z$

για σημείο αναφοράς δυναμικού το κέντρο των αξόνων

$$R_1 = \sqrt{(x-d)^2+(y-h)^2+z^2}$$
$$R_2 = \sqrt{(x-d)^2+(y+h)^2+z^2}$$
$$R_3 = \sqrt{(x+d)^2+(y+h)^2+z^2}$$
$$R_4 = \sqrt{(x+d)^2+(y-h)^2+z^2}$$

Αφού ισχύουν οι προϋποθέσεις θεωρίας μαγν. διπόλου : $\bar{A}_i = \frac{\mu_0}{4\pi}\left(\frac{\bar{m}_i\times\hat{i}_{R_i}}{R_i^2}\right)$, $m_i = I_1 S = I\pi a^2$

όπου $\bar{m}_1 = \hat{i}_y\cdot m_1$, $\hat{i}_{R_1} = \frac{\bar{R}_1}{R_1} = \frac{(x-d)\cdot\hat{i}_x + (y-h)\hat{i}_y + z\cdot\hat{i}_z}{R_1}$

$\bar{m}_2 = -\hat{i}_y\cdot m_2$, $\hat{i}_{R_2} = \frac{\bar{R}_2}{R_2} = \frac{(x-d)\hat{i}_x + (y+h)\hat{i}_y + z\cdot\hat{i}_z}{R_2}$

$\bar{m}_3 = -\hat{i}_y\cdot m_3$, $\hat{i}_{R_3} = \frac{\bar{R}_3}{R_3} = \frac{(x+d)\hat{i}_x + (y+h)\hat{i}_y + z\cdot\hat{i}_z}{R_3}$

$\bar{m}_4 = \hat{i}_y\cdot m_4$, $\hat{i}_{R_4} = \frac{\bar{R}_4}{R_4} = \frac{(x+d)\hat{i}_x + (y-h)\hat{i}_y + z\cdot\hat{i}_z}{R_4}$

$$\Rightarrow \bar{A}(x>0, y>0, z) = \frac{\mu_0 (I\pi a^2)}{4\pi}\left[\frac{-(x-d)\hat{i}_z + z\cdot\hat{i}_x}{R_1^3} + \frac{(x-d)\hat{i}_z - z\hat{i}_x}{R_2^3} + \right.$$

$$\left. + \frac{-(x+d)\hat{i}_z + z\cdot \hat{i}_x}{R_3^3} + \frac{+(x+d)\hat{i}_z - z\cdot\hat{i}_x}{R_4^3} \right]$$

β) Από την θεωρία μαγν. διπόλων: $\bar{H} = \dfrac{3(\bar{m}_i \cdot \hat{i}_{R_i})\hat{i}_{R_i} - \bar{m}_i}{4\pi R_i^3}$

Με επαλληλία: $\bar{H}(x>0, y>0, z) = \bar{H}_1(\Sigma) + \bar{H}_2(\Sigma) + \bar{H}_3(\Sigma) + \bar{H}_4(\Sigma)$

$$\bullet\ \bar{H}_1 = \frac{3\left[(I\pi a^2 \cdot \hat{i}_y)\cdot \frac{\bar{R}_1}{R_1}\right]\cdot \frac{\bar{R}_1}{R_1} - (I\pi a^2 \cdot \hat{i}_y)}{4\pi R_1^3} = \frac{3I\pi a^2\left[(y-h)\bar{R}_1 - R_1^2\hat{i}_y\right]}{4\pi R_1^5} =$$

$$= \frac{1}{4\pi R_1^3}\left\{3\left(\frac{m_0(y-h)}{R_1}\right)\left[\frac{x-d}{R_1}\hat{i}_x + \frac{y-h}{R_1}\hat{i}_y + \frac{z}{R_1}\hat{i}_z\right] - m_0\hat{i}_y\right\}$$

$$\bullet\ \bar{H}_2 = \dots = \frac{3I\pi a^2\left[(y+h)\bar{R}_2 - R_2^2\hat{i}_y\right]}{4\pi R_2^5} = \frac{1}{4\pi R_2^3}\left\{3\left(-\frac{m_0(y+h)}{R_2}\right)\left[\frac{x-d}{R_2}\hat{i}_x + \frac{y+h}{R_2}\hat{i}_y + \frac{z}{R_2}\hat{i}_z\right] + m_0\hat{i}_y\right\}$$

$$\bullet\ \bar{H}_3 = \dots = \frac{3I\pi a^2\left[(y+h)\bar{R}_3 - R_3^2\hat{i}_y\right]}{4\pi R_3^5} = \frac{1}{4\pi R_3^2}\left\{3\left(\frac{m_0(y-h)}{R_3}\right)\left[\frac{x+d}{R_3}\hat{i}_x + \frac{y-h}{R_3}\hat{i}_y + \frac{z}{R_3}\hat{i}_z\right] - m_0\hat{i}_y\right\}$$

$$\bullet\ \bar{H}_4 = \dots = \frac{3I\pi a^2\left[(y-h)\bar{R}_4 - R_4^2\hat{i}_y\right]}{4\pi R_4^5} = \frac{1}{4\pi R_4^2}\left\{3\left(-\frac{m_0(y+h)}{R_4}\right)\left[\frac{x+d}{R_4}\hat{i}_x + \frac{y+h}{R_4}\hat{i}_y + \frac{z}{R_4}\hat{i}_z\right] + m_0\hat{i}_y\right\}$$

Και για $x<0$ ή $y<0 \Rightarrow \bar{H} = 0$

γ) $\bar{K} = \hat{i}_n \times (\bar{H}^+ - \bar{H}^-)$

για $x=0$, $\hat{i}_n = \hat{i}_x \Rightarrow \bar{K}\big|_{x=0} = \hat{i}_x \times \left(\bar{H}^+(x=0) - \cancelto{0}{\bar{H}^-(x=0)}\right) =$

$$= \hat{i}_x \times \frac{3I\pi a^2}{4\pi}\left[\frac{(y-h)\bar{R}_1' - R_1'^2\hat{i}_y}{R_1'^5} + \frac{(y+h)\bar{R}_2' - R_2'^2\hat{i}_y}{R_2'^5} + \frac{(y+h)\bar{R}_3' - R_3'^2\hat{i}_y}{R_3'^5} + \frac{(y-h)\bar{R}_4' - R_4'^2\hat{i}_y}{R_4'^5}\right]$$

όπου $R_1' = R_4' = \sqrt{d^2 + (y-h)^2 + z^2}$

$R_2' = R_3' = \sqrt{d^2 + (y+h)^2 + z^2}$

και $\bar{R}_1' = -d\hat{i}_x + (y-h)\hat{i}_y + z\cdot\hat{i}_z$

$\bar{R}_2' = -d\hat{i}_x + (y+h)\hat{i}_y + z\cdot\hat{i}_z$

$\bar{R}_3' = d\hat{i}_x + (y+h)\hat{i}_y + z\cdot\hat{i}_z$

$R_4' = d\hat{i}_x + (y-h)\hat{i}_y + z\cdot\hat{i}_z$

(*) 2.

για $y=0$, $\hat{i}_n = \hat{i}_y \Rightarrow \bar{K}\big|_{y=0} = \hat{i}_y \times \left(\bar{H}^+(y=0) - \cancel{\bar{H}^-(y=0)}\right) =$

$$= \hat{i}_y \times \frac{3I\pi a^2}{4\pi}\left[\frac{-h\bar{R}_1'' - R_1''^2\hat{i}_y}{R_1''^5} + \frac{h\bar{R}_2'' - \bar{R}_2''\hat{i}_y}{R_2''^5} + \frac{h\bar{R}_3'' - R_3''^2\hat{i}_y}{R_3''^5} + \frac{-h\bar{R}_4'' - R_4''^2\hat{i}_y}{R_4''^5}\right]$$

όπου $R_1'' = R_2'' = \sqrt{(x-d)^2 + h^2 + z^2}$

$R_3'' = R_4'' = \sqrt{(x+d)^2 + h^2 + z^2}$

και $\bar{R}_1'' = (x-d)\hat{i}_x - h\hat{i}_y + z\cdot\hat{i}_z$

$\bar{R}_2'' = (x-d)\hat{i}_x + h\hat{i}_y + z\cdot\hat{i}_z$

$\bar{R}_3'' = (x+d)\hat{i}_x + h\hat{i}_y + z\cdot\hat{i}_z$

$\bar{R}_4'' = (x+d)\hat{i}_x - h\hat{i}_y + z\cdot\hat{i}_z$

$\bar{K}_1\big|_{y=0} = \hat{i}_y \times \left(H_1^+(y=0) - \cancel{H_1^-(y=0)}\right) =$

$$= \frac{1}{4\pi R_1''^3}\left[-3\frac{m_0(-h)}{R_1''}\left(\frac{-d}{R_1''}\right)\hat{i}_z + 3\frac{m_0(-h)}{R_1''}\cdot\left(\frac{z}{R_1''}\right)\hat{i}_x\right]$$

$\bar{K}_3\big|_{y=0} = \hat{i}_y \times \left[H_3^+(y=0) - \cancel{H_3^-(y=0)}\right] =$

$$= \frac{1}{4\pi R_3''^3}\left[3\left(\frac{m_0\cdot h}{R_3''}\right)\left(\frac{x+d}{R_3''}\right)\hat{i}_z + 3\left(\frac{m_0(-h)}{R_3''}\cdot\frac{z}{R_3''}\right)\hat{i}_x\right]$$

Άρα, $K(x,y=0,z) = \left\{\frac{2}{4\pi R_1''^3}\left[3\frac{m_0 h}{R_1''^2}(x-d)\right] + \frac{2}{4\pi R_3''^3}\left[3\frac{m_0 h}{R_3''^2}(x+d)\right]\right\}\hat{i}_z$

$$+ \left\{-\frac{6m_0 h z}{R_1''^5} + \frac{-6m_0 h z}{R_3''^5}\right\}\hat{i}_x$$

(*) $\cdot\ \hat{i}_x \times H_1^+(x=0) = \frac{1}{4\pi R_1^3}\left[3\left(\frac{m_0(y-h)}{R_1}\right)\left(\frac{y-h}{R_1}\right)\hat{i}_z + 3\frac{m_0(y-h)}{R_1}\left(-\frac{z}{R_1}\right)\hat{i}_y - m_0\hat{i}_z\right]$ (ίδιο κ' για H_3)

$\cdot\ \hat{i}_x \times H_2^+(x=0) = \frac{1}{4\pi R_2^3}\left[3\frac{-m_0(y+h)}{R_2}\left(\frac{y+h}{R_2}\right)\hat{i}_z + \frac{3m_0(y+h)}{R_2}\cdot\left(\frac{z}{R_2}\right)\hat{i}_y + m_0\hat{i}_z\right]$ (ίδιο κ' για H_4)

$\cdot\ K(x=0,y,z) = \frac{2}{4\pi R_1'^3}\left[3\left(\frac{m_0(y-h)}{R_1'}\right)\left(\frac{y-h}{R_1'}\right) - m_0\right]\hat{i}_z + \frac{6m_0(y-h)}{R_1'^2}\cdot z\cdot\hat{i}_y +$

$$+ \frac{2}{4\pi R_2'^3}\left[-3\frac{m_0(y+h)}{R_2'}\left(\frac{y+h}{R_2'}\right) + m_0\right]\hat{i}_z + \frac{6m_0(y+h)}{R_2'^2}(-z)\hat{i}_y \rightarrow$$

$$\Rightarrow K(x=0,y,z) = \left\{\frac{2}{4\pi R_1'^3}\left[3\frac{m_0(y-h)^2}{R_1'^2} - m_0\right] + \frac{2}{4\pi R_2'^3}\left[-3\frac{m_0(y+h)^2}{R_2'^2} + m_0\right]\right\}\hat{i}_z +$$

$$+ \left\{\frac{6m_0(y-h)z}{R_1'^5} - \frac{6m_0(y+h)z}{R_2'^5}\right\}\hat{i}_y$$

$8^η$ Άσκηση

α – γ) Η επίλυση φαίνεται στις χειρόγραφες σελίδες.

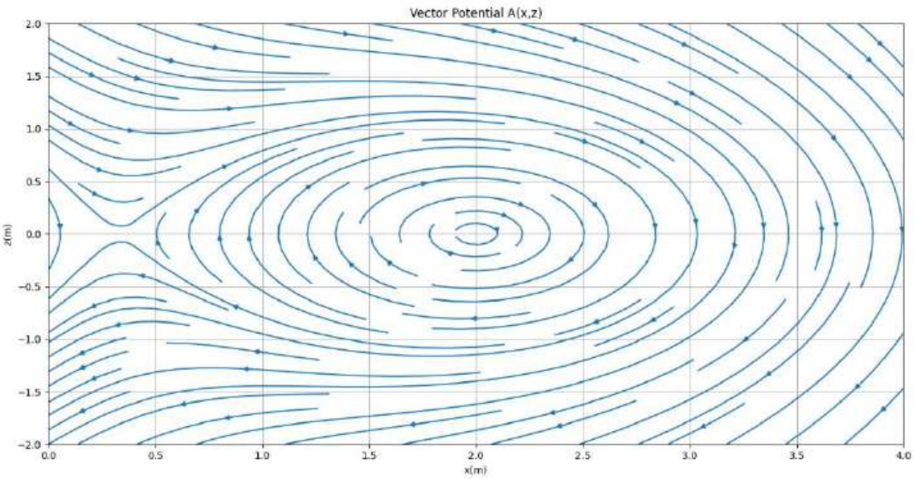
δ) 1ο Σκέλος:

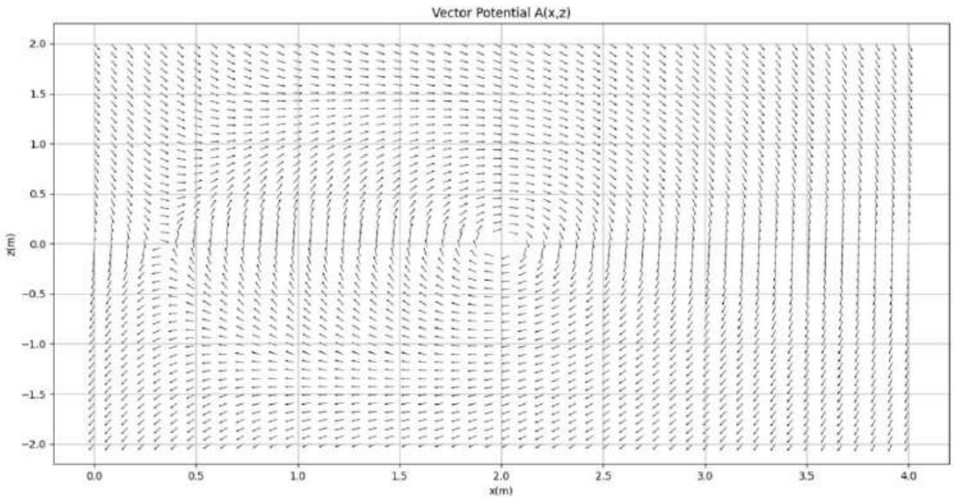
Παρουσιάζεται ο κώδικας **Python**:

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

I = 1
d = 2
h = 1
a = 0.1
M = I*np.pi*a**2
m_0 = 4*np.pi/10000000

def canvas():
    fig, ax = plt.subplots()
    ax.set_title('Vector Potential A(x,z)')
    ax.set_xlabel('x(m)')
    ax.set_ylabel('z(m)')
    ax.grid()
    return ax

def Ax(x, y, z):
    f1 = 1/(np.sqrt((x-d)**2 + (y-h)**2 + z**2))
    f2 = 1/(np.sqrt((x-d)**2 + (y+h)**2 + z**2))
    f3 = 1/(np.sqrt((x+d)**2 + (y-h)**2 + z**2))
    f4 = 1/(np.sqrt((x+d)**2 + (y+h)**2 + z**2))
    return (m_0*M/(4*np.pi))*z*(f1**3 - f2**3 + f3**3 - f4**3)

def Az(x, y, z):
    f1 = 1/(np.sqrt((x-d)**2 + (y-h)**2 + z**2))
    f2 = 1/(np.sqrt((x-d)**2 + (y+h)**2 + z**2))
    f3 = 1/(np.sqrt((x+d)**2 + (y-h)**2 + z**2))
    f4 = 1/(np.sqrt((x+d)**2 + (y+h)**2 + z**2))
    return (m_0*M/(4*np.pi))*(-(x-d)*f1**3 + (x-
d)*f2**3 - (x+d)*f3**3 - (x+d)*f4**3)

def Ax_aux(x, z):
    return Ax(x, 1, z)

def Az_aux(x, z):
    return Az(x, 1, z)

a_x = np.vectorize(Ax_aux)
a_z = np.vectorize(Az_aux)
```

```
x = np.linspace(0, 4, 50)
z = np.linspace(-2, 2, 50)
X, Z = np.meshgrid(x, z)

Ax = a_x(X, Z)
Az = a_z(X, Z)
ax = canvas()
ax.streamplot(X, Z, Ax/(2*((Ax**2 + Az**2)**0.5)),
              Az/(2*((Ax**2 + Az**2)**0.5)))
plt.show()
ax = canvas()
ax.quiver(X, Z, Ax/(2*((Ax**2 + Az**2)**0.5)), Az/(2*((Ax**2 + Az**2)**0.5)),
          (Ax**2 + Az**2)**0.5, cmap='inferno', units='xy', width=0.0035,
          headwidth=3., headlength=4.)
plt.show()
```

Παρουσιάζονται οι γραφικές παραστάσεις:



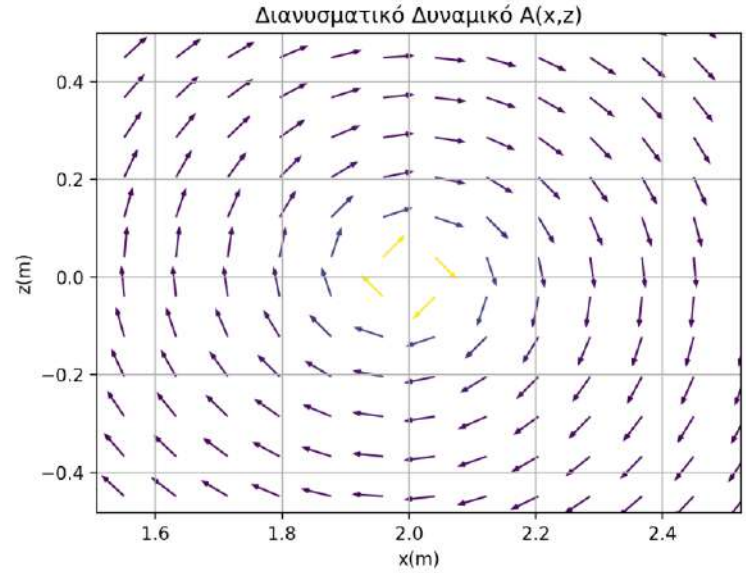


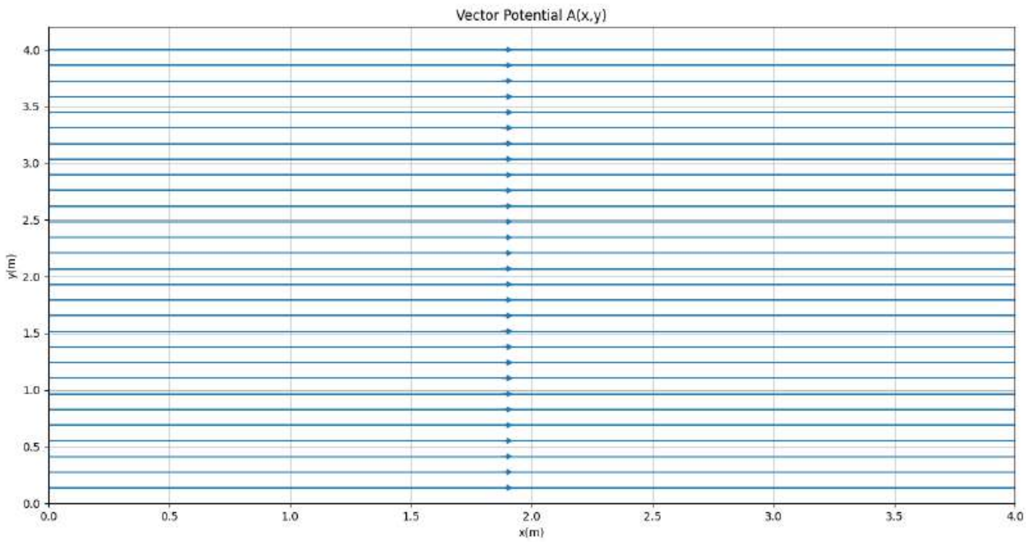
2° Σκέλος:

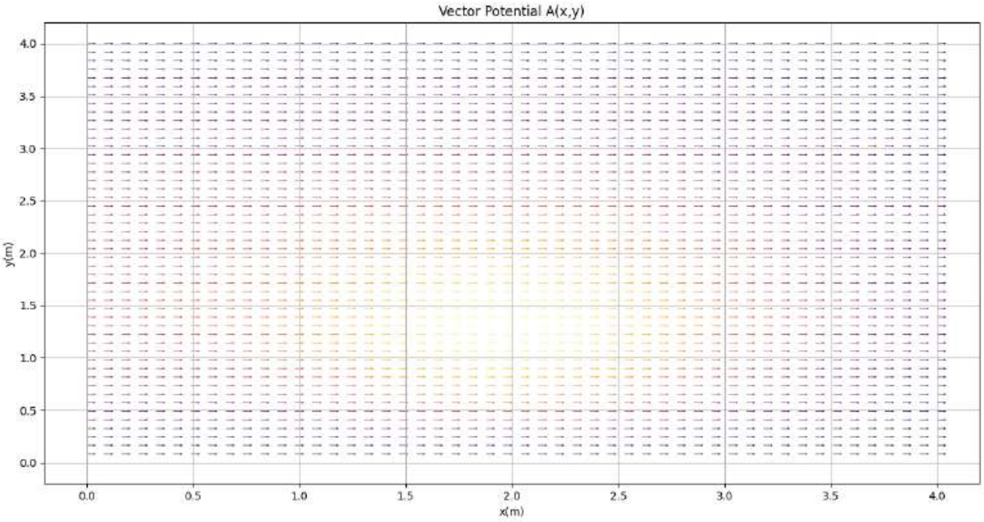
Παρουσιάζεται ο κώδικας **Python**:

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

I = 1
d = 2
a = 0.1
h = 1
M = I*np.pi*a**2
m_0 = 4*np.pi/10000000

def canvas():
    fig, ax = plt.subplots()
    ax.set_title('Vector Potential A(x,y)')
    ax.set_xlabel('x(m)')
    ax.set_ylabel('y(m)')
    ax.grid()
    return ax

def Ax(x, y, z):
    f1 = 1/(np.sqrt((x-d)**2 + (y-h)**2 + z**2))
    f2 = 1/(np.sqrt((x-d)**2 + (y+h)**2 + z**2))
    f3 = 1/(np.sqrt((x+d)**2 + (y-h)**2 + z**2))
    f4 = 1/(np.sqrt((x+d)**2 + (y+h)**2 + z**2))
    return (m_0*M/(4*np.pi))*z*(f1**3 - f2**3 + f3**3 - f4**3)

def Ax_aux(x, y):
    return Ax(x, y, 2)

a_x = np.vectorize(Ax_aux)
x = np.linspace(0, 4, 50)
y = np.linspace(0, 4, 50)
X, Y = np.meshgrid(x, y)
Ax = a_x(X, Y)
numrows = len(Ax)
numcols = len(Ax[0])
Ay = np.array([[0 for i in range(numcols)] for i in range(numrows)])
ax = canvas()
ax.streamplot(X, Y, Ax/(2*((Ax**2 + Ay**2)**0.5)),
              Ay/(2*((Ax**2 + Ay**2)**0.5)))
plt.show()
ax = canvas()
ax.quiver(X, Y, Ax/(2*((Ax**2 + Ay**2)**0.5)), Ay/(2*((Ax**2 + Ay**2)**0.5)),
          (Ax**2 + Ay**2)**0.5, cmap='inferno', units='xy', width=0.0035,
          headwidth=3., headlength=4.)
plt.show()
```

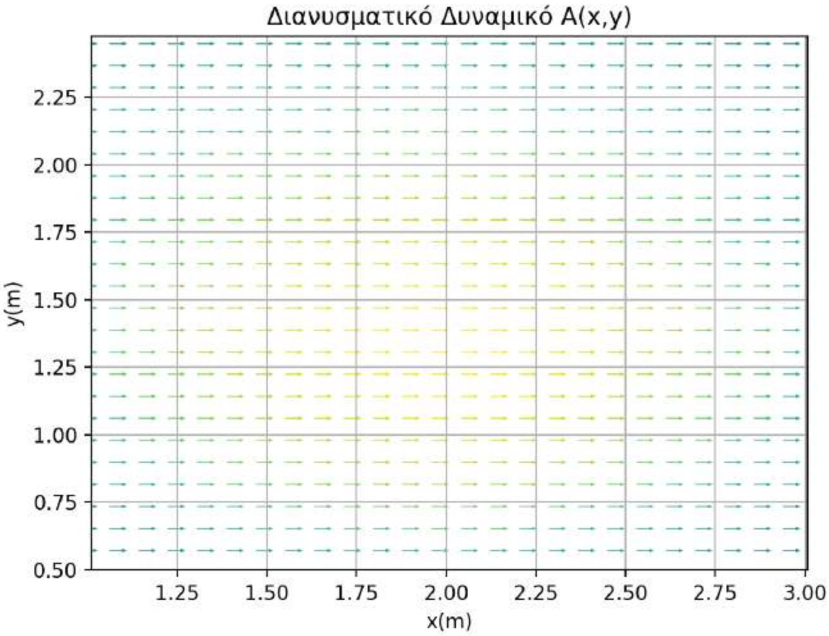
Παρουσιάζονται οι γραφικές παραστάσεις:







Vector Potential A(x,y)
y(m)
x(m)

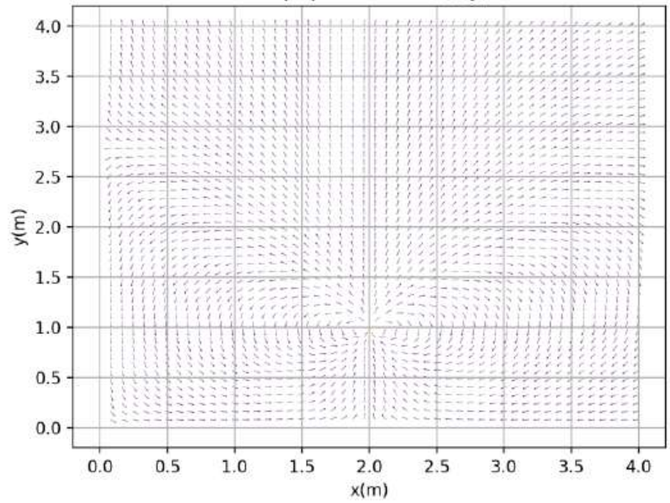


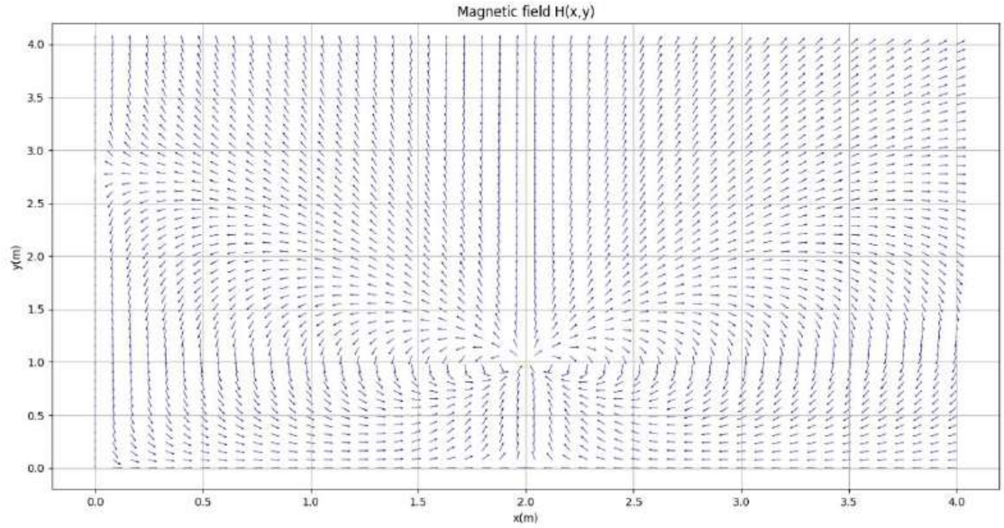
Διανυσματικό Δυναμικό A(x,y)
y(m)
x(m)

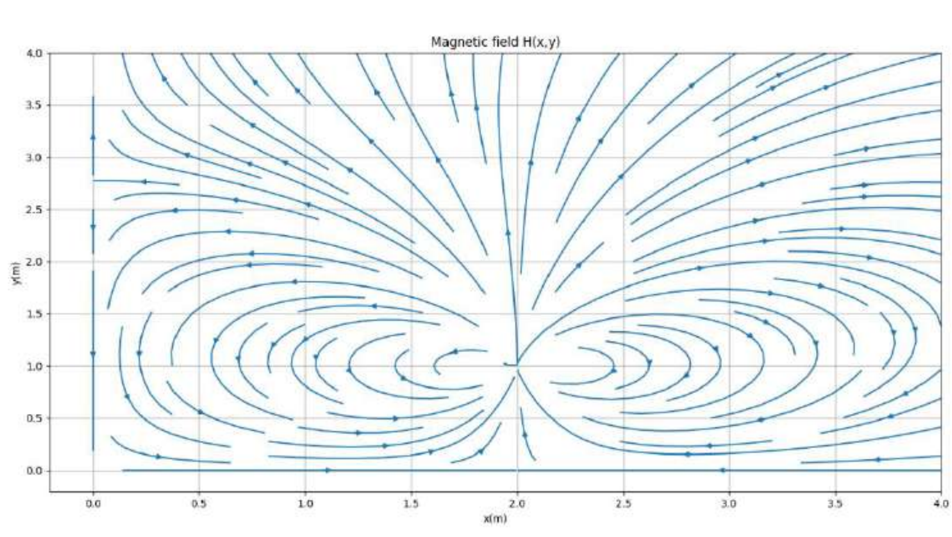
ε) Παρουσιάζεται ο κώδικας **Python**:

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

a = 0.1
I = 1
d = 2
h = 1
M = I*np.pi*a**2
def canvas():
    fig, ax = plt.subplots()
    ax.set_title('Magnetic field H(x,y)')
    ax.set_xlabel('x(m)')
    ax.set_ylabel('y(m)')
    ax.grid()
    return ax
def Hx(x, y):
    f1 = 1/(np.sqrt((x-d)**2 + (y-h)**2))
    f2 = 1/(np.sqrt((x-d)**2 + (y+h)**2))
    f3 = 1/(np.sqrt((x+d)**2 + (y-h)**2))
    f4 = 1/(np.sqrt((x+d)**2 + (y+h)**2))
    return (M/(4*np.pi)) * 3 * ((x-d)*(y-h)*(f1**5) -
    (x-d)*(y+h)*(f2**5) + (x+d)*(y-h)*(f3**5) - (x+d)*(y+h)*(f4**5))
def Hy(x, y):
    f1 = (np.sqrt((x-d)**2 + (y-h)**2))
    f2 = (np.sqrt((x-d)**2 + (y+h)**2))
    f3 = (np.sqrt((x+d)**2 + (y-h)**2))
    f4 = (np.sqrt((x+d)**2 + (y+h)**2))
    return (M/(4*np.pi)) * ((3*(y-h)**2-f1**2)/(f1**5) -
    (3*(y+h)**2-f2**2)/(f2**5) + (3*(y-h)**2-f3**2)/(f3**5) - (3*(y+h)**2-
f4**2)/(f4**5))

h_x = np.vectorize(Hx)
h_y = np.vectorize(Hy)
x = np.linspace(0, 4, 50)
y = np.linspace(0, 4, 50)
X, Y = np.meshgrid(x, y)
Hx = h_x(X, Y)
Hy = h_y(X, Y)
ax = canvas()
ax.quiver(X, Y, Hx/(2*((Hx**2 + Hy**2)**0.5)), Hy/(2*((Hx**2 + Hy**2)**0.5)),
          (Hx**2 + Hy**2)**0.5, cmap='plasma', units='xy', width=0.0035,
          headwidth=3., headlength=4.)
plt.show()
ax = canvas()
ax.streamplot(X, Y, Hx/(2*((Hx**2 + Hy**2)**0.5)),
              Hy/(2*((Hx**2 + Hy**2)**0.5)))
plt.show()
```

Παρουσιάζονται οι γραφικές παραστάσεις:





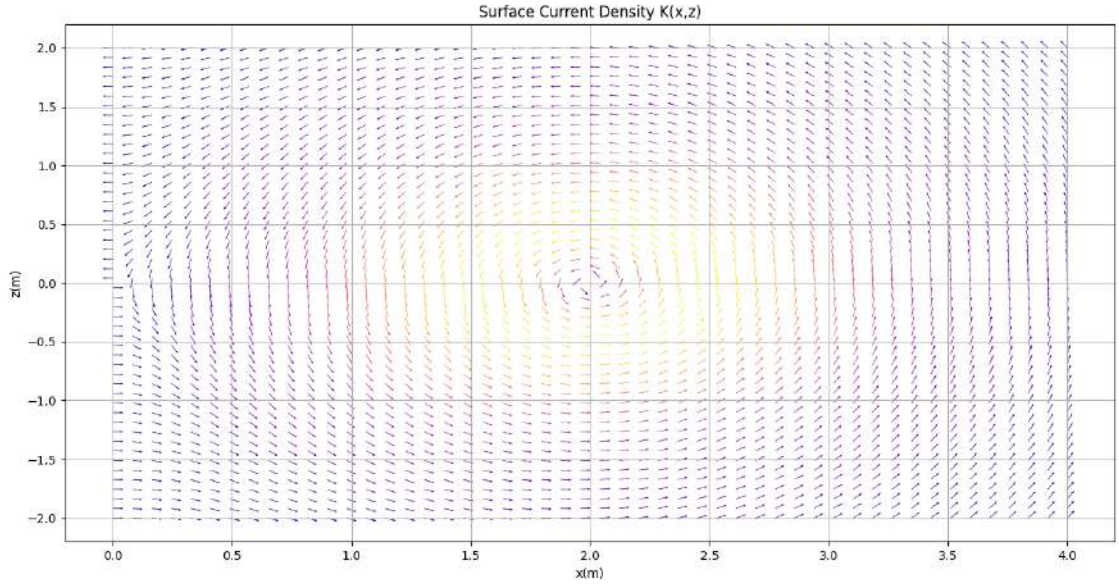


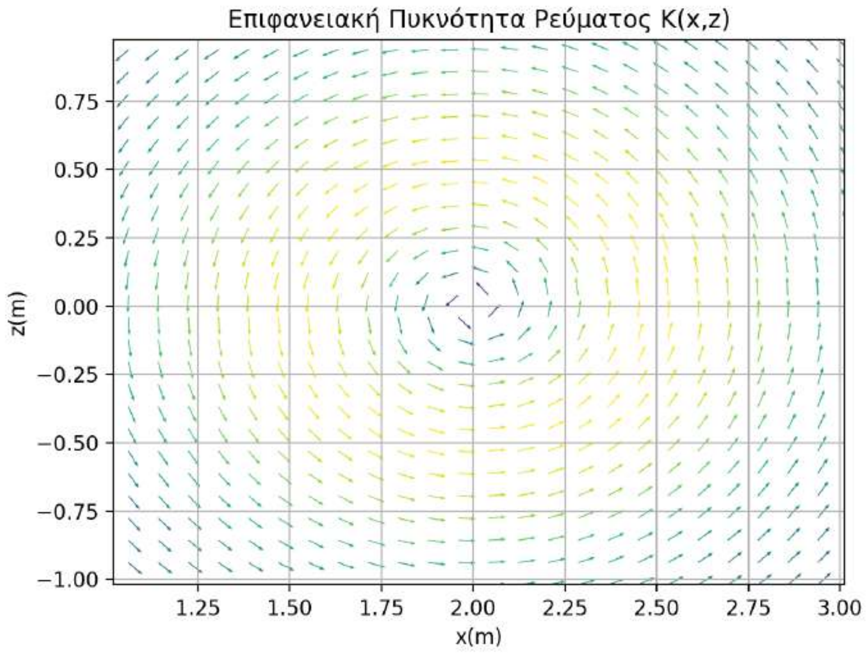
στ) Παρουσιάζεται ο κώδικας **Python**:

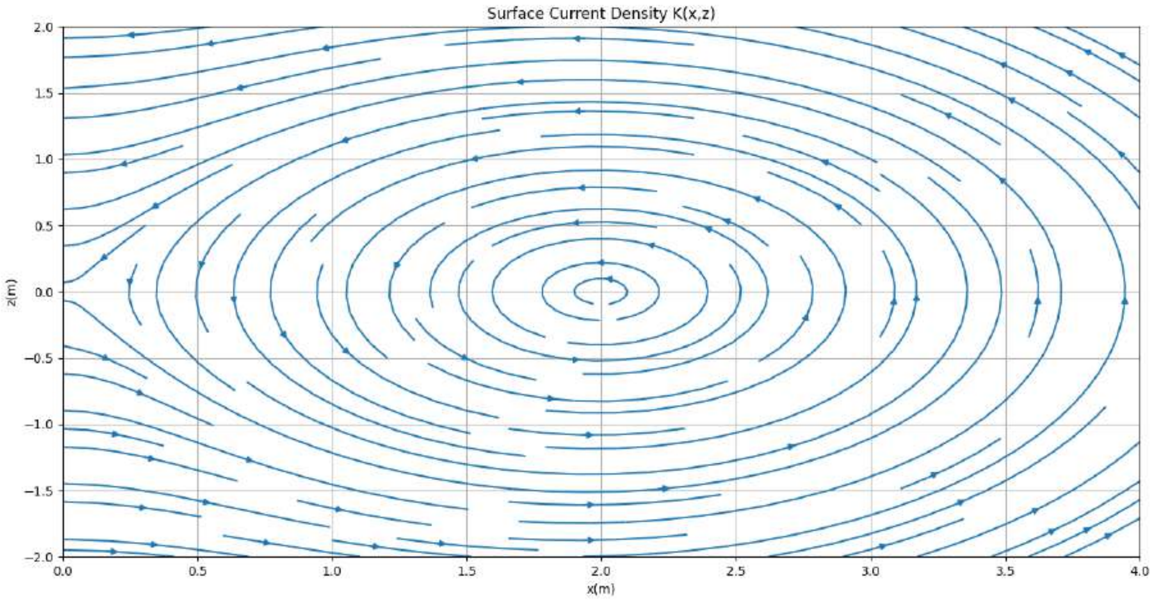
```
import matplotlib.pyplot as plt
import numpy as np

a = 0.1
I = 1
d = 2
h = 1
M = I*np.pi*a**2

def canvas1():
    fig, ax = plt.subplots()
    ax.set_title('Surface Current Density K(x,z)')
    ax.set_xlabel('x(m)')
    ax.set_ylabel('z(m)')
    ax.grid()
    return ax

def canvas2():
    fig, ax = plt.subplots()
    ax.set_title('Surface Current Density K(y,z)')
    ax.set_xlabel('y(m)')
    ax.set_ylabel('z(m)')
    ax.grid()
    return ax

def Kx(x, z):
    f1 = 1/(np.sqrt((x-d)**2 + h**2 + z**2))
    f2 = 1/(np.sqrt((x+d)**2 + h**2 + z**2))
    return M/(4*np.pi) * 6*h*z*((-1)*(f1**5 + f2**5))

def Kz_1(x, z):
    f1 = 1/(np.sqrt((x-d)**2 + h**2 + z**2))
    f2 = 1/(np.sqrt((x+d)**2 + h**2 + z**2))
    return M/(4*np.pi) * 6*h*((x-d)*f1**5 + (x+d)*f2**5)

def Ky(y, z):
    f1 = 1/(np.sqrt(d**2 + (y-h)**2 + z**2))
    f2 = 1/(np.sqrt(d**2 + (y+h)**2 + z**2))
    return M/(4*np.pi) * 6*z*((-1)*((y-h)*f1**5 - (y+h)*f2**5))

def Kz_2(y, z):
    f1 = (np.sqrt(d**2 + (y-h)**2 + z**2))
    f2 = (np.sqrt(d**2 + (y+h)**2 + z**2))
    return M/(4*np.pi) * 2*((3*(y-h)**2-f1**2)/f1**5 - (3*(y+h)**2-f2**2)/f2**5)

Kx = np.vectorize(Kx)
Kz_2 = np.vectorize(Kz_2)
```

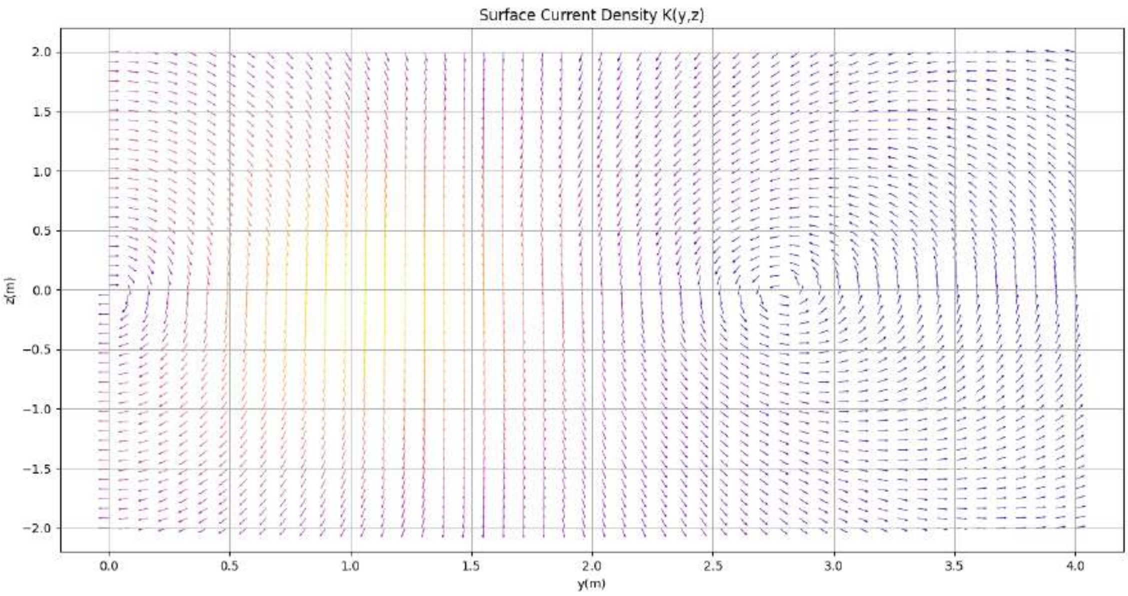
```python
Ky = np.vectorize(Ky)
Kz_1 = np.vectorize(Kz_1)

x = np.linspace(0, 4, 50)
y = np.linspace(0, 4, 50)
z = np.linspace(-2, 2, 50)

# y = 0
X, Z = np.meshgrid(x, z)
Kx = Kx(X, Z)
Kz = Kz_1(X, Z)
ax = canvas1()
ax.quiver(X, Z, Kx/(2*((Kx**2 + Kz**2)**0.5)), Kz/(2*((Kx**2 + Kz**2)**0.5)),
          (Kx**2 + Kz**2)**0.5, cmap='plasma', units='xy', width=0.0035,
          headwidth=3., headlength=4.)
plt.show()
ax = canvas1()
ax.streamplot(X, Z, Kx/(2*((Kx**2 + Kz**2)**0.5)), Kz/(2*((Kx**2 + Kz**2)**0.5)))
plt.show()

# x = 0
Y, Z = np.meshgrid(y, z)
Ky = Ky(Y, Z)
Kz = Kz_2(Y, Z)
ax = canvas2()
ax.quiver(Y, Z, Ky/(2*((Ky**2 + Kz**2)**0.5)), Kz/(2*((Ky**2 + Kz**2)**0.5)),
          (Ky**2 + Kz**2)**0.5, cmap='plasma', units='xy', width=0.0035,
          headwidth=3., headlength=4.)
plt.show()
ax = canvas2()
ax.streamplot(Y, Z, Ky/(2*((Ky**2 + Kz**2)**0.5)), Kz/(2*((Ky**2 + Kz**2)**0.5)))
plt.show()
```

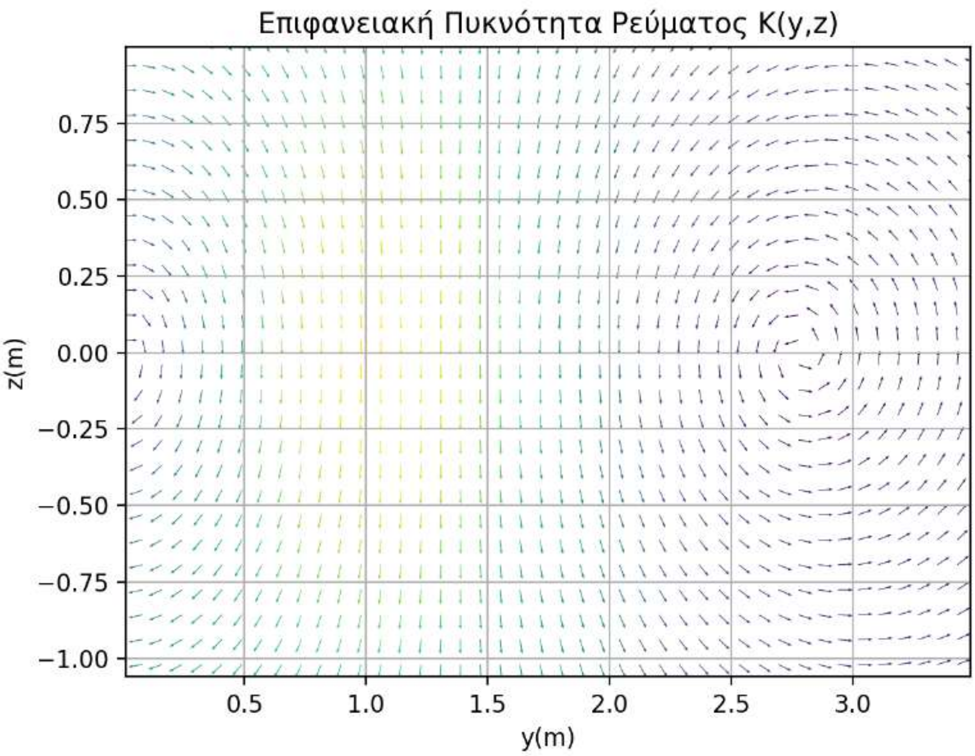
Παρουσιάζονται οι γραφικές παραστάσεις:

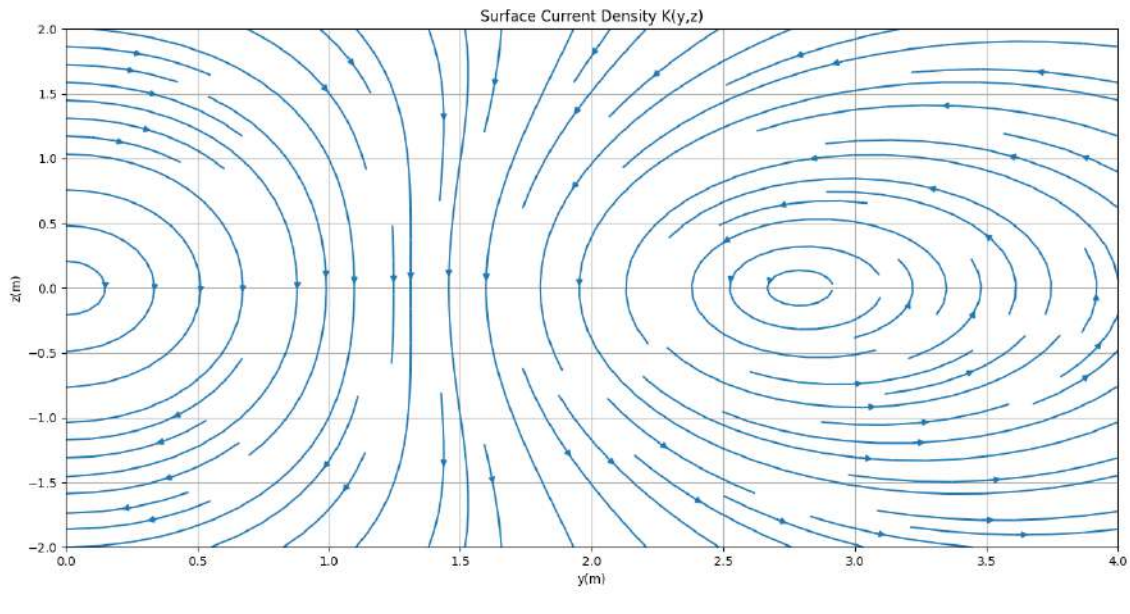




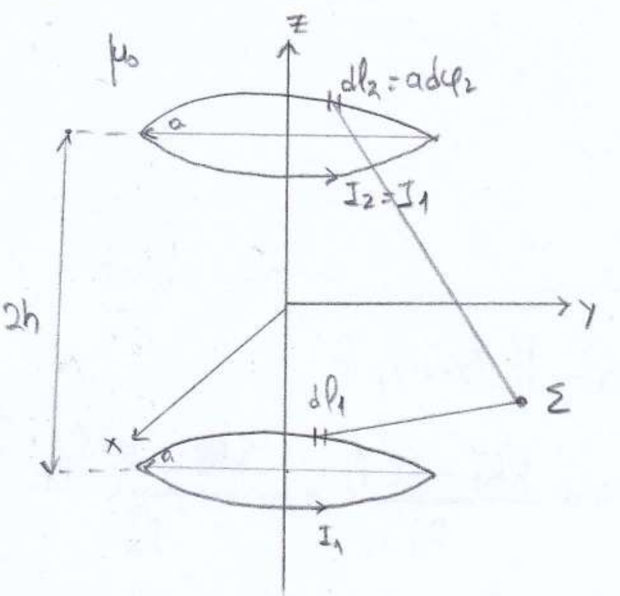

Surface Current Density K(x,z)
z(m)
x(m)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Surface Current Density K(y,z)
z(m)
y(m)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0


Επιφανειακή Πυκνότητα Ρεύματος K(y,z)
z(m)
y(m)
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0.5
1.0
1.5
2.0
2.5
3.0


Surface Current Density K(y,z)
z(m)
y(m)
2.0
1.5
1.0
0.5
0.0
−0.5
−1.0
−1.5
−2.0
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

Άσκηση 9



α) $R_1 = \sqrt{(x-a\cos\varphi)^2 + (y-a\sin\varphi)^2 + (z+h)^2}$

$R_2 = \sqrt{(x-a\cos\varphi)^2 + (y-a\sin\varphi)^2 + (z-h)^2}$

$\bar{d\ell_1} = a\,\bar{d\varphi_1} = a\,d\varphi_1 \cdot \hat{i}_\varphi = a\,d\varphi_1 \left(-\sin\varphi_1 \cdot \hat{i}_x + \cos\varphi_1 \cdot \hat{i}_y\right)$

$\bar{d\ell_2} = a\,d\varphi_2 \left(-\sin\varphi_2 \cdot \hat{i}_x + \cos\varphi_2 \cdot \hat{i}_y\right)$

Με επαλληλία:

$$\bar{A}(\Sigma) = \bar{A}_1(\Sigma) + \bar{A}_2(\Sigma) = \frac{\mu_0}{4\pi}\int_{C_1} I\,\bar{d\ell_1}\left(\frac{1}{R_1} - \frac{1}{R_0^2}\right) + \frac{\mu_0}{4\pi}\int_{C_2} I\,\bar{d\ell_2}\left(\frac{1}{R_2} - \frac{1}{R_0^2}\right) =$$

$$= \frac{\mu_0 I}{4\pi}\left[\int_0^{2\pi} \frac{a\,d\varphi_1\left(-\sin\varphi_1 \hat{i}_x + \cos\varphi_1 \hat{i}_y\right)}{\sqrt{x^2+y^2+a^2-2ax\cos\varphi_1-2ay\sin\varphi_1+(z+h)^2}} + \int_0^{2\pi} \frac{a\,d\varphi_2\left(-\sin\varphi_2 \hat{i}_x + \cos\varphi_2 \hat{i}_y\right)}{\sqrt{x^2+y^2+a^2-2ax\cos\varphi_2-2ay\sin\varphi_2+(z-h)^2}}\right]$$

β) $\bar{B} = \bar{\nabla}\times\bar{A} = \mu_0 \bar{H} \Rightarrow \bar{H} = \frac{1}{\mu_0}\bar{\nabla}\times\bar{A}$

Χρησιμοποιώντας το λυμένο πρόβλημα για ένα κυκλικό βρόχο και με επαλληλία για την περίπτωση των δύο βρόχων, προκύπτει:

$$H_r(r,\theta) = H_{r_1}(r_1,\theta_1) + H_{r_2}(r_2,\theta_2) = \frac{1}{\pi}\frac{a^2\cos\theta_1}{A_1^2 B_1}E(L_1) + \frac{1}{\pi}\frac{a^2\cos\theta_2}{A_2^2 B_2}E(L_2)$$

$$H_\varphi(r,\theta) = \frac{1}{\pi}\frac{1}{2A_1^2 B_1\sin\theta_1}\left[(r_1^2 + a^2\cos 2\theta_1)E(L_1) - A_1^2 K(L_1)\right] +$$

$$+ \frac{1}{\pi}\frac{1}{2A_2^2 B_2\sin\theta_2}\left[(r_2^2 + a^2\cos 2\theta_2)E(L_2) - A_2^2 K(L_2)\right]$$

όπου $A_i^2 = r_i^2 + a^2 - 2ar_i \sin\theta_i$

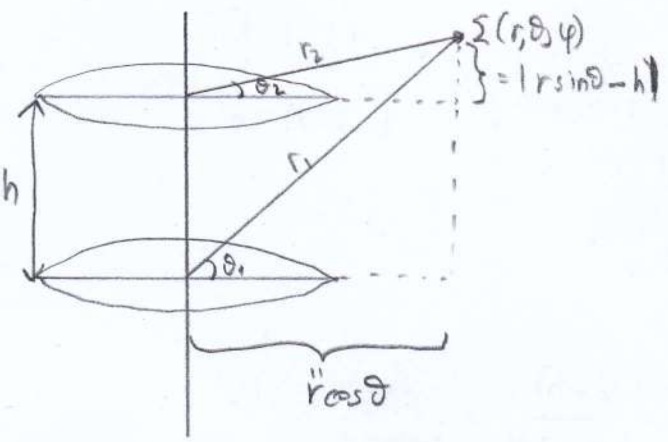
$B_i^2 = r_i^2 + a^2 + 2ar_i \sin\theta_i$

$L = 1 - \dfrac{A_i^2}{B_i^2}$

Για μετατροπή σε καρτεσιανές συν/νες:
$$r = \sqrt{x^2+y^2+z^2}, \quad \theta = \cos^{-1}\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right), \quad \varphi = \ldots$$
$$\left\{\begin{aligned} \hat{i}_r &= \sin\theta\cos\varphi\, \hat{i}_x + \sin\theta\sin\varphi\, \hat{i}_y + \cos\theta \cdot \hat{i}_z \\ \hat{i}_\varphi &= -\sin\varphi\, \hat{i}_x + \cos\varphi\, \hat{i}_y \end{aligned}\right.$$

Και $r_1, \theta_1, r_2, \theta_2$:

- Το λυμένο πρόβλημα θεωρεί σημείο αναφοράς το $(x=0, y=0, z=0)$ όμως σε αυτή την περίπτωση το σημείο αναφοράς του βρόχου 1 είναι το $(x=0, y=0, z=-h)$ και του 2 είναι το $(x=0, y=0, z=h)$, οπότε θα ισχύει.



Άρα
$$r_2(r,\theta) = \sqrt{(r\cos\theta)^2 + (r\sin\theta - h)^2} \qquad r_1(r,\theta) = \sqrt{(r\cos\theta)^2 + (r\sin\theta + h)^2}$$
$$\theta_2(r,\theta) = \sin^{-1}\left(\frac{r\sin\theta - h}{r\cos\theta}\right) \qquad \theta_2(r,\theta) = \sin^{-1}\left(\frac{r\sin\theta + h}{r\cos\theta}\right)$$

γ)
$$\bar{H}(x,y,z) = \ldots = \frac{I}{4\pi}\Bigg[\int_{\varphi_1=0}^{2\pi} \frac{\left[(z+h)\cos\varphi_1\, \hat{i}_x + (z+h)\sin\varphi_1\, \hat{i}_y - (x\cos\varphi_1 + y\sin\varphi_1 - a)\,\hat{i}_z\right] a\, d\varphi_1}{\left[x^2+y^2+a^2 - 2a(x\cos\varphi_1 + y\sin\varphi_1) + (z+h)^2\right]^{3/2}}$$
$$+ \int_{\varphi_2=0}^{2\pi} \frac{\left[(z-h)\cos\varphi_2 \cdot \hat{i}_x + (z-h)\sin\varphi_2\, \hat{i}_y - (x\cos\varphi_2 + y\sin\varphi_2 - a)\,\hat{i}_z\right] a\, d\varphi_2}{\left[x^2+y^2+a^2 - 2a(x\cos\varphi_2 + y\sin\varphi_2) + (z-h)^2\right]^{3/2}}\Bigg]$$

γ1. $H_z(z)$, $x=y=0$: $H_z(z) = \dfrac{I}{4\pi}\left[\displaystyle\int_{\varphi_1=0}^{2\pi} \dfrac{a^2\, d\varphi_2}{[a^2+(z+h)^2]^{3/2}} + \int_{\varphi_2=0}^{2\pi} \dfrac{a^2\, d\varphi_2}{[a^2+(z-h)^2]^{3/2}}\right] \Rightarrow$

$$\Rightarrow H_z(z) = \frac{Ia^2}{_2\cancel{4\pi}}\left[\frac{2\pi}{[a^2+(z+h)^2]^{3/2}} + \frac{2\pi}{[a^2+(z-h)^2]^{3/2}}\right] = \frac{Ia^2}{2}\left[\frac{1}{[a^2+(z+h)^2]^{3/2}} + \frac{1}{[a^2+(z-h)^2]^{3/2}}\right]$$

Σειρά Taylor γύρω από το 0 για την $H_z(z)$:

$$H_z(z) = H_z(0) + \frac{H_z'(0)}{1!}(z-0) + \frac{H_z''(0)}{2!}(z-0)^2 + \frac{H_z'''(0)}{3!}(z-0)^3 + \frac{H^{(4)}(0)}{4!}(z-0)^4 + \dots$$

- $H_z(0) = \dfrac{Ia^2}{\cancel{2}} \cdot \dfrac{\cancel{2}}{(a^2+h^2)^{3/2}} = \dfrac{Ia^2}{(a^2+h^2)^{3/2}}$

- $H_z'(z) = -\dfrac{Ia^2}{2}\left[\dfrac{3(z+h)}{[(z+h)^2+a^2]^{5/2}} + \dfrac{3(z-h)}{[(z-h)^2+a^2]^{5/2}}\right] \Rightarrow H_z'(0) = 0$

- $H_z''(z) = -\dfrac{3}{[(z+h)^2+a^2]^{5/2}} + \dfrac{15(z+h)^2}{[(z+h)^2+a^2]^{7/2}} - \dfrac{3}{[(z-h)^2+a^2]^{5/2}} + \dfrac{15(z-h)^2}{[(z-h)^2+a^2]^{7/2}} \Rightarrow$

$$\Rightarrow H_z''(0) = -\frac{6}{[h^2+a^2]^{5/2}} + \frac{30h^2}{[h^2+a^2]^{7/2}}$$

- $H_z'''(z) = \dfrac{45(z+h)}{[(z+h)^2+a^2]^{7/2}} - \dfrac{105(z+h)^3}{[(z+h)^2+a^2]^{9/2}} + \dfrac{45(z-h)}{[(z-h)^2+a^2]^{7/2}} - \dfrac{105(z-h)^3}{[(z-h)^2+a^2]^{9/2}} \Rightarrow$

$$\Rightarrow H_z'''(0) = 0$$

- $H_z^{(4)}(z) = \dfrac{45}{[(z+h)^2+a^2]^{7/2}} - \dfrac{630(z+h)^2}{[(z+h)^2+a^2]^{9/2}} + \dfrac{945(z+h)^4}{[(z+h)^2+a^2]^{11/2}} +$

$$+ \frac{45}{[(z-h)^2+a^2]^{7/2}} - \frac{630(z-h)^2}{[(z-h)^2+a^2]^{9/2}} + \frac{945(z-h)^4}{[(z-h)^2+a^2]^{11/2}} \Rightarrow$$

$$\Rightarrow H_z^{(4)}(0) = \frac{90}{(h^2+a^2)^{7/2}} - \frac{1260h^2}{(h^2+a^2)^{9/2}} + \frac{1890h^4}{(h^2+a^2)^{11/2}}$$

$\Rightarrow H_z(z) = H_z(0) + \dfrac{H_z''(0)}{2}z^2 + \dfrac{H_z^{(4)}(0)}{24}z^4 + \dots$ (δηλ. οι περιττοί όροι είναι μηδέν)

<u>9η Άσκηση</u>

α – γ) Η επίλυση φαίνεται στις χειρόγραφες σελίδες.

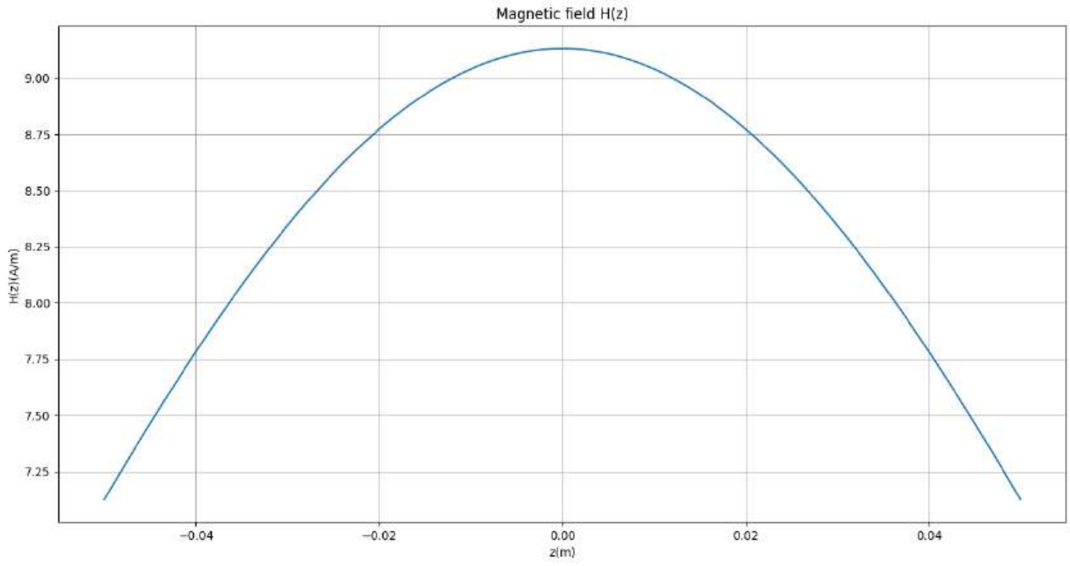
δ) Παρατίθεται ο κώδικας **Python**:

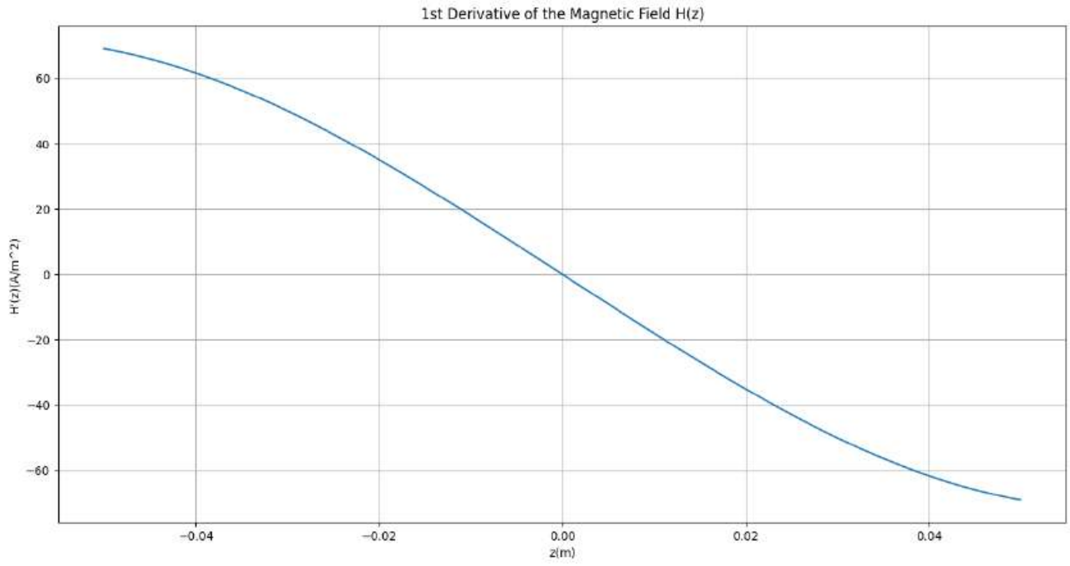
```
import matplotlib.pyplot as plt
import numpy as np

a = 0.1
I1 = 1
h = 0.05
# h = 0.025
# h = 0.1

def canvas():
    fig, ax = plt.subplots()
    ax.grid()
    return ax

def H(z):
    f1 = 1/((a**2 + (z-h)**2)**1.5)
    f2 = 1/((a**2 + (z+h)**2)**1.5)
    return (I1*a**2/2) * (f1 + f2)

def der_H(z):
    f1 = 1/((a**2 + (z-h)**2)**2.5)
    f2 = 1/((a**2 + (z+h)**2)**2.5)
    return (I1*a**2/2) * (-3)*((z-h)*f1 + (z+h)*f2)

def der2_H(z):
    f1 = 1/((a**2 + (z-h)**2)**3.5)
    f2 = 1/((a**2 + (z+h)**2)**3.5)
    f3 = (a**2 - 4*(z-h)**2)
    f4 = (a**2 - 4*(z+h)**2)
    return (I1*a**2/2) * (-3)*(f3*f1 + f4*f2)

def der3_H(z):
    f1 = 1/((a**2 + (z-h)**2)**4.5)
    f2 = 1/((a**2 + (z+h)**2)**4.5)
    f3 = (3*a**2 - 4*(z-h)**2)
    f4 = (3*a**2 - 4*(z+h)**2)
    return (I1*a**2/2) * 15*((z-h)*f3*f1 + (z+h)*f4*f2)

def der4_H(z):
    f1 = 1/((a**2 + (z-h)**2)**5.5)
    f2 = 1/((a**2 + (z+h)**2)**5.5)
    f3 = (a**4 - 12*a**2*(z-h)**2 + 8*(z-h)**4)
    f4 = (a**4 - 12*a**2*(z+h)**2 + 8*(z+h)**4)
```

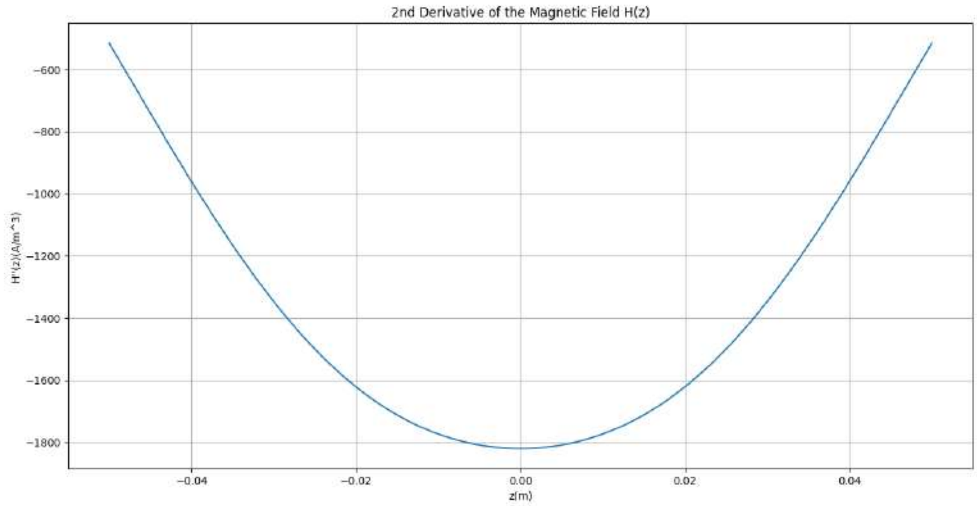
```python
    return (I1*a**2/2) * 45*(f3*f1 + f4*f2)

H = np.vectorize(H)
der_H = np.vectorize(der_H)
der2_H = np.vectorize(der2_H)
der3_H = np.vectorize(der3_H)
der4_H = np.vectorize(der4_H)

z = np.linspace(-2*h, 2*h, 1000)  # 3 times for h = 0.025, 0.05, 0.1

ax = canvas()
ax.set(xlabel='z(m)', ylabel="H(z)(A/m)", title='Magnetic field H(z)')
ax.plot(z, H(z))
plt.show()

ax = canvas()
ax.set(xlabel='z(m)', ylabel="H'(z)(A/m^2)",
       title='1st Derivative of the Magnetic Field H(z)')
ax.plot(z, der_H(z))
plt.show()

ax = canvas()
ax.set(xlabel='z(m)', ylabel="H''(z)(A/m^3)",
       title='2nd Derivative of the Magnetic Field H(z)')
ax.plot(z, der2_H(z))
plt.show()

ax = canvas()
ax.set(xlabel='z(m)', ylabel="H^(3)(z)(A/m^4)",
       title='3rd Derivative of the Magnetic Field H(z)')
ax.plot(z, der3_H(z))
plt.show()

ax = canvas()
ax.set(xlabel='z(m)', ylabel="H^(4)(z)(A/m^5)",
       title='4th Derivative of the Magnetic Field H(z)')
ax.plot(z, der4_H(z))
plt.show()
```

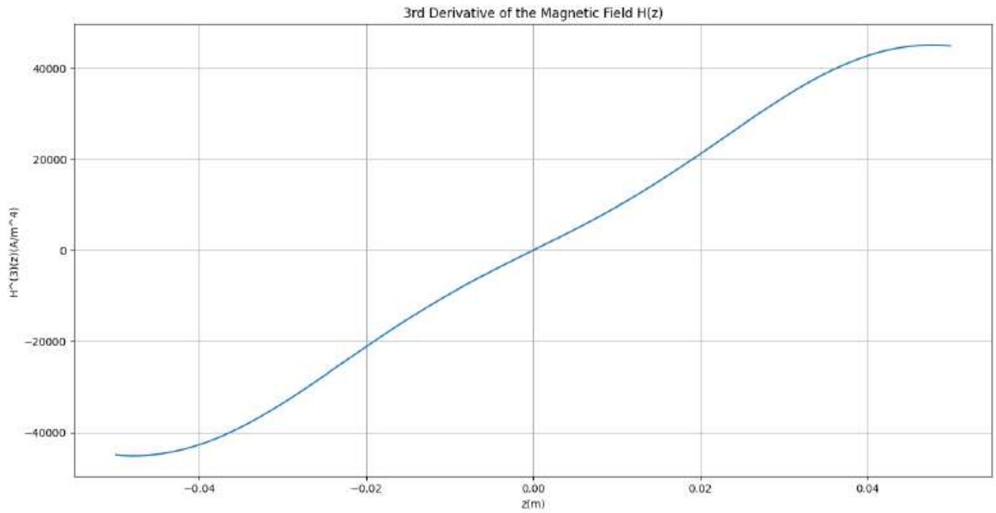
Παρουσιάζονται οι γραφικές παραστάσεις:

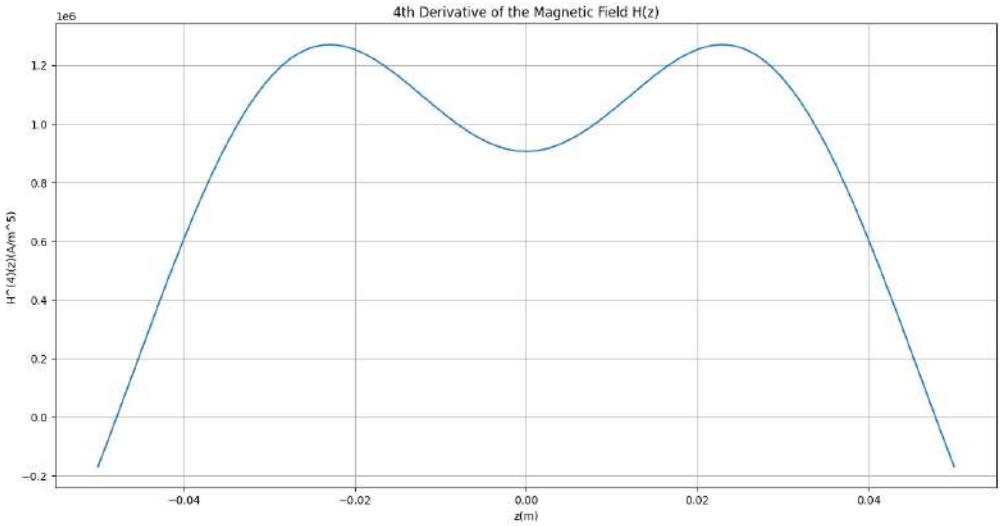
Για h = 0.025:



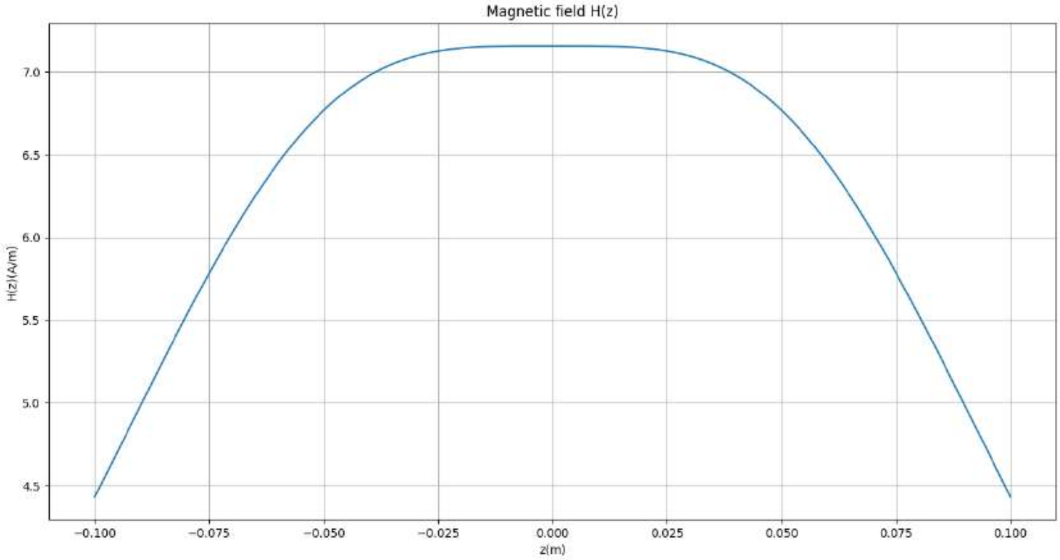



2nd Derivative of the Magnetic Field H(z)
H''(z)(A/m^3)
z(m)

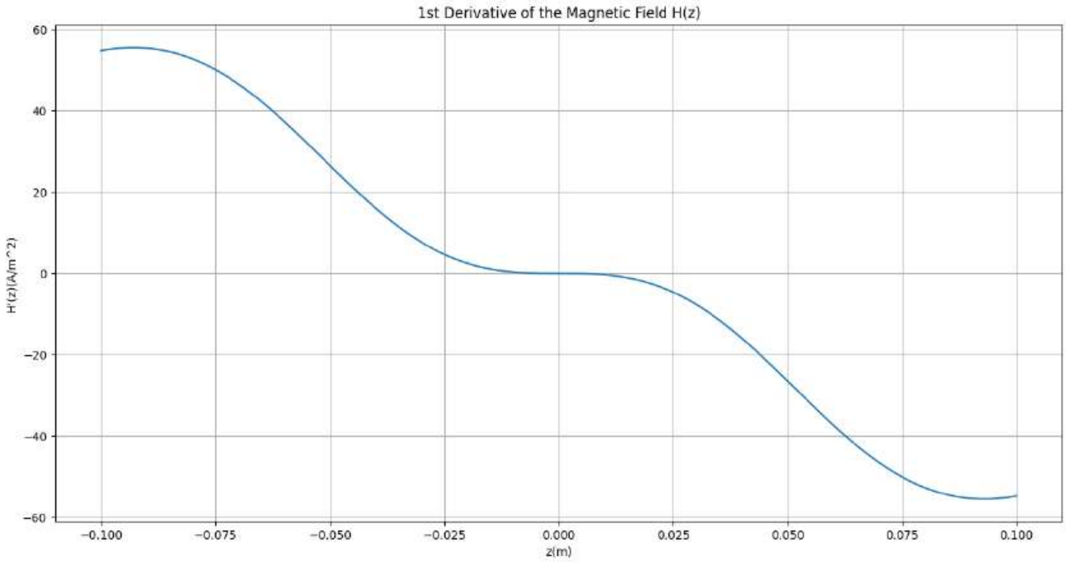

3rd Derivative of the Magnetic Field H(z)
H^(3)(z)(A/m^4)
z(m)


4th Derivative of the Magnetic Field H(z)
1e6
H^(4)(z)(A/m^5)
z(m)

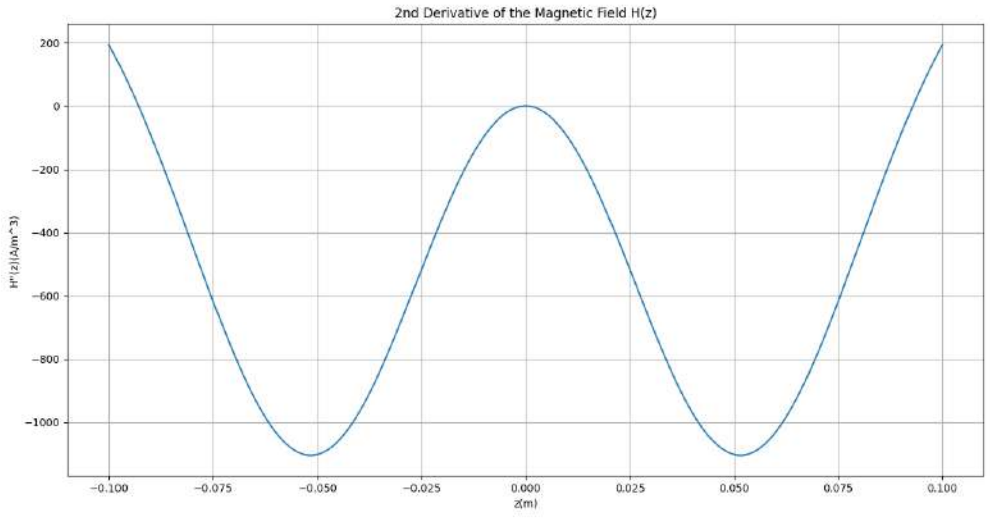
Για h = 0.05:

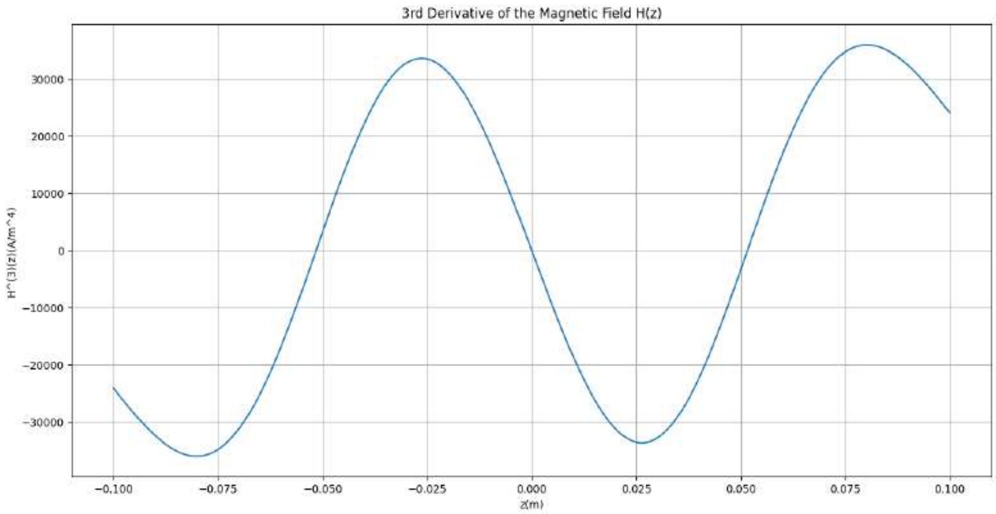


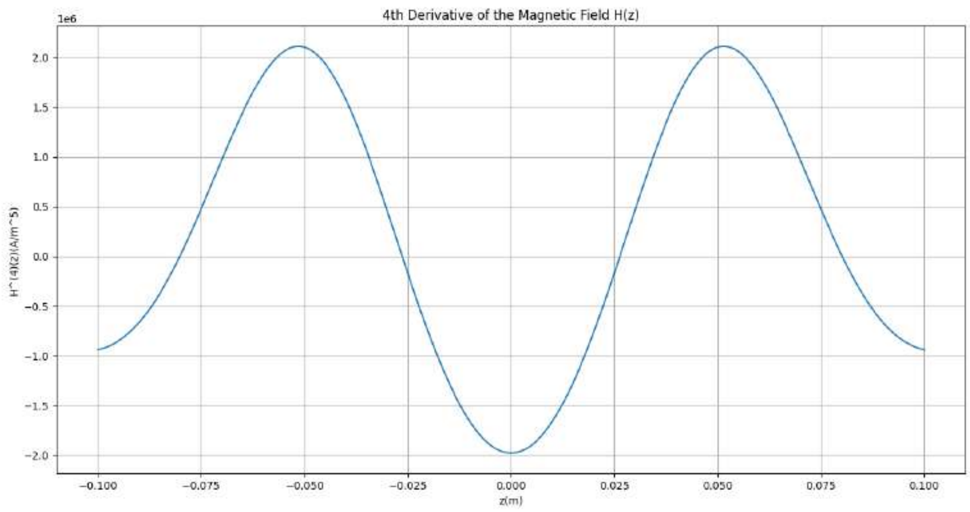
Magnetic field H(z)
H(z)(A/m)
z(m)
7.0
6.5
6.0
5.5
5.0
4.5
−0.100
−0.075
−0.050
−0.025
0.000
0.025
0.050
0.075
0.100



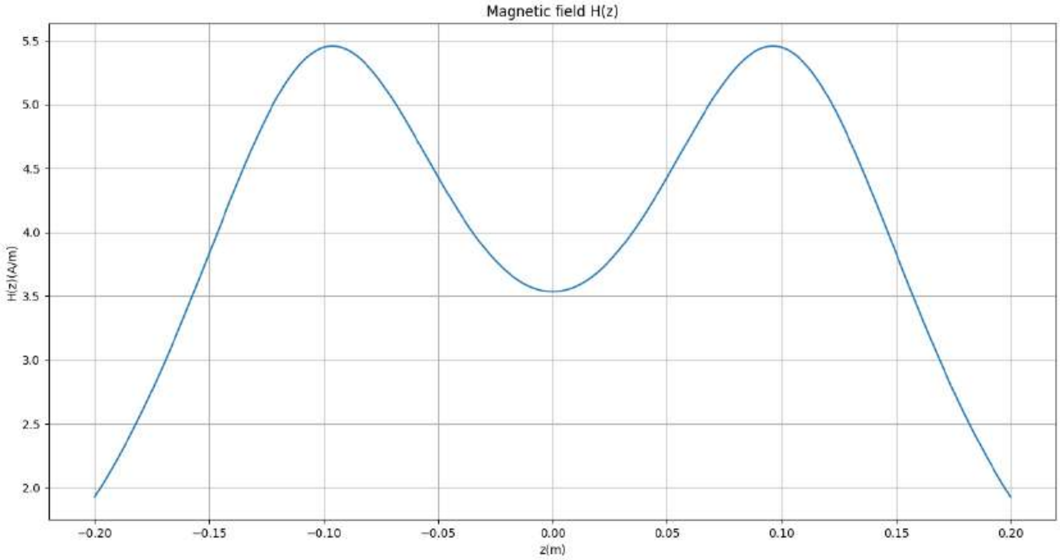
1st Derivative of the Magnetic Field H(z)
H'(z)(A/m^2)
z(m)
60
40
20
0
−20
−40
−60
−0.100
−0.075
−0.050
−0.025
0.000
0.025
0.050
0.075
0.100


2nd Derivative of the Magnetic Field H(z)
H''(z)(A/m^3)
z(m)

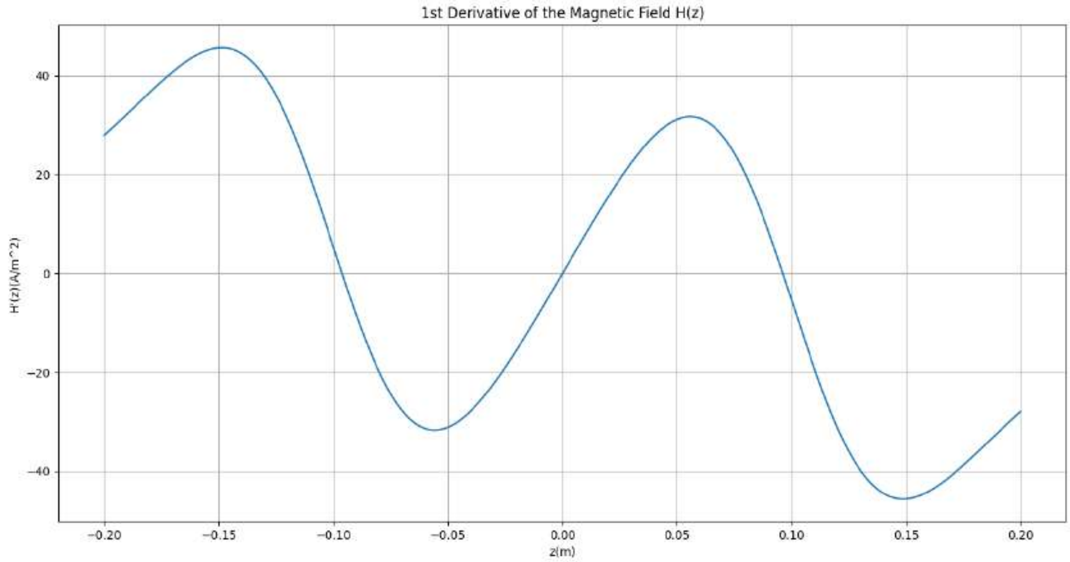

3rd Derivative of the Magnetic Field H(z)
H^(3)(z)(A/m^4)
z(m)


4th Derivative of the Magnetic Field H(z)
1e6
H^(4)(z)(A/m^5)
z(m)

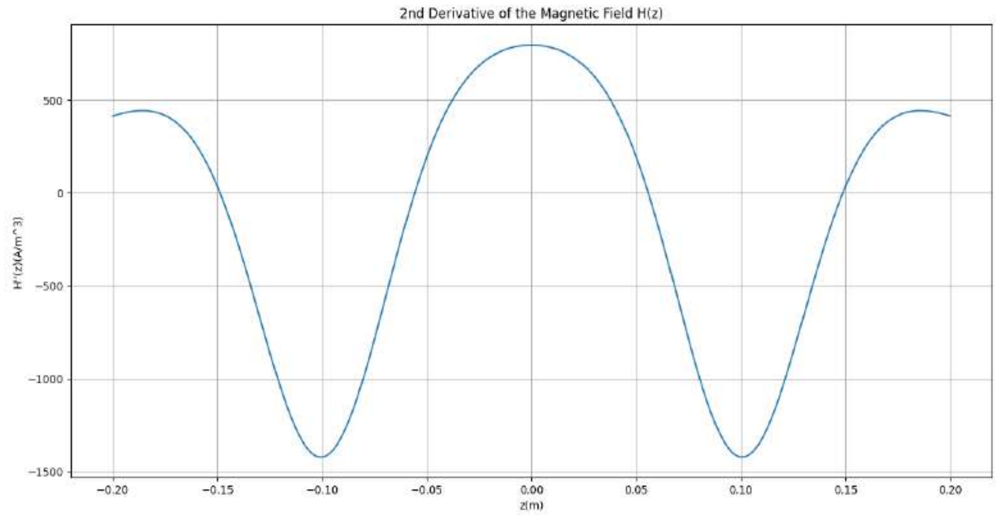
Για h = 0.1:

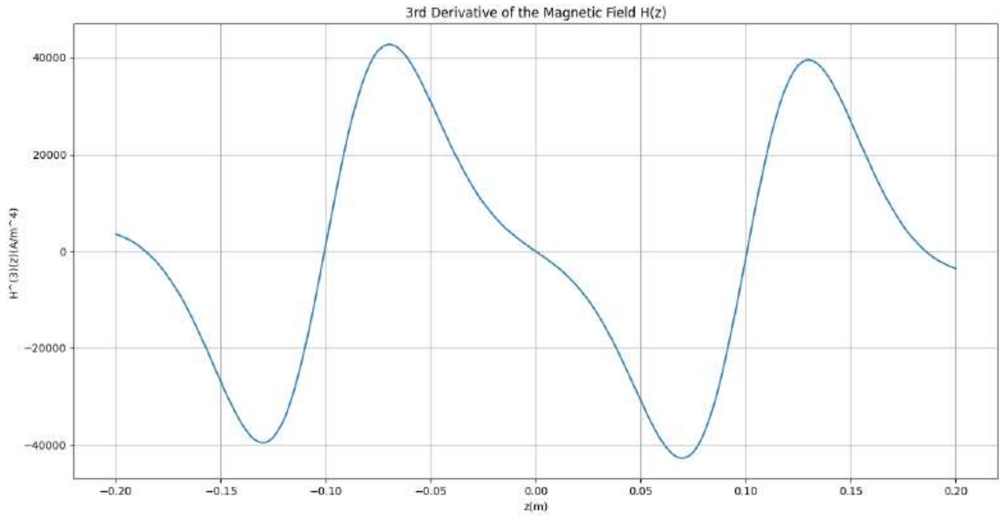


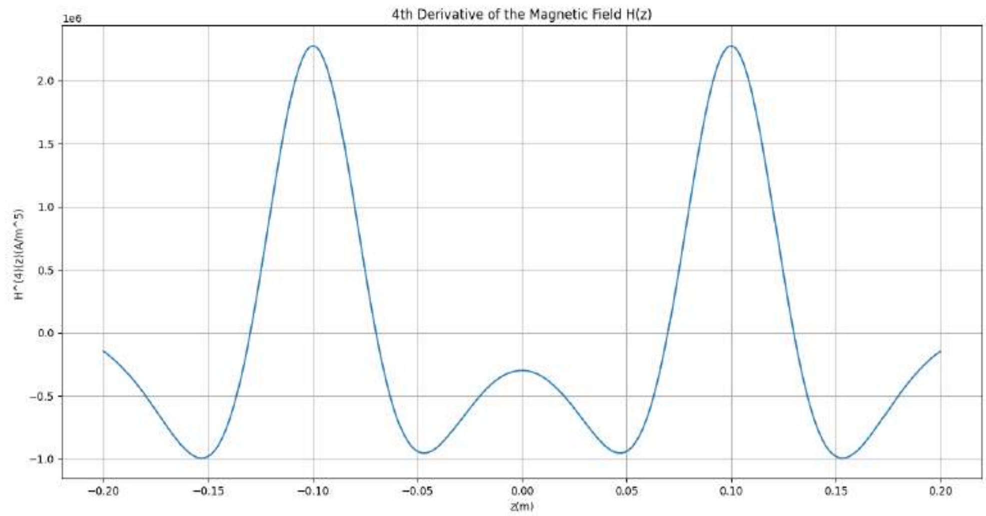
Magnetic field H(z)
H(z)(A/m)
z(m)
5.5
5.0
4.5
4.0
3.5
3.0
2.5
2.0
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20



1st Derivative of the Magnetic Field H(z)
H'(z)(A/m^2)
z(m)
40
20
0
−20
−40
−0.20
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20


2nd Derivative of the Magnetic Field H(z)
H''(z)(A/m^3)
z(m)


3rd Derivative of the Magnetic Field H(z)
H^(3)(z)(A/m^4)
z(m)


4th Derivative of the Magnetic Field H(z)
1e6
H^(4)(z)(A/m^5)
z(m)

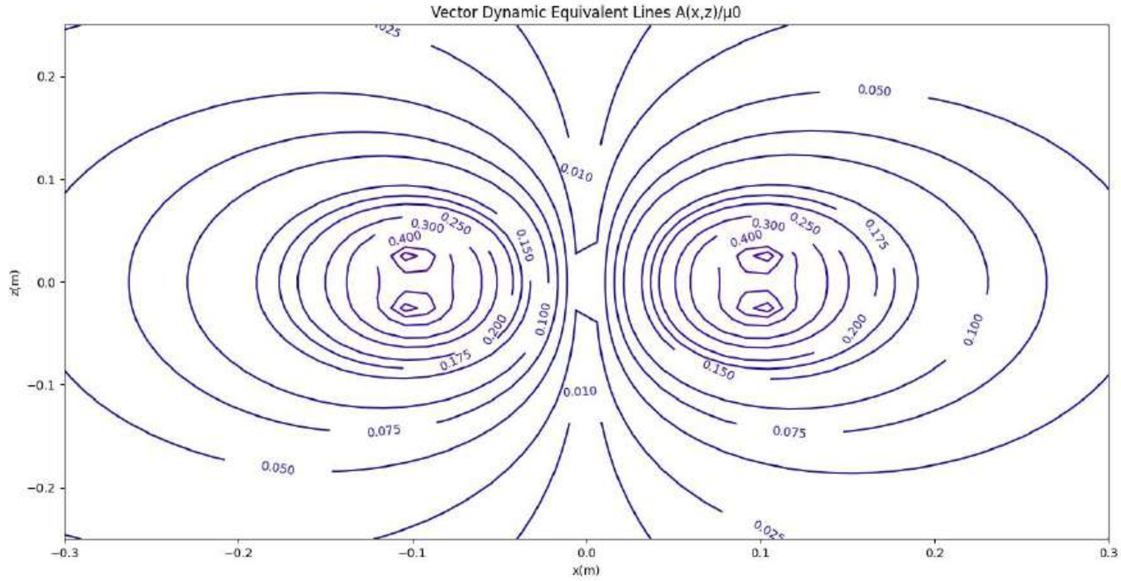
ε) Παρατίθεται ο κώδικας **Python**:

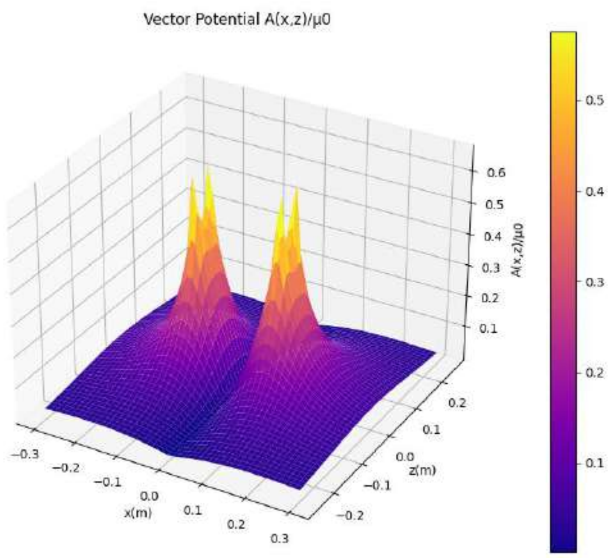
```
from mpl_toolkits.mplot3d import Axes3D
import matplotlib.pyplot as plt
import numpy as np

a = 0.1
I1 = 1
# h = 0.025
# h = 0.05
h = 0.1
monte_carlo = 1000

th = np.linspace(0, 2*np.pi, monte_carlo)

def Ax_aux(x, z, th):
    f1 = 1/(np.sqrt(x**2 + a**2 - 2*a*x*np.cos(th) + (z+h)**2))
    f2 = 1/(np.sqrt(x**2 + a**2 - 2*a*x*np.cos(th) + (z-h)**2))
    Ax_1 = (I1*a/(4*np.pi)) * (-1)*np.sin(th)*f1
    Ax_2 = (I1*a/(4*np.pi)) * (-1)*np.sin(th)*f2
    return Ax_1 + Ax_2

def Ax(x, z):
    val = np.array([Ax_aux(x, z, i) for i in th])
    return (2 * np.pi) * (val.sum() / val.size)

def Ay_aux(x, z, th):
    f1 = 1/(np.sqrt(x**2 + a**2 - 2*a*x*np.cos(th) + (z+h)**2))
    f2 = 1/(np.sqrt(x**2 + a**2 - 2*a*x*np.cos(th) + (z-h)**2))
    Ay_1 = (I1*a/(4*np.pi)) * np.cos(th)*f1
    Ay_2 = (I1*a/(4*np.pi)) * np.cos(th)*f2
    return Ay_1 + Ay_2

def Ay(x, z):
    val = np.array([Ay_aux(x, z, i) for i in th])
    return (2 * np.pi) * (val.sum() / val.size)

a_y = np.vectorize(Ay)
a_x = np.vectorize(Ax)

x = np.linspace(-0.3, 0.3, 50)
z = np.linspace(-0.25, 0.25, 50)
X, Z = np.meshgrid(x, z)
Ay = a_y(X, Z)
Ax = a_x(X, Z)

Ay = Ay * Ay
Ax = Ax * Ax
```

```python
levels = [0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.175, 0.2, 0.25, 0.3,
          0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0, 1.25, 1.5, 2.0, 3.0, 5.0]

fig, ax = plt.subplots()
ax.set(xlabel='x(m)', ylabel="z(m)",
       title='Vector Dynamic Equivalent Lines A(x,z)/μ0')
cs = ax.contour(X, Z, np.power(Ax + Ay, 0.5), levels, cmap='plasma')
ax.clabel(cs, cs.levels, inline=True, fontsize=10)
plt.show()

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.set(xlabel='x(m)', ylabel='z(m)', zlabel='A(x,z)/μ0',
       title='Vector Potential A(x,z)/μ0')
surf = ax.plot_surface(X, Z, np.power(Ax + Ay, 0.5),
                       cmap='plasma', shade=True)
fig.colorbar(surf)
plt.show()
```

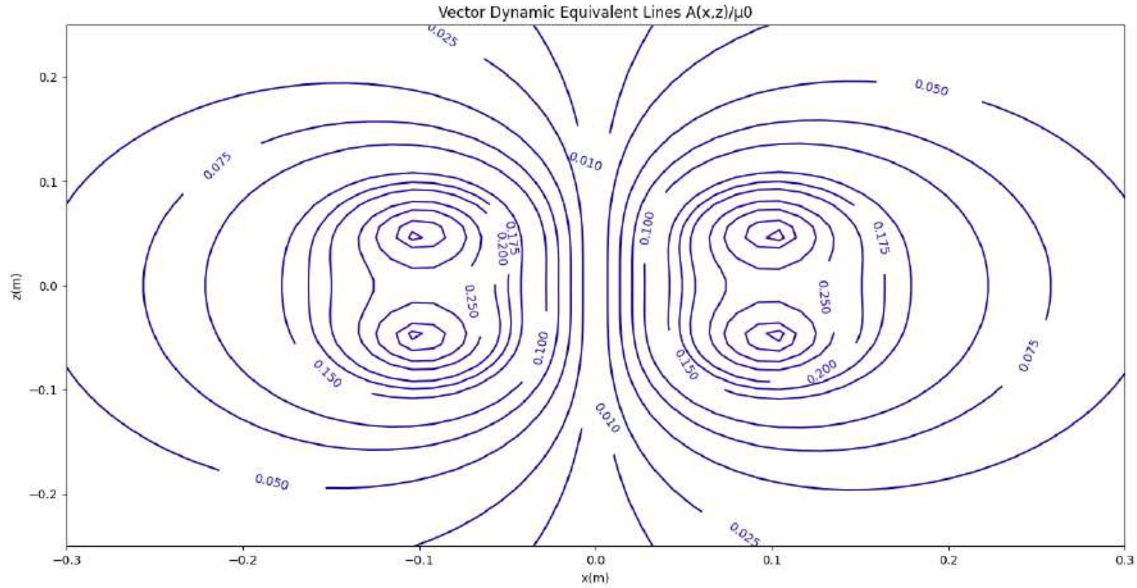
Παρουσιάζονται οι γραφικές παραστάσεις:

Για h = 0.025:

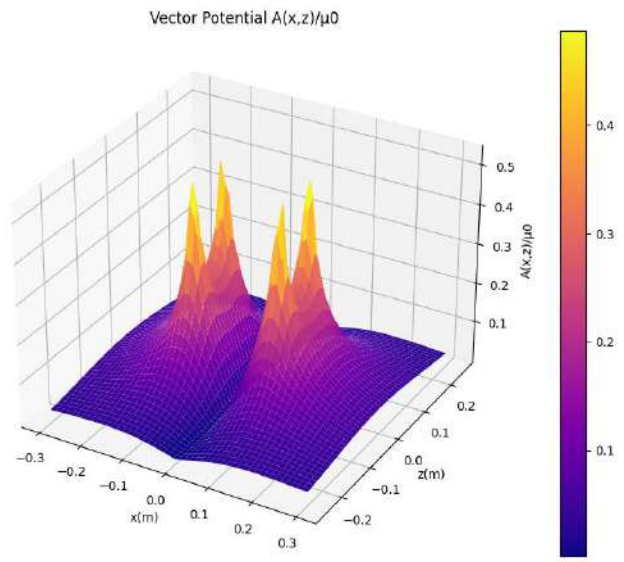




Για h = 0.05:

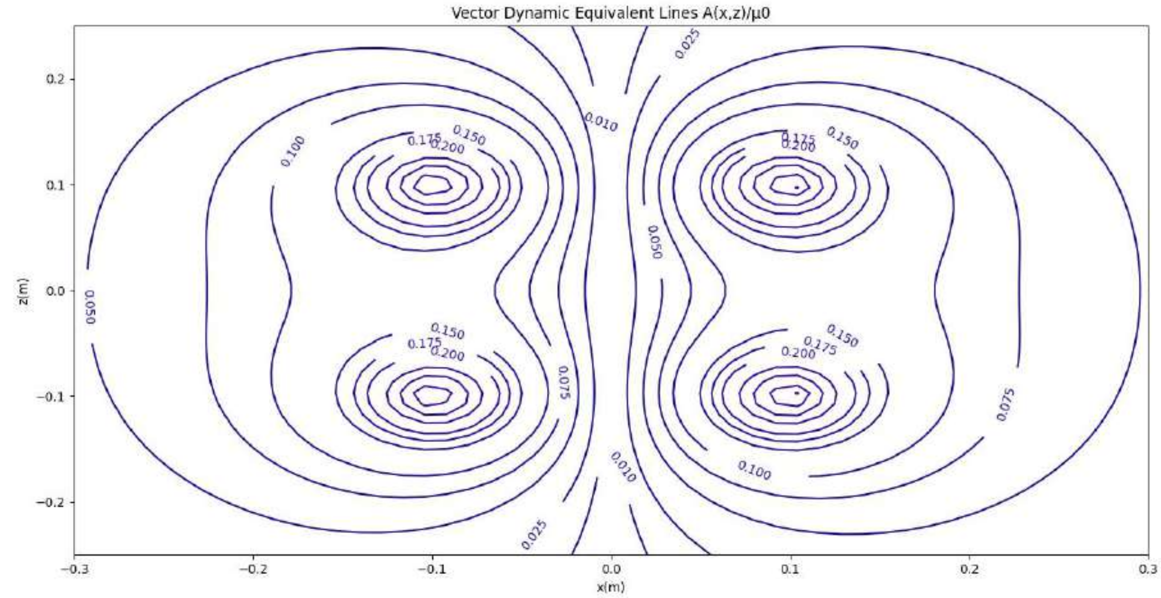


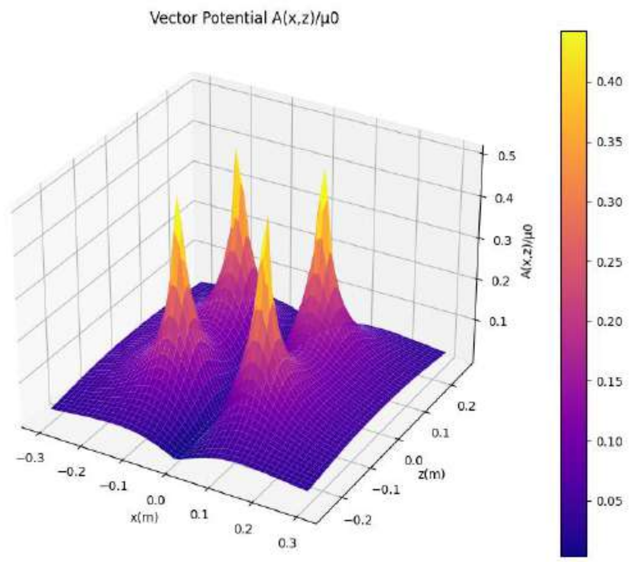
Vector Dynamic Equivalent Lines A(x,z)/μ0
z(m)
x(m)
0.025
0.050
0.075
0.010
0.100
0.150
0.175
0.200
0.250



Vector Potential A(x,z)/μ0
A(x,z)/μ0
x(m)
z(m)

Για h = 0.1:


Vector Dynamic Equivalent Lines A(x,z)/μ0
z(m)
x(m)
0.010
0.025
0.050
0.075
0.100
0.150
0.175
0.200


Vector Potential A(x,z)/μ0
A(x,z)/μ0
x(m)
z(m)

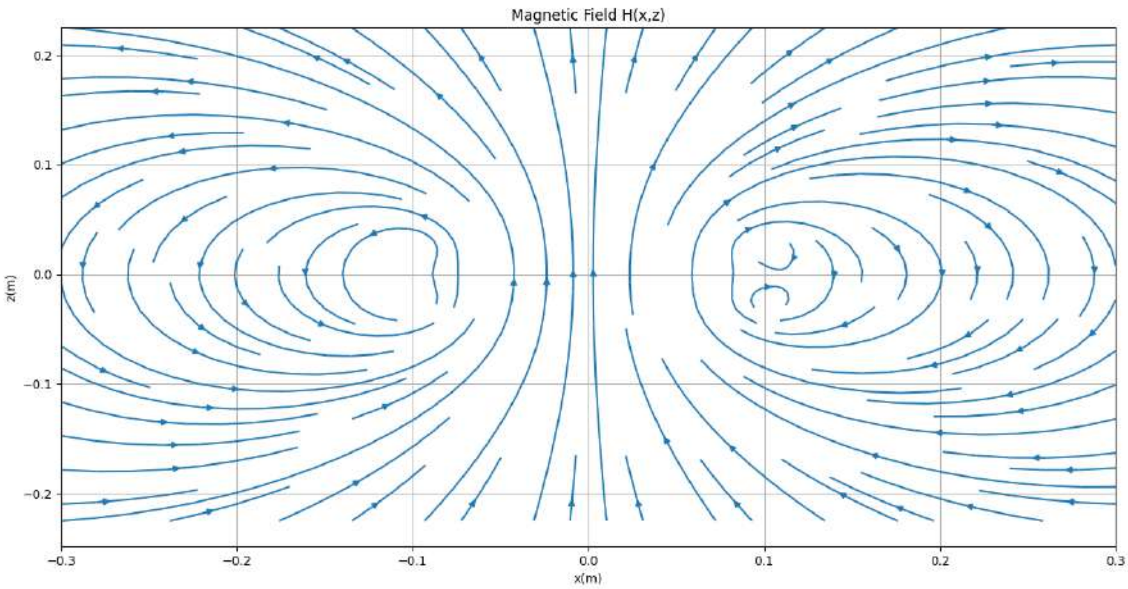
στ) Παρατίθεται ο κώδικας **Python**:

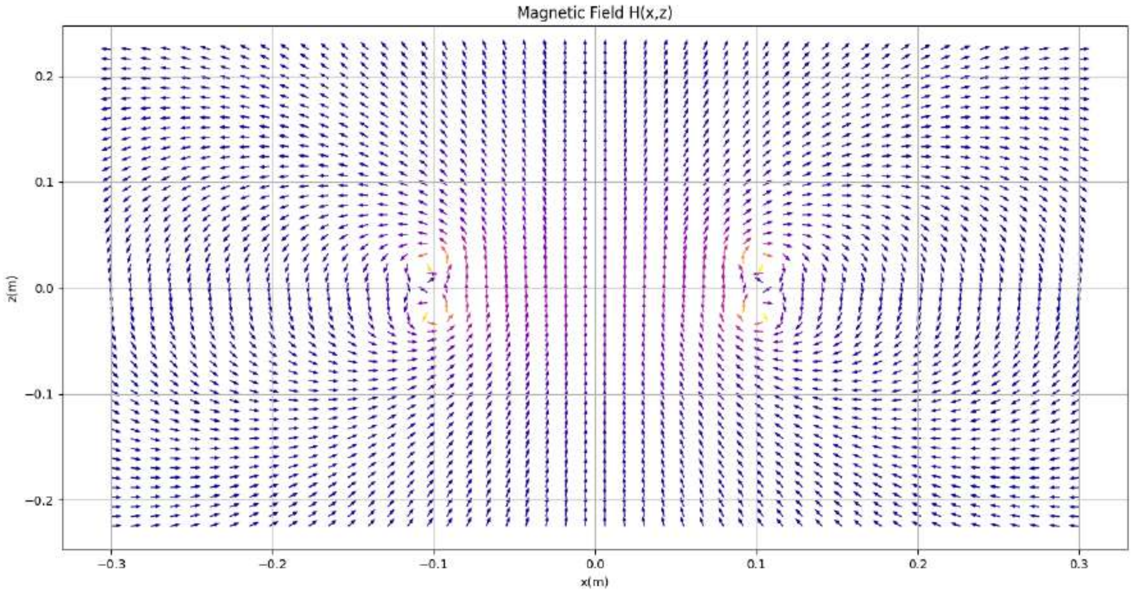
```python
import matplotlib.pyplot as plt
import numpy as np

a = 0.1
I1 = 1
# h = 0.1
# h = 0.05
h = 0.025

monte_carlo = 1000

def canvas():
    fig, ax = plt.subplots()
    ax.set(xlabel='x(m)', ylabel="z(m)", title='Magnetic Field H(x,z)')
    ax.grid()
    return ax

th = np.linspace(0, 2*np.pi, monte_carlo)

def Hx_aux(x, z, th):
    f1 = 1/(np.sqrt(x**2 + a**2 - 2*a*x*np.cos(th) + (z+h)**2))
    f2 = 1/(np.sqrt(x**2 + a**2 - 2*a*x*np.cos(th) + (z-h)**2))
    Hx_1 = (I1*a/(4*np.pi)) * (z+h)*np.cos(th)*f1**3
    Hx_2 = (I1*a/(4*np.pi)) * (z-h)*np.cos(th)*f2**3
    return Hx_1 + Hx_2

def Hx(x, z):
    val = np.array([Hx_aux(x, z, i) for i in th])
    return (2 * np.pi) * (val.sum() / val.size)

def Hz_aux(x, z, th):
    f1 = 1/(np.sqrt(x**2 + a**2 - 2*a*x*np.cos(th) + (z+h)**2))
    f2 = 1/(np.sqrt(x**2 + a**2 - 2*a*x*np.cos(th) + (z-h)**2))
    Hz_1 = (I1*a/(4*np.pi)) * (a - x*np.cos(th))*f1**3
    Hz_2 = (I1*a/(4*np.pi)) * (a - x*np.cos(th))*f2**3
    return Hz_1 + Hz_2

def Hz(x, z):
    val = np.array([Hz_aux(x, z, i) for i in th])
    return (2 * np.pi) * (val.sum() / val.size)

h_x = np.vectorize(Hx)
h_z = np.vectorize(Hz)

x = np.linspace(-0.3, 0.3, 50)
z = np.linspace(-h-0.2, h+0.2, 50)  # 3 times for h = 0.025, 0.05, 0.1
```

```
X, Z = np.meshgrid(x, z)

Hx = h_x(X, Z)
Hz = h_z(X, Z)

ax = canvas()
ax.streamplot(X, Z, Hx/(2*((Hx**2 + Hz**2)**0.5)),
              Hz/(2*((Hx**2 + Hz**2)**0.5)))
plt.show()
ax = canvas()
ax.quiver(X, Z, Hx/(2*((Hx**2 + Hz**2)**0.5)), Hz/(2*((Hx**2 + Hz**2)**0.5)),
          (Hx**2 + Hz**2)**0.5, cmap='plasma', units='xy', width=0.0009, headwidth=3
., headlength=4.)
plt.show()
```

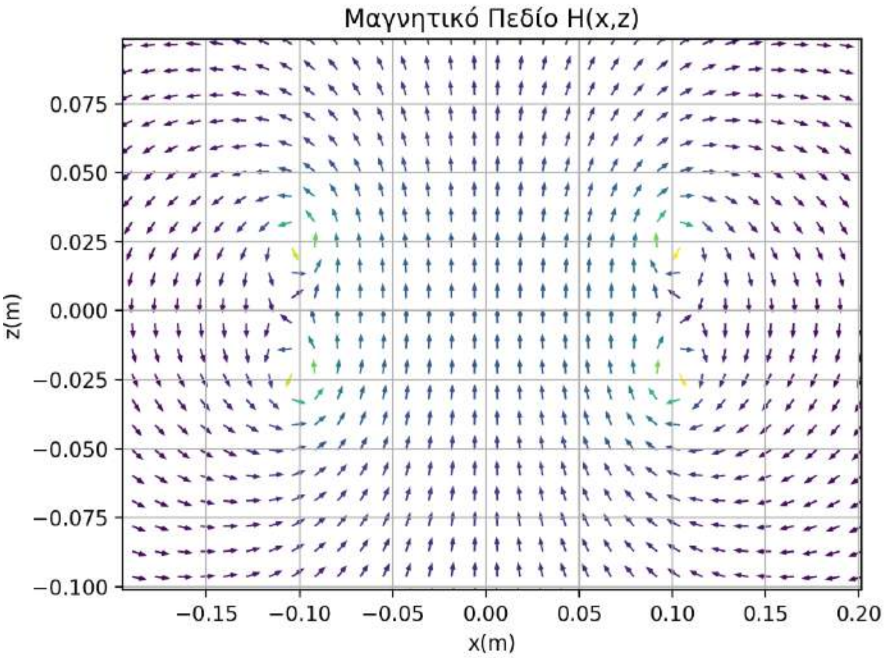
Παρουσιάζονται οι γραφικές παραστάσεις:

Για h = 0.025:



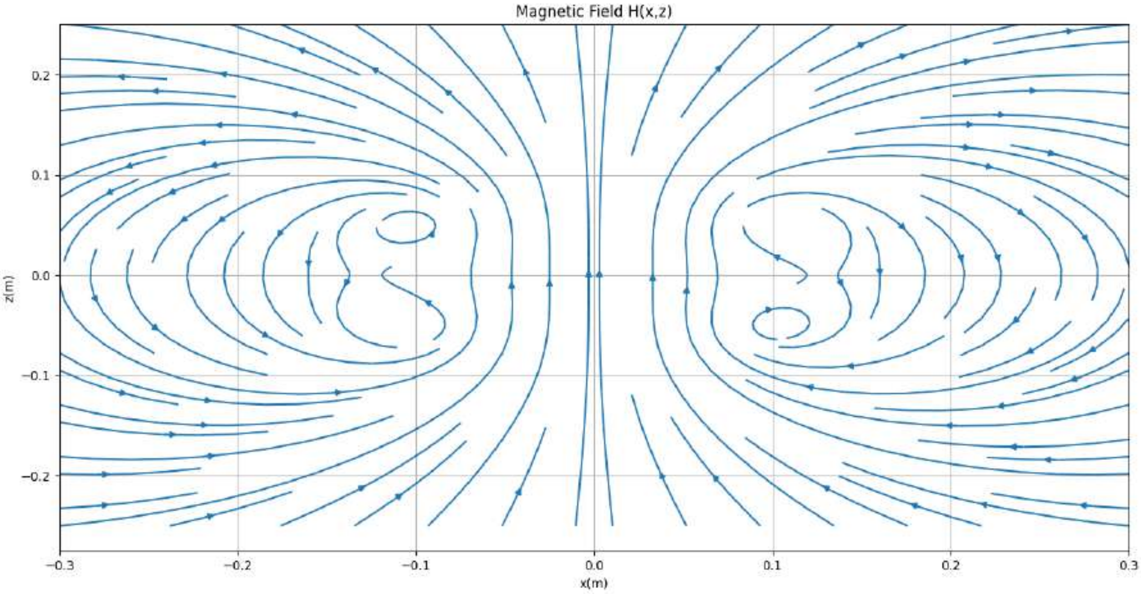


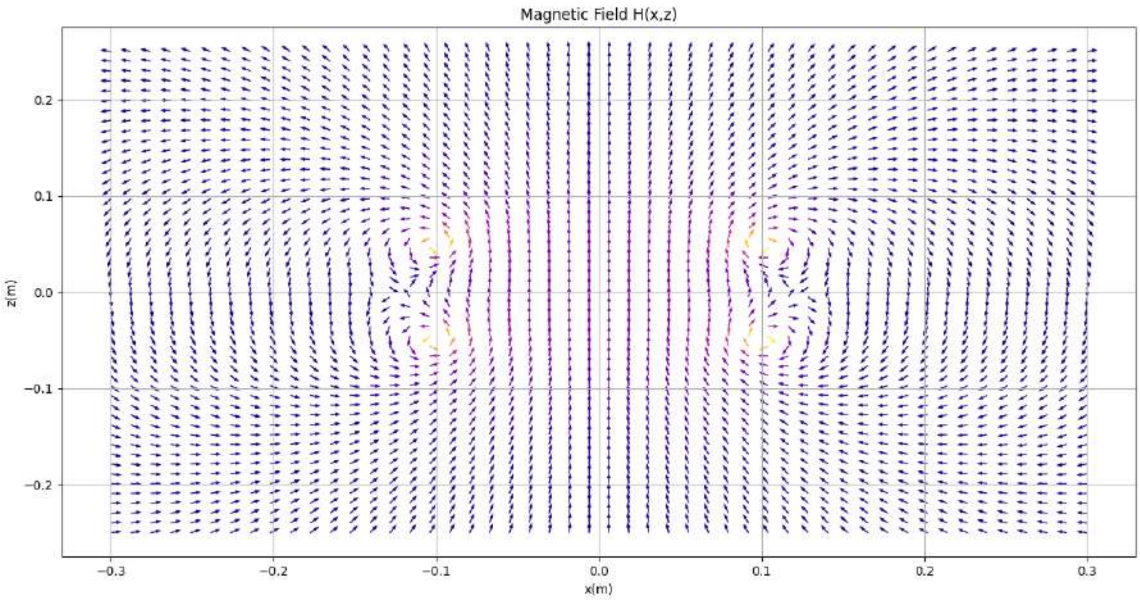
Magnetic Field H(x,z)
0.2
0.1
0.0
−0.1
−0.2
z(m)
−0.3
−0.2
−0.1
0.0
0.1
0.2
0.3
x(m)

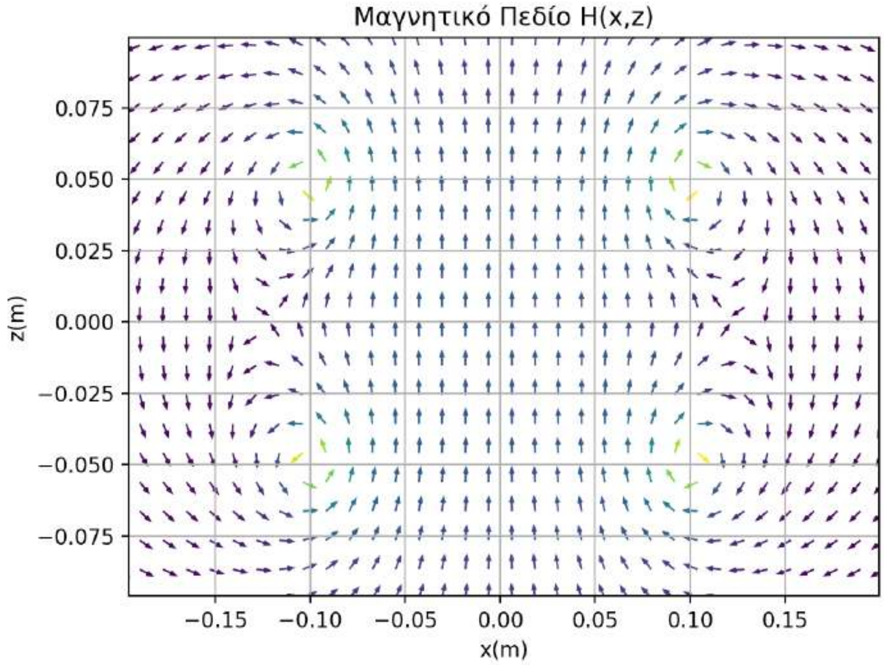


Μαγνητικό Πεδίο H(x,z)
0.075
0.050
0.025
0.000
−0.025
−0.050
−0.075
−0.100
z(m)
−0.15
−0.10
−0.05
0.00
0.05
0.10
0.15
0.20
x(m)

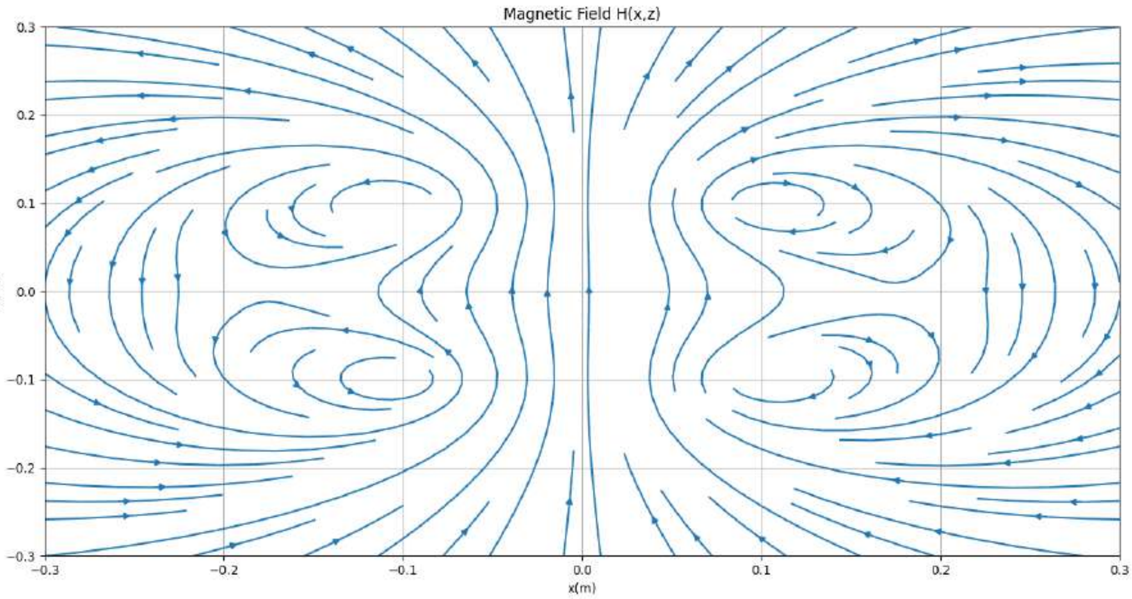
Για h = 0.05:

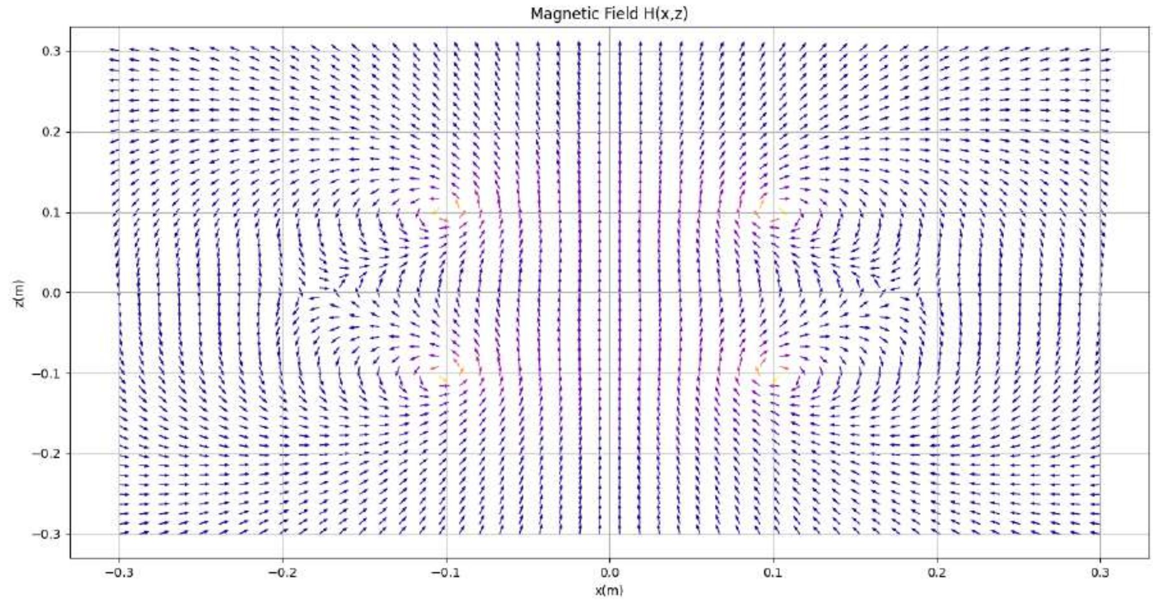


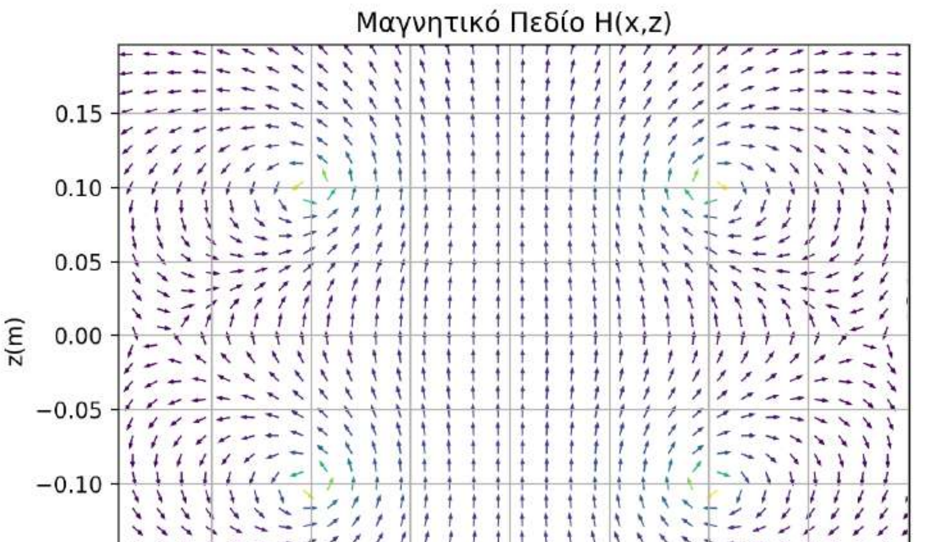




Για h = 0.1:




Magnetic Field H(x,z)
z(m)
x(m)

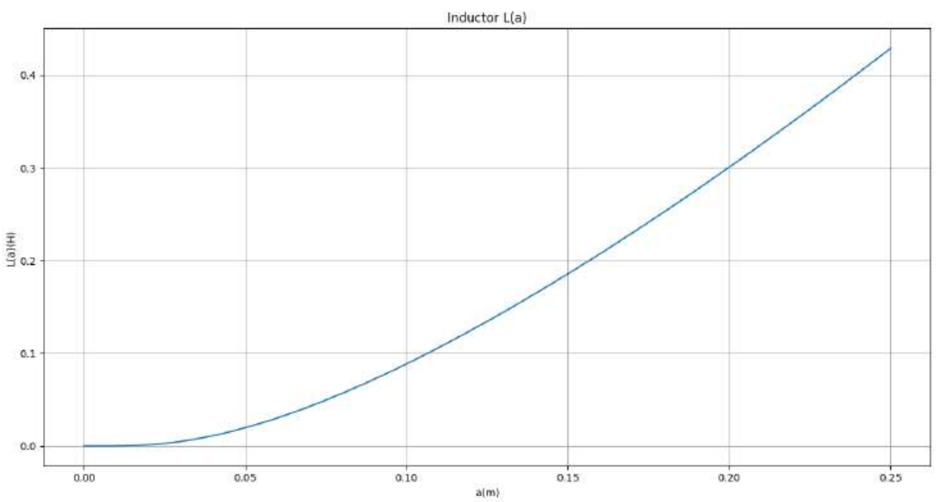

Μαγνητικό Πεδίο H(x,z)
z(m)
x(m)

ζ) Παρατίθεται ο κώδικας **Python**:

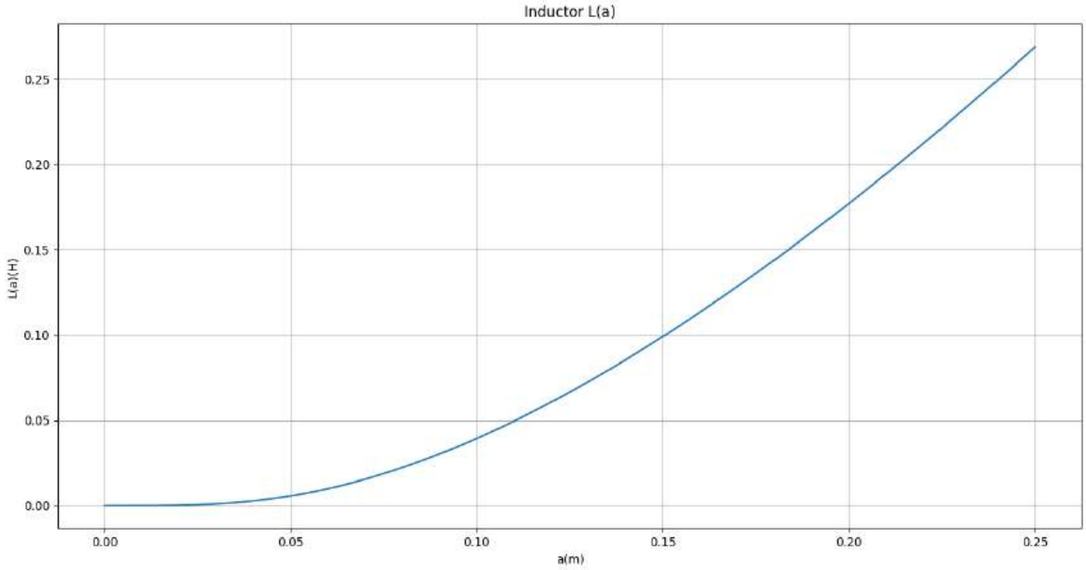
```
import matplotlib.pyplot as plt
import numpy as np

# h = 0.025
# h = 0.05
h = 0.1
monte_carlo = 100

def L_aux(a, th1, th2):
    f1 = 1/(np.sqrt(2*a**2 + 4*h**2 - 2*a**2*np.cos(th1-th2)))
    l = (a**2/(4*np.pi)) * np.cos(th1-th2)*f1
    return l

th1 = np.linspace(0, 2 * np.pi, monte_carlo)
th2 = np.linspace(0, 2 * np.pi, monte_carlo)

def L(a):
    val = np.array([[L_aux(a, i, j) for i in th1] for j in th2])
    return (2*np.pi) * (2*np.pi) * (val.sum() / val.size)
L = np.vectorize(L)
a = np.linspace(0, 0.25, 1000) # for h = 0.025, 0.05, 0.1

fig, ax = plt.subplots()
ax.set(xlabel ='a(m)', ylabel ="L(a)(H)", title ='Inductor L(a)')
ax.grid()
ax.plot(a, L(a))
plt.show()
```

Παρουσιάζονται οι γραφικές παραστάσεις:

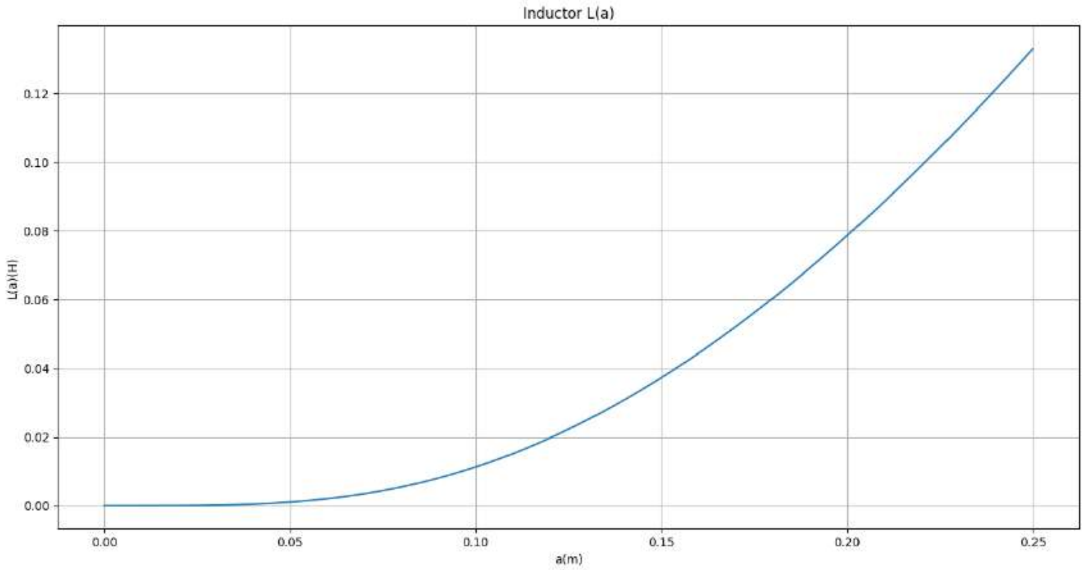
Για h = 0.025:



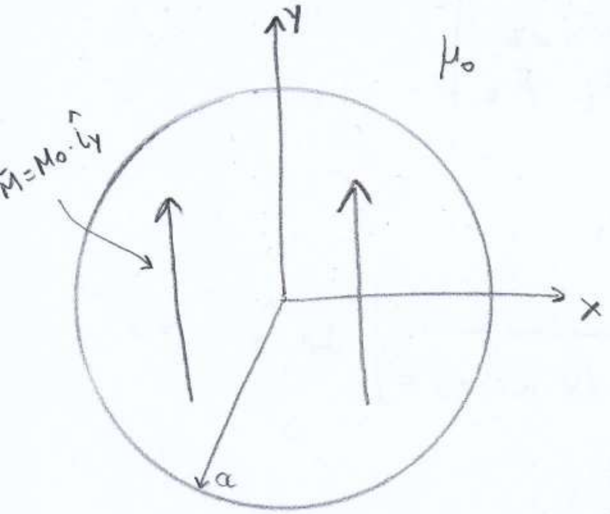
Για h = 0.05:



Για h = 0.1:



Άσκηση 10



• $\bar{J}_m = \bar{\nabla}\times\bar{M} = \begin{vmatrix} \hat{i}_x & \hat{i}_y & \hat{i}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ 0 & M_0 & 0 \end{vmatrix} = 0$

• $\bar{K}_m = -\hat{i}_n\times\bar{M} = -M(\hat{i}_n\times\hat{i}_y) \Rightarrow$

$\Rightarrow \bar{K}_m = -M_0\sin\theta\cdot\hat{i}_z$, όπου θ είναι η γωνία στην επιφάνεια:

↑ επιφανειακό ρεύμα λόγω άπειρου κυλίνδρου.

Όμως, θεωρώντας φ' τη γωνία του r_{Tc} με το x, τότε $\theta+\varphi'=\frac{\pi}{2}$ οπότε $\sin\theta=\cos\varphi'$

• Και επομένως $K_m = -M_0\cos\varphi'\cdot\hat{i}_z$

• Αν χωριστεί ο άπειρος κύλινδρος σε λωρίδες τ.ω.: $dl = a\,d\varphi$ τότε

$K_m\cdot dl = -M_0\cos\varphi'\,a\,d\varphi'$

• Επίσης, $r_T = z'\cdot\hat{i}_z + r_T\cos\varphi\cdot\hat{i}_x + r_T\cdot\sin\varphi\cdot\hat{i}_y \Rightarrow$

$\Rightarrow r_{T_0} = z'\cdot\hat{i}_z + a\cdot\cos\varphi'\cdot i_x + a\sin\varphi\cdot\hat{i}_y$

• Οπότε, $A = \frac{\mu_0\cdot I}{2\pi}\ln\left(\frac{r_{T_0}}{r_T}\right) \overset{r_{T_0}=a}{\Rightarrow} d\vec{A} = \frac{\mu_0}{2\pi}(K_m\cdot dl)\ln\left(\frac{a}{r_T}\right)\hat{i}_z \Rightarrow$

$\Rightarrow d\vec{A} = -\frac{\mu_0 M_0 a}{2\pi}\cos\varphi'\,d\varphi'\ln\left(\frac{a}{r_T}\right)\hat{i}_z \Rightarrow$

$\Rightarrow \vec{A} = \int_0^{2\pi} d\vec{A} = -\frac{\mu_0 M_0 a}{2\pi}\int_0^{2\pi}\cos\varphi'\ln\left(\frac{a}{\sqrt{r_T^2+a^2-2ar_T\cos(\varphi-\varphi')}}\right)d\varphi'\cdot\hat{i}_z$

*Σημείωση: στον κώδικα χρησιμοποιήθηκαν καρτεσιανές συντεταγμένες.

Για το μαγνητικό πεδίο : $B = \nabla \times A = \begin{vmatrix} \hat{i}_x & \hat{i}_y & \hat{i}_z \\ \frac{\partial}{\partial x} & \frac{\partial}{\partial y} & \frac{\partial}{\partial z} \\ A_x & A_y & A_z \end{vmatrix}$

Όμως $A_x = A_y = 0$, οπότε

$$B = \frac{\partial A_z}{\partial y}\hat{i}_x - \frac{\partial A_z}{\partial x}\hat{i}_y$$

όπου $A_z = -\frac{\mu_0}{2\pi} M_0 a \int_{\varphi'=0}^{2\pi} \cos\varphi' \cdot \ln\left[\frac{a}{(x-a\cos\varphi')^2 + (y-a\sin\varphi')^2}\right] d\varphi' \, \hat{i}_z \Rightarrow$

$$\Rightarrow \frac{\partial A_z}{\partial y} = -\frac{\mu_0}{2\pi} M_0 a \int_0^{2\pi} \cos\varphi' \cdot \left(\frac{-2(y-a\sin\varphi')}{2\left[(x-a\cos\varphi')^2 + (y-a\sin\varphi')^2\right]}\right) d\varphi'$$

&

$$\Rightarrow \frac{\partial A_z}{\partial x} = -\frac{\mu_0}{2\pi} M_0 a \int_0^{2\pi} \cos\varphi' \cdot \left(-\frac{2(x-a\cos\varphi')}{2\left[-(x-a\cos\varphi')^2 + (y-a\sin\varphi')^2\right.}\right) d\varphi'$$

και θεωρώντας $r_t > a$: $\overline{H} = \frac{\overline{B}}{\mu_0}$

ενώ για $r_t < a$: $\overline{H} = \frac{\overline{B}}{\mu_0} - M_0 \hat{i}_y$

$10^η$ Άσκηση

α – β) Η επίλυση φαίνεται στις χειρόγραφες σελίδες.

γ – δ) Παρατίθεται ο κώδικας **Matlab**:

Πρώτα, οι αρχικοποιήσεις:

```
clear;

a = 1;
M0 = 1;
nfig = 1;
Npoints = 250;
xmin = -3*a;
ymin = -3*a;
xmax = 3*a;
ymax = 3*a;

xx = xmin: (xmax-xmin) / Npoints:xmax;
yy = ymin: (ymax-ymin) / Npoints:ymax;
[X, Y] = meshgrid(xx, yy);
cont = [-0.9:0.1:0.9];
```

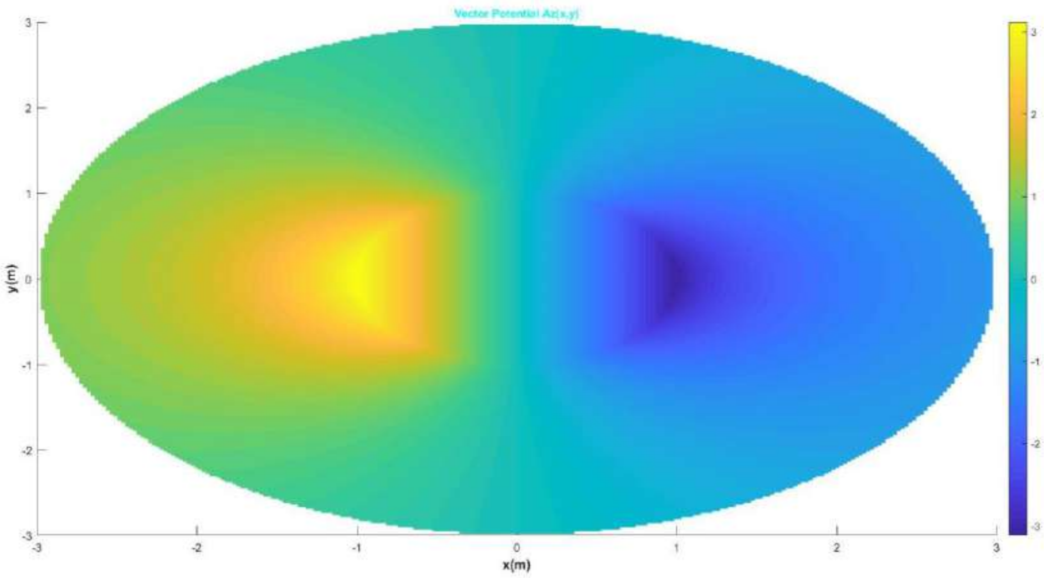
Έπειτα, το κρίσιμο τμήμα του κώδικα:

```
for ix = 1:length(xx)
    for iy = 1:length(yy)
        x0 = X(ix,iy);
        y0 = Y(ix,iy);
       rt = sqrt(x0^2+y0^2);
       if x0 > 0 & y0 >= 0
       f = atan(y0/x0);
       else if x0 <= 0
        f = pi + atan(y0/x0);
           else if x0 > 0 & y0 < 0
         f = 2*pi+atan(y0/x0);
               end
           end
       end
       if ix == (Npoints/2+1) & iy == (Npoints/2+1)
           Az(ix,iy) = 0;
       else if rt < 3*a
       A1 = @(f1) cos(f1)*log(a/sqrt([rt^2+a^2-2*rt*a*cos(f-f1)]));
       % Hx= @(f1) [(y0-a*sin(f1))*cos(f1)]/[rt^2+a^2-2*rt*a*cos(f-f1)];
       % Hy= @(f1) [(x0-a*cos(f1))*cos(f1)]/[rt^2+a^2-2*rt*a*cos(f-f1)];
       Az(ix,iy) = -a*integral(@(f1)A1(f1), 0, 2*pi, 'Arrayvalued', 1);
          else
         Az(ix,iy) = nan;
           end
       end
```

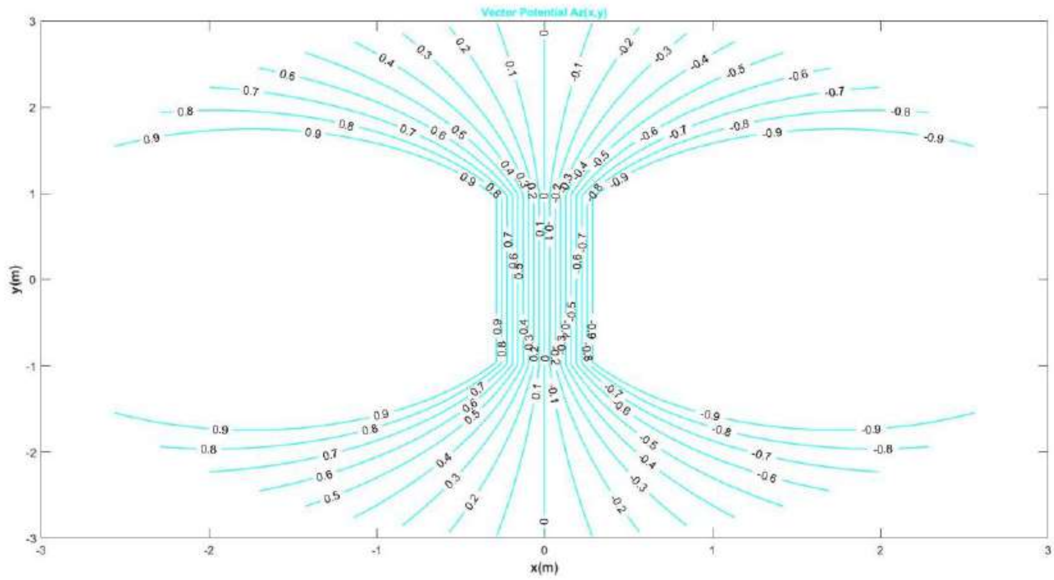
```
        if ix == (Npoints/2+1) & iy == (Npoints/2+1)
            Hxx(ix,iy) = 0;
            Hyy(ix,iy) = 0;
        else
        Hx = @(f1) [(y0-a*sin(f1))*cos(f1)]/[rt^2+a^2-2*rt*a*cos(f-f1)];
        Hy = @(f1) [(x0-a*cos(f1))*cos(f1)]/[rt^2+a^2-2*rt*a*cos(f-f1)];
        Hxx(ix,iy) = a*integral(@(f1)Hx(f1), 0, 2*pi, 'Arrayvalued', 1);
        Hyy(ix,iy) = -a*integral(@(f1)Hy(f1), 0, 2*pi, 'Arrayvalued', 1);
        if rt <= a
         Bxx(ix,iy) = Hxx(ix,iy);
         Byy(ix,iy) = Hyy(ix,iy)-M0;
        else if rt > a
         Bxx(ix,iy) = Hxx(ix,iy);
         Byy(ix,iy) = Hyy(ix,iy);
            end
        end
        end
    end
end
```

Παρουσιάζονται οι γραφικές παραστάσεις:

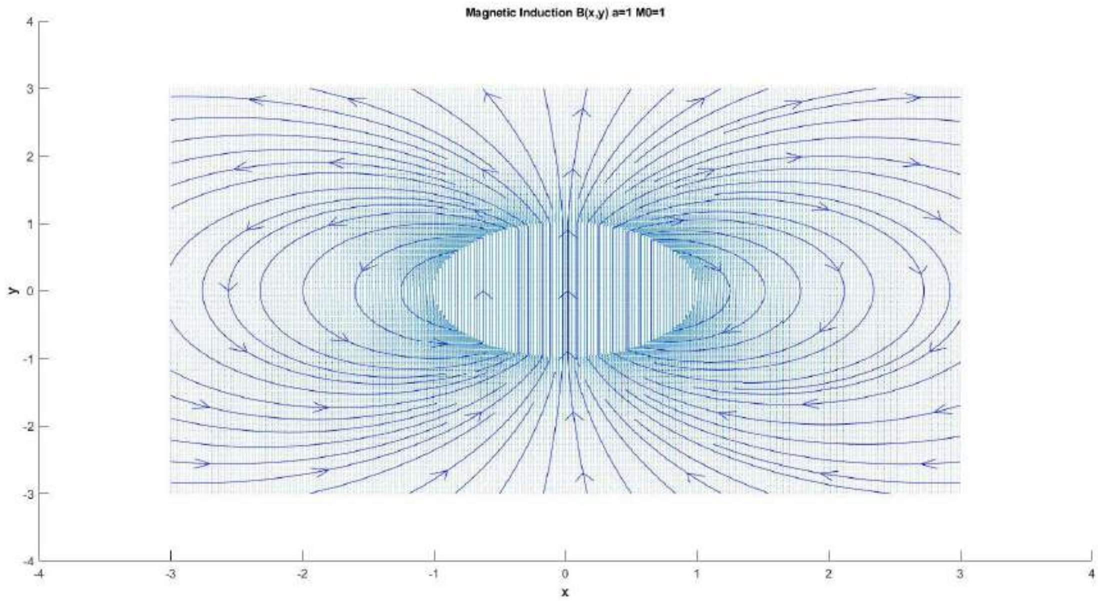
```
t = Npoints/2+1;
Az(t,t) = 0;
figure(nfig);
surface(X,Y,Az) , shading interp
hold on
colorbar
xlabel('x(m)','Fontsize',12,'FontWeight','bold')
ylabel('y(m)','Fontsize',12,'FontWeight','bold')
title(['Vector Potential Az(x,y)'],'Fontsize',10,'FontWeight','bold','Color','c')
hold off
```



```
nfig = nfig + 1;
figure(nfig);
[CS,H] = contour(X,Y,Az,cont,'Linewidth',1,'Color','c');
clabel(CS,H,cont);
xlabel('x(m)','Fontsize',12,'FontWeight','bold')
ylabel('y(m)','Fontsize',12,'FontWeight','bold')
title(['Vector Potential Az(x,y)'],'Fontsize',10,'FontWeight','bold','Color','c')
```



```
nfig = nfig + 1;
figure(nfig);
streamslice(X,Y,Bxx,Byy)
hold on
xlabel('x','Fontsize',12,'FontWeight','bold')
ylabel('y','Fontsize',12,'FontWeight','bold')
title('Magnetic Induction B(x,y) a=1 M0=1', 'Fontsize',10,'FontWeight','bold')
quiver(X,Y,Bxx,Byy);
hold off
```



```
nfig = nfig + 1;
figure(nfig);
streamslice(X,Y,Hxx,Hyy)
hold on
xlabel('x','Fontsize',12,'FontWeight','bold')
ylabel('y','Fontsize',12,'FontWeight','bold')
title('Magnetic Field H(x,y) a=1 M0=1', 'Fontsize',10,'FontWeight','bold')
quiver(X,Y,Hxx,Hyy);
hold off
```