ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ

ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ

ΗΛΕΚΤΡΟΜΑΓΝΗΤΙΚΑ ΠΕΔΙΑ Β

(2020-2021)

2η Σειρά Ασκήσεων

Ονοματεπώνυμο:

- Χρήστος Τσούφης

Αριθμός Μητρώου:

- 03117176

Στοιχεία Επικοινωνίας:

- el17176@mail.ntua.gr

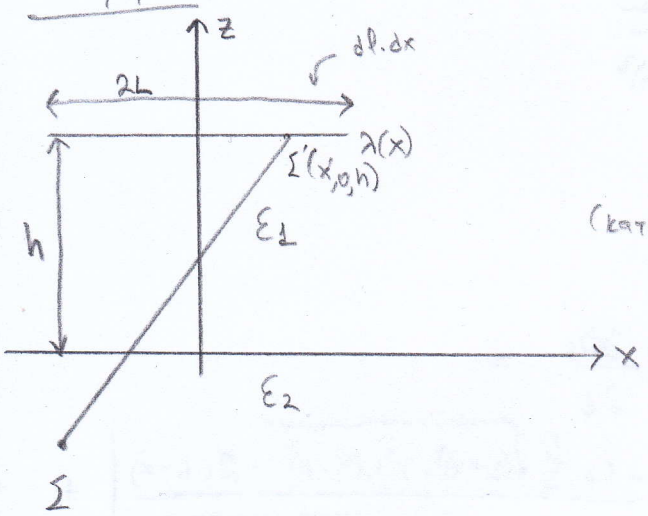
Η/Μ Πεδία Β

Σειρά Ασκήσεων 2η

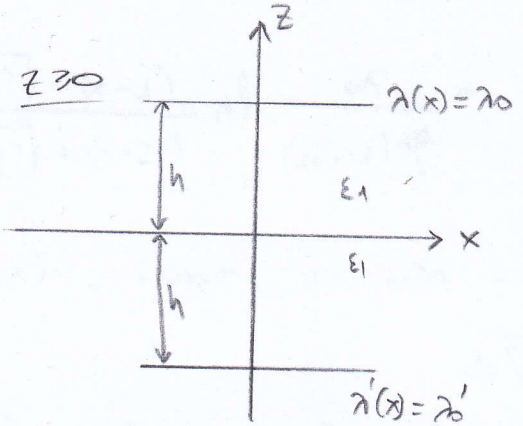
2020 - 2021

Χρήστος Τσαϊτμής
03117176

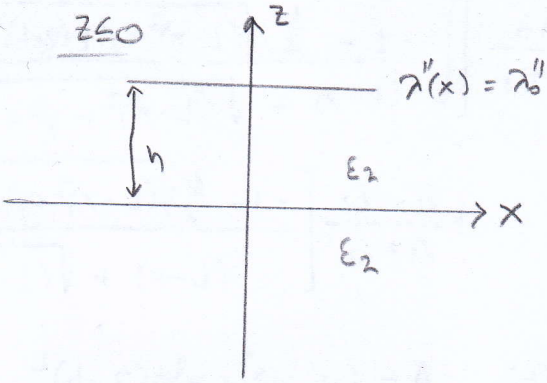
Άσκηση 7



(κατοπτρισμός) ≡



+



α) Ηλεκτροστατικό δυναμικό στο (x, y, z) για $\lambda(x) = \lambda_0$:

- $\hat{i}_n \times (\bar{E}_2 - \bar{E}_1) = 0 \xrightarrow{i_n = i_z} \begin{cases} E_{1x}\big|_{z=0} = E_{2x}\big|_{z=0} \\ E_{1y}\big|_{z=0} = E_{2y}\big|_{z=0} \end{cases}$

- $\hat{i}_n \cdot (\bar{D}_2 - \bar{D}_1) = \cancel{0} \xrightarrow{i_n = i_z} \varepsilon_1 E_{1z}\big|_{z=0} = \varepsilon_2 E_{2z}\big|_{z=0}$

- $\Phi_1(z=0, x, y) = \Phi_2(z=0, x, y)$

- $\lambda'(x) = \lambda_0' = \dfrac{\varepsilon_1 - \varepsilon_2}{\varepsilon_1 + \varepsilon_2}\lambda_0$, $\lambda''(x) = \lambda_0'' = \dfrac{2\varepsilon_2}{\varepsilon_1 + \varepsilon_2}\lambda_0$

- $\Phi_1(x, y, z) = \dfrac{\lambda_0}{4\pi\varepsilon_1}\displaystyle\int_{-L}^{L} \frac{dx'}{\sqrt{(x'-x)^2 + y^2 + (z-h)^2}} + \frac{\lambda_0}{4\pi\varepsilon_1} \cdot \frac{\varepsilon_1 - \varepsilon_2}{\varepsilon_1 + \varepsilon_2} \int_{-L}^{L} \frac{dx'}{\sqrt{(x'-x)^2 + y^2 + (z+h)^2}} =$

$$= \frac{\lambda_0}{4\pi\varepsilon_1}\left[\left[\ln\left[(x'-x) + \sqrt{(x'-x)^2 + y^2 + (z-h)^2}\right]\right]_{-L}^{L} + \frac{\varepsilon_1 - \varepsilon_2}{\varepsilon_1 + \varepsilon_2}\left[\ln\left[(x'-x) + \sqrt{(x'-x)^2 + y^2 + (z+h)^2}\right]\right]\right] =$$

$$= \frac{\lambda_0}{4\pi\varepsilon_1}\left[\ln\frac{(L-x) + \sqrt{(L-x)^2 + y^2 + (z-h)^2}}{(-L-x) + \sqrt{(L+x)^2 + y^2 + (z-h)^2}} + \frac{\varepsilon_1 - \varepsilon_2}{\varepsilon_1 + \varepsilon_2}\ln\frac{(L-x) + \sqrt{(L-x)^2 + y^2 + (z+h)^2}}{(-L-x) + \sqrt{(L+x)^2 + y^2 + (z+h)^2}}\right]$$

• $\Phi_2(x,y,z) = \frac{\lambda_0}{4\pi\varepsilon_2} \cdot \frac{2\varepsilon_2}{\varepsilon_1+\varepsilon_2} \int_{-L}^{L} \frac{dx'}{\sqrt{(x'-x)^2+y^2+(z-h)^2}} =$

για $z \le 0$

$$= \frac{\lambda_0}{2\pi(\varepsilon_1+\varepsilon_2)} \left[\ln(x'-x) + \sqrt{(x'-x)^2+y^2+(z-h)^2} \right]_{-L}^{L} =$$

$$= \frac{\lambda_0}{2\pi(\varepsilon_1+\varepsilon_2)} \ln \frac{(L-x)+\sqrt{(L-x)^2+y^2+(z-h)^2}}{(-L-x)+\sqrt{(L+x)^2+y^2+(z-h)^2}}$$

β) Ηλεκτρικό πεδίο σε τυχαίο (x,y,z):

$$\bar{E} = -\bar{\nabla}\Phi$$

• για $z \ge 0$: $\bar{E} = -\bar{\nabla}\Phi_1 = \hat{i}_x \frac{\partial\Phi_1}{\partial x} + \hat{i}_y \frac{\partial\Phi_1}{\partial y} + \hat{i}_z \frac{\partial\Phi_1}{\partial z} =$

$$= \hat{i}_x \frac{\lambda_0}{4\pi\varepsilon_1} \left[\left[\frac{-1 - \frac{1}{2}\sqrt{(L-x)^2+y^2+(z-h)^2}^{-1} \cdot 2(L-x)}{(L-x)+\sqrt{(L-x)^2+y^2+(z-h)^2}} - \frac{-1+\frac{1}{2}\sqrt{(L+x)^2+y^2+(z-h)^2}^{-1} \cdot 2(L-x)}{(-L-x)+\sqrt{(L+x)^2+y^2+(z-h)^2}} \right] + \right.$$

$$\left. + \frac{\varepsilon_1-\varepsilon_2}{\varepsilon_1+\varepsilon_2} \left[\frac{-1-\frac{1}{2}\sqrt{(L-x)^2+y^2+(z+h)^2}^{-1} \cdot 2(L-x)}{(L-x)+\sqrt{(L-x)^2+y^2+(z+h)^2}} - \frac{-1+\frac{1}{2}\sqrt{(L+x)^2+y^2+(z-h)^2}^{-1} \cdot 2(L+x)}{(-L-x)+\sqrt{(L+x)^2+y^2+(z+h)^2}} \right] \right] +$$

(θέτοντας $A = (L-x)^2+y^2+(z-h)^2$, $A' = (L+x)^2+y^2+(z+h)$)

$$+ \hat{i}_y \frac{\lambda_0}{4\pi\varepsilon_1} \left[\frac{-1+\frac{1}{2}(\sqrt{A})^{-1}2y}{(L-x)+\sqrt{A}} - \frac{-1+\frac{1}{2}(\sqrt{A'})^{-1}2y}{(-L-x)+\sqrt{A'}} + \frac{\varepsilon_1-\varepsilon_2}{\varepsilon_1+\varepsilon_2} \left[\frac{-1+\frac{1}{2}(\sqrt{A})^{-1}2y}{(L-x)+\sqrt{A}} - \frac{-1+\frac{1}{2}(\sqrt{A'})^{-1}2y}{(-L-x)+\sqrt{A'}} \right] \right] +$$

$$+ \hat{i}_z \frac{\lambda_0}{4\pi\varepsilon_1} \left[\frac{-1+\frac{1}{2}(\sqrt{A})^{-1}2(z-h)}{(L-x)+\sqrt{A}} - \frac{-1+\frac{1}{2}(\sqrt{A'})^{-1}2(z-h)}{(-L-x)+\sqrt{A'}} + \frac{\varepsilon_1-\varepsilon_2}{\varepsilon_1+\varepsilon_2} \left[\frac{-1+\frac{1}{2}(\sqrt{A})^{-1}2(z-h)}{(L-x)+\sqrt{A}} - \frac{-1+\frac{1}{2}(\sqrt{A'})^{-1}2(z-h)}{(-L-x)+\sqrt{A'}} \right] \right]$$

Για $z \geq 0$: $\bar{E} = -\bar{\nabla}\Phi_2 = \hat{i}_x \frac{\partial \Phi_2}{\partial x} + \hat{i}_y \frac{\partial \Phi_2}{\partial y} + \hat{i}_z \frac{\partial \Phi_2}{\partial z} =$

$$= \hat{i}_x \frac{\lambda_0}{4\pi(\varepsilon_1+\varepsilon_2)} \left[\frac{-1 - \frac{1}{2}(\sqrt{A})^{-1} \cdot 2(L-x)}{(L-x) + \sqrt{A}} - \frac{-1 + \frac{1}{2}(\sqrt{A'})^{-1} 2(L+x)}{(-L-x) + \sqrt{A'}} \right] +$$

$$\hat{i}_y \frac{\lambda_0}{4\pi(\varepsilon_1+\varepsilon_2)} \left[\frac{-1 + \frac{1}{2}(\sqrt{A})^{-1} 2y}{(L-x) + \sqrt{A}} - \frac{-1 + \frac{1}{2}(\sqrt{A'})^{-1} \cdot 2y}{(-L-x) + \sqrt{A'}} \right] +$$

$$\hat{i}_z \frac{\lambda_0}{4\pi(\varepsilon_1+\varepsilon_2)} \cdot \left[\frac{-1 + \frac{1}{2}(\sqrt{A})^{-1} 2(z-h)}{(L-x) + \sqrt{A}} - \frac{-1 + \frac{1}{2}(\sqrt{A'})^{-1} 2(z-h)}{(-L-x) + \sqrt{A'}} \right]$$

Είδικά για $x=0, y=0$. Δηλ. για $z \in [0,h) \cup (h, +\infty)$:

$$\Phi_1(z) = \frac{\lambda_0}{4\pi\varepsilon_1} \left[\ln \frac{L + \sqrt{L^2+(z-h)^2}}{-L + \sqrt{L^2+(z-h)^2}} + \frac{\varepsilon_1 - \varepsilon_2}{\varepsilon_1 + \varepsilon_2} \ln \frac{L + \sqrt{L^2+(z+h)^2}}{-L + \sqrt{L^2+(z+h)^2}} \right]$$

$$\bar{E}(z) = -\bar{\nabla}\Phi_1 = -\hat{i}_z \frac{\lambda_0}{4\pi\varepsilon_1} \left[\left[\frac{\overbrace{\frac{1}{2}(L^2+(z-h)^2)^{-1/2} 2(z-h)}^{B}}{L + \sqrt{L^2+(z-h)^2}} - \frac{\frac{1}{2}(L^2+(z-h)^2)^{-1/2} 2(z-h)}{-L + \sqrt{L^2+(z-h)^2}} \right] + \right.$$

$$\left. + \varepsilon' \left[\frac{\overbrace{(L^2+(z+h)^2)^{-1/2}(z+h)}^{B'}}{L + \sqrt{L^2+(z+h)^2}} - \frac{(L^2+(z+h)^2)^{-1/2}(z+h)}{-L + \sqrt{L^2+(z+h)^2}} \right] \right] =$$

$$= -\hat{i}_z \frac{\lambda_0}{4\pi\varepsilon_1} \left[\left[\frac{B(-L + \sqrt{L^2+(z-h)^2}) - B(L + \sqrt{L^2+(z-h)^2})}{L^2+(z-h)^2 - L^2} \right] + \varepsilon' \left[\frac{B'(-L + \sqrt{L^2+(z+h)^2}) - B'(L + \sqrt{L^2+(z+h)^2})}{L^2+(z+h)^2 - L^2} \right] \right] =$$

$$= -\hat{i}_z \frac{\lambda_0}{4\pi\varepsilon_1} \left[\frac{-2LB}{(z-h)^2} + \varepsilon' \frac{-2LB'}{(z+h)^2} \right] = -\hat{i}_z \frac{\lambda_0}{4\pi\varepsilon_1} \left[\frac{-2L(z-h)}{(z-h)^2\sqrt{L^2+(z-h)^2}} + \varepsilon' \frac{-2L(z+h)}{(z+h)^2\sqrt{L^2+(z+h)^2}} \right] =$$

$$= \hat{i}_z \frac{\lambda_0}{4\pi\varepsilon_1} \left[\frac{2L}{(z-h)\sqrt{L^2+(z-h)^2}} + \frac{\varepsilon_1 - \varepsilon_2}{\varepsilon_1 + \varepsilon_2} \frac{2L}{(z+h)\sqrt{L^2+(z+h)^2}} \right], \quad z \geq 0$$

• για $z \le 0$: $\vec{E}(z) = -\vec{\nabla}\Phi_2$

οπου $\Phi_2(z) = \frac{\lambda_0}{4\pi(\varepsilon_1+\varepsilon_2)} \ln \frac{L+\sqrt{L^2+(z-h)^2}}{-L+\sqrt{L^2+(z-h)^2}}$

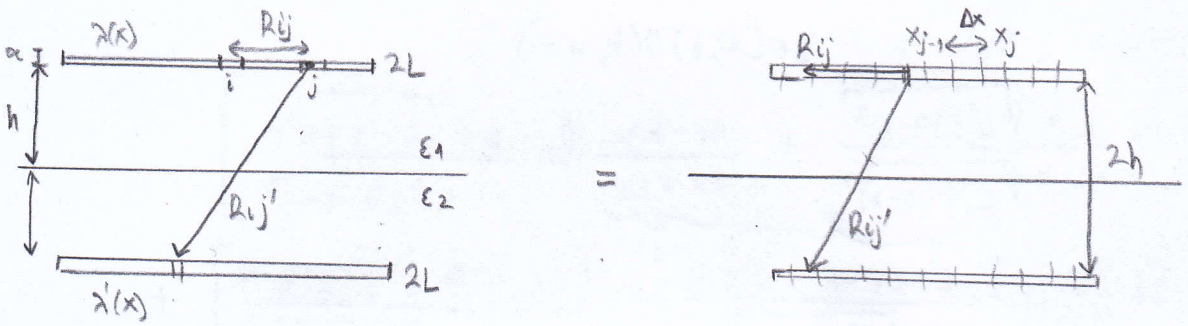
$$\vec{E}(z) = -\vec{\nabla}\Phi_2 = -\hat{i}_z \frac{\lambda_0}{4\pi(\varepsilon_1+\varepsilon_2)} \left[\frac{\frac{1}{2}(L^2+(z-h)^2)^{-1/2} \cdot 2(z-h)}{L+\sqrt{L^2+(z-h)^2}} - \frac{\frac{1}{2}(L^2+(z-h)^2)^{-1/2} 2(z-h)}{-L+\sqrt{L^2+(z-h)^2}} \right] =$$

(B' = denominator of the first term, B'' = denominator of the second term)

$$= -\hat{i}_z \frac{\lambda_0}{4\pi(\varepsilon_1+\varepsilon_2)} \cdot \left[\frac{B'(-L+\sqrt{L^2+(z-h)^2}) - B''(L+\sqrt{L^2+(z-h)^2})}{L^2+(z-h)^2-L^2} \right] = -\hat{i}_z \frac{\lambda_0}{4\pi(\varepsilon_1+\varepsilon_2)} \cdot \frac{-2LB''}{(z-h)^2} =$$

$$= \hat{i}_z \frac{\lambda_0}{4\pi(\varepsilon_1+\varepsilon_2)} \cdot \frac{2L(z-h)}{(z-h)^2\sqrt{L^2+(z-h)^2}} = \hat{i}_z \frac{\lambda_0}{2\pi(\varepsilon_1+\varepsilon_2)} \cdot \frac{L}{(z-h)\sqrt{L^2+(z-h)^2}}$$

γ)



συνολικό φορτίο αγωγού : Q

- Χρησιμοποιώντας την μέθοδο των εικόνων, διαιρείται η ράβδος σε N στοιχειώδη τμήματα.
- Για κάθε τμήμα ισχύει: $\Phi_i = \sum_{j=1}^{N} A_{ij}\lambda_j$ όπου $\begin{cases} A_{ij} = \frac{\Delta x}{4\pi\varepsilon_1}\left[\frac{1}{R_{ij}} + \frac{\varepsilon_1-\varepsilon_2}{\varepsilon_1+\varepsilon_2}\frac{1}{R_{ij'}}\right], i \neq j \\ A_{ii} = \frac{1}{4\pi\varepsilon}\left[2\ln\left(\frac{\Delta x}{a}\right) + \frac{\Delta x(\varepsilon_1-\varepsilon_2)}{\varepsilon_1+\varepsilon_2}\cdot\frac{1}{R_{ii'}}\right] \end{cases}$

οπου $R_{ij} = |\hat{x}_i - \hat{x}_j|$

$R_{ij'} = \sqrt{(x_i-x_j)^2+(2h)^2}$

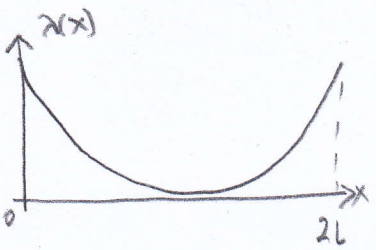
$R_{ii'} = 2h$

• Για N τμήματα (αλλιώς η ράβδος), ισχύει $[\Phi] = [A][\lambda] \Rightarrow [\lambda] = [A]^{-1}[\Phi] \Rightarrow$
$\Rightarrow [\lambda] = [A]^{-1}[\Pi]\Phi_i \Rightarrow [\lambda] = [A]^{-1}[\Pi]\Phi'$, αφού η ράβδος είναι ισοδ. επιφ.: ισχύει $\Phi_1 = \Phi_2 = \ldots = \Phi_N = \Phi'$

• $\Phi(x,y,z>0) = \frac{\Delta x}{4\pi\varepsilon_1} \sum_{j=1}^{N} \lambda_j \left[\frac{1}{R_j} + \frac{\varepsilon_1-\varepsilon_2}{\varepsilon_1+\varepsilon_2}\frac{1}{R_{j'}}\right]$

• Η $\lambda(x)$ θα έχει την μορφή:

Άρα, το ηλ. φορτίο συγκεντρώνεται κυρίως στα άκρα.



7η Άσκηση

α – γ) Η επίλυση φαίνεται στις χειρόγραφες σελίδες.

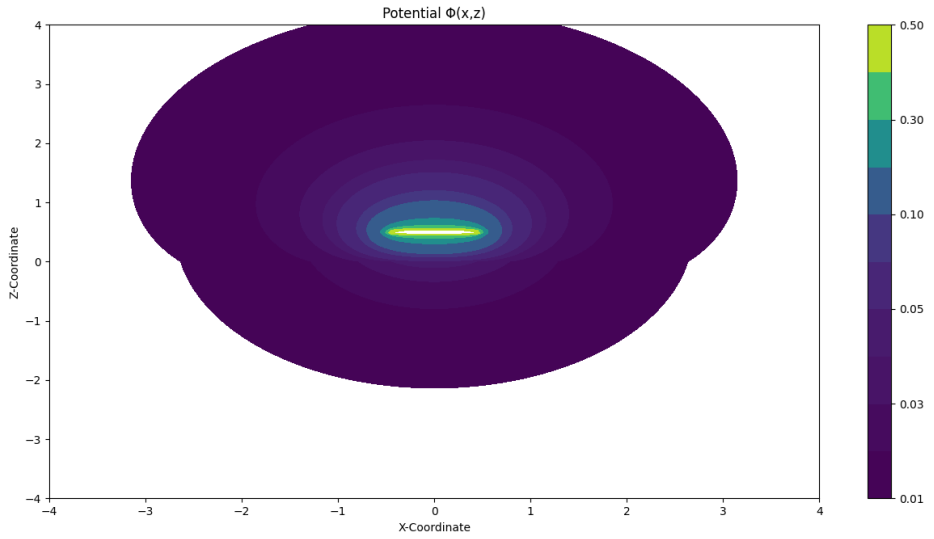
δ) 1ο Σκέλος: Ηλεκτροστατικό Δυναμικό

Παρουσιάζεται αποσπασματικά ο κώδικας **Python**:

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

L = 0.5
h = 0.5

def F1(x, z):
    global L
    global h
    f1 = (np.log((x-L - np.sqrt((x-L)**2 + (z-h)**2))/(x+L - np.sqrt((x+L)**2 + (z-
h)**2))) -
          (2/3)*np.log((x-L - np.sqrt((x-
L)**2 + (z+h)**2))/(x+L - np.sqrt((x+L)**2 + (z+h)**2))))/(4*np.pi)
    return f1

def F2(x, z):
    global L
    global h
    f2 = (np.log((x-L - np.sqrt((x-L)**2 + (z-h)**2))/(x+L - np.sqrt((x+L)**2 + (z-
h)**2))))/(12*np.pi)
    return f2

f1 = np.vectorize(F1)
f2 = np.vectorize(F2)
x = np.linspace(-4, 4, 100)
z_pos = np.linspace(0, 4, 100)
z_neg = np.linspace(-4, 0, 100)
X1, Z1 = np.meshgrid(x, z_pos)
X2, Z2 = np.meshgrid(x, z_neg)
Y1 = f1(X1, Z1)
Y2 = f2(X2, Z2)
fig, ax = plt.subplots(figsize=(10, 10))
cs1 = ax.contourf(X1, Z1, Y1, zdir='xz', offset=11,
                  levels=[0.01, 0.02, 0.03, 0.04, 0.05, 0.075, 0.1, 0.2, 0.3, 0.4, 0
.5],
                  cmap='viridis')
cs2 = ax.contourf(X2, Z2, Y2, zdir='xz', offset=11,
                  levels=[0.01, 0.02, 0.03, 0.04, 0.05, 0.075, 0.1, 0.2, 0.3, 0.4, 0
.5],
                  cmap='viridis')
```

```
cbar = fig.colorbar(cs1)
ax.set_title('Potential Φ(x,z)')
ax.set_xlabel('X-Coordinate')
ax.set_ylabel('Z-Coordinate')
plt.show()
```

Παρουσιάζεται η γραφική παράσταση:



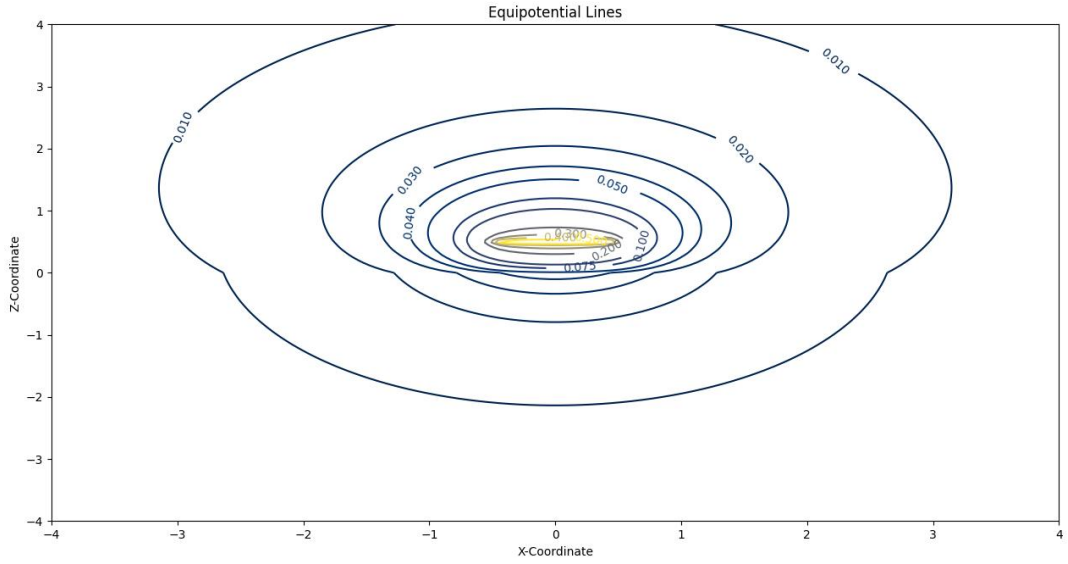
2ο Σκέλος: Ισοδυναμικές Επιφάνειες (Γραμμές)

Παρουσιάζεται αποσπασματικά ο κώδικας **Python**:

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

L = 0.5
h = 0.5

def F1(x, z):
    global L
    global h
    f1 = (np.log((x-L - np.sqrt((x-L)**2 + (z-h)**2))/(x+L - np.sqrt((x+L)**2 + (z-
h)**2))) -
        (2/3)*np.log((x-L - np.sqrt((x-
L)**2 + (z+h)**2))/(x+L - np.sqrt((x+L)**2 + (z+h)**2))))/(4*np.pi)
    return f1

def F2(x, z):
    global L
    global h
```

```
    f2 = (np.log((x-L - np.sqrt((x-L)**2 + (z-h)**2))/(x+L - np.sqrt((x+L)**2 + (z-
h)**2))))/(12*np.pi)
    return f2

f1 = np.vectorize(F1)
f2 = np.vectorize(F2)
x = np.linspace(-4, 4, 100)
z_pos = np.linspace(0, 4, 100)
z_neg = np.linspace(-4, 0, 100)
X1, Z1 = np.meshgrid(x, z_pos)
X2, Z2 = np.meshgrid(x, z_neg)
Y1 = f1(X1, Z1)
Y2 = f2(X2, Z2)

fig, ax = plt.subplots(figsize=(10, 10))
cs1 = ax.contour(X1, Z1, Y1, zdir='xz', offset=11,
                 levels=[0.01, 0.02, 0.03, 0.04, 0.05, 0.075, 0.1, 0.2, 0.3, 0.4, 0.
5],
                 cmap='cividis')
cs2 = ax.contour(X2, Z2, Y2, zdir='xz', offset=11,
                 levels=[0.01, 0.02, 0.03, 0.04, 0.05, 0.075, 0.1, 0.2, 0.3, 0.4, 0.
5],
                 cmap='cividis')

ax.clabel(cs1, cs1.levels, inline=True, fontsize=10)
ax.set_title('Equipotential Lines')
ax.set_xlabel('X-Coordinate')
ax.set_ylabel('Z-Coordinate')
plt.show()
```

Παρουσιάζεται η γραφική παράσταση:



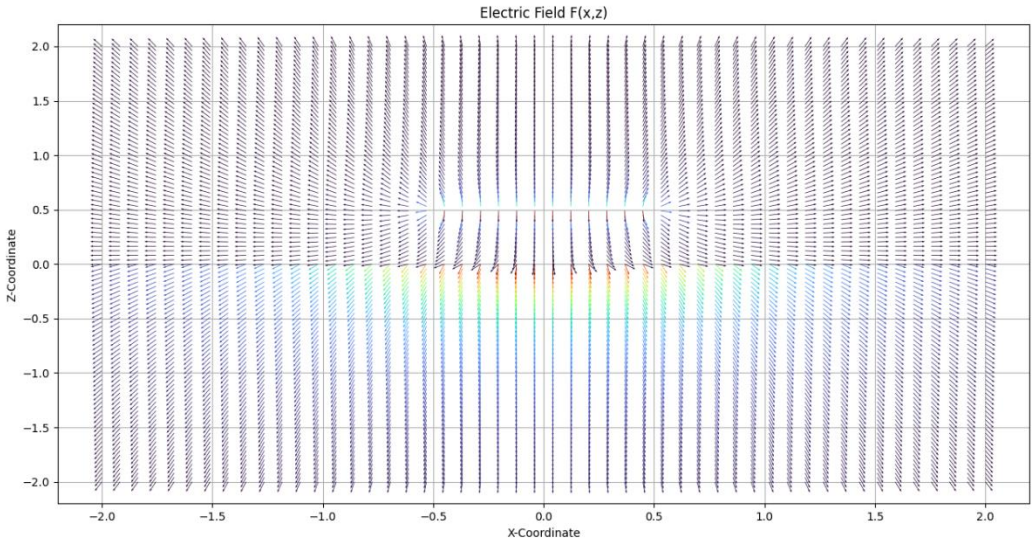
3ο Σκέλος: Ηλεκτρικό Πεδίο

Παρουσιάζεται αποσπασματικά ο κώδικας **Python**:

```
import scipy as sp
import numpy as np
import matplotlib
import matplotlib.pyplot as plt
from scipy import integrate
from mpl_toolkits import mplot3d

L = 0.5
h = 0.5

def E1(x, z):
    global L, h
    Ex1 = ((1/(np.sqrt((x-L)**2 + (z-h)**2)) - 1/(np.sqrt((x+L)**2 + (z-h)**2))) +
        (2/3)*(1/(np.sqrt((x-
L)**2 + (z+h)**2)) - 1/(np.sqrt((x+L)**2 + (z+h)**2))))/(4*np.pi)
    Ez1 = ((1/(z-h))*((x+L)/(np.sqrt((x+L)**2 + (z-h)**2)) - (x-L)/(np.sqrt((x-
L)**2 + (z-h)**2))) +
        (2/3)*((1/(z+h))*((x+L)/(np.sqrt((x+L)**2 + (z+h)**2)) - (x-L)/(np.sqrt((x-
L)**2 + (z+h)**2)))))/(4*np.pi)
    return Ex1, Ez1

def E2(x, z):
    global L, h
    Ex2 = ((1/(np.sqrt((x-L)**2 + (z-h)**2)) - 1/(np.sqrt((x+L)**2 + (z-
h)**2))))/(12*np.pi)
    Ez2 = ((1/(z-h))*((x+L)/(np.sqrt((x+L)**2 + (z-h)**2)) - (x-L)/(np.sqrt((x-
L)**2 + (z-h)**2))))/(12*np.pi)
    return Ex2, Ez2

e1 = np.vectorize(E1)
e2 = np.vectorize(E2)

x = np.linspace(-2, 2, 50)
z_pos = np.linspace(0, 2, 50)
z_neg = np.linspace(-2, 0, 50)
X1, Z1 = np.meshgrid(x, z_pos)
X2, Z2 = np.meshgrid(x, z_neg)
Ex1, Ez1 = e1(X1, Z1)
Ex2, Ez2 = e2(X2, Z2)

fig, ax = plt.subplots(figsize=(10, 10))

ax.quiver(X1, Z1, Ex1/((Ex1**2+Ez1**2)**0.5), Ez1/((Ex1**2+Ez1**2)**0.5), (Ex1**2+Ez
1**2) **
```

```
          0.5, cmap='turbo', units='xy', scale=15, zorder=3, width=0.0035, headwidth
=3., headlength=4.)
ax.quiver(X2, Z2, Ex2/((Ex2**2+Ez2**2)**0.5), Ez2/((Ex2**2+Ez2**2)**0.5), (Ex2**2+Ez
2**2) **
          0.5, cmap='turbo', units='xy', scale=15, zorder=3, width=0.0035, headwidth
=3., headlength=4.)

ax.set_title('Electric Field F(x,z)')
ax.set_xlabel('X-Coordinate')
ax.set_ylabel('Z-Coordinate')
ax.grid()
plt.show()
```

Παρουσιάζεται η γραφική παράσταση:



4ο Σκέλος: Γραμμική Κατανομή Φορτίου

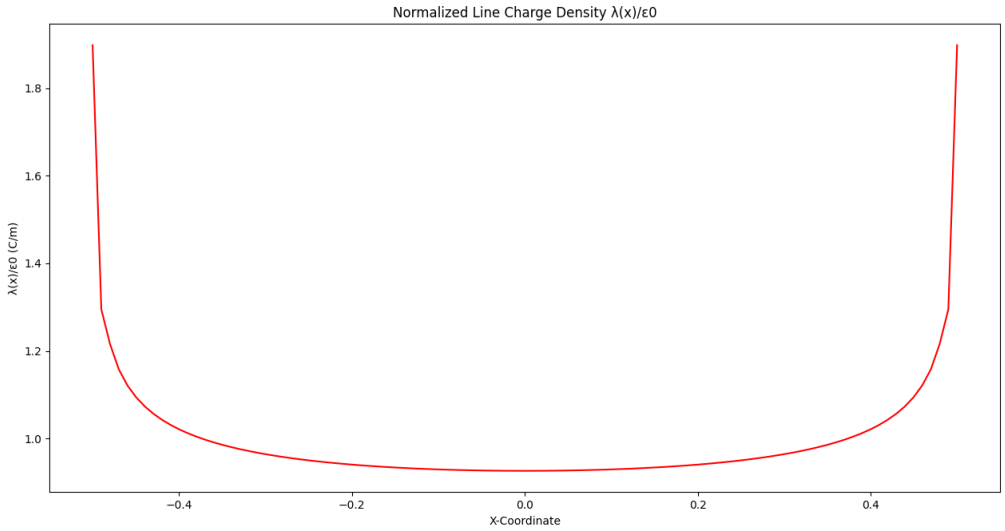
Παρουσιάζεται αποσπασματικά ο κώδικας **Python**:

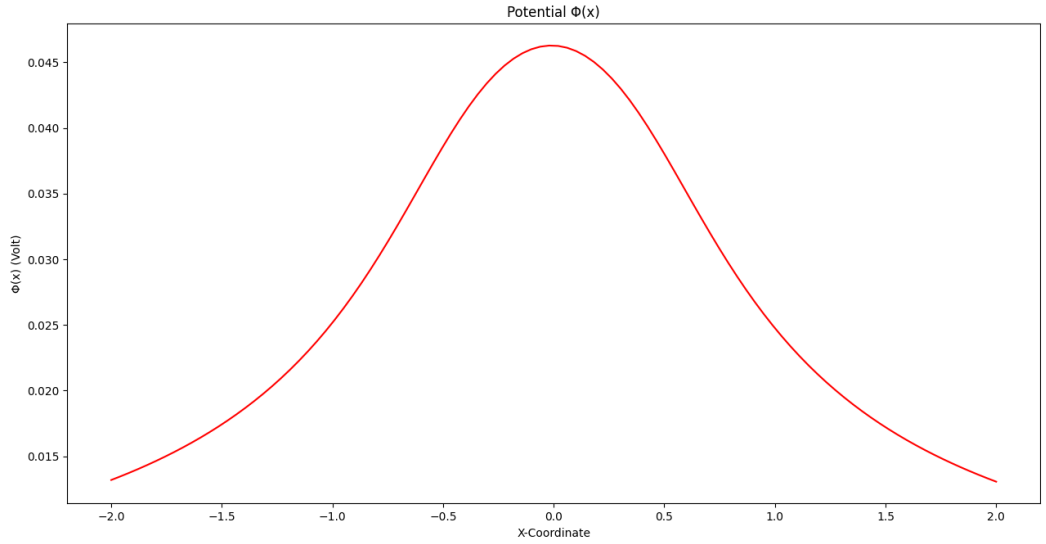
```
import numpy as np
from mpl_toolkits import mplot3d
import matplotlib
import matplotlib.pyplot as plt
from numpy.linalg import inv

L = 0.5
h = 0.5
N = 100
eps1 = 1
eps2 = 5
a = 0.0025
pi = np.pi
Dx = (2*L)/N
#Fi = 1/np.sum()
```

```
def x(i):
    xi = -L + i*Dx
    return xi

x = np.vectorize(x)

def x_bar(i):
    xi_bar = x(i) - Dx/2
    return xi_bar

x_bar = np.vectorize(x_bar)

def A(i, j):
    if (i != j):
        Aij = (Dx/(4*pi*eps1)) * (1/(abs(x_bar(i) - x_bar(j))) + ((eps1 - eps2)/(eps
1 + eps2)) * (1/(np.sqrt((x_bar(i) - x_bar(j))**2 + 4*h**2))))
        return Aij
    else:
        Aii = (1/(4*pi*eps1)) * (np.log((Dx/2 + np.sqrt(a**2 + (Dx/2)**2))/(Dx/-
2 + np.sqrt(a**2 + (Dx/2)**2))) + ((eps1-eps2)*Dx)/(2*h*(eps1+eps2)))
        return Aii

A = np.vectorize(A)
A_inv = inv(np.fromfunction(lambda i, j: A(i, j), (N, N), dtype=int))

I = np.vstack([1 for x in range(N)])

Fi = 1/np.sum(Dx*np.dot(A_inv, I))

def lamdak(i):
    lamda = Fi*(np.dot(A_inv, I)[i][0])
    return lamda

lamda = [n[0] for n in Fi*np.dot(A_inv, I)]
i = np.linspace(-N, N, N)
fig, ax = plt.subplots(figsize=(10, 10))
ax.plot((i/N)*L, lamda, color='red')

ax.set(xlabel='X-
Coordinate', ylabel='λ(x)/ε0 (C/m)', title='Normalized Line Charge Density λ(x)/ε0')

plt.show()

def F(x):
    f = (Dx/(4*eps2*pi)) * np.sum([(lamdak(i) * ((2*eps2)/(eps1 + eps2)) * ((1) / (n
p.sqrt((x - x_bar(i))**2 + h**2)))) for i in range(N)])
    return f
```

```
F = np.vectorize(F)

xv = np.linspace(-2, 2, N)
fig, ax = plt.subplots(figsize=(10, 10))
ax.plot(xv, F(xv), color='red')

ax.set(xlabel='X-Coordinate', ylabel='Φ(x) (Volt)', title='Potential Φ(x)')

plt.show()
```

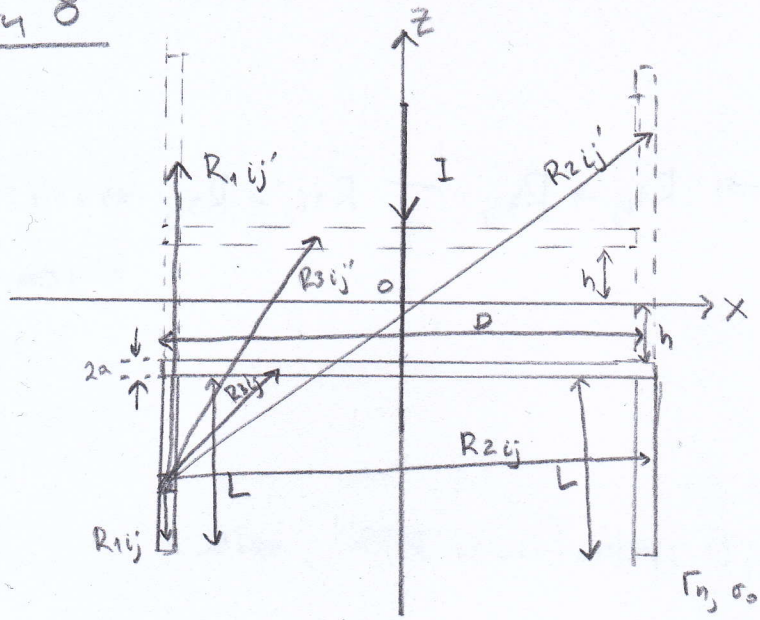
Παρουσιάζονται οι γραφικές παραστάσεις:





Άσκηση 8

α)



$N = 3$

$a = 0{,}005\,m$

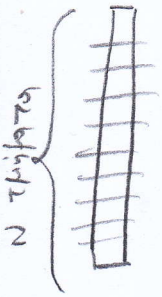
$L = 0{,}99\,m$

$h = 1{,}00\,m$

$D = 1{,}5\,m$

$\sigma_0 = \frac{1}{160}\ \frac{1}{\Omega m}$

$I = 500\,A$



- δραστ. πυκν. ρεύματος σταθερή σε κάθε στρώμα
- η κάθε ράβδος αποτελεί ισοδυναμική επιφάνεια.

Με την μέθοδο των ραών προκύπτει $\Phi_i = \sum_{j=1}^{N} (VDF)_{ij} I_j$ και επειδή $\Delta >> a$

πρώτα,

$$VDF_{ij} = \begin{cases} \frac{1}{4\pi\sigma_0}\Delta z\left(\frac{1}{R_{1ji}} + \frac{1}{R_{1j'i}} + \frac{1}{R_{2ji}} + \frac{1}{R_{2j'i}} + \frac{1}{R_{3ji}} + \frac{1}{R_{3j'i}}\right), & i \neq j \\ \frac{1}{4\pi\sigma_0}\left[\ln\left\{\frac{\frac{\Delta z}{2} + \sqrt{a^2 + \left(\frac{\Delta z}{2}\right)^2}}{-\frac{\Delta z}{2} + \sqrt{a^2 + \left(\frac{\Delta z}{2}\right)^2}}\right. + \frac{\Delta z}{R_{1ii'}} + \frac{\Delta z}{R_{2ii}} + \frac{\Delta z}{R_{2i'i}} + \frac{\Delta z}{R_{3ii}} + \frac{\Delta z}{R_{3ii}}\right], & i = j \end{cases}$$

όπου:

$R_{1ji} = |z_i - z_j|$

$R_{1j'i} = (z_i - z_j) + 2h$

$R_{2ji} = \sqrt{(z_i - z_j)^2 + D^2}$

$R_{2j'i} = \sqrt{[(z_i - z_j) + 2h]^2 + D^2}$

$R_{2ij} = \sqrt{(z_i - z_j)^2 + (D - 2a)^2}$

$R_{2ij'} = \sqrt{(z_i - z_j')^2 + (D - 2a)^2}$

$R_{3ij} = \sqrt{(z_i + h)^2 + \left(a - x_j - \frac{D}{2}\right)^2}$

$R_{3ij'} = \sqrt{(z_i - h)^2 + \left(a - x_j - \frac{D}{2}\right)^2}$

Έπειτα,

$$V_{DF_{ij}} = \frac{1}{4\pi\sigma_0}\left[\sum_{j=1}^{N}\frac{I_j \Delta z}{R_{1ij}} + \sum_{j=1}^{N}\frac{I_j \Delta z}{R_{1ij'}} + \sum_{j=N+1}^{2N}\frac{I_j \Delta x}{|x_i - x_j|} + \sum_{j=N+1}^{2N}\frac{I_j \Delta x}{\sqrt{(2h)^2 + (x_i - x_j)^2}} + \right.$$

$$\left. + \sum_{j=2N+1}^{3N}\frac{I_j \Delta z}{R_{2ij}} + \sum_{j=2N+1}^{3N}\frac{I_j \cdot \Delta z}{R_{2ij'}}\right.$$

όπου

$$R_{1ij} = \sqrt{(z_j - h)^2 + \left(a - x_i - \frac{D}{2}\right)^2}$$

$$R_{1ij'} = \sqrt{(z_j' + h)^2 + \left(a - x_i - \frac{D}{2}\right)^2}$$

$$R_{2ij} = \sqrt{(z_j - h)^2 + \left(\frac{D}{2} - a - x_i\right)^2}$$

$$R_{2ij'} = \sqrt{(z_j' + h)^2 + \left(\frac{D}{2} - a - x_i\right)^2}$$

$\Rightarrow R_{1ij} + R_{3ij} + R_{1ij'} + R_{3ij'} \Rightarrow$ συμμετρικά $\Rightarrow$ ίσα

Άρα, για $i = 2N, \ldots, 2N+1, \ldots, 3N$ για N τμήματα (ανά την ράβδο), ισχύει:

$$[\Phi] = [VDF] \cdot [I] \xrightarrow[(\Phi_1 = \Phi_2 = \ldots \Phi_N)]{\text{ισοδ. επιφ.}} [\hat{I}] = \Phi [VDF]^{-1} [1]$$

οι ράβδοι είναι ίδιες οπότε το ρεύμα μοιράζεται ίσα στις τρεις.

Τέλος

$$\Phi_i = \frac{1}{4\pi\varepsilon_0}\left[\sum_{j=1}^{N} \frac{I_j \Delta z}{R_{1ij}} + \sum_{j=1}^{N} \frac{I_j \Delta z}{R_{1ij'}} + \sum_{j=N+1}^{2N} \frac{I_j \Delta x}{R_{3ij}} + \sum_{j=N+1}^{2N} \frac{I_j \Delta x}{R_{3ij'}} + \right.$$

$$\left. + \sum_{j=2N+1}^{3N} \frac{I_j \Delta z}{|z_i - z_j|} + \sum_{j=2N+1}^{3N} \frac{I_j \Delta z}{|z_i - z_j'|}\right]$$

όπου

$$R_{1ij} = \sqrt{(z_i - z_j)^2 + (D-2a)^2}$$

$$R_{1ij'} = \sqrt{(z_i - z_j)^2 + (D-2a)^2}$$

$$R_{3ij} = \sqrt{(z_i - h)^2 + \left(\frac{D}{2} - a - x_j\right)^2}$$

$$R_{3ij'} = \sqrt{(z_i + h)^2 + \left(\frac{D}{2} - a - x_j\right)^2}$$

Θέτουμε $\Phi = IV$, και ρεύμα I' σε μία ράβδο: $I_i = \hat{I}_i \left[\frac{I'}{\left(\sum_{i=1}^{N} \hat{I}_i\right) \Delta z}\right]$

$$VDF = \begin{bmatrix} \ldots \end{bmatrix} \longrightarrow VDF^{-1} = \begin{bmatrix} \ldots \end{bmatrix}$$

$$[\hat{I}] = [VDF]^{-1} \cdot [1] = \begin{bmatrix} \ldots \end{bmatrix} \longrightarrow I_i = \ldots$$

8η Άσκηση

α) Η επίλυση φαίνεται στις χειρόγραφες σελίδες.

β – γ) Δυναμικό & Αντίσταση Γείωσης & Πίνακας

Παρατίθεται ενδεικτικά ο κώδικας **Python**:

```
import numpy as np
import matplotlib.pyplot as plt
from numpy.linalg import inv

L = 0.99
h = 1
N = 3
I = np.array([0 for i in range(2*N)])
VDF_inv = np.array([[0 for j in range(2*N)] for i in range(2*N)])
D = 1.5
a = 0.005
pi = np.pi
Dz = L/N
s0 = 1/160

def z_bar(i):
    if (i <= N):
        zi = -h + (L/2) - i*Dz
        return zi + (Dz/2)
    return z_bar(i - N)

# the horizontal bar has not been calculated here
def VDF(i, j):
    if ((0 < i <= N and N < j <= 2*N) or (N < i <= 2*N and 0 < j <= N)):
        fraction1 = 1 / (np.sqrt((z_bar(i) - z_bar(j))**2 + D**2))
        fraction2 = 1 / (np.sqrt((z_bar(i) + z_bar(j))**2 + D**2))
        return (Dz / (4*pi*s0)) * (fraction1 + fraction2)
    else:
        fraction1 = 1 / (abs(z_bar(i) + z_bar(j)))
        if (i == j):
            root = np.sqrt(a**2 + (Dz/2)**2)
            log = np.log(((Dz/2) + root) / ((Dz/-2) + root))
            return (1/(4*pi*s0)) * (log + (Dz*fraction1))
        fraction2 = 1 / (abs(z_bar(i) - z_bar(j)))
        return (Dz / (4*pi*s0)) * (fraction1 + fraction2)

def F(x, y, z):
    def r1(i):
        return np.sqrt((x+(D/2))**2 + y**2 + (z - z_bar(i))**2)

    def r2(i):
        return np.sqrt((x-(D/2))**2 + y**2 + (z - z_bar(i))**2)
```

```
    return 2*(Dz / (4*pi*s0)) * np.sum([(1 / r1(i) + 1 / r2(i)) * I[i - 1] for i in
range(1, N + 1)])

F = np.vectorize(F)

def plot_a_lot1():
    x = np.linspace(-3, 3, 100)
    fig, ax = plt.subplots()
    ax.plot(x, F(x, 0, 0))
    ax.set(xlabel='X(m)', ylabel='Phi(V)', title='Potential Φ(x), N=' + str(N))
    ax.grid()
    fig.savefig("ask8_Fi_N=" + str(N) + ".png")
    plt.clf()

def plot_a_lot2():
    x = np.array([z_bar(i) for i in range(1, N + 1)])
    fig, ax = plt.subplots()
    ax.plot(x, I[:N])
    ax.set(xlabel='Z(m)', ylabel='I(A/m)',
           title='Linear Power Distribution I(z), N=' + str(N))
    ax.grid()
    fig.savefig("ask8_I_N=" + str(N) + ".png")
    plt.clf()

def row_sum_vdf(i):
    sum = 0
    for j in range(2*N):
        sum += VDF_inv[i][j]
    return sum

def updateN(n):
    global N
    global Dz
    N = n
    Dz = L / n

# calculation of array
pinakas = [[]]
pinakas[0].append('N')
pinakas[0].append('Phi')
pinakas[0].append('Rg')
pinakas[0].append('F(0,0,0)')

for i in range(16):
    if (i == 0):
        updateN(3)
    else:
        updateN(5 * i)
```

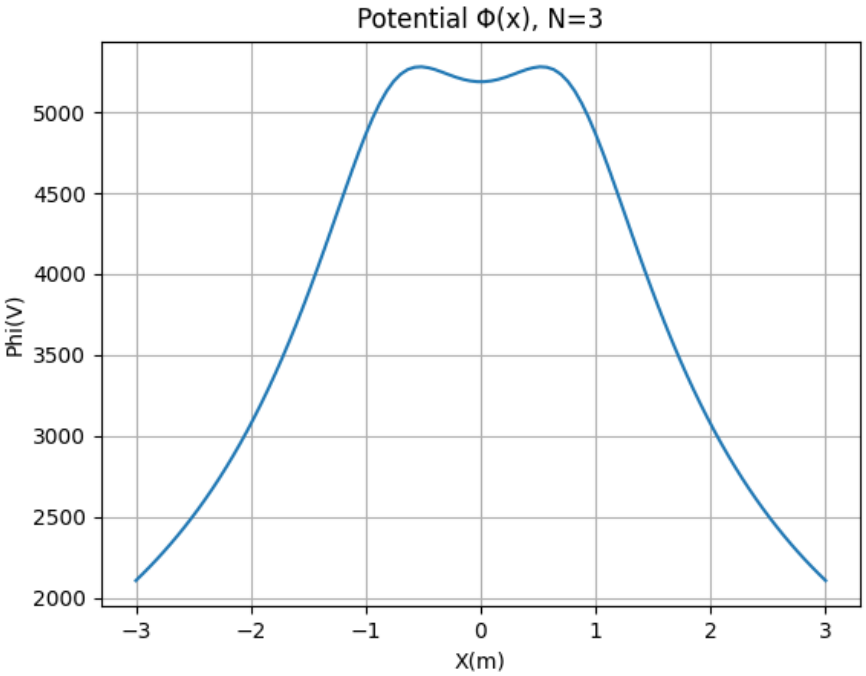
```
    pinakas.append([])
    row = pinakas[i + 1]

    VDF_inv = inv(
        np.array([[VDF(i, j) for j in range(1, 2*N + 1)] for i in range(1, 2*N + 1)]
))

    Phi = 250 / (Dz*np.sum(VDF_inv))

    I = Phi * np.array([row_sum_vdf(i) for i in range(2*N)])

    Rg = Phi / (250)

    row.append(N)
    row.append(Phi)
    row.append(Rg)
    row.append(F(0, 0, 0))

    plot_a_lot1()
    plot_a_lot2()

pinakas = np.array(pinakas)
print(pinakas)
```

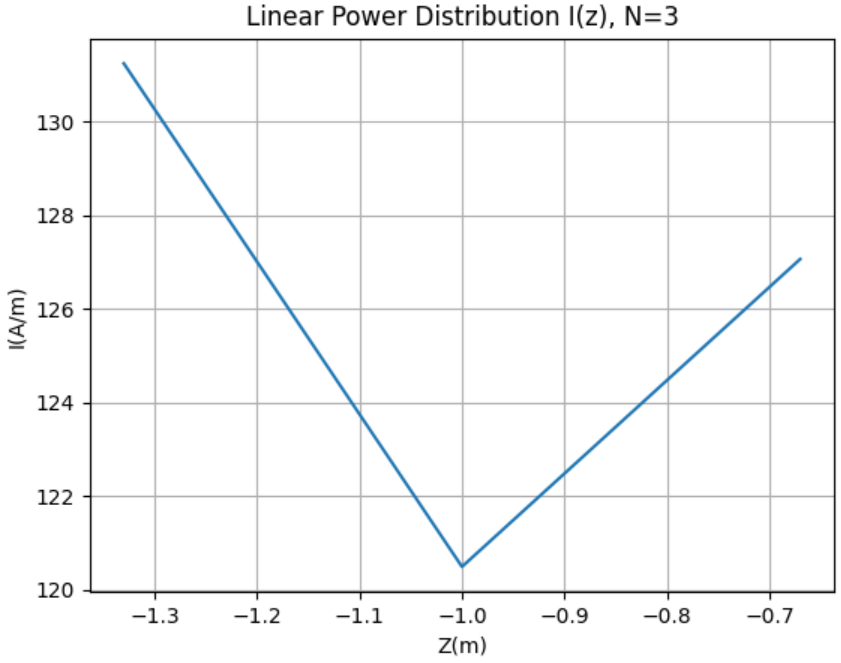
Τέλος, παρουσιάζεται και ο πίνακας:

```
[['N' 'Phi' 'Rg' 'F(0,0,0)']
 ['3' '18637.009046333762' '74.54803618533505' '5187.985389608774']
 ['5' '18418.743854277007' '73.67497541710803' '5196.43358171262']
 ['10' '18228.776902356214' '72.91510760942485' '5201.432957102355']
 ['15' '18159.22601053296' '72.63690404213183' '5203.010550514616']
 ['20' '18127.174547547955' '72.50869819019182' '5203.843748665314']
 ['25' '18113.744289648967' '72.45497715859587' '5204.384558779805']
 ['30' '18111.959792860303' '72.44783917144122' '5204.776776789752']
 ['35' '18118.496500359724' '72.4739860014389' '5205.081476536981']
 ['40' '18131.474215488266' '72.52589686195306' '5205.329417193497']
 ['45' '18149.68398769195' '72.5987359507678' '5205.537958812186']
 ['50' '18172.267398364456' '72.68906959345783' '5205.717742110403']
 ['55' '18198.56720530523' '72.79426882122092' '5205.875718011843']
 ['60' '18228.05163608158' '72.91220654432632' '5206.01666883745']
 ['65' '18260.273337991777' '73.0410933519671' '5206.144034631506']
 ['70' '18294.845733401722' '73.17938293360689' '5206.260391674597']
 ['75' '18331.428514300525' '73.3257140572021' '5206.36774548947']]
```

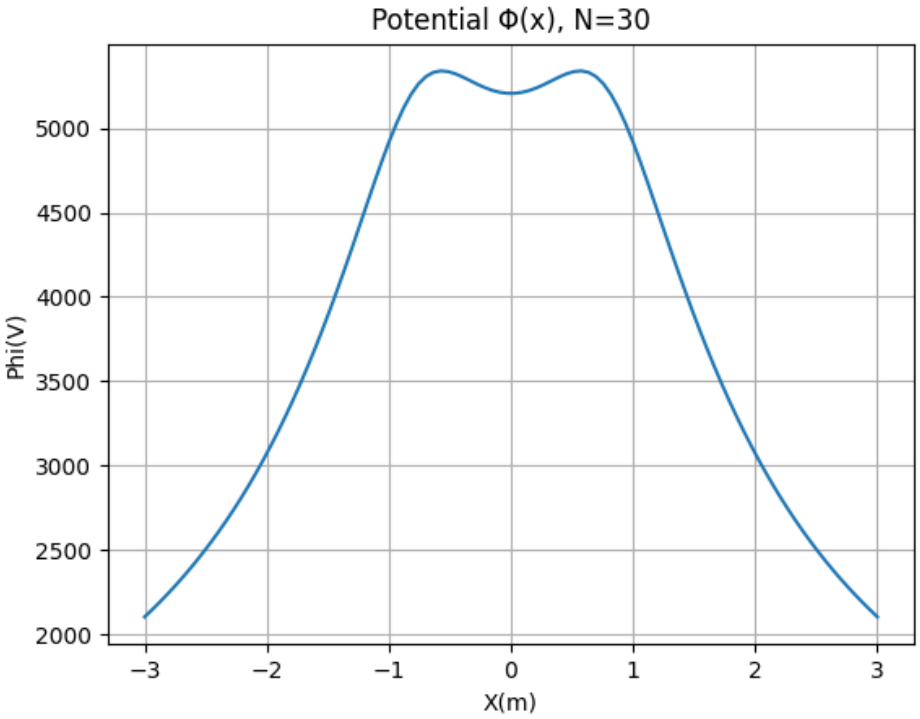
Και η γραφική παράσταση είναι:

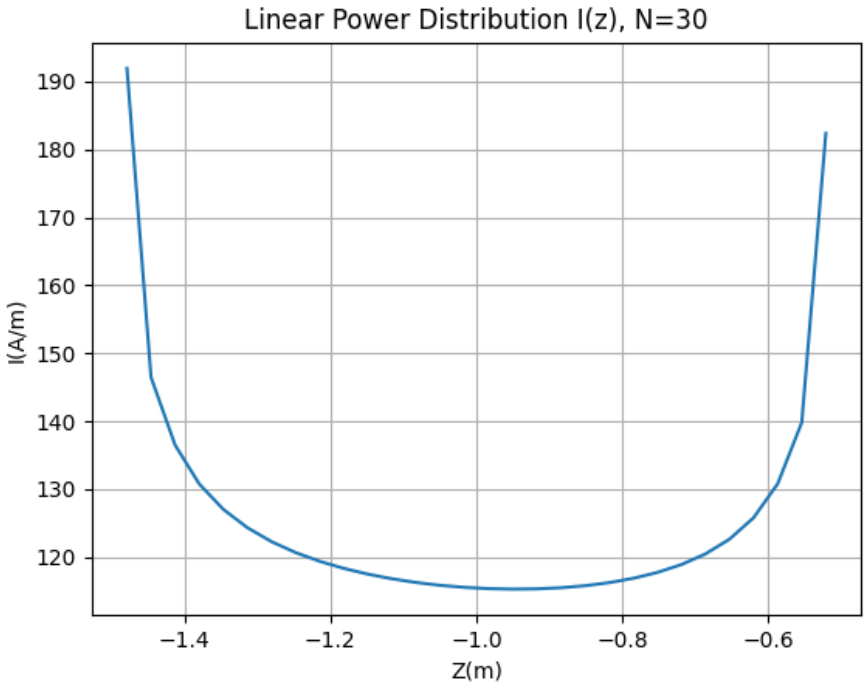
Ενδεικτικά για N = 3:



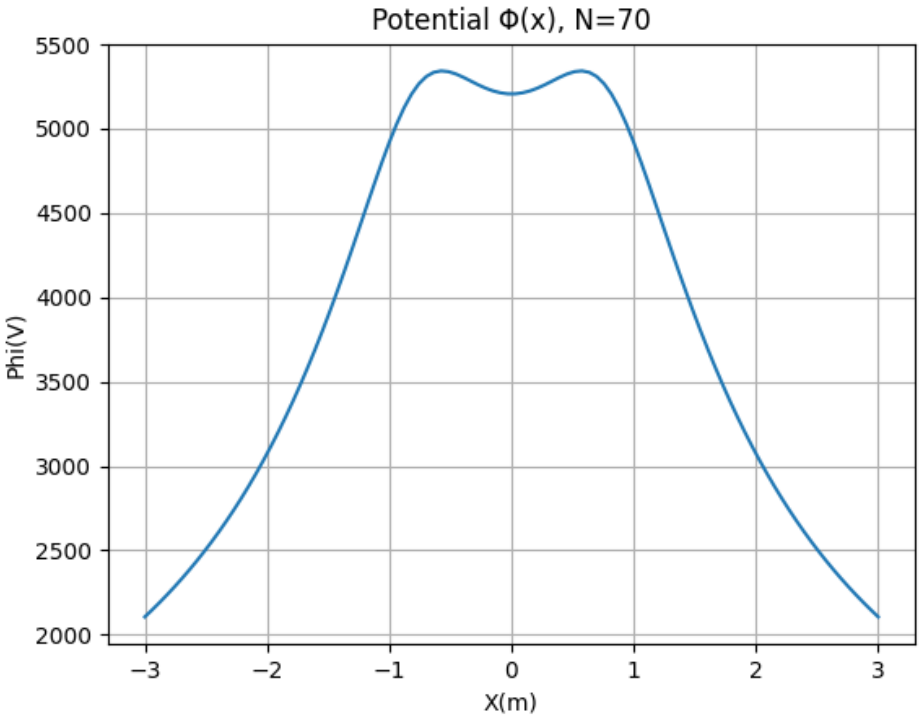


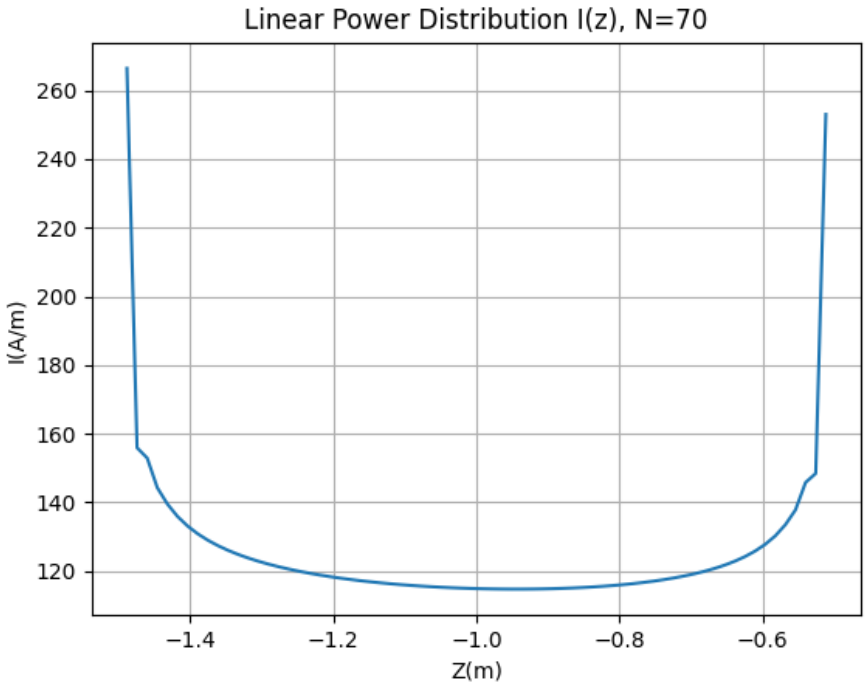
Ενδεικτικά για N = 30:





Ενδεικτικά για N = 70:





Επιπλέον, παρακάτω φαίνεται ένα ενδεικτικό κομμάτι μιας πιο ολοκληρωμένης υλοποίησης σε **Matlab**:

```
clear;
a = 0.005; L = 0.99; h = 1; D = 1.5;
N = 3; V = 1; s0 = 1/160;
Font = 12; I = 500;

Dz = L/N; Dx = D/N;
x = - (0.5)*D:Dx:(0.5)*D;
z = -h-a-L:Dz:-h-a;

for i=1:N
  x(i) = (1/2)*( x(i) + x(i+1) );
  z(i) = (1/2)*( z(i) + z(i+1) );
  z_mirror(i) = -(1/2)*( z(i) + z(i+1) );
endfor

% VDF calculation
for i=1:N
  for j=1:3*N
    if i==j
      R1 = sqrt((Dz/2)^2 + a^2) + (Dz/2);
      R2 = sqrt((Dz/2)^2 + a^2) + -(Dz/2);
      R3 = abs(z(i) - z_mirror(j));
      VDF(i, j) = log(R1/R2) + (Dz/R2);

    else if j<N+1
      R1 = abs(z(i) - z(j));
      R2 = abs(z(i) - z_mirror(j));
      VDF(i, j) = Dz/R1 + Dz/R2;

    else if j<2*N+1
      R1 = sqrt((z(i) + h)^2 + (-D/2 + a - x(j-N))^2);
      R2 = sqrt((z(i) - h)^2 + (-D/2 + a - x(j-N))^2);
      VDF(i, j) = Dx/R1 + Dx/R2;

    else if j<3*N+1
      R1 = sqrt((z(i) - z(j-2*N))^2 + (D - 2*a)^2);
      R2 = sqrt((z(i) - z_mirror(j-2*N))^2 + (D - 2*a)^2);
      VDF(i, j) = Dz/R1 + Dz/R2;
    endif
    endif
    endif
    endif
  endfor
endfor
```