NAME: Utkarsh Jaiswal

ROLL NO: 18EX20030

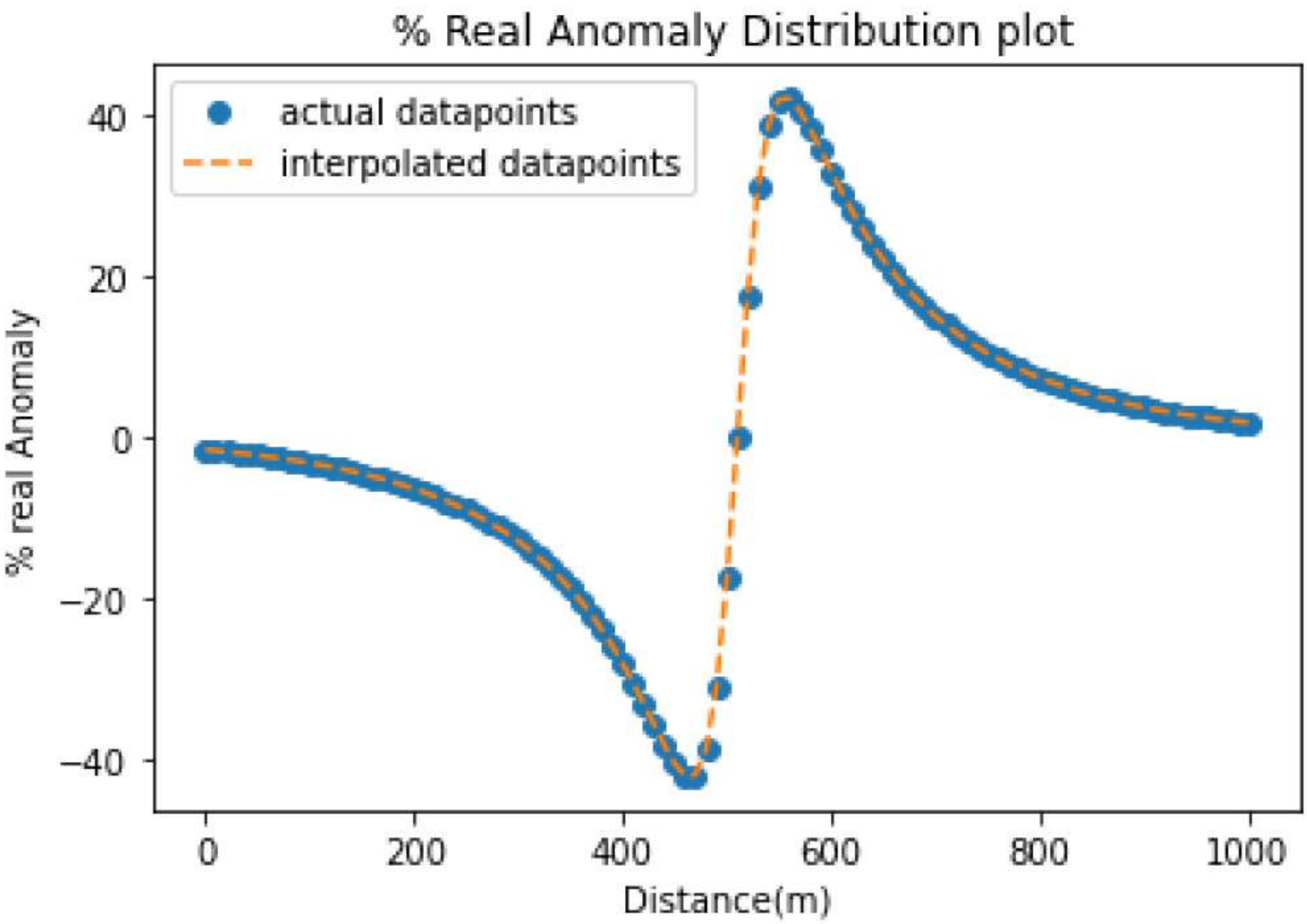
Lab Assignment

```python
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import math
data= pd.read_excel('content/vlfdata.xlsx')
data
```

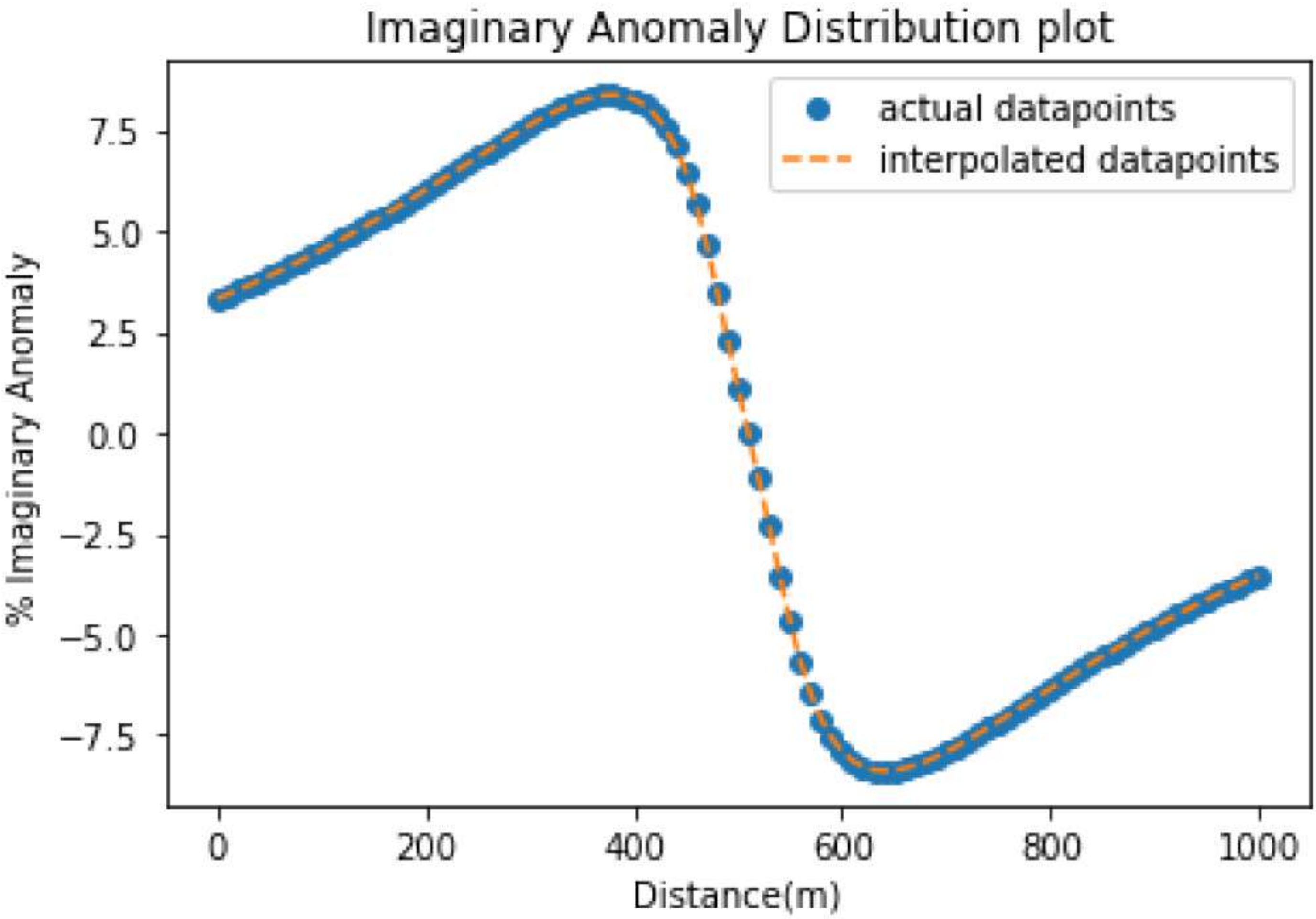
```
     dist     real  Imaginary
0       0  -1.5538     3.3208
1      10  -1.6770     3.4275
2      20  -1.8077     3.5373
3      30  -1.9463     3.6501
4      40  -2.0929     3.7660
..    ...      ...        ...
96    960   2.4119    -4.0129
97    970   2.2466    -3.8909
98    980   2.0912    -3.7719
99    990   1.9445    -3.6560
100  1000   1.8059    -3.5433

[101 rows x 3 columns]
```

```python
real = data['real']
img = data['Imaginary']
```

```python
from scipy.interpolate import interp1d
x = data['dist']
interpol_real= interp1d(x, real)
interpol_img = interp1d(x, img)
xnew= np.arange(0,1001,2)
plt.plot(x,real,'o',xnew, interpol_real(xnew),'--')
plt.legend(['actual datapoints', 'interpolated datapoints'])
plt.xlabel('Distance(m)')
plt.ylabel('% real Anomaly')
plt.title(' % Real Anomaly Distribution plot')
plt.show()
```

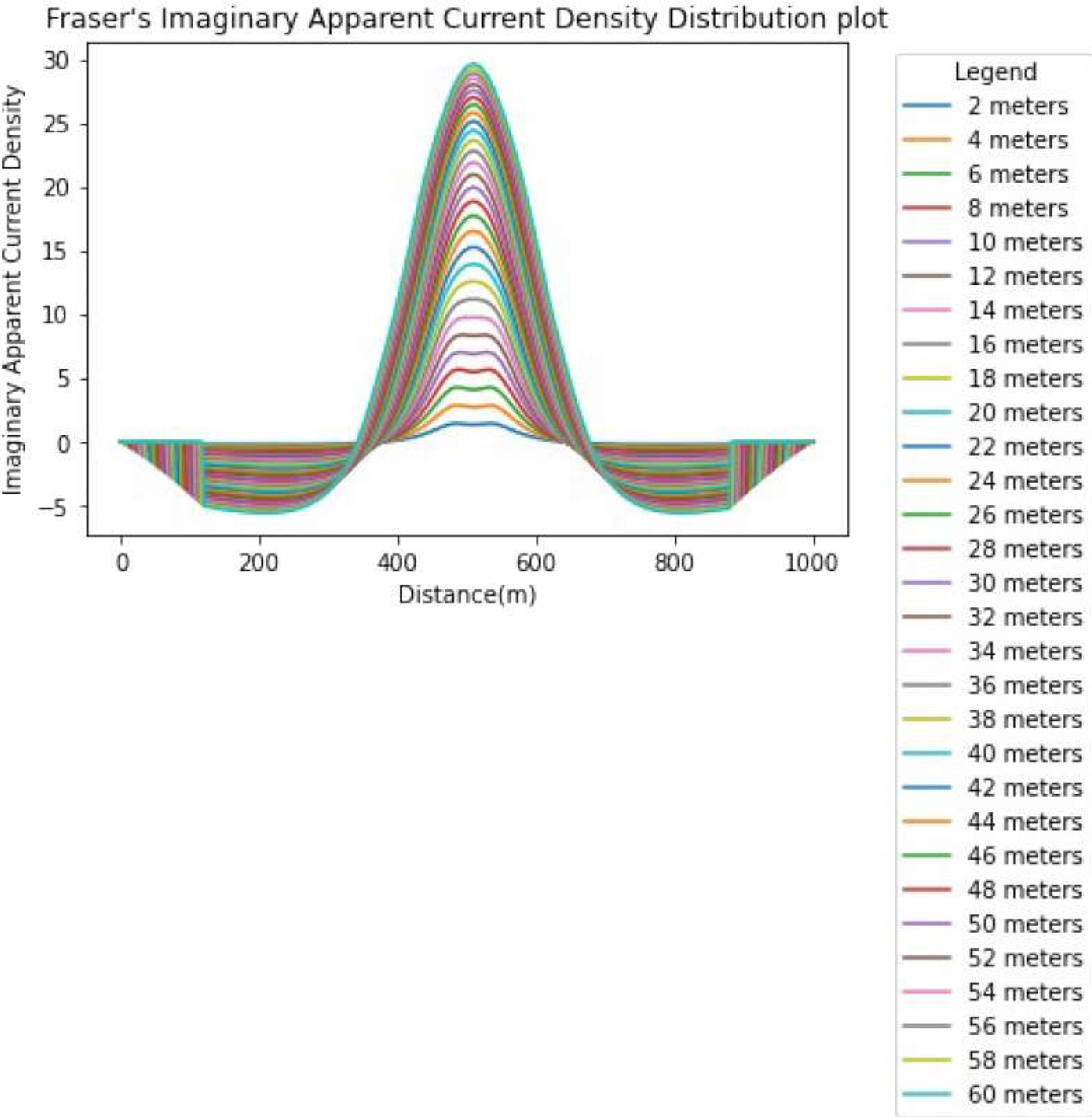

% Real Anomaly Distribution plot
actual datapoints
interpolated datapoints
40
20
0
−20
−40
% real Anomaly
0
200
400
600
800
1000
Distance(m)

```
plt.plot(x, img,'o', xnew,interpol_img(xnew),'--')
plt.legend(['actual datapoints', 'interpolated datapoints'])
plt.xlabel('Distance(m)')
plt.ylabel('% Imaginary Anomaly')
plt.title('Imaginary Anomaly Distribution plot')
plt.show()
```


Imaginary Anomaly Distribution plot
actual datapoints
interpolated datapoints
% Imaginary Anomaly
Distance(m)
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
0
200
400
600
800
1000

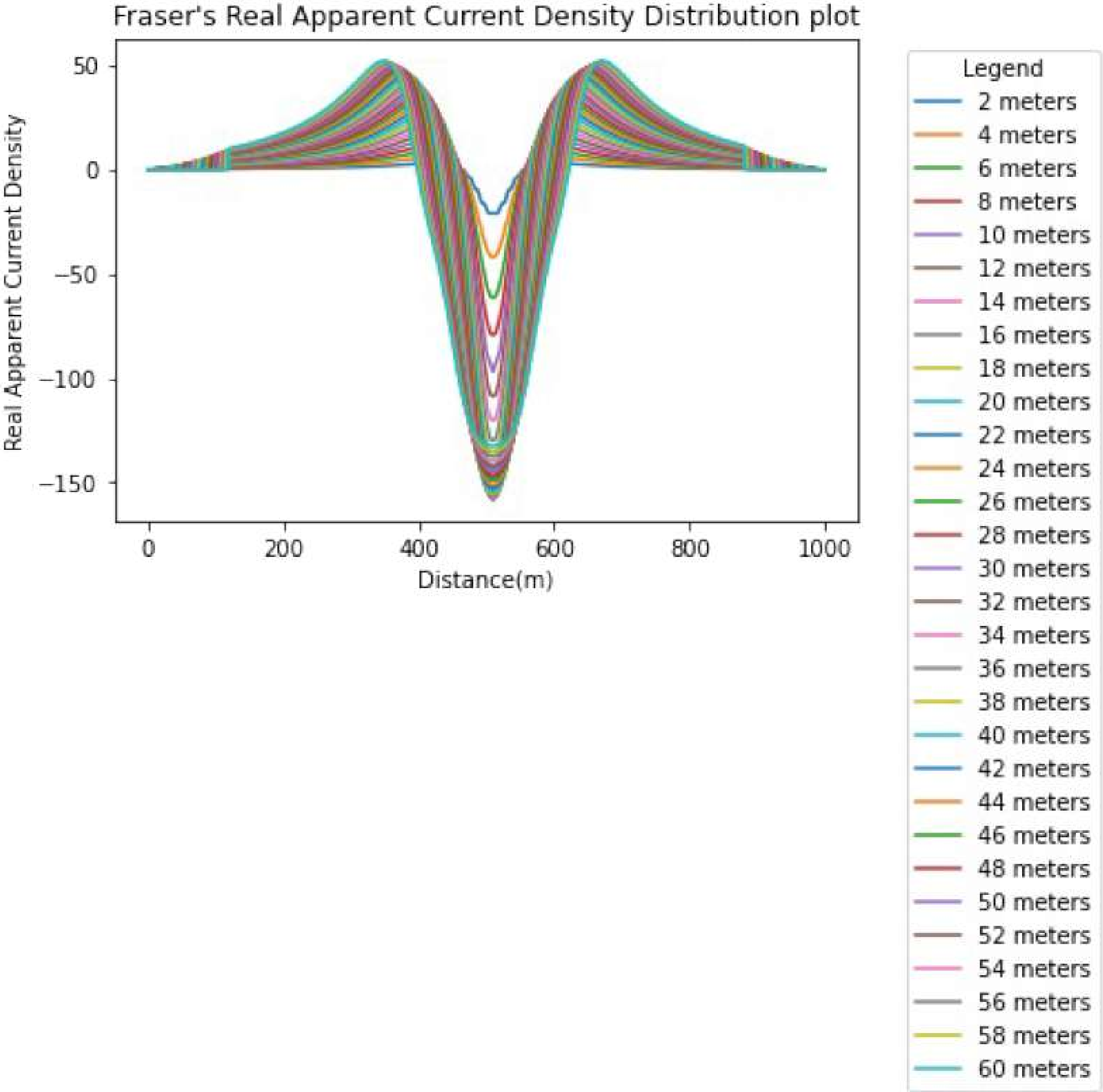
```
real=interpol_real(xnew)
img=interpol_img(xnew)



def fraser_filter(z):
 J_app_real=[]
 J_app_img=[]
 for j in range(1,z+1):
    J_app_real_layer=[]
    J_app_img_layer=[]
    for s in range(2*j):
        J_app_real_layer.append(0)
        J_app_img_layer.append(0)
    for i in range(2*j,501-2*j,1):
        J_app_real_layer.append(real[i-2*j]+real[i-j]-real[i+2*j]-
real[i+j])
        J_app_img_layer.append(img[i-2*j]+img[i-j]-img[i+2*j]-
img[i+j])
    for s in range(2*j):
        J_app_real_layer.append(0)
        J_app_img_layer.append(0)
    J_app_img.append(J_app_img_layer)
    J_app_real.append(J_app_real_layer)
 return J_app_img,J_app_real
```

```
I,R=fraser_filter(30)

for i in range (30):
 plt.plot(xnew, I[i])
plt.legend([str(2*(i+1))+" meters" for i in range(30)],
title='Legend', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xlabel('Distance(m)')
plt.ylabel('Imaginary Apparent Current Density')
plt.title("Fraser's Imaginary Apparent Current Density Distribution
plot")
plt.show()
```

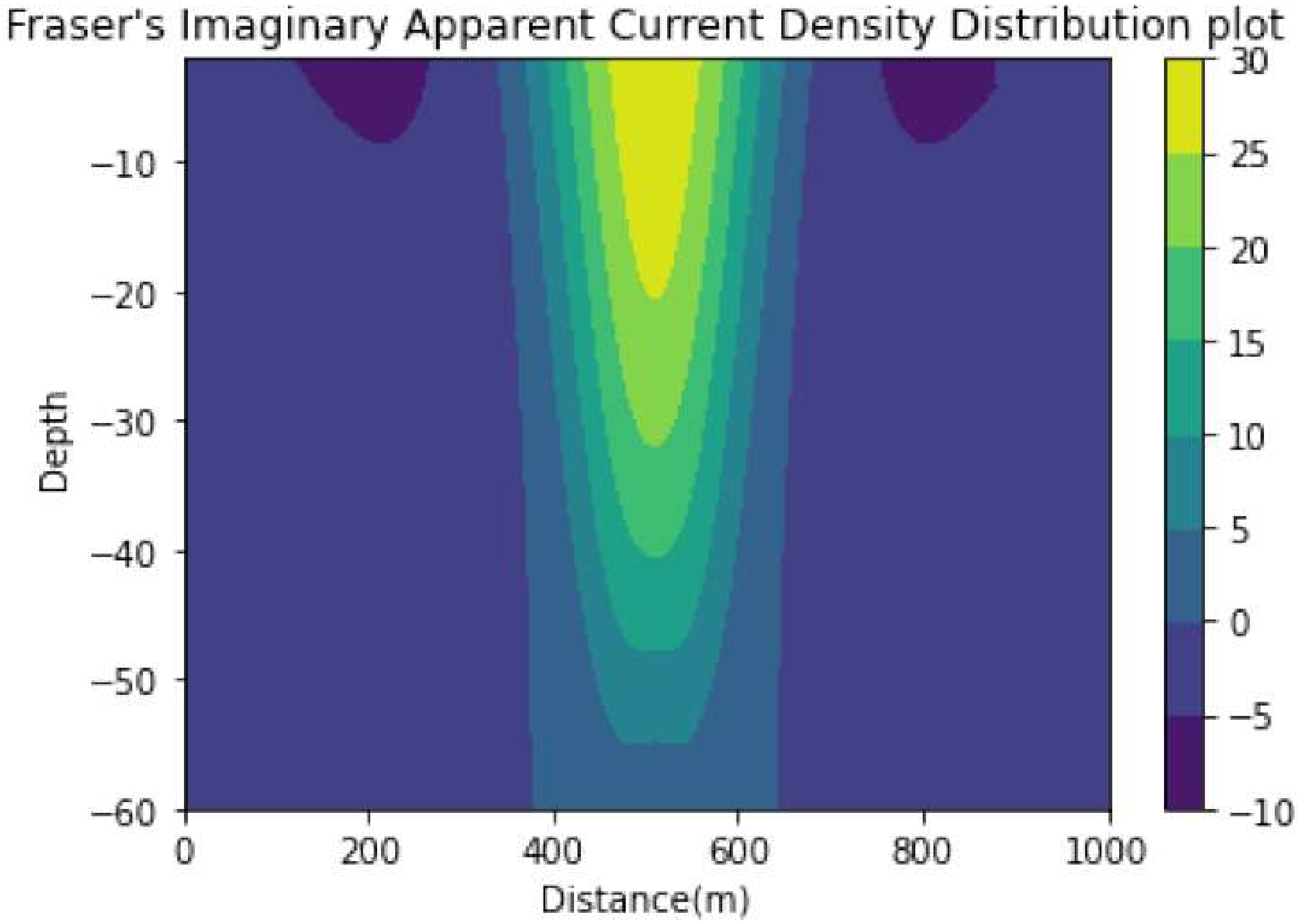

Fraser's Imaginary Apparent Current Density Distribution plot
Imaginary Apparent Current Density
Distance(m)
30
25
20
15
10
5
0
−5
0
200
400
600
800
1000
Legend
2 meters
4 meters
6 meters
8 meters
10 meters
12 meters
14 meters
16 meters
18 meters
20 meters
22 meters
24 meters
26 meters
28 meters
30 meters
32 meters
34 meters
36 meters
38 meters
40 meters
42 meters
44 meters
46 meters
48 meters
50 meters
52 meters
54 meters
56 meters
58 meters
60 meters

```python
for i in range (30):
  plt.plot(xnew, R[i])
plt.legend([str(2*(i+1))+" meters" for i in range(30)],
             title='Legend', bbox_to_anchor=(1.05, 1), loc='upper
```

```
left')
plt.xlabel('Distance(m)')
plt.ylabel('Real Apparent Current Density')
plt.title("Fraser's Real Apparent Current Density Distribution plot")
plt.show()
```

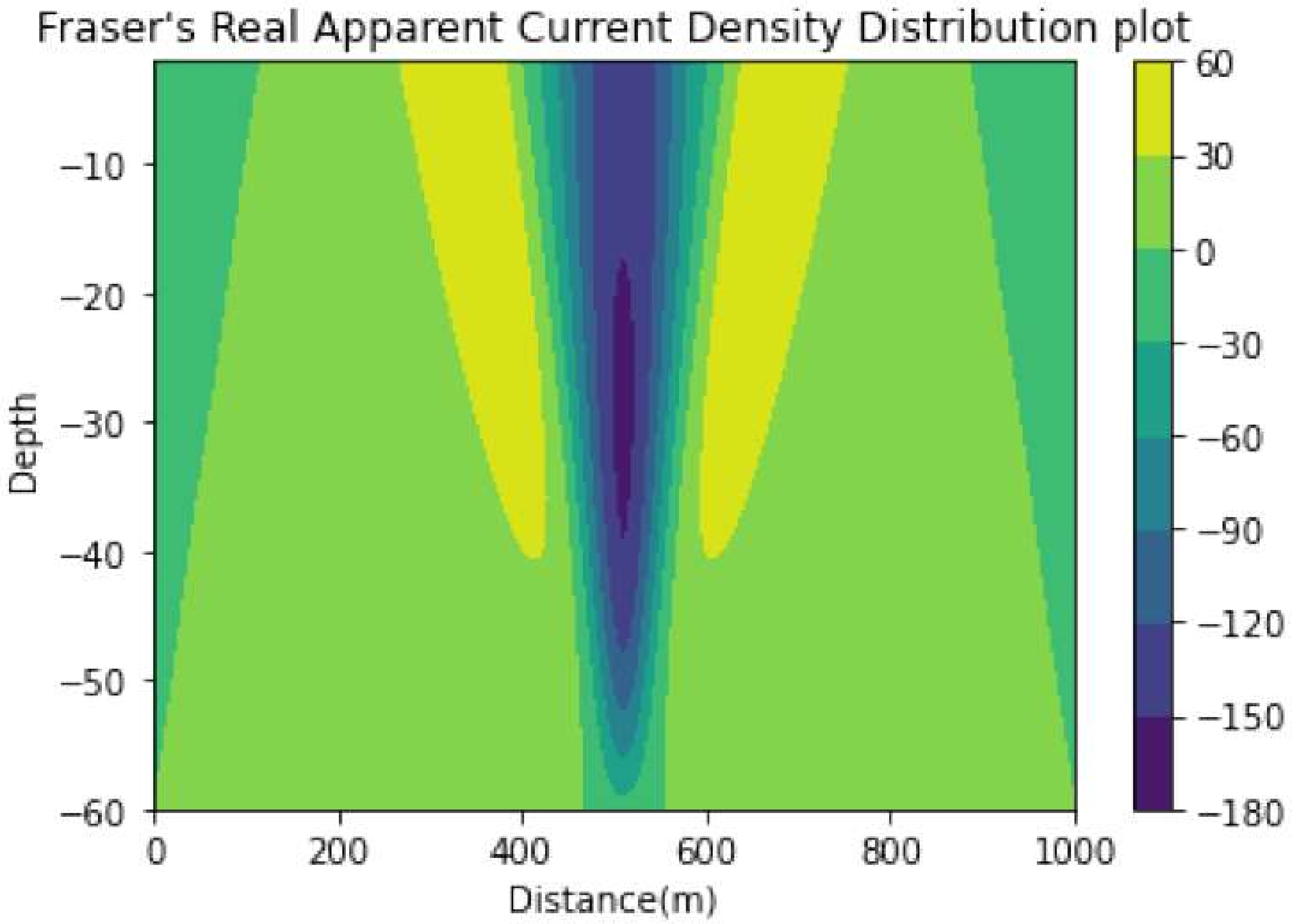

Fraser's Real Apparent Current Density Distribution plot
Real Apparent Current Density
Distance(m)
50
0
−50
−100
−150
0
200
400
600
800
1000
Legend
2 meters
4 meters
6 meters
8 meters
10 meters
12 meters
14 meters
16 meters
18 meters
20 meters
22 meters
24 meters
26 meters
28 meters
30 meters
32 meters
34 meters
36 meters
38 meters
40 meters
42 meters
44 meters
46 meters
48 meters
50 meters
52 meters
54 meters
56 meters
58 meters
60 meters

```
a = -np.array(np.arange(60,1,-2))
b = np.array(xnew)
b,a = np.meshgrid(b, a)
plt.contourf(b, a,np.array(I))
plt.colorbar()
plt.xlabel('Distance(m)')
plt.ylabel('Depth')
plt.title("Fraser's Imaginary Apparent Current Density Distribution
plot")
plt.show()
```



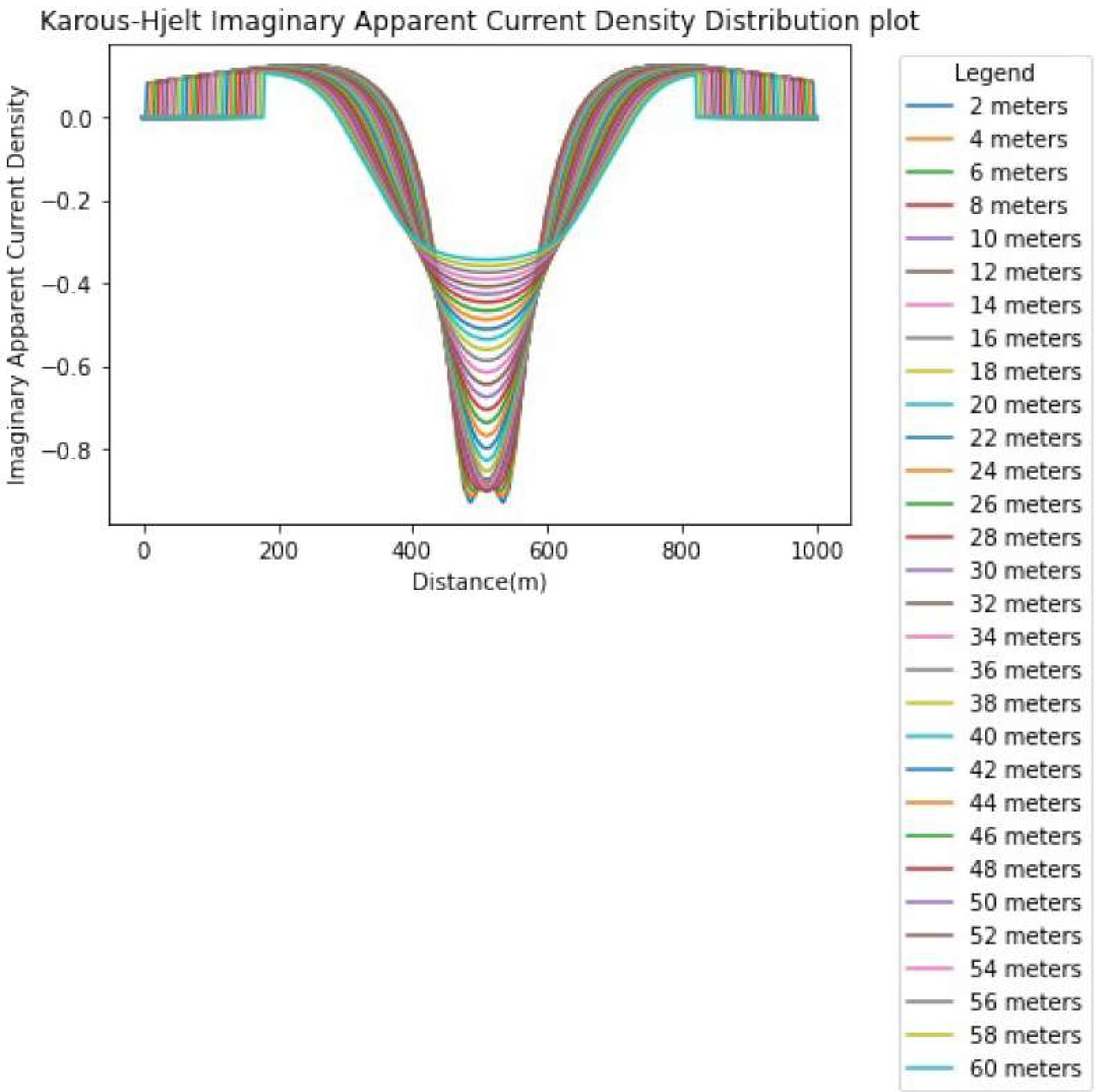
Fraser's Imaginary Apparent Current Density Distribution plot
Depth
Distance(m)
−10
−20
−30
−40
−50
−60
0
200
400
600
800
1000
30
25
20
15
10
5
0
−5
−10

```
plt.contourf(b, a,np.array(R))
plt.colorbar()
plt.xlabel('Distance(m)')
plt.ylabel('Depth')
plt.title("Fraser's Real Apparent Current Density Distribution plot")
plt.show()
```


Fraser's Real Apparent Current Density Distribution plot
Depth
−10
−20
−30
−40
−50
−60
0
200
400
600
800
1000
Distance(m)
60
30
0
−30
−60
−90
−120
−150
−180

```
def KH_filter(z):
  J_app_real=[]
  J_app_img=[]
  for j in range(1,z+1):
    J_app_real_layer=[]
    J_app_img_layer=[]
    for s in range(3*j):
      J_app_real_layer.append(0)
      J_app_img_layer.append(0)
    for i in range(3*j,501-3*j,1):
      J_app_real_layer.append((2*np.pi/j)*(-0.102*real[i-3*j]
+0.059*real[i-2*j]-0.561*real[i-j]+0.102*real[i+3*j]-0.059*real[i+2*j]
+0.561*real[i+j]))
      J_app_img_layer.append((2*np.pi/j)*(-0.102*img[i-3*j]
+0.059*img[i-2*j]-0.561*img[i-j]+0.102*img[i+3*j]-0.059*img[i-2*j]
+0.561*img[i-j]))
    for s in range(3*j):
      J_app_real_layer.append(0)
      J_app_img_layer.append(0)
    J_app_img.append(J_app_img_layer)
    J_app_real.append(J_app_real_layer)
  return J_app_img,J_app_real


I_kh,R_kh=KH_filter(30)
```

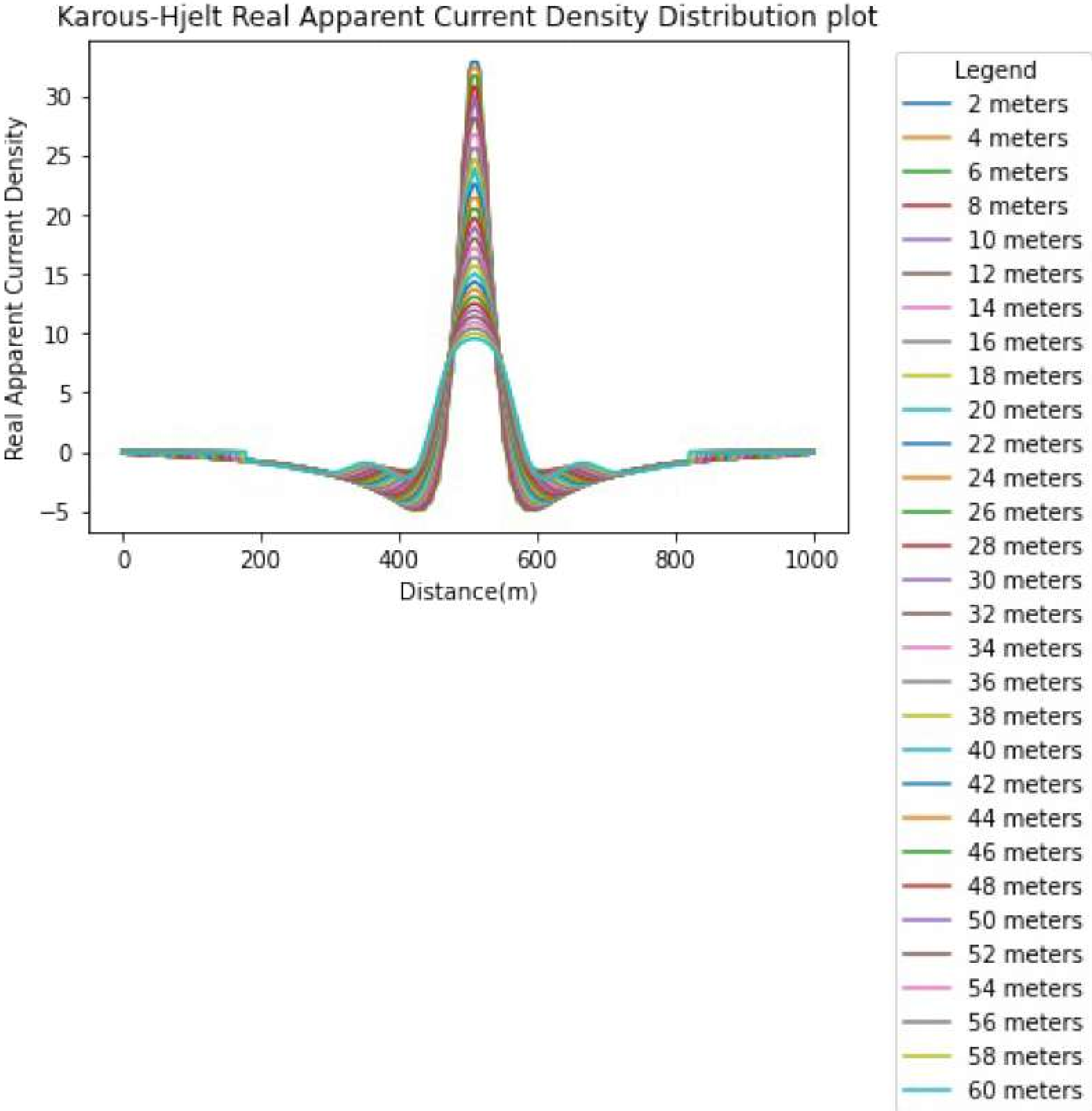
```
for i in range (30):
  plt.plot(xnew, I_kh[i])
plt.legend([str(2*(i+1))+" meters" for i in range(30)],
title='Legend', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xlabel('Distance(m)')
plt.ylabel('Imaginary Apparent Current Density')
plt.title("Karous-Hjelt Imaginary Apparent Current Density
Distribution plot")
plt.show()
```



Karous-Hjelt Imaginary Apparent Current Density Distribution plot
Imaginary Apparent Current Density
0.0
−0.2
−0.4
−0.6
−0.8
0
200
400
600
800
1000
Distance(m)
Legend
2 meters
4 meters
6 meters
8 meters
10 meters
12 meters
14 meters
16 meters
18 meters
20 meters
22 meters
24 meters
26 meters
28 meters
30 meters
32 meters
34 meters
36 meters
38 meters
40 meters
42 meters
44 meters
46 meters
48 meters
50 meters
52 meters
54 meters
56 meters
58 meters
60 meters

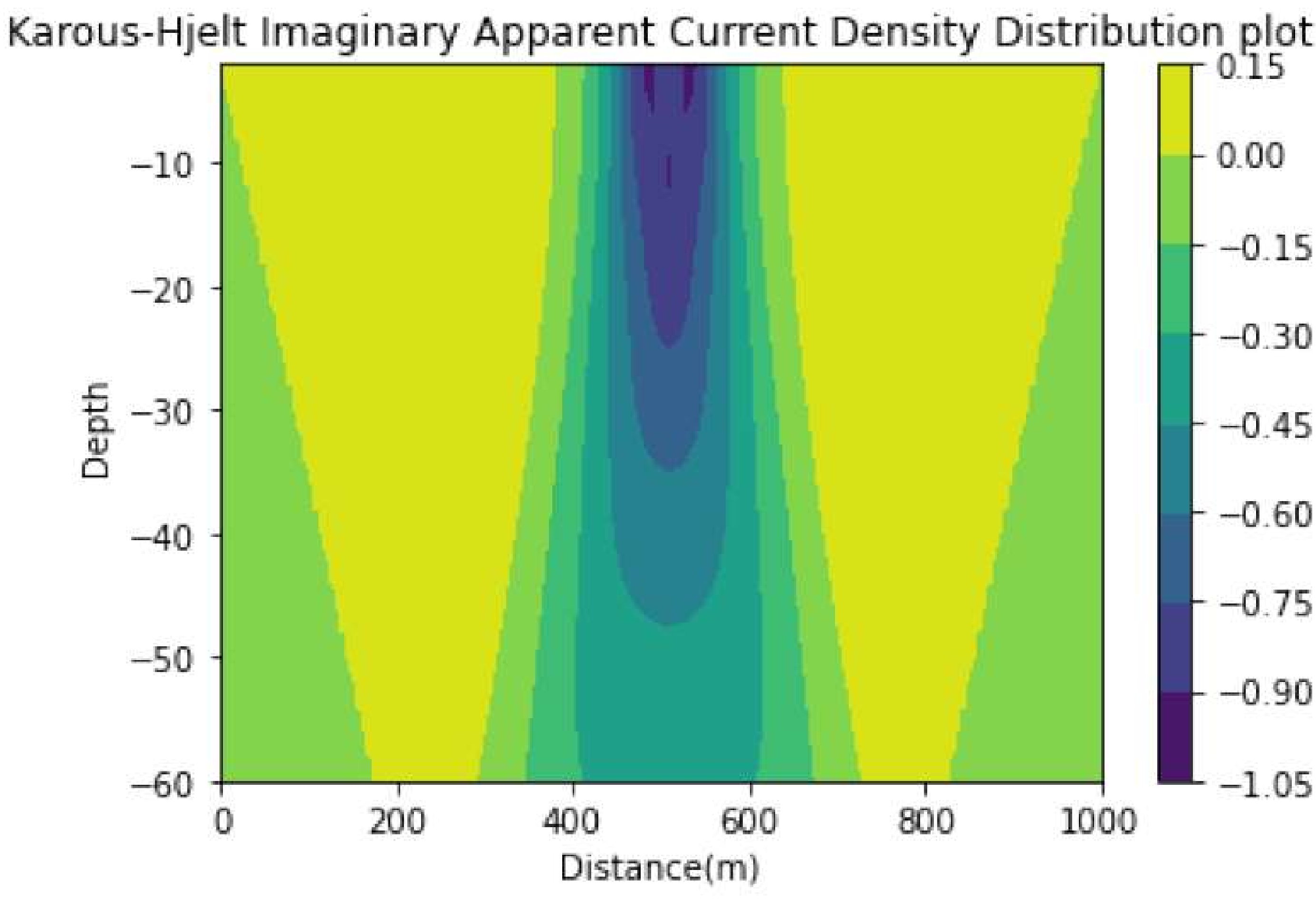
```
for i in range (30):
  plt.plot(xnew, R_kh[i])

plt.legend([str(2*(i+1))+" meters" for i in range(30)],
           title='Legend', bbox_to_anchor=(1.05, 1), loc='upper left')
plt.xlabel('Distance(m)')
plt.ylabel('Real Apparent Current Density')
```

```
plt.title("Karous-Hjelt Real Apparent Current Density Distribution
plot")
plt.show()
```



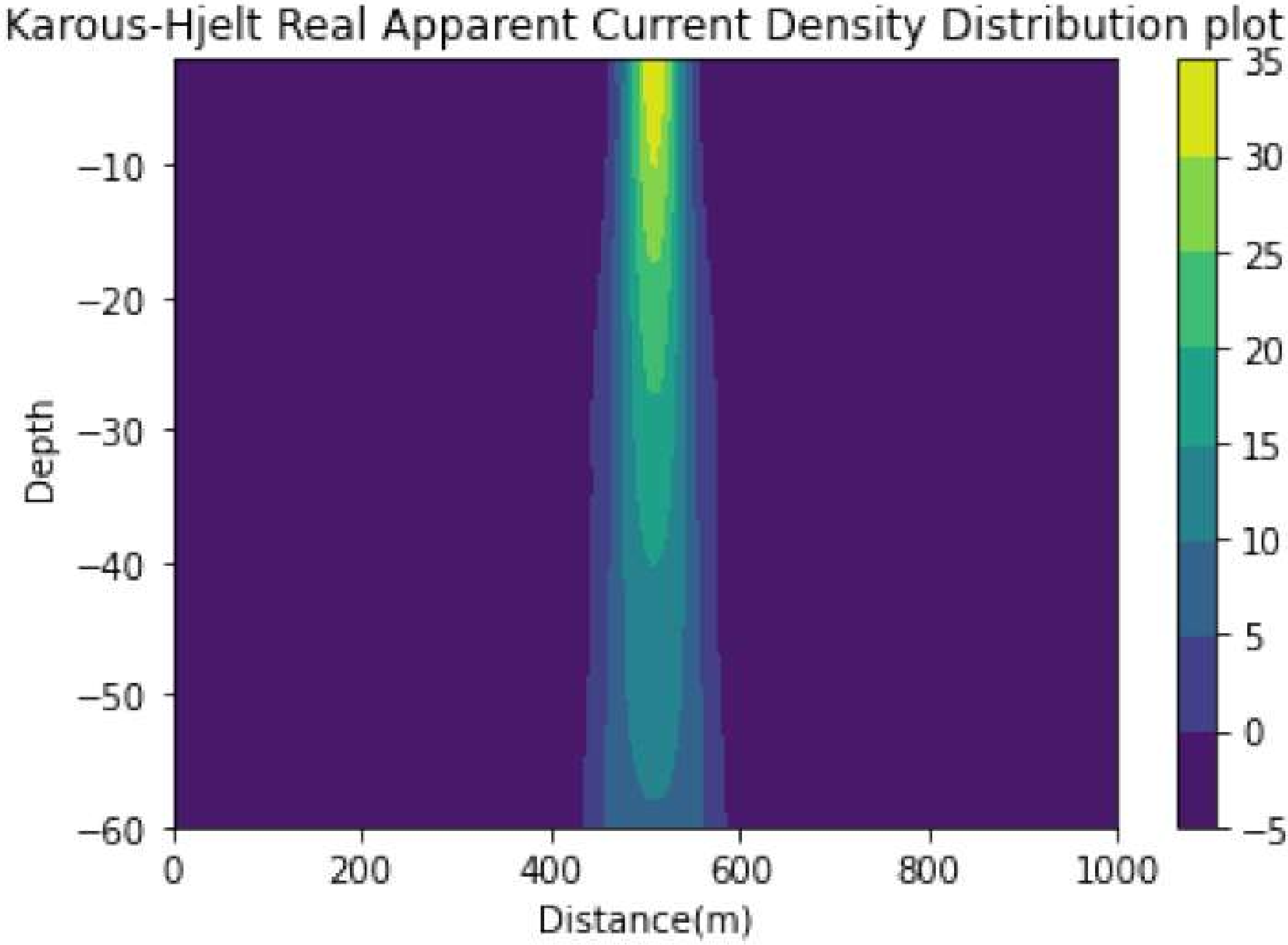
Karous-Hjelt Real Apparent Current Density Distribution plot
Real Apparent Current Density
30
25
20
15
10
5
0
−5
0
200
400
600
800
1000
Distance(m)
Legend
2 meters
4 meters
6 meters
8 meters
10 meters
12 meters
14 meters
16 meters
18 meters
20 meters
22 meters
24 meters
26 meters
28 meters
30 meters
32 meters
34 meters
36 meters
38 meters
40 meters
42 meters
44 meters
46 meters
48 meters
50 meters
52 meters
54 meters
56 meters
58 meters
60 meters

```
a = -np.array(np.arange(2,61,2))
b = np.array(xnew)
b,a = np.meshgrid(b, a)
plt.contourf(b, a,np.array(I_kh))
plt.colorbar()
plt.xlabel('Distance(m)')
plt.ylabel('Depth')
plt.title("Karous-Hjelt Imaginary Apparent Current Density
Distribution plot")
plt.show()
```



Karous-Hjelt Imaginary Apparent Current Density Distribution plot
Depth
−10
−20
−30
−40
−50
−60
0
200
400
600
800
1000
Distance(m)
0.15
0.00
−0.15
−0.30
−0.45
−0.60
−0.75
−0.90
−1.05

```
plt.contourf(b, a,np.array(R_kh))
plt.colorbar()
plt.xlabel('Distance(m)')
plt.ylabel('Depth')
plt.title("Karous-Hjelt Real Apparent Current Density Distribution
plot")
plt.show()
```


Karous-Hjelt Real Apparent Current Density Distribution plot
Depth
−10
−20
−30
−40
−50
−60
0
200
400
600
800
1000
Distance(m)
35
30
25
20
15
10
5
0
−5