

```

1 import numpy as np
2 import matplotlib.pyplot as plot
3 from math import exp
4 from numpy.linalg import inv
5
6 def nX(xval):
7     if abs(xval) >= 1:
8         return 0
9     return exp(1 / (abs(xval) ** 2 - 1))
10 X = []
11 F = []
12 for i in range(-50, 51):
13     Xi = [1]
14     F.append(nX(i/50))
15     for j in range(1, 11):
16         Xi.append((i/50) ** j)
17     X.append(Xi)
18
19 X = np.array(X)
20 F = np.array(F)
21
22 XTX = np.matmul(np.transpose(X), X)
23
24 XTF = np.matmul(np.transpose(X), F)
25
26 A = np.matmul(inv(XTX), XTF)
27
28 if __name__ == '__main__':
29     x = np.linspace(-1, 1, 101)
30     P10 = A[0] + A[1] * x ** 1 + A[2] * x ** 2 + A[3]
31     * x ** 3 + A[4] * x ** 4 + A[5] * x ** 5 + A[6] * x
32     ** 6 + A[7] * x ** 7 + A[8] * x ** 8 + A[9] * x ** 9
33     + A[10] * x ** 10
34     print(A)
35     vals = []
36     for i in range(-50, 51):
37         vals.append(nX(i / 50))
38     N_plot, = plot.plot(x, vals, color="black", label
39     = "$\eta(x)", alpha = 0.75)
40     P10_plot, = plot.plot(x, P10, ls = "dashed",
41     color = "red", label = "$P_{10,101}(x)$")
42
43 plot.xlabel("x")

```

```
40     plot.ylabel("y")
41     plot.title("10th Order Least Squares
Approximation for  $\eta(x)$  with 101 Fitting Points")
42     plot.legend(handles = [P10_plot, N_plot])
43     plot.show()
44
```

```

1 import numpy as np
2 import matplotlib.pyplot as plot
3 from math import exp
4 from numpy.linalg import inv
5 import scipy.special as sp
6 from code2pdf.code2pdf import main
7 ifile, ofile, size = "p10b.py", "p10b.pdf", "A4"
8
9 X = []
10 F = []
11 for i in range(0, 1000):
12     Xi = [1]
13     F.append(sp.jv(0, i/100))
14     for j in range(1, 9):
15         Xi.append((i/100) ** j)
16     X.append(Xi)
17
18 X = np.array(X)
19 F = np.array(F)
20
21 XTX = np.matmul(np.transpose(X), X)
22
23 XTF = np.matmul(np.transpose(X), F)
24
25 A = np.matmul(inv(XTX), XTF)
26
27 if __name__ == '__main__':
28     x = np.linspace(0, 10, 1001)
29     P10 = A[0] + A[1] * x ** 1 + A[2] * x ** 2 + A[3]
30     ] * x ** 3 + A[4] * x ** 4 + A[5] * x ** 5 + A[6] * x
31     ** 6 + A[7] * x ** 7 + A[8] * x ** 8
32     print(A)
33
34     J0_plot, = plot.plot(x, sp.jv(0, x), color="black
35     ", label="$J_0(x)", alpha = 0.75)
36     P8_plot, = plot.plot(x, P10, ls = "dashed", color
37     = "red", label = "$P_{8,1001}(x)$")
38
39     #plot.ylim(-0.5, 20)
40     plot.xlabel("x")
41     plot.ylabel("y")
42     plot.title("8th Order Least Squares Approximation
43     for $J_0(x)$ with 1001 Fitting Points")
44     plot.legend(handles = [P8_plot, J0_plot])

```

```
40     plot.show()  
41
```