

```

1  from numpy import linspace, zeros, exp, linalg, dot, eye
2  from matplotlib.pyplot import style, figure, axes
3
4  # Функция подготавливает матрицу СЛАУ
5  def system_matrix(N,x,h,N_s,s,l_y,l_z) :
6      A = zeros((3*N_s,N+1))
7      for j in range(0,3*N_s,3) :
8          A[j,0] = 1/2*(s[int(j/3)] - x[0])*h/((s[int(j/3)] - x[0])**2 + l_y**2 +
9              l_z**2)**3/2
10         for n in range(1,N) :
11             A[j,n] = (s[int(j/3)] - x[n])*h/((s[int(j/3)] - x[n])**2 + l_y**2 +
12                 l_z**2)**3/2
13             A[j,N] = 1/2*(s[int(j/3)] - x[N])*h/((s[int(j/3)] - x[N])**2 + l_y**2 +
14                 l_z**2)**3/2
15         for j in range(1,3*N_s,3) :
16             A[j,0] = 1/2*l_y*h/((s[int((j-1)/3)] - x[0])**2 + l_y**2 + l_z**2)**3/2
17             for n in range(1,N) :
18                 A[j,n] = l_y*h/((s[int((j-1)/3)] - x[n])**2 + l_y**2 + l_z**2)**3/2
19             A[j,N] = 1/2*l_y*h/((s[int((j-1)/3)] - x[N])**2 + l_y**2 + l_z**2)**3/2
20         for j in range(2,3*N_s,3) :
21             A[j,0] = 1/2*l_z*h/((s[int((j-2)/3)] - x[0])**2 + l_y**2 + l_z**2)**3/2
22             for n in range(1,N) :
23                 A[j,n] = l_z*h/((s[int((j-2)/3)] - x[n])**2 + l_y**2 + l_z**2)**3/2
24             A[j,N] = 1/2*l_z*h/((s[int((j-2)/3)] - x[N])**2 + l_y**2 + l_z**2)**3/2
25         return A
26
27 def rho_model(x) :
28     return 2*exp(-(x - 0.382)**2/0.009) + 1.2*exp(-(x - 0.618)**2/0.018)
29
30 a = 0.; b = 1.
31
32 N_model = 100
33
34 h_model = (b - a)/N_model
35 x_model = linspace(a,b,N_model+1)
36
37 X_model = zeros(N_model+1)
38 for n in range(N_model+1) :
39     X_model[n] = rho_model(x_model[n])
40
41 c = -1.; d = 2.
42 l_y = 0.8 ; l_z = 0.2
43
44 N_s = 200
45 x_s = linspace(c,d,N_s)
46
47 A = system_matrix(N_model,x_model,h_model,N_s,x_s,l_y,l_z)
48
49 B = dot(A,X_model)
50
51 N = 20
52 h = (b - a)/N
53 x = linspace(a,b,N+1)
54
55 A = system_matrix(N,x,h,N_s,x_s,l_y,l_z)
56
57 alpha = 10**(-13)
58 X_inv = linalg.solve(dot(A.T,A) + alpha*eye(N+1),dot(A.T,B))
59
60 # Отрисовка решения
61 style.use('dark_background')
62
63 fig = figure()
64 ax = axes(xlim=(a,b), ylim=(-0.5,2.5))
65 ax.set_xlabel('x'); ax.set_ylabel('rho')
66 ax.plot(x_model,X_model,'-g',lw=2)
67 ax.plot(x,X_inv,'-or',lw=2)
68
69 # Листинг программы, реализующей приближённое решение

```

```
67 # интегрального уравнения Фредгольма 1-го рода
68 # на примере решения обратной задачи электростатики
69 # (с регуляризацией)
```