

```

1  from numpy import linspace, zeros, log, exp
2  from matplotlib.pyplot import plot, xscale, yscale
3
4  def u(x) :
5      return 1/(1 + x**2)
6
7  def Integration(u,a,b,N) :
8      h = (b - a)/N
9      x = linspace(a,b,N+1)
10     integral = 0.
11     for n in range(1,N+1) :
12         integral = integral + (u(x[n-1]) + u(x[n]))/2*h
13     return integral
14
15     a = -1.; b = 1.
16
17     N = 1
18     r = 2; S = 3
19     p = 2; q = 2
20
21     U = zeros((S,S))
22     R = zeros((S,S))
23     p_eff = zeros((S,S))
24
25     for s in range(S) :
26         U[s,0] = Integration(u,a,b,r**s*N)
27
28     for s in range(1,S) :
29         for l in range(s) :
30             R[s,l] = (U[s,l] - U[s-1,l])/(r**(p + l*q) - 1)
31             U[s,l+1] = U[s,l] + R[s,l]
32
33     for s in range(2,S) :
34         for l in range(s-1) :
35             p_eff[s,l] = log(abs(R[s-1,l]/R[s,l]))/log(r)
36
37     # Функция выводит форматированную таблицу
38     def PrintTriangular(A,i) :
39         print(' ',end=' ')
40         for l in range(len(A)) :
41             print(' p={0:<4d}'.format(p + l*q),end=' ')
42         print()
43         for m in range(len(A)) :
44             print(' s={0:<2d}'.format(m),end=' ')
45             for l in range(m + 1 - i) :
46                 print('{0:7.4f}'.format(A[m,l]),end=' ')
47             print()
48         print()
49
50     print('Таблица приближённых значений интеграла:')
51     PrintTriangular(U,0)
52     print('Таблица оценок ошибок:')
53     PrintTriangular(R,1)
54     print('Таблица эффективных порядков точности:')
55     PrintTriangular(p_eff,2)
56
57     plot([r**s*N for s in range(1,S)],abs(R[1:,0]),'-bo')
58     xscale('log'); yscale('log')
59
60     # Листинг программы, реализующей приближённое вычисление интеграла
61     # с помощью рекуррентного сгущения сеток и многократного повышения
62     # точности по Ричардсону (с вычислением эффективных порядков точности)

```