

```

1  from numpy import linspace, zeros, log, exp
2  from matplotlib.pyplot import plot, xscale, yscale
3
4  def u(x) :
5      return exp(x)
6
7  def Differentiation(u,x,h):
8      dif = (u(x) - u(x - h))/h
9      return dif
10
11  x = 0.
12
13  h = 0.1
14  r = 2; S = 5
15  p = 1; q = 1
16
17  D = zeros((S,S))
18  R = zeros((S,S))
19  p_eff = zeros((S,S))
20
21  for s in range(S) :
22      D[s,0] = Differentiation(u,x,h/r**s)
23
24  for s in range(1,S) :
25      for l in range(s) :
26          R[s,l] = (D[s,l] - D[s-1,l])/(r**(p + l*q) - 1)
27          D[s,l+1] = D[s,l] + R[s,l]
28
29  for s in range(2,S) :
30      for l in range(s-1) :
31          p_eff[s,l] = log(abs(R[s-1,l]/R[s,l]))/log(r)
32
33  # Функция выводит форматированную таблицу
34  def PrintTriangular(A,i) :
35      print(' ',end=' ')
36      for l in range(len(A)) :
37          print(' p={0:<4d}'.format(p + l*q),end=' ')
38      print()
39      for m in range(len(A)) :
40          print(' s={0:<2d}'.format(m),end=' ')
41          for l in range(m + 1 - i) :
42              print('{0:7.4f}'.format(A[m,l]),end=' ')
43          print()
44      print()
45
46  print('Таблица приближённых значений интеграла:')
47  PrintTriangular(D,0)
48  print('Таблица оценок ошибок:')
49  PrintTriangular(R,1)
50  print('Таблица эффективных порядков точности:')
51  PrintTriangular(p_eff,2)
52
53  plot([r**s*(1- 0)/h for s in range(1,S)],abs(R[1:,0]),'-bo')
54  xscale('log'); yscale('log')
55
56  # Листинг программы, реализующей приближённое вычисление производной
57  # с помощью рекуррентного сгущения сеток и многократного повышения
58  # точности по Ричардсону (с вычислением эффективных порядков точности)

```