

```

1  from numpy import *
2  from matplotlib.pyplot import *
3
4  # Функция f возвращает значение правой части решаемого ОДУ
5  def f(u,lambda):
6      f = lambda*u
7      return f
8
9  # Определение входных данных задачи
10 t_0 = 0.; T = 1.; lambda = -10.; u_0 = 1.
11
12 # Определение числа интервалов сетки,
13 # на которой будет искомое приближённое решение
14 M = 20
15
16 # Определение сетки
17 tau = (T - t_0)/M
18 t = linspace(t_0,T,M + 1)
19
20 # Выделение памяти под массив сеточных значений решения ОДУ
21 u = zeros(M + 1)
22
23 # Задание начального условия
24 u[0] = u_0
25
26 # Реализация схемы Эйлера
27 for m in range(M):
28     u[m + 1] = u[m] + tau*f(u[m],lambda)
29
30 # Отрисовка решения
31 figure()
32 # Приближённое решение
33 plot(t,u,'-ro',markersize=5,label='приближённое решение')
34 # Точное решение
35 plot(t,u_0*exp(lambda*(t-t_0)),'-g',label='точное решение')
36 title('График функции u(t), найденной по схеме ERK1')
37 xlabel('t'); ylabel('u')
38 xlim((t_0,T)); ylim((0,u_0))
39 legend()

```

Комментарий к файлу:

Листинг программы, реализующей решение задачи Коши для ОДУ, возникающего при моделировании процесса радиоактивного распада. Для простоты реализована схема Эйлера.