```python
#! python3.7
# -*- coding: utf-8 -*-
from numpy import zeros, linspace, linalg, eye
from matplotlib.pyplot import style, figure, axes

def f(y,h,N,eps,u_left,u_right) :
    f = zeros(N-1)
    f[0] = eps*(y[1] - 2*y[0] + u_left)/h**2 - y[0]*(y[1] - u_left)/(2*h) - y[0]
    for n in range(1,N-2) :
        f[n] = eps*(y[n+1] - 2*y[n] + y[n-1])/h**2 - y[n]*(y[n+1] - y[n-1])/(2*h) - y[n]
    f[N-2] = eps*(u_right - 2*y[N-2] + y[N-3])/h**2 - y[N-2]*(u_right - y[N-3])/(2*h) - y[N-2]
    return f

def f_y(y,h,N,eps,u_left,u_right) :
    f_y = zeros((N-1,N-1))
    f_y[0,0] = eps*(-2/h**2) - (y[1] - u_left)/(2*h) - 1.
    for n in range(1,N-1) :
        f_y[n,n-1] = eps/h**2 + y[n]/(2*h)
    for n in range(1,N-2) :
        f_y[n,n] = eps*(-2)/h**2 - (y[n+1] - y[n-1])/(2*h) - 1.
    for n in range(N-2) :
        f_y[n,n+1] = eps/h**2 - y[n]/(2*h)
    f_y[N-2,N-2] = eps*(-2/h**2) - (u_right - y[N - 3])/(2*h) - 1.
    return f_y

# Определение входных данных задачи
a = 0.; b = 1.; eps = 0.1
u_left = 4.; u_right = -3.5

# Число итераций метода Ньютона
S = 10

# Определение числа интервалов сетки,
# на которой будет искаться приближённое решение
N = 50

# Определение сетки
h = (b - a)/N
x = linspace(a,b,N+1)

# Выделение памяти под массивы сеточных значений приближённых решений краевой задачи
# и соответствующей нелинейной системы на каждой итерации метода Ньютона.
# В строке с номером s хранятся сеточные значения соответствующего приближения на s-ой итерации
u = zeros((S+1,N+1)); y = zeros((S+1,N-1))

# Начальное приближение является нулевым вектором
y[0] = u[0,1:N]

for s in range(S) :
    # Реализация итерации метода Ньютона
    y[s+1] = y[s] - linalg.solve(f_y(y[s],h,N,eps,u_left,u_right),f(y[s],h,N,eps,u_left,u_right))
    # Заполнение массива сеточных значений решения
    # очередного приближения решения краевой задачи
    u[s+1,0] = u_left
    u[s+1,1:N] = y[s+1,:]
    u[s+1,N] = u_right

# Отрисовка решения
style.use('dark_background')

fig1 = figure()
ax1 = axes(xlim=(a,b), ylim=(-6,8.))
ax1.set_xlabel('x'); ax1.set_ylabel('y')
ax1.plot(x,u[S],'-ow',markersize=5)

```

```
66  # Листинг программы, реализущей приближённое решение краевой задачи
67  # для ОДУ второго порядка с помощью сеточного метода
```