Постановка задачи:

Задание: Решить указанную в варианте задачу:

1. С помощью метода конечных разностей.

2. С помощью метода Галеркина с 10-ю алгебраическими базисными функциями (т.е. многочленами).

Вывести графики полученных решений и сравнить решения.

Вариант 1. $$\begin{aligned}&(2xu'(x))' - u(x) = 6x^2 + x + 2, \quad 0 \le x \le 1,\\ &u'(0) = 1, \quad u'(1) + u(1) = 10.\end{aligned}$$

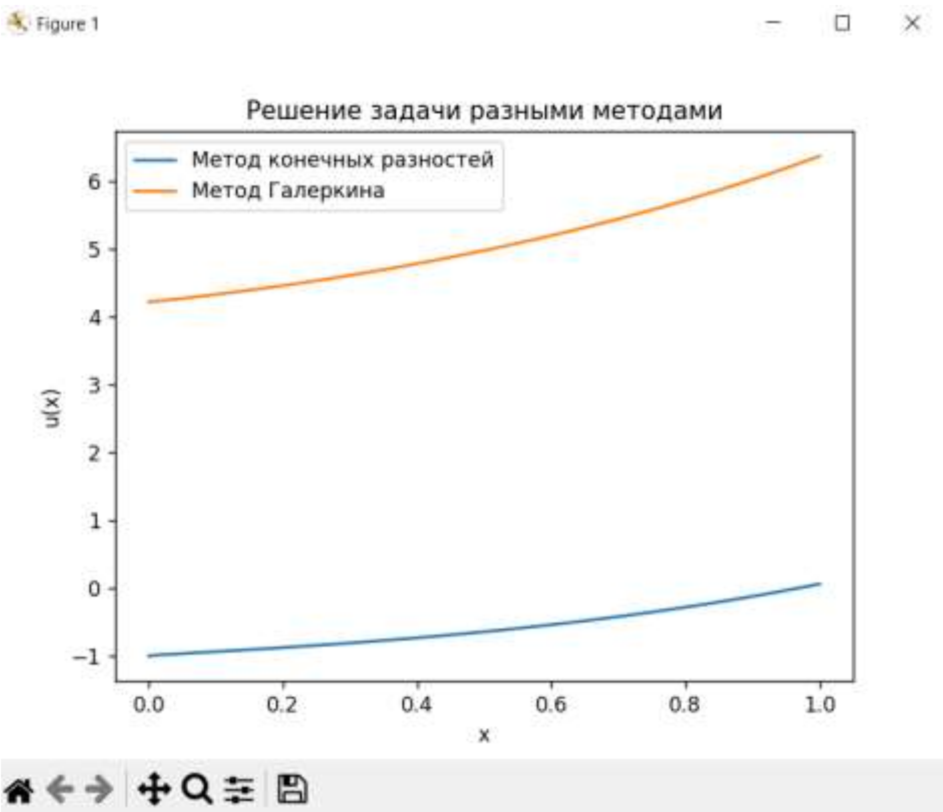
Программная реализация:

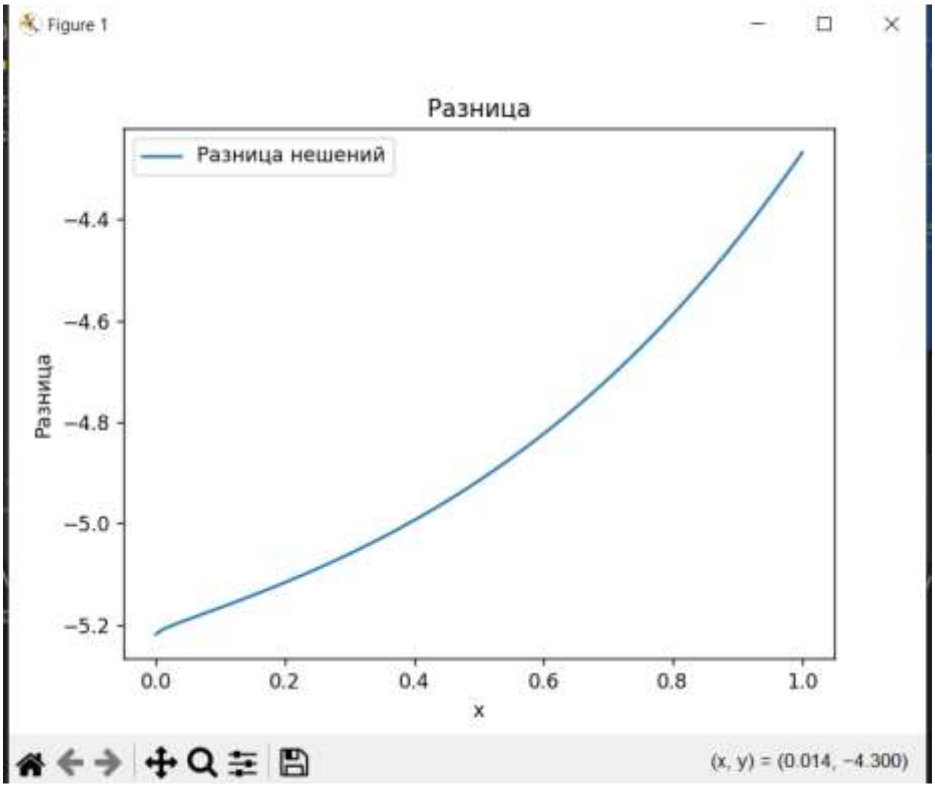
```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import quad
from scipy.interpolate import interp1d


# Метод конечных разностей
def finite_difference_method(N):
    x = np.linspace(0, 1, N + 1)
    h = 1.0 / N

    A = np.zeros((N + 1, N + 1))
    b = np.zeros(N + 1)

    # Внутренние узлы
    for i in range(1, N):
        xi = x[i]
        A[i, i - 1] = ((2 * xi) / (h ** 2)) - (1/h)
        A[i, i] = ((-4 * xi) / (h ** 2)) - 1
        A[i, i + 1] = ((2 * xi) / (h ** 2)) + (1/h)
        b[i] = 6 * (xi ** 2) + xi + 2

    # Граничные условия
    A[0, 0] = -1 / h
    A[0, 1] = 1 / h
    b[0] = 1

    A[N, N - 1] = 1 / (2*h)
    A[N, N] = 3 / (2*h)
    b[N] = 10

    # Решение системы
    u = np.linalg.solve(A, b)
```

```python
    return x, u


# Метод Галеркина
def phi(i, x):
    if i==0:
        return x+8
    else:
        return -i-2+(x**(i+1))
    # return x**i

def dphi(i, x):
    if i == 0:
        return 1
    else:
        return (i+1)*(x**(i))
    # if i==0:
    #     return 0
    # else:
    #     return i * x**(i-1)

def dphi2(i, x):
    if i == 0:
        return 0
    else:
        return i*(i+1)*(x**(i-1))
    # return i * (i-1) * x**(i-2)

def first(i,j):
    integrative_first = lambda x: dphi(j,x)*phi(i,x)
    return quad(integrative_first, 0,1)[0]

def second(i,j):
    integrative_second = lambda x: dphi2(j,x)*phi(i,x)
    return quad(integrative_second, 0, 1)[0]

def third(i,j):
    integrative_third = lambda x: phi(j,x)*phi(i,x)
    return quad(integrative_third, 0, 1)[0]

def fourth(i):
    integrative_fourth = lambda x: ((6*(x**2)) + x + 2)*phi(i,x)
    return quad(integrative_fourth, 0,1)[0]

def galerkin_method(n_basis):
    A = np.zeros((n_basis, n_basis))
    B = np.zeros(n_basis)

    for i in range(n_basis):
        for j in range(n_basis):
            A[i, j] = 2*first(i,j) + 2*second(i,j) - third(i,j)
        B[i] = fourth(i)
```

```
    # Граничные условия
    A[0, :] = [dphi(i, 0) for i in range(0, n_basis)]
    B[0] = 1

    A[-1, :] += [dphi(i, 1) + phi(i, 1) for i in range(0,
n_basis)]
    B[-1] += 10

    c = np.linalg.solve(A, B)

    def u(x):
        return sum(c[i] * phi(i, x) for i in range(0, n_basis))

    x_vals = np.linspace(0, 1, 1000)
    u_vals = [u(xi) for xi in x_vals]
    return x_vals, u_vals


# Параметры
N = 100
M = 10

# Решение методами
x_finite_difference, solution_finite_difference =
finite_difference_method(N)
x_galerkin, solution_galerkin = galerkin_method(M)


res_diff = solution_finite_difference -
solution_galerkin[:len(solution_finite_difference)]
norm_diff = np.linalg.norm(res_diff)

# Графики
plt.plot(x_finite_difference, solution_finite_difference,
label='Метод конечных разностей')
plt.plot(x_galerkin, solution_galerkin, label='Метод Галеркина')
plt.xlabel('x')
plt.ylabel('u(x)')
plt.legend()
plt.title('Решение задачи разными методами')
plt.show()

print(f'Норма разности решений: {norm_diff:.6f}')
plt.plot(x_finite_difference, res_diff, label='Разница нешений')
plt.xlabel('x')
plt.ylabel('Разница')
plt.legend()
plt.title('Разница')
plt.show()
```

Полученные результаты:






```
C:\MyProgramming\For-BSU\BSU-SixthTerm
Норма разности решений: 48.873742
```

Значение нормы разности решений между методом конечных разностей и методом Галеркина составляет приблизительно 48.87. Это означает, что решения, полученные этими двумя методами, достаточно различны на всем интервале.

Вероятно, это связано с тем, что метод конечных разностей и метод Галеркина являются разными численными методами, и каждый из них имеет свои ограничения и приближения при решении дифференциальных уравнений. Различия в результатах могут быть вызваны различными аппроксимациями и численными алгоритмами, используемыми в этих методах.