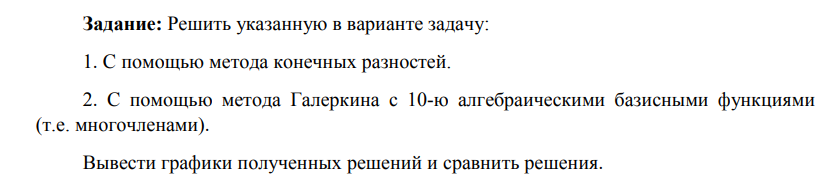
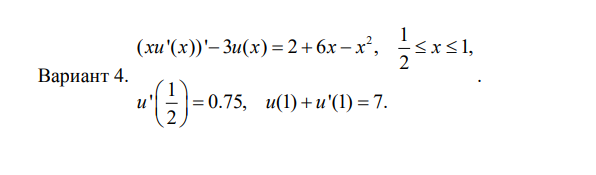
**Лабораторная работа № 4 Граничная задача для ОДУ**

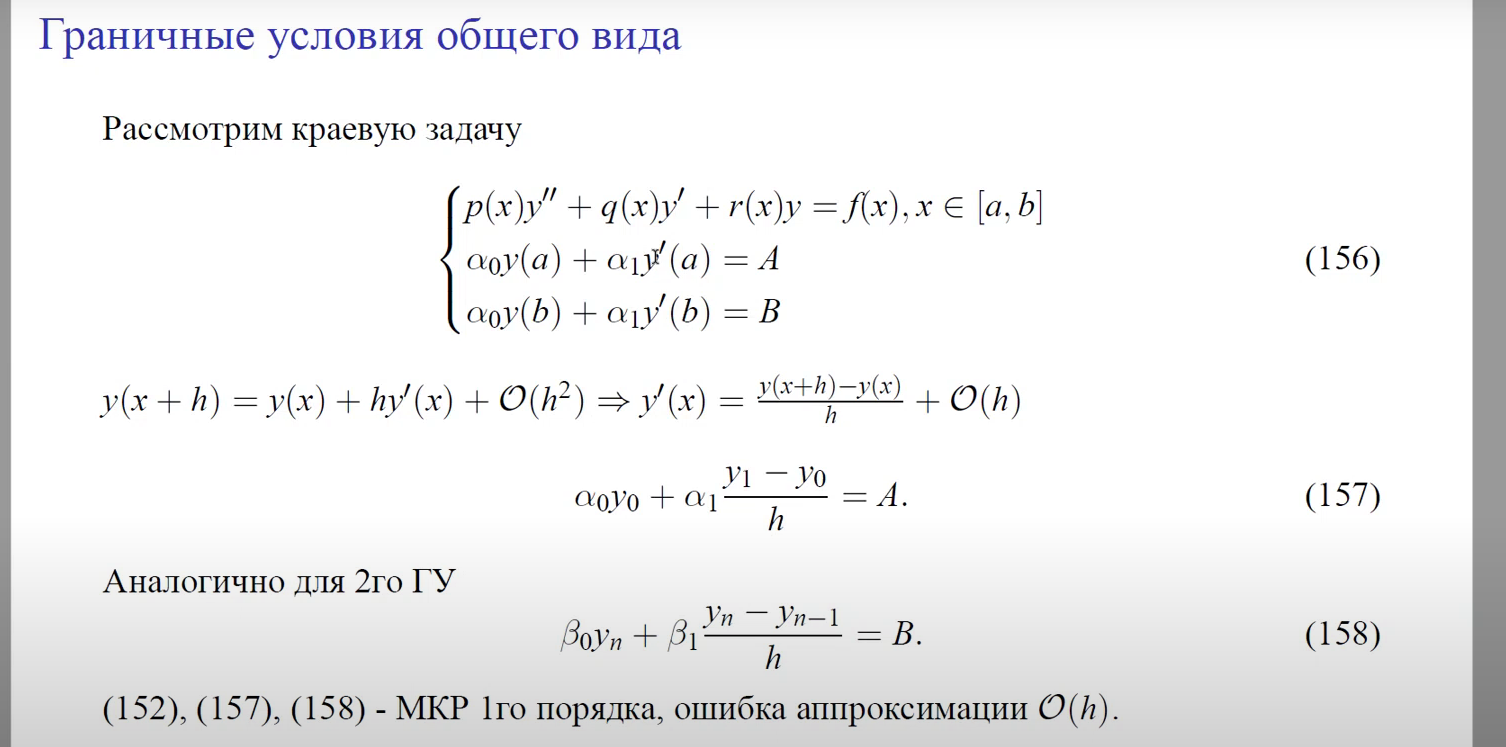
Выполнил студент 3 курса 4 группы ФПМИ БГУ Видевич Александр.

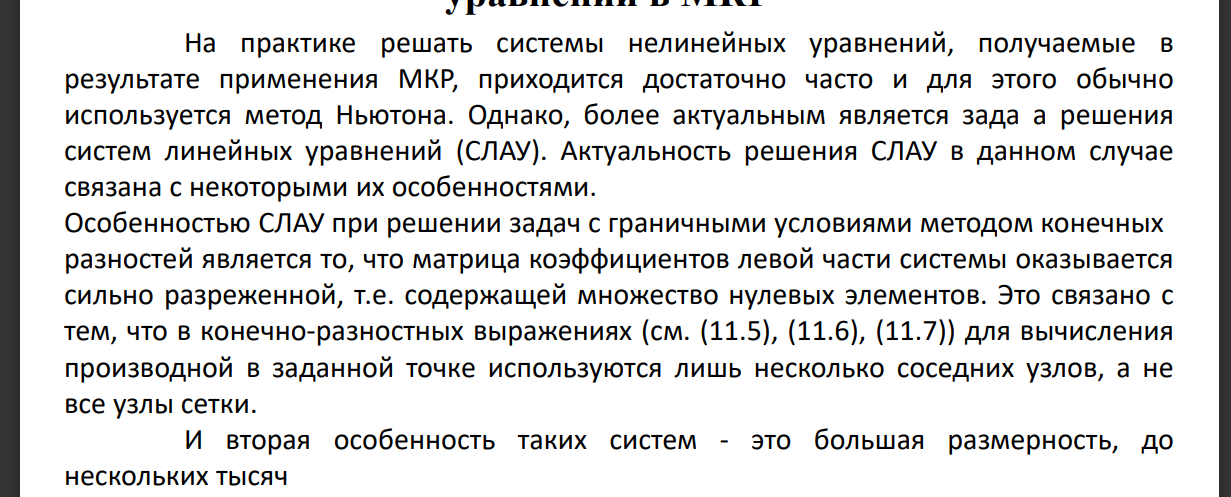
**Задача.**

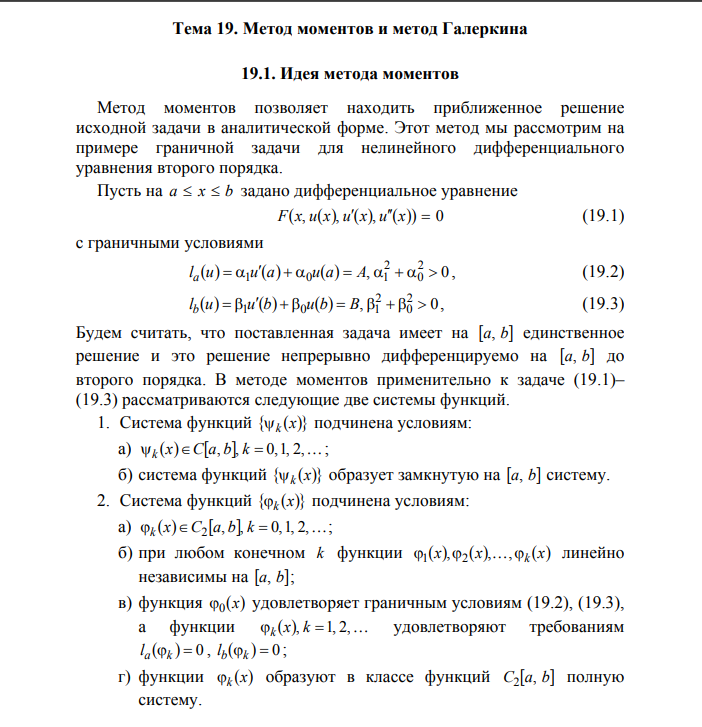




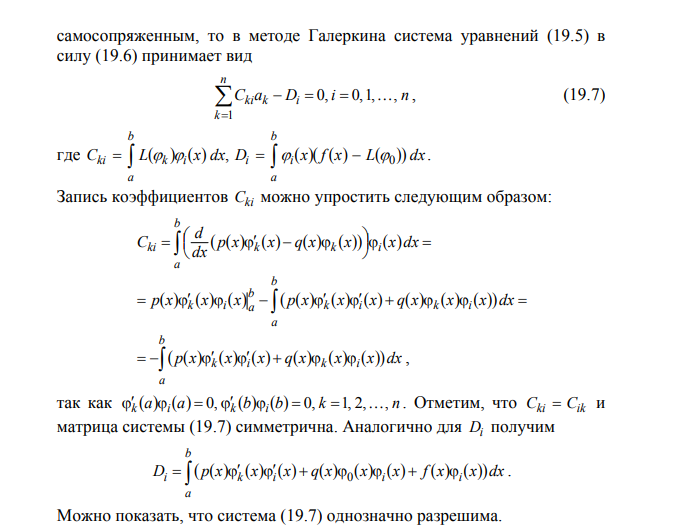
**Использованная теория.**

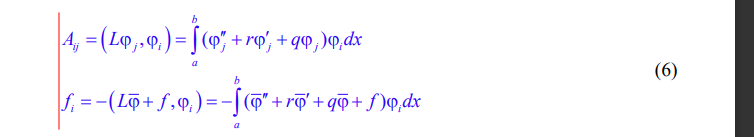
****

****

****

****

****

****

**Листинг программы**

data.py

x\_lower = *1*/*2*

x\_upper = *1*

N = *100*

finite\_difference\_method.py

*import* numpy *as* np

*import* data

*# x[i] \* (u[i+1] - 2u[i] + u[i-1]) / h^2 + (u[i+1] - u[i-1]) / (2h) - 3 \* u[i] ≈ 2 + 6 \* x[i] - x[i]^2 = f[i]*

*# x[i]/h^2 + 1/2h*

*# -2 \* x[i] / h\*\*2 -3*

*# x[i]/h^2 - 1/2h*

def finite\_difference\_method():

    h = (data.x\_upper - data.x\_lower) / (data.N - *1*)

    x = np.linspace(data.x\_lower, data.x\_upper, data.N)

    A = np.zeros((data.N, data.N))

    A[*0*][*0*] = -*1*/h

    A[*0*][*1*] = *1*/h

    A[-*1*][-*1*] = (h + *1*) / h

    A[-*1*][-*2*] = -*1* / h

    b = np.zeros(data.N)

    b[*0*] = *0.75*

    b[-*1*] = *7*

*for* i *in* *range*(*1*, data.N - *1*):

        b[i] = *2* + *6* \* x[i] - x[i]\*\**2*

        A[i][i+*1*] = x[i] / h\*\**2* + *1* / (*2* \* h)

        A[i][i] = -*2* \* x[i] / h\*\**2* - *3*

        A[i][i-*1*] = x[i]/h\*\**2* - *1*/(*2*\*h)

    y = np.linalg.solve(A, b)

*return* x, y

galerkin\_method.py

*import* numpy *as* np

*import* matplotlib.pyplot *as* plt

*import* scipy.integrate *as* spi

*import* data

def fi(k, x):

*if* k == *0*:

*return* *0.75* \* x + *5.5*

*else*:

*return* -(data.x\_upper-data.x\_lower)\*\*(k+*1*)-(k+*1*)\*(data.x\_upper-data.x\_lower)\*\*k+(x-data.x\_lower)\*\*(k+*1*)

def d\_fi(k, x):

    h = *0.001*

*return* (fi(k, x+h)-fi(k, x-h))/(*2*\*h)

def d2\_fi(k, x):

    h = *0.001*

*return* (d\_fi(k, x+h)-d\_fi(k, x-h))/(*2*\*h)

def r(x):

*return* *1*/x

def q(x):

*return* -*3*/x

def integrate(f):

*return* spi.quad(f, data.x\_lower, data.x\_upper)[*0*]

def get\_func\_to\_integrate\_fi(i):

    def func(x):

*return* (d2\_fi(*0*, x)+r(x)\*d\_fi(*0*, x)+q(x)\*fi(*0*, x) - *2*/x - *6* + x)\*fi(i, x)

*return* func

def f(i):

*return* -*1* \* integrate(get\_func\_to\_integrate\_fi(i))

def get\_func\_to\_integrate\_Aij(i, j):

    def func(x):

*return* (d2\_fi(j, x)+r(x)\*d\_fi(j, x)+q(x)\*fi(j, x))\*fi(i, x)

*return* func

def galerkin\_method():

    n = *10*

    G = []

*for* i *in* *range*(*1*, n):

        tmp = []

*for* j *in* *range*(*1*, n):

            tmp.append(integrate(get\_func\_to\_integrate\_Aij(i, j)))

        G.append(tmp[:])

    K = []

*for* i *in* *range*(*1*, n):

        K.append(f(i))

    G = np.asarray(G, dtype=np.float64)

    K = np.asarray(K, dtype=np.float64)

    coeff = np.linalg.solve(G, K)

    def u(x):

        sum = fi(*0*, x)

*for* i *in* *range*(n-*1*):

            sum += coeff[i]\*fi(i+*1*, x)

*return* sum

    x = np.linspace(data.x\_lower, data.x\_upper, data.N)

    y = *map*(lambda x: u(x), x)

*return* x, *list*(y)

main.py

*import* matplotlib.pyplot *as* plt

*from* finite\_difference\_method *import* finite\_difference\_method

*from* galerkin\_method *import* galerkin\_method

*import* numpy *as* np

x1, u1 = finite\_difference\_method()

x2, u2 = galerkin\_method()

print(np.linalg.norm(np.array(u1) - np.array(u2)))

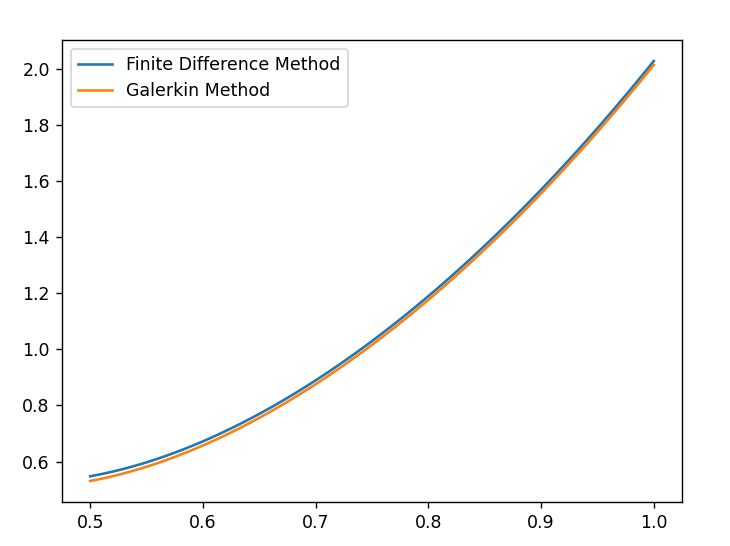
plt.plot(x1, u1, label='Finite Difference Method')

plt.plot(x2, u2, label='Galerkin Method')

plt.legend()

plt.show()

**Выходные данные.**

****

