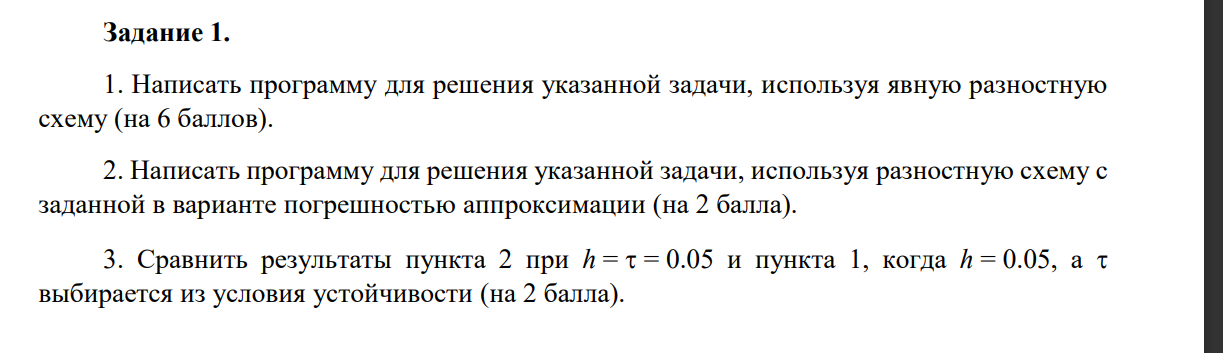
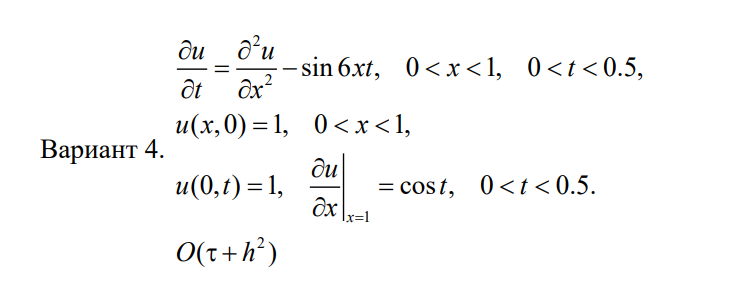
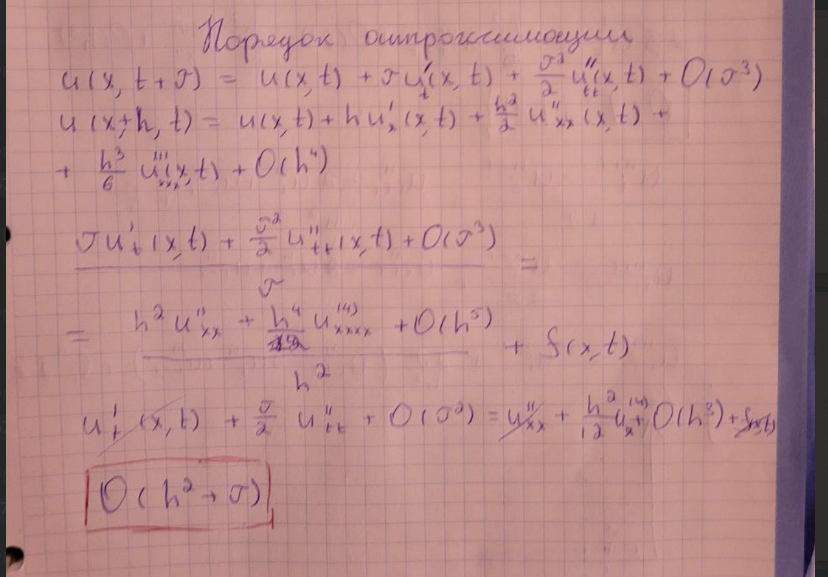
**Лабораторная работа № 5 Метод конечных разностей**

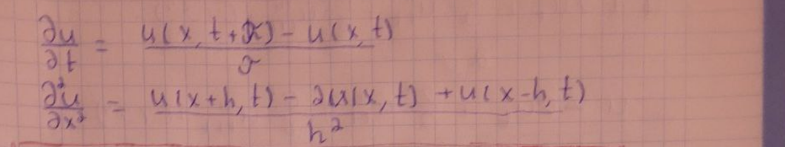
Выполнил студент 3 курса 4 группы ФПМИ БГУ Видевич Александр.

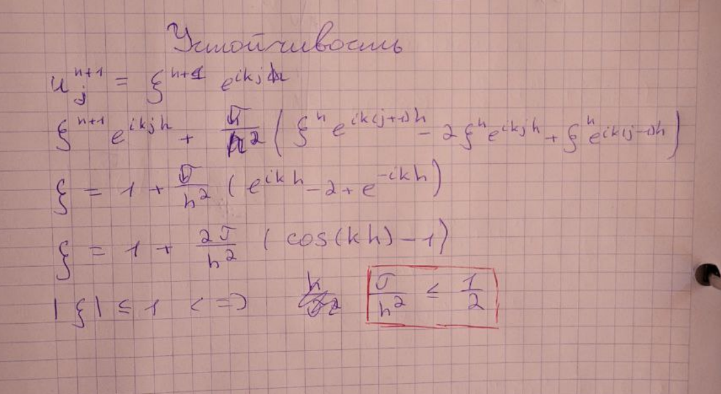
**Задача 1** ****

****

**Использованная теория.**







**Листинг программы**

*import* numpy *as* np

*import* matplotlib.pyplot *as* plt

*from* mpl\_toolkits.mplot3d *import* *Axes3D*

b = *1.0*

bt = *0.5*

N = *20*

M = *400*

h = b / N

r = bt / M

x = np.linspace(*0*, b, N+*1*)

t = np.linspace(*0*, bt, M+*1*)

u = np.zeros((N+*1*, M+*1*))

def f(x, t):

*return* -np.sin(*6*\*x\*t)

u[:, *0*] = np.ones(N+*1*)

u[*0*, :] = np.ones(M+*1*)

def g1(t):

*return* *1*

def g2(t):

*return* np.cos(t)

*for* n *in* *range*(*0*, M):

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

        u[i, n+*1*] = u[i, n] + r / h\*\**2* \* \

            (u[i+*1*, n] - *2*\*u[i, n] + u[i-*1*, n]) + r \* f(x[i], t[n])

    u[N, n+*1*] = u[N-*1*, n+*1*] + h \* g2(t[n+*1*])

X, T = np.meshgrid(x, t)

U = u.T

fig = plt.figure(figsize=(*10*, *7*))

ax = fig.add\_subplot(*111*, projection='3d')

ax.plot\_surface(X, T, U, cmap='viridis')

ax.set\_xlabel('X (пространственная координата)')

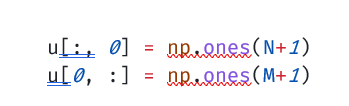
ax.set\_ylabel('T (время)')

ax.set\_zlabel('Температура u(x,t)')

ax.set\_title('Распределение температуры в стержне')

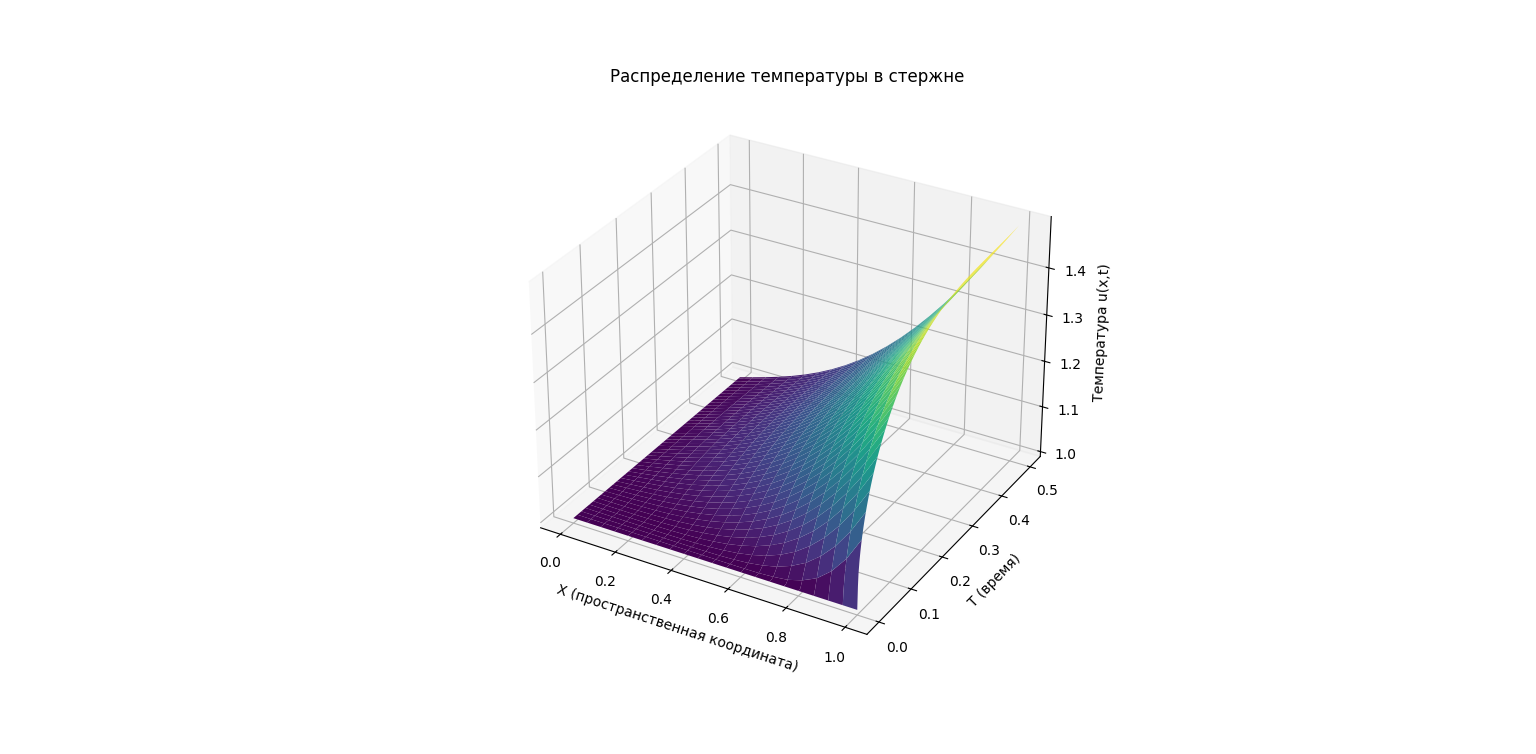
plt.show()

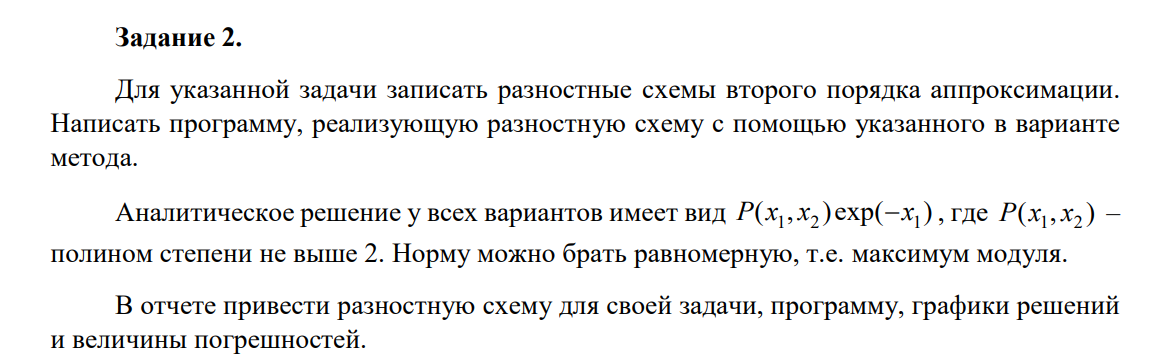
**Разностная схема**

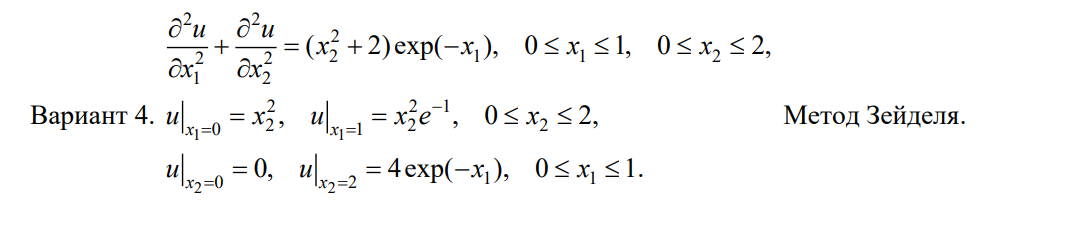




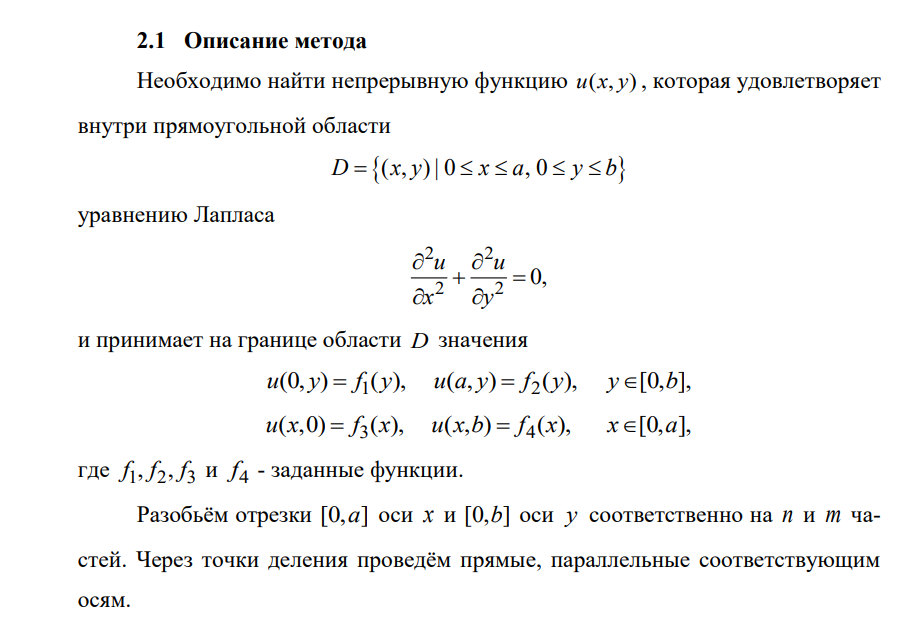
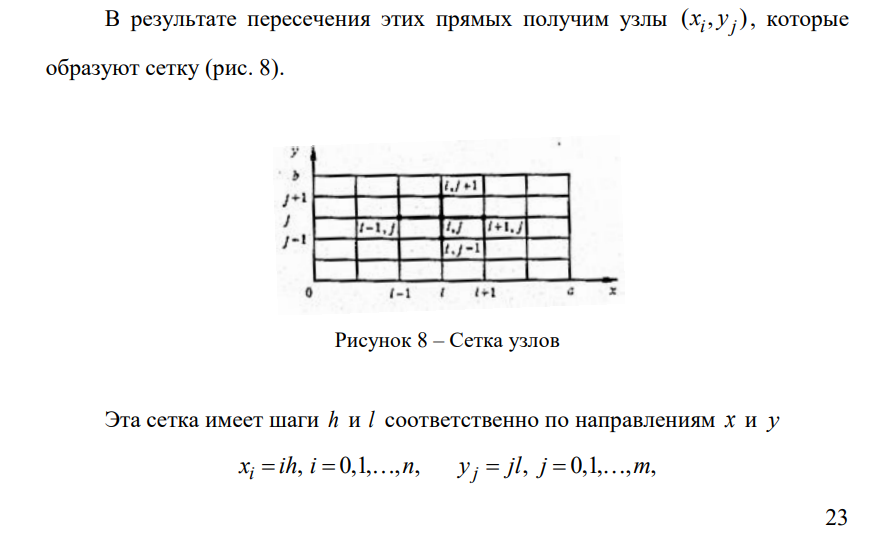
**Выходные данные.**

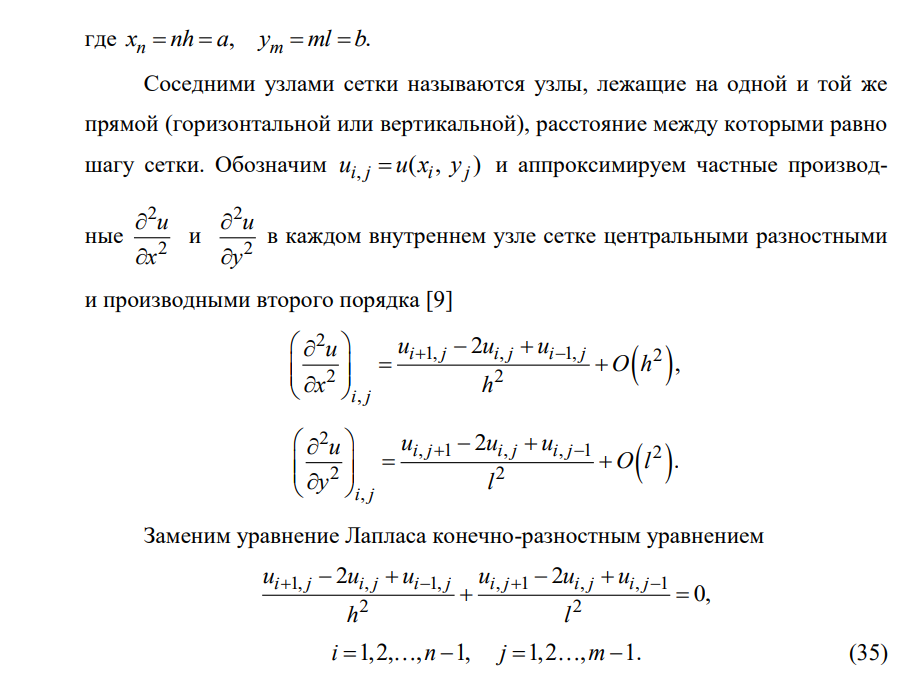


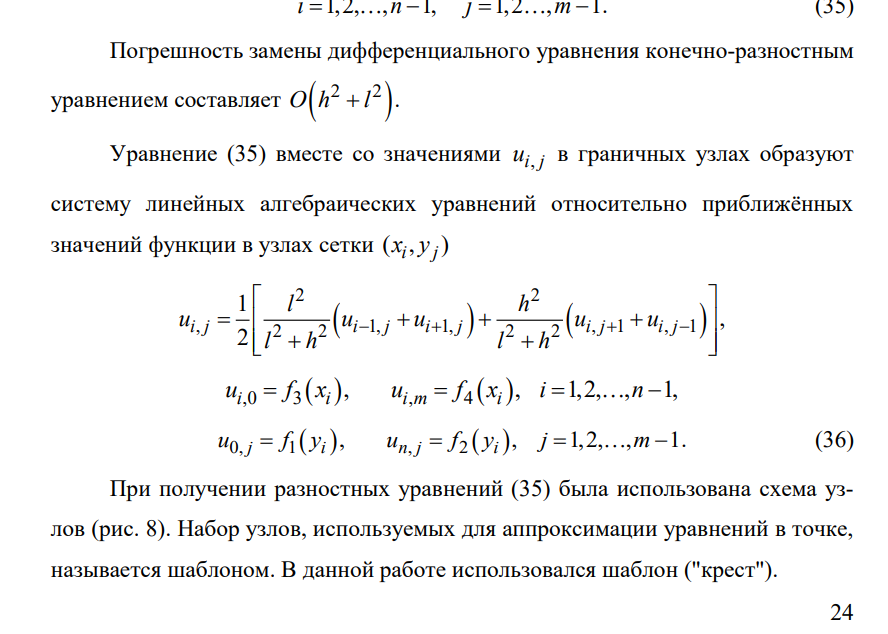
**Задача 2** ****

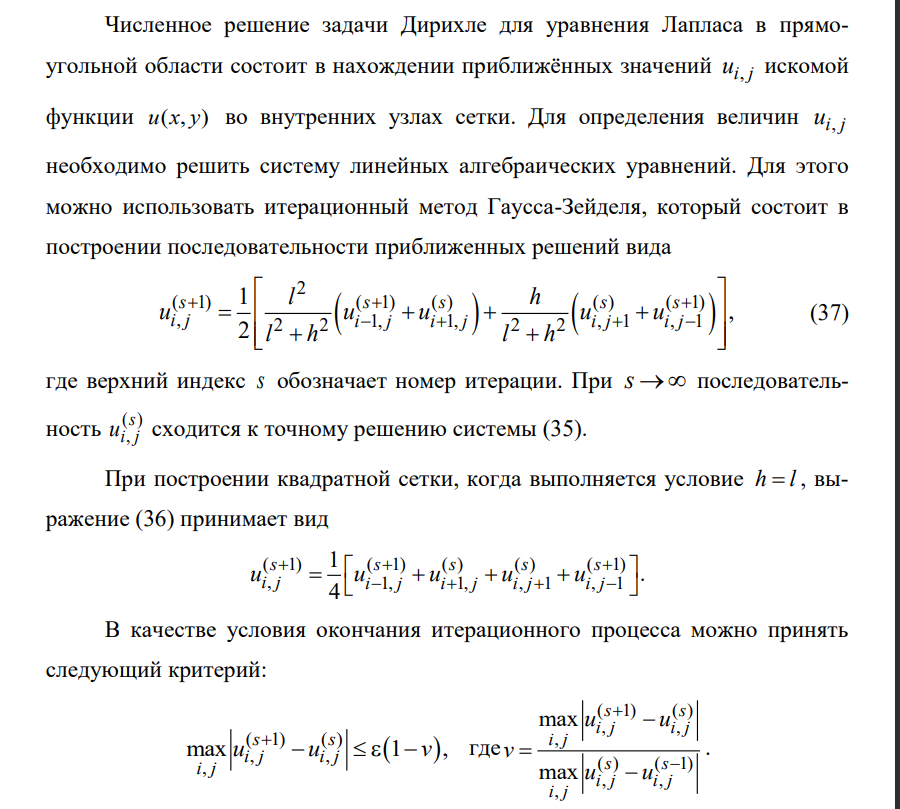


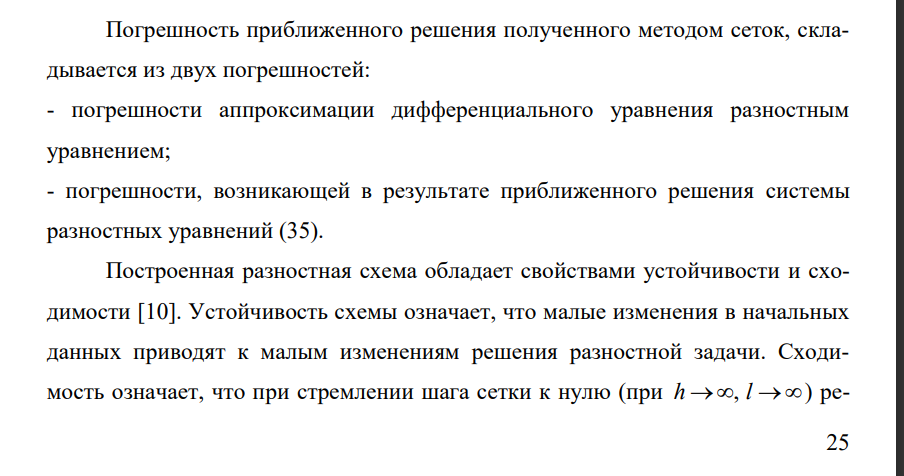
**Использованная теория.**

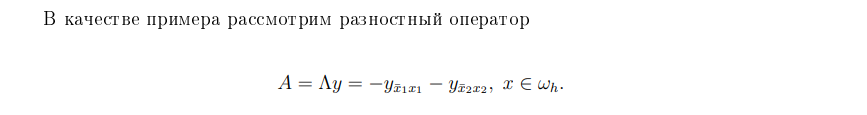
 

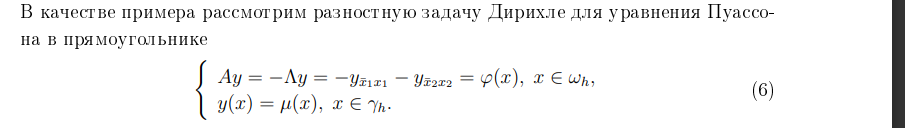


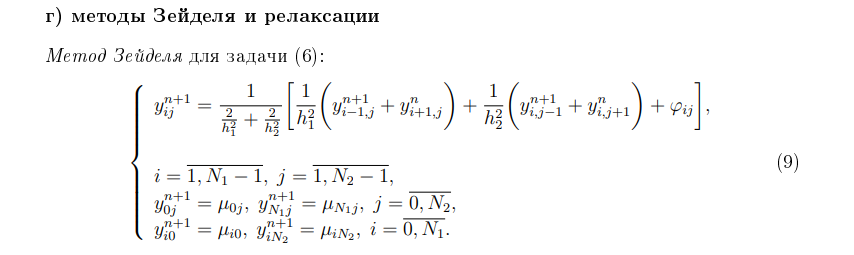




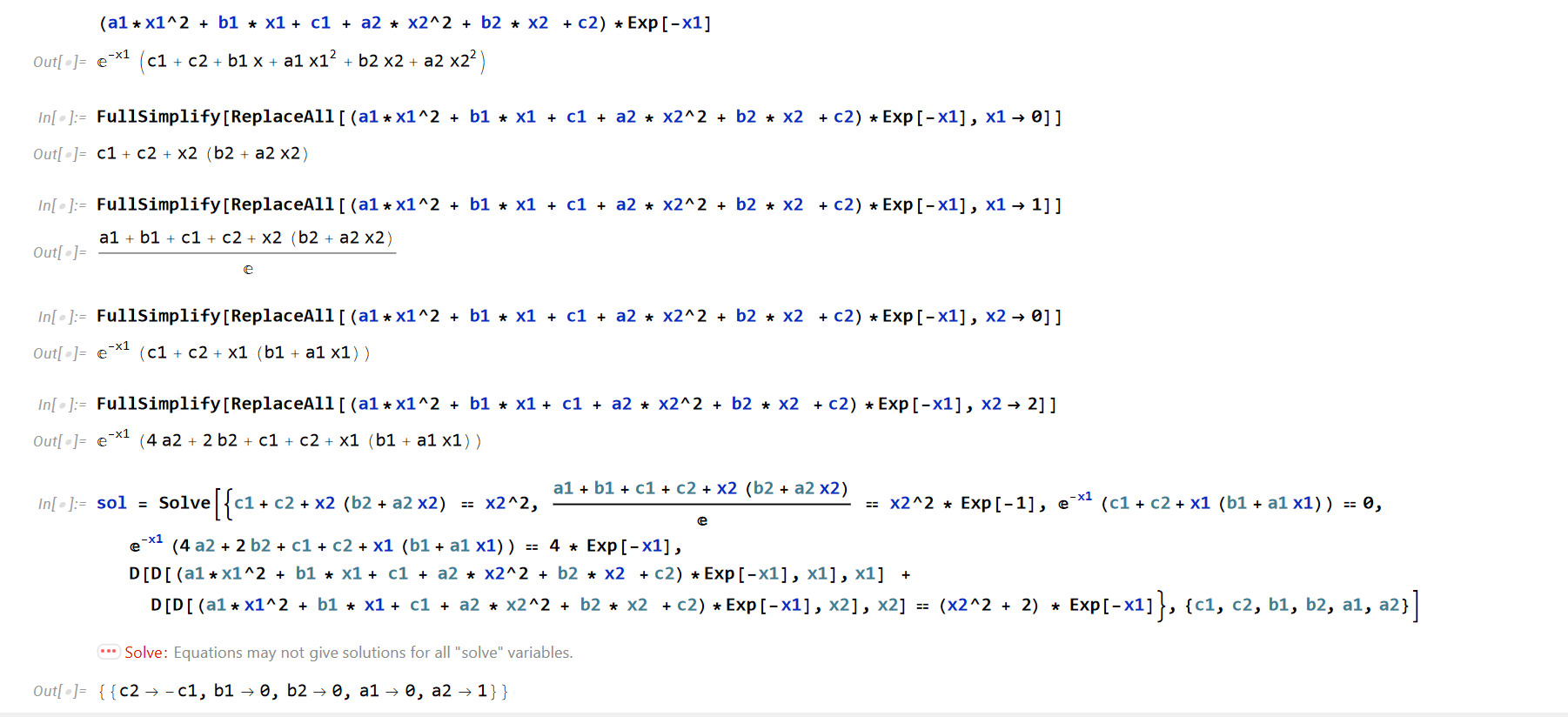


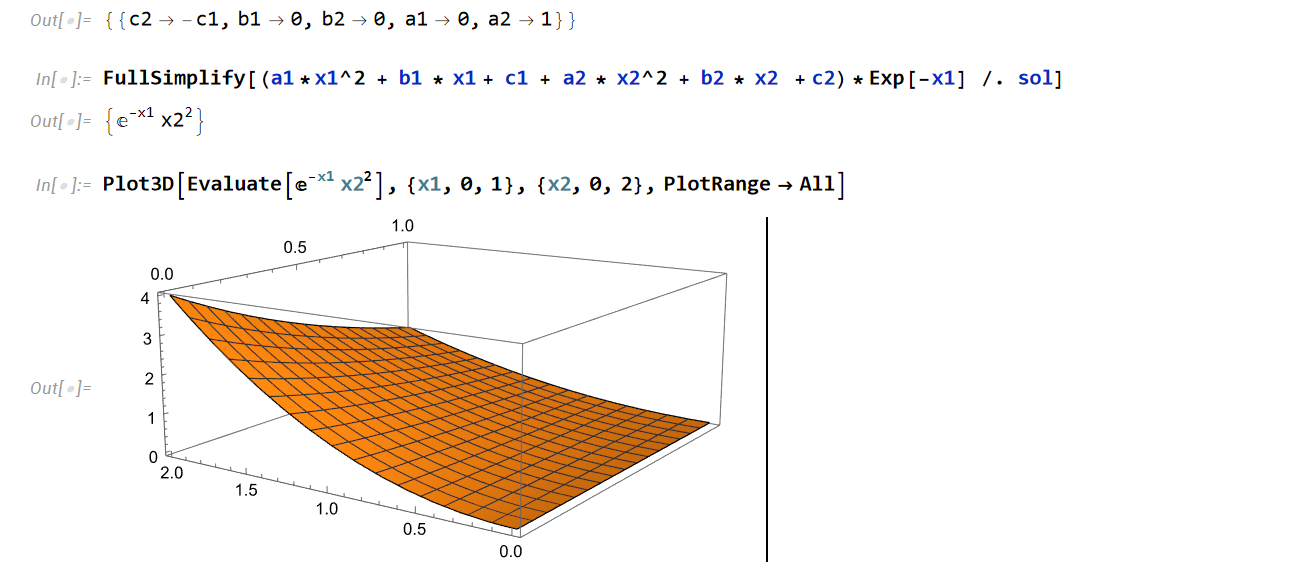






**Нахождение аналитического решения**

****

****

**Листинг программы**

*from* mpl\_toolkits.mplot3d *import* *Axes3D*

*import* matplotlib.pyplot *as* plt

*import* numpy *as* np

n = *2*

m = *1*

h = *0.1*

x1 = np.linspace(*0*, m, *int*(m/h)+*1*)

x2 = np.linspace(*0*, n, *int*(n/h)+*1*)

u = np.zeros((len(x1), len(x2)))

fi = np.zeros((len(x1), len(x2)))

def f(x1, x2):

*return* -(x2\*\**2* + *2*) \* np.exp(-x1)

def f1(x1):

*return* *0*

def f2(x2):

*return* x2\*\**2*

def f3(x1):

*return* *4*\*np.exp(-x1)

def f4(x2):

*return* x2\*\**2*\*np.exp(-*1*)

*for* i *in* *range*(len(x1)):

*for* j *in* *range*(len(x2)):

        fi[i][j] = f(x1[i], x2[j])

*for* i *in* *range*(len(x1)):

    u[i][*0*] = f1(x1[i])

*for* i *in* *range*(len(x2)):

    u[*0*][i] = f2(x2[i])

*for* i *in* *range*(len(x1)):

    u[i][-*1*] = f3(x1[i])

*for* i *in* *range*(len(x2)):

    u[-*1*][i] = f4(x2[i])

eps = *0.001*

u\_prev\_prev = np.copy(u)

u\_prev = np.copy(u)

k = *0*

*while* *True*:

*for* i *in* *range*(*1*, len(x1)-*1*):

*for* j *in* *range*(*1*, len(x2)-*1*):

            u[i][j] = *1*/*4* \* (u[i-*1*][j] + u[i+*1*][j] + u[i]

                             [j-*1*] + u[i][j+*1*] + h \* h \* fi[i][j])

    k += *1*

*if* k != *1*:

        v = np.max(np.abs(u-u\_prev)) / np.max(np.abs(u\_prev-u\_prev\_prev))

*if* (np.max(np.abs(u-u\_prev)) <= eps \* (*1*-v)):

*break*

    u\_prev\_prev = np.copy(u\_prev)

    u\_prev = np.copy(u)

def plot\_3d\_matrix(u):

    x = np.linspace(*0*, *1*, u.shape[*1*])

    y = np.linspace(*0*, *2*, u.shape[*0*])

    X, Y = np.meshgrid(x, y)

    fig = plt.figure()

    ax = fig.add\_subplot(*111*, projection='3d')

    ax.plot\_surface(X, Y, u, cmap='viridis')

    ax.set\_xlabel('X')

    ax.set\_ylabel('Y')

    ax.set\_zlabel('Z')

    plt.show()

print(u)

plot\_3d\_matrix(u)

def real\_solution(x1, x2):

*return* np.exp(-x1) \* x2\*\**2*

sol = np.zeros((len(x1), len(x2)))

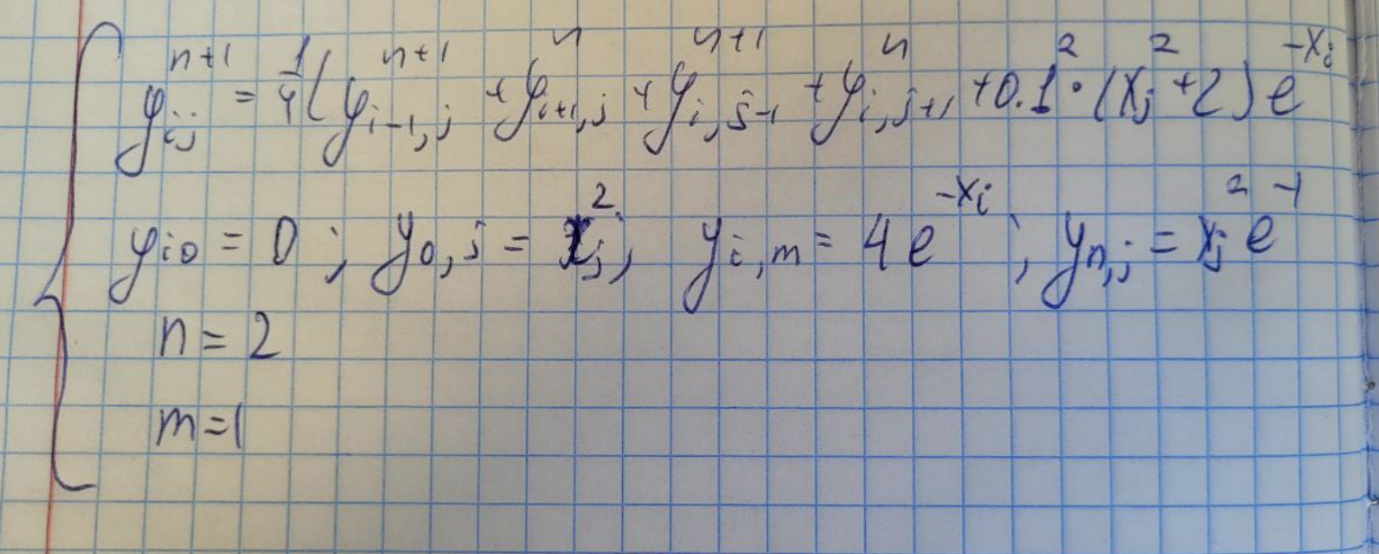
*for* i *in* *range*(len(x1)):

*for* j *in* *range*(len(x2)):

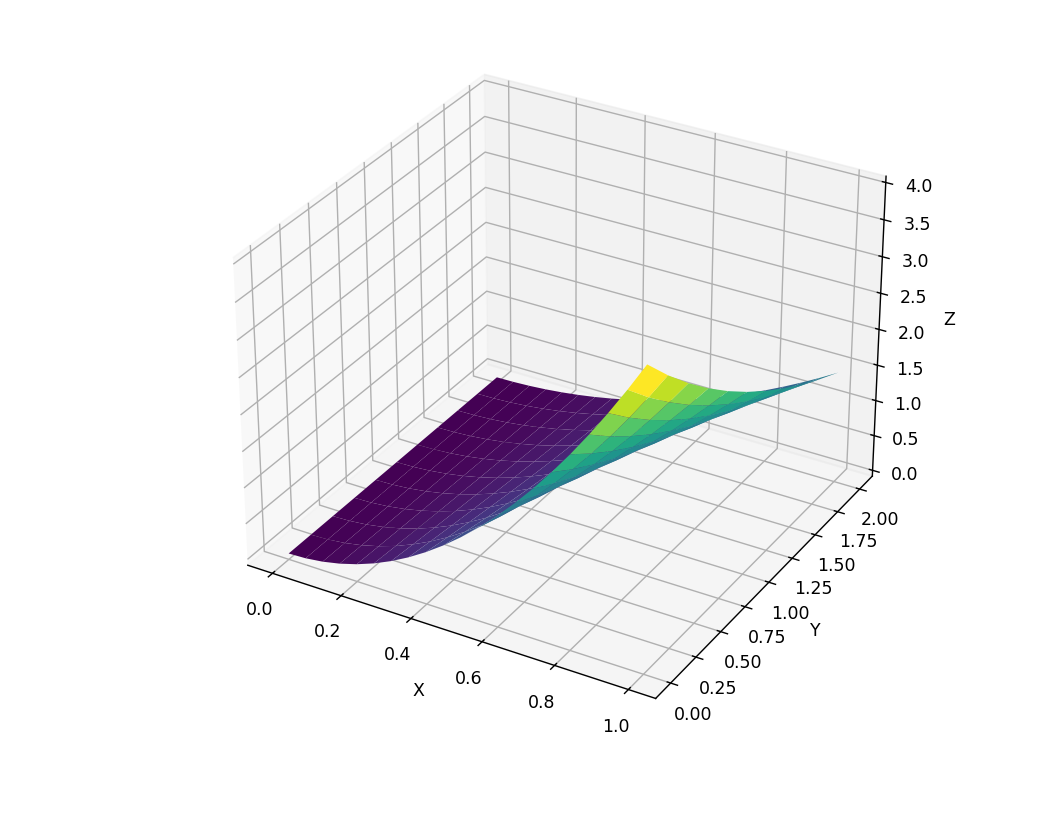
        sol[i][j] = real\_solution(x1[i], x2[j])

print(np.max(np.abs(sol-u)))

**Разностная схема**



**Выходные данные.**



[[0.00000000e+00 1.00000000e-02 4.00000000e-02 9.00000000e-02

1.60000000e-01 2.50000000e-01 3.60000000e-01 4.90000000e-01

6.40000000e-01 8.10000000e-01 1.00000000e+00 1.21000000e+00

1.44000000e+00 1.69000000e+00 1.96000000e+00 2.25000000e+00

2.56000000e+00 2.89000000e+00 3.24000000e+00 3.61000000e+00

4.00000000e+00]

[0.00000000e+00 8.98707421e-03 3.60762038e-02 8.12685098e-02

1.44564896e-01 2.25966053e-01 3.25472459e-01 4.43084377e-01

5.78801850e-01 7.32624702e-01 9.04552539e-01 1.09458475e+00

1.30272052e+00 1.52895883e+00 1.77329842e+00 2.03573786e+00

2.31627542e+00 2.61490900e+00 2.93163587e+00 3.26645192e+00

3.61934967e+00]

[0.00000000e+00 8.07418791e-03 3.25327341e-02 7.33776939e-02

1.30610731e-01 2.04233118e-01 2.94245735e-01 4.00649061e-01

5.23443171e-01 6.62627735e-01 8.18202018e-01 9.90164889e-01

1.17851483e+00 1.38324994e+00 1.60436794e+00 1.84186615e+00

2.09574145e+00 2.36599014e+00 2.65260765e+00 2.95558803e+00

3.27492301e+00]

[0.00000000e+00 7.25714525e-03 2.93436267e-02 6.62621742e-02

1.18015004e-01 1.84603811e-01 2.66029767e-01 3.62293505e-01

4.73395118e-01 5.99334156e-01 7.40109629e-01 8.95720020e-01

1.06616329e+00 1.25143690e+00 1.45153781e+00 1.66646248e+00

1.89620682e+00 2.14076610e+00 2.40013481e+00 2.67430635e+00

2.96327288e+00]

[0.00000000e+00 6.53098471e-03 2.64831409e-02 5.98595716e-02

1.06662799e-01 1.66894754e-01 2.40556765e-01 3.27649549e-01

4.28173204e-01 5.42127207e-01 6.69510419e-01 8.10321099e-01

9.64556916e-01 1.13221497e+00 1.31329180e+00 1.50778340e+00

1.71568520e+00 1.93699200e+00 2.17169795e+00 2.41979644e+00

2.68128018e+00]

[0.00000000e+00 5.88970603e-03 2.39250833e-02 5.41092929e-02

9.64449052e-02 1.50933887e-01 2.17577588e-01 2.96376734e-01

3.87331415e-01 4.90441090e-01 6.05704587e-01 7.33120118e-01

8.72685298e-01 1.02439716e+00 1.18825219e+00 1.36424632e+00

1.55237493e+00 1.75263285e+00 1.96501436e+00 2.18951319e+00

2.42612264e+00]

[0.00000000e+00 5.32619640e-03 2.16425223e-02 4.89518963e-02

8.72566905e-02 1.36558718e-01 1.96859220e-01 2.68158860e-01

3.50457716e-01 4.43755276e-01 5.48050451e-01 6.63341583e-01

7.89626462e-01 9.26902346e-01 1.07516598e+00 1.23441362e+00

1.40464103e+00 1.58584348e+00 1.77801574e+00 1.98115211e+00

2.19524654e+00]

[0.00000000e+00 4.83234036e-03 1.96078676e-02 4.43289968e-02

7.89976880e-02 1.23615437e-01 1.78183267e-01 2.42701719e-01

3.17170852e-01 4.01590237e-01 4.95958966e-01 6.00275662e-01

7.14538494e-01 8.38745189e-01 9.72893056e-01 1.11697899e+00

1.27099948e+00 1.43495059e+00 1.60882793e+00 1.79262661e+00

1.98634122e+00]

[0.00000000e+00 4.39928387e-03 1.77932564e-02 4.01836278e-02

7.15717832e-02 1.11958777e-01 1.61345330e-01 2.19731819e-01

2.87118282e-01 3.63504411e-01 4.48889561e-01 5.43272754e-01

6.46652692e-01 7.59027766e-01 8.80396069e-01 1.01075540e+00

1.15010325e+00 1.29843678e+00 1.45575275e+00 1.62204734e+00

1.79731586e+00]

[0.00000000e+00 4.01781216e-03 1.61711685e-02 3.64609471e-02

6.48878566e-02 1.01452436e-01 1.46155050e-01 1.98995892e-01

2.59974978e-01 3.29092148e-01 4.06347070e-01 4.91739241e-01

5.85267996e-01 6.86932510e-01 7.96731801e-01 9.14664737e-01

1.04073002e+00 1.17492616e+00 1.31725139e+00 1.46770340e+00

1.62627864e+00]

[0.00000000e+00 3.67879441e-03 1.47151776e-02 3.31091497e-02

5.88607106e-02 9.19698603e-02 1.32436599e-01 1.80260926e-01

2.35442842e-01 2.97982347e-01 3.67879441e-01 4.45134124e-01

5.29746395e-01 6.21716256e-01 7.21043705e-01 8.27728743e-01

9.41771369e-01 1.06317158e+00 1.19192939e+00 1.32804478e+00

1.47151776e+00]]

0.0008487442997374139