Московский Авиационный Институт
(Национальный исследовательский
университет)

Лабораторная работа №6
По курсу «Численные методы»

Студент: Ивченко А.В.
Группа: М8О-408Б-20
Преподаватель: Пивоваров Д. Е.

Москва, 2023

Задание:

Используя явную схему крест и неявную схему, решить начально-краевую задачу для дифференциального уравнения гиперболического типа. Аппроксимацию второго начального условия произвести с первым и со вторым порядком. Осуществить реализацию трех вариантов аппроксимации граничных условий, содержащих производные: двухточечная аппроксимация с первым порядком, трехточечная аппроксимация со вторым порядком, двухточечная аппроксимация со вторым порядком. В различные моменты времени вычислить погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением U(x,t) . Исследовать зависимость погрешности от сеточных параметров τ , h

Вариант:

9.

$$\frac{\partial^2 u}{\partial t^2} + 3\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial x} - u + \sin x \exp(-t),$$

$$u(0,t) = \exp(-t),$$

$$u(\pi,t) = -\exp(-t),$$

$$u(x,0) = \cos x,$$

$$u_t(x,0) = -\cos x.$$

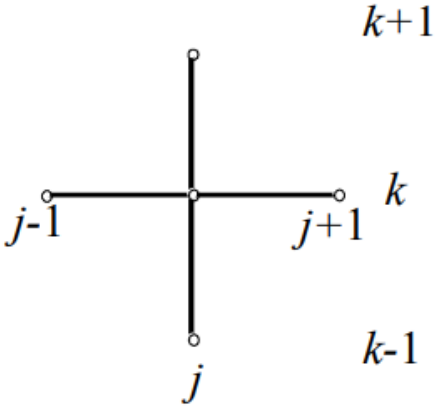
Аналитическое решение: $U(x,t) = \exp(-t)\cos x$.

Теория:

На пространственно-временной сетке с помощью отношений конечных разностей запишем наше ду.

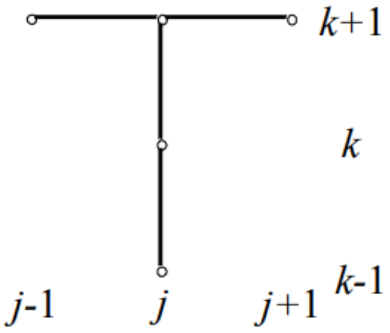
Явная схема крест:

$$\frac{u_j^{k+1} - 2u_j^k + u_j^{k-1}}{\tau^2} + 3\frac{u_j^{k+1} - u_j^k}{\tau} = \frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2} + \frac{u_{j+1}^k - u_{j-1}^k}{2h} - u_j^k + sin(x)e^{-t}$$



Неявная схема:

$$\frac{u_j^{k+1}-2u_j^k+u_j^{k-1}}{\tau^2}+3\frac{u_j^{k+1}-u_j^k}{\tau}=\frac{u_{j+1}^{k+1}-2u_j^{k+1}+u_{j-1}^{k+1}}{h^2}+\frac{u_{j+1}^{k+1}-u_{j-1}^{k+1}}{2h}-u_j^{k+1}+sin(x)e^{-t}$$



Граничные условия:

Вариант содержит граничное условие с производной:

$$\frac{\partial u(x,0)}{\partial t}=-\cos(x)$$

Двухточечная аппроксимация с первым порядком:

$$\frac{u_j^1-u_j^0}{\tau}=-\cos(x)$$

Двухточечная аппроксимация со вторым порядком:

$$u_j^1=u_j^0+\tau\frac{\partial u}{\partial t}+\frac{\tau^2}{2}\frac{\partial^2 u}{\partial t^2}$$

Из исходного уравнения:

$$\frac{\partial^2 u}{\partial t^2}=\frac{\partial^2 u}{\partial x^2}+\frac{\partial u}{\partial x}-u+\sin sin\,(x)\,e^{-t}=-\cos cos\,(x)-\sin sin\,(x)-\cos cos\,(x)$$

$$u_j^1=\cos cos\,(x)-\tau\cos cos\,(x)-\frac{\tau^2}{2}\Big(-\cos cos\,(x)-\sin sin\,(x)-\cos cos\,(x)$$

Код программы:

```
def neyavnaya1(n):
    sigma = tau**2/h/h
    u = [0]*n
    for i in range(n):
        u[i] = [0]*n

    for i in range(n):
        u[0][i] = np.cos(x[i])
    for j in range(n):
        u[1][j] = u[0][j] -np.cos(x[j])*tau
```

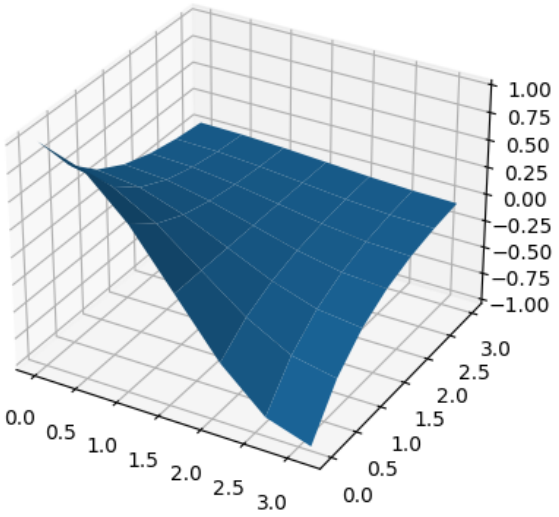
```
        #u[1][j] = u[0][j] -tau*np.cos(x[j]) +
tau**2/2*(-np.cos(x[j])-np.sin(x[j])-np.cos(x[j])+np.sin(x[j])*np.exp(-t[0])+3*
np.cos(x[j]))
        #u[1][j] = u[0][j] -tau*np.cos(x[j]) +
tau**2/2*((9*u[0][0]**2+16*u[0][1]**2+u[0][2]**2-24*u[0][0]*u[0][1]+6*u[0][0]*u
[0][2]-8*u[0][1]*u[0][2])/4/h/h+(-3*u[0][0]+4*u[0][1]-u[0][2])/2/h-u[0][j]+np.s
in(x[j])*np.exp(-t[0])+3*np.cos(x[j]))

    for i in range(n):
        u[i][0] = np.exp(-t[i])
        u[i][n-1] = -np.exp(-t[i])


    for k in range(1,n-1):
        A = [[0 for j in range(n+1)] for i in range(n)]
        A[0][0] = 1
        A[0][1] = 0
        A[0][n] = np.exp(-t[k+1])
        for j in range(1,n-1):
            A[j][j-1] = sigma
            A[j][j] = -1-3*tau-2*sigma-tau**2/h-1
            A[j][j+1] = sigma+1/h
            A[j][n] =
-2*u[k][j]-3*tau*u[k][j]-tau**2*np.sin(x[j])*np.exp(-t[k+1])+u[k-1][j]
        A[-1][n-2] = 0
        A[-1][n-1] = 1
        A[-1][n] = -np.exp(-t[k+1])

        #print(A)
        res = gauss(A)
        #print(res)

        for j in range(n):
            u[k+1][j] = res[j]
    return u

def yavnaya(n):
    u = [0]*n
    for i in range(n):
        u[i] = [0]*n

    for i in range(n):
        u[i][0] = np.exp(-t[i])
        u[i][n-1] = -np.exp(-t[i])

    for i in range(n):
        u[0][i] = np.cos(x[i])

    for j in range(n):
        #u[1][j] = u[0][j] -np.cos(x[j])*tau
        u[1][j] = u[0][j] -tau*np.cos(x[j]) +
tau**2/2*(-np.cos(x[j])-np.sin(x[j])-np.cos(x[j])+np.sin(x[j])*np.exp(-t[0])+3*
np.cos(x[j]))
        #u[1][j] = u[0][j] -tau*np.cos(x[j]) +
tau**2/2*((9*u[0][0]**2+16*u[0][1]**2+u[0][2]**2-24*u[0][0]*u[0][1]+6*u[0][0]*u
[0][2]-8*u[0][1]*u[0][2])/4/h/h+(-3*u[0][0]+4*u[0][1]-u[0][2])/2/h-u[0][j]+np.s
in(x[j])*np.exp(-t[0])+3*np.cos(x[j]))
    for j in range(1,n-1):
        for k in range(1,n-1):
            #u[k+1][j] =
((2*u[k][j]-u[k-1][j])/tau/tau+3*u[k][j]/tau+(u[k][j+1]-2*u[k][j]+u[k][j-1])/h/
h+(u[k][j+1]-u[k][j])/h-u[k][j]+np.sin(x[j])*np.exp(-t[k]))/(1/tau/tau+3/tau)
            u[k+1][j] =
(u[k][j]*(2/tau/tau+3/tau-2/h/h-1/h-1)+u[k-1][j]*(-1/tau/tau)+u[k][j+1]*(1/h/h+
```

```
1/h)+u[k][j-1]*(-1/h/h)+np.sin(x[j])*np.exp(-t[k+1]))/(1/tau/tau+3/tau)
    return u
```

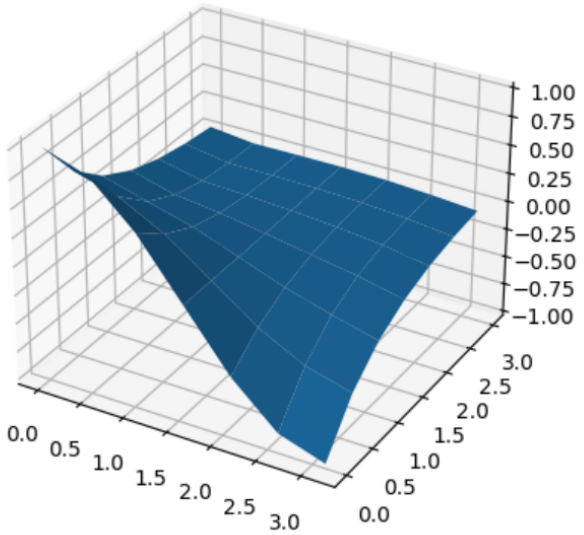
Результаты:

Аналитическое решение

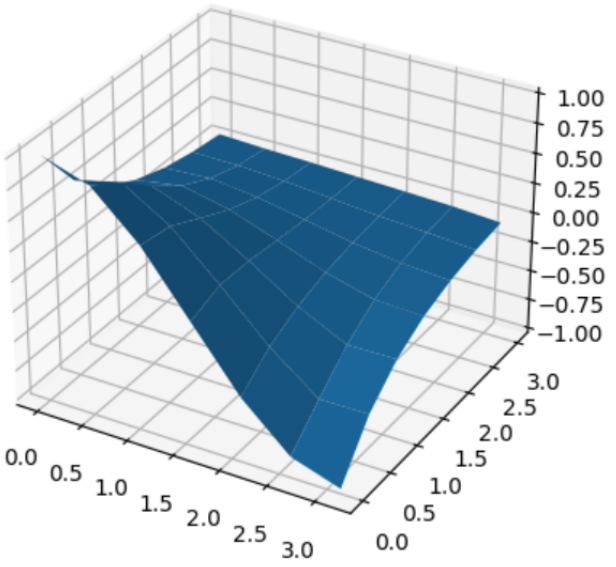
```
In [323]: def isxF(x,t):
              U = [0]*n
              for i in range(n):
                  U[i] = [0]*n

              for i in range(n): #t
                  for j in range(n):  #x
                      U[i][j] = np.exp(-t[i])*np.cos(x[j])
              return U

          U = isxF(x,t)
          #print(U)
          fig = plt.figure()
          ax = plt.axes(projection ='3d')
          ax.plot_surface(x_plt,t_plt,np.array(U))
          plt.show()
```



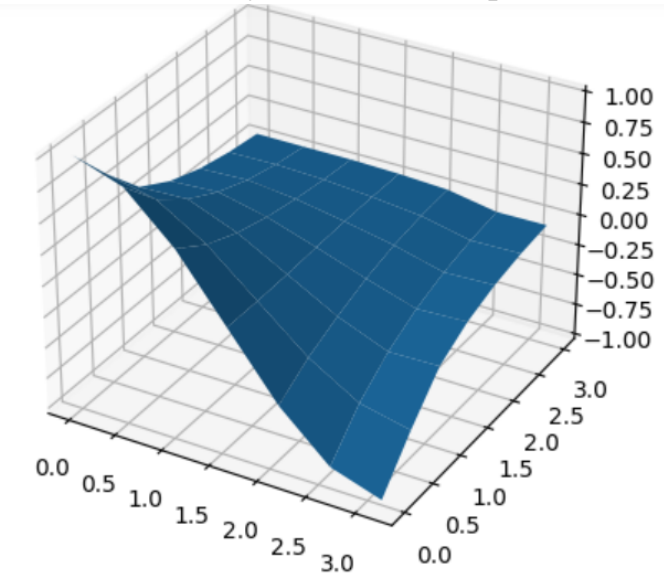
Неявная схема, двухточечная аппроксимация с первым порядком:



Неявная схема, двухточечная аппроксимация со вторым порядком:



Явная схема, двухточечная аппроксимация со вторым порядком:



Вывод:

В ходе проделанной работы, я успешно провела эксперимент с двумя методами решения начально-краевой задачи для дифференциального уравнения гиперболического типа. Результаты показали, что неявная схема дала более точное решение по сравнению с явной схемой. Мне удалось достичь более высокой точности при использовании неявного метода.