Московский авиационный институт

(национальный исследовательский институт)

Лабораторная работа №2

По курсу

Численные методы

Выполнил: Смирнов Д.А.

Группа: М8О-403Б-18

Москва 2021

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. Исследовать зависимость погрешности от сеточных параметров.

Вариант 8

$$\frac{\partial^2 u}{\partial t^2} + 2\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2} + 2\frac{\partial u}{\partial x} - 3u$$

$$u(0,t) = 0$$

$$u(\pi,t) = 0$$

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

$$u_t(x,0) = 2e^{-x} sinx$$

Аналитическое решение:

$u(x,t) = e^{-t-x} sinx * sin(2t)$.

Теоретический материал

Гиперболическое уравнение имеет вид:

$$\frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, 0 < x < l, t = 0$$

Первая начально-краевая задача для волнового уравнения имеет вид:

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, 0 < x < l, t = 0(1) \\ u(0,t) = \varphi_0(t), x = 0, t > 0(2) \\ u(l,t) = \varphi_l(t), x = l, t > 0(3) \\ u(x,0) = \psi_1(x), 0 \le x \le l, t = 0(4) \\ \frac{\partial u(x,0)}{\partial t} = \psi_2(x), 0 \le x \le l, t = 0(5) \end{cases}$$

Вторая начально-краевая задача для волнового уравнения имеет вид:

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, 0 < x < l, t = 0 \\ \frac{\partial u(0,t)}{\partial x} = \varphi_0(t), x = 0, t > 0 \\ \frac{\partial u(l,t)}{\partial x} = \varphi_l(t), x = l, t > 0 \\ u(x,0) = \psi_1(x), 0 \le x \le l, t = 0 \\ \frac{\partial u(x,0)}{\partial t} = \psi_2(x), 0 \le x \le l, t = 0 \end{cases}$$

Третья начально-краевая задача для волнового уравнения имеет вид:

$$\begin{cases} \frac{\partial^2 u}{\partial t^2} = a^2 \frac{\partial^2 u}{\partial x^2}, 0 < x < l, t = 0 \\ \alpha \frac{\partial u(0,t)}{\partial x} + \beta u(0,t) = \varphi_0(t), x = 0, t > 0 \\ \gamma \frac{\partial u(l,t)}{\partial x} + \delta u(l,t) = \varphi_l(t), x = l, t > 0 \\ u(x,0) = \psi_1(x), 0 \le x \le l, t = 0 \\ \frac{\partial u(x,0)}{\partial t} = \psi_2(x), 0 \le x \le l, t = 0 \end{cases}$$

В данном варианте – первая начально-краевая задача

Разностные схемы для аппроксимации:

$$\frac{u_j^{k+1} - 2u_j^k + u_j^{k-1}}{\tau^2} = a^2 \frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2} + O(\tau^2 + h^2), j = 1, \ldots, N-1, k = 1,2, \ldots$$

Схема является явной. С ее помощью решение $u_j^{k+1}, j = 1, \ldots, N-1$, определяется сразу, поскольку значения сеточных функции , на нижних временных слоях должны быть известны. В соответствии с шаблоном для этой схемы порядок аппроксимации равен двум, как по пространственной, так и по временной переменной. При этом явная конечно-разностная схема для волнового уравнения условно устойчива с условием $\sigma = \frac{a^2\tau^2}{h^2} < 1$, накладываемым на сеточные характеристики τ и h .

$$\frac{u_j^{k+1} - 2u_j^k + u_j^{k-1}}{\tau^2} = a^2 \frac{u_{j+1}^{k+1} - 2u_j^{k+1} + u_{j-1}^{k+1}}{h^2} + O(\tau^2 + h^2), j = 1, \ldots, N-1, k = 1,2, \ldots$$

Данная является неявной схемой и обладает абсолютной устойчивостью. Ее можно свести к СЛАУ с трехдиагональной матрицей, решаемой методом прогонки.

Формулы методов решения:

Неявный метод:

$$\frac{u_j^{k+1}-2u_j^{k}+u_j^{k-1}}{\tau^2}+2\frac{u_j^{k+1}-u_j^{k-1}}{2\tau}=\frac{u_{j-1}^{k+1}-2u_j^{k+1}+u_{j+1}^{k+1}}{h^2}+2\frac{u_{j+1}^{k+1}-u_{j-1}^{k+1}}{2h}-3u_j^{k}$$

$$u_{j-1}^{k+1}\left(-\frac{1}{h^2}+\frac{1}{h}\right)+u_j^{k+1}\left(\frac{1}{\tau^2}+\frac{1}{\tau}+\frac{2}{h^2}\right)+u_{j+1}^{k+1}\left(-\frac{1}{h^2}-\frac{1}{h}\right)=$$

$$=u_j^{k}\left(\frac{2}{\tau^2}-3\right)+u_j^{k-1}\left(-\frac{1}{\tau^2}+\frac{1}{\tau}\right)$$

Явный метод:

$$u_j^{k+1}=\frac{u_{j+1}^{k}\cdot\left(\frac{a^2\tau}{h^2}+\frac{\tau^2 b}{2h}\right)}{(\tau+1)}+\frac{u_j^{k}\left(2-2\frac{a^2\tau}{h^2}+c\tau^2\right)}{\tau+1}+\frac{u_{j-1}^{k}\cdot\left(\frac{a^2\tau}{h^2}-\frac{\tau^2 b}{2h}\right)}{\tau+1}+$$

$$+\frac{u_j^{k-1}(-1)(1-\tau)}{\tau+1}$$

$a=1$

$b=2$

$c=-3$

Первый слой:

Первый порядок точности:

$$u_j^1=0+\tau(2e^{-x}\sin x)$$

Второй порядок точности:

$$u_j^1=0+\tau(2e^{-x}\sin x)+\frac{\tau^2}{2}(2e^{-x}\sin x)''=2\tau e^{-x}\sin x-2\tau^2 e^{-x}\cos x$$

Программа

Определяем начально-краевые условия

```
# значение на левом конце
def Ux1(t):
    return 0

# значение на правом конце
def Ux2(t):
    return 0

# функция в начальный момент времени
def U(x):
    return 0

# производная по времени функции в начальный момент времени
def Ut(x):
    return 2*np.exp(-x)*np.sin(x)
```

Явная схема:

```
def explicit(t, m, n, x0, xl, aprox_f):

    space_step = (xl - x0) / (m - 1)
    time_step = t / (n - 1)


    Uarray = autofill(x0, space_step, m, n, param_a, time_step, aprox_f)

    sigma = param_a**2 * time_step**2 / space_step**2

    for k in range(1, n - 1):
        for j in range(1, m - 1):

            Uarray[k + 1][j] = \
                Uarray[k][j + 1] *(sigma + time_step**2*param_b/(2*space_step))/(time_step+1) +\
                Uarray[k][j] * (2 - 2*sigma + param_c*time_step**2) /(time_step+1) + \
                Uarray[k][j - 1] *(sigma - time_step**2*param_b/(2*space_step))/(time_step+1) + \
                (Uarray[k - 1][j] * (-1)*(1 - time_step))/(time_step+1)

        Uarray[k + 1][0] = Ux1(time_step*k)
        Uarray[k + 1][m - 1] = Ux2(time_step*k)
```

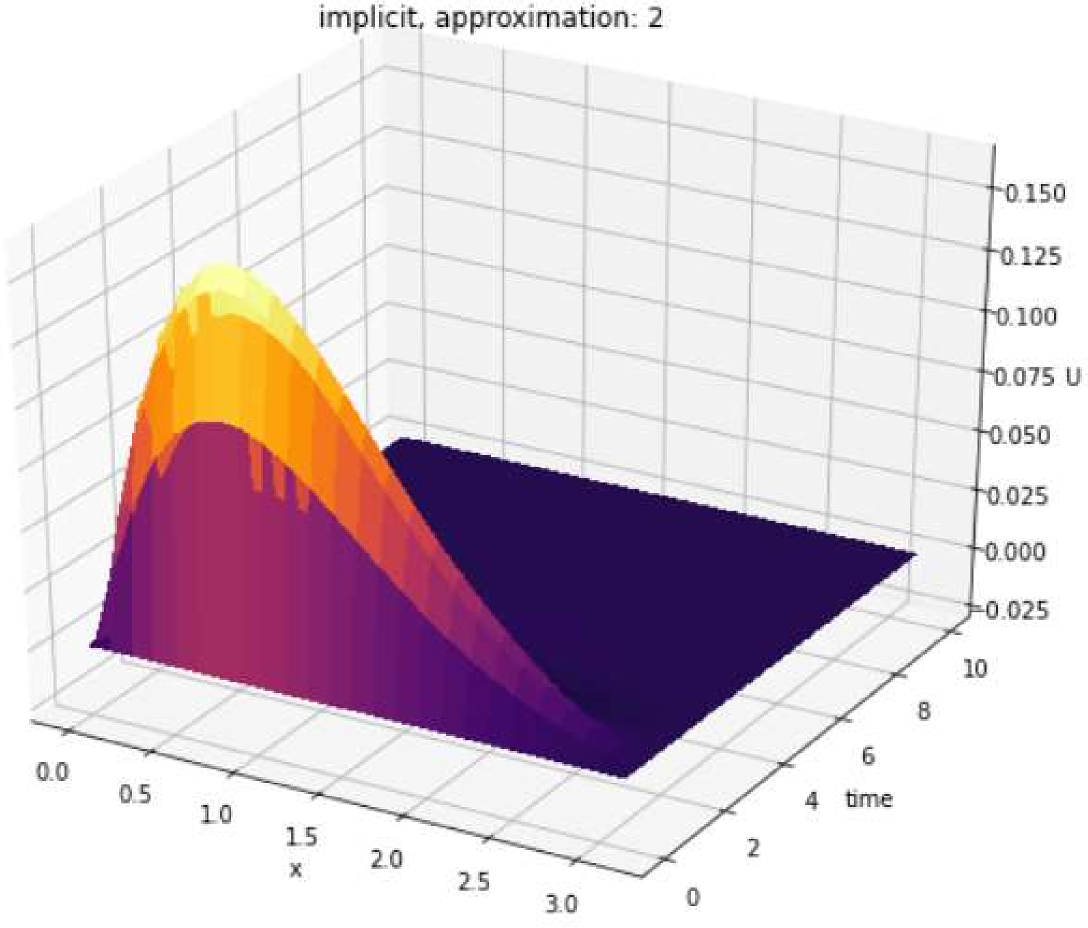
```
    return Uarray
```

Неявная схема:

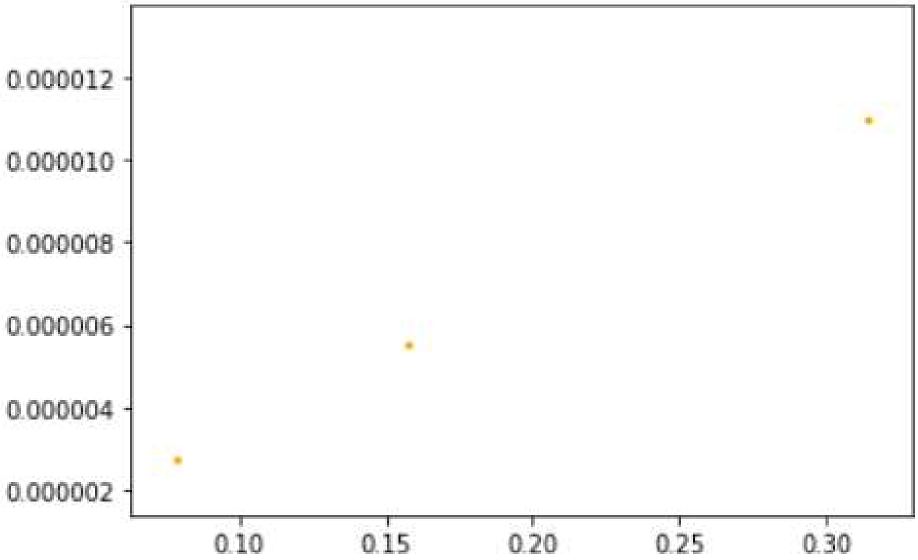
```
def implicit(t, m, n, x0, xl, aprox_f):

    space_step = (xl - x0) / (m - 1)
    time_step = t / (n - 1)


    Uarray = autofill(x0, space_step, m, n, param_a, time_step, aprox_f)

    sigma = param_a**2 * time_step**2 / space_step**2

    alpha = 0
    betta = 1
    gamma = 0
    delta = 1
    tmp = param_b*time_step**2/(2*space_step)

    for k in range(1, n - 1):
        a = np.zeros(m)
        b = np.zeros(m)
        c = np.zeros(m)
        d = np.zeros(m)

        for j in range(1, m - 1):
            a[j] = -sigma + tmp
            b[j] = (time_step + 1 + 2*sigma - param_c*time_step**2)
            c[j] = -sigma - tmp
            d[j] = Uarray[k - 1][j]*(time_step - 1) + 2*Uarray[k][j]

        b[0] = betta - alpha / space_step
        c[0] = alpha / space_step
        d[0] = Ux1((k + 1) * time_step)

        a[m - 1] = - gamma / space_step
        b[m - 1] = delta + gamma / space_step
        d[m - 1] = Ux2((k + 1) * time_step)


        Y = runMethod(a, b, c, d, m)
        Uarray[k + 1] = Y
    return Uarray
```

Результат работы:

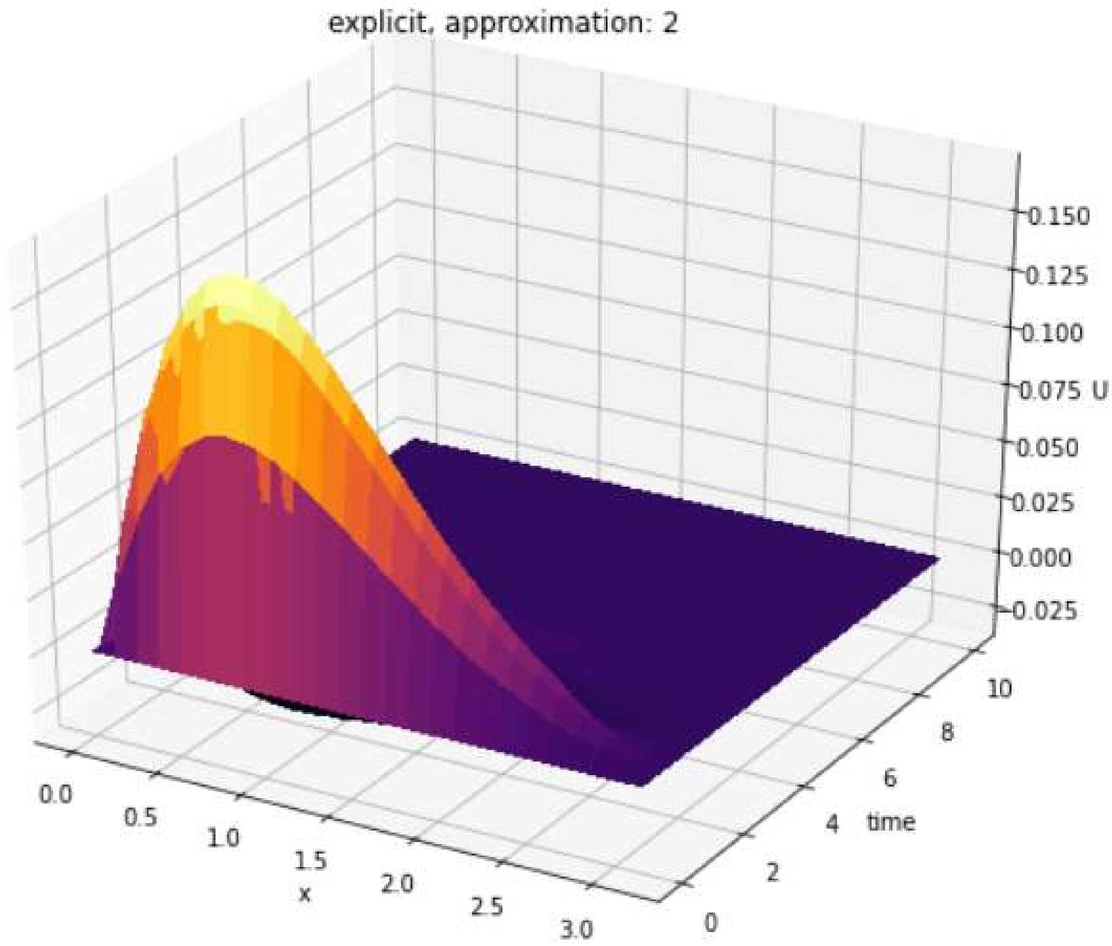
Неявный метод:



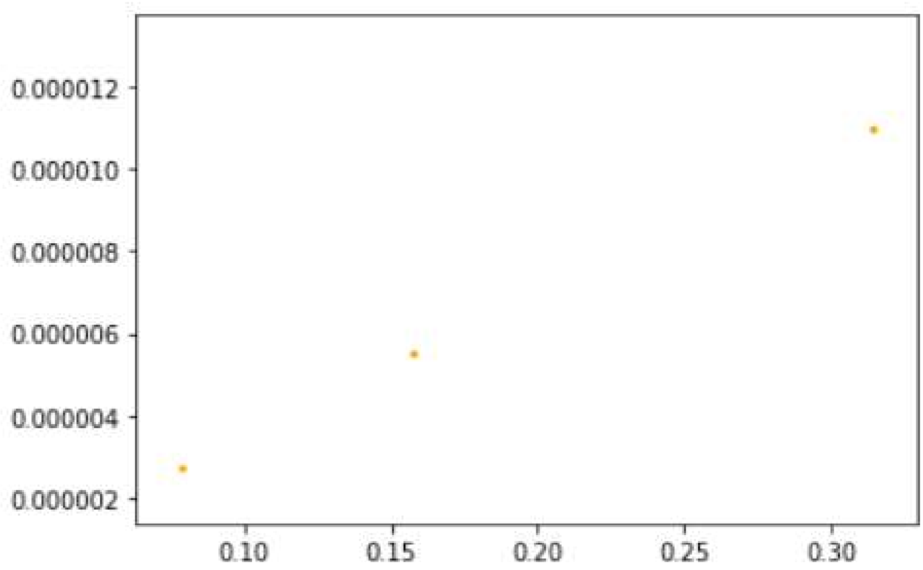
Изменение ошибки в зависимости от размера шага по пространству из П/10, П/20 и П/40:


0.000012
0.000010
0.000008
0.000006
0.000004
0.000002
0.10
0.15
0.20
0.25
0.30

Явный метод:


explicit, approximation: 2
0.150
0.125
0.100
0.075
U
0.050
0.025
0.000
-0.025
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x
0
2
4
6
8
10
time

Изменение ошибки в зависимости от размера шага по пространству из П/10, П/20 и П/40:



Выводы:

Были реализованы два численных метода на питоне для решения задачи гиперболического типа.

Первый - Явный метод. Он использует в правой части значения только с предыдущего временного слоя, и получает явной выражения для решения на новом временном слое.

Второй - Неявный метод. Использует в правой части значения с текущего слоя, что приводит к решению СЛАУ с помощью метода прогонки.

Оба метода дали хорошую сходимость. Ошибка решений падает с уменьшением размеров шагов по времени и пространству.

Были построены графики решений и ошибок по сравнению с аналитическим решением.