Московский авиационный институт

(Национальный исследовательский университет)

Институт №8 «Компьютерные науки и прикладная математика»

Кафедра вычислительной математики и программирования

Лабораторная работа №6

по курсу «Численные методы»

Студент: Мариечев К.Д.

Группа: М8О-409Б-20

Проверил: Пивоваров Д.Е.

Дата:

Оценка:

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:

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial u}{\partial x},\; a > 0,\; b > 0.$$

$$u_x(0,t) - u(0,t) = -\exp(-at)(\cos(bt) + \sin(bt)),$$
$$u_x(\pi,t) - u(\pi,t) = \exp(-at)(\cos(bt) + \sin(bt)),$$

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

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

Теория

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

$$u(0,t) = \varphi_0(t), \quad x = 0, \quad t > 0;$$
$$u(l,t) = \varphi_l(t), \quad x = l, \quad t > 0;$$
$$u(x,0) = \psi_1(x), \quad 0 \le x \le l, \quad t = 0;$$
$$\frac{\partial u(x,0)}{\partial t} = \psi_2(x), \quad 0 \le x \le l, \quad t = 0,$$

Пусть N разбиений по x, k разбиений по t. Тогда $h=l/N$, $\tau=T/k$, $x_i=l+ih$, $t_k=k\,\tau$.

u(x,o) заполняем 0-ой слой

Далее заполняем 1-ый слой, используя последнее условие

1 способ аппроксимации

$$\frac{u_j^1 - u_j^0}{\tau} = \psi_2(x_j)$$

Тогда

$$u_j^1 = \psi_1(x_j) + \psi_2(x_j)\tau \,.$$

2 способ аппроксимации

Для повышения порядка аппроксимации разложим u_j^1 в ряд Тейлора на точном решении по времени в окрестности $t=0$:

$$u_j^1 = u(x_j, 0+\tau) = u_j^0 + \left.\frac{\partial u}{\partial t}\right|_j^0 \tau + \left.\frac{\partial^2 u}{\partial t^2}\right|_j^0 \frac{\tau^2}{2} + O(\tau^3)\,,$$

где, согласно исходному уравнению

$$\left.\frac{\partial^2 u}{\partial t^2}\right|_j^0 = a^2 \left.\frac{\partial^2 u}{\partial x^2}\right|_j^0 + b\left.\frac{\partial u}{\partial x}\right|_j^0 + c u_j^k + f_j^k = a^2 \psi_1''(x_j) + b\psi_1'(x_j) + c\psi_1(x_j) + f_j^k - 3\left.\frac{\partial\ u}{\partial t}\right|_j^0$$

Окончательно получаем

$$u_j^1 = \psi_1(x_j) + \psi_2(x_j)\tau + (a^2\psi_1''(x_j) + b\psi_1'(x_j) + c\psi_1(x_j) + f_j^k - 3\psi_2(x_j)\,)\frac{\tau^2}{2}\,.$$

Явная схема

$$\frac{u_j^{k+1} - 2u_j^k + u_j^{k-1}}{\tau^2} - 3\,\frac{u_j^{k+1} - u_j^{k-1}}{2\tau} = \frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2} + \frac{u_{j+1}^k - u_{j-1}^k}{2h} - u_j^{\,k} + f_j^{\,k}, \quad j = \overline{1, N-1}; \quad k = 1, 2, \ldots$$

где $f_j^k = f(x_j, t^k)$.

Выразим u_j^{k+1}

$${}_j^{k+1} = \left(\frac{u_{j+1}^k - 2u_j^k + u_{j-1}^k}{h^2} + \frac{u_{j+1}^k - u_{j-1}^k}{2h} - u_j^k + f_i^k + \frac{2u_j^k - u_j^{k-1}}{\tau^2} + 3\,\frac{u_j^{k-1}}{2\tau} \right) * \frac{2\tau^2}{2+3\tau}$$

Неявная схема

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

Составим систему линейных уравнений, где коэффициенты

u_{i-1}^{k+1}: $\quad -\frac{1}{h^2} + \frac{1}{2h}$

При u_i^{k+1}: $\quad \frac{1}{\tau^2} + \frac{3}{2\tau} + \frac{2}{h^2} + 1$

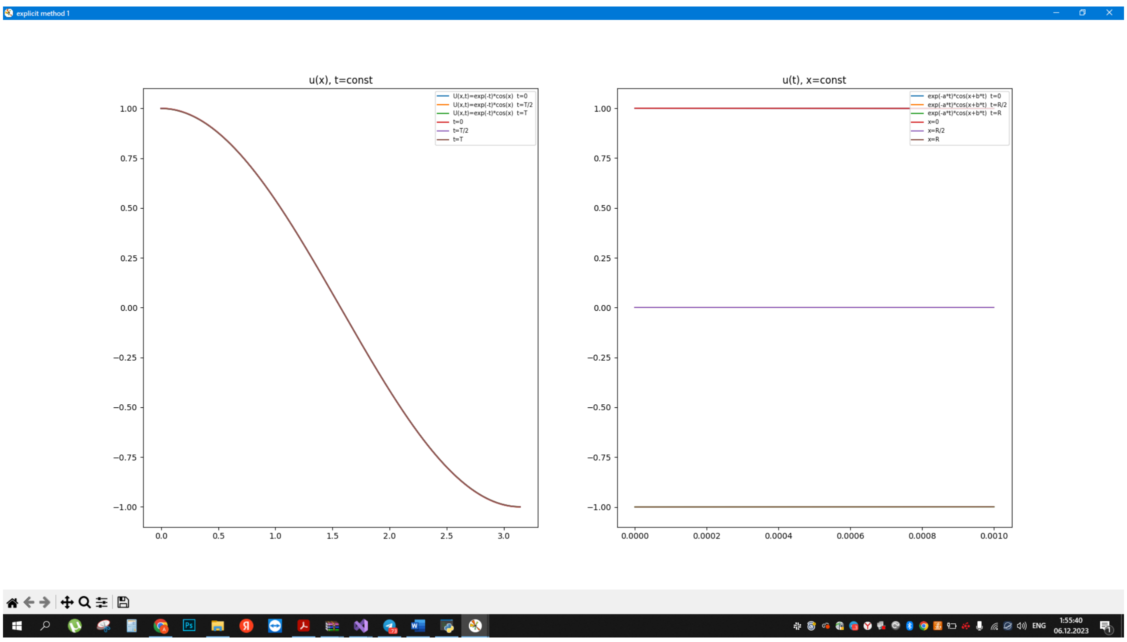
При u_{i+1}^{k+1}: $\quad -\frac{1}{h^2} - \frac{1}{2h}$

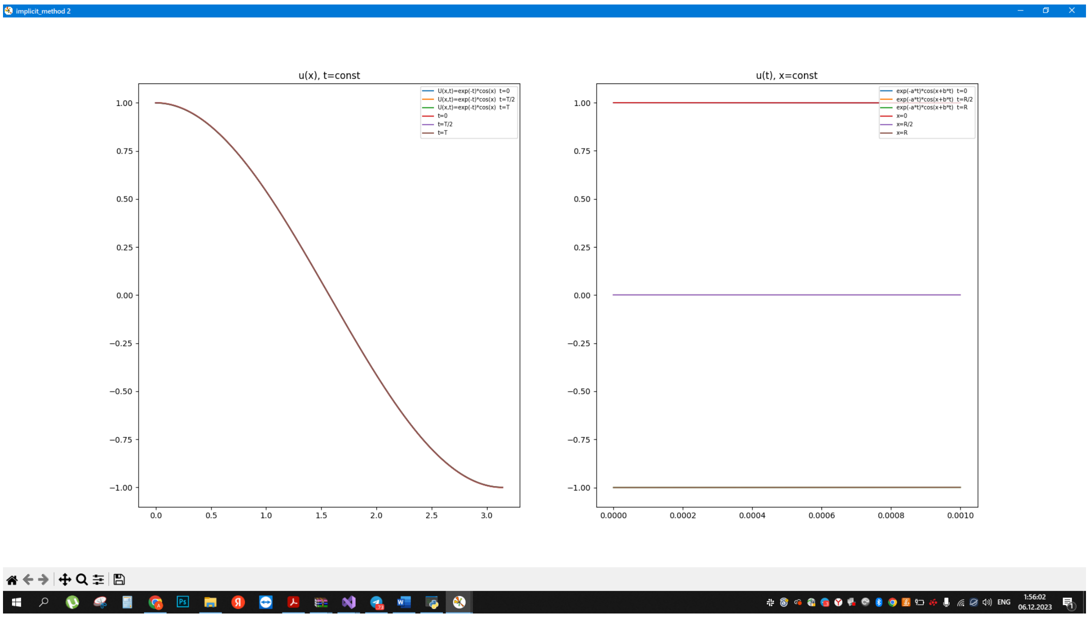
Правая часть: $\frac{2u_i^k - u_i^{k-1}}{\tau^2} + \frac{3u_i^k}{2\tau} + f_i^{k+1}$

Матрица коэффициентов имеет трехдиагональный вид. Найдем u^{k+1}, решив СЛАУ методом прогонки.

Решение

При k=100, N=100, T=0.001






```
C:\Users\Home PC\AppData\Local\Programs\Python\Python311\python.exe
0.03s - Debugger warning: It seems that frozen modules are being used, which may
0.00s - make the debugger miss breakpoints. Please pass -Xfrozen_modules=off
0.00s - to python to disable frozen modules.
0.00s - Note: Debugging will proceed. Set PYDEVD_DISABLE_FILE_VALIDATION=1 to disable this validation.

tau <= h
1e-05 <= 0.0031415926535897933

eroor explicit method 1
2.052191063445079e-09

eroor explicit method 2
2.724222925390113e-13

eroor implicit_method 1
2.048851501781290 5e-09словия со вторым порядком

eroor implicit_method 2
3.3780960651644615e-12
```

```
solve(f1,f2):
u=np.zeros((K+1,N+1))

# заполняем 1 слой
k=0
for i in range(N+1):
    x=L+i*h
    u[k][i]=psi1(x)

# заполняем 2 слой
k=1
for i in range(N+1):
    if f1==1:
        x=L+i*h
        u[k][i]=psi1(x)+tau*psi2(x)
    if f1==2:
        x=L+i*h
        t=k*tau
        #u[k][i]=psi1(x)+tau*psi2(x)+tau**2*(-math.cos(x)-math.sin(x)-
psi1(x)+f(x,0)-psi2(x))/2
        if i==0:
            u[k][i]=phi1(t)
        if i==N:
           u[k][i]=phi2(t)
```

```
            else :
               u[k][i]=psi1(x)+tau*psi2(x)+tau**2*(dpsi1_dx(x)+d2psi1_dx2(x)-
psi1(x)+f(x,0)-3*psi2(x))/2


    # заполняем k+1 слой

    for k in range(1,K):
        if f2==1:
            for i in range(1,N):
                u[k+1][i]=((u[k][i+1]-2*u[k][i]+u[k][i-1])/h**2+(u[k][i+1]-u[k][i-
1])/(2*h)-u[k][i]+f(L+i*h,k*tau)+(2*u[k][i]-u[k-1][i])/tau**2+3*u[k-
1][i]/(2*tau))*((2*tau**2)/(2+3*tau))

            t=tau*(k+1)
            u[k+1][0]=phi1(t)
            u[k+1][N]=phi2(t)
        if f2==2:
            A=np.zeros((N+1,N+1))
            B=np.zeros((N+1,1))
            t=tau*(k+1)

            A[0][0]=1
            B[0][0]=phi1(t)

            A[N][N]=1
            B[N][0]=phi2(t)

            for i in range(1,N):
                x=L+i*h
                A[i][i-1]=-1/h**2+1/(2*h)
                A[i][i]=1/tau**2+3/(2*tau)+2/h**2+1
                A[i][i+1]=-1/h**2-1/(2*h)
                B[i][0]=(2*u[k][i]-u[k-1][i])/tau**2+3*u[k-1][i]/(2*tau)+f(x,t)
            X=running_method(A,B,N+1)

            for i in range(0,N+1):
                u[k+1][i]=X[i]
    return(u)
```

Вывод

В данной работе была решена начально-краевая задача для ДУ гиперболического типа двумя способами:

- с помощью явной конечно-разностной схемы
- с помощью неявной конечно-разностной схемы

С помощью каждого метода получилось решить заданное ДУ с хорошей точностью.

Явная конечно-разностная схема легко считается, но она не всегда устойчива и, соответственно, не всегда гарантирует хороший результат.

Неявная конечно-разностная схема считается более сложным образом, но зато абсолютно устойчива