Московский Авиационный Институт
(Национальный исследовательский университет)

Лабораторная работа №8
По курсу «Численные методы»

Студент:	Ивченко А.В.
Группа:	М8О-408Б_20
Преподаватель:	Пивоваров Д. Е.

Москва, 2023

Задание:

Используя схемы переменных направлений и дробных шагов, решить двумерную начально-краевую задачу для дифференциального уравнения параболического типа. 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, hx , hy

Вариант:

9.

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial^2 u}{\partial y^2} + \sin x \sin y(\mu\cos\mu t + (a+b)\sin\mu t),$$

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

$$u(\frac{\pi}{2},y,t) = \sin y \sin(\mu t),$$

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

$$u_y(x,\pi,t) = -\sin x \sin(\mu t),$$

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

Аналитическое решение: $U(x,y,t) = \sin x \sin y \sin(\mu t)$.

Теория:

В схеме метода переменных направлений (МПН), как и во всех методах расщепления, шаг по времени τ разбивается на число независимых пространственных переменных (в двумерном случае - на два). На каждом дробном временном слое один из пространственных дифференциальных операторов аппроксимируется неявно (по соответствующему координатному направлению осуществляются скалярные прогонки), а остальные явно. На следующем дробном шаге следующий по порядку дифференциальный оператор аппроксимируется неявно, а остальные – явно и т.д

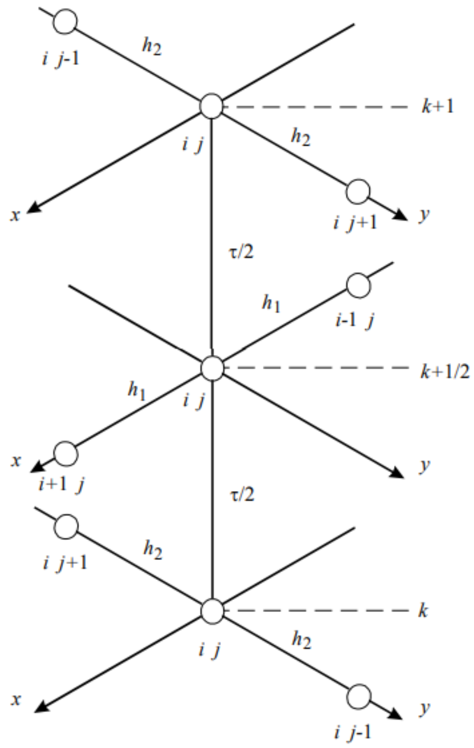
Положим, $h_x = h_y = h$

Схема МНП имеет вид:

$$\frac{u_{i,j}^{k+1/2} - u_{i,j}^{k}}{\tau/2} = \frac{a}{h^2}\left(u_{i+1,j}^{k+1/2} - 2u_{i,j}^{k+1/2} + u_{i-1,j}^{k+1/2}\right) + \frac{b}{h^2}\left(u_{i,j+1}^{k} - 2u_{i,j}^{k} + u_{i,j-1}^{k}\right)$$
$$+ \sin sin\left(x_i\right) \sin sin\left(y_j\right)\left(\mu \cos cos\left(\mu t^{k+1/2}\right) + (a+b) \sin sin\left(\mu t^{k+1/2}\right)\right)$$

$$\frac{u_{i,j}^{k+1} - u_{i,j}^{k+1/2}}{\tau/2} = \frac{a}{h^2}\left(u_{i+1,j}^{k+1/2} - 2u_{i,j}^{k+1/2} + u_{i-1,j}^{k+1/2}\right) + \frac{b}{h^2}\left(u_{i,j+1}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j-1}^{k+1}\right) +$$

$$+ \sin sin\left(x_i\right) \sin sin\left(y_j\right)\left(\mu \cos cos\left(\mu t^{k+1/2}\right) + (a + b)\sin sin\left(\mu t^{k+1/2}\right)\right)$$



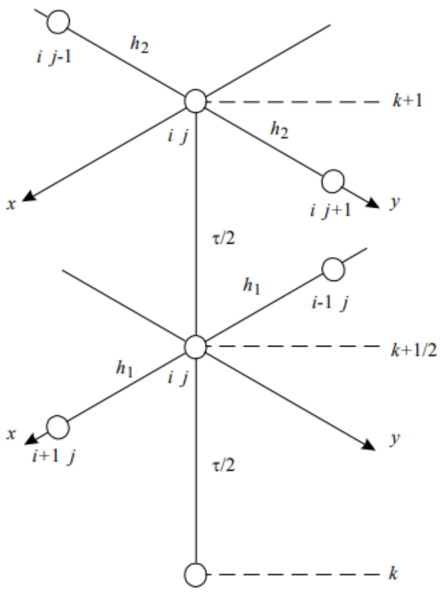
В отличие от МПН метод дробных шагов (МДШ) использует только неявные конечно разностные операторы, что делает его абсолютно устойчивым в задачах, не содержащих смешанные производные. Он обладает довольно значительным запасом устойчивости и в задачах со смешанными производными

МДШ схема имеет вид:

$$\frac{u_{i,j}^{k+1/2} - u_{i,j}^{k}}{\tau} = \frac{a}{h^2}\left(u_{i+1,j}^{k+1/2} - 2u_{i,j}^{k+1/2} + u_{i-1,j}^{k+1/2}\right) + \frac{b}{h^2}\left(u_{i+1,j}^{k+\frac{1}{2}} - 2u_{i,j}^{k+\frac{1}{2}} + u_{i-1,j}^{k+\frac{1}{2}}\right) + \frac{\sin sin\left(x_i\right)\sin sin\left(y_j\right)\left(\mu\cos cos\left(\mu t^{k}\right) + (a+b)\sin sin\left(\mu t^{k}\right)\right)}{2}$$

$$\frac{u_{i,j}^{k+1} - u_{i,j}^{k+1/2}}{\tau} = \frac{a}{h^2}\left(u_{i,j+1}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j-1}^{k+1}\right) + \frac{b}{h^2}\left(u_{i,j+1}^{k+1} - 2u_{i,j}^{k+1} + u_{i,j-1}^{k+1}\right)$$

$$\frac{\sin sin\left(x_i\right)\sin sin\left(y_j\right)\left(\mu\cos cos\left(\mu t^{k+1}\right) + (a+b)\sin sin\left(\mu t^{k+1}\right)\right)}{2}$$



Код программы:

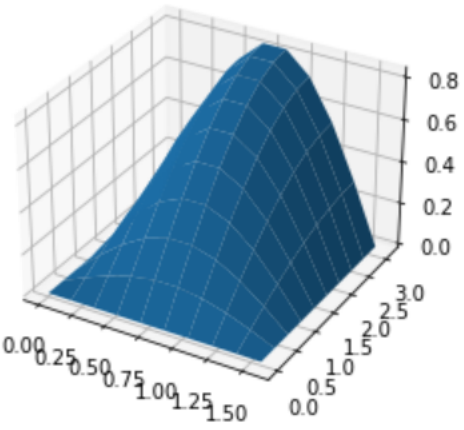
```
def mpn(n):
    u = np.zeros((n,n,n))
    for i in range(n):
        for j in range(n):
            u[i][j][0] = 0

    for k in range(1,n):
        u1 = np.zeros((n,n))
        u2 = np.zeros((n,n))
        tau2 = t[k-1] + tau/2
        for i in range(n):
            u1[i][0] = 0
            u1[0][i] = 0
            u1[-1][i] = np.sin(y[i])*np.sin(mu*tau2)
            u2[i][0] = 0

        for j in range(n-1):
            a = np.zeros(n)
            b = np.zeros(n)
            c = np.zeros(n)
            d = np.zeros(n)
            a[0] = 0
            b[0] = 1
            c[0] = 0
            d[0] = 0
            for i in range(1,n-1):
                a[i] = acoef/h1/h1*tau/2
                b[i] = -(2/tau + 2*acoef/h1/h1)*tau/2
                c[i] = acoef/h1/h1*tau/2
                d[i] = -(2*u[i][j][k-1]/tau +
bcoef/h2/h2*(u[i][j+1][k-1]-2*u[i][j][k-1]+u[i][j-1][k-1])-f(x[i],y[j],tau2))*t
au/2
            a[-1] = 0
            b[-1] = 1
            c[-1] = 0
            d[-1] = np.sin(y[j])*np.sin(mu*tau2)

            result = slau([a,b,c,d])
            #print(result)
            for i in range(n):
```

```
                u1[i][j] = result[i]
                u1[i][-1] = -np.sin(x[i])*np.sin(mu*tau2)*h1 + u1[i][-2]


        for i in range(n-1):
            a = np.zeros(n)
            b = np.zeros(n)
            c = np.zeros(n)
            d = np.zeros(n)
            a[0] = 0
            b[0] = 1
            c[0] = 0
            d[0] = 0
            for j in range(1,n-1):
                a[j] = bcoef/h2/h2*tau/2
                b[j] = -(2/tau + 2*bcoef/h2/h2)*tau/2
                c[j] = bcoef/h2/h2*tau/2
                d[j] = -(u1[i][j]/tau/2 +
acoef/h1/h1*(u1[i+1][j]-2*u1[i][j]+u1[i-1][j]) + f(x[i],y[j],tau2))*tau/2
            a[-1] = -1/h2
            b[-1] = 1/h2
            c[-1] = 0
            d[-1] = -np.sin(x[i])*np.sin(mu*t[k])

            result2 = slau([a,b,c,d])
            #print(result2)
            for j in range(n):
                u2[i][j] = result2[j]
                u2[0][j] = 0
                u2[-1][j] = np.sin(y[j])*np.sin(t[k])

        for i in range(n):
            u2[i][-1] = -np.sin(x[i])*np.sin(mu*t[k])*h2 + u2[i][-2]

        for i in range(n):
            for j in range(n):
                u[i][j][k] = u2[i][j]
    return u

def mdsh(n):
    u = np.zeros((n,n,n))
    for i in range(n):
        for j in range(n):
            u[i][j][0] = 0

    for k in range(1,n):
        u1 = np.zeros((n,n))
        u2 = np.zeros((n,n))
        tau2 = t[k-1] + tau/2
        for i in range(n):
            u1[i][0] = 0
            u1[0][i] = 0
            u1[-1][i] = np.sin(y[i])*np.sin(mu*tau2)
            u2[i][0] = 0

        for j in range(n-1):
            a = np.zeros(n)
            b = np.zeros(n)
            c = np.zeros(n)
            d = np.zeros(n)
            a[0] = 0
            b[0] = 1
            c[0] = 0
            d[0] = 0
```

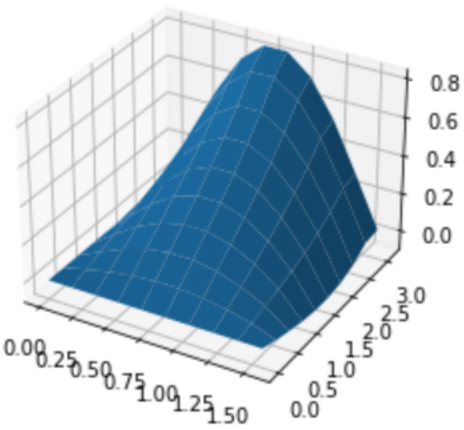
```
            for i in range(1,n-1):
                a[i] = acoef/h1/h1
                b[i] = -(1/tau + 2*acoef/h1/h1)
                c[i] = acoef/h1/h1
                d[i] = -(f(x[i],y[j],t[k-1])/2+u[i][j][k-1]/tau)
            a[-1] = 0
            b[-1] = 1
            c[-1] = 0
            d[-1] = np.sin(y[j])*np.sin(mu*tau2)

            result = slau([a,b,c,d])
            #print(result)
            for i in range(n):
                u1[i][j] = result[i]
                u1[i][-1] = -np.sin(x[i])*np.sin(mu*tau2)*h1 + u1[i][-2]


        for i in range(n-1):
            a = np.zeros(n)
            b = np.zeros(n)
            c = np.zeros(n)
            d = np.zeros(n)
            a[0] = 0
            b[0] = 1
            c[0] = 0
            d[0] = 0
            for j in range(1,n-1):
                a[j] = bcoef/h2/h2
                b[j] = -(1/tau + 2*bcoef/h2/h2)
                c[j] = bcoef/h2/h2
                d[j] = -(f(x[i],y[j],t[k])/2+u1[i][j]/tau)
            a[-1] = -1/h2
            b[-1] = 1/h2
            c[-1] = 0
            d[-1] = -np.sin(x[i])*np.sin(mu*t[k])

            result2 = slau([a,b,c,d])
            #print(result2)
            for j in range(n):
                u2[i][j] = result2[j]
                u2[0][j] = 0
                u2[-1][j] = np.sin(y[j])*np.sin(t[k])

        for i in range(n):
            u2[i][-1] = -np.sin(x[i])*np.sin(mu*t[k])*h2 + u2[i][-2]

        for i in range(n):
            for j in range(n):
                u[i][j][k] = u2[i][j]
    return u
```

Результат:

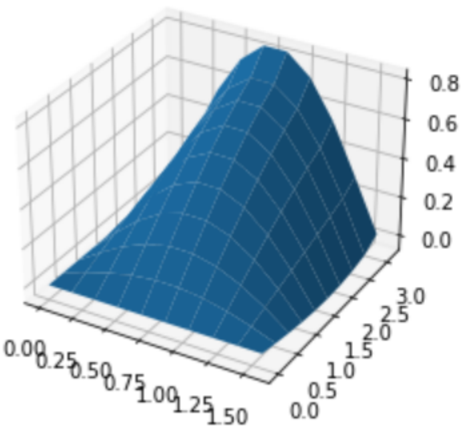
Аналитическое решение:



МПН:



МДШ:



Вывод:

В ходе проделанной работы, мною были проведены эксперименты с схемами переменных направлений и дробных шагов для решения двумерной начально-краевой задачи дифференциального уравнения параболического типа. Обе схемы дали не очень точное решение.