Московский авиационный институт
(национальный исследовательский университет)

Факультет информационных технологий и прикладной математики
Кафедра вычислительной математики и программирования

Отчет по лабораторной работе
№8 по курсу «Численные методы»

Дата: 10.12.2023

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:

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

$$u(0, y, t) = \cos(\mu_2 y)\exp(-(\mu_1^2 + \mu_2^2)at)$$

$$u(\frac{\pi}{2}\mu_1, y, t) = \cos(\mu_1\frac{\pi}{2})\cos(\mu_2 y)\exp(-(\mu_1^2 + \mu_2^2)at)$$

$$u(x, 0, t) = \cos(\mu_1 x)\exp(-(\mu_1^2 + \mu_2^2)at)$$

$$u(x, \frac{\pi}{2}\mu_2, t) = \cos(\mu_1 x)\cos(\mu_2\frac{\pi}{2})\exp(-(\mu_1^2 + \mu_2^2)at)$$

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

$$U = \cos(\mu_1 x)\cos(\mu_2 y)\exp(-(\mu_1^2 + \mu_2^2)at)$$

Решение: Из экономичных конечно-разностных схем, получивших наибольшее распространение, в данном разделе рассматриваются схема метода переменных направлений и схема метода дробных шагов.

В схеме метода переменных направлений (МПН), как и во всех методах расщепления, шаг по времени τ разбивается на число независимых пространственных переменных. На каждом дробном временном слое один из пространственных дифференциальных операторов аппроксимируется неявно, а остальные явно. На следующем дробном шаге следующий по порядку дифференциальный оператор аппроксимируется неявно, а остальные – явно и т.д. В двумерном случае схема метода переменных направлений имеет вид:

$$\frac{u_{ij}^{k+1/2}-u_{ij}^{k}}{\tau/2}=\frac{a}{h_1^2}\left(u_{i+1j}^{k+1/2}-2u_{ij}^{k+1/2}+u_{i-1j}^{k+1/2}\right)+\frac{a}{h_2^2}\left(u_{ij+1}^{k}-2u_{ij}^{k}+u_{ij-1}^{k}\right)+f_{ij}^{k+1/2},$$

$$(5.78)$$

$$\frac{u_{ij}^{k+1}-u_{ij}^{k+1/2}}{\tau/2}=\frac{a}{h_1^2}\left(u_{i+1j}^{k+1/2}-2u_{ij}^{k+1/2}+u_{i-1j}^{k+1/2}\right)+\frac{a}{h_2^2}\left(u_{ij+1}^{k+1}-2u_{ij}^{k+1}+u_{ij-1}^{k+1}\right)+f_{ij}^{k+1/2}. \quad (5.79)$$

В двумерном случае схема МПН абсолютна устойчива. К достоинствам метода переменных направлений можно отнести высокую точность, поскольку метод имеет второй порядок точности по времени. К недостаткам можно отнести условную устойчивость при числе пространственных переменных больше двух. Кроме этого, МПН условно устойчив в задачах со смешанными производными уже в двумерном случае.

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

$$\frac{u_{ij}^{k+1/2}-u_{ij}^{k}}{\tau}=\frac{a}{h_1^2}\left(u_{i+1j}^{k+1/2}-2u_{ij}^{k+1/2}+u_{i-1j}^{k+1/2}\right)+\frac{f_{ij}^{k}}{2},$$

$$\frac{u_{ij}^{k+1}-u_{ij}^{k+1/2}}{\tau}=\frac{a}{h_2^2}\left(u_{i\,j+1}^{k+1}-2u_{ij}^{k+1}+u_{i\,j-1}^{k+1}\right)+\frac{f_{ij}^{k+1}}{2}.$$

К достоинствам схемы МДШ можно отнести простоту в алгоритмизации и программировании и абсолютную устойчивость с большим запасом устойчивости даже для задач, содержащих смешанные производные. К недостаткам МДШ относятся следующие: на каждом дробном шаге достигается частичная аппроксимация, полная аппроксимация достигается на последнем дробном шаге, т.е. имеет место суммарная аппроксимация; схема имеет первый порядок точности по времени.

Код МПН:

```
private double[,,] MPN()
{
    double[,,] u = new double[Nx + 1, Ny + 1, K +
    1];double[] b = new double[Nx + 1];
    double[] a = new
    double[Nx];double[] c =
    new double[Nx];
```

```
            double[] d = new double[Nx + 1];
            double[,] u1 = new double[Nx + 1, Ny +
            1];double[,] u2 = new double[Nx + 1,
            Ny + 1];double[] x;
            double sigx = this.a * tau / (2 * Math.Pow(hx,
            2)); double  sigy  =  this.a  *  tau  /  (2  *
            Math.Pow(hy, 2));for (int i = 0; i <= Nx; i++)
                for  (int  j  =  0;  j  <=  Ny;  j++)  u[i,  j,  0]  =  Math.Cos(nu1  *  hx  *  i)  *
                Math.Cos(nu2 *
hy *
j);         for (int k = 1; k <= K; k++)
            {

                double t = k * tau - tau / 2;
                for (int j = 1; j <= Ny - 1; j++)
                {
                    for (int i = 0; i <= Nx - 1; i++)
                    {
                        a[i] = -sigx; b[i] = 1 + 2 * sigx; c[i] = -sigx; d[i] = sigy * (u[i, j
                        + 1,
k - 1] - 2 * u[i, j, k - 1] + u[i, j - 1, k - 1]) + u[i, j, k - 1];
                    }
                    b[0] = 1; c[0] = 0; d[0] = Phi1(hy * j, t);
                    a[Nx - 1] = 0; b[Nx] = 1; d[Nx] = Phi2(hy * j,
                    t);x = Progon(a, b, c, d).ToArray();
                    for (int i = 0; i <= Nx; i++)
                    {
                        u1[i, j] = x[i];
                        u1[i, 0] = Phi3(hx * i, t);
                        u1[i, Ny] = Phi4(hx * i, t);
                    }
                }
                for (int j = 0; j <= Ny; j++)
                {
                    u1[0, j] = Phi1(hy * j, t);
                    u1[Nx, j] = Phi2(hy * j, t);
                }
                for (int i = 1; i <= Nx - 1; i++)
                {
                    for (int j = 0; j <= Ny - 1; j++)
                    {
                        a[j] = -sigy; b[j] = 1 + 2 * sigy; c[j] = -sigy; d[j] = sigx * (u1[i +
                        1, j]
- 2 * u1[i, j] + u1[i - 1, j]) + u1[i, j];
                    }
                    b[0] = 1; c[0] = 0; d[0] = Phi3(hx * i, k * tau);
                    a[Ny - 1] = 0; b[Ny] = 1; d[Ny] = Phi4(hx * i, k *
                    tau);x = Progon(a, b, c, d).ToArray();
                    for (int j = 0; j <= Ny; j++)
                    {
                        u2[i, j] = x[j];
                        u2[0, j] = Phi1(hy * j, k * tau);
                        u2[Nx, j] = Phi2(hy * j, k * tau);
                    }
                }
                for (int i = 0; i <= Nx; i++)
                {
                    u1[i, 0] = Phi3(hx * i, k * tau);
                    u1[i, Ny] = Phi4(hx * i, k * tau);
                }
                for (int i = 0; i <= Nx; i++)
                    for (int j = 0; j <= Ny; j++) u[i, j, k] = u2[i, j];
            }
            return u;
        }
```

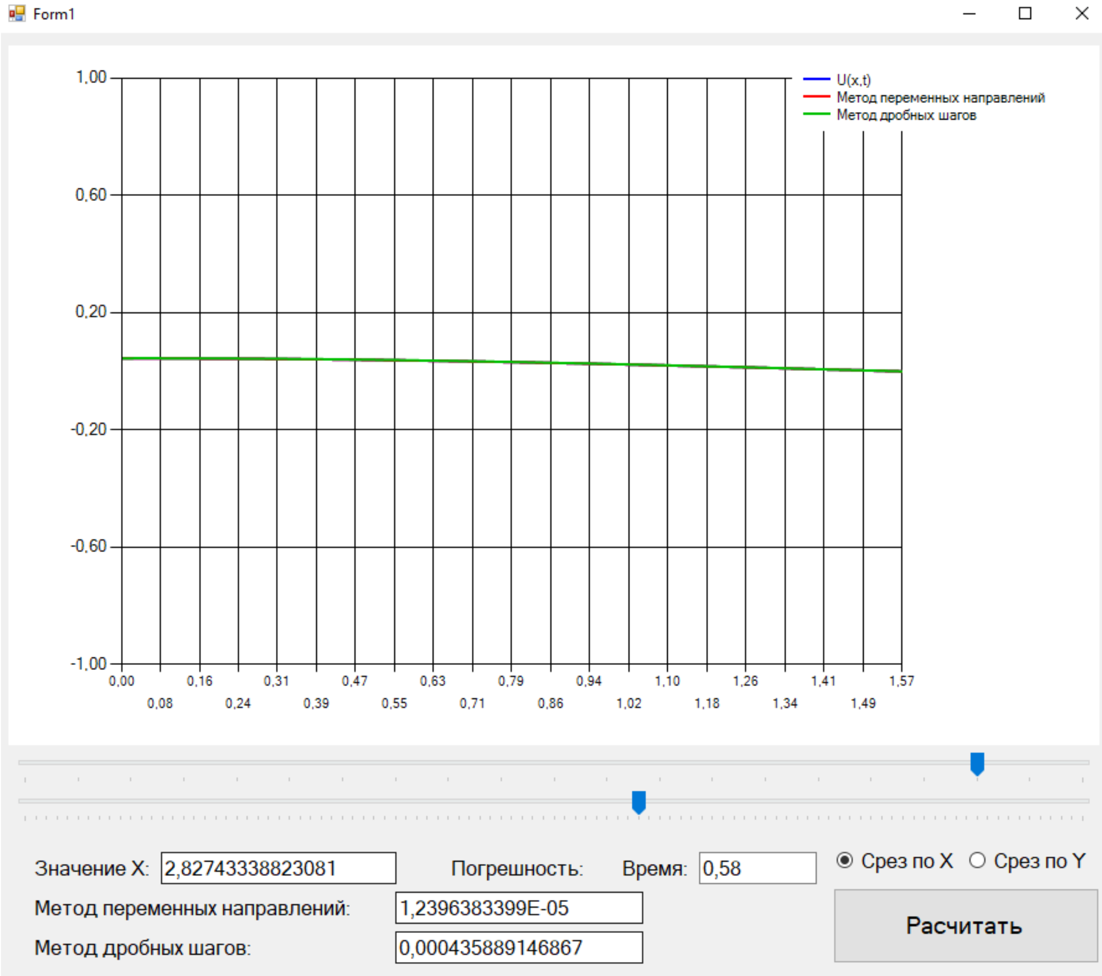
Код МДШ:

```
        private double[,,] MDS()
        {
```

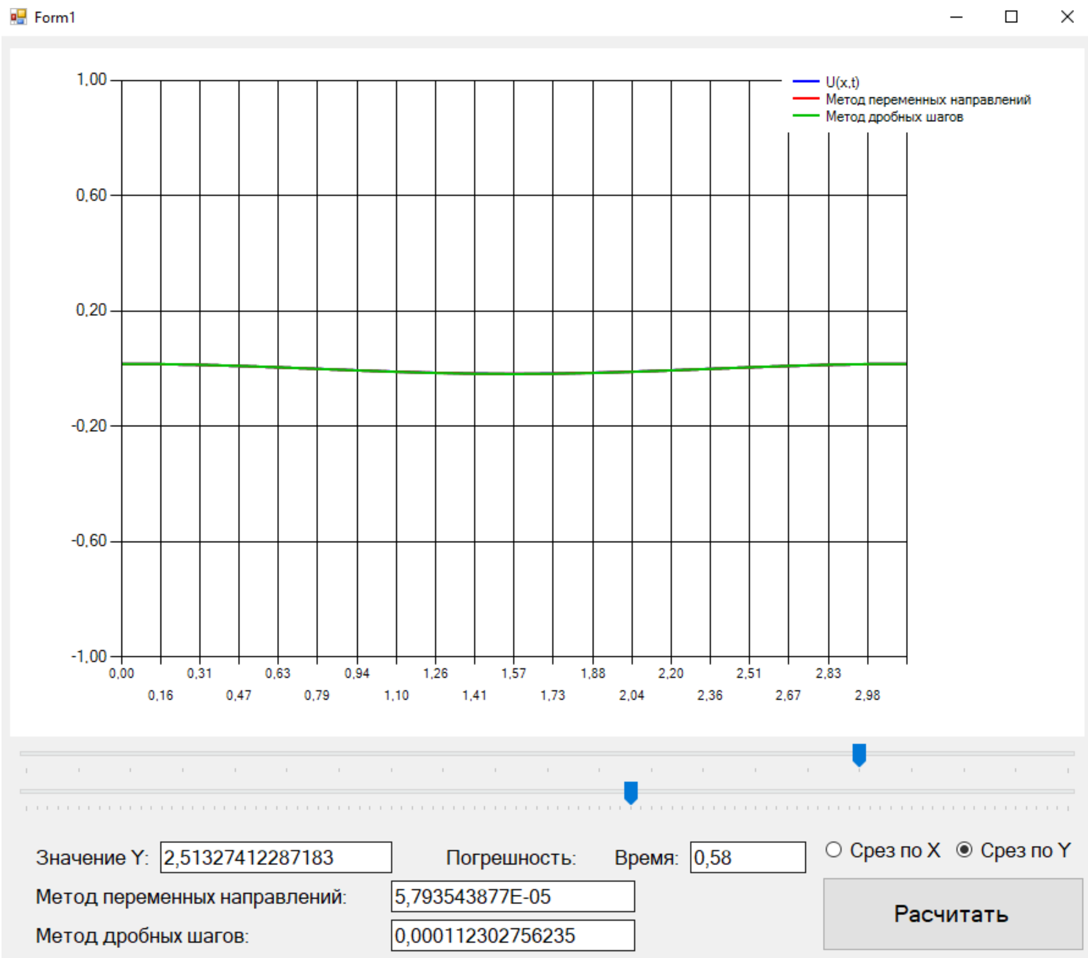
```
double[,,] u = new double[Nx + 1, Ny + 1, K + 1];
```

```
double[] b = new double[Nx +
1];double[] a = new
double[Nx]; double[] c = new
double[Nx]; double[] d = new
double[Nx + 1];
double[,] u1 = new double[Nx + 1, Ny +
1]; double[,] u2 = new double[Nx + 1,
Ny + 1];double[] x;
double sigx = this.a * tau / Math.Pow(hx,
2); double sigy = this.a * tau /
Math.Pow(hy, 2);for (int i = 0; i <= Nx;
i++)
hy *     for (int j = 0; j <= Ny; j++) u[i, j, 0] = Math.Cos(nu1 * hx * i) *
j);      Math.Cos(nu2 *

for (int k = 1; k <= K; k++)
{

    double t = k * tau - tau / 2;
    for (int j = 1; j <= Ny - 1; j++)
    {
        for (int i = 0; i <= Nx - 1; i++)
        {
            a[i] = -sigx; b[i] = 1 + 2 * sigx; c[i] = -sigx; d[i] = u[i, j, k - 1];
        }
        b[0] = 1; c[0] = 0; d[0] = Phi1(hy * j, t);
        a[Nx - 1] = 0; b[Nx] = 1; d[Nx] = Phi2(hy * j,
        t);x = Progon(a, b, c, d).ToArray();
        for (int i = 0; i <= Nx; i++)
        {
            u1[i, j] = x[i];
            u1[i, 0] = Phi3(hx * i, t);
            u1[i, Ny] = Phi4(hx * i, t);
        }
    }
    for (int j = 0; j <= Ny; j++)
    {
        u1[0, j] = Phi1(hy * j, t);
        u1[Nx, j] = Phi2(hy * j, t);
    }
    for (int i = 1; i <= Nx - 1; i++)
    {
        for (int j = 0; j <= Ny - 1; j++)
        {
            a[j] = -sigy; b[j] = 1 + 2 * sigy; c[j] = -sigy; d[j] = u1[i, j];
        }
        b[0] = 1; c[0] = 0; d[0] = Phi3(hx * i, k * tau);
        a[Ny - 1] = 0; b[Ny] = 1; d[Ny] = Phi4(hx * i, k *
        tau);x = Progon(a, b, c, d).ToArray();
        for (int j = 0; j <= Ny; j++)
        {
            u2[i, j] = x[j];
            u2[0, j] = Phi1(hy * j, k * tau);
            u2[Nx, j] = Phi2(hy * j, k * tau);
        }
    }
    for (int i = 0; i <= Nx; i++)
    {
        u1[i, 0] = Phi3(hx * i, k * tau);
        u1[i, Ny] = Phi4(hx * i, k * tau);
    }
    for (int i = 0; i <= Nx; i++)
        for (int j = 0; j <= Ny; j++) u[i, j, k] = u2[i, j];
}
return u;
}
```

Результаты:



Form1
U(x,t)
Метод переменных направлений
Метод дробных шагов
1,00
0,60
0,20
-0,20
-0,60
-1,00
0,00 0,08 0,16 0,24 0,31 0,39 0,47 0,55 0,63 0,71 0,79 0,86 0,94 1,02 1,10 1,18 1,26 1,34 1,41 1,49 1,57
Значение X: 2,82743338823081
Погрешность:
Время: 0,58
Срез по X
Срез по Y
Метод переменных направлений: 1,2396383399E-05
Метод дробных шагов: 0,000435889146867
Расчитать



Form1
U(x,t)
Метод переменных направлений
Метод дробных шагов
1,00
0,60
0,20
-0,20
-0,60
-1,00
0,00 0,16 0,31 0,47 0,63 0,79 0,94 1,10 1,26 1,41 1,57 1,73 1,88 2,04 2,20 2,36 2,51 2,67 2,83 2,98
Значение Y: 2,51327412287183
Погрешность:
Время: 0,58
Срез по X
Срез по Y
Метод переменных направлений: 5,793543877E-05
Метод дробных шагов: 0,000112302756235
Расчитать

Вывод: Мной было реализовано решение краевой задачи для дифференциального уравнения параболического типа 2D, с использованием схемы переменных направлений и дробных шагов, а также вычислена погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением.