Отчет по вычислительному практикуму

Шилов Максим

Дано уравнение теплопроводности:

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + f(x,t) \quad 0 \le x \le 1, \quad 0 \le t \le 1$$

$$u(x,0) = \mu(x), \quad u(0,t) = \mu_1(t), \quad u(1,t) = \mu_2(t)$$

Явная схема:

$$\frac{u_k^{j+1} - u_k^j}{\tau} = a\frac{u_{k+1}^j - 2u_k^j + u_{k-1}^j}{h^2} + f_k^j$$

$$u(x,t) = x^4 + tx + t^2 - te^x, \quad a = 0.014$$

Описание метода

Явная разностная схема реализуется при помощи пересчета верхнего слоя через нижний, значение в каждом неизвестном узле на (n+1)-м слое выражается через три уже вычисленных ранее значения с предыдущего n-го слоя.

Погрешность аппроксимации: O(τ, h^2).

Условие устойчивости: разностная схема устойчива, если при переходе от предыдущего шага к последующему ошибка не возрастает.

Для явной схемы:

$$\frac{a\tau}{h^2} \le \frac{1}{2} \Rightarrow h \ge \sqrt{2a\tau}$$

Имеем $u(x,t) = x^4 + tx + t^2 - te^x, \quad a = 0.014$
Выразим $f(x,t)$:
$\frac{\partial u}{\partial t} = x + 2t - e^x$
$\frac{\partial u}{\partial x} = 4x^3 + t - te^x$
$\frac{\partial^2 u}{\partial x^2} = 12x^2 - te^x$
$x + 2t - e^x = a(12x^2 - te^x) + f(x,t)$
Следовательно, $f(x,t) = x + 2t - e^x - a(12x^2 - te^x)$

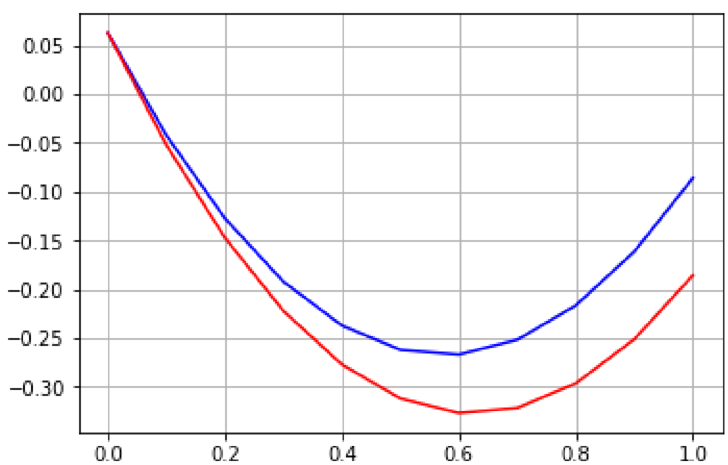
Схема реализации

Функция $f(x,t)$ задана, значения величин u^j считаем известными (как посчитанные на предыдущем шаге). Выразим u_k^{j+1}:
$u_k^{j+1} = u_k^j + \frac{a\tau}{h^2}(u_{k+1}^j - 2u_k^j + u_{k-1}^j) + \tau f_k^j$
Знаем значения на нулевом слое из начальных и краевых условий.

Ниже приведена таблица значений погрешности при различных h и τ:

$h\backslash\tau$	0.1	0.025	0.01	0.001
0.1	0.099631	0.024676	0.009706	0.000727
0.04	false	0.024935	0.009946	0.000955
0.01	false	false	false	0.000996
0.001	false	false	false	false

График приближенного и точного решения
для значений t от 0 до 1 с шагом 0.1 для x = 0.5
false - не удовлетворяет условию устойчивости



Код программы (Python 3.6)

```
import numpy as np
import matplotlib.pyplot as plt

def funk_u(t, x):
    return x**4+t*x+t**2-t*np.exp(x)

def funk_f(x, t):
    a = 0.014
    return x+2*t-np.exp(x)-a*(12*x**2-t*np.exp(x))

a = 0.014
n = 10
m = 10

tau = 1/n
h = 1/m

u = np.zeros((m+1, n+1))

# t = 0
for i in range(m+1):
    u[i][0] = funk_u(0, i*h)

# x = 0
for i in range(n+1):
    u[0][i] = funk_u(i*tau, 0)

# x = 1
for i in range(n+1):
    u[m][i] = funk_u(i*tau, 1)

for j in range(1, n+1):
    for i in range(1, m):
        u[i][j]=((a*tau)/h**2)*(u[i+1][j-1] - 2*u[i][j-1] + u[i-1][j-1] ) + \
        funk_f(i*h, (j-1)*tau)*tau + u[i][j-1]
```

```
# график приближенного и точного решения

t = np.arange(0, 1+tau, tau)
x = 0.5
y = x**4+t*x+t**2-t*np.exp(x)
fig = plt.figure()
plt.plot(t, y, color='blue')
plt.plot(t, u[5], color='red')
plt.grid(True)
plt.show()
```