Отчет по вычислительному практикуму

Шилов Максим

Дано уравнение переноса в виде:

$$\frac{\partial u}{\partial t} + C(x,t)\frac{\partial u}{\partial x} = 0 \quad 0 \le x \le 1, \quad 0 \le t \le 1$$
$$u(x,0) = \mu(x), \quad u(0,t) = \mu_1(t)$$

Схема Лакса:

$$u_k^{j+1} = \frac{u_{k+1}^j + u_{k-1}^j}{2} - \frac{\tau C}{2h}(u_{k+1}^j - u_{k-1}^j) \qquad (1)$$

$$u(x,t) = x^3 - \frac{sin(2\pi t)}{2} + x - 3.5t$$

Описание метода

Погрешность аппроксимации $O(\tau + h)$ в точке (x_k, t_j) при условии $\tau = O(h)$.

Устойчивость условная: $\frac{|C|\tau}{h} \le 1$.

Схема реализации

Недотающее граничное условие можно вычислить с помощью многочлена Лагранжа:

$$u_M^{j+1} = 2u_{M-1}^{j+1} - u_{M-2}^{j+1}; \quad \text{(многочлен 1-й степени)}$$

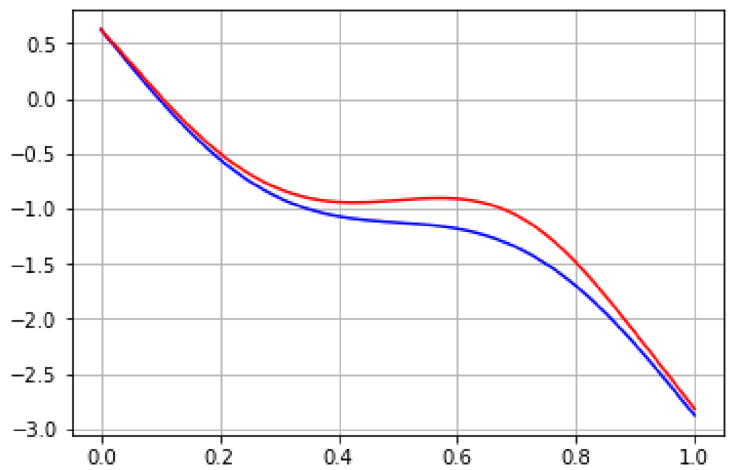
Значения величин u_{k-1}^j, u_{k+1}^j считаем известными, как посчитанные на предыдущем шаге. u_k^{j+1} - выражено в схеме. Значения на нулевом слое находим из начальных условий.

Ниже приведена таблица значений погрешности при различных h и τ:

$h \backslash \tau$	$1/2^4$	$1/2^5$	$1/2^6$	$1/2^7$	$1/2^8$	$1/2^9$
$1/2$	0.484428	0.350003	0.360096	0.367577	0.371292	0.373146
$1/2^2$	false	0.468720	0.542660	0.615828	0.663604	0.690586
$1/2^3$	false	false	0.454929	0.571145	0.705317	0.810278
$1/2^4$	false	false	false	0.362881	0.494208	0.639935
$1/2^5$	false	false	false	false	0.250219	0.376269
$1/2^6$	false	false	false	false	false	0.155282

В обозначениях 'false' - не удовлетворяет условию устойчивости.

График
приближенного и точного решения для значений t от 0 до 1
с шагом $1/2^7$ для x = 0.5



Код программы (Python 3.6)

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

def funk_u(x, t):
    return x**3 - np.sin(np.pi*2*t)/2 + x - 3.5*t

def funk_C(x, t):
    return (np.pi*np.cos(2*np.pi*t) + 3.5)/(3*x**2 + 1)

n = 128   # tau
m = 16    # h
tau = 1/n
h = 1/m

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

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

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

# схема
if (np.pi + 3.5)*tau/h <= 1:
    for j in range(0, n):
        for k in range(1, m):
            u[k][j+1] = (u[k+1][j] + u[k-1][j])/2 - \
                tau*funk_C(k*h, j*tau)/(2*h)*(u[k+1][j] - u[k-1][j])
        # многочлен Лагранжа 1й степени
        u[m][j+1] = 2*u[m-1][j+1] - u[m-2][j+1]

else:
    print('Схема неустойчива')

# график
t = np.arange(0, 1+tau, tau)
x = 0.5
u_real_x = x**3 - np.sin(np.pi*2*t)/2 + x - 3.5*t
fig = plt.figure()
plt.plot(t, u_real_x, color='blue')
plt.plot(t, u[int(m*x)], color='red')
plt.grid(True)
plt.show()
```