Московский авиационный институт
(Национальный исследовательский университет)
Факультет прикладной математики и физики
Кафедра вычислительной математики и программирования

Лабораторная работа № 5

по курсу «Численные методы»

Студент: Аксенов А. Е.
Группа: М80-408Б-20
Преподаватель: Пивоваров Д. Е.
Оценка:

Москва, 2023

Лабораторная №5

Задание

Используя явную и неявную конечно-разностные схемы, а также схему Кранка - Николсона, решить начально-краевую задачу для дифференциального уравнения параболического типа. 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 1

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

$$u(0,t) = 0,$$
$$u(1,t) = 0,$$

$$u(x,0) = \sin(2\pi x).$$

Аналитическое решение: $U(x,t) = \exp(-4\pi^2 at)\sin(2\pi x)$.

Теория

Классическим примером уравнения параболического типа является уравнение теплопроводности (диффузии). В одномерном по пространству случае однородное (без источников энергии) уравнение теплопроводности имеет вид

$$\frac{\partial u}{\partial t} = a^2\frac{\partial^2 u}{\partial x^2}, \qquad 0 < x < l, t > 0. \qquad (1)$$

Граничные условия первого рода:

$$u(0,t) = \varphi_0(t), x = 0, t > 0$$

$$u(l,t) = \varphi_l(t), x = l, t > 0 \qquad (2)$$

Начальные условия:

$$u(x,0) = \psi(t), 0 \le x \le l, t = 0 \qquad (4)$$

Задача (1) - (4) называется первой начально-краевой задачей для уравнения теплопроводности.

Если же на границах $x = 0, x = l$ заданы значения производных искомых функций по пространственной переменной

$$\frac{\partial u(0,t)}{\partial t} = \varphi_0(t), x = 0, t > 0 \qquad (5)$$

$$\frac{\partial u(l,t)}{\partial t} = \varphi_l(t), x = l, t > 0 \qquad (6)$$

То есть граничные условия второго рода, то задачу (1), (5), (6), (4) называют второй начально-краевой задачей для уравнения теплопроводности (1).

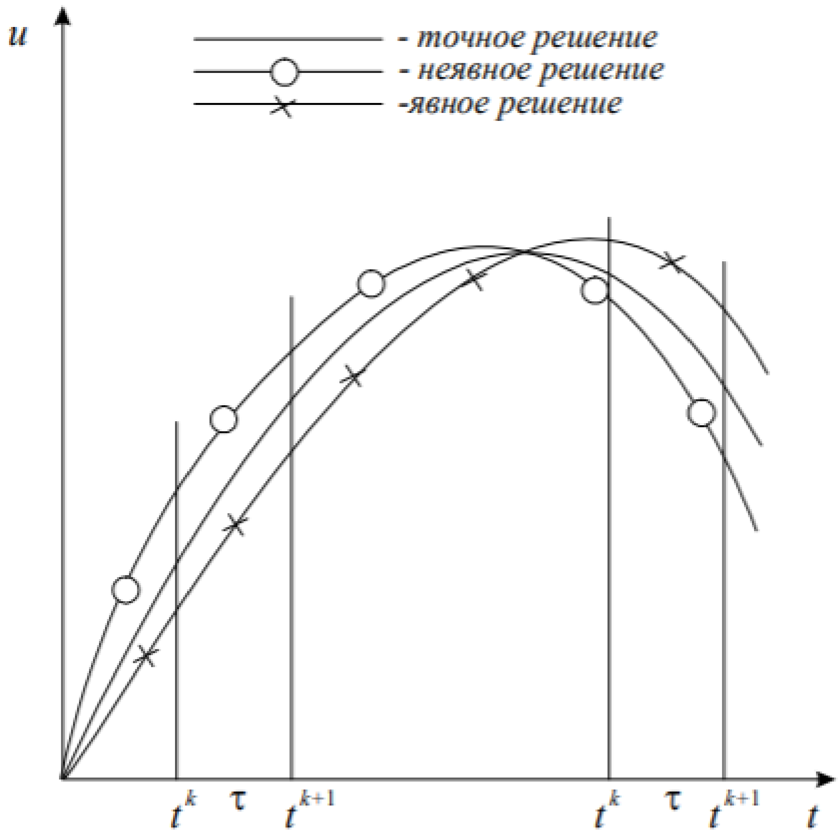
Если на границах заданы линейные комбинации искомой функции и ее производной по пространственной переменной.

$$\alpha \frac{\partial u(0,t)}{\partial x} + \beta u(0,t) = \varphi_0(t), \qquad x = 0, \qquad t > 0 \qquad (7)$$

$$\gamma \frac{\partial u(0,t)}{\partial x} + \delta u(0,t) = \varphi_l(t), \qquad x = l, \qquad t > 0 \qquad (8)$$

То есть граничные условия третьего рода, то задачу (1), (7), (8), (4) называют третьей начально-краевой задачей для уравнения теплопроводности (1).

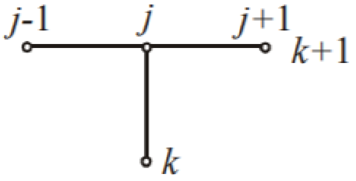
Рассмотрим три схемы для решения задачи:



Неявная схема

$$\frac{u_j^{k+1}-u_j^k}{\tau}=a^2\frac{u_{j+1}^{k+1}-2u_j^{k+1}+u_{j-1}^{k+1}}{h^2}+O(\tau+h^2),\quad j=\overline{1,N-1},\quad k=\overline{0,K-1},$$

$$u_0^{k+1}=\varphi_0(t^{k+1}),\quad u_N^{k+1}=\varphi_l(t^{k+1}),\quad k=\overline{0,K-1};\quad u_j^0=\psi(x_j),\quad j=\overline{0,N}.$$

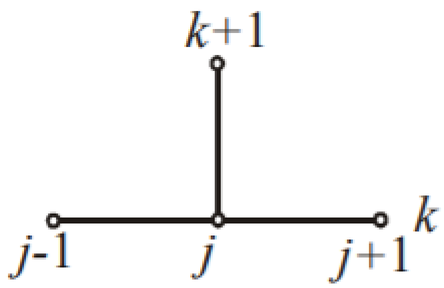


шаблон неявной схемы

Явная схема

$$\frac{u_j^{k+1}-u_j^k}{\tau}=a^2\frac{u_{j+1}^{k}-2u_j^{k}+u_{j-1}^{k}}{h^2}+O(\tau+h^2),\quad j=\overline{1,N-1},\quad k=\overline{0,K-1},$$

$$u_0^{k}=\varphi_0(t^{k}),\quad u_N^{k}=\varphi_l(t^{k}),\quad k=\overline{0,K};\quad u_j^0=\psi(x_j),\quad j=\overline{0,N},$$



шаблон явной схемы

Схема Кранка-Николсона

$$\frac{u_j^{k+1}-u_j^k}{\tau}=\theta a^2\frac{u_{j+1}^{k+1}-2u_j^{k+1}+u_{j-1}^{k+1}}{h^2}+(1-\theta)a^2\frac{u_{j+1}^{k}-2u_j^{k}+u_{j-1}^{k}}{h^2},$$

где θ - вес неявной части конечно-разностной схемы, $1-\theta$ - вес для явной части, причем $0\le\theta\le1$. При $\theta=1$ имеем полностью неявную схему, при $\theta=0$ - полностью явную схему, и при $\theta=1/2$ - схему Кранка-Николсона. Для схемы Кранка-Николсона ($\theta=1/2$) порядок аппроксимации составляет $O(\tau^2+h^2)$, т.е. на один порядок по времени выше, чем обычные явная или неявная схемы. Неявно-явная схема с весами (5.20) абсолютно устойчива при $1/2 \le \theta \le 1$ и условно устойчива с условием $\sigma \le \frac{1}{2}$ при $0 \le \theta \le \frac{1}{2}$

Явный метод

```
l_start = 0.0
l_finish = 1.0
t_start = 0.0
t_finish = 1.0

def explicit(N = 30, K = 300, a = 1.0, theta = 0.0):
    N = N - 1
    K = K - 1
    h = (l_finish - l_start) / N
    tau = (t_finish - t_start) / K
    sigma = a * a * tau / (h * h)
    X = []
    Y = []
    ans = []

    x = list(np.arange(l_start, l_finish + h/2, h))
    last_line = list(map(init_cond, x))
    ans.append(list(last_line))
    X.append(x)
    Y.append([0.0 for _ in x])

    for t in np.arange(t_start + tau, t_finish + tau/2, tau):
        ans_line = [None for _ in last_line]
        for i in range(1, len(x) - 1):
            ans_line[i] = sigma * last_line[i - 1]
            ans_line[i] += (1 - 2 * sigma) * last_line[i]
            ans_line[i] += sigma * last_line[i + 1]
        ans_line[0] = cond_1(t)
        ans_line[-1] = cond_2(t)

        ans.append(ans_line)
        X.append(x)
        Y.append([t for _ in x])
        last_line = ans[-1]

    return X, Y, ans
```

Неявный и KN

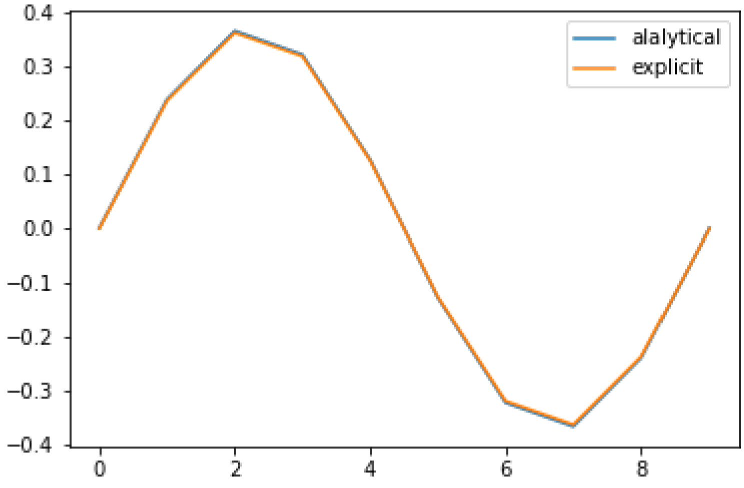
```
l_start = 0.0
l_finish = 1.0
t_start = 0.0
t_finish = 1.0

def unexplicit(N = 30, K = 300, a = 1.0, theta = 0.5):
    N = N - 1
    K = K - 1
    h = (l_finish - l_start) / N
    tau = (t_finish - t_start) / K
    sigma = a * a * tau / (h * h)
    X = []
    Y = []
    ans = []

    x = list(np.arange(l_start, l_finish + h/2, h))
    last_line = list(map(init_cond, x))
    ans.append(list(last_line))
    X.append(x)
    Y.append([0.0 for _ in x])

    for t in np.arange(t_start + tau, t_finish + tau/2, tau):
        coeff_a = sigma * theta
        coeff_b = -1 - 2 * sigma * theta

        A = [
            (coeff_a, coeff_b, coeff_a)
            for _ in range(1, len(x)-1)
        ]
        coeff_d = [
            -(last_line[i] + (1 - theta) * sigma* (last_line[i-1] - 2*last_line[i] + last_line[i+1]))
            for i in range(1, len(x)-1)
        ]
        A.insert(0, (0,0,cond_1(0)))
        coeff_d.insert(0, cond_1(0))

        A.append((0,0,cond_2(0)))
        coeff_d.append(cond_2(0))

        ans.append(race_method(A, coeff_d))
        X.append(x)
        Y.append([t for _ in x])
        last_line = ans[-1]

    return X, Y, ans
```

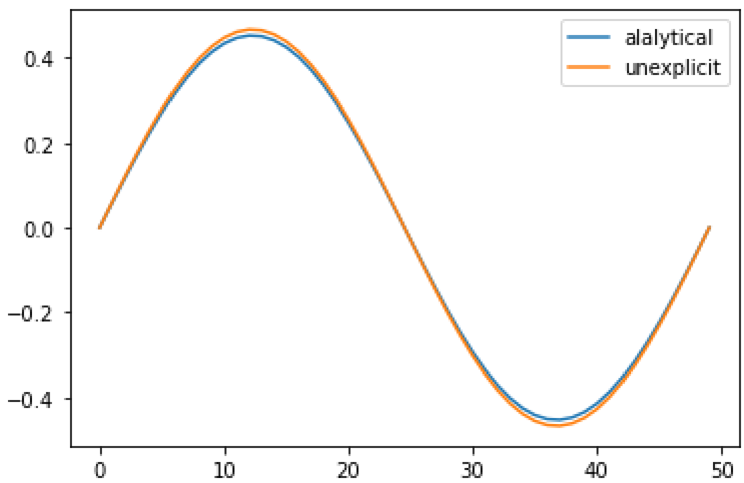
Прогонка

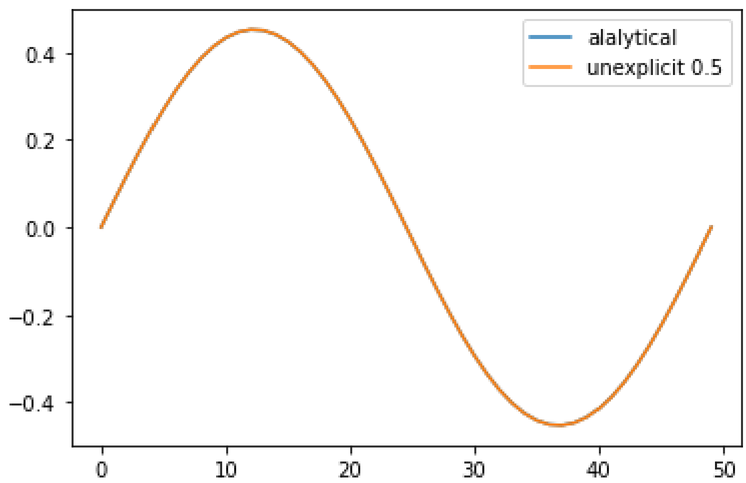
```
def race_method(A, b):
    P = [-item[2] for item in A]
    Q = [item for item in b]
    eps = 1e-10 if abs(A[0][1]) < 1e-8 else 0
    P[0] /= (A[0][1] + eps)
    Q[0] /= (A[0][1] + eps)
    for i in range(1, len(b)):
        z = (A[i][1] + A[i][0] * P[i-1])
        P[i] /= (z + eps)
        Q[i] -= A[i][0] * Q[i-1]
        Q[i] /= (z + eps)
    x = [item for item in Q]
    for i in range(len(x) - 2, -1, -1):
        x[i] += P[i] * x[i + 1]
    return x
```

Погрешности

Численные и аналитические кривые в последний момент времени







Вывод

В ходе выполнения лабораторной работы я познакомился со следующими схемами аппроксимации: 1) Явная 2) Неявная 3) Кранекера – Николсона. Также мной были построены графики зависимости ошибки от размера шага по пространству. С увеличением размера шага растет ошибка, что говорит о том, что алгоритмы сходятся. Также я отрисовал графики аналитического и численного решения в последний момент времени. Графики оказались близки друг к другу. Данный факт говорит о том, что алгоритмы сходятся.