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

Лабораторная работа № 6

по курсу «Численные методы»

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

Москва, 2023

Лабораторная №6

Задание

Используя явную схему крест и неявную схему, решить начально-краевую задачу для дифференциального уравнения гиперболического типа. Аппроксимацию второго начального условия произвести с первым и со вторым порядком. 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

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

$u(0,t) = -\sin(at),$
$u(\pi,t) = \sin(at),$

$u(x,0) = \sin x\,,$

$u_t(x,0) = -a\cos x\,.$

Аналитическое решение: $U(x,t) = \sin(x - at)$

Теория

Классическим примером уравнения гиперболического типа является волновое уравнение, которое в области имеет вид: $0 < x < l, t > 0$

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

Первая начально-краевая задача для волнового уравнения имеет вид:

$$\begin{cases} \dfrac{\partial^2 u}{\partial^2 t} = a^2 \dfrac{\partial^2 u}{\partial x^2}, 0 < x < l, t = 0 & (1) \\ u(0,t) = \varphi_0(t), x = 0, t > 0 & (2) \\ u(l,t) = \varphi_l(t), x = l, t > 0 & (3) \\ u(x,0) = \psi_1(x), 0 \le x \le l, t = 0 & (4) \\ \dfrac{\partial u(x,0)}{\partial t} = \psi_2(x), 0 \le x \le l, t = 0 & (5) \end{cases}$$

Вторая начально-краевая задача для волнового уравнения имеет вид:

$$\begin{cases} \dfrac{\partial^2 u}{\partial^2 t} = a^2 \dfrac{\partial^2 u}{\partial x^2}, 0 < x < l, t = 0 \\ \dfrac{\partial u(0,t)}{\partial x} = \varphi_0(t), x = 0, t > 0 \\ \dfrac{\partial u(l,t)}{\partial x} = \varphi_l(t), x = l, t > 0 \\ u(x,0) = \psi_1(x), 0 \le x \le l, t = 0 \\ \dfrac{\partial u(x,0)}{\partial t} = \psi_2(x), 0 \le x \le l, t = 0 \end{cases}$$

Третья начально-краевая задача для волнового уравнения имеет вид:

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

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

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

$$u(x,0) = \psi_1(x), 0 \le x \le l, t = 0$$

$$\frac{\partial u(x,0)}{\partial t} = \psi_2(x), 0 \le x \le l, t = 0$$

Разностные схемы для аппроксимации

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

Схема является явной. С ее помощью решение $u_j^{k+1}, j = 1, \ldots, N-1$, определяется сразу, поскольку значения сеточных функции , на нижних временных слоях должны быть известны. В соответствии с шаблоном для этой схемы порядок аппроксимации равен двум, как по пространственной, так и по временной переменной. При этом явная конечно-разностная схема для волнового уравнения условно устойчива с условием $\sigma = \frac{a^2\tau^2}{h^2} < 1$, накладываемым на сеточные характеристики τ и h .

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

Данная является неявной схемой и обладает абсолютной устойчивостью. Ее можно свести к СЛАУ с трехдиагональной матрицей, решаемой методом прогонки.

Ключевые моменты программы

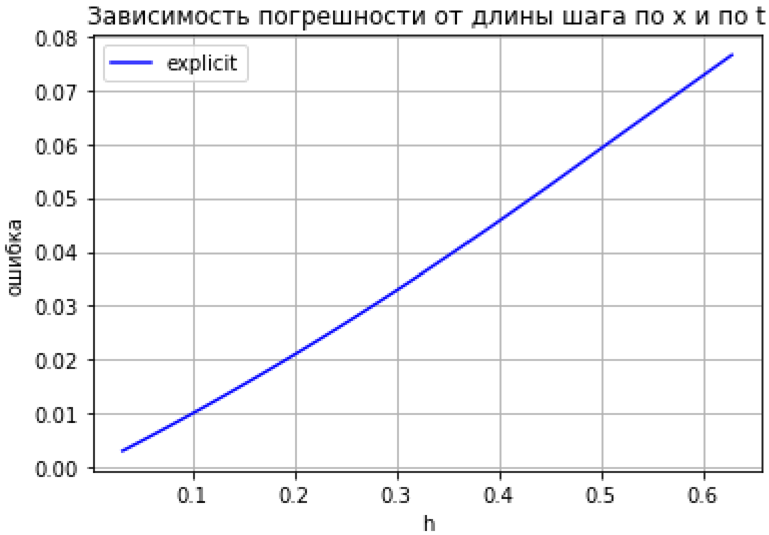
Явная схема

```
l_start = 0.0
l_finish = math.pi
t_start = 0.0
t_finish = 1.0

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

    x = list(np.linspace(l_start, l_finish, N))
    ans.append(list(map(init_cond_1, x, len(x)*[a])))
    ans.append(list(map(lambda lam: init_cond_1(lam, a) + tau*init_cond_2(lam, a) + a*a*tau*tau*diff2_init_cond(lam, a)/2, x
    X = [x, x]
    T.append([0.0 for _ in x])
    T.append([tau for _ in x])

    for t in np.linspace(t_start + 2*tau, t_finish, K):
        ans_last_line_1 = ans[-1]
        ans_last_line_2 = ans[-2]
        ans_line = [None for _ in ans_last_line_1]
        for i in range(1, len(x) - 1):
            ans_line[i] = 2 * (1 - sigma) * ans_last_line_1[i]
            ans_line[i] += sigma * (ans_last_line_1[i+1] + ans_last_line_1[i-1])
            ans_line[i] -= ans_last_line_2[i]
        ans_line[0] = cond_1(t, a)
        ans_line[-1] = cond_2(t, a)

        ans.append(ans_line)
        X.append(x)
        T.append([t for _ in x])

    return X, T, ans
```

Неявная схема

```
l_start = 0.0
l_finish = math.pi
t_start = 0.0
t_finish = 1.0

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

    x = list(np.linspace(l_start, l_finish, N+1))
    ans.append(list(map(init_cond_1, x, len(x)*[a])))
    ans.append(list(map(lambda lam: init_cond_1(lam, a) + tau*init_cond_2(lam, a) + a*a*tau*tau*diff2_init_cond(lam, a)/2, x
    X = [x, x]
    T.append([0.0 for _ in x])
    T.append([tau for _ in x])

    for t in np.linspace(t_start + 2*tau, t_finish, K):
        ans_last_line_1 = ans[-1]
        ans_last_line_2 = ans[-2]
        coeff_a = 1
        coeff_b = -2 - 1 / sigma

        A = [
            (coeff_a, coeff_b, coeff_a)
            for _ in range(1, len(x)-1)
        ]
        coeff_d = [
            (ans_last_line_2[i] - 2 * ans_last_line_1[i]) / sigma
            for i in range(1, len(x) - 1)
        ]
        A.insert(0, (0, coeff_b, coeff_a))
        coeff_d.insert(0, (cond_1(t-tau, a) - 2*cond_1(t, a)) / sigma)

        A.append((coeff_a, coeff_b, 0))
        coeff_d.append((cond_2(t-tau, a) - 2*cond_2(t, a)) / sigma)

        ans.append(race_method(A, coeff_d))
        X.append(x)
        T.append([t for _ in x])

    return X, T, ans
```

<u>Ошибки</u>

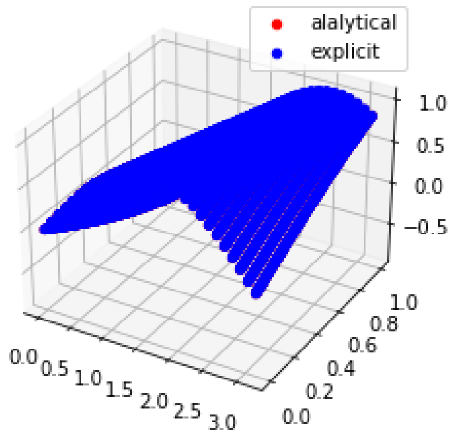

Зависимость погрешности от длины шага по x и по t
explicit
ошибка
h
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07
0.08
0.1
0.2
0.3
0.4
0.5
0.6


Зависимость погрешности от длины шага по x и по t
implicit
ошибка
h
0.000
0.001
0.002
0.003
0.004
0.005
0.006
0.007
0.008
0.1
0.2
0.3
0.4
0.5
0.6

Численные и аналитические решения

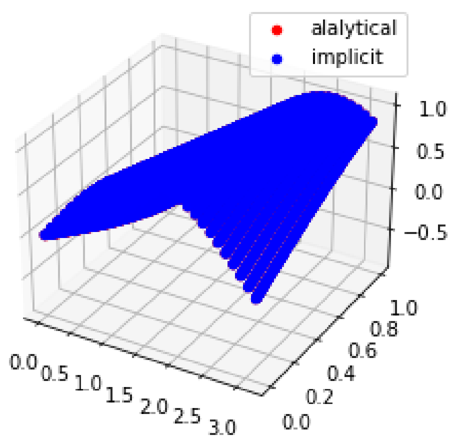
```
R3_plot(explicit, "explicit")
```

```
RMSE =  0.010940135873761359
```




```
R3_plot(implicit, "implicit")
```

```
RMSE =  0.0018102814339126048
```



Вывод

Были реализованы методы решения задачи гиперболического типа:

· Явная схема

· Неявная схема

Также были построены графики зависимости ошибки от размера шага по пространству. График возрастает, что говорит о том, что алгоритм сходится. Также мной были отрисованы графики аналитического и численного решения. Графики почти совпадают. Данный факт говорит о том, что алгоритмы сходятся.