Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

« Российский химико-технологический университет имени Д.И. Менделеева »

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

Вариант 22

Выполнил студент группы КС-36: **Золотухин Андрей Александрович**
Ссылка на репозиторий: https://github.com/CorgiPuppy/num-methods-eq-math-phys-chem-labs
Принял: Лебедев Данила Александрович
Дата сдачи: 02.04.25

Москва

2025

Оглавление

Описание задачи

Вариант	Уравнение	Интервалы переменных	Начальные и граничные условия
22	$\frac{\partial u}{\partial t} = \frac{\partial^2 u}{\partial x^2}$	x in [0, 1] t in [0, 1]	$u(t=0,x) = e^x$ $u(t,x=0) = e^t$ $u(t,x=1) = e^{t+1}$

Для заданного уравнения:

1. записать неявную разностную схему;
2. определить порядок аппроксимации разностной схемы;
3. доказать абсолютную устойчивость разностной схемы (с помощью метода гармоник);
4. привести схему к виду, удобному для использования метода прогонки;
5. проверить сходимость прогонки;
6. найти α_1, β_1, u_N^{n+1}
7. записать рекуррентное прогоночное соотношение;
8. составить алгоритм (блок-схему) расчёта;
9. построить программу на любом удобном языке программирования;
10. провести численный расчёт с использованием различных значений Δt*(0.1, 0.01, 0.001)*, $h = 0.1$
11. составить отчёт о проделанной работе;
12. сравнить результаты расчётов заданий №1 и №2 друг с другом, а также с истинными значениями функции u и в соответствующих точках разностной сетки (*истинное решение уравнения будет выдано преподавателем после выполнения расчётов по явной и неявной разностным схемам).*

Выполнение задачи

Задание 1

Записать неявную разностную схему:

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{h^2}. \tag{1}$$

В записанной разностной схеме уравнения 1 аппроксимация второй производной функции *u(t, x)* по координате рассматривается на *n+1*-м шаге по времени, т.е. относительно точки t^{n+1}, для которой рассматривается аппроксимация всего уравнения. Такая разностная схема называется **неявной**.

Задание 2

Определить порядок аппроксимации разностной схемы уравнения 1: Для этого запишу разложение значений u_j^{n+1}, u_{j+1}^{n+1}, u_{j-1}^{n+1} в ряд Тейлора относительно точки (t^{n+1}, x_j):

$$u_j^n = u_j^{n+1} - \frac{\partial u}{\partial t}|_j^{n+1}\ \Delta t + \frac{\partial^2 u}{\partial t^2}|_j^{n+1}\ \frac{(\Delta t)^2}{2!} - \frac{\partial^3 u}{\partial t^3}|_j^{n+1}\ \frac{(\Delta t)^3}{3!} + ..., \tag{2}$$

$$u_{j+1}^{n+1} = u_j^{n+1} + \frac{\partial u}{\partial x}|_j^{n+1}\ h + \frac{\partial^2 u}{\partial x^2}|_j^{n+1}\ \frac{h^2}{2!} + \frac{\partial^3 u}{\partial x^3}|_j^{n+1}\ \frac{h^3}{3} + \frac{\partial^4 u}{\partial x^4}|_j^{n+1}\ \frac{h^4}{4} + ..., \tag{3}$$

$$u_{j-1}^{n+1} = u_j^{n+1} - \frac{\partial u}{\partial x}|_j^{n+1}\ h + \frac{\partial^2 u}{\partial x^2}|_j^{n+1}\ \frac{h^2}{2!} - \frac{\partial^3 u}{\partial x^3}|_j^{n+1}\ \frac{h^3}{3} + \frac{\partial^4 u}{\partial x^4}|_j^{n+1}\ \frac{h^4}{4} + ... \tag{4}$$

Подставляя зависимости уравнения 2 - уравнения 4 в разностную схему уравнения 1, получаю:

$$\frac{\partial u}{\partial t}|_j^{n+1} - \frac{\partial^2 u}{\partial t^2}|_j^{n+1}\ \frac{(\Delta t)}{2!} + \frac{\partial^3 u}{\partial t^3}|_j^{n+1}\ \frac{(\Delta t)^2}{3!} - ... = \frac{\partial^2 u}{\partial x^2}|_j^{n+1}\ \frac{h^2}{2!} + \frac{\partial^4 u}{\partial x^4}|_j^{n+1}\ \frac{h^4}{4} + ...$$

$$\Rightarrow \frac{\partial u}{\partial t}|_j^{n+1} + O(\Delta t) = \frac{\partial^2 u}{\partial x^2}|_j^{n+1} + O(h^2).$$

Таким образом, неявная разностная схема уравнения 1 аппроксимирует исходное дифференциальное уравнение с первым порядком по времени и со вторым порядком по координате, что записывается в следующем виде:

$$O(\Delta t) + O(h^2) \text{ или } O(\Delta t, h^2).$$

Задание 3

Доказать абсолютную устойчивость разностной схемы уравнения 1 (с помощью метода гармоник):

Представляю решение разностной схемы в виде гармоники:

$$u_j^n = \lambda^n e^{i\alpha j}. \tag{5}$$

Подставляя уравнения 5 в разностную схему уравнения 1, получаю:

$$\frac{\lambda^{n+1}e^{i\alpha j}-\lambda^n e^{i\alpha j}}{\Delta t}=\frac{\lambda^{n+1}e^{i\alpha(j+1)}-2\lambda^{n+1}e^{i\alpha j}+\lambda^{n+1}e^{i\alpha(j-1)}}{h^2}.$$

Упрощаю полученное выражение, деля левую и правую его части на $\lambda^n e^{i\alpha j}$:

$$\frac{\lambda-1}{\Delta t}=\frac{\lambda e^{i\alpha}-2\lambda+\lambda e^{-i\alpha}}{h^2}$$

Преобразую комплексные числа из экспоненциальной формы в тригонометрическую:

$$e^{\pm i\alpha}=\cos\alpha\pm i\sin\alpha \Rightarrow \frac{\lambda-1}{\Delta t}=\lambda\frac{2\cos\alpha-2}{h^2}.$$

Используя тригонометрические тождества

$$\cos\alpha=\cos^2\frac{\alpha}{2}-\sin^2\frac{\alpha}{2}=1-2\sin^2\frac{\alpha}{2},$$

получаю формулу, из которой затем выражаю λ:

$$\frac{\lambda-1}{\Delta t}=\frac{-4\lambda\sin^2\frac{\alpha}{2}}{h^2}\Rightarrow\lambda=\frac{1}{1+\frac{4\Delta t\sin^2\frac{\alpha}{2}}{h^2}}.$$

С учётом необходимого условия устойчивости разностных схем $|\lambda|\leq 1$ имею:

$$-1\leq\frac{1}{1+\frac{4\Delta t\sin^2\frac{\alpha}{2}}{h^2}}\leq 1.$$

В полученном двойном неравенстве левое и правое условие выполняются автоматически.

Для любых значений Δt и h неравенство выполняется. Следовательно, разностная схема **абсолютно устойчива**.

Задание 4

Привести схему к виду, удобному для использования метода прогонки:

Преобразую неявную разностную схемы уравнения 1, группируя в левой части члены, содержащие значение функции *u(t, x)* на *(n + 1)* шаге по времени, а в правой части - все остальные члены:

$$-\frac{\Delta t}{h^2}u_{j+1}^{n+1}+\left(1+2\frac{\Delta t}{h^2}\right)u_j^{n+1}-\frac{\Delta t}{h^2}u_{j-1}^{n+1}=u_j^n. \tag{6}$$

Введу следующие обозначения:

$$a_j=-\frac{\Delta t}{h^2};b_j=\left(1+2\frac{\Delta t}{h^2}\right);c_j=-\frac{\Delta t}{h^2};\xi_j^n=u_j^n. \tag{7}$$

С учётом обозначений уравнения 7 равенство уравнения 6 будет иметь вид:

$$a_j u_{j+1}^{n+1}+b_j u_j^{n+1}+c_j u_{j-1}^{n+1}=\xi_j^n$$

Данное преобразование называется *преобразованием неявной разностной схемы к виду, удобному для использования метода прогонки.*

Задание 5

Проверить сходимость прогонки:

Теорема. Достаточным условием сходимости метода прогонки к решению исходной дифференциальной задачи является выполнение неравенства:

$$|a_j| + |c_j| < |b_j|$$

Легко видеть, что для разностной схемы достаточное условие сходимости прогонки выполняется:

$$|a_j| + |c_j| = 2\frac{\Delta t}{h^2} < 1 + 2\frac{\Delta t}{h^2} = |b_j|$$

Задание 6

Найти α_1, β_1, u_N^{n+1}:

Для реализации неявной разностной схемы требуется ввести некоторое дополнительно условие, связывающее значения функции *u(t, x)* на *(n+1)*-м шаге по времени. Представлю это дополнительное условие в виде линейной зависимости

$$u_j^{n+1} = \alpha_j u_{j+1}^{n+1} + \beta_j, \tag{8}$$

справедливой для любого значений *j = 1, ..., N-1.*

Соотношение уравнения 8 называют **рекуррентным прогоночным соотношением**, а коэффициенты *a_j*, β_j - **прогоночными коэффициентами**.

Для определния прогоночных коэффициентов на 1-м шаге по координате x, использую рекуррентное прогоночное соотношение уравнения 8, записанное для *j = 1*:

$$u_1^{n+1} = \alpha_1 u_2^{n+1} + \beta_1,$$

и левое граничное условие:

$$u_1^{n+1} = e^{(n+1)\Delta t}$$

Сравнивая эти два соотношения, получаю:

$$\alpha_1 = 0, \beta_1 = e^{(n+1)\Delta t}.$$

Значение функции *u(t, x)* на *(n+1)*-м шаге по времени в крайней правой точке, которое можно определить из правого граничного условия:

$$u_N^{n+1} = e^{(n+1)\Delta t + 1}.$$

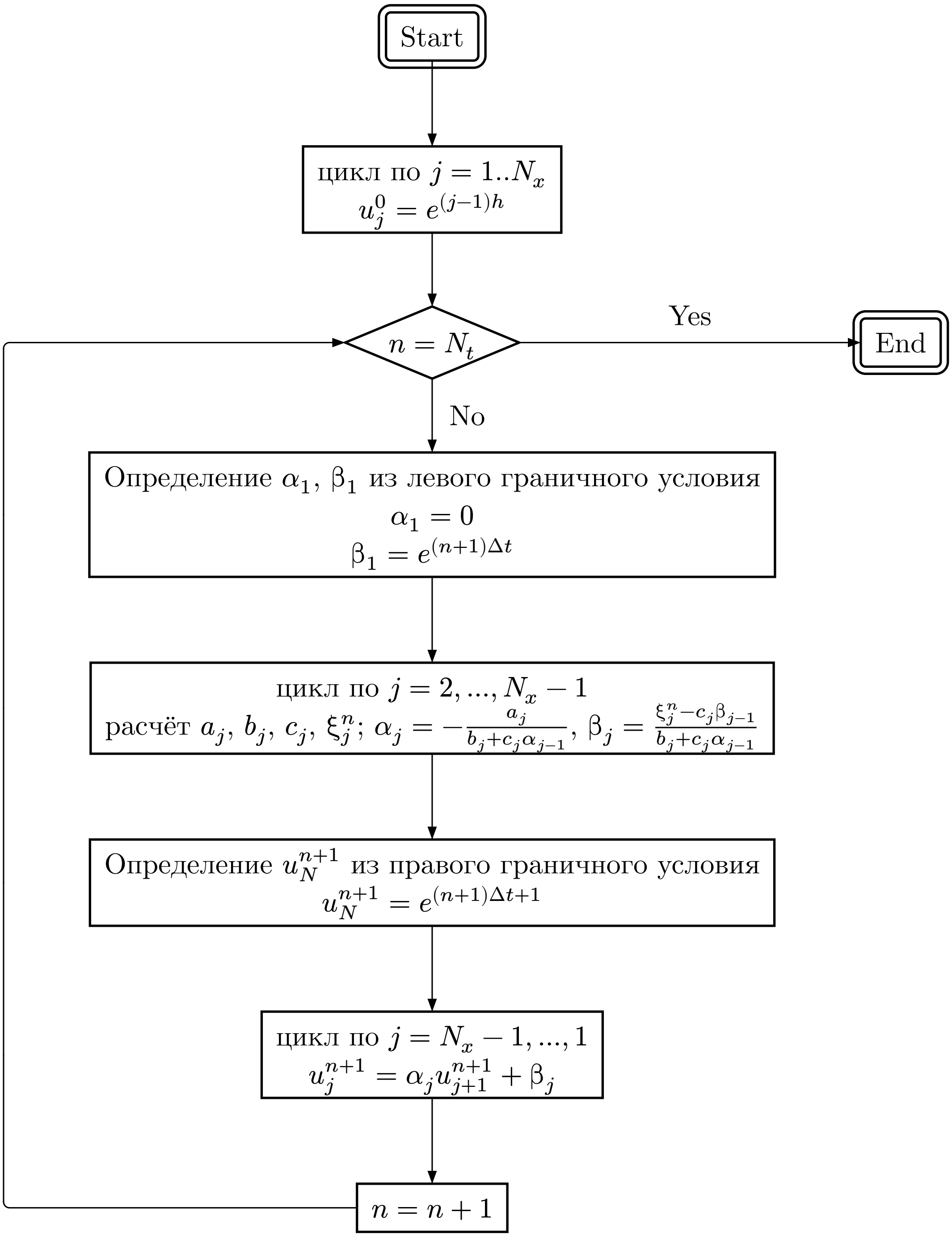
Задание 7

Записать рекуррентное прогоночное соотношение:

Соотношение уравнения 8 является **рекуррентным прогоночным соотношением**.

Задание 8

Составить алгоритм (блок-схему) расчёта:



Задание 9

Построить программу на любом удобном языке программирования:

```
#include <iostream>
#include <cmath>
#include <fstream>

#include "../include/Constants.h"

int main() {
  int N_x = 1 + (Constants::x_end - Constants::x_start) / Constants::h;
  int N_t[Constants::amount_of_delta_t] = {0};
  for (int i = 0; i < Constants::amount_of_delta_t; i++)
    N_t[i] = 1 + (Constants::t_end - Constants::t_start) / Constants::delta_t[i];
```

```
  double u[N_x][N_t[0]] = {0.0};
  for (int j = 0; j <= N_x - 1; j++) {
    u[0][j] = std::exp(j * Constants::h);
  }

  double a[N_x] = {0};
  double b[N_x] = {0};
  double c[N_x] = {0};
  double ksi[N_x] = {0};
  double alpha[N_x] = {0};
  double beta[N_x] = {0};

  int n = 0;
  while (!(n == (N_t[0] - 1))) {
    alpha[0] = 0.0;
    beta[0] = std::exp((n + 1) * Constants::delta_t[2]);

    for (int j = 1; j <= N_x - 2; j++) {
      a[j] = - (Constants::delta_t[2]) / (std::pow(Constants::h, 2));
      b[j] = 1 + 2 * (Constants::delta_t[2]) / (std::pow(Constants::h, 2));
      c[j] = - (Constants::delta_t[2]) / (std::pow(Constants::h, 2));
      ksi[j] = u[n][j];
      alpha[j] = - (a[j]) / (b[j] + c[j] * alpha[j - 1]);
      beta[j] = (ksi[j] - c[j] * beta[j - 1]) / (b[j] + c[j] * alpha[j - 1]);
    }

    u[n + 1][N_x - 1] = std::exp((n + 1) * Constants::delta_t[2] + 1);

    for (int j = N_x - 2; j >= 0; j--) {
      u[n + 1][j] = alpha[j] * u[n + 1][j + 1] + beta[j];
    }

    n++;
  }

  std::ofstream file (Constants::path);
  for (int n = 0; n < N_t[0]; n++) {
    double t = (n + 1) * Constants::delta_t[2];
    for (int j = 0; j < N_x; j++) {
      double x = j * Constants::h;
      file << t << " " << x << " " << u[n][j] << "\n";
    }
    file << "\n";
  }
  file.close();

  return 0;
}
```

Задание 11

Составить отчёт о проделанной работе. График функции $u(t, x)$ (Рис. 1).

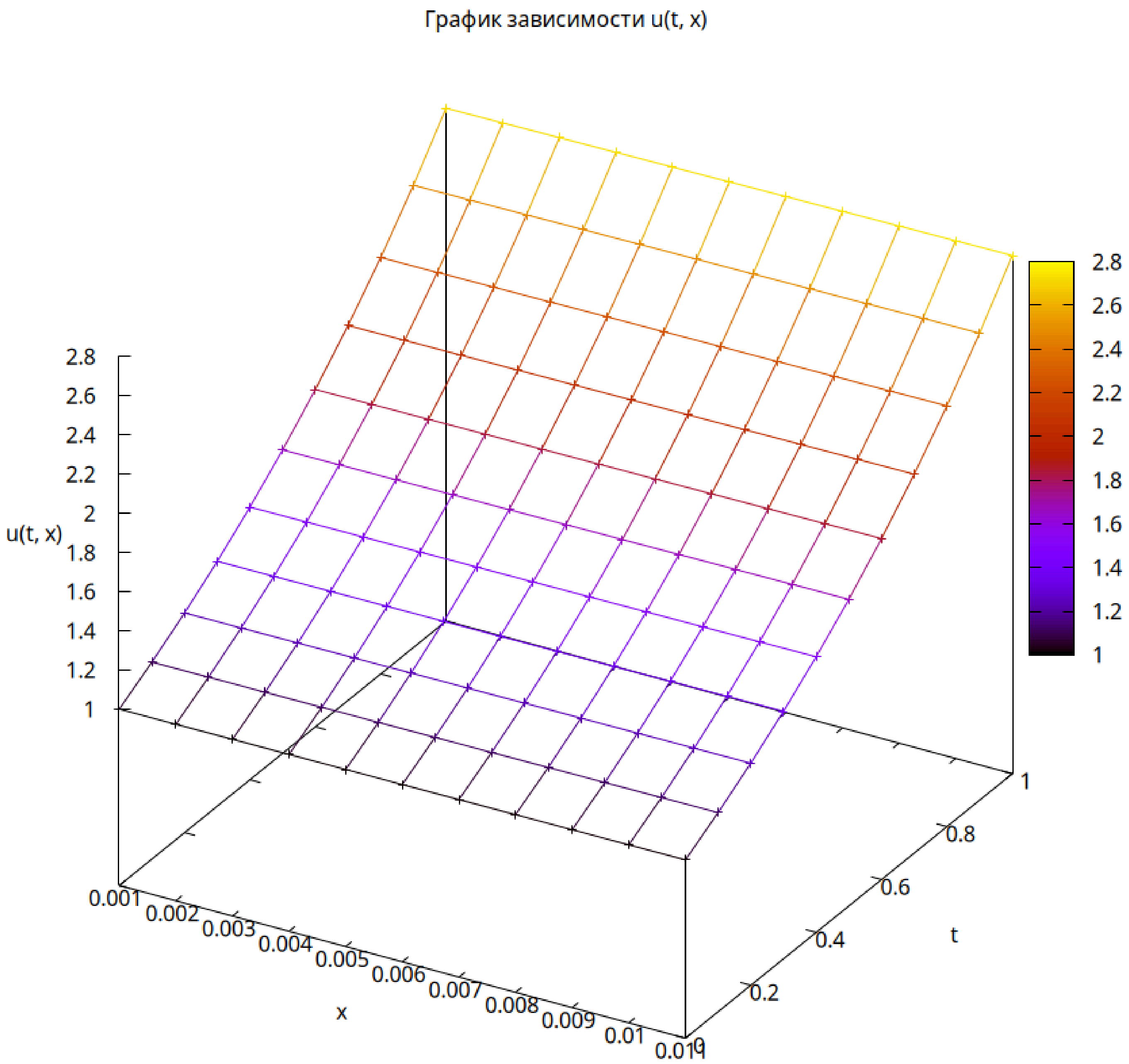


Рис. 1: График функции *u(t, x)*.