МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

Институт №8 «Компьютерные науки и прикладная математика»
Кафедра 806 «Вычислительная математика и программирование»

Лабораторная работа №*5*
по курсу «Численные методы»

Численное решение уравнений параболического типа.

Выполнил: *К. А. Полонский*
Группа: *М8О-408Б-20*
Преподаватель: Д. Е. Пивоваров

Москва, 2023

Условие

1. 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.
2. Вариант 10:

$$\frac{\partial u}{\partial t} = a\frac{\partial^2 u}{\partial x^2} + b\frac{\partial u}{\partial x} + cu, \quad a > 0, \quad b > 0, \quad c < 0.$$

$$u_x(0,t) + u(0,t) = \exp((c-a)t)(\cos(bt) + \sin(bt)),$$

$$u_x(\pi,t) + u(\pi,t) = -\exp((c-a)t)(\cos(bt) + \sin(bt)),$$

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

Аналитическое решение: $U(x,t) = \exp((c-a)t)\sin(x+bt)$.

Метод решения

Программа позволяет пользователю с помощью консольного ввода выбрать режим ввода параметров и метод решения параболического уравнения.

Сеточная функция представлена матрицей U размерности KxN, где K — число временных слоёв, N — число пространственных шагов.

Явная конечно-разностная схема была записана в форме:

$$u_j^{k+1} = (\sigma + \mu) * u_{j+1}^k + (1 - 2 * \sigma + \tau * c) * u_j^k + (\sigma - \mu) * u_{j-1}^k.$$

Неявная конечно-разностная схема была записана в форме:

$$a_j u_{j-1}^{k+1} + b_j u_j^{k+1} + c_j u_{j+1}^{k+1} = d_j,\ j = 1 \ldots N - 2,\ k = 0,\ 1,\ 2,\ \ldots$$

где (для двухточечной аппроксимации с первым порядком)

$$a_j = \frac{a*\tau}{h^2} - \frac{\tau*b}{2*h},$$

$$b_j = \tau * \frac{c-2*a}{h^2} - 1,$$

$$c_j = \frac{a*\tau}{h^2} + \frac{\tau*b}{2*h},\ j = 1,\ldots, N - 2.$$

$$a_0 = 0,$$

$$b_0 = \beta_0 - \frac{\alpha_0}{h},$$

$$c_0 = \frac{\alpha_0}{h},$$

$a_{N-1} = -\frac{\alpha_1}{h}$,

$b_{N-1} = \beta_1 + \frac{\alpha_1}{h}$,

$c_{N-1} = 0$.

Схема Кранка-Николсона была записана в форме:

$a_j u_{j-1}^{k+1} + b_j u_j^{k+1} + c_j u_{j+1}^{k+1} = d_j,\ j = 1\ldots N - 2,\ k = 0,\ 1,\ 2,\ \ldots$

где

$a_j = \sigma\frac{a*\tau}{h^2} - \frac{\tau*b}{2*h}$,

$b_j = \tau * \frac{c-\sigma*2*a}{h^2} - 1$,

$c_j = \sigma\frac{a*\tau}{h^2} + \frac{\tau*b}{2*h},\ j = 1,\ldots,N - 2.$

$a_0 = \beta_0 - \frac{3\alpha_0}{2h}$,

$b_0 = 2\frac{\alpha_0}{h}$,

$c_0 = -\frac{\alpha_0}{2h}$,

$a_{N-1} = \frac{\alpha_1}{2h}$,

$b_{N-1} = -\frac{2\alpha_1}{h}$,

$c_{N-1} = \beta_1 + \frac{3\alpha_1}{2h}$.

В конце работы программа записывает параметры условия, сеточную функцию и вектор ошибок в файл для скрипта отрисовки графиков.

Описание программы

Программа состоит из одного файла lab5.cpp, включающего функции:

- double phi0(double t) — функция граничного условия
- double phi1(double t) — функция граничного условия
- double psi(double x) — функция начального условия
- double analSol(double x, double t) — функция аналитического решения
- std::vector<std::vector<double>> explicitMethod(int approx) — функция для запуска явной конечно-разностной схемы, принимающая вид аппроксимации граничных условий
- void explicit2Points1Order(std::vector<std::vector<double>>& U) — функция, осуществляющая двухточечную аппроксимацию с первым порядком

- void explicit3Points2Order(std::vector<std::vector<double>>& U) — функция, осуществляющая трёхточечную аппроксимацию со вторым порядком
- void explicit2Points2Order(std::vector<std::vector<double>>& U) — функция, осуществляющая двухточечную аппроксимацию со вторым порядком
- double tridiagonalAlgo(std::vector<double>& a, std::vector<double>& b, std::vector<double>& c, std::vector<double>& d, std::vector<double>& x, int step, double prevP, double prevQ) — функция метода прогонки для решения СЛАУ с трёхдиагональной матрицей
- std::vector<std::vector<double>> implicitMethod(int approx) — функция для запуска неявной конечно-разностной схемы, принимающая вид аппроксимации граничных условий
- void implicit2Points1Order(std::vector<std::vector<double>>& U, std::vector<double>& lower, std::vector<double>& main, std::vector<double>& upper, std::vector<double>& coeffs, bool getA, int k) — функция, осуществляющая двухточечную аппроксимацию с первым порядком и принимающая три массива, соответствующие диагоналям матрицы СЛАУ, а также вектор свободных коэффициентов
- void implicit3Points2Order(std::vector<std::vector<double>>& U, std::vector<double>& lower, std::vector<double>& main, std::vector<double>& upper, std::vector<double>& coeffs, bool getA, int k) — функция, осуществляющая трёхточечную аппроксимацию со вторым порядком и принимающая три массива, соответствующие диагоналям матрицы СЛАУ, а также вектор свободных коэффициентов
- void implicit2Points2Order(std::vector<std::vector<double>>& U, std::vector<double>& lower, std::vector<double>& main, std::vector<double>& upper, std::vector<double>& coeffs, bool getA, int k) — функция, осуществляющая двухточечную аппроксимацию со вторым порядком и принимающая три массива, соответствующие диагоналям матрицы СЛАУ, а также вектор свободных коэффициентов
- std::vector<std::vector<double>> CrankNicolsonMethod() — функция запуска метода Кранка-Николса
- std::vector<double> getError(std::vector<std::vector<double>>& U) — функция, вычисляющая погрешность

Результаты

Для построения графиков функций (аналитического решения и численного) была написана программа на языке Python, использующая библиотеки numpy и matplotlib. Графики были построены для временного слоя timeSlice = 200, оранжевый цвет использовался для аналитического решения, чёрный — для численного.

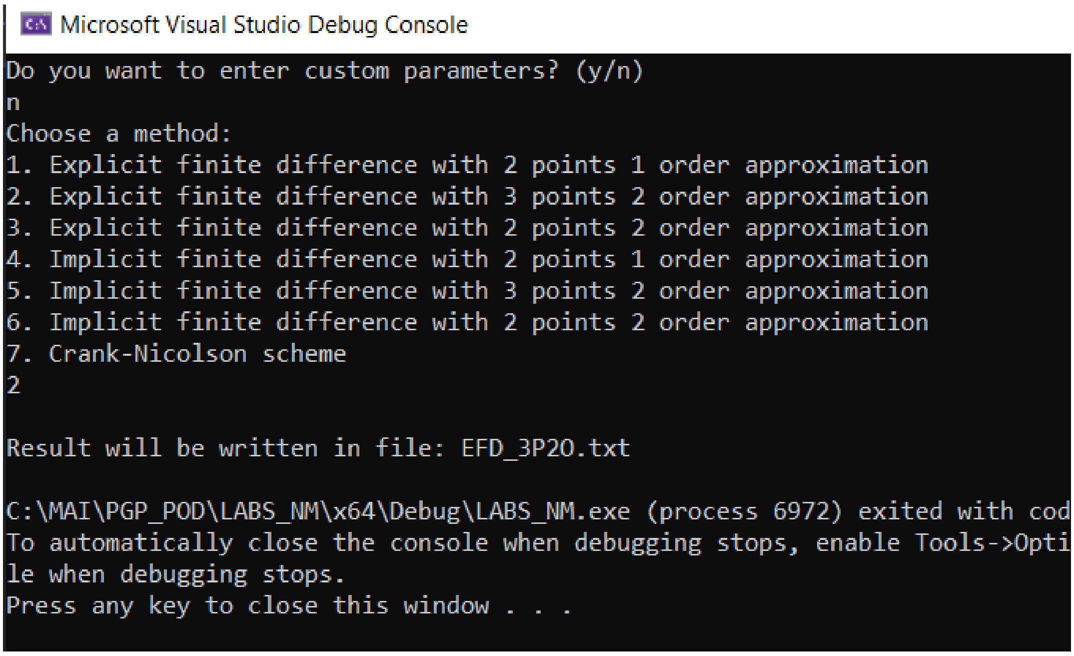

Рис. 1. Консольное взаимодействие программы с пользователем.

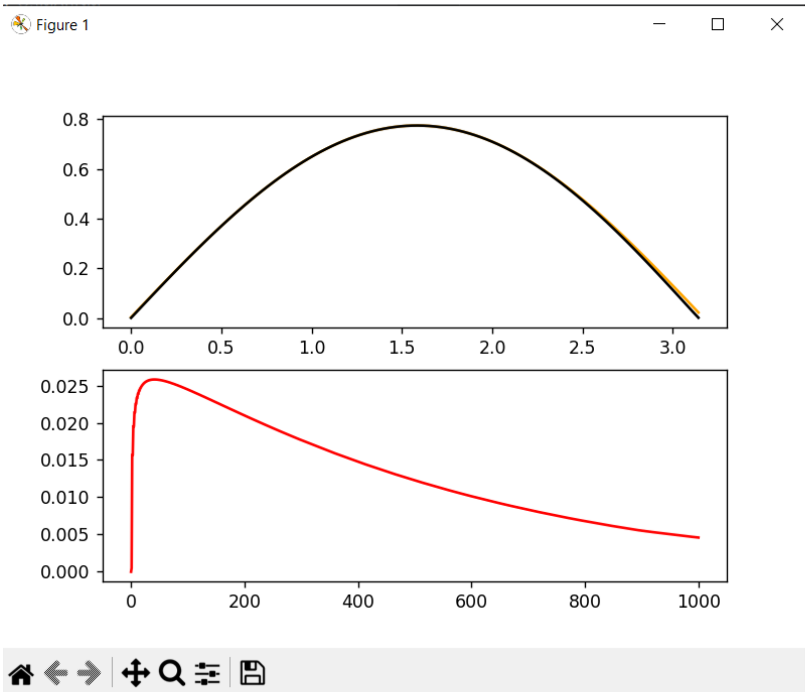

Рис. 2. График численного решения явной конечно-разностной схемой с двухточечной аппроксимацией второго порядка.

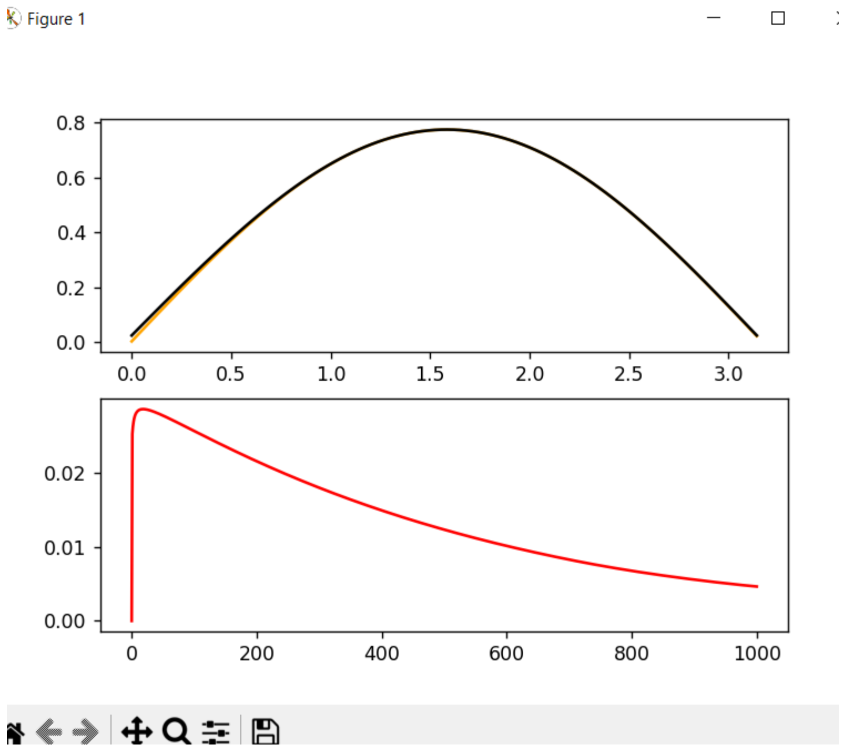

Рис. 3. График численного решения неявной конечно-разностной схемой с трёхточечной аппроксимацией второго порядка.

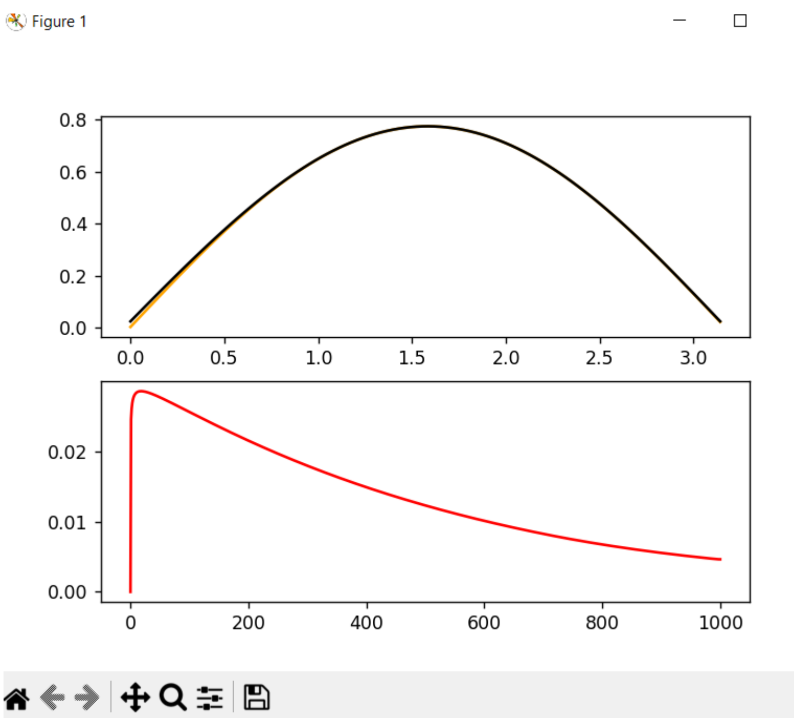

Рис. 4. График численного решения схемой Кранка-Николсона.

Выводы

В ходе выполнения данной лабораторной работы я освоил численные методы решения уравнений параболического типа, а именно явную и неявную конечно-разностные схемы и схему Кранка-Николсона. Практическое применение решения данной задачи лежит в области моделирования физических процессов и может применяться как при исследовательской деятельности, так и при разработке специализированного ПО.