

МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)

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

Лабораторная работа №7
по курсу «Численные методы»

Численное решение уравнений эллиптического типа.

Выполнил: *К. А. Полонский*

Группа: *М8О-408Б-20*

Преподаватель: *Д. Е. Пивоваров*

Москва, 2023

Условие

1. Решить краевую задачу для дифференциального уравнения эллиптического типа. Аппроксимацию уравнения произвести с использованием центрально-разностной схемы. Для решения дискретного аналога применить следующие методы: метод простых итераций (метод Либмана), метод Зейделя, метод простых итераций с верхней релаксацией. Вычислить погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением $U(x, y)$. Исследовать зависимость погрешности от сеточных параметров h_x, h_y .
2. Вариант 10:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2 \frac{\partial u}{\partial x} - 2 \frac{\partial u}{\partial y} - 4u,$$

$$u(0, y) = \exp(-y) \cos y,$$

$$u\left(\frac{\pi}{2}, y\right) = 0,$$

$$u(x, 0) = \exp(-x) \cos x,$$

$$u\left(x, \frac{\pi}{2}\right) = 0.$$

Аналитическое решение: $U(x, y) = \exp(-x - y) \cos x \cos y$.

Метод решения

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

Сеточная функция представлена матрицей U размерности $N_x \times N_y$, где N_x — число шагов по оси X , N_y — число шагов по оси Y .

Метод Либмана был записан в форме:

$$u_{ij}^n = \delta * \left((hxc + ac) * u_{i-1,j}^{n-1} + (hxc - ac) * u_{i+1,j}^{n-1} + (hyc + bc) * u_{i,j-1}^{n-1} \right) + \delta * (hyc - bc) * u_{i,j+1}^{n-1}, \text{ где}$$

n — номер итерации,

$$\delta = \frac{1}{\frac{2}{h_x^2} + \frac{2}{h_y^2} + c}, hxc = \frac{1}{h_x^2}, hyc = \frac{1}{h_y^2}, ac = \frac{a}{2h_x}, bc = \frac{b}{2h_y}.$$

Метод Зейделя был записан следующем виде:

$$u_{ij}^n = \delta * \left((hxc + ac) * u_{i-1,j}^{n-1} + (hxc - ac) * u_{i+1,j}^{n-1} + (hyc + bc) * u_{i,j-1}^{n-1} \right) + \delta * (hyc - bc) * u_{i,j+1}^{n-1}, \text{ где}$$

n — номер итерации,

$$\delta = \frac{1}{\frac{2}{h_x^2} + \frac{2}{h_y^2} + c}, hxc = \frac{1}{h_x^2}, hyc = \frac{1}{h_y^2}, ac = \frac{a}{2h_x}, bc = \frac{b}{2h_y}.$$

Метод простых итераций с верхней релаксацией был записан в форме:

$$u_{ij}^n += \omega * \{ \delta * [(hxc + ac) * u_{i-1,j}^{n-1} + (hxc - ac) * u_{i+1,j}^{n-1} + (hyc + bc) * u_{i,j-1}^{n-1} + \delta * (hyc - bc) * u_{i,j+1}^{n-1}] - u_{ij}^n \}, \text{ где}$$

n — номер итерации,

$$\delta = \frac{1}{\frac{2}{h_x^2} + \frac{2}{h_y^2} + c}, hxc = \frac{1}{h_x^2}, hyc = \frac{1}{h_y^2}, ac = \frac{a}{2h_x}, bc = \frac{b}{2h_y}.$$

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

Описание программы

Программа состоит из одного файла lab7.cpp, включающего функции:

- double phi0(double y) — функция граничного условия
- double phi1(double y) — функция граничного условия
- double phi2(double x) — функция начального условия
- double phi3(double x) — функция начального условия
- double analSol(double x, double y) — функция аналитического решения
- std::vector<std::vector<double>> solveEllipEquation(int meth) — функция для запуска решения эллиптического уравнения, принимающая вид метода
- void interpolate(std::vector<std::vector<double>>& U) — функция, осуществляющая заполнение сетки начальными приближёнными значениями
- void methLiebmann(std::vector<std::vector<double>>& U) — функция, осуществляющая решение методом Либмана
- void methSeidel(std::vector<std::vector<double>>& U) — функция, осуществляющая решение методом Зейделя
- void methUpperRelax(std::vector<std::vector<double>>& U) — функция, осуществляющая решение методом простых итераций с верхней релаксацией
- std::vector<double> getError(std::vector<std::vector<double>>& U) — функция, вычисляющая погрешность относительно аналитического решения
- double getMaxInaccuracy(std::vector<std::vector<double>>& oldU, std::vector<std::vector<double>>& newU) — функция, вычисляющая максимальную погрешность вычислений текущей итерации на основе предыдущей

Результаты

Для построения графиков функций (аналитического решения и численного) была написана программа на языке Python, использующая библиотеки numpy и matplotlib.

Графики были построены для среза по оси Y sliceY = 10, оранжевый цвет использовался для аналитического решения, чёрный — для численного.

```
Microsoft Visual Studio Debug Console
Do you want to enter custom parameters? (y/n)
n
Choose a method:
1. Liebmann method
2. Seidel method
3. Simple iterations with upper-relaxation
2
Result will be written in file: Seid.txt
C:\MAT\NM\LABS_NM\x64\Debug\LABS_NM.exe (process 6156) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .
```

Рис. 1. Консольное взаимодействие программы с пользователем.

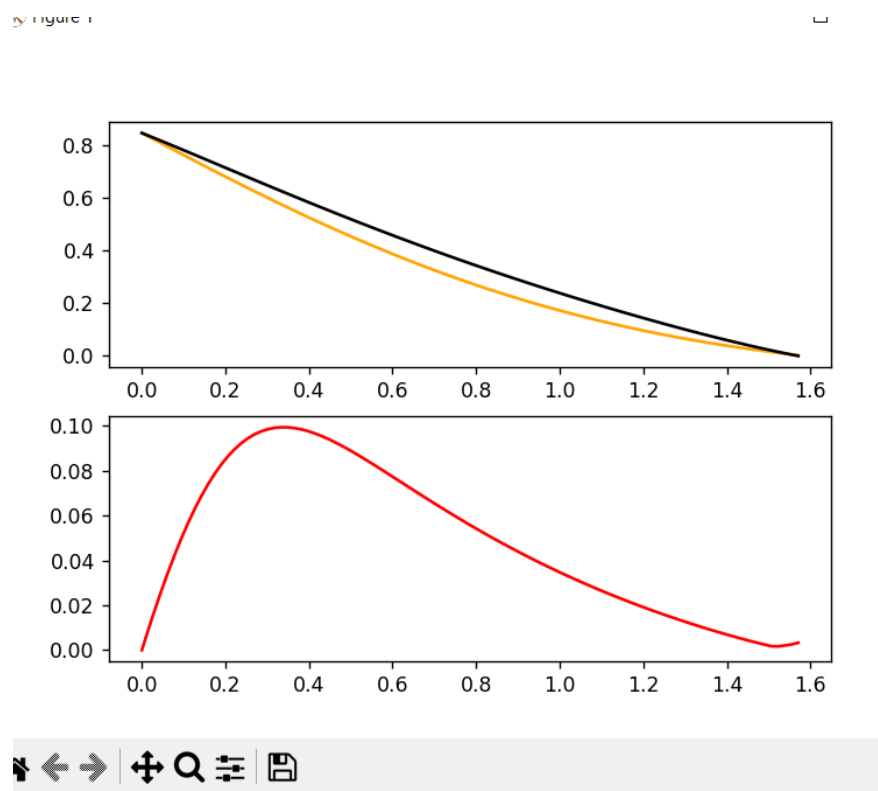


Рис. 2. График численного решения методом Либмана.

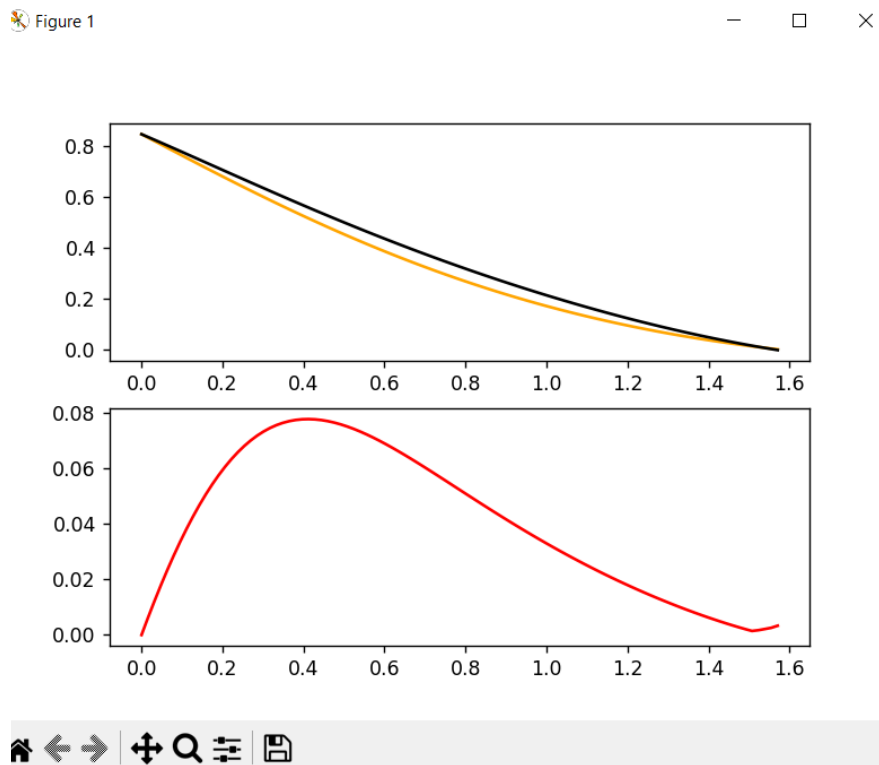


Рис. 3. График численного решения методом Зейделя.

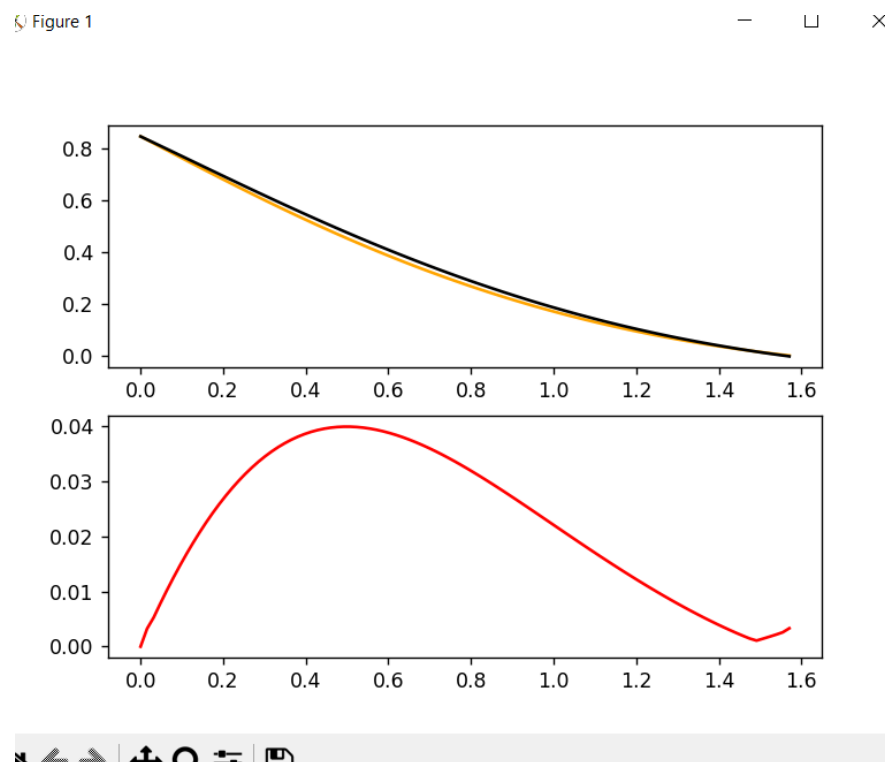


Рис. 4. График численного решения методом простых итераций с верхней релаксацией.

Выводы

В ходе выполнения данной лабораторной работы я освоил численные методы решения уравнений эллиптического типа. Практическое применение решения данной задачи

лежит в области моделирования физических процессов и иллюстрации физических явлений (в частности, процесс течения идеальной жидкости и распределение напряжённостей электрического и магнитного полей) и может применяться как при исследовательской деятельности, так и при разработке специализированного ПО.