



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ _____ «Информатика и системы управления»

КАФЕДРА _____ «Теоретическая информатика и компьютерные технологии»

**Отчет по лабораторной работе № 4
«Численное решение краевой задачи для
линейного дифференциального уравнения
второго порядка методом прогонки»**

*по курсу
«Численные методы»*

Выполнил:

Студент группы ИУ9-62Б

Караник А.А.

Проверила:

Домрачева А.Б.

2024 г.

Цель работы

Целью данной работы является реализация программы, вычисляющая численное решение краевой задачи для линейного дифференциального уравнения второго порядка методом прогонки.

Постановка задачи

- 1) Написать и отладить процедуру для решения трехдиагональной линейной системы методом прогонки.
- 2) Решить аналитически задачу Коши

$$y'' + py' + qy = f(x), \quad y(0) = y_0, y'(0) = y'_0$$

p	q	$f(x)$	y_0	y'_0
0	-4	$16xe^{2x}$	0	3

По найденному решению Коши $y(x)$ вычислить $b = y(1)$.

- 3) Найти численное решение $(x_i, y_i), i = 0, \dots, n$ краевой задачи для того же уравнения с краевыми условиями $y(0) = a, y(1) = b$ при $n = 10$.

- 4) Найти погрешность численного решения $||y - \tilde{y}|| = \max |y(x_i) - \tilde{y}_i|, 0 \leq i \leq n$

Теоретические сведения

Краевая задача для линейного дифференциального уравнения второго порядка имеет вид

$$y'' + p(x)y' + q(x)y = f(x)$$
$$y(0) = a, y(1) = b$$

Приближенным численным решением задачи называется сеточная функция $(x_i, y_i), i = 0, \dots, n$, заданная в $(n + 1)$ -й точке $x_i = ih, h = \frac{1}{n}$

Пусть $p_i = p(x_i), q_i = q(x_i), f_i = f(x_i)$ значения коэффициентов уравнения в точках x_i (узлах сетки), $i = 0, \dots, n$. Применяя разностную аппроксимацию производных по формулам численного дифференцирования, получим приближенную систему уравнений относительно ординат сеточной функции y_i :

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p_i \frac{y_{i+1} - y_{i-1}}{2h} + q_i y_i = f_i$$

Или после преобразований

$$y_{i-1} \left(1 - \frac{h}{2} p_i\right) + y_i (h^2 q_i - 2) + y_{i+1} \left(1 + \frac{h}{2} p_i\right) = h^2 f_i, \quad i = 1, \dots, n-1$$

С краевыми условиями

$$y_0 = a, y_n = b$$

Система является разностной системой с краевыми условиями и представляет собой трехдиагональную систему линейных алгебраических уравнений $(n+1)$ -го порядка, Трехдиагональную систему следует решать методом прогонки.

Погрешность численного решения вычисляется:

$$||y - \tilde{y}|| = \max |y(x_i) - \tilde{y}_i|, 0 \leq i \leq n$$

Реализация

Аналитическое решение:

$$y(x) = e^{-2x}(e^{4x}(2x^2 - x + 1) - 1)$$

Исходный код программы:

```
/**
 * Laboratory work: 4
 * Discipline: Numerical methods
 * Copyright (c) 2024, Andrey Karanik
 */

#include <iostream>
#define _USE_MATH_DEFINES
#include <cmath>
#include <iomanip>
#include <vector>

double* solve(double *a, double *b, double *c, double *d, int n) {
    double *alpha = new double[n - 1];
    double *beta = new double[n - 1];
    double *x = new double[n];

    for (int i = 1; i < n - 1; i++) {
        if (std::fabs(b[i]) < std::fabs(a[i - 1]) + std::fabs(c[i])) {
            std::cout << "The condition is not satisfied." << std::endl;
            break;
        }
        if (std::fabs(b[i]) / (std::fabs(c[i])) < 1) {
            std::cout << "The condition is not satisfied." << std::endl;
            break;
        }
        if (std::fabs(b[i]) / (std::fabs(a[i - 1])) < 1) {
            std::cout << "The condition is not satisfied." << std::endl;
            break;
        }
    }
}
```

```

    }

    if (b[0] != 0) {
        alpha[0] = -c[0] / b[0];
    } else {
        std::cout << "ERROR: b[0] = 0" << std::endl;
        return x;
    }

    beta[0] = d[0] / b[0];

    for (int i = 1; i < n - 1; i++) {
        alpha[i] = -(c[i]) / (alpha[i - 1] * a[i - 1] + b[i]);
        beta[i] = (d[i] - a[i - 1] * beta[i - 1]) / (alpha[i - 1] * a[i - 1] + b[i]);
    }
    beta[n - 1] = (d[n - 1] - a[n - 2] * beta[n - 2]) / (alpha[n - 2] * a[n - 2] + b[n - 1]);

    x[n - 1] = beta[n - 1];

    std::cout << "";

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

    delete[] alpha;
    delete[] beta;

    return x;
}

double f(double x) {
    return 16 * x * exp(2 * x);
}

double y(double x) {
    return exp(-2 * x) * (exp(4 * x) * (2 * x * x - x + 1) - 1);
}

int main(int argc, char *argv[]) {

    int n = 10;
    double p = 0;
    double q = -4;

    double h = 1.0f / n;
    double* xi = new double[n + 1];
    double* yi = new double[n + 1];
    double* ai = new double[n - 2];
    double* bi = new double[n - 1];
    double* ci = new double[n - 2];
    double* di = new double[n - 1];

    for (int i = 0; i <= n; i++) {
        xi[i] = i * h;
    }

    double a = y(0);
    double b = y(1);
    yi[0] = a;
    yi[n] = b;

    for (int i = 0; i < n - 2; i++) {
        ai[i] = 1 - (h / 2) * p;
        ci[i] = 1 + (h / 2) * p;
    }

    for (int i = 0; i < n - 1; i++) {
        bi[i] = h * h * q - 2;
    }

```

```

}

di[0] = h * h * f(xi[1]) - yi[0] * (1 - (h / 2) * p);
di[n - 2] = h * h * f(xi[n - 1]) - yi[n] * (1 + (h / 2) * p);

for (int i = 1; i < n - 2; i++) {
    di[i] = h * h * f(xi[i + 1]);
}

double* solution = solve(ai, bi, ci, di, n - 1);

for (int i = 1; i < n; i++) {
    yi[i] = solution[i - 1];
}

std::cout << "xi y*(xi) y(xi) |y*(xi)-y(xi)|" << std::endl;

for (int i = 0; i <= n; i++) {
    std::cout << xi[i] << " " << yi[i] << " " << y(xi[i]) << " " << fabs(y(xi[i]) - yi[i]) << std::endl;
}

std::cout << std::endl;

double error = 0;
for (int i = 0; i <= n; i++) {
    if (error < fabs(y(xi[i]) - yi[i])) {
        error = fabs(y(xi[i]) - yi[i]);
    }
}

std::cout << "Error of numerical solution: " << error << std::endl;

return 0;
}

```

Тестирование

Вывод программы:

```

xi y*(xi) y(xi) |y*(xi)-y(xi)|
0 0 0 0
0.1 0.312909 0.30496 0.00794951
0.2 0.657877 0.642486 0.0153917
0.3 1.0769 1.05465 0.0222461
0.4 1.62646 1.59817 0.0282896
0.5 2.38351 2.3504 0.0331081
0.6 3.45337 3.41734 0.0360289
0.7 4.98009 4.94406 0.0360276
0.8 7.16019 7.12859 0.031602
0.9 10.2607 10.2401 0.0206015
1 14.6428 14.6428 7.74772e-07

```

Error of numerical solution: 0.0360289

Вывод

В процессе выполнения лабораторной работы была разработана программа, вычисляющая численное решение краевой задачи для линейного дифференциального уравнения второго порядка методом прогонки. Также было найдено аналитическое решение задачи Коши, численное решение краевой задачи и посчитана погрешность численного решения.