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 Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

Лабораторная работа № 4

«Численное решение краевой задачи для линейного дифференциального уравнения второго порядка методом прогонки»

по курсу «Численные методы»

Выполнил:

студент группы ИУ9-61Б

Окутин Денис

Проверила:

Домрачева А. Б.

Москва, 2024

1. Цель

Целью данной работы является вычисление численного решения задачи для линейного дифференциального уравнения второго порядка методом прогонки.

2. Постановка задачи

Дано: краевая задача для линейного дифференциального уравнения (ДУ) второго порядка

$y'' + p(x)y' + q(x)y = f(x)$

$y(0) = \boldsymbol{a}$

$y(1) = \boldsymbol{b}$

Задание:

- Найти аналитическое решение задачи Коши:

  $y'' + p(x)y' + q(x)y = f(x),\ y(0) = y_0,\ y'(0) = y'_0$;
- По найденному решению задачи Коши вычислить $b = y(1)$;
- С помощью метода прогонки найти численное решение $(x_i, y_i),\ i = \overline{0,n},\ n = 10$ краевой задачи для того же уравнения с краевыми условиями $y(0) = a,\ y(1) = b$;
- Вычислить $|y_i - \widetilde{y_i}|,\ i = \overline{0,n}$, найти погрешность численного решения $\|y - \tilde{y}\| = \max\limits_{0 \le i \le n} |y_i - \widetilde{y_i}|$ и сравнить (здесь y – аналитическое решение, $\tilde{y}$ – численное решение).

Индивидуальный вариант: $p(x) = -1, q(x) = 0, f(x) = 3,$

$y_0 = 0, y'_0 = 2$

Краевая задача имеет вид:

$y'' - 1y' = 5 * e^x - 3 * x - 5$

$y(0) = 0$

$y(1) = b$

3. Основные теоретические сведения

Метод прогонки

Пусть требуется решить краевую задачу на отрезке [0,1] (т.е. краевые условия ДУ заданы в точках $x = 0, x = 1$). Тогда отрезок разбивается на n равных отрезков длины $h = \frac{1-0}{n} = \frac{1}{n}$. Получаем разбиение отрезка точками $x_i = ih,\ h = \frac{1}{n},\ i = \overline{0,n}$.

Приближенный численным решением краевой задачи для ДУ второго порядка называется сеточная функция $(x_i, y_i),\ i = \overline{0,n}$, заданная в точках $x_i = ih,\ h = \frac{1}{n}$.

Обозначим значения коэффициентов уравнения в точках $x_i,\ i = \overline{0,n}$ через $p_i = p(x_i),\ q_i = q(x_i),\ f_i = f(x_i)$. При помощи разностной аппроксимации производных получаем приближенную систему уравнений относительно y_i:

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

После преобразования система имеет вид:

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

с краевыми условиями $y_0 = \boldsymbol{a},\ y_n = \boldsymbol{b}$

Данная система имеет порядок $n - 1$ и представляет собой трехдиагональную систему линейных алгебраических уравнений, ее необходимо решить методом прогонки.

Напомним, что метод прогонки позволяет решать системы вида $A\bar{x} = \bar{d}$, где A - трехдиагональная матрица:

$$\begin{pmatrix} b_1 & c_1 & 0 & \cdots & \cdots & 0 \\ a_1 & b_2 & c_2 & \cdots & \cdots & 0 \\ 0 & a_2 & b_3 & c_3 & \cdots & 0 \\ \vdots & \cdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & a_{n-2} & b_{n-1} & \vdots \\ 0 & \cdots & \cdots & \cdots & a_{n-1} & b_n \end{pmatrix}$$

где a – массив элементов под главной диагональю, b – массив элементов главной диагонали, c – массив элементов над главной диагональю.

Для рассматриваемой задачи элементы массивов a, b, c, d будут иметь вид:

$$a_i = 1 - \frac{h}{2}p_i, \qquad i = \overline{1, n-2}$$

$$b_i = h^2 q_i - 2, \qquad i = \overline{1, n-1}$$

$$c_i = 1 + \frac{h}{2}p_i, \qquad i = \overline{1, n-2}$$

$$d_i = h^2 f_i, \qquad i = \overline{2, n-2}$$

Поскольку $y_0 = \boldsymbol{a},\ y_n = \boldsymbol{b}$, то

$$d_1 = h^2 f_1 - y_0\left(1 - \frac{h}{2}p_1\right) = h^2 f_1 - \boldsymbol{a}(1 - \frac{h}{2}p_1), \qquad i = 1$$

$$d_{n-1} = h^2 f_{n-1} - y_n\left(1 + \frac{h}{2}p_{n-1}\right) = h^2 f_1 - \boldsymbol{b}\left(1 + \frac{h}{2}p_{n-1}\right), \qquad i = n-1$$

4. Реализация

Аналитическое решение для задачи Коши найдено с помощью онлайн сервиса по решению ДУ.

По вычисленному решению находим $\boldsymbol{b} = y(1)$:

Листинг 1. Метод прогонки для решения краевой задачи для ДУ второго порядка

```
import math
from decimal import Decimal, getcontext

getcontext().prec = 25



def forward(a, b, c, d):
    if b[0] == 0 or b[len(b) - 1] == 0:
        raise Exception("invalid b data, cant
calculate forward")
    if abs(c[0]) / abs(b[0]) > 1 or abs(a[len(b) - 1
```

```
- 1]) / abs(b[len(b) - 1]) > 1:
        raise Exception("invalid matrix data, cant
calculate forward")

    y = [Decimal(0.0)] * len(b)
    alpha = [Decimal(0.0)] * len(b)
    beta = [Decimal(0.0)] * len(b)
    y[0] = b[0]
    alpha[0] = -c[0] / b[0]
    beta[0] = d[0] / b[0]

    n = len(b) - 1
    for i in range(len(b)):
        if 1 < i < n and abs(b[i]) < abs(a[i - 1]) +
abs(c[i]):
            raise Exception("invalid matrix data,
cant calculate forward")

        if i == 0:
            continue
        elif i == n:
            y[n] = b[n] + a[n - 1] * alpha[n - 1]
            beta[n] = (d[n] - a[n - 1] * beta[n - 1])
/ y[n]
        else:
            y[i] = b[i] + alpha[i - 1] * a[i - 1]
            alpha[i] = -c[i] / y[i]
            beta[i] = (d[i] - a[i - 1] * beta[i - 1])
/ y[i]
```

```
    return alpha, beta


def backward(alpha, beta):
    x = [0] * len(beta)
    n = len(beta) - 1
    x[n] = beta[n]
    for i in range(n - 1, -1, -1):
        x[i] = alpha[i] * x[i + 1] + beta[i]

    return x


def f():
    return 3.0


# через вольфрам альфа
def analytical(x):
    return 5 * (math.e ** x) - 3 * x - 5


p = -1.0
q = 0.0
a = analytical(0)
b = analytical(1)


def solve(n, h, a, p):
    as_ = []
```

```
    bs = []
    cs = []
    ds = []

    for i in range(1, n - 1):
        as_.append(1 - h / 2 * p)
        cs.append(1 + h / 2 * p)

    for i in range(1, n):
        bs.append(h * h * q - 2)

    ds.append(h * h * f() - a * (1 - h / 2 * p))
    for i in range(2, n):
        ds.append(h * h * f())
    ds[-1] = h * h * f() - b * (1 + h / 2 * p)

    alpha, beta = forward(as_, bs, cs, ds)
    ys = [a]
    ys.extend(backward(alpha, beta))
    ys.append(b)

    return ys


def main():
    print(f"y'' + {p}y' + {q}y = 5 * e^x - 3x - 5")
    print(f"y(0) = {a}\ny(1) = {b}")

    n = 40
    h = 1.0 / float(n)
```

```
    print(a,b)

    xs = []
    for i in range(n + 1):
        xs.append(float(i) * h)

    ys = solve(n, h, a, p)

    for i in range(len(ys)):
        print(f"x={float(i) * h:.2f},
y={analytical(xs[i]):.6f},"
              f" y*={ys[i]:.6f}, |y-y*|={abs(ys[i] -
analytical(xs[i])):.6f}")

    maxInaccuracy = 0.0
    for i in range(0, len(ys), 4):
        if abs(ys[i] - analytical(xs[i])) >
maxInaccuracy:
            maxInaccuracy = abs(ys[i] -
analytical(xs[i]))
    print(f"y-y={maxInaccuracy:.6f}")

if __name__ == "__main__":
    main()
```

5. Результаты

Таблица 1 - Результаты метода прогонки

Значение x	Значение y (аналитическое решение)	Значение $\tilde{y}$ (численное решение)	$\lvert y_i - \tilde{y}_i \rvert$

0	0.000000	0.000000	0
0.1	0.225855	0.225840	0.000015
0.2	0.507014	0.506986	0.000028
0.3	0.849294	0.849255	0.000039
0.4	1.259123	1.259076	0.000047
0.5	1.743606	1.743554	0.000053
0.6	2.310594	2.310540	0.000054
0.7	2.968764	2.968713	0.000051
0.8	3.727705	3.727663	0.000041
0.9	4.598016	4.597991	0.000025
1.0	5.591409	5.591409	0

$$\|y-\tilde{y}\| = \max_{0\le i\le n}|y_i-\tilde{y_i}| = 0.000054$$

6. Вывод

В ходе выполнения лабораторной работы был реализован метод приближенного численного решения краевой задачи для дифференциального уравнения второго порядка методом прогонки.

Вследствие сравнения результатов работы методов сделан вывод о том, что метод прогонки обладает достаточно высокой точностью в решении поставленной в условии задачи. Кроме того, вычислительная погрешность обусловлена малым количеством разбиений рассматриваемого отрезка. С увеличением числа разбиений, погрешность уменьшается.