



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

«Сравнительный анализ методов численного решения краевой
задачи для линейного дифференциального уравнения второго
порядка»

по курсу «Численные методы»

Выполнила:

студент группы ИУ9-61Б

Яровикова Анастасия

Проверила:

Домрачева А. Б.

Москва, 2023

1. Цель

Целью данной работы является сравнение по точности решения двух методов численного решения краевой задачи для линейного дифференциального уравнения второго порядка:

1. Метод прогонки
2. Метод стрельбы

2. Постановка задачи

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

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

$$y(0) = a$$

$$y(1) = b$$

Задание:

- Найти аналитическое решение задачи Коши:

$$y'' + p(x)y' + q(x)y = f(x), \quad y(0) = y_0, \quad y'(0) = y'_0;$$

- По найденному решению задачи Коши вычислить $b = y(1)$;
- С помощью метода прогонки и метода стрельбы найти численное решение (x_i, y_i) , $i = \overline{0, n}$, $n = 10$ краевой задачи для того же уравнения с краевыми условиями $y(0) = a$, $y(1) = b$;
- Для каждого метода вычислить $|y_i - \tilde{y}_i|$, $i = \overline{0, n}$, найти погрешность численного решения $\|y - \tilde{y}\| = \max_{0 \leq i \leq n} |y_i - \tilde{y}_i|$ и сравнить (здесь y – аналитическое решение, \tilde{y} – численное решение).

Индивидуальный вариант: $p(x) = 0, q(x) = 4, f(x) = 3 \sin(2x)$,

$$y_0 = 2, y'_0 = 0.75$$

Краевая задача имеет вид:

$$y'' + 4y = 3 \sin(2x)$$

$$y(0) = 2$$

$$y(1) = b$$

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 :

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

После преобразования система имеет вид:

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

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

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:

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

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

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

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

Поскольку $y_0 = a$, $y_n = b$, то

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

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

Метод стрельбы

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 $x_i = ih$, $h = \frac{1}{n}$, $i = \overline{0, n}$.

При помощи разностной аппроксимации производных получаем формулы для производных y_i :

$$y'_i = \frac{y_{i+1} - y_{i-1}}{2h}, \quad y''_i = \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}, \quad i = \overline{1, n-1}$$

Обозначим значения $y_0(x_i) = y_0[i]$, $y_1(x_i) = y_1[i]$, а значения коэффициентов уравнения в точках x_i , $i = \overline{0, n}$ через $p_i = p(x_i)$, $q_i = q(x_i)$, $f_i = f(x_i)$.

Ищем решения y_i , удовлетворяющие условиям:

$$y_0[0] = a, \quad y_0[1] = D_0, \quad y_1[0] = 0, \quad y_1[1] = D_1 \neq 0$$

где $D_0 = a + O(h)$, $D_1 = O(h)$. За $O(h)$ возьмем значение $O(h) = h$.

Для определения y_0 и y_1 получаем уравнения:

$$\frac{y_0[i+1] - 2y_0[i] + y_0[i-1]}{h^2} + p_i \frac{y_0[i+1] - y_0[i-1]}{2h} + q_i y_0[i] = f_i,$$

$$\frac{y_1[i+1] - 2y_1[i] + y_1[i-1]}{h^2} + p_i \frac{y_1[i+1] - y_1[i-1]}{2h} + q_i y_1[i] = 0.$$

Используя условия $y_0[0] = \mathbf{a}, y_0[1] = D_0, y_1[0] = 0, y_1[1] = D_1 \neq 0$,
имеем:

$$y_0[i+1] = \frac{(f_i h^2 + (2 - q_i h^2) y_0[i] - (1 - p_i \frac{h}{2}) y_0[i-1])}{1 + \frac{p_i h}{2}},$$

$$y_1[i+1] = \frac{\left((2 - q_i h^2) y_1[i] - \left(1 - p_i \frac{h}{2} \right) y_1[i-1] \right)}{1 + \frac{p_i h}{2}},$$

$$i = \overline{1, n-1};$$

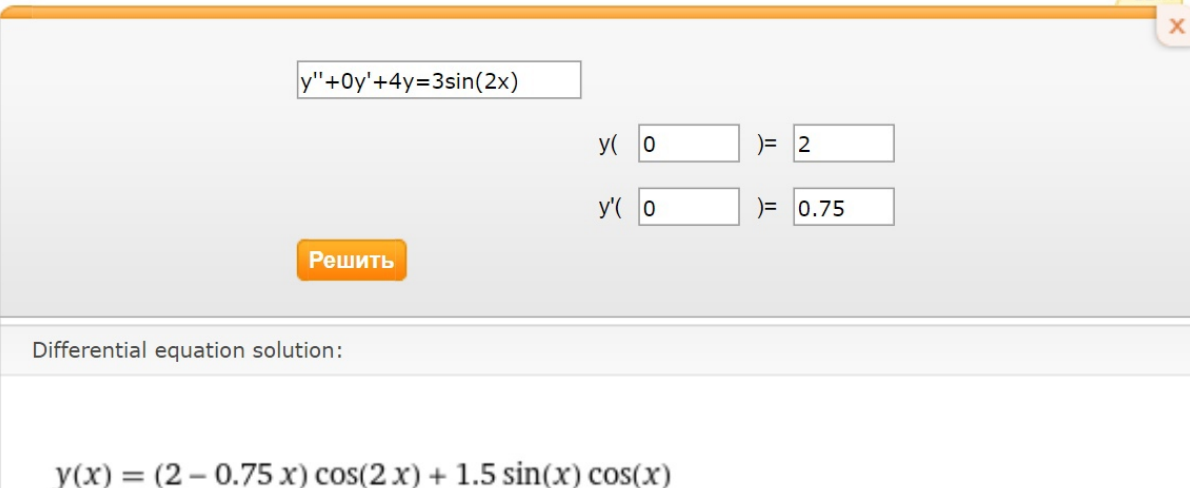
Далее последовательно определяем $y_0[2], \dots, y_0[n], y_1[2], \dots, y_1[n]$ и находим $C_1 = (\mathbf{b} - y_0[n])/y_1[n]$.

Искомое решение ищется по формулам:

$$y[i] = y_0[i] + C_1 y_1[i], \quad i = \overline{1, n}.$$

4. Реализация

Аналитическое решение для задачи Коши найдено с помощью WolframAlpha:



The screenshot shows the WolframAlpha interface. At the top, the differential equation $y'' + 0y' + 4y = 3\sin(2x)$ is entered in the input box. Below the input box, there are two rows of initial conditions: $y(0) = 2$ and $y'(0) = 0.75$. A blue button labeled "Решить" (Solve) is located below the input box. Below the button, the text "Differential equation solution:" is displayed. At the bottom, the solution is given as $y(x) = (2 - 0.75 x) \cos(2 x) + 1.5 \sin(x) \cos(x)$.

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

$$\mathbf{b} = 1.25 \cos(2) + 1.5 \sin(1) \cos(1)$$

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

```
package main

import (
    "fmt"
    "math"
)

func direct(b, a, c, d []float64, size int) (alpha,
beta []float64) {
    alpha = append(alpha, -c[0] / b[0])
    beta = append(beta, d[0] / b[0])
    var y float64
    for i := 1; i < size - 1; i++ {
        y = a[i - 1] * alpha[i - 1] + b[i]
        alpha = append(alpha, -c[i] / y)
        beta = append(beta, (d[i] - a[i - 1] * beta[i
- 1]) / y)
    }
    y = a[size - 2] * alpha[size - 2] + b[size - 1]
    beta = append(beta, (d[size - 1] - a[size - 2] *
beta[size - 2]) / y)
    return alpha, beta
}
```

```

func reverse(alpha, beta []float64, size int) (x
[]float64) {
    x = make([]float64, size)
    x[size - 1] = beta[size - 1]
    for i := size - 2; i >= 0; i-- {
        x[i] = alpha[i] * x[i + 1] + beta[i]
    }
    return x
}

func f(x float64) float64 {
    return 3 * math.Sin(2 * x)
}

func analytical(x float64) float64 {
    return (2 - 0.75 * x) * math.Cos(2 * x) + 1.5 *
math.Sin(x) * math.Cos(x)
}

var (
    n = 10
    p = 0.0
    q = 4.0
    a = analytical(0)
    b = analytical(1)
)

func main() {
    fmt.Println("МЕТОД ПРОГОКНИ")
}

```

```

    fmt.Printf("y'' + %0.1fy' + %0.1fy = (2 -
0.75*x)*cos(2x) + 1.5*sin(x)*cos(x)\n", p, q)
    fmt.Printf("y(0) = %f\n", a, b)
    h := 1.0 / float64(n)
    xs := make([]float64, 0, n)
    for i := 0; i < n + 1; i++ {
        xs = append(xs, float64(i) * h)
    }

    as := make([]float64, 0, n - 2)
    bs := make([]float64, 0, n - 1)
    cs := make([]float64, 0, n - 2)
    ds := make([]float64, 0, n - 1)

    for i := 1; i < n - 1; i++ {
        as = append(as, 1 - h / 2 * p)
        cs = append(cs, 1 + h / 2 * p)
    }

    for i := 1; i < n; i++ {
        bs = append(bs, h * h * q - 2)
    }

    ds = append(ds, h * h * f(0) - a * (1 - h / 2 *
p))
    for i := 2; i < n; i++ {
        ds = append(ds, h * h * f(float64(i) * h))
    }
    ds[len(ds) - 1] = h * h * f(float64(len(ds) - 1)
* h) - b * (1 + h / 2 * p)

```



```

alpha, beta := direct(bs, as, cs, ds, len(ds))
ys := []float64{a}
ys = append(ys, reverse(alpha, beta, len(ds))...)
ys = append(ys, b)

maxInaccuracy := 0.0
for i := 0; i < len(ys); i++ {
    fmt.Printf("x=%.1f,   y=%.6f,   y*=%.6f   |y-
y*|=%.6f\n",
        float64(i) * h,   analytical(xs[i]),
ys[i], math.Abs(ys[i] - analytical(xs[i])))
    if math.Abs(ys[i] - analytical(xs[i])) >
maxInaccuracy {
        maxInaccuracy = math.Abs(ys[i] -
analytical(xs[i]))
    }
}
fmt.Printf("||y-y*||=%.6f\n", maxInaccuracy)
}

```

Листинг 2. Метод стрельбы для решения краевой задачи для ДУ второго порядка

```

package main
package main

import (
    "fmt"
    "math"

```

```

)

func f(x float64) float64 {
    return 3 * math.Sin(2 * x)
}

func analytical(x float64) float64 {
    return (2 - 0.75 * x) * math.Cos(2 * x) + 1.5 *
math.Sin(x) * math.Cos(x)
}

var (
    n = 10
    p = 0.0
    q = 4.0
    a = analytical(0)
    b = analytical(1)
    ys = make([][]float64, 2)
)

func getC1() float64 {
    return (b - ys[0][n]) / ys[1][n]
}

func getYi(i int) float64 {
    return ys[0][i] + getC1() * ys[1][i]
}

func main() {
    fmt.Println("МЕТОД СТРЕЛЬБЫ")

```

```

    fmt.Printf("y'' + %0.1fy' + %0.1fy = (2 -
0.75*x)*cos(2x) + 1.5*sin(x)*cos(x)\n", p, q)
    fmt.Printf("y(0) = %f\n", a)
    fmt.Printf("Количество разбиений: %d\n", n)
    h := 1.0 / float64(n)
    fmt.Println(h)
    delta := h
    xs := make([]float64, 0, n + 1)
    for i := 0; i <= n; i++ {
        xs = append(xs, float64(i) * h)
    }
    for i := 0; i < 2; i++ {
        ys[i] = make([]float64, 2, n)
    }
    ys[0][0], ys[0][1] = a, a + delta
    ys[1][0], ys[1][1] = 0, delta

    for i := 1; i < n; i++ {
        ys[0] = append(ys[0],
            (h * h * f(xs[i]) + (2.0 - q * h *
h) * ys[0][i] - (1.0 - h / 2 * p) * ys[0][i - 1]) /
(1 + h / 2 * p))
        ys[1] = append(ys[1],
            ((2.0 - q * h * h) * ys[1][i] - (1.0
- h / 2 * p) * ys[1][i - 1]) / (1 + h / 2 * p))
    }

    y := make([]float64, 0, n + 1)
    for i := 0; i <= n; i++ {
        y = append(y, getYi(i))
    }

```

```

    }
    maxInaccuracy := 0.0
    for i := range y {
        fmt.Printf("x=%.1f,   y=%.6f,   y*=%.6f   |y-
y*|=%.6f\n",
            float64(i) * h, analytical(xs[i]), y[i],
            math.Abs(y[i] - analytical(xs[i])))
        if math.Abs(y[i] - analytical(xs[i])) >
maxInaccuracy {
            maxInaccuracy = math.Abs(y[i] -
analytical(xs[i]))
        }
    }
    fmt.Printf("||y-y*||=%.6f\n", maxInaccuracy)
}

```

5. Результаты

Таблица 1 - Результаты метода прогонки

Значение x	Значение y (аналитическое решение)	Значение \tilde{y} (численное решение)	$ y_i - \tilde{y}_i $
0	2.000000	2.000000	0
0.1	2.035630	2.035760	0.000130
0.2	1.996027	1.996174	0.000147
0.3	1.888453	1.888599	0.000147
0.4	1.722418	1.722550	0.000132
0.5	1.509094	1.509200	0.000106
0.6	1.260684	1.260758	0.000074
0.7	0.989789	0.989829	0.000040
0.8	0.708801	0.708810	0.000009
0.9	0.429343	0.429326	0.000016
1.0	0.161790	0.161790	0

$$\|y - \tilde{y}\| = \max_{0 \leq i \leq n} |y_i - \tilde{y}_i| = 0.000147$$

Таблица 2 - Результаты метода стрельбы

Значение x	Значение y (аналитическое решение)	Значение \tilde{y} (численное решение)	$ y_i - \tilde{y}_i $
0	2.000000	2.000000	0
0.1	2.035630	2.036225	0.000595
0.2	1.996027	1.996961	0.000934
0.3	1.888453	1.889501	0.001049
0.4	1.722418	1.723401	0.000982
0.5	1.509094	1.509885	0.000790
0.6	1.260684	1.261217	0.000534
0.7	0.989789	0.990063	0.000274
0.8	0.708801	0.708869	0.000068
0.9	0.429343	0.429308	0.000035
1.0	0.161790	0.161790	0

$$\|y - \tilde{y}\| = \max_{0 \leq i \leq n} |y_i - \tilde{y}_i| = 0.001049$$

6. Вывод

В ходе выполнения лабораторной работы были реализованы методы приближенного численного решения краевой задачи для дифференциального уравнения второго порядка: метод прогонки и метод стрельбы.

Вследствие сравнения результатов работы методов сделан вывод о более высокой точности метода прогонки в отличие от метода стрельбы. Такой результат, прежде всего, связан с присутствием случайных значений в методе стрельбы ($D_0 = a + O(h), D_1 = O(h)$), в то время как в методе прогонки их нет. Кроме того вычислительная погрешность обусловлена малым количеством разбиений рассматриваемого отрезка. С увеличением числа разбиений, погрешность уменьшается.