МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

имени М.В.Ломоносова

Факультет вычислительной математики и кибернетики

Компьютерный практикум по учебному курсу

«ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ»

ЗАДАНИЕ № 2.

Численные методы решения дифференциальных уравнений

ОТЧЕТ

о выполненном задании

студента 204 учебной группы факультета ВМК МГУ

Гаухова Владислава Константиновича

гор. Москва

2020 г.

Оглавление

Подвариант 1

Постановка задачи

Рассматривается обыкновенное дифференциальное уравнение первого порядка, разрешенное относительно производной и имеющее вид:

$$\frac{dy}{dx} = f(x,y),\ \ x_0 < x, \tag{1}$$

с дополнительным начальным условием, заданным в точке $x = x_0$:

$$y(x_0) = y_0. \tag{2}$$

Предполагается, что правая часть уравнения (1) функция $f = f(x,y)$ такова, что гарантирует существование и единственность решения задачи Коши (1)-(2).

В том случае, если рассматривается не одно дифференциальное уравнение вида (1), а система обыкновенных дифференциальных уравнений первого порядка, разрешенных относительно производных неизвестных функций, то соответствующая задача Коши имеет вид (на примере двух дифференциальных уравнений):

$$\begin{cases} \dfrac{dy_1}{dx} = f_1(x,y_1,y_2), \\ \dfrac{dy_2}{dx} = f_2(x,y_1,y_2),\ \ x > x_0. \end{cases} \tag{3}$$

Дополнительные (начальные) условия задаются в точке $x = x_0$:

$$y_1(x_0) = y_1^{(0)},\ \ y_2(x_0) = y_2^{(0)}. \tag{4}$$

Также предполагается, что правые части уравнений из (3) заданы так, что это гарантирует существование и единственность решения задачи Коши (3)-(4), но уже для системы обыкновенных дифференциальных уравнений первого порядка в форме, разрешенной относительно производных неизвестных функций.

Цели практической работы:

1)Реализовать функции для решения следующих задач: Решение задачи Коши для обыкновенного дифференциального уравнения первого порядка и системы ОДУ первого порядка методом Рунге-Кутта второго и четвёртого порядка точности и численный расчёт конечно-разностного уравнения;

2)Найти численное решение задачи Коши и построить его график;

3)Найденное численное решение сравнить с точным решением дифференциального уравнения.

Описание алгоритмов

Метод Рунге-Кутта

Решение задачи Коши предполагает нахождение приближённого значения функции в точке за счёт разложения функции в ряд и аппроксимации значения этого ряда. Пусть m – число точек разбиения отрезка, на котором решается поставленная задача, $h = \frac{a-b}{m}$ – диаметр разбиения, $x_i = a + ih$, $y_i = y(x_i)$, $i = \overline{1, n}$ – сетка и сеточная функция.

Метод Рунге-Кутта второго порядка точности для рекуррентного вычисления сеточной функции в случае системы ОДУ первого порядка примет следующий вид:

$$\begin{cases} y_{i+1} = y_i + \frac{h}{2}(k_1 + k_2) \\ k_1 = f(x_i, y_i) \\ k_2 = f(x_i, y_i + hk_1) \end{cases}$$

Аналогичный вид имеет метод Рунге-Кутта четвёртого порядка точности:

$$\begin{cases} y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) \\ k_1 = f(x_i, y_i) \\ k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right) \\ k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right) \\ k_4 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_3\right) \end{cases}$$

Пусть дана система дифференциальных уравнений с начальными условиями:

$$\begin{cases} \mathrm{u}'(\mathrm{x}) = \mathrm{f1}(\mathrm{x}, \mathrm{u}(\mathrm{x}), \mathrm{v}(\mathrm{x})) \\ \mathrm{v}'(\mathrm{x}) = \mathrm{f2}(\mathrm{x}, \mathrm{u}(\mathrm{x}), \mathrm{v}(\mathrm{x})) \\ \mathrm{u}(\mathrm{x0}) = \mathrm{u0} \\ \mathrm{v}(\mathrm{x0}) = \mathrm{v0} \end{cases}$$

Для ее решения воспользуемся следующими рекуррентными соотношениями схемы Рунге-Кутта:

2 порядка:

$$\begin{cases} u_{i+1} = u_i + h * \dfrac{k_1 + k_2}{2} \\ v_{i+1} = v_i + h * \dfrac{m_1 + m_2}{2} \\ k_1 = f_1(x_i, u_i, v_i) \\ k_2 = f_1(x_{i+1}, u_i + h * k_1, v_i + h * m_1) \\ m_1 = f_2(x_i, u_i, v_i) \\ m_2 = f_2(x_{i+1}, u_i + h * k_1, v_i + h * m_1) \end{cases}$$

4 порядка:

$$\begin{cases} u_{i+1} = u_i + h * \dfrac{k_1 + 2k_2 + 2k_3 + k_4}{6} \\ v_{i+1} = v_i + h * \dfrac{m_1 + 2m_2 + 2m_3 + m_4}{6} \\ k_1 = f_1(x_i, u_i, v_i) \\ k_2 = f_1(x_i + h/2, u_i + h * k_1/2, v_i + h * m_1/2) \\ k_3 = f_1(x_i + h/2, u_i + h * k_2/2, v_i + h * m_2/2) \\ k_4 = f_1(x_i + h, u_i + h * k_3, v_i + h * m_3) \\ m_1 = f_2(x_i, u_i, v_i) \\ m_2 = f_2\left(x_i + \dfrac{h}{2}, u_i + \dfrac{h * k_1}{2}, v_i + \dfrac{h * m_1}{2}\right) \\ m_3 = f_2\left(x_i + \dfrac{h}{2}, u_i + \dfrac{h * k_2}{2}, v_i + \dfrac{h * m_2}{2}\right) \\ m_4 = f_2(x_i + h, u_i + h * k_3, v_i + h * m_3) \end{cases}$$

Тестирование

№1

y’=(x-x^2)y, y(0)=1

Аналитическое решение: $y = e^{(3x^2-2x^3)/6}$

Решение методом Рунге-Кутта на (0;1] при n = 10

2 порядок:

x = 0.1000 y = 1.0045 correct = 1.0047

x = 0.2000 y = 1.0171 correct = 1.0175

x = 0.3000 y = 1.0361 correct = 1.0367

x = 0.4000 y = 1.0597 correct = 1.0604

x = 0.5000 y = 1.0860 correct = 1.0869

x = 0.6000 y = 1.1129 correct = 1.1140

x = 0.7000 y = 1.1382 correct = 1.1396

x = 0.8000 y = 1.1595 correct = 1.1611

x = 0.9000 y = 1.1740 correct = 1.1759

x = 1.0000 y = 1.1793 correct = 1.1814

Погрешность: 0.0010677

4 порядок:

x = 0.1000 y = 1.0047 correct = 1.0047

x = 0.2000 y = 1.0175 correct = 1.0175

x = 0.3000 y = 1.0367 correct = 1.0367

x = 0.4000 y = 1.0604 correct = 1.0604

x = 0.5000 y = 1.0869 correct = 1.0869

x = 0.6000 y = 1.1140 correct = 1.1140

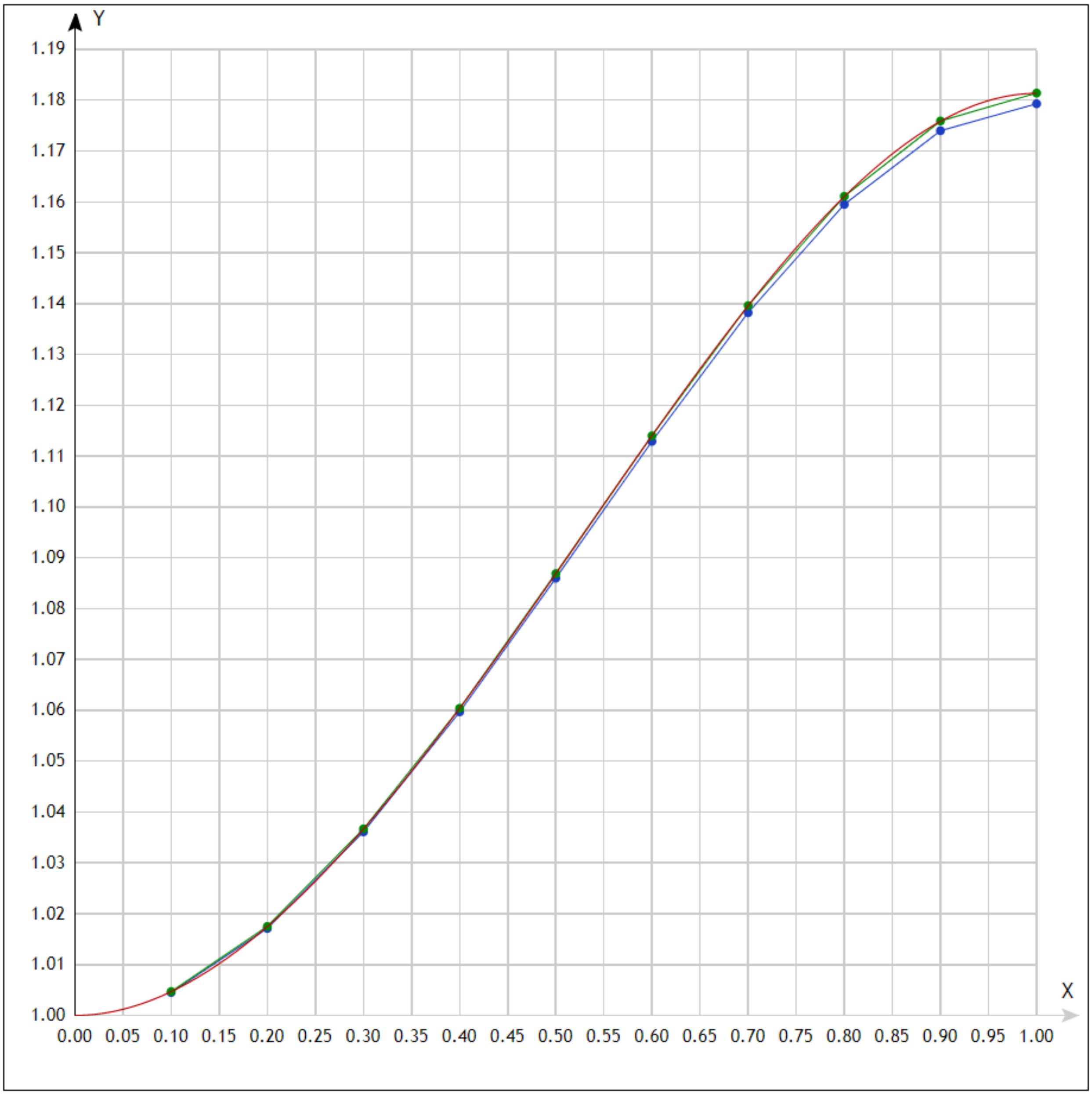
x = 0.7000 y = 1.1396 correct = 1.1396

x = 0.8000 y = 1.1611 correct = 1.1611

x = 0.9000 y = 1.1759 correct = 1.1759

x = 1.0000 y = 1.1814 correct = 1.1814

Погрешность: 0.000000



Синяя – 2 порядок; Зелёная – 4 порядок; Красная - аналитическое решение

№2

y'=x + sin(x), y(0) = 1

Аналитическое решение: y = 2 + x^2/2 – cos(x)

Решение методом Рунге-Кутта на (0;1] при n = 10

2 порядок

x = 0.1000 y = 1.0100 correct = 1.0100

x = 0.2000 y = 1.0399 correct = 1.0399

x = 0.3000 y = 1.0896 correct = 1.0897

x = 0.4000 y = 1.1589 correct = 1.1589

x = 0.5000 y = 1.2473 correct = 1.2474

x = 0.6000 y = 1.3545 correct = 1.3547

x = 0.7000 y = 1.4800 correct = 1.4802

x = 0.8000 y = 1.6230 correct = 1.6233

x = 0.9000 y = 1.7831 correct = 1.7834

x = 1.0000 y = 1.9593 correct = 1.9597

Погрешность: 0.0001519

4 порядок:

x = 0.1000 y = 1.0100 correct = 1.0100

x = 0.2000 y = 1.0399 correct = 1.0399

x = 0.3000 y = 1.0897 correct = 1.0897

x = 0.4000 y = 1.1589 correct = 1.1589

x = 0.5000 y = 1.2474 correct = 1.2474

x = 0.6000 y = 1.3547 correct = 1.3547

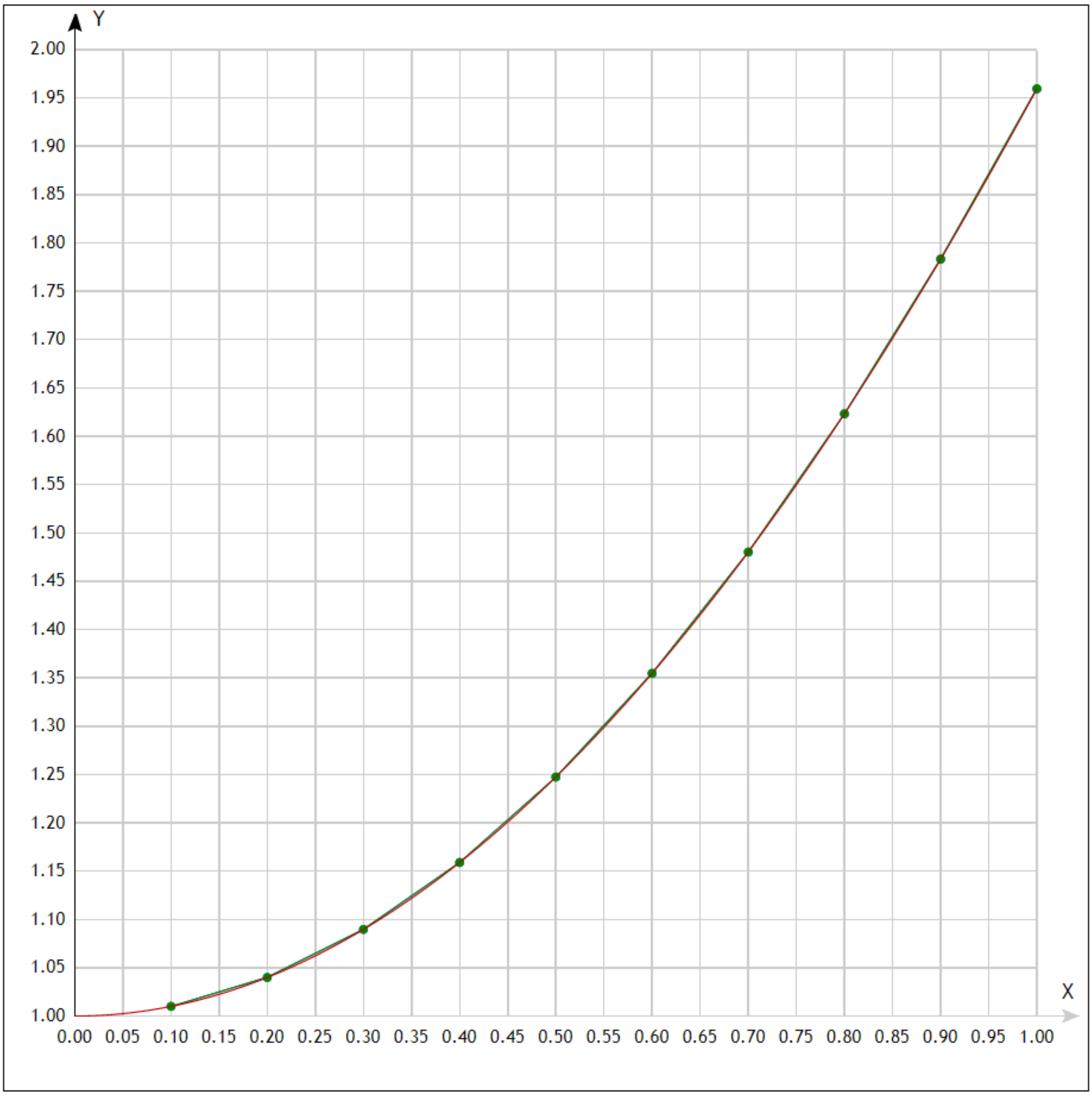
x = 0.7000 y = 1.4802 correct = 1.4802

x = 0.8000 y = 1.6233 correct = 1.6233

x = 0.9000 y = 1.7834 correct = 1.7834

x = 1.0000 y = 1.9597 correct = 1.9597

Погрешность: 0.000000



Синяя – 2 порядок; Зеленая – 4 порядок; Красная – аналитическое решение

№3

y’=x + sin(x), y(0) = 1

Аналитическое решение: y = 2 + x^2/2 – cos(x)

Решение методом Рунге-Кутта на (0;10] при n = 10

2 порядок:

x = 1.0000 y = 1.9207 correct = 1.9597

x = 2.0000 y = 4.2961 correct = 4.4161

x = 3.0000 y = 7.3213 correct = 7.4900

x = 4.0000 y = 10.5135 correct = 10.6536

x = 5.0000 y = 14.1556 correct = 14.2163

x = 6.0000 y = 19.0365 correct = 19.0398

x = 7.0000 y = 25.7252 correct = 25.7461

x = 8.0000 y = 34.0484 correct = 34.1455

x = 9.0000 y = 43.2492 correct = 43.4111

x = 10.0000 y = 52.6832 correct = 52.8391

Погрешность: 0.0967699

4 порядок:

x = 1.0000 y = 1.9599 correct = 1.9597

x = 2.0000 y = 4.4167 correct = 4.4161

x = 3.0000 y = 7.4907 correct = 7.4900

x = 4.0000 y = 10.6542 correct = 10.6536

x = 5.0000 y = 14.2166 correct = 14.2163

x = 6.0000 y = 19.0398 correct = 19.0398

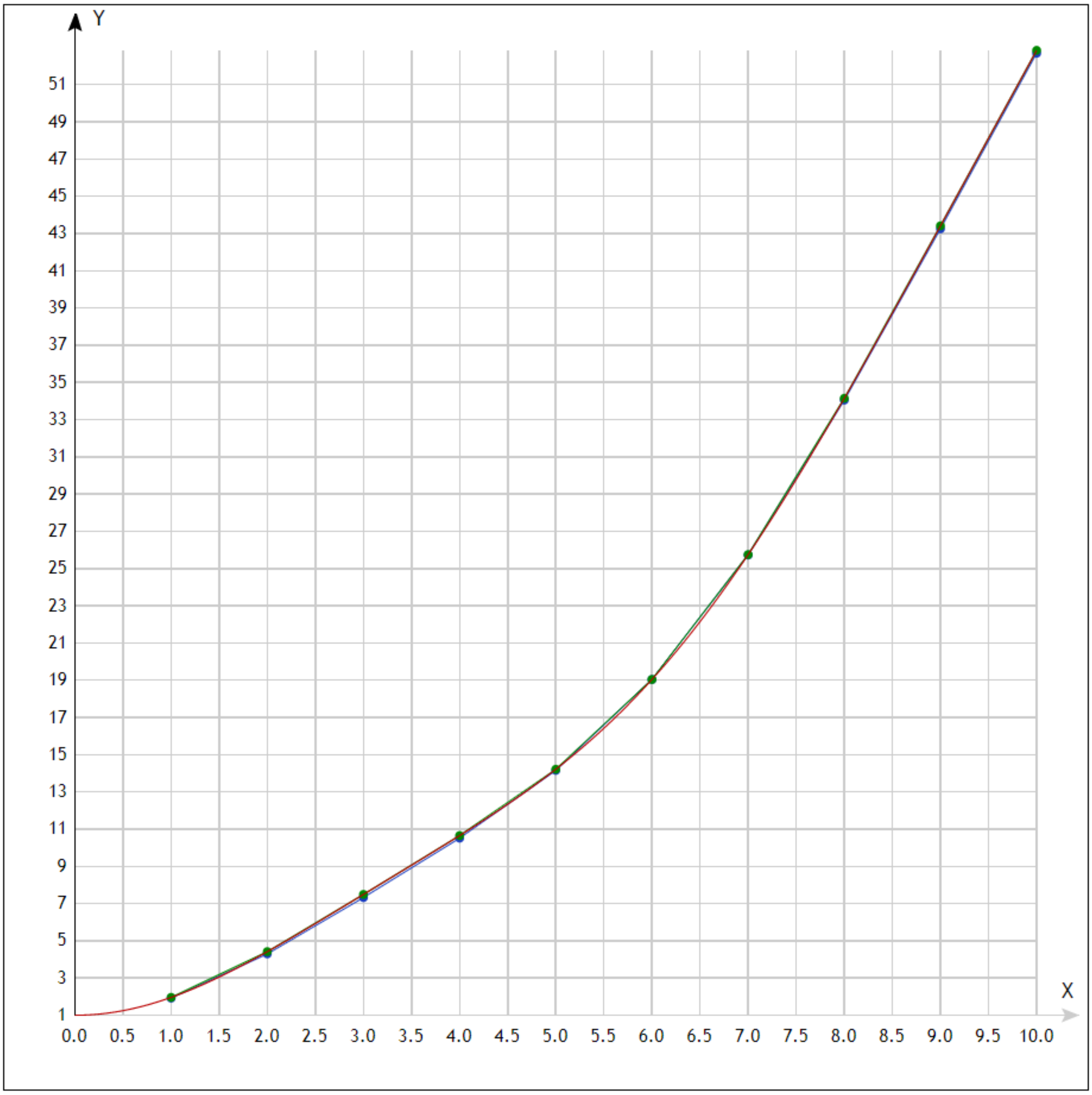
x = 7.0000 y = 25.7462 correct = 25.7461

x = 8.0000 y = 34.1459 correct = 34.1455

x = 9.0000 y = 43.4118 correct = 43.4111

x = 10.0000 y = 52.8397 correct = 52.8391

Погрешность: 0.0004086



Синяя – 2 порядок; Зеленая – 4 порядок; Красная – аналитическое решение

№4

$$\begin{cases} f_1(x,u,v) = \cos(1.5x - v) - u \\ f_2(x,u,v) = -v^2 + 2.3u - 1.2 \end{cases}$$

Решение не найдено (Wolfram Alpha)

2 порядок:

x = 0.1000 u = 0.2307 v = 0.8492

x = 0.2000 u = 0.2233 v = 0.7195

x = 0.3000 u = 0.2241 v = 0.6066

x = 0.4000 u = 0.2301 v = 0.5076

x = 0.5000 u = 0.2390 v = 0.4199

x = 0.6000 u = 0.2492 v = 0.3414

x = 0.7000 u = 0.2594 v = 0.2705

x = 0.8000 u = 0.2687 v = 0.2056

x = 0.9000 u = 0.2766 v = 0.1453

x = 1.0000 u = 0.2826 v = 0.0882

4 порядок:

x = 0.1000 u = 0.2298 v = 0.8495

x = 0.2000 u = 0.2218 v = 0.7198

x = 0.3000 u = 0.2223 v = 0.6068

x = 0.4000 u = 0.2280 v = 0.5075

x = 0.5000 u = 0.2369 v = 0.4195

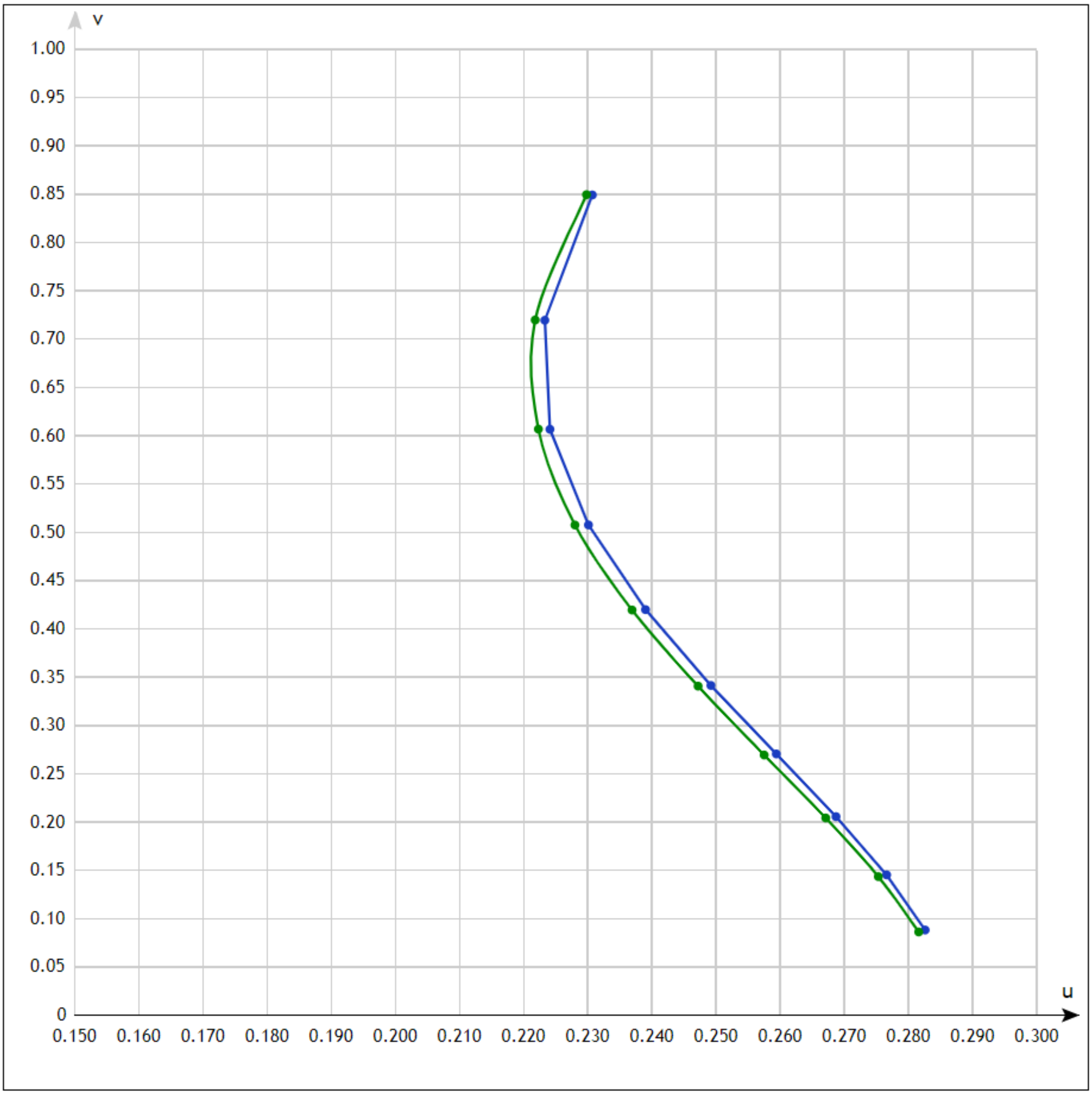
x = 0.6000 u = 0.2472 v = 0.3407

x = 0.7000 u = 0.2575 v = 0.2694

x = 0.8000 u = 0.2671 v = 0.2042

x = 0.9000 u = 0.2753 v = 0.1435

x = 1.0000 u = 0.2816 v = 0.0862



Синая – 2 порядок; зелёная – 4 порядок

№5

$$\begin{cases} f_1(x,u,v) = tg(u) \\ f_2(x,u,v) = v \end{cases}$$

Аналитическое решение:

$$\begin{cases} u = arcsin(e^x * sin\left(\frac{1}{4}\right)) \\ v = e^x \end{cases}$$

2 порядок:

x = 0.1000 u = 0.2769 correct u =0.2770 v = 1.1050 correct v = 1.1052

x = 0.2000 u = 0.3069 correct u =0.3070 v = 1.2210 correct v = 1.2214

x = 0.3000 u = 0.3403 correct u =0.3405 v = 1.3492 correct v = 1.3499

x = 0.4000 u = 0.3778 correct u =0.3780 v = 1.4909 correct v = 1.4918

x = 0.5000 u = 0.4198 correct u =0.4202 v = 1.6474 correct v = 1.6487

x = 0.6000 u = 0.4671 correct u =0.4677 v = 1.8204 correct v = 1.8221

x = 0.7000 u = 0.5208 correct u =0.5215 v = 2.0116 correct v = 2.0138

x = 0.8000 u = 0.5821 correct u =0.5831 v = 2.2228 correct v = 2.2255

x = 0.9000 u = 0.6529 correct u =0.6542 v = 2.4562 correct v = 2.4596

x = 1.0000 u = 0.7358 correct u =0.7376 v = 2.7141 correct v = 2.7183

Погрешность: 0.0023930

4 порядок

x = 0.1000 u = 0.2770 correct u = 0.2770 v = 1.1052 correct v =1.1052

x = 0.2000 u = 0.3070 correct u = 0.3070 v = 1.2214 correct v =1.2214

x = 0.3000 u = 0.3405 correct u = 0.3405 v = 1.3499 correct v =1.3499

x = 0.4000 u = 0.3780 correct u = 0.3780 v = 1.4918 correct v =1.4918

x = 0.5000 u = 0.4202 correct u = 0.4202 v = 1.6487 correct v =1.6487

x = 0.6000 u = 0.4677 correct u = 0.4677 v = 1.8221 correct v =1.8221

x = 0.7000 u = 0.5215 correct u = 0.5215 v = 2.0138 correct v =2.0138

x = 0.8000 u = 0.5831 correct u = 0.5831 v = 2.2255 correct v =2.2255

x = 0.9000 u = 0.6542 correct u = 0.6542 v = 2.4596 correct v =2.4596

x = 1.0000 u = 0.7376 correct u = 0.7376 v = 2.7183 correct v =2.7183

Погрешность 0.000011

Выводы

В ходе данной практической работы были реализованы методы Рунге-Кутта 2-го и 4-го порядков для решения задачи Коши с дифференциальным уравнением первого порядка. Численные методы позволяют с достаточно высокой точностью получать решения ОДУ и систем ОДУ. Также достоинством методов является простота их реализации и высокая скорость вычислений. При тестировании было показано, что для более точного нахождения решения предпочтительнее использовать метод 4-го порядка.

Код программы

Метод Рунге-Кутта 2 порядка

```
void rk2(double x0, double y0, double h, int n, double (*f)())
{
    double y = y0;
    for (int i = 1; i <= n; i++) {
        double x = x0 + i * h;
        double k1 = f(x, y);
        double k2 = f(x, y + h * k1);
        y += (k1 + k2) * h / 2;
        printf("%.4lf %.4lf\n", x, y);
    }
}
```

Метод Рунге-Кутта 4 порядка

```
void rk4(double x0, double y0, double h, int n, double (*f)())
{
    double y = y0;
    for (int i = 1; i <= n; i++) {
        double x = x0 + i * h;
        double k1 = f(x, y);
        double k2 = f(x + h/2, y + h*k1 / 2);
        double k3 = f(x + h/2, y + h*k2 / 2);
        double k4 = f(x + h/2, y + h*k3 / 2);
        y += (k1 + 2*k2 + 2*k3 + k4) * h / 6;
        printf("%.4lf %.4lf\n", x, y)
    }
}
```

Метод Рунге-Кутта 2 порядка для системы

```
void rk2_sys(double x0, double u0, double v0, double h, int n, double (*f1)(), double (*f2)())
{
    double u = u0;
    double v = v0;
    for (int i = 1; i <= n; i++) {
```

```
        double x = x0 + i * h;
        double k1 = f1(x, u, v);
        double m1 = f2(x, u, v);
        double k2 = f1(x + h, u + h * k1, v + h * m1);
        double m2 = f2(x + h, u + h * k1, v + h * m1);
        u += (k1 + k2) * h / 2;
        v += (m1 + m2) * h / 2;
        printf("%.4lf %.4lf %.4lf\n", x, u, v);
    }
}
```

Метод Рунге-Кутта 4 порядка для системы

```
void rk4_sys(double x0, double u0, double v0, double h, int n, double (*f1)(), double (*f2)())
{
    double u = u0;
    double v = v0;
    for (int i = 1; i <= n; i++) {
        double x = x0 + i * h;
        double k1 = f1(x, u, v);
        double m1 = f2(x, u, v);
        double k2 = f1(x + h/2, u + h * k1/2, v + h * m1/2);
        double m2 = f2(x + h/2, u + h * k1/2, v + h * m1/2);
        double k3 = f1(x + h/2, u + h * k2/2, v + h * m2/2);
        double m3 = f2(x + h/2, u + h * k2/2, v + h * m2/2);
        double k4 = f1(x + h, u + h * k3, v + h * m3);
        double m4 = f2(x + h, u + h * k3, v + h * m3);
        u += (k1 + 2*k2 + 2*k3 + k4) * h / 6;
        v += (m1 + 2*m2 + 2*m3 + m4) * h / 6;
        printf("%.4lf %.4lf %.4lf\n", x, u, v);
    }
}
```

Подвариант №2

Постановка задачи

Рассматривается линейное дифференциальное уравнение II порядка вида:

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

с дополнительными условиями в граничных точках:

$$\begin{cases}\sigma_1 y(a) + \gamma_1'(a) = \delta_1 \\ \sigma_2 y(b) + \gamma_2'(b) = \delta_2\end{cases}$$

Цели практической работы:

1. Решить краевую задачу методом конечных разностей, аппроксимировав ее разностной схемой второго порядка точности (на равномерной сетке), полученную систему конечно-разностных уравнений решить методом прогонки.
2. Найти разностное решение и построить его график.
3. Найденное разностное решение сравнить с точным решением дифференциального уравнения, полученным при помощи онлайн системы Wolfram Alpha.

Описание алгоритмов

Разобьем отрезок [a, b] на сетку из n частей:

$$x_i = a + hi, \qquad h = \frac{b-a}{n}, \quad 0 \le i \le n$$

Приблизим производные в исходном уравнении конечно-разностными отношениями:

$$y''(x_i) = \frac{y(x_{i+1})+1—2y(x_i)+y(x_{i-1})}{h^2}, \quad 1 \le i \le n-1$$

$$y'(x_i) = \frac{y(x_{i+1})-y(x_{i-1})}{2h} \quad 1 \le i \le n-1$$

$$y'(x_0) = \frac{y(x_1)-y(x_0)}{h}$$

$$y'(x_n) = \frac{y(x_n)-y(x_{n-1})}{h}$$

$$\frac{y(x_{i+1})+1—2y(x_i)+y(x_{i-1})}{h^2} + p(x_i)\frac{y(x_{i+1})-y(x_{i-1})}{2h} + q(x_i)y(x_i) = f(x_i), \quad 0 \le i < n$$

Приблизим производные в дополнительных условиях:

$$\begin{cases}\sigma_1 y(x_0) + \gamma_1 y'(x_0) = \delta_1, \\ \sigma_2 y(x_n) + \gamma_2 y'(x_n) = \delta_2.\end{cases} \Rightarrow \begin{cases}\sigma_1 y(x_0) + \gamma_1 \frac{y(x_1)-y(x_0)}{h} = \delta_1, \\ \sigma_2 y(x_n) + \gamma_2 \frac{y(x_n)-y(x_{n-1})}{h} = \delta_2.\end{cases}$$

Соберем коэффициенты во всех полученных равенствах при соответствующих $y_i = y(x_i)$ и получим систему линейных уравнений с постоянными коэффициентами имеющую трехдиагональную матрицу. Решим данную систему методом прогонки. Решение будем искать в виде:
$y_{i-1} = \alpha_i y_i + \beta_i$

С прогоночными коэффициентами:
$\alpha_i = \frac{-k_1}{k_2\alpha_{i-1}+k_3}$;
$\beta_i = \frac{f(x_{i-1})-k_2\beta_{i-1}}{k_2\alpha_{i-1}+k_3}$;

$k_1 = \frac{1}{h^2} + \frac{p(x_i)}{2h}$;
$k_2 = \frac{1}{h^2} - \frac{p(x_i)}{2h}$;
$k_3 = -\frac{2}{h^2} + q(x_i)$.

Тестирование

Для тестирования метода конечных разностей и метода прогонки будем использовать тесты, предложенные в варианте 5 и краевую задачу для дополнительного тестирования. Проверку результатов будем проводить через Wolfram Alpha.

№1

$$\begin{cases} y'' + 2y' - xy = x^2 \\ y'(0.6) = 0.7 \\ y(0.9) - 0.5y'(0.9) = 1 \end{cases}$$

Решение не найдено (Wolfram Alpha)

Решение на отрезке [0.6; 0.9] с n = 10:

x = 0.900 y = 1.366

x = 0.870 y = 1.344

x = 0.840 y = 1.323

x = 0.810 y = 1.302

x = 0.780 y = 1.281

x = 0.750 y = 1.260

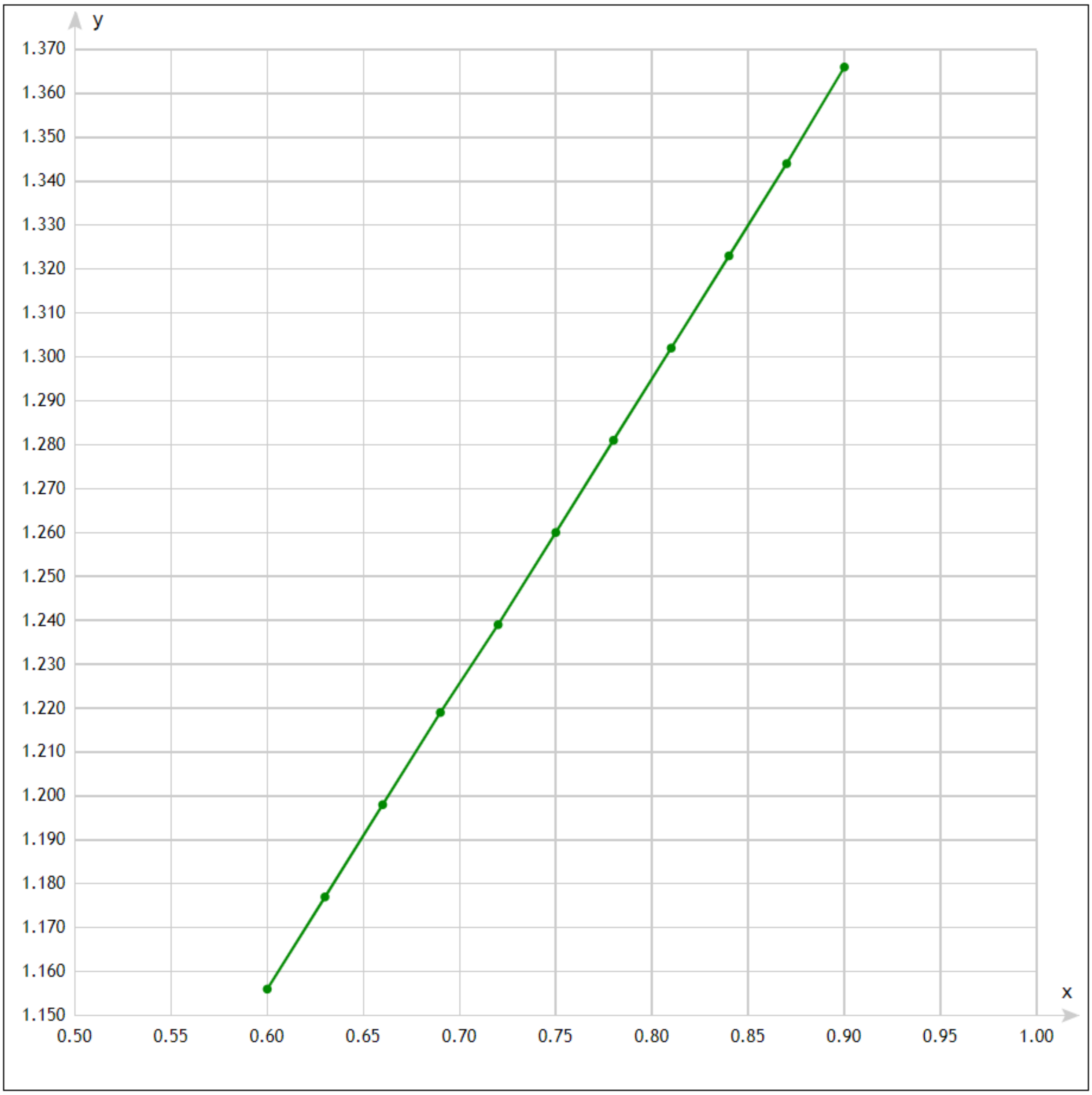
x = 0.720 y = 1.239

x = 0.690 y = 1.219

x = 0.660 y = 1.198

x = 0.630 y = 1.177

x = 0.600 y = 1.156



№2

y''+y'=1, y'(0)=0, y(1)=1

$$\begin{cases} y'' + y' = 1 \\ y'(0) = 0 \\ y(1) = 1 \end{cases}$$

Решение(Wolfram Alpha): $y = x + e^{-x} - \frac{1}{e}$

Решение на отрезке [0; 1] с n = 10:

x = 1.000 y = 1.000 correct = 1.000

x = 0.900 y = 0.941 correct = 0.939

x = 0.800 y = 0.886 correct = 0.881

x = 0.700 y = 0.835 correct = 0.829

x = 0.600 y = 0.790 correct = 0.781

x = 0.500 y = 0.751 correct = 0.739

x = 0.400 y = 0.718 correct = 0.702

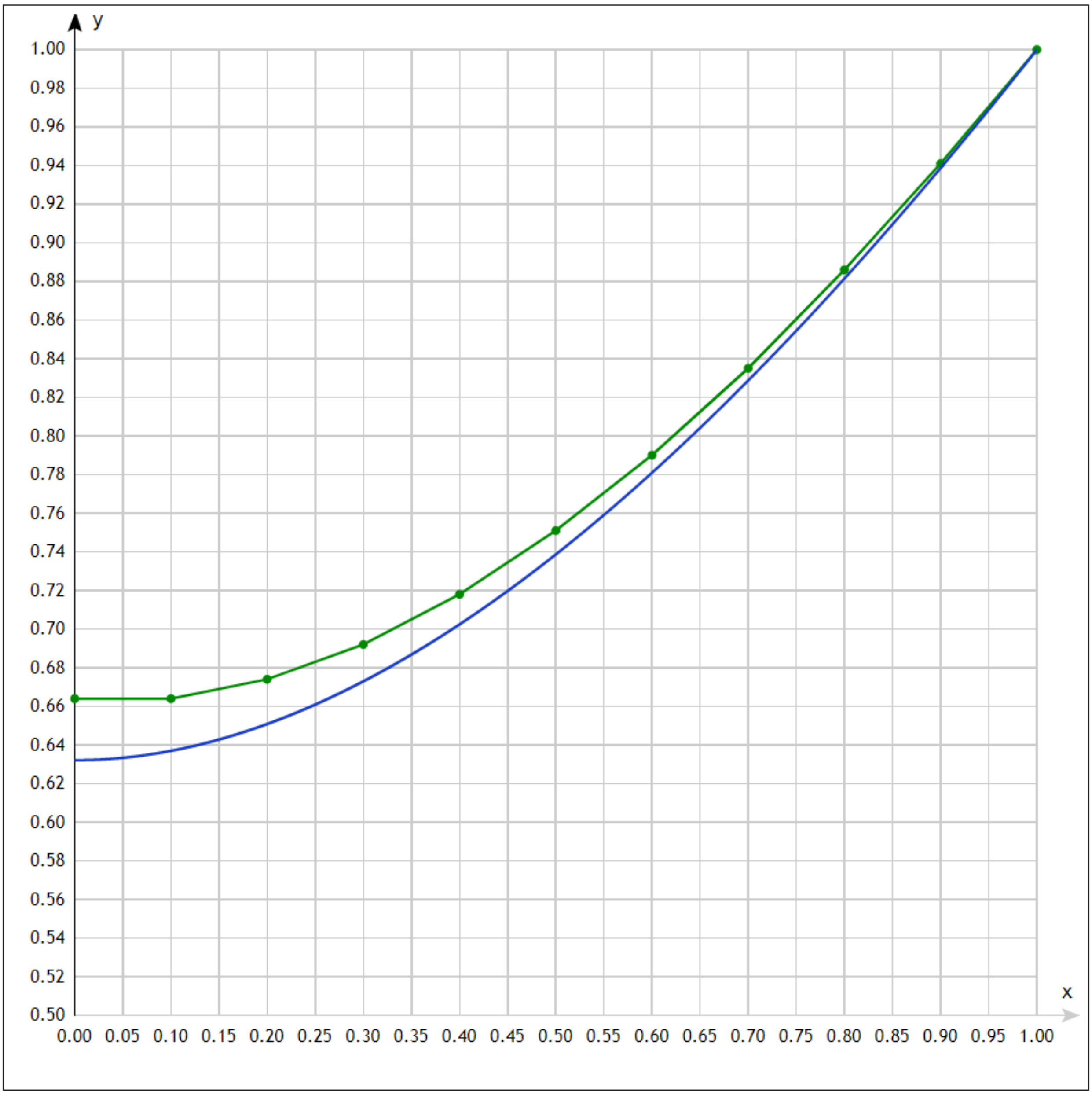
x = 0.300 y = 0.692 correct = 0.673

x = 0.200 y = 0.674 correct = 0.651

x = 0.100 y = 0.664 correct = 0.637

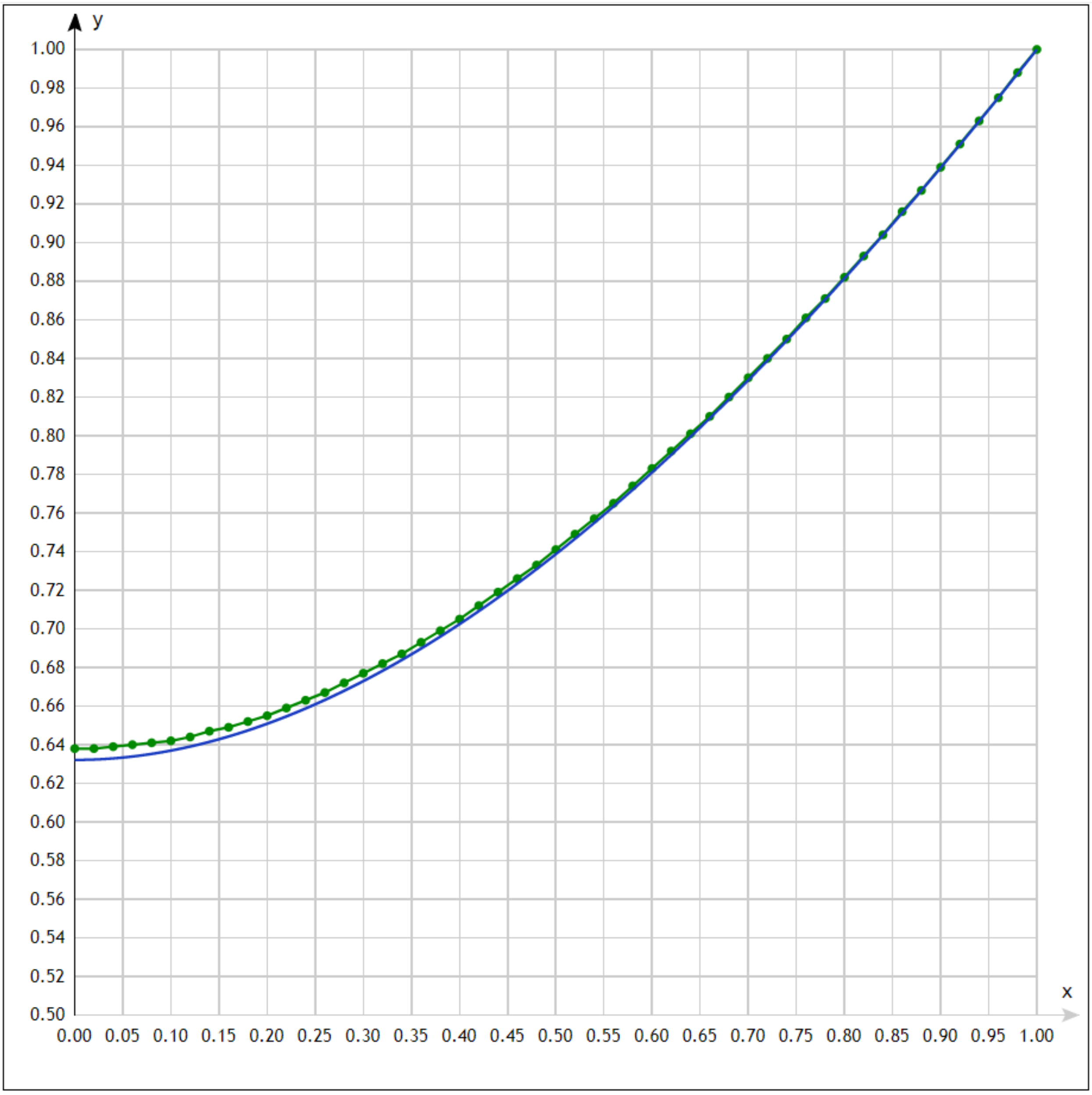
x = 0.000 y = 0.664 correct = 0.632

Погрешность: 0.0149262776



Зеленая - метода конечных разностей и прогонки, синяя – точное решение

Для n=50:



y
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
0.82
0.80
0.78
0.76
0.74
0.72
0.70
0.68
0.66
0.64
0.62
0.60
0.58
0.56
0.54
0.52
0.50
x
0.00 0.05 0.10 0.15 0.20 0.25 0.30 0.35 0.40 0.45 0.50 0.55 0.60 0.65 0.70 0.75 0.80 0.85 0.90 0.95 1.00

Выводы

В данной работе был реализован способ решения краевой задачи методом конечных разностей, полученная система конечно-разностных уравнений была решена методом прогонки. Тестирование показало, что данные методы позволяют с высокой точностью находить решения линейных ДУ данного вида. Также точность растёт с увеличением числа шагов.

Код программы

Метод конечных разностей

```
void method(int n, double a, double b, double sigma1, double sigma2, double gamma1, double gamma2, double delta1, double delta2, double (*p)(), double (*q)(), double (*f)())
{
    double alpha[n], beta[n];
    double h = (b - a) / n;
    double x = a + h;
    alpha[0] = -gamma1/(sigma1*h - gamma1);
    beta[0] = delta1/(sigma1 - gamma1/h);
    for (int i = 1; i < n; i++) {
        double k1 = 1/(h*h) + p(x)/(2*h);
        double k2 = 1/(h*h) - p(x)/(2*h);
        double k3 = -2/(h*h) + q(x);
        alpha[i] = - k1/(k2*alpha[i - 1] + k3);
        beta[i] = (f(x) - beta[i - 1]*k2)/(k2*alpha[i - 1] + k3);
        x += h;
    }
    double y = (delta2*h + gamma2*beta[n - 1]) / (sigma2*h + gamma2 - gamma2*alpha[n - 1]);
    printf("x = %.3lf y = %.3lf correct = %.3lf\n", x, y, correct(x));
    double er = fabs(y - correct(x));
    for (int i = n - 1; i >= 0; i--) {
        x -= h;
        y = alpha[i] * y + beta[i];
        er += fabs(y - correct(x));
        printf("x = %.3lf y = %.3lf correct = %.3lf\n", x, y, correct(x));
    }
    printf("error rate = %.10lf\n", er/n);
}
```