МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
имени М.В. Ломоносова

Факультет вычислительной математики и кибернетики

Компьютерный практикум по курсу «ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ»

ЗАДАНИЕ №2

Численные методы решения дифференциальных уравнений.

ОТЧЕТ

о выполненном задании
студента 205 группы факультета ВМК МГУ
Петрова Ильи Алексеевича

Москва, 2014 г.

<u>Подвариант №1.</u>

Методы Рунге-Кутта второго и четвертого порядка точности для численного решения задачи Коши для дифференциального уравнения (системы дифференциальных уравнений) первого порядка.

Постановка задачи:

Рассматривается ОДУ 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 $f(x,y)$ 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):

$$\begin{cases} \frac{dy_1}{dx} = f_1(x, y_1, y_2) \\ \frac{dy_2}{dx} = f_2(x, y_1, y_2) \end{cases}, \; x_0 < x \tag{3}$$

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

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

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. Заметим, что к подобным задачам сводятся многие важные задачи, возникающие в механике (уравнения движения материальной точки), небесной механике, химической кинетике, гидродинамике и т.п.

Цели и задачи практической работы:

1) Решить задачу Коши (1)-(2) (или (3)-(4)) наиболее известными и широко используемыми на практике методами Рунге-Кутта второго и четвертого порядка точности, аппроксимировав дифференциальную задачу соответствующей разностной схемой (на равномерной сетке); полученное конечно-разностное уравнение (или уравнения в случае системы), представляющее фактически некоторую рекуррентную формулу, просчитать численно;
2) Найти численное решение задачи и построить его график;
3) Найденное численное решение сравнить с точным решением дифференциального уравнения (подобрать специальные тесты, где аналитические решения находятся в классе элементарных функций, при проверке можно использовать ресурсы on-line системы http://www.wolframalpha.com или пакета Mapie и т.п.).

Описание методов решения

1)Метод Рунге-Кутты второго порядка точности

Даны: функция f и (x_0, y_0). Требуется найти аппроксимацию решения задачи Коши (1)(2). Выберем число отрезков разбиения n (чем больше, тем выше точность). Пусть требуется найти решение на отрезке $[a, b], a = x_0$. Тогда длина одного отрезка разбиения $h = (b - a)/n$. Аппроксимация состоит из приближённых значений $y(x_i),\ i = \overline{0, n}$. Они находятся с помощью рекуррентной формулы:

$$k_1 = f(x_i, y_i)$$
$$k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} * k_1\right)$$
$$y_{i+1} = y_i + h * k_2$$

Для системы (3)(4):

$$k_1 = f_1\left(x_i, y_1^{(i)}, z_i\right) \qquad l_1 = f_2\left(x_i, y_1^{(i)}, y_2^{(i)}\right)$$
$$k_2 = f_1\left(x_{i+1}, y_1^{(i)} + hk_1, z_i + hl_1\right) \qquad l_2 = f_2\left(x_{i+1}, y_1^{(i)} + hk_1, y_2^{(i)} + hl_i\right)$$
$$y_1^{(i+1)} = y_1^{(i)} + \frac{h}{2}(k_1 + k_2) \qquad y_2^{(i+1)} = y_2^{(i)} + \frac{h}{2}(l_1 + l_2)$$

Можно доказать («Вводные лекции по численным методам», стр. 162), что погрешность решения зависит от h как $O(h^2)$ при условии, что f (f_1, f_2) имеет непрерывные вторые частные производные.

2)Метод Рунге-Кутты четвёртого порядка точности

Производит вычисления с более высокой точностью $O(h^4)$

$$
\begin{aligned}
&k_1 = f(x_i, y_i)\\
&k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_1\right)\\
&k_3 = f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2}k_2\right)\\
&k_4 = f(x_i + h, y_i + hk_3)\\
&y_{i+1} = y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)
\end{aligned}
$$

Для системы (3)(4):

$$
\begin{aligned}
&k_1 = f_1\left(x_i, y_1^{(i)}, y_2^{(i)}\right) && l_1 = f_2\left(x_i, y_1^{(i)}, y_2^{(i)}\right)\\
&k_2 = f_1\left(x_i + \frac{h}{2}, y_1^{(i)} + \frac{h}{2}k_1, y_2^{(i)} + \frac{h}{2}l_1\right) && l_2 = f_2\left(x_i + \frac{h}{2}, y_1^{(i)} + \frac{h}{2}k_1, y_2^{(i)} + \frac{h}{2}l_1\right)\\
&k_3 = f_1\left(x_i + \frac{h}{2}, y_1^{(i)} + \frac{h}{2}k_2, y_2^{(i)} + \frac{h}{2}l_2\right) && l_3 = f_2\left(x_i + \frac{h}{2}, y_1^{(i)} + \frac{h}{2}k_2, y_2^{(i)} + \frac{h}{2}l_2\right)\\
&k_4 = f_1\left(x_i + h, y_1^{(i)} + hk_3, y_2^{(i)} + hl_3\right) && l_4 = f_2\left(x_i + h, y_1^{(i)} + hk_3, y_2^{(i)} + hl_3\right)\\
&y_1^{(i+1)} = y_1^{(i)} + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4) && y_2^{(i+1)} = y_2^{(i)} + \frac{h}{6}(l_1 + 2l_2 + 2l_3 + l_4)
\end{aligned}
$$

Описание программы

Программа реализует метод Рунге — Кутта 2-ого и 4-ого порядка точности для заранее заданного уравнения или системы. Порядок точности и тип решаемой задачи (одно уравнение или система) задаются пользователем в начале работы. Программа выводит результат в виде: x: <точка>counted: <вычисленное знач.> given: <точное решение>. Функции правой части задаются непосредственно в коде программы в функции func(x,y) для одного уравнения и в функциях sys_1(x,u,v) и sys_2(x,u,v) для системы из двух уравнений. Начальные условия задаются в виде констант Y0 для одного уравнения и U0, V0 для системы (левая граница задается константой LEFT). Точное решение для уравнения задается в функции sol(x).

Тесты

В таблицах 1) – метод Рунге-Кутты второго, 2) - четвёртого порядка точности.

I. <u>Таблица 1, Вариант 3</u>

$$\frac{dy}{dx} = -y - x^2, (x_0, y_0) = (0, 10)$$

точное решение:

$$y(x) = -x^2 + 2x - 2 + 12e^{-x}$$

n	Отрезок	Погрешность 1)	Погрешность 2)
10	[0;10]	0.74706	0.30361
100	[0;10]	0.00402	0.000002
1000	[0;10]	0.000038	0.0000

Примечание: графики находятся в Приложении 1.

II. Таблица 1, Вариант 2

$$\frac{dy}{dx} = \sin(x) - y, (x_0, y_0) = (0, 10)$$

точное решение:

$$y(x) = -0.5 * \cos(x) + 0.5 * \sin(x) + 12.5 * e^{-x}$$

n	Отрезок	Погрешность 1)	Погрешность 2)
10	[0;10]	0.535774	0.099155
100	[0;10]	0.054583	0.052540
1000	[0;10]	0.050263	0.050248

Примечание: графики находятся в Приложении 1.

III. Таблица 2б Вариант 17

$$\begin{cases} \frac{dy_1}{dx} = \sin(1.4 * y_1^2) - x + y_2 \\ \frac{dy_2}{dx} = x + y_1 - 2.2 * y_2^2 + 1 \end{cases}$$

n	Отрезок	Погрешность 1)	Погрешность 2)
10	[0;10]	N/A	N/A
100	[0;10]	N/A	N/A
1000	[0;10]	N/A	N/A

Погрешность не подсчитывалась для этого теста с системой ОДУ, т.к. для него решения в классе элементарных функций не существует.

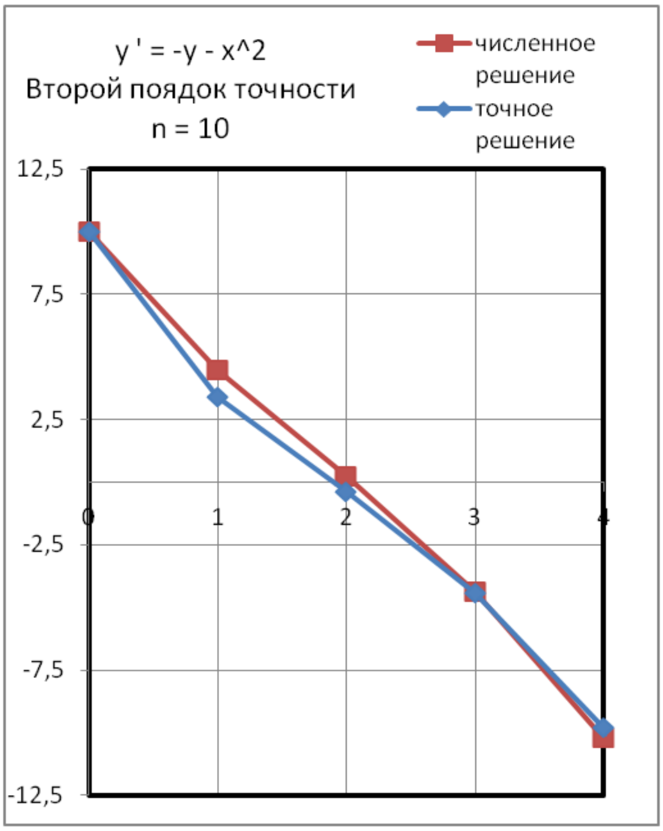
Вывод

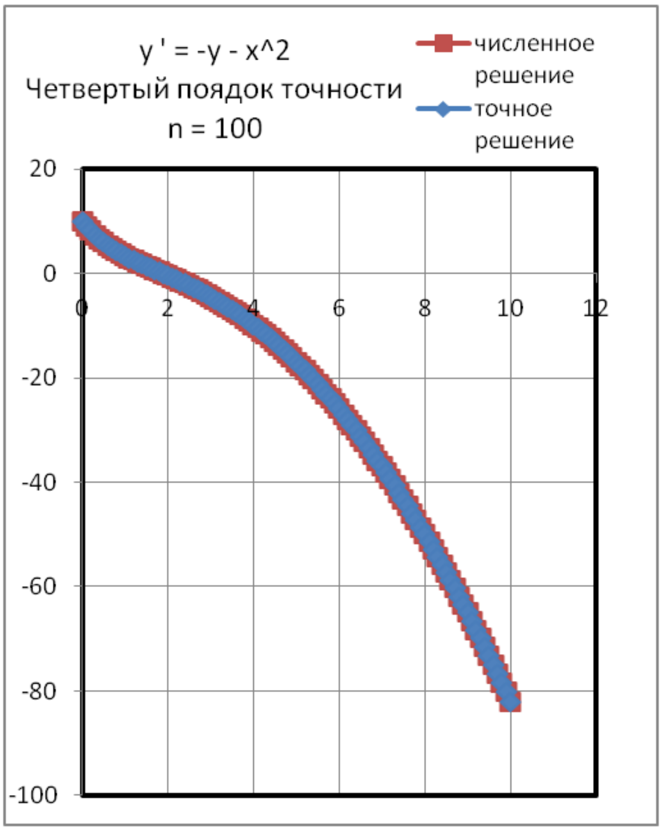
В ходе практической работы были реализованы метод Рунге-Кутты второго и четвёртого порядков точности, применительно как к «простым» ОДУ первого порядка, разрешённым относительно производной, так и к соответствующим системам.

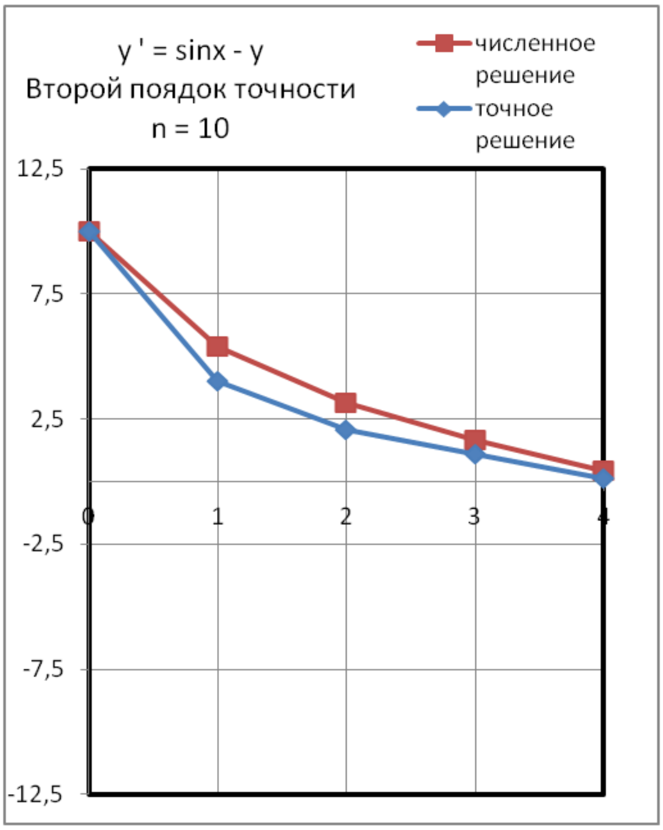
Тестирование показало, что метод Рунге-Кутты четвёртого порядка точности действительно намного более точный, чем метод второго порядка точности. В то время, как второму хватает 100 итераций для получения приемлемого результата, первому требуется порядка 10000 итераций.

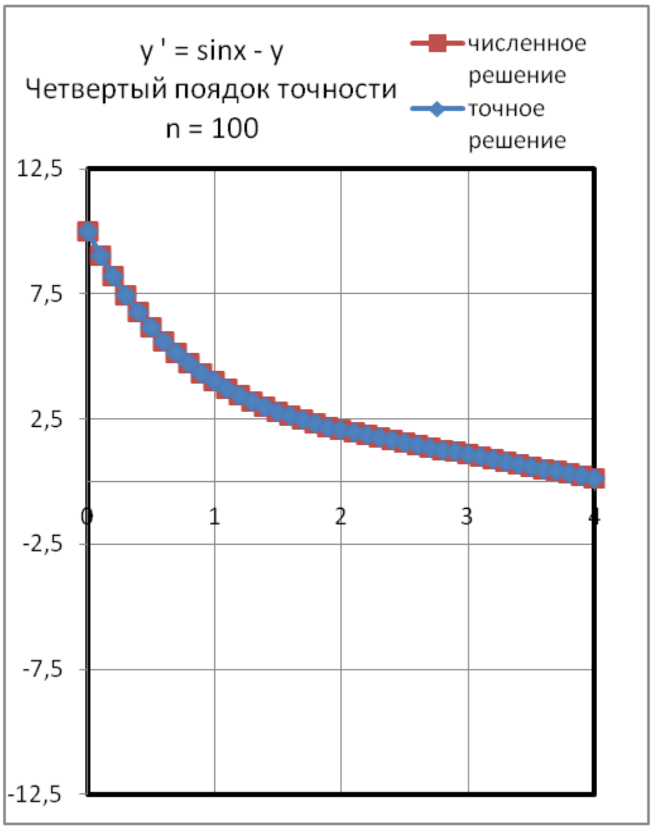
В применении к системам из двух ОДУ первого порядка, метод Рунге-Кутты четвёртого порядка показывает ещё большее преимущество.

Приложение 1:









<u>Подвариант №2.</u>

Решение краевой задачи для ОДУ второго порядка, разрешенного относительно старшей производной.

Постановка задачи:

Рассматривается линейное дифференциальное уравнение второго порядка вида:

$$y'' + p(x) * y' + q(x) * y = -r(x),\ 0 < x < 1 \qquad (1)$$

с дополнительными условиями в граничных точках:

$$\begin{cases} \sigma_1 * y(0) + \gamma_1 * y'(0) = \delta_1, \\ \sigma_2 * y(0) + \gamma_2 * y'(0) = \delta_2, \end{cases} (2)$$

Задачи практической работы

1) Решить краевую задачу (1)-(2) методом конечных разностей, аппроксимировав её разностной схемой второго порядка точности (на равномерной сетке); полученную систему конечно-разностных уравнений решить методом прогонки.
2) Найти разностное решение задачи и построить её график.
3) Найденное решение сравнить с точным решением дифференциального уравнениям (подобрать специальные тесты, где аналитические решения находятся в классе элементарных функций, при проверке можно использовать ресурсы on-line системы http://www.wolfra malpha.com или пакета Mapie и т.п.).

Описание метода:

Перейдем от исходного уравнения к разностному:

$$\frac{y_{i+1} - 2y_i + y_{i-1}}{h^2} + p(x_i)\frac{y_{i+1} - y_{i-1}}{2h} + q(x_i)y_i = -r(x_i)$$

Приведем уравнение к трехдиагональному виду (выразим y):

$$a_i y_{i-1} - b_i y_i + c_i y_{i+1} = d_i$$

где:

$$a_i = \frac{1}{h^2} - \frac{p(x_i)}{2h}, b_i = \frac{2}{h^2} - q(x_i), c_i = \frac{1}{h^2} + \frac{p(x_i)}{2h}, d_i = r_i$$

Получившуюся СЛАУ удобно решить методом прогонки (т.к. матрица системы - трехдиагональная). Для начала требуется определить прогоночные коэффициенты (это прямой ход метода):

$$\alpha_{i+1} = \frac{c_i}{b_i - a_i\alpha_i},$$

$$\beta_{i+1} = \frac{\beta_i a_i - d_i}{b_i - a_i\alpha_i},$$

Начальные значения определяются, исходя из краевых условий:

$$\alpha_1 = \frac{-\gamma_1}{\sigma_1 h - \gamma_1}$$

$$\beta_1 = \frac{\delta_1 h}{\sigma_1 h - \gamma_1}$$

После вычисления всех коэффициентов необходимо произвести обратный ход, т.е. по ним вычислить значения $y_{j-1}, j = \overline{n,2}$, значение y_nопределяем исходя из начальных условий:

$$y_n = \frac{\gamma_2\beta_n + \delta_2 h}{\gamma_2(1 - \alpha_n) + \sigma_2 h}$$

Формула для вычисления y_{j-1}:

$$y_{j-1} = y_j\alpha_j + \beta_j, j = \overline{n,2}$$

Описание программы

Программа решает заданную краевую задачу вышеизложенными методами. Число отрезков разбиения (SIZE) задается пользователем. Программа выводи результат в виде: x: <точка>counted: <вычисленное знач.> given: <точное решение>. Сама задача задается в программе следующим образом: функции p_func(x), q_func(x) и r_func(x) отвечают, соответственно, за функции p(x), q(x) и r(x) уравнения (1). Коэффициенты граничных условий (2) задаются в коде программы с помощью директивы #define. Точное решение задаётся в функции sol(x).

Тестирование

I. <u>Собственный тест.</u>

условие:

$$y'' - 2*x*y' - 2*y = -4*x, \; 0 < x < 1$$

$$\begin{cases} y(0) - y'(0) = 0, \\ y(1) = 1 + e = 3.718 \end{cases}$$

точное решение:

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

n	Отрезок	Погрешность
10	[0;1]	0.026064
100	[0;1]	0.002049
1000	[0;1]	0.000165

Примечание: графики находятся в Приложении 2.

II. Вариант 5

$$y'' + 2y' - x * y = x^2, \ 0.6 < x < 0.9$$

$$\begin{cases} y'(0.6) = 0.7, \\ y(0.9) - 0.5y'(0.9) = 1 \end{cases}$$

точное решение:

$$\text{не выражается в элементарных функциях}$$

n	Отрезок	Погрешность
10	[0.6;0.9]	N/A
100	[0.6;0.9]	N/A
1000	[0.6;0.9]	N/A

Вывод

В данной работе был реализован метод прогонки, позволяющий находить решения краевой задачи с точностью до $\overline{o}(h^2)$, где $h = \frac{x_n - x_1}{n}$. Такая точность позволяет получать достаточно хорошие приближения решений, в том числе и таких, которые не выражаются в элементарных функциях.

Приложение 2