МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ им. В. Г. ШУХОВА» (БГТУ им. В.Г. Шухова)

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

Лабораторная работа №4

по дисциплине: «Вычислительная математика»

тема: Численные методы решения задачи Коши

Выполнил: ст. группы ПВ-211

Чувилко Илья Романович

Проверил:

Бондаренко Татьяна Владимировна

Белгород 2023 г.

Вариант 23

Цель работы: изучить численные методы решения задачи Коши; получить практические навыки приближенного решения дифференциальных уравнений с помощью ЭВМ.

Ход работы:

23	$y' - \frac{2}{x+1} y = e^x (1+x)^2, \quad y\|_{x=0} = 1, 0 \le x \le 2$	$\varphi(x) = e^x (x+1)^2$

1. Вычислить «вручную» приближенное решение y(x) задачи Коши методом последовательного дифференцирования.

   Замечание. Ряд Тейлора ограничить значением производной третьего порядка.

2. Вычислить значение функции φ(x), которая является точным решением задачи Коши и функции y(x), которая является приближенным решением задачи Коши по методу последовательного дифференцирования, в точке x = b. *Замечание. $x = b$ – правый конец указанного в задании отрезка, которому принадлежит значение x, $a \le x \le b$, $x = b = x0 + ih$, $h > 0$ — шаг сетки, $x0 = a$*

3. Определить относительную и абсолютную погрешности вычисления приближенного решения задачи Коши методом последовательного дифференцирования. Значения погрешностей внести в соответствующие ячейки таблицы 4.

Формула для ячейки C2: =1+3*B2+(7/2)*СТЕПЕНЬ(B2; 2) + (13/6)*СТЕПЕНЬ(B2; 3)

i	xi	yi	φi		Абс. Погрешность	Отн. Погрешность
0	0	1	1		0	0
1	0,2	1,757333	1,75882		0,00148663841731	0,00084524763261
2	0,4	2,898667	2,923976		0,02530974071022	0,00865593191736
3	0,6	4,528	4,664624		0,1366241289997	0,02928941865869
4	0,8	6,749333	7,210753		0,46141927498226	0,06399044594182
5	1	9,666667	10,87313		1,20646064716951	0,11095801716902
6	1,2	13,384	16,06937		2,68536590604489	0,16711088177006
7	1,4	18,00533	23,35795		5,352618475692	0,22915615716031
8	1,6	23,63467	33,4825		9,84783252224431	0,29411880118871
9	1,8	30,376	47,42924		17,0532361209975	0,35955114430881
10	2	38,33333	66,5015		28,1681715570425	0,42357194176998

4. Вычислить «вручную» приближенное решение y(x) задачи Коши четырьмя численными методами решения:

   – методом Эйлера;

$$f(x_i, y_i) = e^x (1+x)^2 + \frac{2y}{(x+1)^2}$$

Формула для ячейки C3: =C2 + (2/10) * (EXP(B2)*СТЕПЕНЬ(1+ B2; 2) + (2*C2)/СТЕПЕНЬ(B2+1;2))

i	xi	yi	ϕi		Абс. Погрешность	Отн. Погрешность
0	0	1	1		0	0
1	0,2	1,6	1,75882		0,15881997175065	0,09029916324669
2	0,4	2,396208	2,9239764		0,52776796858232	0,18049665765114
3	0,6	3,470026	4,6646241		1,19459827836351	0,25609743579054
4	0,8	4,945142	7,2107526		2,26561039271756	0,3141988798929
5	1	6,997804	10,873127		3,87532319193329	0,35641293255179
6	1,2	9,87221	16,069366		6,19715590918448	0,3856503078851
7	1,4	13,90197	23,357952		9,45598350724846	0,40482930971691
8	1,6	19,53897	33,482499		13,9435260599279	0,41644221302769
9	1,8	27,39163	47,429236		20,0376108980793	0,42247382704965
10	2	38,275	66,501505		28,2265005441297	0,42444904954647
0	0	1	1		0	0
1	0,4	1,6	2,9239764		1,32397640737689	0,45279996241989
2	0,8	2,511326	7,2107526		4,69942671459532	0,65172485728825
3	1,2	4,263517	16,069366		11,8058492568689	0,73468046753655
4	1,6	7,829747	33,482499		25,6527526106602	0,76615405755482
5	2	14,98955	66,501505		51,5119598602452	0,77459840863992

– методом Эйлера-Коши;

Формула ячейки C3: =C2+(1/2)*(B3-B2)*((EXP(B2)*СТЕПЕНЬ(1+B2;2)+(2*C2)/(B2+1))+(EXP(B3)*СТЕПЕНЬ(1+B3; 2)+(2*(EXP(B2)*СТЕПЕНЬ(1+B2;2)+(2*C2)/(B2+1)))/(B3+1))

i	xi	yi	ϕi		Абс. Погрешность	Отн. Погрешность
0	0	1	1		0	0
1	0,2	1,9758819972	1,7588199718		0,21706202542442	0,12341344134748
2	0,4	3,4951833918	2,9239764074		0,57120698443664	0,1953528020936
3	0,6	5,7429923009	4,664624129		1,07836817188509	0,23118007840781
4	0,8	8,9643334017	7,2107526083		1,75358079336752	0,24318970412953
5	1	13,485870744	10,873127314		2,61274343015658	0,24029364825259
6	1,2	19,743161509	16,069365906		3,67379560322879	0,22862106847955
7	1,4	28,315534024	23,357951809		4,9575822154714	0,21224387549065
8	1,6	39,971070818	33,482499189		6,48857162942893	0,19378994360068
9	1,8	55,724761872	47,429236121		8,29552575083874	0,17490321222286
10	2	76,913685271	66,50150489		10,4121803801685	0,15657059787342
0	0	0	1		1	1
1	0,4	1,0705095672	2,9239764074		1,85346684018723	0,63388570287747
2	0,8	4,3929320887	7,2107526083		2,81782051963055	0,39078036270181
3	1,2	12,223669879	16,069365906		3,84569602686702	0,23931846778225
4	1,6	28,538342649	33,482499189		4,94415653986662	0,14766390381946
5	2	60,317000446	66,50150489		6,18450444412079	0,09299796229146

– модифицированным методом Эйлера;

Формула ячейки E3: =E2+(B3-B2)*(EXP(C2)*СТЕПЕНЬ(1+C2; 2) + (2*(E2+((B3-B2)/2)*D2))/C2)

i	xi	x(i+1/2)	f(xi, yi)	yi	ϕi	Абс. Погрешность	Отн. Погрешность
0	0	0,1	3	1	1	0	0
1	0,2	0,3	4,69027	1,7588685466	1,758819971751	4,857484957549E-05	2,7617863315E-05
2	0,4	0,5	7,10122	2,9240727072	2,923976407377	9,629984382398E-05	3,293454885E-05
3	0,6	0,7	10,4956	4,664767296	4,664624129	0,0001431670255406	3,0692081844E-05
4	0,8	0,9	15,2229	7,2109417293	7,210752608316	0,0001891209661588	2,6227632042E-05
5	1	1,1	21,7465	10,873361381	10,87312731384	0,0002340676021651	2,152716467E-05
6	1,2	1,3	30,6781	16,069643788	16,06936590604	0,0002778816586332	1,7292633714E-05
7	1,4	1,5	42,8232	23,358272222	23,35795180903	0,000320413011702	1,3717513176E-05
8	1,6	1,7	59,2385	33,482860681	33,48249918891	0,0003614922477166	1,0796453565E-05
9	1,8	1,9	81,3075	47,429637057	47,429236121	0,0004009356076156	8,4533431361E-06
10	2	1	110,836	66,50194344	66,50150489038	0,0004385494539526	6,594579396E-06
0	0	0,2	3	1	1	0	0
1	0,4	0,6	7,10165	2,9243704674	2,923976407377	0,0003940600191924	0,00013476853582
2	0,8	1	15,2236	7,2115273432	7,210752608316	0,0007747348520552	0,00010744160757
3	1,2	1,4	30,6789	16,07050539	16,06936590604	0,0011394843444066	7,0910348988E-05
4	1,6	1,8	59,2394	33,483982876	33,48249918891	0,0014836867042161	4,4312304641E-05
5	2	1	110,837	66,503306336	66,50150489038	0,0018014455330507	2,7088793495E-05

– методом Рунге-Кутты.

i	xi	m1	m2	m3	m4	yi	ϕi	Абс. Погрешность	Отн. Погрешность
0	0	3	3,7008931745	3,8283282971	4,7015960708	1	1	0	0
1	0,2	4,6899332503	5,7084326035	5,8651248117	7,1121091638	1,7586679671	1,7588199718	0,00015200461376	8,6424202708E-05
2	0,4	7,1006048006	8,5545566662	8,748416915	10,506278202	2,9236398753	2,9239764074	0,00033653208819	0,00011509398207
3	0,6	10,494708563	12,541555211	12,782360699	15,233574483	4,6640675474	4,664624129	0,00055658154992	0,00011931969962
4	0,8	15,221794878	18,070869895	18,370772529	21,757219863	7,2099380431	7,2107526083	0,00081456526514	0,00011296536012
5	1	21,74514201	25,673637782	26,047780236	30,688975673	10,872014696	10,873127314	0,00111261781182	0,00010232730471
6	1,2	30,676559682	36,050204695	36,517478174	42,834125799	16,067913153	16,069365906	0,00145275273115	9,040510619E-05
7	1,4	42,82138086	50,121159047	50,705141302	59,249532359	23,356114861	23,357951809	0,00183694842252	7,8643386096E-05
8	1,6	59,236523797	69,093138983	69,823258626	81,318438776	33,480231991	33,482499189	0,00226719778087	6,7712919758E-05
9	1,8	81,305300824	94,543558926	95,456542244	110,84670425	47,426490584	47,429236121	0,00274553685971	5,7887014092E-05
10	2	110,83365878	128,5295359	129,67120539	150,18643318	66,498230831	66,50150489	0,00327405927419	4,9232859912E-05
	0,2								
i	xi	m1	m2	m3	m4	yi	ϕi	Абс. Погрешность	Отн. Погрешность
0	0	3	4,4254866384	4,9006488512	7,1529186081	1	1	0	0
1	0,4	5,7242235306	8,5458962445	9,2513144229	13,500417892	1,9601729862	2,9239764074	0,96380342113167	0,32962079266443
2	0,8	11,419058251	16,944414042	18,0494852	26,075974232	3,7874750782	7,2107526083	3,42327753015587	0,47474621805886
3	1,2	22,769580552	33,294745326	35,048939455	49,936200645	7,3702361104	16,069365906	8,69912979563052	0,54134866593324
4	1,6	44,520966729	64,039379798	66,827724522	93,888903298	14,350007802	33,482499189	19,1324913864906	0,57141766146379
5	2	84,960267396	120,34126151	124,76388577	173,07127076	27,688143758	66,50150489	38,8133611323268	0,58364635802315
	0,4								

5. Сравнить полученные в пункте 4 значения приближенного решения дифференциального уравнения y(x) с точным значением решения дифференциального уравнения φ(x) в точке x = b.
6. Определить относительную и абсолютную погрешности вычисления приближенного решения задачи Коши заданными численными методами. Значения погрешностей внести в соответствующие ячейки табл. 4.1.

Погрешность	Вычислительный метод				
	Послед. Дифференцирования	Эйлера	Эйлера-Коши	мод. Эйлера	Рунге-Кутты
H=0,4					
Δ	28,168171557	51,51195986	6,1845044441	0,0018014455	38,813361132
δ	0,4235719418	0,7745984086	0,0929979623	2,708879E-05	0,583646358
H=0,2					
Δ	28,168171557	28,226500544	10,41218038	0,0004385495	0,0032740593
δ	0,4235719418	0,4244490495	0,1565705979	0,0024368275	4,923286E-05

7. Описать в модуле функции, каждая из которых возвращает приближенное значение решения задачи Коши в точке x = b с точностью ⌜ , реализующие метод Эйлера, метод Эйлера-Коши, модифицированный метод Эйлера и метод РунгеКутты. Оценка точности вычисления должна осуществляться по принципу Рунге.

Код программы:

```
#include <iostream>
#include "vector"
#include "cmath"

using namespace std;

typedef double diff_equation(double x, double y);

typedef double methodOfSolution(diff_equation func, double x0, double y0, double target, double step);

class KoshiSolver {
public:
    static double euler(diff_equation func, double x0, double y0, double target, double step) {
        //x, y
        vector<vector<double>> table{{x0, y0}};

        while (table[table.size() - 1][0] < target) {
            vector<double> previous_row = table[table.size() - 1];
            double next_x = previous_row[0] + step;
            double next_y = previous_row[1] + step * func(previous_row[0], previous_row[1]);
            table.push_back({next_x, next_y});
        }

        return table[table.size() - 1][1];
    }

    static double eulerKoshi(diff_equation func, double x0, double y0, double target, double step) {
        //x, y
        vector<vector<double>> table{{x0, y0}};

        while (table[table.size() - 1][0] < target) {
            vector<double> previous_row = table[table.size() - 1];
            double next_x = previous_row[0] + step;

            double intermediate_value = func(previous_row[0], previous_row[1]);
            double next_y = previous_row[1] + step / 2 *
                    (intermediate_value + func(next_x, previous_row[1] + step * intermediate_value));
            table.push_back({next_x, next_y});
        }

        return table[table.size() - 1][1];
    }

    static double modifiedEuler(diff_equation func, double x0, double y0, double target, double step) {
        //x, y
        vector<vector<double>> table{{x0, y0}};

        while (table[table.size() - 1][0] < target) {
            vector<double> previous_row = table[table.size() - 1];
            double next_x = previous_row[0] + step;

            double intermediate_value = func(previous_row[0], previous_row[1]);
```

```
        double next_y = previous_row[1] + step * func(previous_row[0] + step / 2,
                                                      previous_row[1] + step / 2 * intermediate_value);
        table.push_back({next_x, next_y});
    }

    return table[table.size() - 1][1];
}

static double rungeKutta(diff_equation func, double x0, double y0, double target, double step) {
    //x, y
    vector<vector<double>> table{{x0, y0}};

    while (table[table.size() - 1][0] < target) {
        vector<double> previous_row = table[table.size() - 1];
        double next_x = previous_row[0] + step;

        double m1 = func(previous_row[0], previous_row[1]);
        double m2 = func(previous_row[0] + step / 2, previous_row[1] + step / 2 * m1);
        double m3 = func(previous_row[0] + step / 2, previous_row[1] + step / 2 * m2);
        double m4 = func(previous_row[0] + step, previous_row[1] + step * m3);
        double next_y = previous_row[1] + step / 6 * (m1 + 2 * m2 + 2 * m3 + m4);

        table.push_back({next_x, next_y});
    }

    return table[table.size() - 1][1];
}

static double findError(diff_equation func, double x0, double y0, double target, double step,
                        methodOfSolution method, double method_p) {
    double valueStep = method(func, x0, y0, target, step);
    double valueStepDividedBy2 = method(func, x0, y0, target, step / 2);

    return (valueStepDividedBy2 - valueStep) / (std::pow(2, method_p) - 1);
}

double solveKoshiWithPrecision(diff_equation func, double x0, double y0, double target, double step,
                               const string& method, double precision) {
    if (method == "Euler") {
        double currentError = findError(func, x0, y0, target, step, euler, 1);
        while (currentError > precision) {
            step /= 2;
            currentError = findError(func, x0, y0, target, step, euler, 1);
        }
        return euler(func, x0, y0, target, step);

    } else if (method == "Euler Koshi") {
        double currentError = findError(func, x0, y0, target, step, eulerKoshi, 2);
        while (currentError > precision) {
            step /= 2;
            currentError = findError(func, x0, y0, target, step, eulerKoshi, 2);
        }
        return eulerKoshi(func, x0, y0, target, step);

    } else if (method == "Modified Euler") {
        double currentError = findError(func, x0, y0, target, step, modifiedEuler, 2);
        while (currentError > precision) {
            step /= 2;
            currentError = findError(func, x0, y0, target, step, modifiedEuler, 2);
        }
```

```
            return modifiedEuler(func, x0, y0, target, step);

        } else if (method == "Runge Kutta") {
            double currentError = findError(func, x0, y0, target, step, rungeKutta, 4);
            while (currentError > precision) {
                step /= 2;
                currentError = findError(func, x0, y0, target, step, rungeKutta, 4);
            }
            return rungeKutta(func, x0, y0, target, step);

        } else {
            return 0;
        }
    }
};

double target_function(double x, double y) {
    return std::pow(M_E, x) * std::pow(x + 1, 2) + (2*y)/(x+1);
}

int main() {
    system("chcp 65001");
    KoshiSolver solver;
    double x0 = 0;
    double y0 = 1;

    cout << "Точность для пятого знака"
    << "\nМетод Эйлера: " << solver.solveKoshiWithPrecision(target_function, x0, y0, 2, 1, "Euler", 0.0001)
    << "\nМетод Эйлера Коши: " << solver.solveKoshiWithPrecision(target_function, x0, y0, 2, 1, "Euler Koshi",
0.0001)
    << "\nМетод модифицированный Эйлера: " << solver.solveKoshiWithPrecision(target_function, x0, y0, 2, 1,
"Modified Euler", 0.0001)
    << "\nМетод Рунге Кутта: " << solver.solveKoshiWithPrecision(target_function, x0, y0, 2, 1, "Runge Kutta",
0.0001);
}
```

Результат работы программы:

```
D:\BGTU\VicMat\Lab4\Example\Code\cmake-build-debug\Code.exe
Active code page: 65001
Точность для пятого знака
Метод Эйлера: 66.5013
Метод Эйлера Коши: 66.5012
Метод модифицированный Эйлера: 66.5014
Метод Рунге Кутта: 66.5009
Process finished with exit code 0
```