**МИНОБРНАУКИ РОССИИ**

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 №5

дисциплина: Вычислительная математика

тема: «Численные методы решения задачи Коши»

|  |  |
| --- | --- |
|  | Выполнил: ст. группы ПВ-21  Браткова Ирина Олеговна  Проверил: Бондаренко Т.В. |

Белгород 2017

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

**Задания к работе**

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(х), 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 х, a ≤ x ≤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. Вычислить «вручную» приближенное решение y(x) задачи Коши четырьмя численными методами решения: методом Эйлера; методом Эйлера-Коши; модифицированным методом Эйлера; методом Рунге-Кутты. Сначала выполнить вычисления с шагом h = 0,2, а затем с шагом h = 0,1. Вычисления вручную можно выполнить с помощью MS Excel или другой программы и обязательно их включать в отчет.
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 таблицы 4.

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 Рунге-Кутты. Оценка точности вычисления должна осуществляться по принципу Рунге.

1. Составить программу для вычисления приближенных значений решения задачи Коши с заданной точностью на отрезке [a, b] с шагом h для соответствующего варианта задания с использованием всех функций, описанных в модуле. Результат работы программы таблица значений приближенного решения задачи Коши для заданного отрезка a ≤ x ≤ b. Предусмотреть возможность сохранения результата работы программы в файл.

Задания 1-2

Задание 4

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Eiler |  |  |  | Eiler-Koshi |  |
|  | h | 0,2 |  |  | h | 0,2 |
| i | x | y |  | i | x | y |
| 0 | 1,570796 | 0 |  | 0 | 1,570796 | 0 |
| 1 | 1,770796 | 0,628319 |  | 1 | 1,770796 | 0,648522 |
| 2 | 1,970796 | 1,297045 |  | 2 | 1,970796 | 1,294309 |
| 3 | 2,170796 | 1,913458 |  | 3 | 2,170796 | 1,860173 |
| 4 | 2,370796 | 2,368299 |  | 4 | 2,370796 | 2,287302 |
| 5 | 2,570796 | 2,5413 |  | 5 | 2,570796 | 2,567604 |
|  |  |  |  |  |  |  |
|  | Eiler |  |  |  | Eiler-Koshi |  |
|  | h | 0,1 |  |  | h | 0,1 |
| i | x | y |  | i | x | y |
| 0 | 1,570796 | 0 |  | 0 | 1,570796 | 0 |
| 1 | 1,670796 | 0,314159 |  | 1 | 1,670796 | 0,321749 |
| 2 | 1,770796 | 0,643497 |  | 2 | 1,770796 | 0,653625 |
| 3 | 1,870796 | 0,977552 |  | 3 | 1,870796 | 0,985846 |
| 4 | 1,970796 | 1,304761 |  | 4 | 1,970796 | 1,308894 |
| 5 | 2,070796 | 1,612641 |  | 5 | 2,070796 | 1,613923 |
| 6 | 2,170796 | 1,888001 |  | 6 | 2,170796 | 1,893251 |
| 7 | 2,270796 | 2,117163 |  | 7 | 2,270796 | 2,141057 |
| 8 | 2,370796 | 2,286197 |  | 8 | 2,370796 | 2,354476 |
| 9 | 2,470796 | 2,381152 |  | 9 | 2,470796 | 2,535608 |
| 10 | 2,570796 | 2,388263 |  | 10 | 2,570796 | 2,69578 |
|  |  |  |  |  |  |  |
|  | Modif Eiler |  |  |  | Modif Eiler |  |
|  | h | 0,2 |  |  | h | 0,1 |
| i | x | y |  | i | x | y |
| 0 | 1,570796 | 0 |  | 0 | 1,570796 | 0 |
| 1 | 1,770796 | 0,66498 |  | 1 | 1,670796 | 0,323754 |
| 2 | 1,970796 | 1,33939 |  | 2 | 1,770796 | 0,659149 |
| 3 | 2,170796 | 1,927141 |  | 3 | 1,870796 | 0,995243 |
| 4 | 2,370796 | 2,327187 |  | 4 | 1,970796 | 1,320273 |
| 5 | 2,570796 | 2,443618 |  | 5 | 2,070796 | 1,621882 |
|  |  |  |  | 6 | 2,170796 | 1,887377 |
|  |  |  |  | 7 | 2,270796 | 2,104002 |
|  |  |  |  | 8 | 2,370796 | 2,259239 |
|  |  |  |  | 9 | 2,470796 | 2,341125 |
|  |  |  |  | 10 | 2,570796 | 2,338601 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Runge-Kutta |  |  |  |  |  |
|  | h | 0,2 |  |  |  |  |
| i | x | y | m1 | m2 | m3 | m4 |
| 0 | 1,570796 | 0 | 3,141593 | 3,293378 | 3,291855 | 3,337538 |
| 1 | 1,770796 | 0,654987 | 3,338224 | 3,268606 | 3,270759 | 3,076952 |
| 2 | 1,970796 | 1,304783 | 3,078794 | 2,753588 | 2,771354 | 2,311424 |
| 3 | 2,170796 | 1,852787 | 2,315711 | 1,717971 | 1,768318 | 1,031653 |
| 4 | 2,370796 | 2,196785 | 1,041605 | 0,172188 | 0,281748 | -0,73103 |
| 5 | 2,570796 | 2,2374 |  |  |  |  |
|  |  |  |  |  |  |  |
|  | Runge-Kutta |  |  |  |  |  |
|  | h | 0,1 |  |  |  |  |
| i | x | y | m1 | m2 | m3 | m4 |
| 0 | 1,570796 | 0 | 3,141593 | 3,229681 | 3,229461 | 3,292496 |
| 1 | 1,670796 | 0,32254 | 3,292537 | 3,329319 | 3,329041 | 3,338132 |
| 2 | 1,770796 | 0,654996 | 3,338222 | 3,318517 | 3,318769 | 3,269204 |
| 3 | 1,870796 | 0,986363 | 3,269362 | 3,188966 | 3,190434 | 3,07853 |
| 4 | 1,970796 | 1,304808 | 3,078783 | 2,934585 | 2,938068 | 2,761262 |
| 5 | 2,070796 | 1,597897 | 2,761655 | 2,551724 | 2,558159 | 2,315078 |
| 6 | 2,170796 | 1,852838 | 2,315676 | 2,039325 | 2,049829 | 1,740323 |
| 7 | 2,270796 | 2,056743 | 1,741231 | 1,399041 | 1,41498 | 1,040105 |
| 8 | 2,370796 | 2,1969 | 1,041487 | 0,635289 | 0,658408 | 0,220332 |
| 9 | 2,470796 | 2,261053 | 0,222458 | -0,24476 | -0,21209 | -0,71034 |
| 10 | 2,570796 | 2,237693 | -0,70699 | -1,23132 | -1,18561 | -1,74066 |

**Задание 3, 5-6**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Погрешность | Вычислительный метод | | | | |
| Последовательного дифференцирования | Эйлера | Эйлера-Коши | Модифицированный Эйлера | Рунге-Кутта |
| h=0.2 | | | | | |
| δ | 14,88 | 16.5 | 14.7 | 9 | 0,01 |
| Δ | 0,333 | 0.3036 | 0.3299 | 0,2059 | 0,0003 |
| h=0.1 | | | | | |
| δ | 14,88 | 6.72786 | 20 | 4,5 | 0,000849 |
| Δ | 0,333 | 0.15 | 0.458 | 0,1009 | 0,000019 |

**Программа**

*Заголовочный файл:*

#ifndef V\_55\_H\_INCLUDED #define V\_55\_H\_INCLUDED

#include <stdlib.h>#include <math.h>

typedef struct **//структура матрица размера m на n**

{

double \*\*a; int m, n; } Matr;

typedef double(\*func\_f)(double, double); **//тип функции и ее описание**

double fun(double x, double y);

void getmem\_matr(Matr \*a, int m, int n); **//выделение и высвобождение памяти**

void freemem\_matr(Matr \*a);

**/\*Решение задачи коши методом Рунге-Кутты, где**

**f - функция (первая произовдая) , a(b)- нижняя(верхняя) граница x , eps - точность ,**

**y0 - начальное условие задачи Коши , n-количество на сколько разбивается отрезок \*/**

double cauchyRK(func\_f f, Matr \*yOut, double a, double b , double y0, double n);

**/\*Решение задачи коши методом Эйлера \*/**

double cauchyEul(func\_f f, Matr \*yOut, double a, double b, double y0, double n);

**/\*Решение задачи коши методом Эйлера-Коши\*/**

double cauchyEC(func\_f f , Matr \*yOut, double a, double b , double y0, double n);

**/\*Решение задачи коши методом Эйлера-Коши модифицированный\*/**

double cauchyECM(func\_f f, Matr \*yOut, double a, double b, double y0, double n);

**/\*Вычисление в зависимости от метода\*/**

double cauchy(func\_f f, int m, Matr \*yOut, double a, double b, double y0, double n, double eps);

**/\*Ввод в файл\*/**void printFL (Matr a)

#endif

*Файл C*

#include "v\_55.h" #include "math.h"

double fun(double x, double y) **//функция варианта**

{ return 2\*x\*sin(x)+y/tan(x); }

void getmem\_matr(Matr \*a, int m, int n) **//выделение памяти**

{

a->a=(double\*\*)calloc(m,sizeof(double\*));

int i;

for (i=0; i<=m-1; i++)

a->a[i]= (double \*) calloc(n,sizeof(double));

a->m = m; //записываем

a->n = n;

}

void freemem\_matr(Matr \*a) **//освбождение памяти**

{ int i;

for (i=0; i<=a->m-1; i++) free(a->a[i]);

free(a->a);

}

**/\*Решение задачи коши методом Рунге-Кутты, где**

**f - функция (первая произовдая) , a(b)- нижняя(верхняя) граница x , eps - точность ,**

**y0 - начальное условие задачи Коши , n-количество на сколько разбивается отрезок \*/**

double cauchyRK(func\_f f, Matr \*yOut, double a, double b , double y0, double n)

{

freemem\_matr(yOut); getmem\_matr(yOut, n+1, 2);

double y=y0, x=a, k1, k2, k3, k4, h=(b-a)/n;

yOut->a[0][1]=y0; yOut->a[0][0]=a;

int i;

for (i=1; i<=n; i++)

{

k1 = f(x, y);

k2 = f(x+h/2, y+h/2\*k1);

k3 = f(x+h/2, y+h/2\*k2);

k4 = f(x+h, y+h\*k3);

x+=h;

y+=h/6\*(k1+2\*k2+2\*k3+k4);

yOut->a[i][0]=x;

yOut->a[i][1]=y;

}

return y;

}

double cauchyEul(func\_f f, Matr \*yOut, double a, double b, double y0, double n)

{

freemem\_matr(yOut);

getmem\_matr(yOut, n+1, 2);

double y=y0, x=a, h=(b-a)/n;

yOut->a[0][1]=y0;

yOut->a[0][0]=a;

int i;

for (i=1; i<=n; i++)

{

y+=h\*f(x, y); x+=h;

yOut->a[i][0]=x; yOut->a[i][1]=y;

}

return y;

}

double cauchyEC(func\_f f , Matr \*yOut, double a, double b , double y0, double n)

{

freemem\_matr(yOut);

getmem\_matr(yOut, n+1, 2);

double y=y0, x=a, h=(b-a)/n, fi;

yOut->a[0][1]=y0;

yOut->a[0][0]=a;

int i;

for (i=1; i<=n; i++)

{

fi=f(x, y);

y+=h/2\*(fi+f(x+h, y+h\*fi));

x+=h;

yOut->a[i][0]=x;

yOut->a[i][1]=y;

}

return y;

}

double cauchyECM(func\_f f, Matr \*yOut, double a, double b, double y0, double n)

{ freemem\_matr(yOut);

getmem\_matr(yOut, n+1, 2);

double y=y0, x=a, h=(b-a)/n;

yOut->a[0][1]=y0; yOut->a[0][0]=a;

int i;

for (i=1; i<=n; i++)

{

y+=h\*f(x+h/2, y+h/2\*f(x, y)); x+=h;

yOut->a[i][0]=x; yOut->a[i][1]=y;

}

return y;

}

double cauchy(func\_f f, int m, Matr \*yOut, double a, double b, double y0, double n, double eps)

{

int p;

double (\*f1)(func\_f f, Matr \*yOut, double a, double b, double y0, double n) ;

switch (m)

{

case 1:

p=1; f1 = cauchyEul;

break;

case 2:

p=2; f1 = cauchyEC;

break;

case 3:

p=2; f1 = cauchyECM;

break;

case 4:

p=4; f1 = cauchyRK;

break;

}

double yh = f1(f, yOut, a, b, y0, n**); //высчитать функцию в точке**

n = 2\*n; **//увеличить n на два**

double yh\_2 = f1( f, yOut, a , b , y0 , n ); **//высчитать функцию в новой точке**

double eps1 = fabs(yh - yh\_2)/(pow(2, p)-1); **//получить новый эпсилон для сравнения**

while (eps1>eps) **//пока новыый эпсилон больше введенногоо**

{

yh = yh\_2; **//переприсвоить функцию**

n = 2\*n; **//снова увеличить n**

yh\_2 = f1( f, yOut, a , b , y0 , n ); **//пересчитать**

eps1 = fabs(yh - yh\_2)/(pow(2,p)-1); **//вычислить новый эпсилон**

}

return yh+(yh - yh\_2)/(pow(2,p)-1);

}

void printFL (Matr a)

{

FILE \*S1;

S1 = freopen("Rez1.txt", "w", stdout);

printMatr(a);

fclose(S1);

}

*Главная функция:*

#include <stdio.h>#include <stdlib.h>#include "v\_55.h"

int main()

{ Matr a; int m, n; printf("Input n >> "); scanf("%d",&n);

double eps, an = 1.5707963, bn = 2.5707963, yo = 0;

getmem\_matr(&a, n+1, 2); //столбцов всегда 2

printf("Input eps >> "); scanf("%lf",&eps);

printf("1 - Eiler \n"); printf("2 - Eiler-Koshi \n");

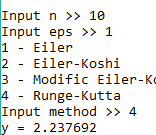
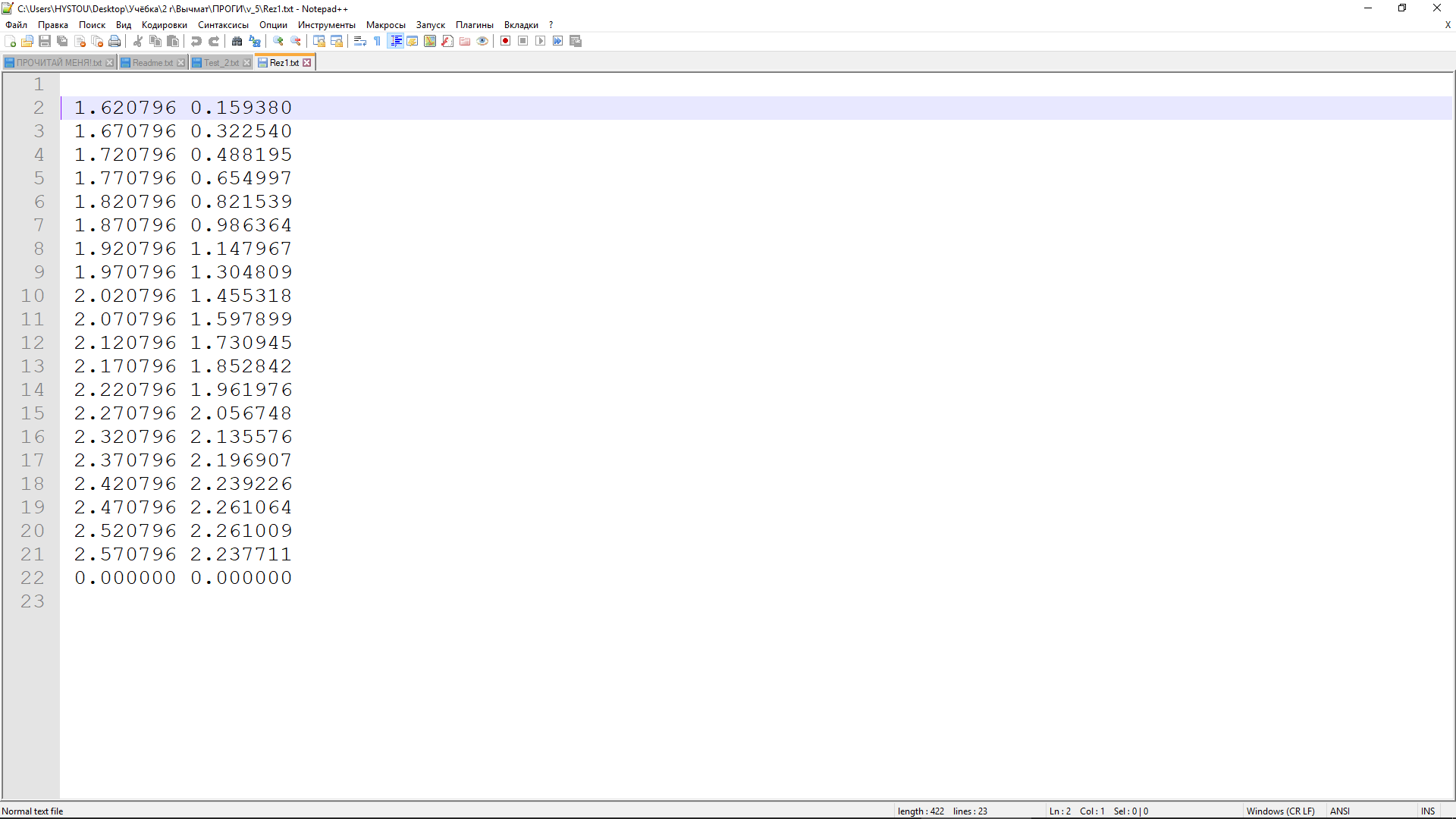
printf("3 - Modific Eiler-Koshi \n"); printf("4 - Runge-Kutta \n");

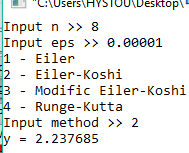
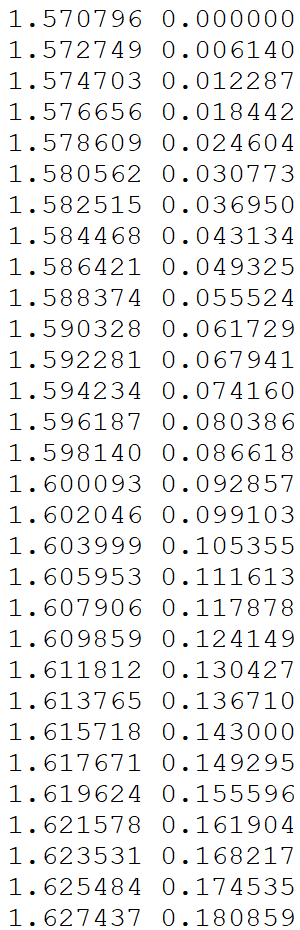
printf("Input method >> "); scanf("%d",&m);

printf("y = %lf\n", cauchy(fun, m, &a, an, bn, yo, n, eps)); printFL(a); return 0; }

*Пример работы:*

*Если нужная точность не была достигнута, функция высчитывается при новом n, таким образом, увеличивается количество строк матрицы файла.*

Т.к. в последнем случае был задан очень малый эпсилон, количество записей в таблице очень велико.