**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 №4

Дисциплина: Вычислительная математика

по теме «Численные методы решения задачи Коши»

Выполнил: ст. группы ВТ-22  
Макаров Даниил Сергеевич

Проверила: Бондаренко Т.В.

**Белгород 2018**

**Лабораторная работа №4**

**Численные методы решения задачи Коши**

**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.

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.

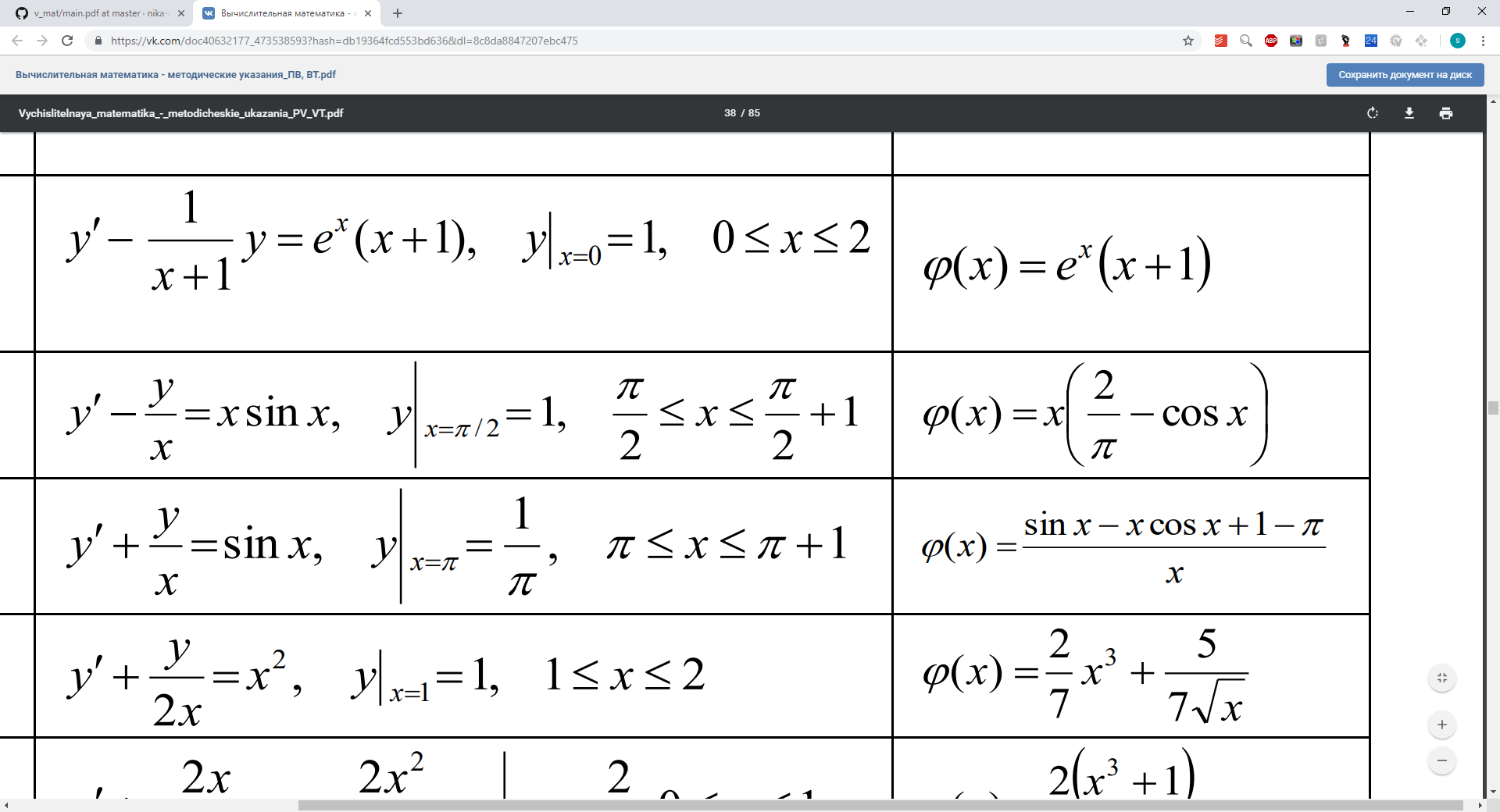
6. Определить относительную и абсолютную погрешности вычисления приближенного решения задачи Коши заданными численными методами. Значения погрешностей внести в соответствующие ячейки таблицы 4.1.

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

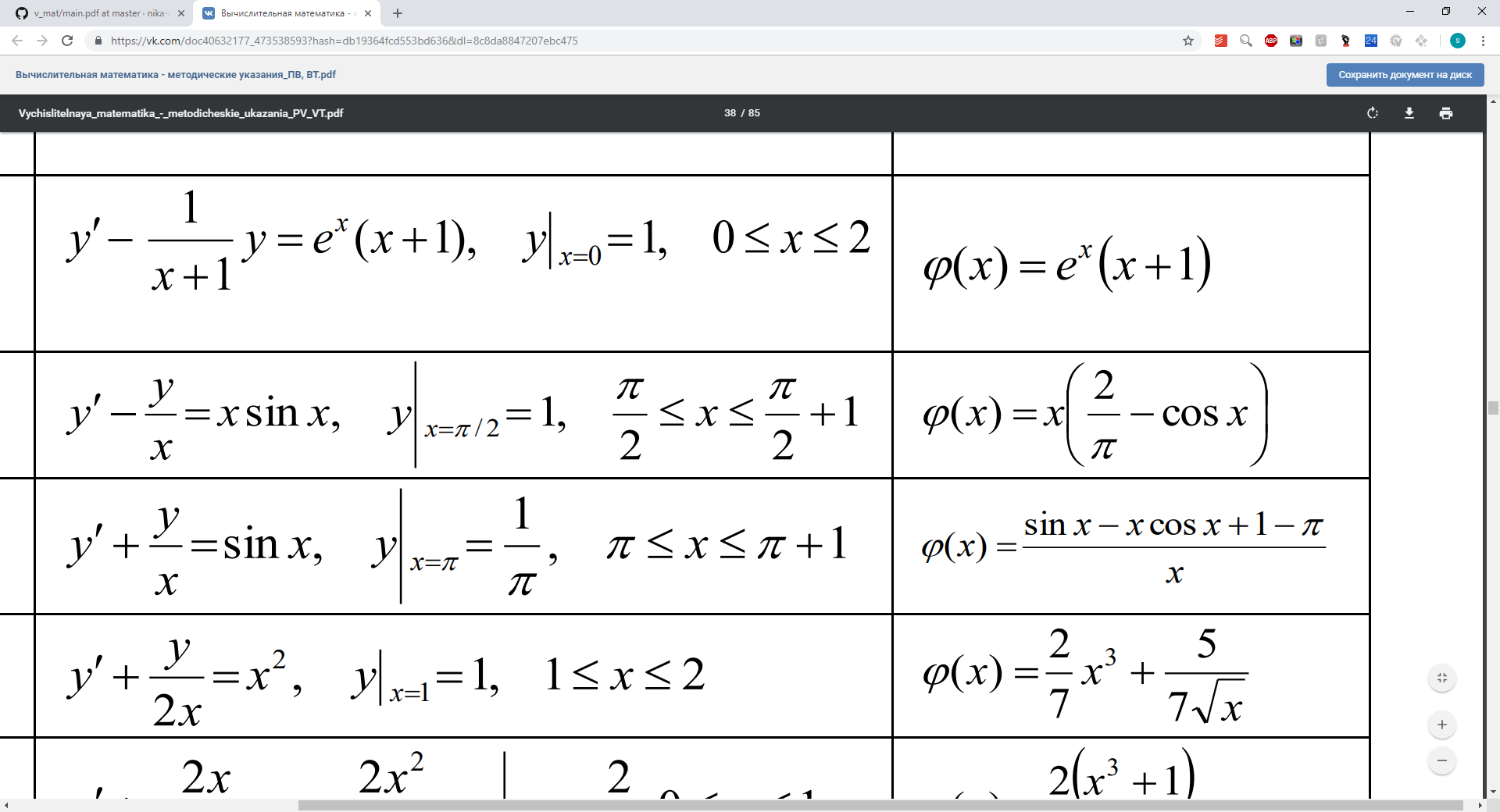
8. Составить программу для вычисления приближенных значений решения задачи Коши с точностью ε на отрезке *[a, b]* с шагом h для соответствующего варианта задания с использованием всех функций, описанных в модуле. Результат работы программы таблица значений приближенного решения задачи Коши для заданного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

Вариант задание 7.

Задача Коши



Точное решение



Выполнение задания

x0=pi/2 y(pi/2)=1

Ряд Тейлора

Первая производная

Вторая производная

Третья производная

Подставим полученные значения в ряд Тейлора

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Погрешность | Вычислительный метод | | | | |
| Последовательного дифференцирования | Эйлера | Эйлера-Коши | Модифицированный метод Эйлера | Рунге-Кутта |
| h=0.2 | | | | | |
| Δ | 4,496 | 0,050535 | 0,003641 | 0,019944 | 1,065367 |
| δ | 118,35% | 1,33% | 0,10% | 0,52% | 28,04% |
| h=0.1 | | | | | |
| Δ | 4,496 | 0,104673 | 0,016823 | 0,035173 | 1,061488 |
| δ | 118,35% | 2,76% | 0,44% | 0,93% | 27,94% |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Метод Элейра | | | | | | | |
| h | | 0,1 | | h | | 0,2 | |
| i | x\_i | y\_i | f(x,y) | i | x\_i | y\_i | f(x,y) |
| 0 | 1,570796 | 1 | 2,207416 | 0 | 1,570796 | 1 | 2,207416 |
| 1 | 1,670796 | 1,220742 | 2,393084 | 1 | 1,770796 | 1,441483 | 2,549529 |
| 2 | 1,770796 | 1,46005 | 2,560014 | 2 | 1,970796 | 1,951389 | 2,805376 |
| 3 | 1,870796 | 1,716051 | 2,704524 | 3 | 2,170796 | 2,512464 | 2,949028 |
| 4 | 1,970796 | 1,986504 | 2,823194 | 4 | 2,370796 | 3,10227 | 2,960285 |
| 5 | 2,070796 | 2,268823 | 2,912923 | 5 | 2,570796 | 3,694327 | 2,826043 |
| 6 | 2,170796 | 2,560116 | 2,970979 |  |  |  |  |
| 7 | 2,270796 | 2,857213 | 2,995044 |  |  |  |  |
| 8 | 2,370796 | 3,156718 | 2,983251 |  |  |  |  |
| 9 | 2,470796 | 3,455043 | 2,934224 |  |  |  |  |
| 10 | 2,570796 | 3,748465 | 2,847102 |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод Элейра-Коши | | | | | | | | | | | | | |
| h | | 0,1 | | | | | h | | 0,2 | | | | |
| i | x\_k | y\_k | f(x,y) | x\_k+h | y\_k+hf | f(x,y) | i | x\_k | y\_k | f(x,y) | x\_k+h | y\_k+hf | f(x,y) |
| 0 | 1,5708 | 1,0000 | 2,2074 | 1,6708 | 1,2207 | 2,3931 | 0 | 1,5708 | 1,0000 | 2,2074 | 1,7708 | 1,4415 | 2,5495 |
| 1 | 1,6708 | 1,2300 | 2,3986 | 1,7708 | 1,4699 | 2,5656 | 1 | 1,7708 | 1,4757 | 2,5688 | 1,9708 | 1,9895 | 2,8247 |
| 2 | 1,7708 | 1,4782 | 2,5703 | 1,8708 | 1,7353 | 2,7148 | 2 | 1,9708 | 2,0150 | 2,8377 | 2,1708 | 2,5826 | 2,9813 |
| 3 | 1,8708 | 1,7425 | 2,7187 | 1,9708 | 2,0144 | 2,8373 | 3 | 2,1708 | 2,5969 | 2,9879 | 2,3708 | 3,1945 | 2,9992 |
| 4 | 1,9708 | 2,0203 | 2,8403 | 2,0708 | 2,3043 | 2,9301 | 4 | 2,3708 | 3,1957 | 2,9997 | 2,5708 | 3,7956 | 2,8654 |
| 5 | 2,0708 | 2,3088 | 2,9322 | 2,1708 | 2,6020 | 2,9903 | 5 | 2,5708 | 3,7822 |  | 2,7708 | 3,7822 |  |
| 6 | 2,1708 | 2,6049 | 2,9916 | 2,2708 | 2,9041 | 3,0157 |  |  |  |  |  |  |  |
| 7 | 2,2708 | 2,9053 | 3,0162 | 2,3708 | 3,2069 | 3,0044 |  |  |  |  |  |  |  |
| 8 | 2,3708 | 3,2063 | 3,0042 | 2,4708 | 3,5068 | 2,9552 |  |  |  |  |  |  |  |
| 9 | 2,4708 | 3,5043 | 2,9542 | 2,5708 | 3,7997 | 2,8670 |  |  |  |  |  |  |  |
| 10 | 2,5708 | 3,7954 |  | 2,6708 | 3,7954 |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Модифицированный метод Эйлера | | | | | | | | | | | | | |
| h | | 0,1 | | | | | h | | 0,2 | | | | |
| i | x\_k | y\_k | f(x,y) | x\_k+h/2 | y\_k+h/2f | f(x,y) | i | x\_k | y\_k | f(x,y) | x\_k+h/2 | y\_k+h/2f | f(x,y) |
| 0 | 1,5708 | 1,0000 | 2,2074 | 1,6208 | 1,0810 | 2,2858 | 0 | 1,5708 | 1,0000 | 2,2074 | 1,6708 | 1,1671 | 2,3610 |
| 1 | 1,6708 | 1,2286 | 2,3978 | 1,7208 | 1,3146 | 2,4654 | 1 | 1,7708 | 1,4722 | 2,5669 | 1,8708 | 1,6593 | 2,6742 |
| 2 | 1,7708 | 1,4751 | 2,5685 | 1,8208 | 1,5662 | 2,6243 | 2 | 1,9708 | 2,0070 | 2,8336 | 2,0708 | 2,2141 | 2,8865 |
| 3 | 1,8708 | 1,7376 | 2,7160 | 1,9208 | 1,8336 | 2,7589 | 3 | 2,1708 | 2,5843 | 2,9821 | 2,2708 | 2,8114 | 2,9749 |
| 4 | 1,9708 | 2,0134 | 2,8369 | 2,0208 | 2,1145 | 2,8660 | 4 | 2,3708 | 3,1793 | 2,9928 | 2,4708 | 3,4264 | 2,9226 |
| 5 | 2,0708 | 2,3000 | 2,9280 | 2,1208 | 2,4061 | 2,9426 | 5 | 2,5708 | 3,7638 |  | 2,6708 | 4,0309 |  |
| 6 | 2,1708 | 2,5943 | 2,9867 | 2,2208 | 2,7053 | 2,9861 |  |  |  |  |  |  |  |
| 7 | 2,2708 | 2,8929 | 3,0108 | 2,3208 | 3,0090 | 2,9946 |  |  |  |  |  |  |  |
| 8 | 2,3708 | 3,1924 | 2,9983 | 2,4208 | 3,3134 | 2,9664 |  |  |  |  |  |  |  |
| 9 | 2,4708 | 3,4890 | 2,9480 | 2,5208 | 3,6151 | 2,9004 |  |  |  |  |  |  |  |
| 10 | 2,5708 | 3,7791 |  | 2,6208 | 3,9101 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Метод Рунге-Кутта | | | | | | | | | | | | | |
| h | | 0,1 | | | | | h | | 0,2 | | | | |
| i | x\_k | y\_k | m1 | m2 | m3 | m4 | i | x\_k | y\_k | m1 | m2 | m3 | m4 |
| 0 | 1,5708 | 1,0000 | 2,2074 | 2,3038 | 2,3068 | 2,3990 | 0 | 1,5708 | 1,0000 | 2,2074 | 2,3931 | 2,4042 | 2,4042 |
| 1 | 1,6708 | 1,1536 | 2,3529 | 2,4402 | 2,4428 | 2,5249 | 1 | 1,7708 | 1,3136 | 2,4773 | 2,6218 | 2,6296 | 2,6296 |
| 2 | 1,7708 | 1,3163 | 2,4788 | 2,5552 | 2,5573 | 2,6275 | 2 | 1,9708 | 1,6589 | 2,6570 | 2,7467 | 2,7510 | 2,7510 |
| 3 | 1,8708 | 1,4866 | 2,5819 | 2,6455 | 2,6472 | 2,7039 | 3 | 2,1708 | 2,0224 | 2,7233 | 2,7474 | 2,7484 | 2,7484 |
| 4 | 1,9708 | 1,6629 | 2,6590 | 2,7083 | 2,7095 | 2,7512 | 4 | 2,3708 | 2,3880 | 2,6590 | 2,6100 | 2,6080 | 2,6080 |
| 5 | 2,0708 | 1,8434 | 2,7075 | 2,7411 | 2,7419 | 2,7671 | 5 | 2,5708 | 2,7375 | 2,4539 | 2,3283 | 2,3236 | 2,3236 |
| 6 | 2,1708 | 2,0260 | 2,7249 | 2,7416 | 2,7420 | 2,7498 |  |  |  |  |  |  |  |
| 7 | 2,2708 | 2,2086 | 2,7094 | 2,7082 | 2,7081 | 2,6976 |  |  |  |  |  |  |  |
| 8 | 2,3708 | 2,3890 | 2,6594 | 2,6395 | 2,6391 | 2,6096 |  |  |  |  |  |  |  |
| 9 | 2,4708 | 2,5648 | 2,5739 | 2,5348 | 2,5341 | 2,4853 |  |  |  |  |  |  |  |
| 10 | 2,5708 | 2,7336 | 2,4523 | 2,3939 | 2,3928 | 2,3246 |  |  |  |  |  |  |  |

#include "koshi.h"

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

double \*\*method\_euler(func f, double a, double b, double y0,int\* count){

double x = a;

double y = y0;

double h = (b-a)/(\*count);

double \*\*result = malloc\_array(\*count);

result[0][0] = x;

result[0][1] = y;

for (int i = 0; i < \*count; i++){

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

x+=h;

result[i+1][0] = x;

result[i+1][1] = y;

}

return result;

}

double \*\*method\_euler\_cauchy(func f, double a, double b, double y0,int\* count){

double x = a;

double y = y0;

double h = (b-a)/(\*count);

double \*\*result = malloc\_array(\*count);

result[0][0] = x;

result[0][1] = y;

for (unsigned k = 0; k < \*count; k++){

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

x+=h;

result[k+1][0] = x;

result[k+1][1] = y;

}

return result;

}

double \*\*method\_mod\_euler(func f, double a, double b, double y0,int\* count){

double x = a;

double y = y0;

double h = (b-a)/(\*count);

double \*\*result = malloc\_array(\*count);

result[0][0] = x;

result[0][1] = y;

for (unsigned k = 0; k < \*count; k++){

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

result[k+1][0] = x;

result[k+1][1] = y;

x+=h;

}

return result;

}

double \*\*method\_runge\_kutta(func f, double a, double b, double y0,int\* count){

double x = a;

double y = y0;

double h = (b-a)/(\*count);

double m1, m2, m3, m4;

double \*\*result = malloc\_array(\*count);

result[0][0] = x;

result[0][1] = y;

for (unsigned k = 0; k < \*count; k++){

m1 = f(x, y);

m2 = f(x+h/2, y+h\*m1/2);

m3 = f(x+h/2, y+h\*m2/2);

m4 = f(x+h, y+h\*m3);

y = y+h/6\*(m1+2\*m2+2\*m3+m4);

x+=h;

result[k+1][0] = x;

result[k+1][1] = y;

}

return result;

}

void free\_array(double \*\*a, unsigned n){

for (unsigned i = 0; i<n; i++){

free(a[i]);

}

free(a);

}

double\*\* malloc\_array(unsigned n){

double \*\*result = (double \*\*)malloc(sizeof(double \*)\*(n+1));

for (int i = 0; i<=n; i++){

result[i] = (double \*)malloc(sizeof(double)\*2);

}

return result;

}

double \*\*runge\_eps(method\_func method,func fx, double a, double b, double y0,double eps,int \*count,int acc\_order){

double \*\*result1, \*\*result2;

double delta;

result1 = method(fx,a,b,y0,count);

(\*count)\*=2;

result2 = method(fx,a,b,y0,count);

delta = fabs(result2[\*count][1]-result1[\*count/2][1])/((1<<acc\_order)-1);

while (delta > eps){

free\_array(result1, \*count/2+1);

result1=result2;

(\*count)\*=2;

result2 = method(fx,a,b,y0,count);

delta = abs(result2[\*count][1]-result1[\*count/2][1])/((1<<acc\_order)-1);

}

free\_array(result1, \*count/2+1);

return result2;

}

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#include "koshi.h"

#define EPS 0.001

double var\_7(double x,double y){

return x\*sin(x)+y/x;

}

void print\_result(double \*\*table,int step){

printf("x%0\* y%0\*\n",5,5 );

for (int i=0;i<step;i++){

printf("%f %f\n",table[i][0],table[i][1]);

}

printf("\n");

}

int main(){

double y0,a,b;

int step,temp\_step;

double \*\*result;

y0=1;

a=1.570796327;

b=2.570796327;

step=2;

func f=var\_7;

method\_func method;

method=method\_euler;

temp\_step=step;

result=runge\_eps(method,var\_7,a,b,y0,EPS,&temp\_step,1);

printf("\nEuler's method calc on %d steps \n",temp\_step);

print\_result(result,temp\_step);

free(result);

method=method\_euler\_cauchy;

temp\_step=step;

result=runge\_eps(method,var\_7,a,b,y0,EPS,&temp\_step,2);

printf("\nEuler-Cauchy method calc on %d steps\n",temp\_step);

print\_result(result,temp\_step);

free(result);

method=method\_mod\_euler;

temp\_step=step;

result=runge\_eps(method,var\_7,a,b,y0,EPS,&temp\_step,2);

printf("\nmodificaton of Euler's method calc on %d steps\n",temp\_step);

print\_result(result,temp\_step);

 free(result);

method=method\_runge\_kutta;

temp\_step=step;

result=runge\_eps(method,var\_7,a,b,y0,EPS,&temp\_step,4);

printf("\nRunge-Kutta method calc on %d steps \n",temp\_step);

print\_result(result,temp\_step);

free(result);

}