МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Вятский государственный университет»

(ФГБОУ ВО «ВятГУ»)

Факультет автоматики и вычислительной техники

Кафедра ЭВМ

Отчёт

Лабораторная работа № 4 по дисциплине

«Вычислительная математика»

«Численное интегрирование и решение дифференциальных уравнений»

Вариант 7

Выполнил студент группы ИВТб-2301-04-00 / Жеребцов К. А./

Проверил преподаватель / Исупов К. С./

Киров 2021

**Цель работы**

Изучить методы численного интегрирования и решения дифференциальных уравнений.

**Задание**

1. Вычислить определённый интеграл с точностью до 0,0001. Выбрать значение n, обеспечивающее заданную точность, из формулы остаточного члена.

Задание:

Определённый интеграл от функции: 1/sqrt(3\*x^2-1)

Пределы интегрирования: [1,4;2,1]

Использовать формулу Симпсона.

2. Вычислить определённый интеграл с точностью до 0,0001 по другой квадратурной формуле:

Задание:

Определённый интеграл от функции: lg(x^2+1)/x

Пределы интегрирования: [0,8;1,6]

Использовать формулу трапеций.

В качестве начального шага взять число, близкое к Е^(1/m), где m=2. Для приближённой оценки погрешности применить принцип Рунге.

3. Вычислить определённый интеграл по квадратурной формуле Гаусса. Для оценки погрешности взять различное количество узлов:

n1=5; n2=8.

Квадратурная формула Гаусса с 5 узлами:

x1=-x5=-0,90618 A1=A5=0,23698

x2=-x4=-0,538469 A2=A4=0,47863

x3=0 A3=0,56889

Квадратурная формула Гаусса с 8 узлами:

x1=-x8=-0,96028986 A1=A8=0,10122854

x2=-x7=-0,79666648 A2=A7=0,22238103

x3=-x6=-0,52553242 A3=A6=0,31370664

x4=-x5=-0,18343464 A4=A5=0,36268378

Задание:

Определённый интеграл от функции: sqrt(x^2+1)/(x+2)

Пределы интегрирования: [0,2;2,4]

4. Определить значения всех интегралов, обратившись к встроенным функциям Mathcad.

5. Решить обыкновенное дифференциальное уравнение. Решение представить в табличной и графической формах. Для оценки погрешности выполнить расчёт с шагом h и с шагом h/2.

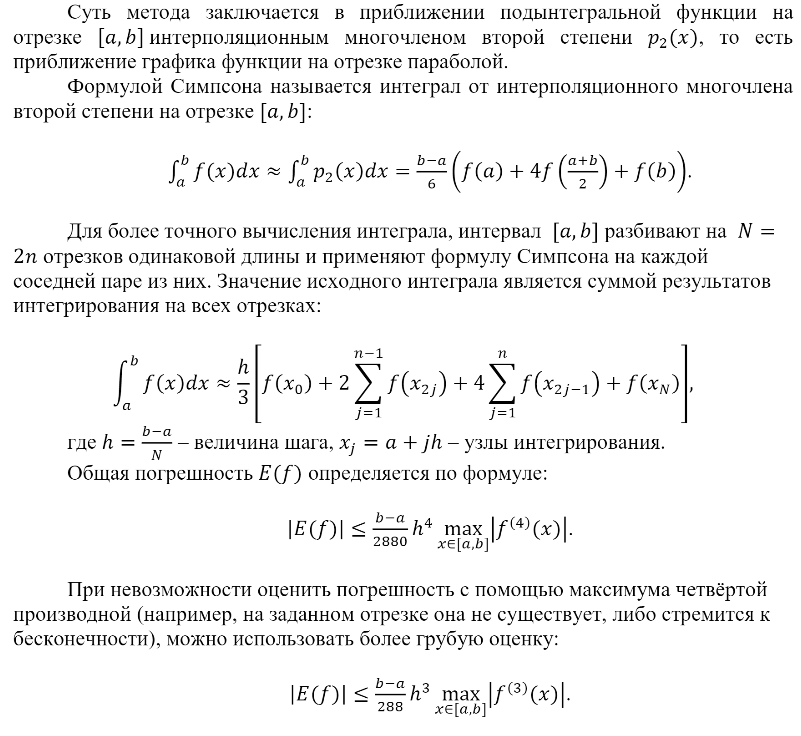
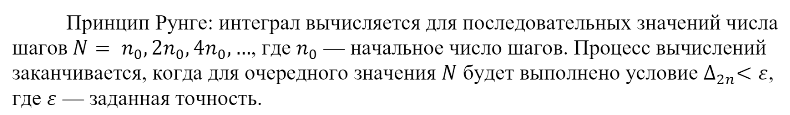
Задание:

По формуле 2-го порядка точности решить дифференциальное уравнение

y'=2 \* x^2 + 3 \* y

a=1/2; y(0)=0; h=0,1; 0<=x<=1

**Ход выполнения работы:**

1. Метод Симпсона

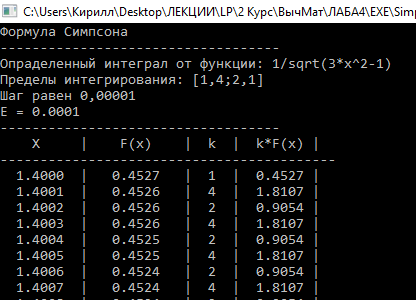


Рис. 1 – Результат выполнения программы.

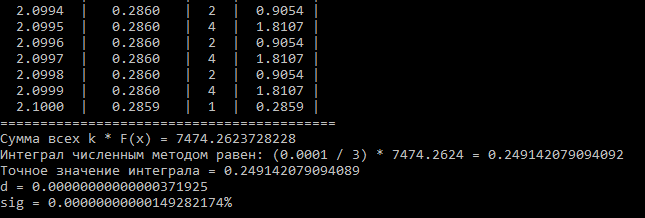


Рис. 2 – Результат выполнения программы.

Результат проверки:

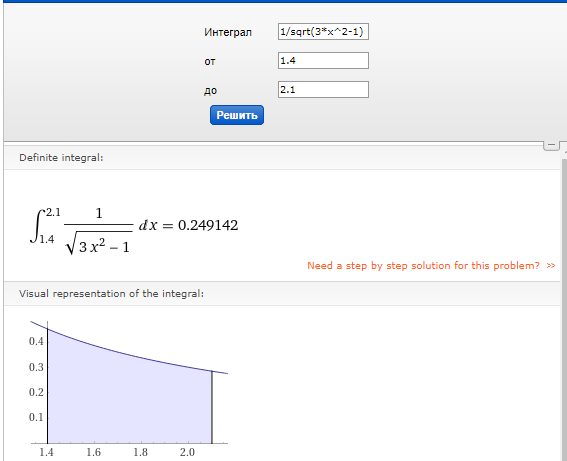


Рис. 3 – Результат проверки.

Листинг:

**program** simps;

**uses**

crt;

**const**

resh = 0.2491420790940887;

**var**

a: real = 1.4;

a1: real;

b: real = 2.1;

e: real = 0.0001;

s: real = 0;

f: real;

h: real = 0.0001;

i: integer = 1;

**function** integral(x: real): real;

**begin**

result := 1 / sqrt(3 \* (x \* x) - 1);

**end**;

**procedure** fsimps();

**begin**

a1 := a + h;

**while** a1 < b **do**

**begin**

**if** i **mod** 2 <> 0 **then**

**begin**

s := s + 4 \* integral(a1);

writeln(' ', a1:1:4, ' | ', integral(a1):1:4, ' | ', 4, ' | ', 4 \* integral(a):1:4, ' |');

**end**;

**if** i **mod** 2 = 0 **then**

**begin**

s := s + 2 \* integral(a1);

writeln(' ', a1:1:4, ' | ', integral(a1):1:4, ' | ', 2, ' | ', 2 \* integral(a):1:4, ' |');

**end**;

inc(i);

a1 := a1 + h;

**end**;

s := s + integral(a) + integral(b);

f := (h / 3) \* s;

**end**;

**begin**

writeln('Формула Симпсона');

Writeln('-----------------------------------');

writeln('Опраделенный интеграл от функции: 1/sqrt(3\*x^2-1)');

writeln('Пределы интегрирования: [1,4;2,1]');

writeln('Шаг равен 0,00001');

writeln('E = 0.0001');

Writeln('-----------------------------------');

writeln(' X | F(x) | k | k\*F(x) |');

Writeln('------------------------------------------');

writeln(' ', a:1:4, ' | ', integral(a):1:4, ' | ', 1, ' | ', integral(a):1:4, ' |');

fsimps;

writeln(' ', b:1:4, ' | ', integral(b):1:4, ' | ', 1, ' | ', integral(b):1:4, ' |');

Writeln('==========================================');

writeln('Сумма всех k \* F(x) = ', s:1:10);

writeln('Интеграл численным методом равен: (', h, ' / ', 3, ') \* ', s:1:4, ' = ', f:1:15);

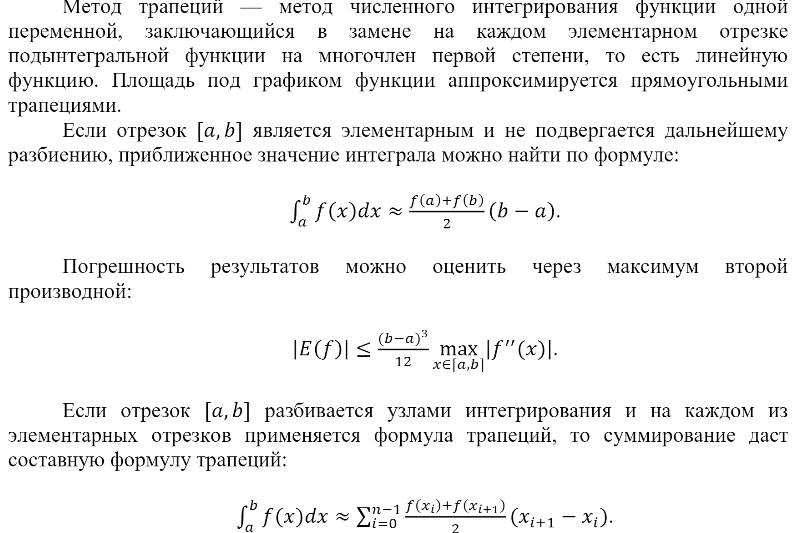
Writeln('Точное значение интеграла = ', resh:1:15);

writeln('d = ', abs(resh - f):1:20);

writeln('sig = ', abs(resh - f) / resh \* 100:1:20, '%');

readln();

**end**.

1. Интерполяция по Ньютону

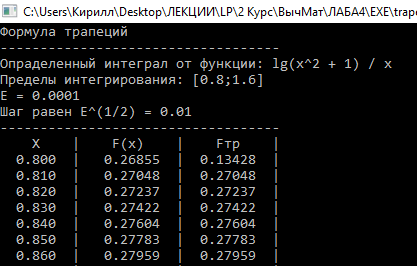


Рис. 4 – Результат выполнения программы.

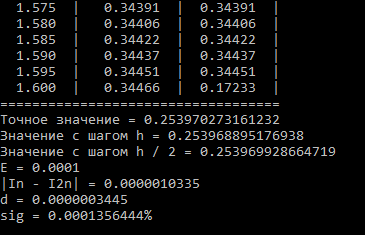


Рис. 5 – Результат выполнения программы.

Проверка:

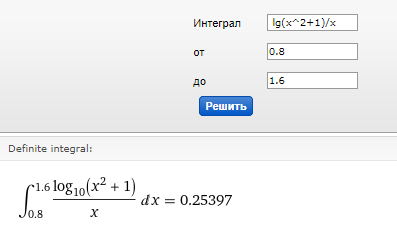


Рис. 6 – Результат вычислений.

Листинг:

**program** trapec;

**uses**

crt;

**const**

resh = 0.2539702731612316;

**var**

a: real = 0.8;

a1: real;

b: real = 1.6;

e: real = 0.0001;

s: real = 0;

ss: real = 0;

f, ff: real;

h: real = 0.01;

hh: real;

i: integer = 1;

**function** integral(x: real): real;

**begin**

result := (ln((x \* x) + 1) / ln(10)) / x;

**end**;

**procedure** ftrapec;

**begin**

hh := h / 2;

**repeat**

a1 := a + h;

writeln(' ', a:1:3, ' | ', integral(a):1:5, ' | ', 0.5 \* integral(a):1:5, ' |');

**while** a1 < b **do**

**begin**

s := s + integral(a1);

writeln(' ', a1:1:3, ' | ', integral(a1):1:5, ' | ', integral(a1):1:5, ' |');

a1 := a1 + h;

**end**;

writeln(' ', b:1:3, ' | ', integral(b):1:5, ' | ', 0.5 \* integral(b):1:5, ' |');

s := s + 0.5 \* integral(a) + 0.5 \* integral(b);

f := h \* s;

Writeln('===================================');

s := 0;

a1 := a + hh;

writeln(' ', a:1:3, ' | ', integral(a):1:5, ' | ', 0.5 \* integral(a):1:5, ' |');

**while** a1 < b - 0.001 **do**

**begin**

s := s + integral(a1);

writeln(' ', a1:1:3, ' | ', integral(a1):1:5, ' | ', integral(a1):1:5, ' |');

a1 := a1 + hh;

**end**;

writeln(' ', b:1:3, ' | ', integral(b):1:5, ' | ', 0.5 \* integral(b):1:5, ' |');

s := s + 0.5 \* integral(a) + 0.5 \* integral(b);

ff := hh \* s;

**if** abs(f - ff) > e **then**

**begin**

h := hh;

hh := hh / 2;

**end**;

**until** abs(f - ff) < e;

**end**;

**begin**

writeln('Формула трапеций');

Writeln('-----------------------------------');

writeln('Опраделенный интеграл от функции: lg(x^2 + 1) / x');

writeln('Пределы интегрирования: [0.8;1.6]');

writeln('E = 0.0001');

writeln('Шаг равен E^(1/2) = 0.01');

Writeln('-----------------------------------');

writeln(' X | F(x) | Fтр |');

ftrapec;

Writeln('===================================');

writeln('Точное значение = ', resh);

writeln('Значение с шагом h = ', f);

writeln('Значение с шагом h / 2 = ', ff);

writeln('E = 0.0001');

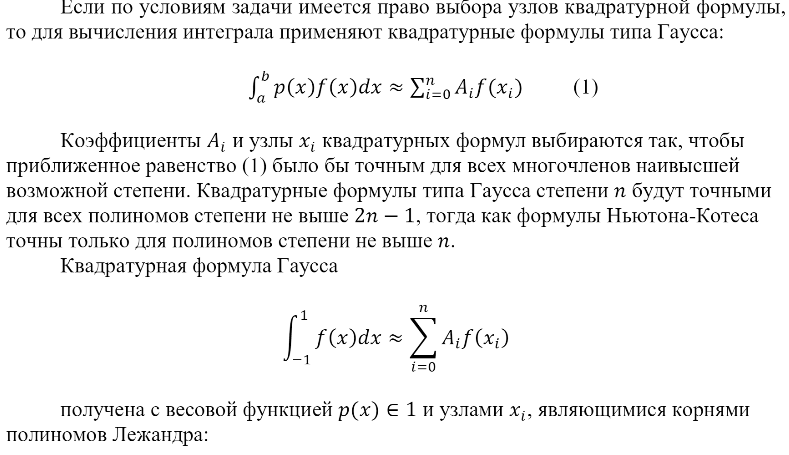
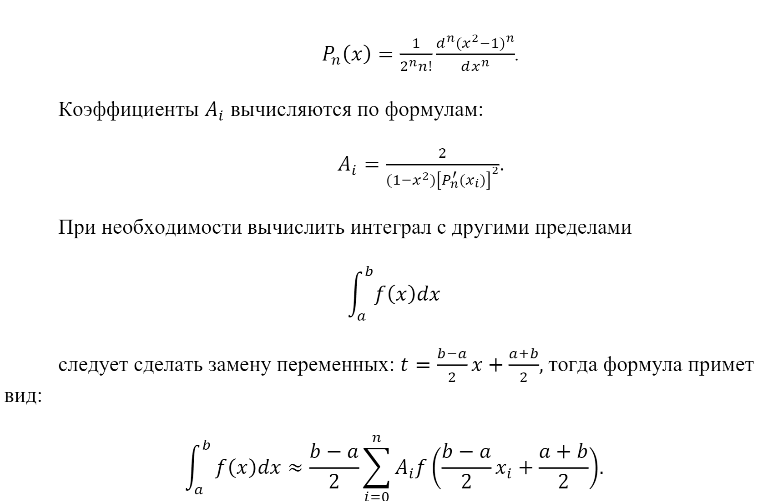
writeln('|In - I2n| = ', Abs(f - ff):1:10);

writeln('d = ', abs(resh - ff):1:10);

writeln('sig = ', abs(resh - ff) / resh \* 100:1:10, '%');

readln();

**end**.

1. Метод Гаусса

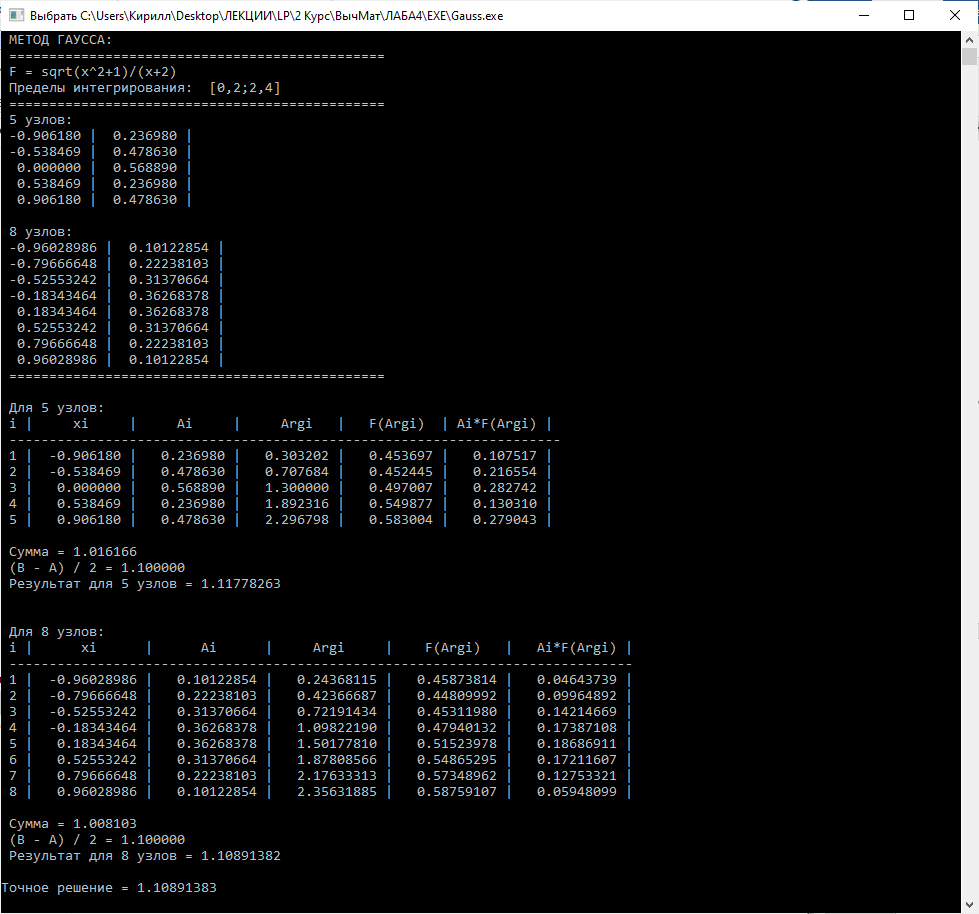


Рис. 7 – Результат выполнения программы.

Проверка:

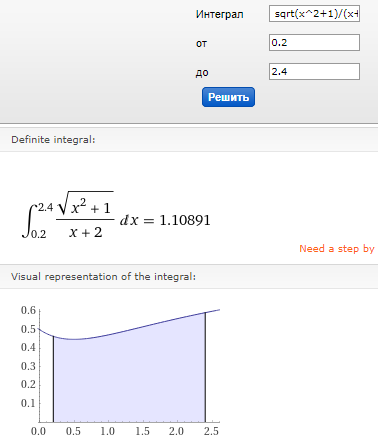


Рис. 8 – Результат проверки.

Листинг:

**program** Gauss;

**uses**

CRT;

**const**

A: Real = 0.2; {Нижний предел}

B: Real = 2.4; {Верхний предел}

resh: real = 1.10891382735557; {точное решение}

{таблица для 5 узлов}

tab1: **array**[1..5, 1..2] **of** real = (

(-0.90618, 0.23698),

(-0.538469, 0.47863),

(0, 0.56889),

(0.538469, 0.23698),

(0.90618, 0.47863));

{таблица для 8 узлов}

tab2: **array**[1..8, 1..2] **of** real = (

(-0.96028986, 0.10122854),

(-0.79666648, 0.22238103),

(-0.52553242, 0.31370664),

(-0.18343464, 0.36268378),

(0.18343464, 0.36268378),

(0.52553242, 0.31370664),

(0.79666648, 0.22238103),

(0.96028986, 0.10122854));

**var**

i, j: integer;

**function** F(X: real): real; {подинтегральная функция}

**begin**

F := sqrt((x \* x) + 1) / (x + 2);

**end**;

**function** arg(X: real): real; {аргументы}

**begin**

arg := ((b - a) / 2) \* x + (a + b) / 2;

**end**;

**procedure** node5(); {процедура для 5 узлов}

**var**

s: real = 0;

TabFor5: **array**[1..5, 1..5] **of** real;

i, j: integer;

integral: real;

**begin**

**for** i := 1 **to** 5 **do** {копирование точек в таблицу}

**begin**

**for** j := 1 **to** 2 **do**

**begin**

TabFor5[i, j] := tab1[i, j];

**end**;

**end**;

**for** i := 1 **to** 5 **do** {вычисление остальных столбцов и подсчет суммы элементов последнего}

**begin**

TabFor5[i, 3] := arg(tab1[i, 1]);

TabFor5[i, 4] := F(TabFor5[i, 3]);

TabFor5[i, 5] := TabFor5[i, 4] \* Tab1[i, 2];

s:= s + TabFor5[i, 5];

**end**;

writeln(' Для 5 узлов:');

writeln(' i | xi | Ai | Argi | F(Argi) | Ai\*F(Argi) |');

writeln(' ---------------------------------------------------------------------');

**for** i:=1 **to** 5 **do**

**begin**

write(' ', i, ' | ');

**for** j:= 1 **to** 5 **do**

**begin**

**if** tabfor5[i, j] < 0 **then**

write(' ', tabfor5[i, j]:0:6, ' | ')

**else**

write(' ', tabfor5[i, j]:0:6, ' | ')

**end**;

writeln();

**end**;

writeln();

writeln(' Сумма = ', s:0:6);

writeln(' (B - A) / 2 = ', (b - a)/2:0:6);

integral:= s \* ((b - a)/2);

writeln(' Результат для 5 узлов = ', integral:0:8);

**end**;

**procedure** node8(); {процедура для 8 узлов}

**var**

s: real = 0;

TabFor8: **array**[1..8, 1..5] **of** real;

i, j: integer;

integral: real;

**begin**

**for** i := 1 **to** 8 **do**

**begin**

**for** j := 1 **to** 2 **do**

**begin**

TabFor8[i, j] := tab2[i, j];

**end**;

**end**;

**for** i := 1 **to** 8 **do**

**begin**

TabFor8[i, 3] := arg(tab2[i, 1]);

TabFor8[i, 4] := F(TabFor8[i, 3]);

TabFor8[i, 5] := TabFor8[i, 4] \* Tab2[i, 2];

s:= s + TabFor8[i, 5];

**end**;

writeln(' Для 8 узлов:');

writeln(' i | xi | Ai | Argi | F(Argi) | Ai\*F(Argi) |');

writeln(' ------------------------------------------------------------------------------');

**for** i:=1 **to** 8 **do**

**begin**

write(' ', i, ' | ');

**for** j:= 1 **to** 5 **do**

**begin**

**if** tabfor8[i, j] < 0 **then**

write(' ', tabfor8[i, j]:0:8, ' | ')

**else**

write(' ', tabfor8[i, j]:0:8, ' | ')

**end**;

writeln();

**end**;

writeln();

writeln(' Сумма = ', s:0:6);

writeln(' (B - A) / 2 = ', (b - a)/2:0:6);

integral:= s \* ((b - a)/2);

writeln(' Результат для 8 узлов = ', integral:0:8);

**end**;

**begin**

writeln(' МЕТОД ГАУССА:');

writeln(' ===============================================');

writeln(' F = sqrt(x^2+1)/(x+2)');

writeln(' Пределы интегрирования: [0,2;2,4]');

writeln(' ===============================================');;

writeln(' 5 узлов:');

**for** i := 1 **to** 5 **do**

**begin**

**for** j := 1 **to** 2 **do**

**begin**

**if** tab1[i, j] < 0 **then**

write(' ', tab1[i, j]:0:6, ' |')

**else**

write(' ', tab1[i, j]:0:6, ' |')

**end**;

writeln;

**end**;

writeln();

writeln(' 8 узлов:');

**for** i := 1 **to** 8 **do**

**begin**

**for** j := 1 **to** 2 **do**

**begin**

**if** tab2[i, j] < 0 **then**

write(' ', tab2[i, j]:0:8, ' |')

**else**

write(' ', tab2[i, j]:0:8, ' |')

**end**;

writeln;

**end**;

writeln(' ===============================================');

writeln();

node5();

writeln();

writeln();

node8();

writeln();

Writeln('Точное решение = ', resh:0:8);

readln();

**end**.

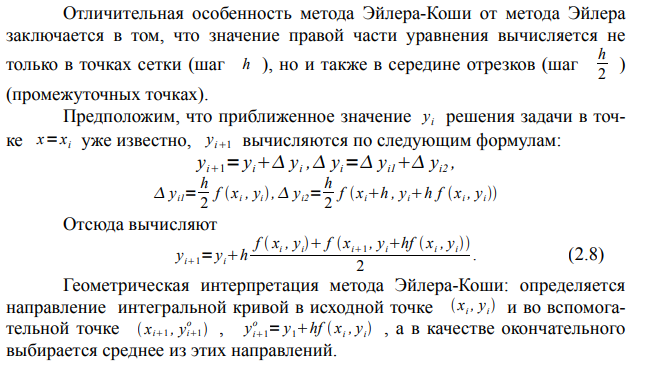
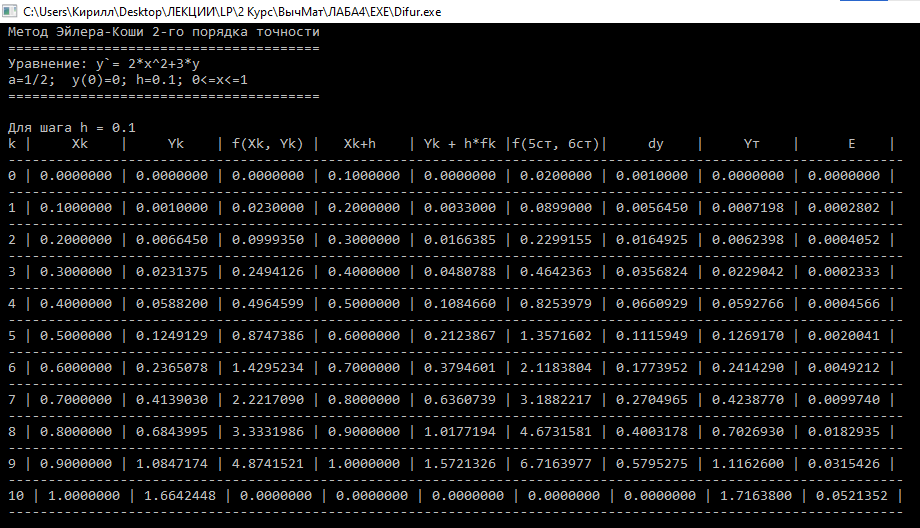
1. Метод Эйлера-Коши 2-го порядка точности

Рис. 9 – Результат выполнения программы.

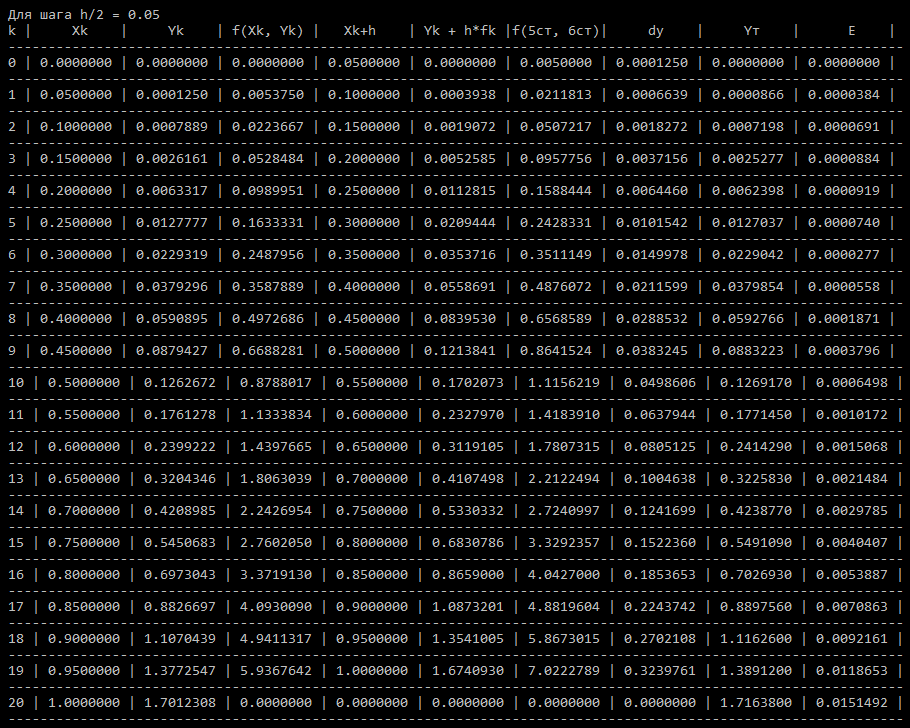


Рис. 10 – Результат выполнения программы.

Проверка:

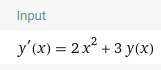
****

Рис. 11 – Исходные данные.

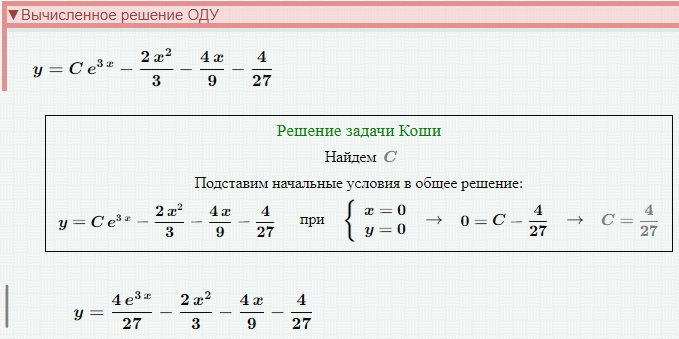


Рис. 12 – Результат вычислений.

Листинг:

**Program** difur;

**uses**

crt;

**var**

alf: real = 0.5; {альфа}

x0: real = 0; y0: real = 0; {начальное условие}

hh: real = 0.1; {шаг}

a: real = 0; {левая граница}

b: real = 1; {правая граница}

tab: **array**[0..49, 0..8] **of** real; {таблица}

i, j: integer;

xx, yy: **array** [0..49] **of** real;

n: integer;

Yt1: **array**[0..10] **of** real = (0,

0.0007198,

0.0062398,

0.0229042,

0.0592766,

0.126917,

0.241429,

0.423877,

0.702693,

1.11626,

1.71638);

Yt2: **array**[0..20] **of** real = (0,

0.0000865545,

0.0007198,

0.00252773,

0.0062398,

0.0127037,

0.0229042,

0.0379854,

0.0592766,

0.0883223,

0.126917,

0.177145,

0.241429,

0.322583,

0.423877,

0.549109,

0.702693,

0.889756,

1.11626,

1.38912,

1.71638);

**function** f(x, y: real): real; {правая часть уравнения}

**begin**

result:= 2 \* x \* x + 3 \* y;

**end**;

**procedure** dif(h: real);

**var**

i, j: integer;

x: real;

dy: real;

**begin**

tab[0,0]:= x0;

tab[0,1]:= y0;

tab[0,2]:= f(tab[0,0], tab[0,1]);

tab[0,3]:= tab[0,0] + h;

tab[0,4]:= tab[0,1] + h \* tab[0,2];

tab[0,5]:= f(tab[0,3], tab[0,4]);

tab[0,6]:= (h/2) \* (tab[0,5] + tab[0,2]);

x:= x0;

**for** i:= 1 **to** n **do**

**begin**

x:= x + h;

**for** j:= 0 **to** 8 **do**

**begin**

**if** j = 0 **then** tab[i,j]:= x; {вычисление Х}

**if** j = 1 **then** tab[i,j]:= tab[i-1, j] + ((h/2) \* ( f(tab[i-1,j-1], tab[i-1, j]) + f(tab[i-1,j-1] + h, tab[i-1, j] + (h \* f(tab[i-1,j-1], tab[i-1, j]))) )); {вычисление Y}

**if** j = 2 **then** tab[i,j]:= f(tab[i,j-2], tab[i,j-1]);

**if** j = 3 **then** tab[i,j]:= tab[i, j-3] + h;

**if** j = 4 **then** tab[i,j]:= tab[i,j-3] + h \* tab[i,j-2];

**if** j = 5 **then** tab[i,j]:= f(tab[i,j-2], tab[i,j-1]);

**if** j = 6 **then** tab[i,j]:= (h/2) \* (tab[i,j-1] + tab[i,j-4]);

**end**;

**end**;

**for** i:=0 **to** n **do** tab[i,8]:= abs(tab[i,1] - tab[i,7]);

**for** j:=2 **to** 6 **do**

**begin**

tab[n,j]:=0;

**end**;

writeln(' k | Xk | Yk | f(Xk, Yk) | Xk+h | Yk + h\*fk |f(5ст, 6ст)| dy | Yт | E |');

writeln(' ----------------------------------------------------------------------------------------------------------------');

**for** i:= 0 **to** n **do**

**begin**

write(' ',i,' |');

**for** j:= 0 **to** 8 **do**

**begin**

write(' ', tab[i,j]:0:7, ' |');

**end**;

writeln;

writeln(' ----------------------------------------------------------------------------------------------------------------');

**end**;

**end**;

**begin**

writeln(' Метод Эйлера-Коши 2-го порядка точности');

writeln(' =======================================');

writeln(' Уравнение: y`= 2\*x^2+3\*y');

writeln(' a=1/2; y(0)=0; h=0.1; 0<=x<=1');

writeln(' =======================================');

writeln();

writeln(' Для шага h = 0.1');

n:= round((b - a)/hh);

**for** i:= 0 **to** n **do** tab[i,7]:= yt1[i];

dif(hh);

writeln();

writeln();

writeln();

writeln(' Для шага h/2 = 0.05');

n:= round((b - a)/(hh/2));

**for** i:= 0 **to** n **do** tab[i,7]:= yt2[i];

dif(hh/2);

readln();

**end**.

**Вывод**

В ходе данной лабораторной работы были подробно изучены методы численного интегрирования. Также были изучены численные методы решения дифференциальных уравнений. После изучения материала по данным темам, была написана программа, в которой реализовано решение заданий, предоставленных на лабораторную работу, при помощи соответствующих методов. Полученные при решении ответы можно считать правильными, т.к. они совпадают с ответами, которые получаются при решении соответствующих заданий при помощи сервиса Wolfram Alpha.