***Лабораторная работа № 3***

Индивидуальное задание №10

***Тема:*** Детерминированные вычислительные процессы с управлением по аргументу. Численное интегрирование.

***Цель:*** Научиться реализовывать алгоритмы детерминированных вычислительных процессов управление которыми осуществляется по аргументу. Разработать алгоритмы, основанные на ДЦВП для вычисления определенного интеграла.

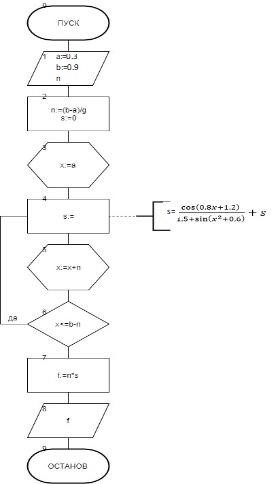
***Используемое оборудование:*** ПК, PascalABC

***Задание №1:***

***Постановка задачи:*** Написать программу для вычисления определенного интеграла методом прямоугольника левых частей.

***Математическая модель:***

***Блок-схема:***

******

***Список идентификаторов:***

|  |  |  |
| --- | --- | --- |
| ***Имя*** | ***Смысл*** | ***Тип*** |
| a | Значение нижнего предела интегрирования | real |
| b | Значение верхнего предела интегрирования | real |
| g | Значение количества шагов | real |
| n | Значение величины шага | real |
| s | Значение суммы функции | real |
| f | Результат | real |
| x | Параметр цикла | real |

***Код программы:***

**program** zadanie1;

**const**

a=0.3;

b=0.9;

**var** g,s,x,n,f: real;

**begin**

writeln('Количество шагов');

readln(g);

s:=0;

n:=(b-a)/g;

x:=a;

**while** x<=b-n **do**

**begin**

s:=s+(cos(0.8\*x+1.2))/(1.5+sin(x\*x+0.6));

x:=x+n;

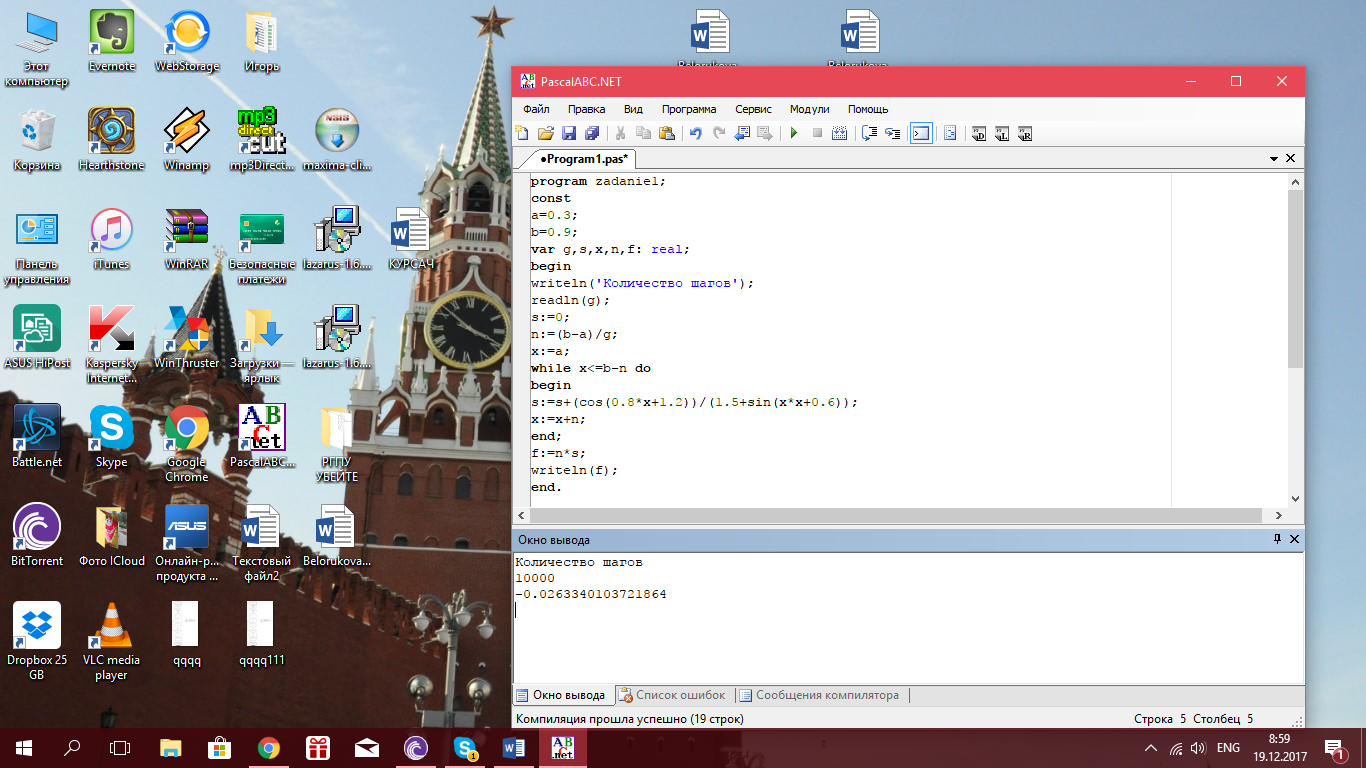
**end**;

f:=n\*s;

writeln(f);

**end**.

***Результат вычислений:***



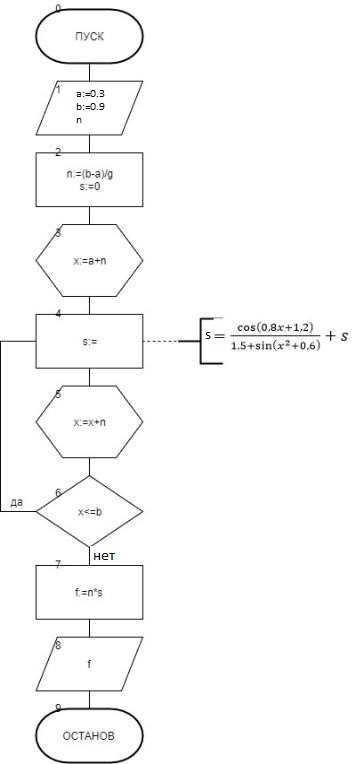
***Анализ результатов вычислений:*** Программа была получена при помощи использования детерминированного циклического процесса. Мы использовали конструкцию While для создания цикла (из-за шага n типа “real”). Переменные a, b, g, которые мы вводили с клавиатуры помогли высчитать шаг(n). Задача цикла заключалась в накоплении суммы (s типа “real”). Диапазон значений был от a до b-n т.к. мы использований метод левых частей прямоугольников.

***Задание №2:***

***Постановка задачи:*** Написать программу для вычисления определенного интеграла методом прямоугольника правых частей.

***Математическая модель:***

***Блок-схема:***

******

***Список идентификаторов:***

|  |  |  |
| --- | --- | --- |
| ***Имя*** | ***Смысл*** | ***Тип*** |
| a | Значение нижнего предела интегрирования | real |
| b | Значение верхнего предела интегрирования | real |
| g | Значение количества шагов | real |
| n | Значение величины шага | real |
| s | Значение суммы функции | real |
| f | Результат | real |
| x | Параметр цикла | real |

***Код программы:***

**program** zadanie2;

**const**

a=0.3;

b=0.9;

**var** g,s,x,n,f: real;

**begin**

writeln('Количество шагов');

readln(g);

s:=0;

n:=(b-a)/g;

x:=a+n;

**while** x<=b **do**

**begin**

s:=s+(cos(0.8\*x+1.2))/(1.5+sin(x\*x+0.6));

x:=x+n;

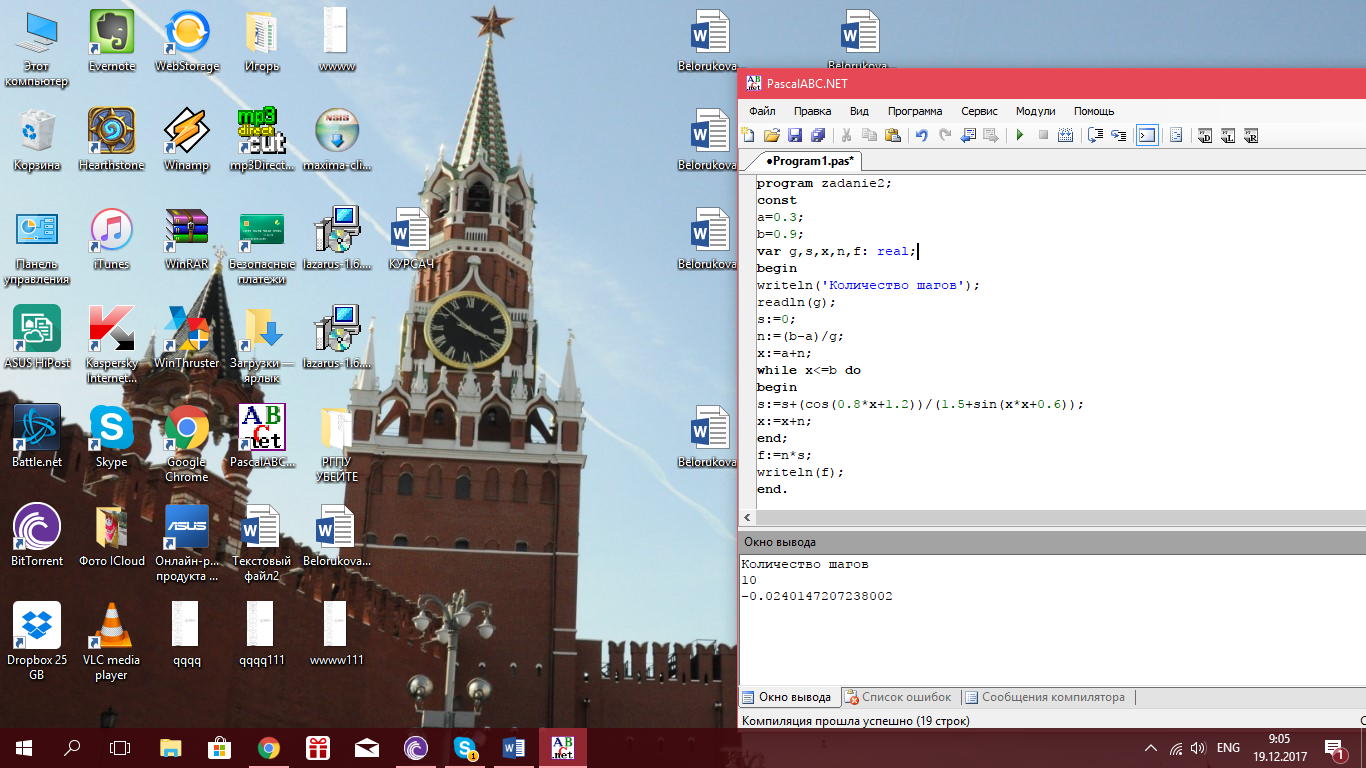
**end**;

f:=n\*s;

writeln(f);

**end**.

***Результат вычислений:***



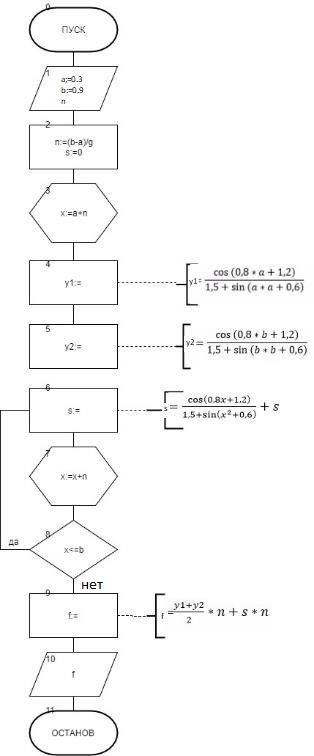
***Анализ результатов вычислений:*** Программа была получена при помощи использования детерминированного циклического процесса. Мы использовали конструкцию While для создания цикла (из-за шага n типа “real”). Переменные a, b, g, которые мы вводили с клавиатуры помогли высчитать шаг(n). Задача цикла заключалась в накоплении суммы (s типа “real”). Диапазон значений был от a+n до b т.к. мы использований метод правых частей прямоугольников.

***Задание №3:***

***Постановка задачи:*** Написать программу для вычисления определенного интеграла методом трапеций.

***Математическая модель:***

***Блок-схема:***

******

***Список идентификаторов:***

|  |  |  |
| --- | --- | --- |
| ***Имя*** | ***Смысл*** | ***Тип*** |
| a | Значение нижнего предела интегрирования | real |
| b | Значение верхнего предела интегрирования | real |
| g | Значение количества шагов | real |
| n | Значение величины шага | real |
| x | Параметр цикла | real |
| s | Значение суммы функции | real |
| y1 | Значение функции при аргументе равном, а | real |
| y2 | Значение функции при аргументе равном, b | real |
| f | Результат | real |

***Код программы:***

**program** zadanie2;

**const**

a=0.3;

b=0.9;

**var** g,s,x,n,f,y1,y2: real;

**begin**

writeln('Количество шагов');

readln(g);

s:=0;

n:=(b-a)/g;

x:=a+n;

y1:=(cos(0.8\*a+1.2))/(1.5+sin(a\*a+0.6));

y2:=(cos(0.8\*b+1.2))/(1.5+sin(b\*b+0.6));

**while** x<=b-n **do**

**begin**

s:=s+(cos(0.8\*x+1.2))/(1.5+sin(x\*x+0.6));

x:=x+n;

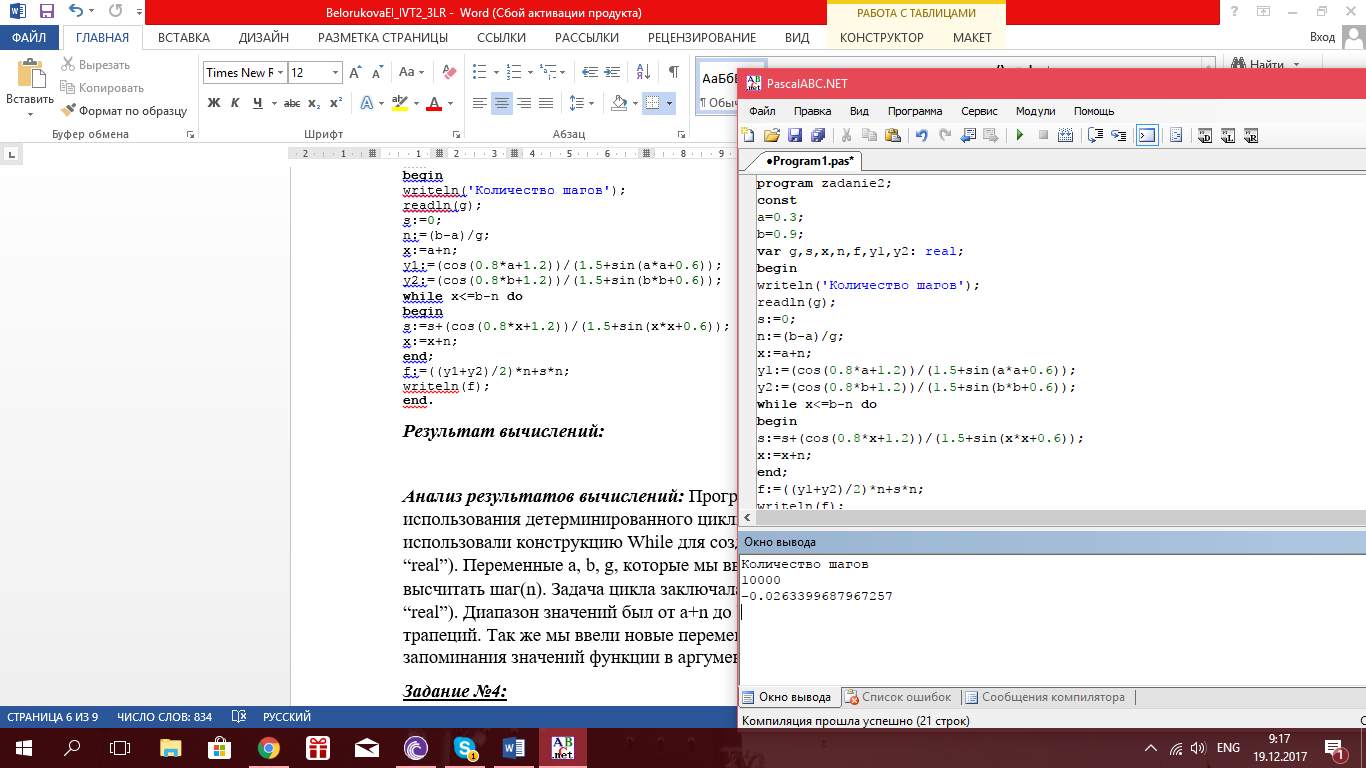
**end**;

f:=((y1+y2)/2)\*n+s\*n;

writeln(f);

**end**.

***Результат вычислений:***



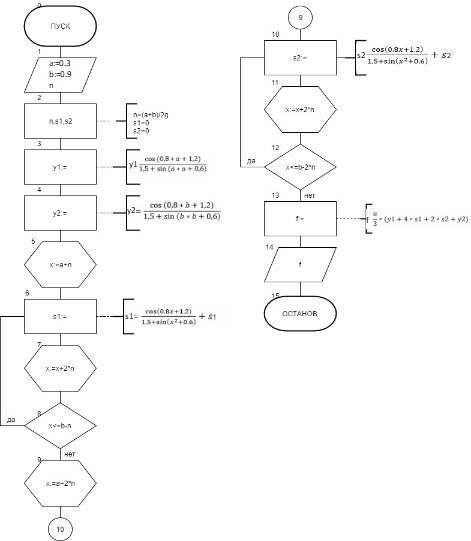
***Анализ результатов вычислений:*** Программа была получена при помощи использования детерминированного циклического процесса. Мы использовали конструкцию While для создания цикла (из-за шага n типа “real”). Переменные a, b, g, которые мы вводили с клавиатуры помогли высчитать шаг(n). Задача цикла заключалась в накоплении суммы (s типа “real”). Диапазон значений был от a+n до b-n т.к. мы использований метод трапеций. Так же мы ввели новые переменные y1, y2 типа “real” для запоминания значений функции в аргументах a и b (начальные значения).

***Задание №4:***

***Постановка задачи:*** Написать программу для вычисления определенного интеграла методом парабол.

***Математическая модель:***

***Блок-схема:***

******

***Список идентификаторов:***

|  |  |  |
| --- | --- | --- |
| ***Имя*** | ***Смысл*** | ***Тип*** |
| a | Значение нижнего предела интегрирования | real |
| b | Значение верхнего предела интегрирования | real |
| g | Значение количества шагов | real |
| n | Значение величины шага | real |
| x | Параметр цикла | real |
| s1 | Значение суммы функции в четных позициях | real |
| s2 | Значение суммы функции в нечетных позициях | real |
| y1 | Значение функции при аргументе равном, а | real |
| y2 | Значение функции при аргументе равном, b | real |
| f | Результат | real |

***Код программы:***

**program** zadanie2;

**const**

a=0.3;

b=0.9;

**var** g,s1,s2,x,n,f,y1,y2: real;

**begin**

writeln('Количество шагов');

readln(g);

s1:=0;

s2:=0;

n:=(b-a)/2\*g;

x:=a+n;

y1:=(cos(0.8\*a+1.2))/(1.5+sin(a\*a+0.6));

y2:=(cos(0.8\*b+1.2))/(1.5+sin(b\*b+0.6));

**while** x<=b-n **do**

**begin**

s1:=s1+(cos(0.8\*x+1.2))/(1.5+sin(x\*x+0.6));

x:=x+2\*n;

**end**;

x:=a+2\*n;

**while** x<=b-2\*n **do**

**begin**

s2:=s2+(cos(0.8\*x+1.2))/(1.5+sin(x\*x+0.6));

x:=x+2\*n;

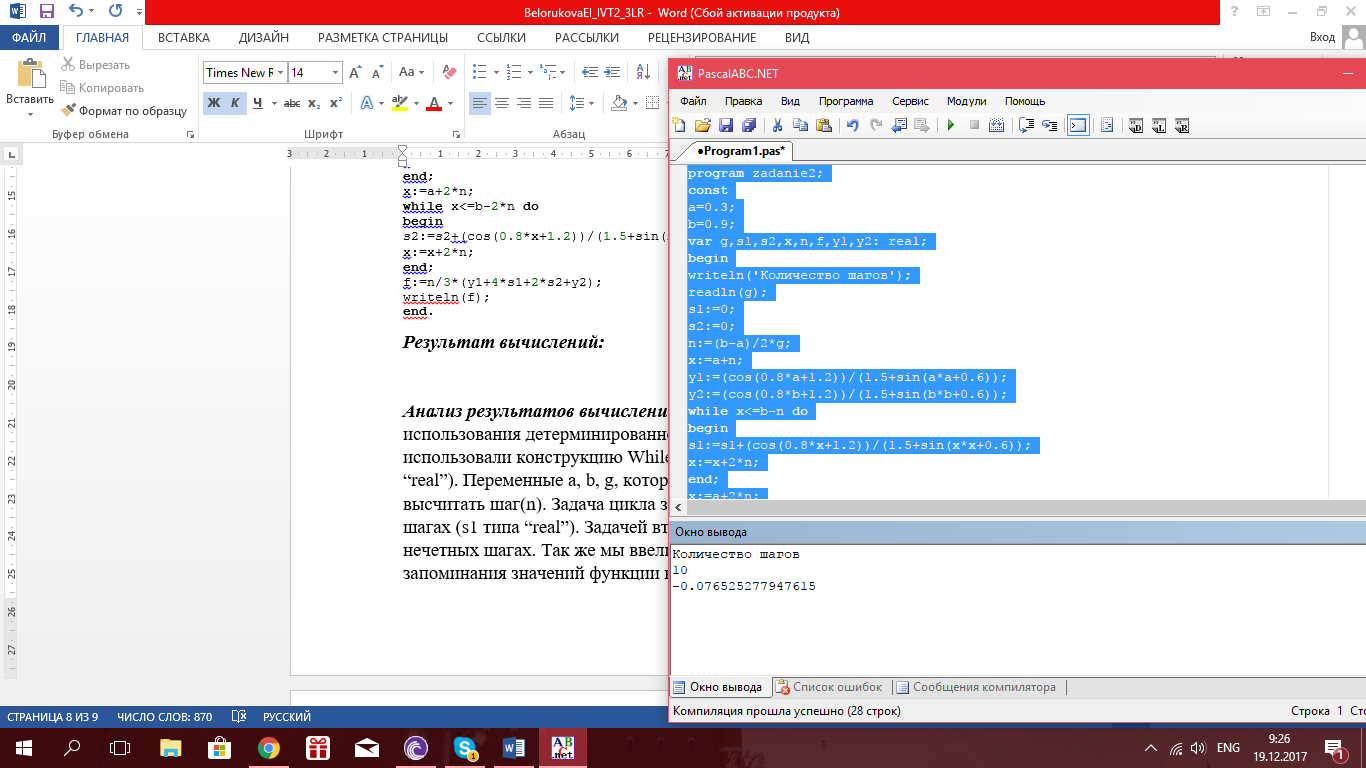
**end**;

f:=n/3\*(y1+4\*s1+2\*s2+y2);

writeln(f);

**end**.

***Результат вычислений:***



***Анализ результатов вычислений:*** Программа была получена при помощи использования детерминированного циклического процесса. Мы использовали конструкцию While для создания цикла (из-за шага n типа “real”). Переменные a, b, g, которые мы вводили с клавиатуры помогли высчитать шаг(n). Задача цикла заключалась в накоплении суммы в четных шагах (s1 типа “real”). Задачей второго цикла было накопление суммы в нечетных шагах. Так же мы ввели новые переменные y1, y2 типа “real” для запоминания значений функции в аргументах a и b (начальные значения). Изменение аргумента происходило по формуле x = x + 2h. Каждый цикл имел свой диапазон значений (1. От a+n до b-n, 2. От a+2n до b-2n) т.к. мы пользовались методом парабол (метод Симпсона)

***Таблица сравнения результатов для разных методов:***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N  Количество разбиений | H  шаг | I  Метод левых частей прямоугольника | I  Метод правых частей прямоугольника | I  Метод трапеций | I  Метод парабол |
| 10 | 0,06 | -0.0131310006486793 | -0.0240147207238002 | -0.0190894251880069 | -0.076525277947615 |
| 100 | 0,006 | -0.0249287204040626 | -0.0261100985220978 | -0.0255245628579953 | -0.0255245628579953 |
| 1000 | 0,0006 | -0.0261979428768588 | -0.0263170082981475 | -0.0262575271222521 | -0.0263170082981475 |
| 10000 | 0,00006 | -0.0263340103721864 | -0.026345927221265 | -0.0263399687967257 | -0.0263340103721864 |

***Вывод:*** Наиболее точным способом вычисления определенного интеграла является метод Симпсона (метод парабол). Мы произвели вычисления всех 4 методов при помощи ДЦВП. При использовании всех этих 4 методов мы получали одинаковые значения до 3 знаков после запятой. Мы научились реализовывать алгоритмы детерминированных вычислительных процессов управление которыми осуществляется по аргументу. Разработать алгоритмы, основанные на ДЦВП для вычисления определенного интеграла. Увеличить же точность можно за счет уменьшения шага, получая в результате большее количество точек разбиения