Лабораторная работа №3. Детерминированные вычислительные процессы

с управлением по аргументу. Численное интегрирование.

Цель л/р: рассмотреть методы численного интегрирования с использованием ДЦВП с управлением по аргументу средствами Lazarus и Free Pascal.

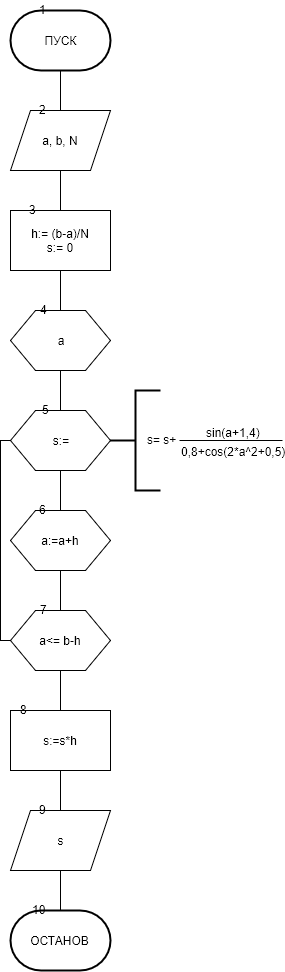
Используемое оборудование: ПК, Lazarus, Word, Draw.io

Задание 1.

1. Написать программу для вычисления определенного интеграла из индивидуального задания методом прямоугольника левых частей. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы 3.



1. Математическая модель:
2. Блок-схема



1. Список идентификаторов

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | Левая граница интервала интегрирования | real |
| b | Правая граница интервала интегрирования | real |
| N | Количество разбиений интервала | word |
| h | Величина шага | real |
| s | Результат | real |

1. Код программы:

program Project1;

var a, b, h, s: real;

N: word;

begin

writeln('Enter N');

readln(N);

a:= 0.4;

b:= 1.0;

h:=(b-a)/N;

s:= 0;

while a <= b-h do begin

s:=s+(sin(a+1.4))/(0.8+cos(2\*a\*a+0.5));

a:=a+h;

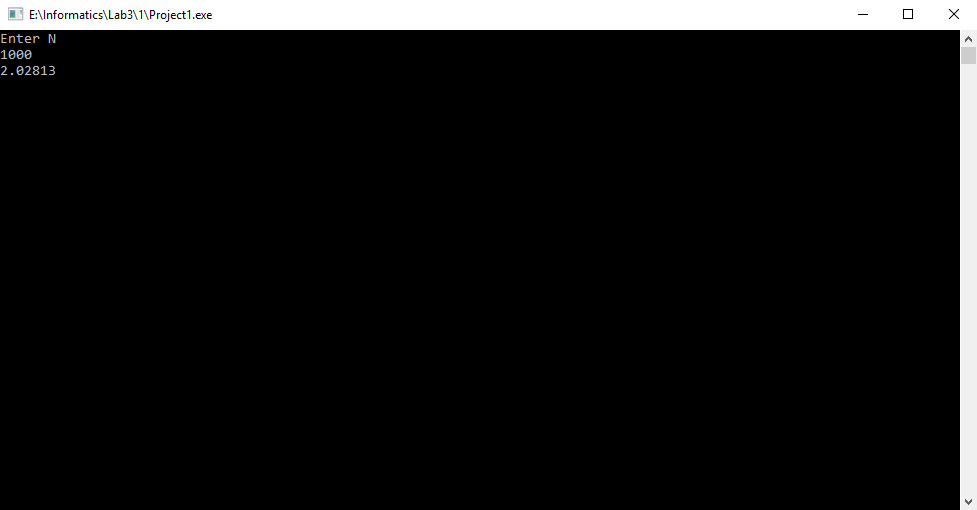
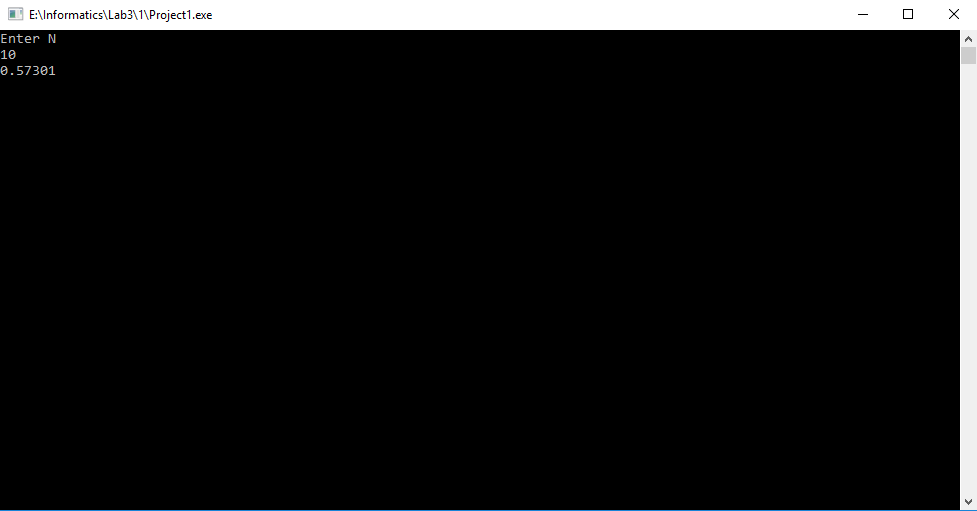
end;

s:= s\*h;

writeln(s:5:5);

readln();

end.

1. Результаты выполненной работы: 
2. Анализ результатов вычисления:

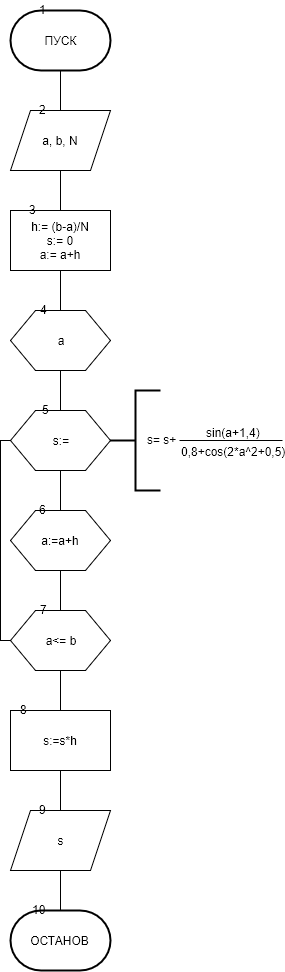
Входная переменная N хранит значения до 65535, следовательно, программа поддерживает до 65535 разбиений интегрируемого интервала. Цикл вычислений проходит с a до b-h. Переменная a после вычисления величины шага используется в качестве счётчика и аргумента выражения, заключенного в теле цикла. Точность вычисления напрямую зависит от количества разбиений интервала – величины N. Чем больше значение N, тем более точным окажется результат.

Задание 2.

1. Написать программу для вычисления определенного интеграла из индивидуального задания методом прямоугольника правых частей. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы 3.



1. Математическая модель:
2. Блок-схема



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | Левая граница интегрирования | real |
| b | Правая граница интегрирования | real |
| N | Количество разбиений интервала | word |
| h | Величина шага | real |
| s | Результат | real |

1. Код программы:

program project2;

var a, b, h, s: real;

N: word;

begin

writeln('Enter N');

readln(N);

a:= 0.4;

b:= 1.0;

h:=(b-a)/N;

s:= 0;

a:= a+h;

while a <= b do begin

s:=s+(sin(a+1.4))/(0.8+cos(2\*a\*a+0.5));

a:=a+h;

end;

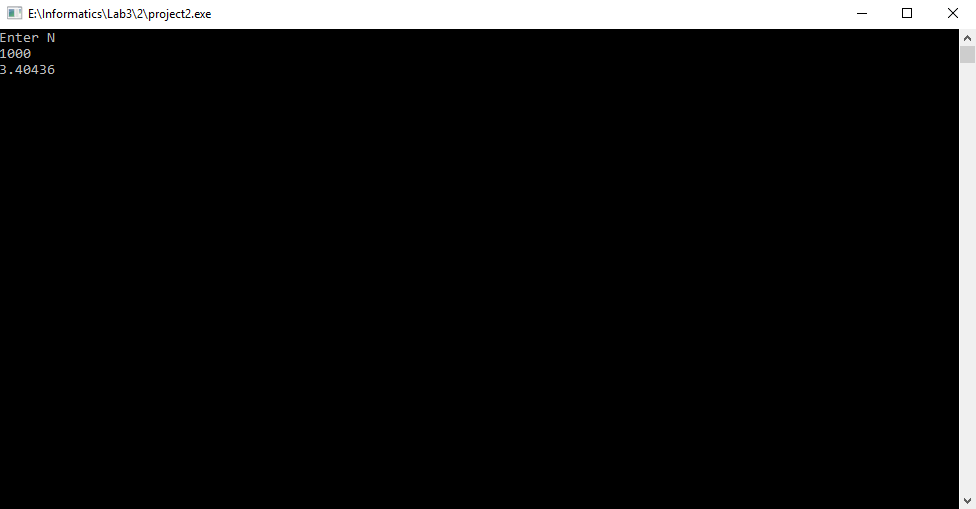
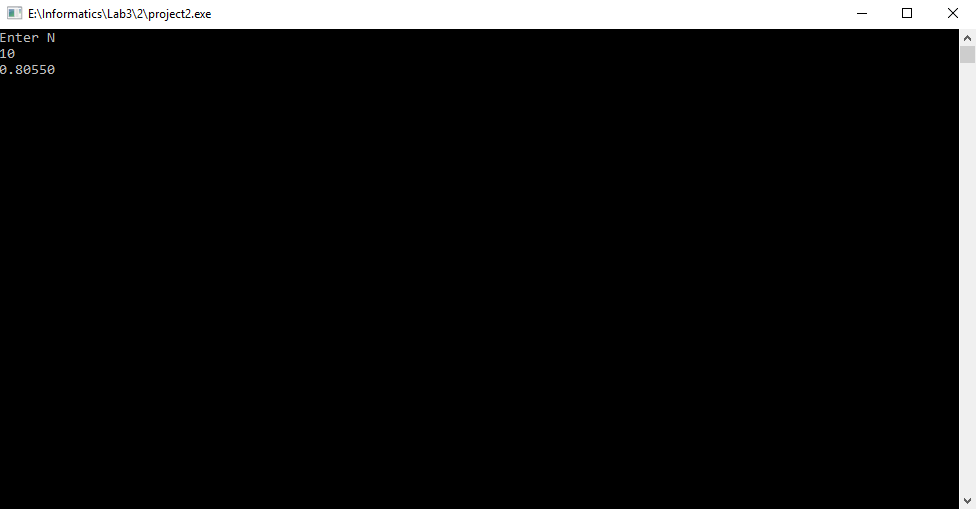
s:= s\*h;

writeln(s:5:5);

readln();

end.

1. Результаты вычисления:



1. Анализ результатов вычисления:

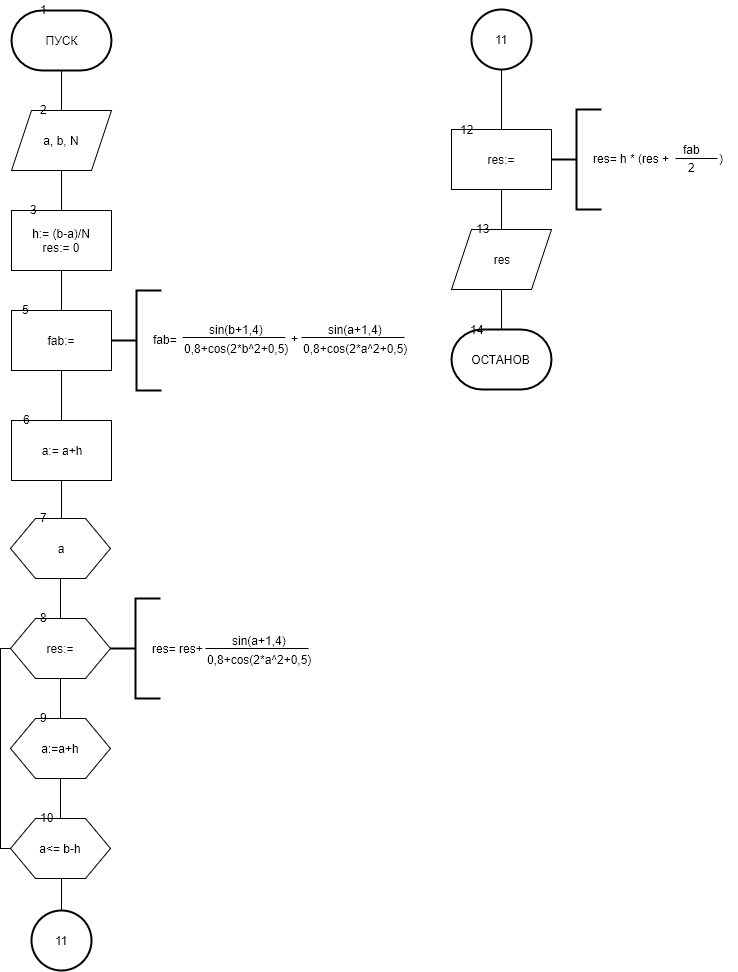
Входная переменная N хранит значения до 65535, следовательно, программа поддерживает до 65535 разбиений интегрируемого интервала. Цикл вычислений проходит a + h до b. Точность нахождения определенного интеграла определяется количеством разбиений N.

Задание 3.

1. Написать программу для вычисления определенного интеграла из индивидуального задания методом трапеций. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы 3.



1. Математическая модель:
2. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | Левая граница интегрируемого интервала | real |
| b | Правая граница интегрируемого интервала | real |
| h | Величина шага | real |
| N | Количество разбиений интервала | word |
| fab | Сумма значений функции в точках a и b | real |
| res | Результат | real |

1. Код программы:

program Project3;

var a, b, h, fab, res: real;

N: word;

begin

writeln('Enter N');

readln(N);

a:= 0.4;

b:= 1.0;

h:=(b-a)/N;

fab:=(sin(b+1.4))/(0.8+cos(2\*b\*b+0.5)) + (sin(a+1.4))/(0.8+cos(2\*a\*a+0.5));

a:= a+h;

res:= 0;

while a <= b-h do begin

res:= res+(sin(a+1.4))/(0.8+cos(2\*a\*a+0.5));

a:=a+h;

end;

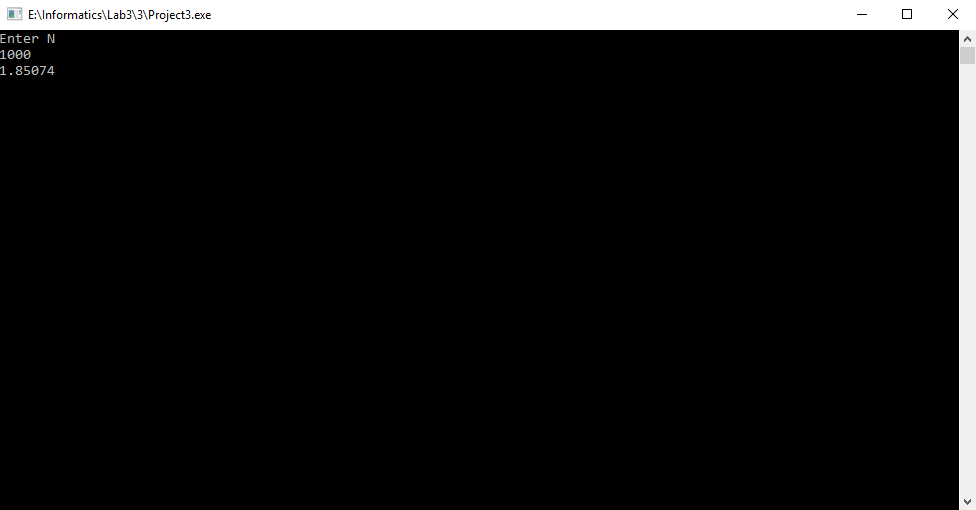
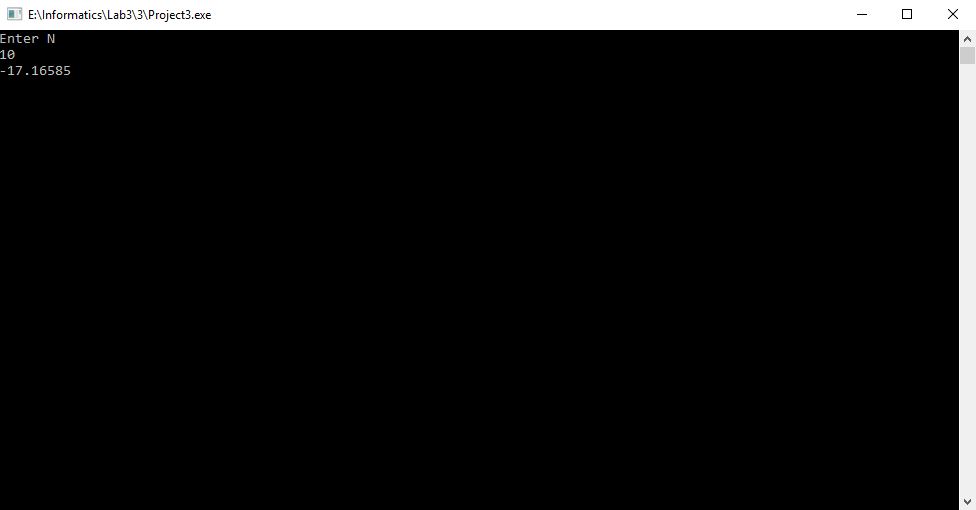
res:=h\*(fab/2+res);

writeln(res:5:5);

readln();

end.

1. Результаты вычисления:

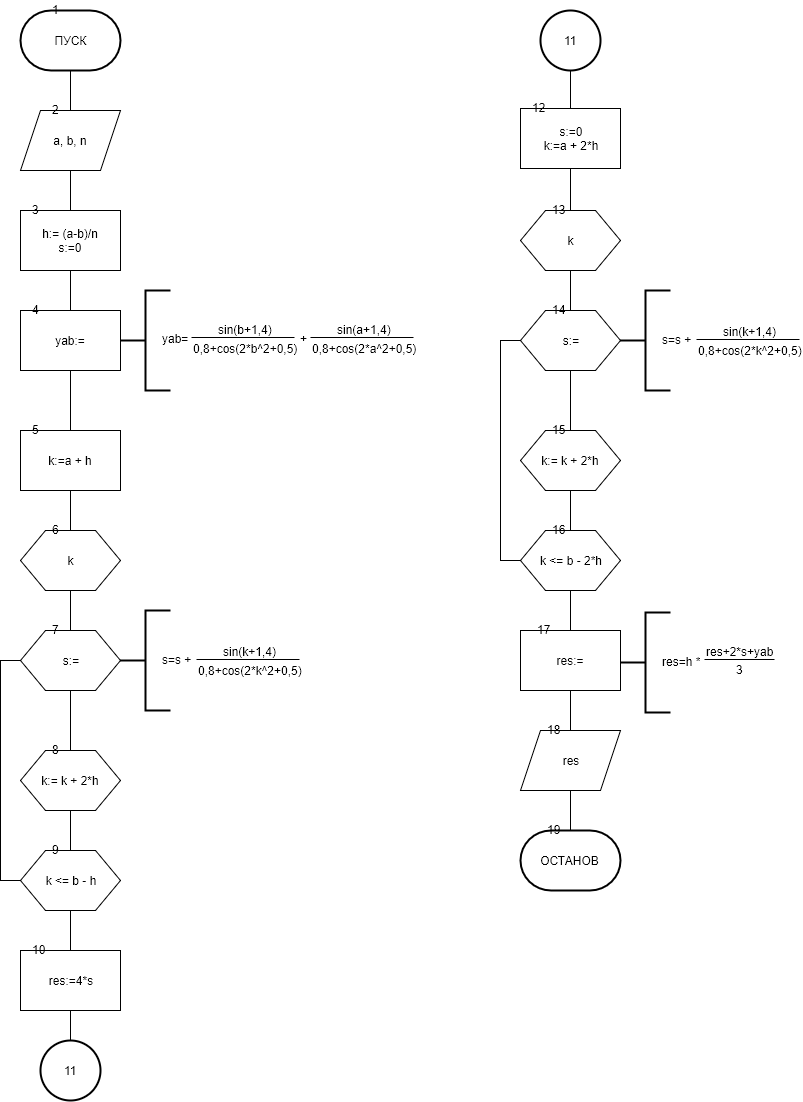


1. Анализ результатов вычисления:

Программа поддерживает до 65535 разбиений интегрируемого интервала. Цикл вычислений проходит от a + h до b - h, значения функции на границах интервала вычисляются отдельно и делятся надвое в соответствии с методом трапеций. Точность нахождения определенного интеграла определяется количеством разбиений N. Чем больше значение N, тем более точным окажется результат.

Задание 4.

1. Написать программу для вычисления определенного интеграла из индивидуального задания методом парабол. Протестировать программу на определенном интеграле, вычисленным в ходе выполнения самостоятельной работы 3.
2. Математическая модель:
3. Блок-схема:



1. Список идентификаторов:

|  |  |  |
| --- | --- | --- |
| Имя | Смысл | Тип |
| a | Левая граница интегрируемого интервала | real |
| b | Правая граница интегрируемого интервала | real |
| n | Количество разбиений интервала | word |
| h | Величина шага | real |
| k | Переменная-счётчик для подсчёта сумм значений элементов с нечётными чётными индексами в первом и втором цикле соответственно | real |
| yab | Сумма значений функции в крайних точках интервала | real |
| s | Сумма значений первого цикла, затем - второго | real |
| res | Результат | real |

1. Код программы:

program Project4;

var a, b, h, s, res, yab, k: real;

n: word;

begin

a:= 0.4;

b:= 1;

writeln('Enter n');

readln(n);

h:=(b-a)/n;

yab:=(sin(a+1.4))/(0.8+cos(2\*a\*a+0.5))+(sin(b+1.4))/(0.8+cos(2\*b\*b+0.5));

k:= a+h;

s:=0;

while k <= b-h do begin

s:= s + (sin(k+1.4))/(0.8+cos(2\*k\*k+0.5));

k:= k + 2\*h;

end;

res:=s\*4;

s:=0;

k:= a+2\*h;

while k <= b-2\*h do begin

s:= s + (sin(k+1.4))/(0.8+cos(2\*k\*k+0.5));

k:= k + 2\*h;

end;

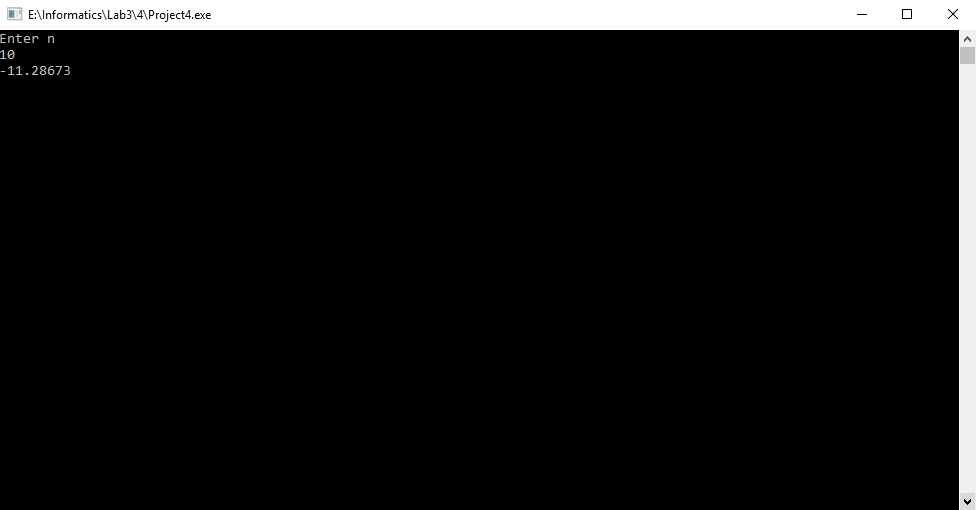
res:= h\*(res+s\*2+yab)/3;

writeln(res:5:5);

readln();

end.

1. Результаты вычислений:



1. Анализ результатов вычисления:

Программа поддерживает до 65535 разбиений интегрируемого интервала. В первом цикле подсчитывается сумма y с нечётными индексами от a + h до b – h. Полученная сумма записывается в переменную s, значение которой затем присваивается переменной res и обнуляется для подсчёта суммы значений второго цикла. Во втором цикле суммируются значения y с чётными индексами от a + 2\*h до b – 2\*h. Значения функции на границах интервала рассчитываются отдельно. Конечное значение рассчитывается согласно методу Симпсона. Точность метода растёт с повышением количества разбиений интервала.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N  Количество разбиений | H  шаг | I  Метод левых частей прямоугольников | I  Метод правых частей прямоугольников | I  Метод трапеций | I  Метод парабол |
| 10 | 0.06 | 0.57301 | 0.80550 | -17.16585 | -11.28673 |
| 100 | 0.006 | 1.19952 | 1.49990 | -0.57437 | -0.11032 |
| 1000 | 0.0006 | 2.02813 | 3.40436 | 1.85074 | 3.69314 |
| 10000 | 0.00006 | 8.08480 | 8.04932 | 8.06706 | 6.11274 |

Вывод:

1. Наиболее точным методом интегрирования для данной функции является метод трапеций, т.к. диапазон значений при различном количестве разбиений интервала в сравнении с остальными методами является наименьшим.
2. Повысить точность любого из методов можно увеличивая количество разбиений интегрируемого интервала. Чем выше количество разбиений, тем меньше погрешность аппроксимации подынтегральной функции.