## Численное интегрирование

Постоянный шаг.

1. Тема: Численное интегрирование с постоянным шагом.
2. Цель: Изучить несколько методов численного интегрирования с постоянным шагом и реализовать их в виде программы.
3. Используемое оборудование: ПК, текстовый редактор Notepad++, браузер.

**Метод левых частей прямоугольника.**

1. Постановка задачи: реализовать с помощью языка JavaScript вычисление интеграла, используя метод левых частей прямоугольника.
2. Код программы (приведены фрагменты, относящиеся только непосредственно к вычислению):

var func = (inp\_func.value).replace(/X(?!p)/g,'x');

var a = eval(inp\_a.value);

var b = eval(inp\_b.value);

var n = +inp\_n.value;

var sum = 0;

var h = (b-a)/n;

var x = a;

var s;

while (x <= b-h) {

sum = sum + eval(func);

x = x + h;

}

s = sum \* h;

1. Результат работы программы:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Подынтегр. выражение | Нижний предел интегрирования | Верхний предел интегрирования | Количество разбиений | Результат |
| Math.exp(Math.sqrt(x)) | 1 | 5 | 100000 | 23.13033070411247 |

Ответ, полученный с помощью онлайн-сервиса по вычислению интегралов:

23.1304634677793

1. Анализ результатов вычисления:

Данный метод не слишком-то эффективен в нашем случае. При довольно большом количестве разбиений точность оставляет желать лучшего.

1. Вывод:

Реализовал вычисление интеграла методом прямоугольника левых частей. Реализация включена в меню в раздел (Численное интегрирование/С постоянным шагом)

**Метод правых частей прямоугольника.**

1. Постановка задачи: реализовать с помощью языка JavaScript вычисление интеграла, используя метод правых частей прямоугольника.
2. Код программы (приведены фрагменты, относящиеся только непосредственно к вычислению):

var func = (inp\_func.value).replace(/X(?!p)/g,'x');

var a = eval(inp\_a.value);

var b = eval(inp\_b.value);

var n = +inp\_n.value;

var sum = 0;

var h = (b-a)/n;

var x = a+h;

var s;

while (x <= b) {

sum = sum + eval(func);

x = x + h;

}

s = sum \* h;

1. Результат работы программы:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Подынтегр. выражение | Нижний предел интегрирования | Верхний предел интегрирования | Количество разбиений | Результат |
| Math.exp(Math.sqrt(x)) | 1 | 5 | 100000 | 23.1305962316 |

Ответ, полученный с помощью онлайн-сервиса по вычислению интегралов:

23.1304634677793

1. Анализ результатов вычисления:

В конкретно нашем случае данный метод показал странный результат, который превзошёл результат, полученный из Интернета. К тому же, при довольно большом количестве разбиений точность оставляет желать лучшего.

1. Вывод:

Реализовал вычисление интеграла методом прямоугольника правых частей. Реализация включена в меню в раздел (Численное интегрирование/С постоянным шагом)

**Метод трапеций.**

1. Постановка задачи: реализовать с помощью языка JavaScript вычисление интеграла, используя метод трапеций.
2. Код программы (приведены фрагменты, относящиеся только непосредственно к вычислению):

var func = (inp\_func.value).replace(/X(?!p)/g,'x');

var a = eval(inp\_a.value);

var b = eval(inp\_b.value);

var n = +inp\_n.value;

var sum = 0;

var h = (b-a)/n;

var x = a+h;

var s,fa,fb;

while (x <= b-h) {

sum = sum + eval(func);

x = x + h;

}

fa = eval(func.replace(/x(?!p)/g,'a'));

fb = eval(func.replace(/x(?!p)/g,'b'));

s = h\*((fa+fb)/2+sum);

1. Результат работы программы:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Подынтегр. выражение | Нижний предел интегрирования | Верхний предел интегрирования | Количество разбиений | Результат |
| Math.exp(Math.sqrt(x)) | 1 | 5 | 100000 | 23.13046346778813 |

Ответ, полученный с помощью онлайн-сервиса по вычислению интегралов:

23.1304634677793

1. Анализ результатов вычисления:

Если метод парабол и превосходит в точности метод трапеций, то в нашем случае, ненамного.

1. Вывод:

Реализовал вычисление интеграла методом парабол. Реализация включена в меню в раздел (Численное интегрирование/С постоянным шагом)

**Метод парабол.**

1. Постановка задачи: реализовать с помощью языка JavaScript вычисление интеграла, используя метод парабол.
2. Код программы (приведены фрагменты, относящиеся только непосредственно к вычислению):

var func = (inp\_func.value).replace(/X(?!p)/g,'x');

var a = eval(inp\_a.value);

var b = eval(inp\_b.value);

var n = +inp\_n.value;

var sum = 0, sum2 = 0;

var h = (b-a)/n;

var x = a+h;

var s,fa,fb;

while (x <= b-h) {

sum = sum + eval(func);

x = x + 2 \* h;

}

x = a+2\*h;

while (x <= b-2\*h) {

sum2 = sum2 + eval(func);

x = x + 2 \* h;

}

fa = eval(func.replace(/x(?!p)/g,'a'));

fb = eval(func.replace(/x(?!p)/g,'b'));

s = h/3\*(fa+fb+4\*sum+2\*sum2);

1. Результат работы программы:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Подынтегр. выражение | Нижний предел интегрирования | Верхний предел интегрирования | Количество разбиений | Результат |
| Math.exp(Math.sqrt(x)) | 1 | 5 | 100000 | 23.130463467856234 |

Ответ, полученный с помощью онлайн-сервиса по вычислению интегралов:

23.1304634677793

1. Анализ результатов вычисления:

По сравнению с предыдущими двумя методами, текущий показывает самый лучший результат в нашем случае.

1. Вывод:

Реализовал вычисление интеграла методом трапеций. Реализация включена в меню в раздел (Численное интегрирование/С постоянным шагом)

Переменный шаг.

1. Тема: Численное интегрирование с переменным шагом.
2. Цель: Изучить несколько методов численного интегрирования с переменным шагом и реализовать их в виде программы.
3. Используемое оборудование: ПК, текстовый редактор Notepad++, браузер.

**Метод левых частей прямоугольника.**

1. Постановка задачи: реализовать с помощью языка JavaScript вычисление интеграла, используя метод левых частей прямоугольника.
2. Код программы (приведены фрагменты, относящиеся только непосредственно к вычислению):

var func = (inp\_func.value).replace(/X(?!p)/g,'x');

var a = eval(inp\_a.value);

var b = eval(inp\_b.value);

var e = eval(inp\_e.value);

var pr = (inp\_pr.value).replace(/x(?!p)/g,'i');

var sum = 0;

var x = a;

var s, r, eps;

var array = [];

var i=a;

var I2n=1;

var In=0;

while(i<b){

array.push(eval(pr));

i+=1

}

r = (((b-a)\*(b-a))/2)\*Math.abs.apply(null,array);

h = (b-a)/r;

do{

x=a;

while (x <= b-h) {

sum += eval(func);

x += h;

}

I2n = (sum \* h);

R=Math.abs(I2n-In);

In=I2n;

sum=0;

h=h/2;

} while(R > e)

1. Результат работы программы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Подынтегр. выражение | Нижний предел интегрирования | Верхний предел интегрирования | Точность | Первая производная | Результат |
| Math.exp(Math.sqrt(x)) | 1 | 5 | 0.00001 | Math.exp(Math.sqrt(x))/(2\*Math.sqrt(x)) | 23.130439885437443 |

Ответ, полученный с помощью онлайн-сервиса по вычислению интегралов:

23.1304634677793

1. Анализ результатов вычисления:

Требуемая точность достигнута.

1. Вывод:

Реализовал вычисление интеграла методом прямоугольника левых частей. Реализация включена в меню в раздел (Численное интегрирование/С переменным шагом)

**Метод правых частей прямоугольника.**

1. Постановка задачи: реализовать с помощью языка JavaScript вычисление интеграла, используя метод правых частей прямоугольника.
2. Код программы (приведены фрагменты, относящиеся только непосредственно к вычислению):

var func = (inp\_func.value).replace(/X(?!p)/g,'x');

var a = eval(inp\_a.value);

var b = eval(inp\_b.value);

var e = eval(inp\_e.value);

var pr = (inp\_pr.value).replace(/x(?!p)/g,'i');

var sum = 0;

var x = a;

var s, r, eps;

var array = [];

var i=a;

var I2n=1;

var In=0;

while(i<b){

array.push(eval(pr));

i+=1

}

r = (((b-a)\*(b-a))/2)\*Math.abs.apply(null,array);

h = (b-a)/r;

do{

x=a+h;

while (x <= b) {

sum += eval(func);

x += h;

}

I2n = (sum \* h);

R=Math.abs(I2n-In);

In=I2n;

sum=0;

h=h/2;

} while(R > e)

1. Результат работы программы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Подынтегр. выражение | Нижний предел интегрирования | Верхний предел интегрирования | Точность | Первая производная | Результат |
| Math.exp(Math.sqrt(x)) | 1 | 5 | 0.00001 | Math.exp(Math.sqrt(x))/(2\*Math.sqrt(x)) | 23.130458516811192 |

Ответ, полученный с помощью онлайн-сервиса по вычислению интегралов:

23.1304634677793

1. Анализ результатов вычисления:

Требуемая точность достигнута.

1. Вывод:

Реализовал вычисление интеграла методом прямоугольника правых частей. Реализация включена в меню в раздел (Численное интегрирование/С переменным шагом)

**Метод трапеции.**

1. Постановка задачи: реализовать с помощью языка JavaScript вычисление интеграла, используя метод трапеции.
2. Код программы (приведены фрагменты, относящиеся только непосредственно к вычислению):

var func = (inp\_func.value).replace(/x(?!p)/g,'x');

var a = eval(inp\_a.value);

var b = eval(inp\_b.value);

var e = eval(inp\_e.value);

var pr = (inp\_pr.value).replace(/x(?!p)/g,'i');

var sum = 0;

var In=1;

var I2n=0;

var fa,fb,x;

var i = a;

var array = [];

while(i<b){

array.push(eval(pr));

i+=1;

}

var n = (((b-a)\*(b-a)\*(b-a))/12)\*Math.abs(Math.max.apply(null,array));

var h = (b-a)/n;

do{

x = a+h;

sum = 0;

while (x <= (b-h)) {

sum += eval(func);

x += h;

}

fa = Math.abs(eval(func.replace(/x(?!p)/g,'a')));

fb = Math.abs(eval(func.replace(/x(?!p)/g,'b')));

I2n = h\*((fa+fb)/2+sum);

R = Math.abs(I2n-In);

In=I2n;

h=h/2;

}while(R > e/2)

1. Результат работы программы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Подынтегр. выражение | Нижний предел интегрирования | Верхний предел интегрирования | Точность | Вторая производная | Результат |
| Math.exp(Math.sqrt(x)) | 1 | 5 | 0.00001 | (Math.exp(Math.sqrt(x))\*(Math.sqrt(x)-1))/(4\*Math.pow(x,3/2)) | 23.1304534677793 |

Ответ, полученный с помощью онлайн-сервиса по вычислению интегралов:

23.1304634677793

1. Анализ результатов вычисления:

Требуемая точность достигнута.

1. Вывод:

Реализовал вычисление интеграла методом трапеции. Реализация включена в меню в раздел (Численное интегрирование/С переменным шагом)

**Метод парабол.**

1. Постановка задачи: реализовать с помощью языка JavaScript вычисление интеграла, используя метод парабол.
2. Код программы (приведены фрагменты, относящиеся только непосредственно к вычислению):
3. Результат работы программы:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Подынтегр. выражение | Нижний предел интегрирования | Верхний предел интегрирования | Точность | Четвёртая производная | Результат |
| Math.exp(Math.sqrt(x)) | 1 | 5 | 0.00001 | (Math.exp(Math.sqrt(x))\*(Math.pow(x,3/2)-6\*x+15\*Math.sqrt(x)-15))/(16\*Math.pow(x,7/2)) | 23.1304534677793 |

Ответ, полученный с помощью онлайн-сервиса по вычислению интегралов:

23.1304634677793

1. Анализ результатов вычисления:

Требуемая точность достигнута.

1. Вывод:

Реализовал вычисление интеграла методом трапеции. Реализация включена в меню в раздел (Численное интегрирование/С переменным шагом)