Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение высшего образования  
«Петрозаводский государственный университет»  
Физико-технический институт

Сафонов Владислав Денисович

ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ №1

«Определенный интеграл методом Симпсона»

отчет обучающегося 1 курса очной формы обучения  
по направлению подготовки 09.03.01 "ИВТ"

Петрозаводск 2023 г.

**Цель:** составить программу на языке Программирования высокого уровня, вычисляющую определенный интеграл

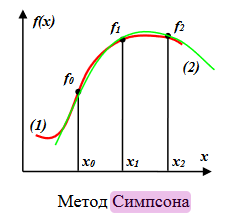
f(x) = |ln(x)| от (1/e) до e

методом Симпсона с точностью 0.00001. С помощью любой экспертной системы найти решение этого интеграла и сравнить полученные результаты.

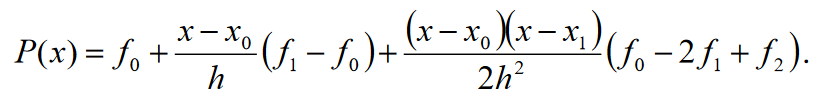
**Сведения о численном методе:**

Подинтегральная функция f(x) заменяется интерполяционным полиномом второй степени P(x) –параболой, проходящей через три узла, например, как показано на рисунке ((1) – функция, (2) – полином).

Рассмотрим два шага интегрирования (h = const = xi+1 – xi), то есть три узла x0, x1, x2, через которые проведем параболу,

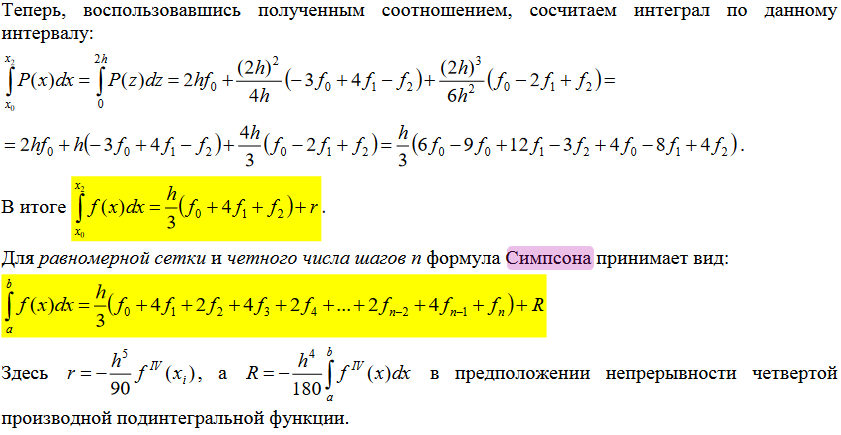


воспользовавшись уравнением Ньютона:



Пусть z = x – x0, тогда:





**Словесно-формульный алгоритм:**

1. Ввести нижний (down) и верхний (up) пределы интегрирования и требуемую точность (epsilon);

2. Ввести начальное число разбиений n (n%2=0);

3. Задать подынтегральную функцию:f = |ln(x)|;

4. Повторять

4.1. Вычисление шага интегрирования h=(up-down)/n;

4.2. Задаем начальное значение x=down

4.3. S=0;

4.4. Считаем значение f (down), f(up);

4.5. Прибавляем f(down), f(up) к S ;

4.6. Повторять для i от 1 до n-1

4.6.1. x+=h (сдвиг на шаг);

4.6.2 Если i четное: вхождение значения с кф \*2   
 (S += f(x) \*2)

4.6.3 Иначе: вхождение значения с кф \*4 (f(x) \*4)

4.7. Расчет значения интеграла: prevSum = S\*2\*h/6;

4.8. Увеличение числа разбиений: n=n+2;

4.9. Вычисление нового шага интегрирования h=(b-a)/n

4.10. S=0;

4.11. Повторять для i от 1 до n-1

4.11.1. Расчет суммы значений функции в новых узлах

4.12. Расчет нового значения интеграла: currentSum = S\*2\*h/6;

4.13. Увеличение числа разбиений: n=n+2;

пока модуль(prevSum - currentSum)>epsilon;

5. Вывод значения интеграла Sum2 и окончательного числа разбиений n.

**Листинг программы:**

double simpsonMethod(double down, double up, int n){  
  
 double h = (double) ((up - down) / n);  
 double x = down;  
 double S = 0.0;

S += (function(down) + function(up));  
 for (int i = 1; i <= n-1; i++){  
 x += h;  
 if ((i % 2) == 0){  
 S += 2\*function(x);  
 }  
 else {  
 S += 4\*function(x);  
 }  
}

return S\*2\*h/6;

}

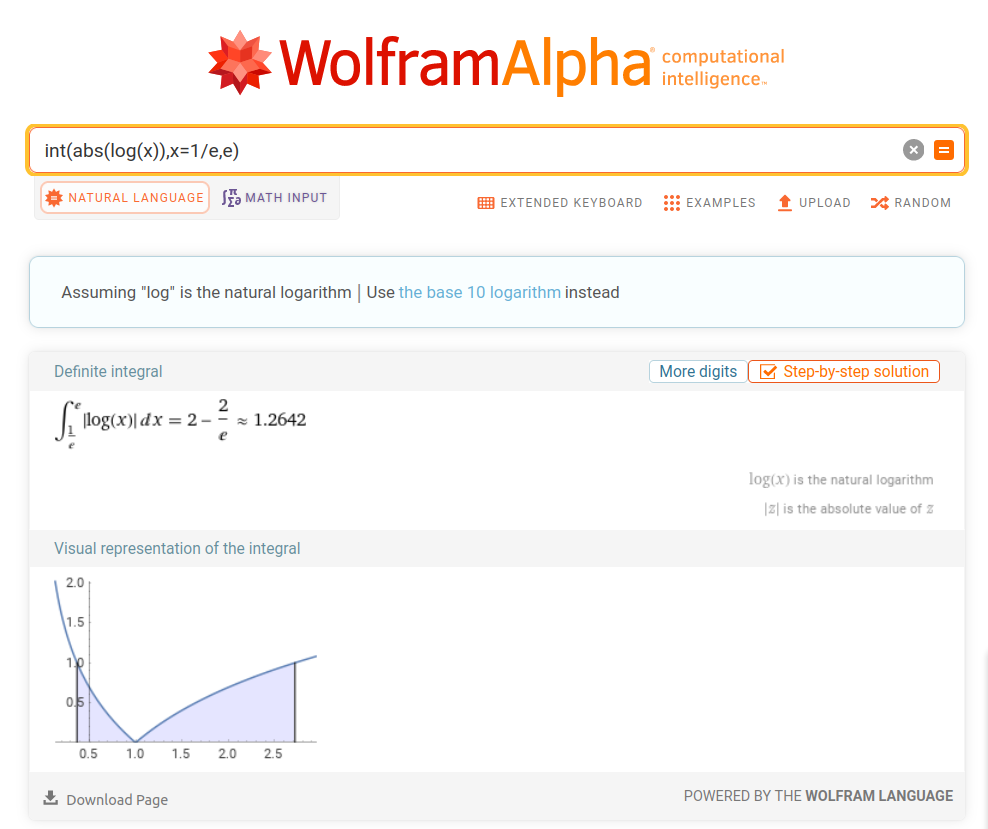
int main(){  
 const double exp = 2.71828;  
 double prevSum = simpsonMethod(1/exp, exp, n), currentSum;  
 for (int i = 1;;i++){  
 n += i;  
 currentSum = simpsonMethod(1/exp, exp, n);

if (fabs(prevSum-currentSum) > epsilon){  
 prevSum = currentSum;  
 }

else {  
 printf("Result: %lf, n = %d.\n", currentSum, n); break;  
 }  
 }

}

**Результаты расчета в экспертной системе:**



**Результаты работы программы:**



**Вывод:**

С помощью ЯП Си я написал программу, которая считает данный интеграл с точностью 0,00001 методом Симпсона.

Данные которые получились у меня отличаются от данных полученные в WolframAlpha (в интернете являются более точными). Это связано с заданной точностью, если взять большее количество разбиений, то результаты будут совпадать.