|  |  |  |
| --- | --- | --- |
| ё | | |
| Федеральное государственное бюджетное  образовательное учреждение высшего образования «Новосибирский государственный технический университет» | | |
|  | | |
| Кафедра прикладной математики | | |
| Лабораторная работа № 4 | | |
| по дисциплине «Программирование вычислений» | | |
| **Численное интегрирование** | | |
|  | | |
|  | Бригада 5 | Гриневич Юлия |
| Группа ПМ-21 | Егупов Иван |
| Вариант 2, 5.4, 5.6 | Порсин Данил |
|  |  |
|  |  |
|  |  |
| Преподаватели | Рояк Светлана Хаимовна |
|  |  |
| Новосибирск, 2024 | | |



1. **Цель работы**

Изучение методов численного интегрирования, оценки порядка точности, оценки погрешности по правилу Рунге, уточнения значений по Ричардсону.

1. **Математическая модель**

Входные данные: интегрируемая на отрезке интегрирования функция f(x), пределы интегрирования [a,b] этой функции.

Выходные данные: результат вычисления интеграла заданной функции f(x).

* Метод трапеций

Усложненная квадратурная формула для равномерной сетки из N отрезков:

где h – шаг сетки.

Порядок точности – 1

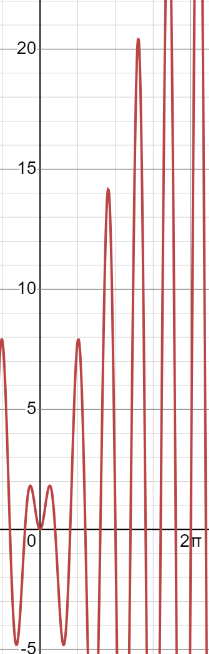
Порядок метода (порядок аппроксимации) - 2

* Метод Ньютона – Котеса

Усложненная квадратурная формула для равномерной сетки из N отрезков и с n точками на каждом из отрезков (считая начало и конец отрезка):

Порядок точности – (n-1)

Порядок метода (порядок аппроксимации): для 4 узлов – 4, для 6 узлов – 5

Выбранная осциллирующая функция 5x\*sin(5x):  


1. **Текст программы**

program main

implicit none

common/data/a,b,seg

real a,b

real nc6

real grid

dimension grid(2\*\*28)

integer seg

call input

call make\_grid(grid(1))

call trapez(grid,seg+1)

print\*,'---------------'

call nc4(grid,seg+1)

print\*,'---------------'

print\*,nc6(grid,seg+1)

pause

end

!-------------------------------------------------

subroutine input

implicit none

common/data/a,b,seg

real a,b

integer seg

print\*,'enter begin and end of the integration interval'

read\*,a,b

print\*,'enter the number of segments'

read\*,seg

end

!-------------------------------------------------

subroutine make\_grid(grid)

implicit none

common/data/a,b,seg

real a,b

real h

real grid

dimension grid(2\*\*28)

integer seg

integer i

h=(b-a)/seg

do i=0,seg

grid(i+1)=a+i\*h

end do

end

!------------------------------------------------

subroutine trapez(grid,len)

implicit none

common/data/a,b,seg

real a,b,h,res,f

integer len,seg

real grid

dimension grid(len)

integer i

h = grid(2)-grid(1)

res = h\*(f(grid(1))+f(grid(len)))/2.0

do i = 2,len-1

res = res + h\*f(grid(i))

end do

print\*,res

end

!-------------------------------------------------

real function nc6(grid,len)

implicit none

common/data/a,b,seg

real a,b,f,h

integer seg,len

real grid

dimension grid(len)

real sum\_i,sum\_j

integer coefs

dimension coefs(6)

integer i,j

h=(grid(2)-grid(1))/5

coefs(1)=19

coefs(2)=75

coefs(3)=50

coefs(4)=50

coefs(5)=75

coefs(6)=19

sum\_i=0

sum\_j=0

do i=1,seg

do j=0,5

sum\_j=sum\_j+coefs(j+1)\*f(grid(i)+j\*h)

end do

sum\_i=sum\_i+sum\_j

sum\_j=0

end do

nc6=5\*h\*sum\_i/288

end

!-------------------------------------------------

subroutine nc4(grid,len)

implicit none

common/data/a,b,seg

real a,b,f,h,res

integer seg,len

real grid

dimension grid(len)

real sum\_i,sum\_j

integer coefs

dimension coefs(4)

integer i,j

h=(grid(2)-grid(1))/3

coefs(1)=1

coefs(2)=3

coefs(3)=3

coefs(4)=1

sum\_i=0

sum\_j=0

do i=1,seg

do j=0,3

sum\_j=sum\_j+coefs(j+1)\*f(grid(i)+j\*h)

end do

sum\_i=sum\_i+sum\_j

sum\_j=0

end do

res=3\*h\*sum\_i/8

print\*,res

end

!---------------------------------------------------

real function f(x)

implicit none

real x

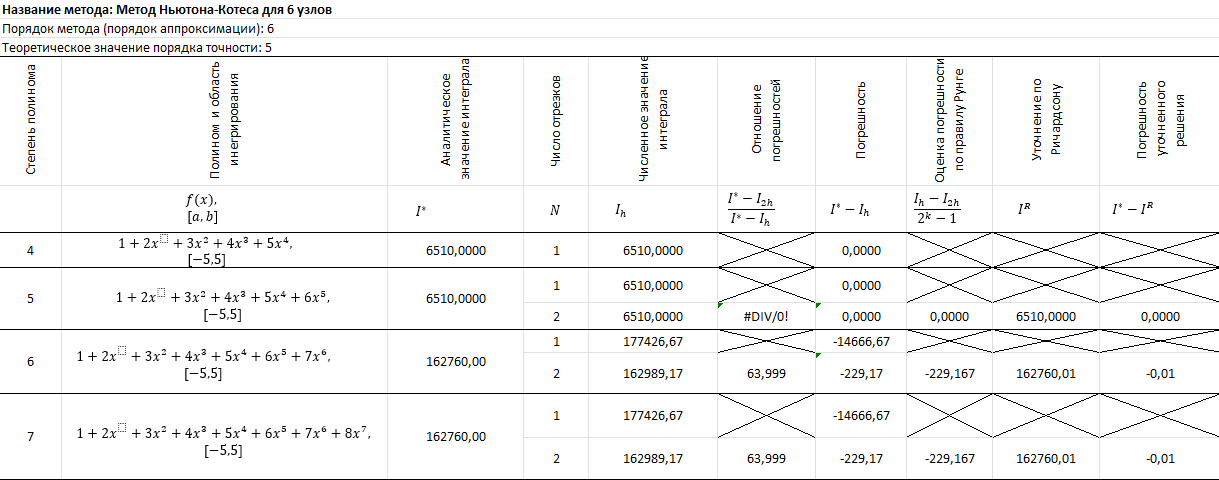
f = 0.25\*x\*\*7

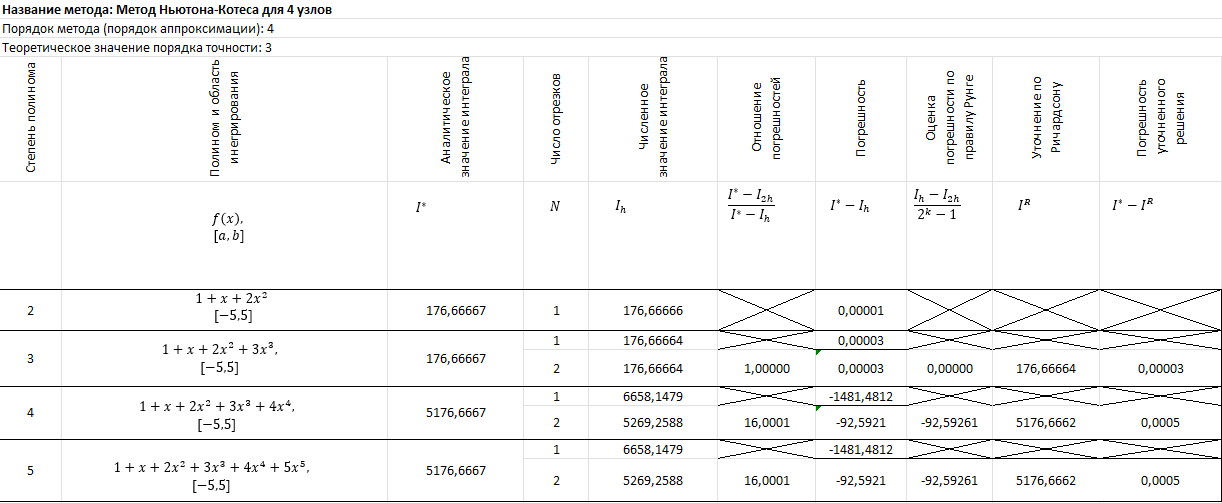
end

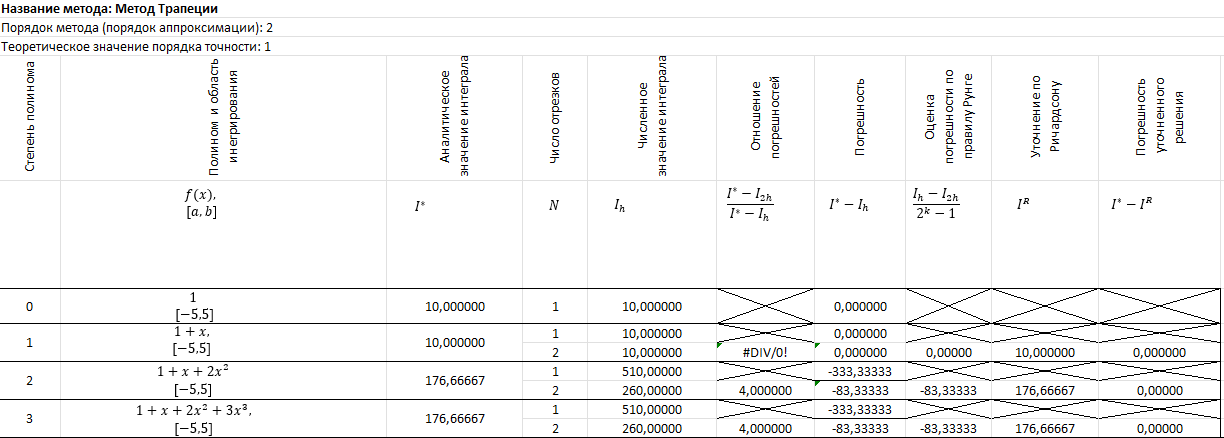
!--------------------------------------------------

1. **Таблицы**

**Верификация программы:**

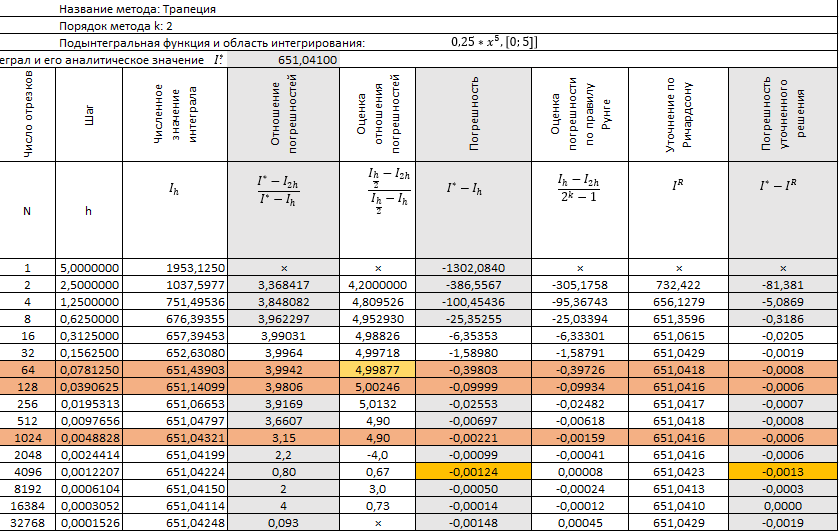


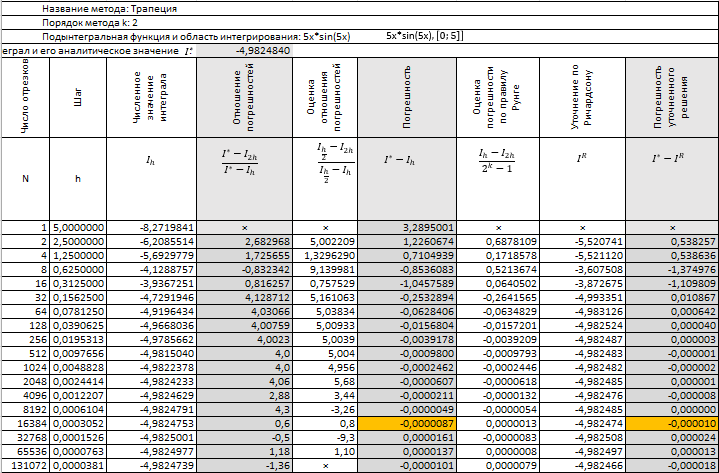


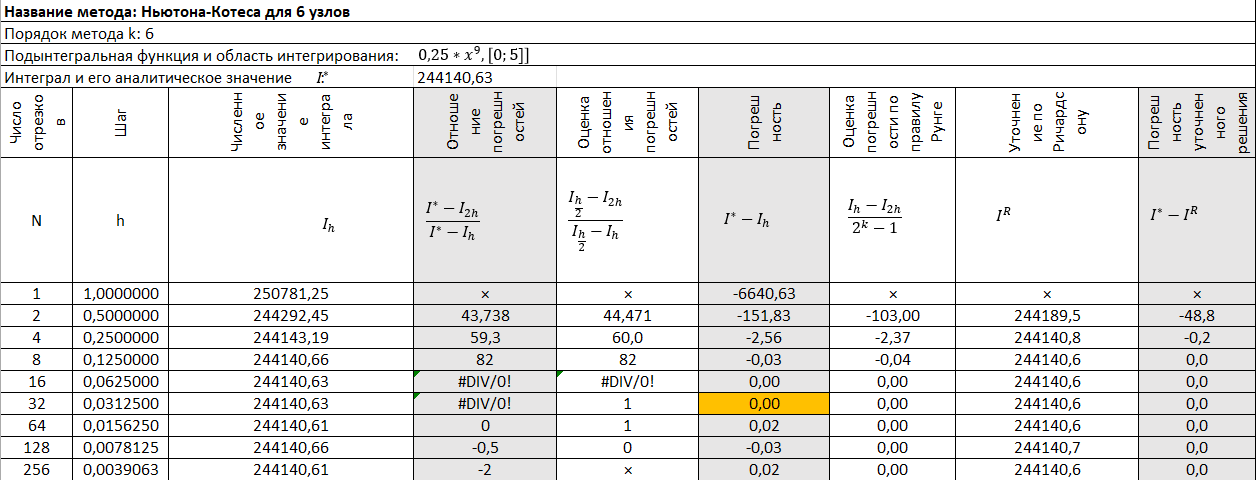


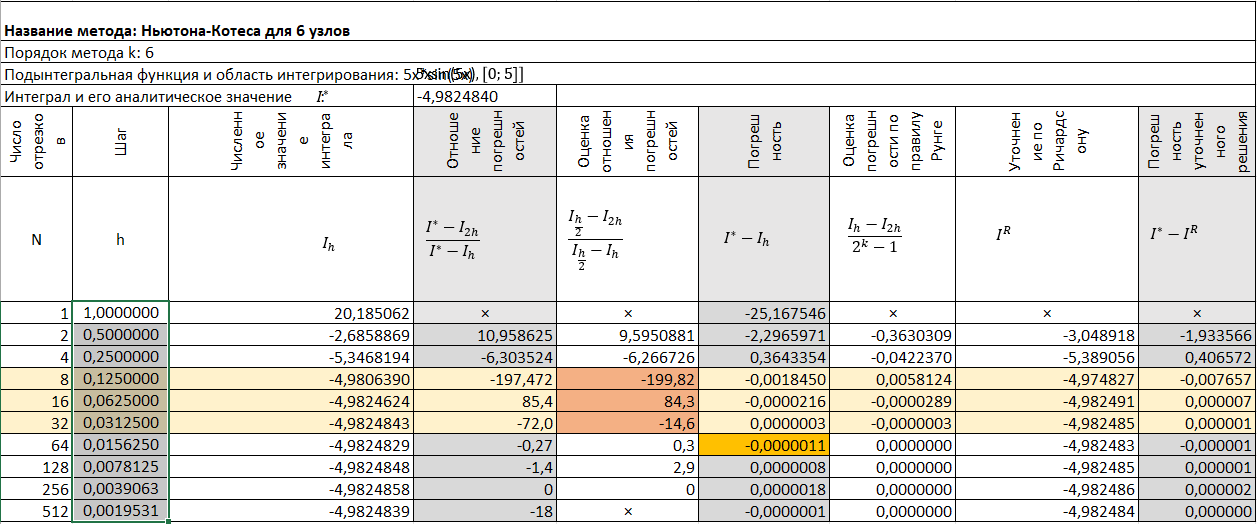
**Вывод**: точность вычисления каждого из методов соответствовала теоретическому порядку точности. То есть каждый метод правильно исчислял интеграл от полинома с максимальной степенью, равной порядку точности метода.

**Изучение порядка аппроксимации:**

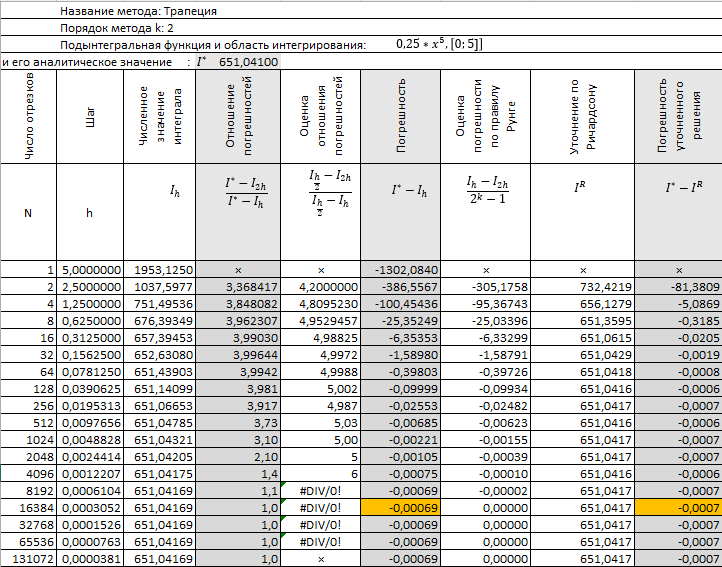


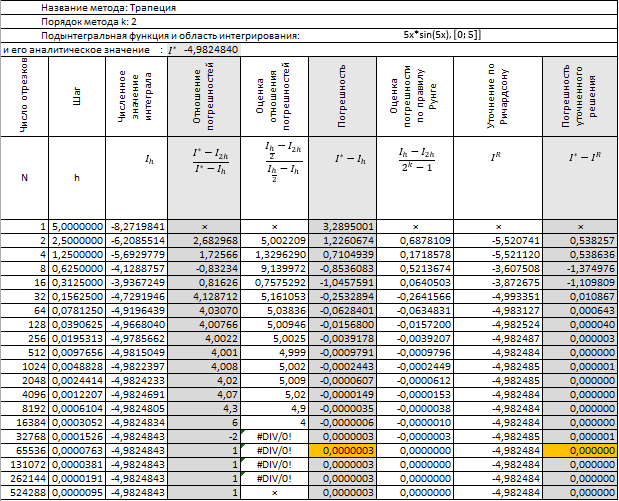


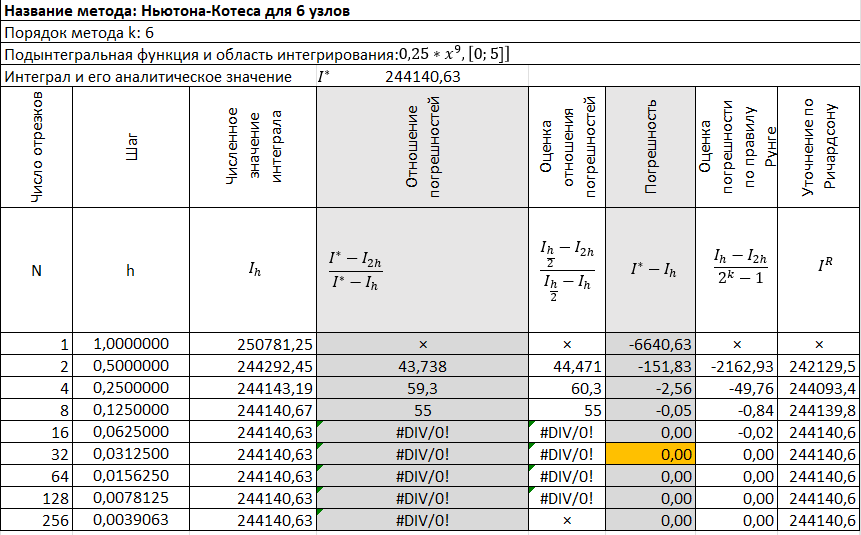


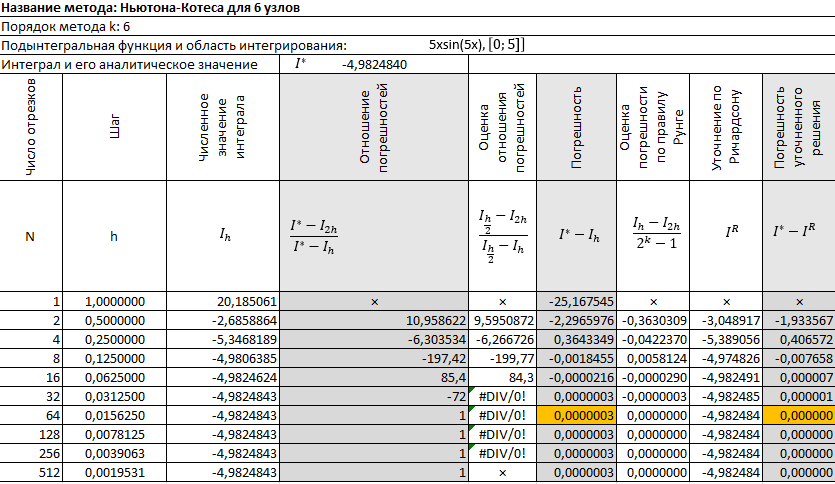


**Таблицы с использованием переменной накопления real\*8**



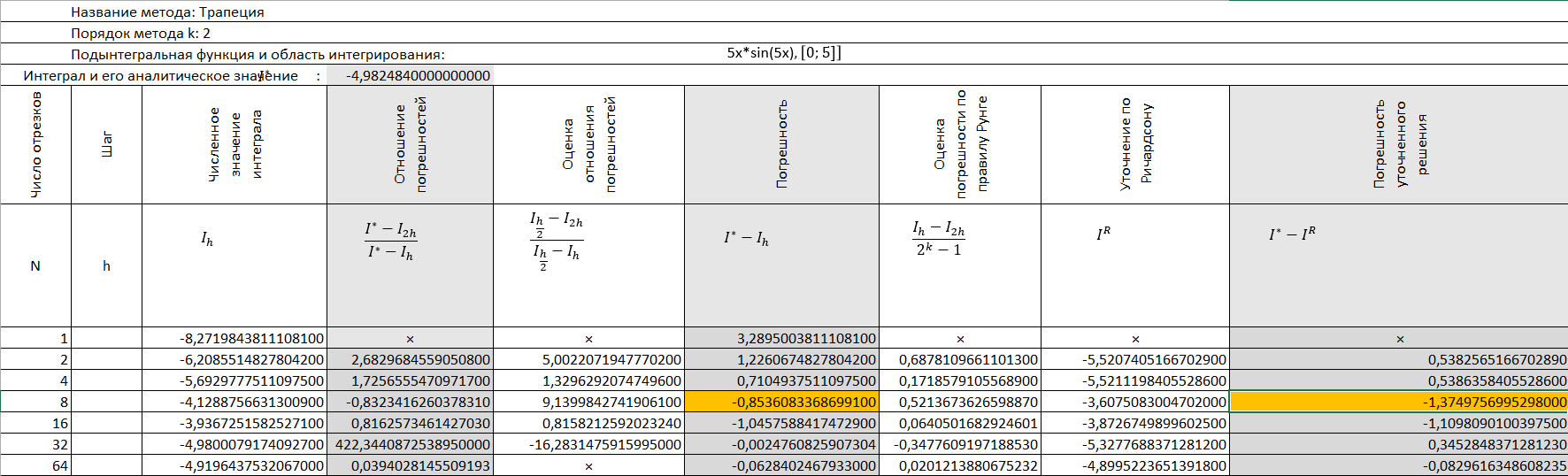


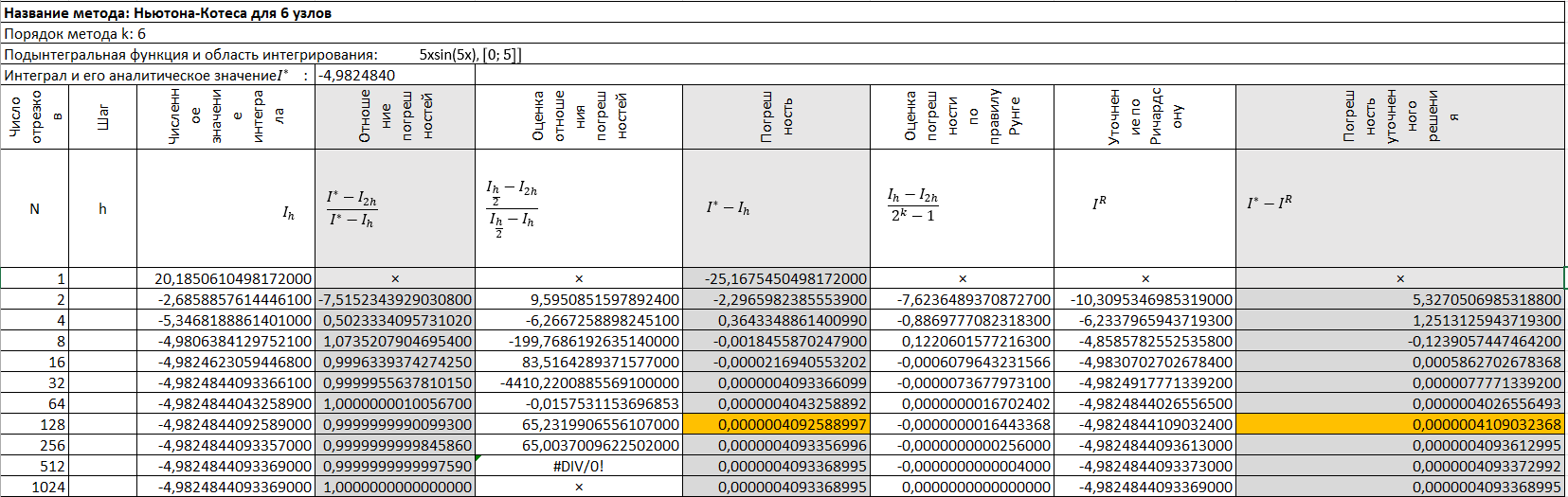




**Вывод:** Использую переменную накопления с большей длиной слова, была получена меньшая погрешность при вычислениях. Погрешность уточненного решения также оказалась меньше. Используя переменную накопления двойной точности, мы можем доверять 8 значащей цифре, так как погрешность вычислений будет скапливаться в последующих разрядах, что даёт возможность получить более точное значение при вычислении

**Таблицы для осциллирующей функции с использованием в коде программы вычислений с двойной точностью:**





**Вывод:** Длина слова, т.е. количество битов в ячейке памяти, может влиять на точность вычислений в численном интегрировании. Чем длиннее слово, тем больше битов доступно для представления чисел с плавающей точкой. Это может позволить использовать больше разрядов для хранения значений, что может привести к более точным результатам вычислений.