Министерство образования и науки Российской Федерации

Новосибирский государственный технический университет

Кафедра прикладной математики

Программирование вычислений

Лабораторная работа №4

Факультет: ПМИ

Группа: ПМ-01

Студенты: Александров М.Е.

Жигалов П.С.

Преподаватели: Вагин Д.В.

Рояк С.Е.

Вариант: 11

Новосибирск

2012

**Цели работы**

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

**Задание**

1. Запрограммировать указанный метод, используя его усложненную квадратурную формулу.

2. Провести практическое исследование порядка точности указанного метода для многочленов степени m .

3. Провести исследование порядка малости остаточного члена для нескольких функций. Среди них обязательно должна быть сильно осциллирующая (колеблющаяся) функция (4-5 колебаний на отрезке интегрирования).

4. Исследовать влияние длины слова при реализации численного интегрирования, для чего сравнить реализации методов с одинарной и двойной точностью на нескольких функциях. Замечание: При задании констант двойной точности на языке Фортран необходимо задавать их в виде PI=3,14159265358979D00, а также все переменные объявлять как REAL\*8.

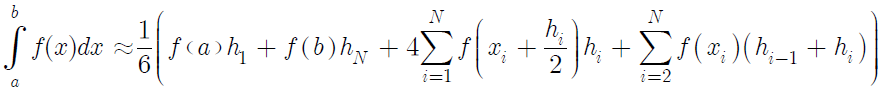
5. Провести исследование на неравномерных сетках. Выяснить, в каких промежутках сгущение сетки дает наилучшие результаты: в окрестности экстремумов, на местах осцилляции или в местах постоянства. Результаты сравнить по точности и по вычислительным затратам, например, при различном разбиении отрезка, но с равным количеством узлов, получены существенно разные по точности результаты; или при практически равной точности необходимо существенно разное количество узлов для различных разбиений. Выяснить, в каких промежутках (ближе к точкам экстремума или в местах быстрого роста функции) требуется увеличение числа отрезков интегрирования.

Вариант 11. Методы Симпсона и Гаусса-3.

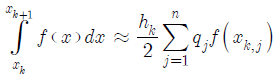
**Анализ задачи**

Основные формулы, использованные при выполнении лабораторной работы:

1. Усложнённая квадратурная формула Симпсона:

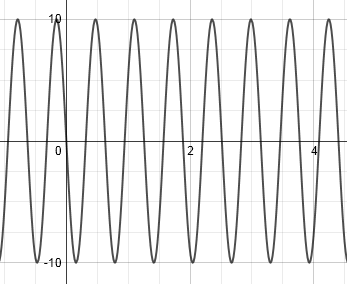
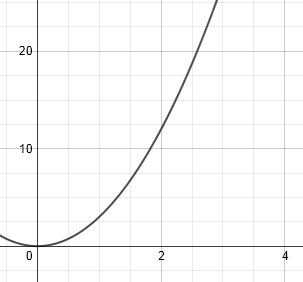


1. Усложнённая квадратурная формула Гаусса:

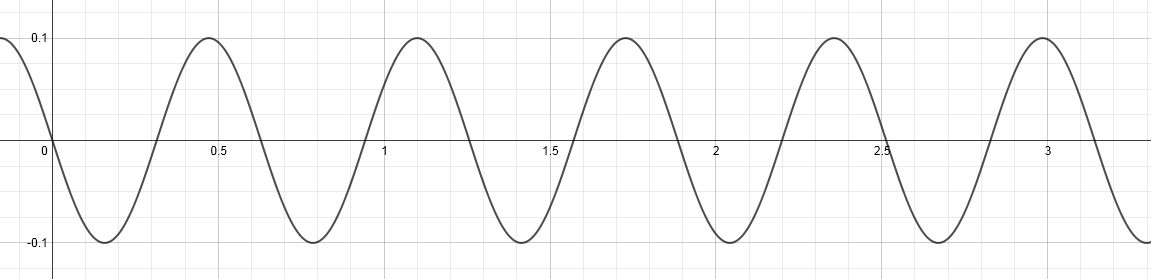
, где 

Графики интегрируемых функций:

Сильно осциллирующая функция Полиномиальная функция



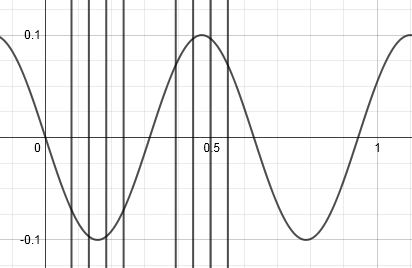
Для построения неравномерной сетки рассмотрим сильно осциллирующую на промежутке [0,] фун-кцию . Ее график выглядит следующим образом:



Построим такое разбиение заданного отрезка, чтобы количество узлов сетки в окрестности экстремумов этой функции было больше, чем на промежутках возрастания и убывания.

Экстремумы этой функции будут в точках , где . Тогда за окрестность экстремума можно взять промежуток .

Разобъем эту окрестность на три равных отрезка, длина каждого будет составлять . Получившееся разбиение будет выглядеть так:



Таким образом, искомое разбиение построено.

Разбиение отрезка такое, чтобы количество узлов сетки на промежутках возрастания и убывания этой функции было больше, чем в окрестности экстремумов строится аналогичным образом, только разбивать будем не окрестность экстремума, а промежуток между окрестностями соседних экстремумов.

**Текст программы**

!Вспомогательная функция, вычисляющая значение осцилирующей функции

real\*8 function f(x) !4

implicit none

real\*8 x !4

f=-10D00\*dsin(10D00\*x) !8

!f=-10.\*sin(10.\*x) !4

end

!Вспомогательная функция, вычисляющая значение полиномиальной функции

real\*8 function g(x,n) !4

implicit none

real\*8 x !4

integer n

g=(n+1)\*x\*\*n

end

!Функия реализующая метод Симпсона для многочлена степени n

real\*8 function simppoly(n,nn,a,b) !8

implicit none

real\*8 sum1,sum2,a,b !8

real\*8 h,g !4

integer n,nn,i

h=(b-a)/nn

sum2=0.0D00

sum1=0.0D00

do i=2,nn

sum2=sum2+g((a+(i-1.0)\*h),n) !8

!sum2=sum2+dble(g((real(a)+(i-1)\*h),n)) !4

enddo

do i=1,nn

sum1=sum1+g((a+(i-1.0)\*h+h/2.0D00),n) !8

!sum1=sum1+dble(g((real(a)+(i-1)\*h+h/2.0),n)) !4

enddo

simppoly=h\*(g(a,n)+g(b,n)+4.0D00\*sum1+2.0D00\*sum2)/6.0D00 !8

!simppoly=dble(g(real(a),n)+g(real(b),n))+4.0D00\*sum1+2.0D00\*sum2 !4

!simppoly=dble(h)\*simppoly/6.0D00 !4

end

!Функия реализующая метод Симпсона для осцилирующей функции

real\*8 function simposcil(nn,a,b) !8

implicit none

real\*8 sum1,sum2,a,b !8

real\*8 h,f !4

integer nn,i

h=(b-a)/nn

sum1=0.0D00

sum2=0.0D00

do i=2,nn

sum2=sum2+f(a+(i-1.0)\*h) !8

!sum2=sum2+dble(f(real(a)+(i-1)\*h)) !4

enddo

do i=1,nn

sum1=sum1+f(a+(i-1.0)\*h+h/2.0D00) !8

!sum1=sum1+dble(f(real(a)+(i-1)\*h+h/2.0)) !4

enddo

simposcil=h\*(f(a)+f(b)+4.0D00\*sum1+2.0D00\*sum2)/6.0D00 !8

!simposcil=dble(f(real(a))+f(real(b)))+4.0D00\*sum1+2.0D00\*sum2 !4

!simposcil=dble(h)\*simposcil/6.0D00 !4

return

end

!Функия реализующая метод Сипсона для осцилирующей функции с неравномерной сеткой

real\*8 function simposcil2()

implicit none

real\*8 sum1,sum2

real\*8 h,f

integer i

real\*8 w,pi

dimension w(42),h(41)

pi=3.14159265358979D00

w(1)=0

w(2)=pi/40.0D0

w(3)=pi/40.0D0+1.0D0\*pi/60.0D0

w(4)=pi/40.0D0+2.0D0\*pi/60.0D0

w(5)=3.0D0\*pi/40.0D0

w(6)=5.0D0\*pi/40.0D0

w(7)=5.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(8)=5.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(9)=7.0D0\*pi/40.0D0

w(10)=9.0D0\*pi/40.0D0

w(11)=9.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(12)=9.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(13)=11.0D0\*pi/40.0D0

w(14)=13.0D0\*pi/40.0D0

w(15)=13.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(16)=13.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(17)=15.0D0\*pi/40.0D0

w(18)=17.0D0\*pi/40.0D0

w(19)=17.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(20)=17.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(21)=19.0D0\*pi/40.0D0

w(22)=21.0D0\*pi/40.0D0

w(23)=21.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(24)=21.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(25)=23.0D0\*pi/40.0D0

w(26)=25.0D0\*pi/40.0D0

w(27)=25.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(28)=25.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(29)=27.0D0\*pi/40.0D0

w(30)=29.0D0\*pi/40.0D0

w(31)=29.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(32)=29.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(33)=31.0D0\*pi/40.0D0

w(34)=33.0D0\*pi/40.0D0

w(35)=33.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(36)=33.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(37)=35.0D0\*pi/40.0D0

w(38)=37.0D0\*pi/40.0D0

w(39)=37.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(40)=37.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(41)=39.0D0\*pi/40.0D0

w(42)=40.0D0\*pi/40.0D0

do i=1,41

h(i)=w(i+1)-w(i)

enddo

sum1=0.0D00

sum2=0.0D00

do i=2,41

sum2=sum2+f(w(i))\*(h(i-1)+h(i))

enddo

do i=1,41

sum1=sum1+f(w(i)+h(i)/2.0D0)\*h(i)

enddo

simposcil2=(f(w(1))\*h(1)+f(w(42))\*h(41)+4.0D0\*sum1+sum2)/6.0D0

return

end

!Функия реализующая метод Сипсона для осцилирующей функции с неравномерной сеткой

real\*8 function simposcil3()

implicit none

real\*8 sum1,sum2

real\*8 h,f

integer i

real\*8 w,pi

dimension w(42),h(41)

pi=3.14159265358979D00

w(1)=0

w(2)=1.0D0\*pi/60.0D0

w(3)=1.0D0\*pi/40.0D0

w(4)=3.0D0\*pi/40.0D0

w(5)=3.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(6)=3.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(7)=5.0D0\*pi/40.0D0

w(8)=7.0D0\*pi/40.0D0

w(9)=7.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(10)=7.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(11)=9.0D0\*pi/40.0D0

w(12)=11.0D0\*pi/40.0D0

w(13)=11.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(14)=11.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(15)=13.0D0\*pi/40.0D0

w(16)=15.0D0\*pi/40.0D0

w(17)=15.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(18)=15.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(19)=17.0D0\*pi/40.0D0

w(20)=19.0D0\*pi/40.0D0

w(21)=19.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(22)=19.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(23)=21.0D0\*pi/40.0D0

w(24)=23.0D0\*pi/40.0D0

w(25)=23.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(26)=23.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(27)=25.0D0\*pi/40.0D0

w(28)=27.0D0\*pi/40.0D0

w(29)=27.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(30)=27.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(31)=29.0D0\*pi/40.0D0

w(32)=31.0D0\*pi/40.0D0

w(33)=31.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(34)=31.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(35)=33.0D0\*pi/40.0D0

w(36)=35.0D0\*pi/40.0D0

w(37)=35.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(38)=35.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(39)=37.0D0\*pi/40.0D0

w(40)=39.0D0\*pi/40.0D0

w(41)=39.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(42)=40.0D0\*pi/40.0D0

do i=1,41

h(i)=w(i+1)-w(i)

enddo

sum1=0.0D00

sum2=0.0D00

do i=2,41

sum2=sum2+f(w(i))\*(h(i-1)+h(i))

enddo

do i=1,41

sum1=sum1+f(w(i)+h(i)/2.0D0)\*h(i)

enddo

simposcil3=(f(w(1))\*h(1)+f(w(42))\*h(41)+4.0D0\*sum1+sum2)/6.0D0

return

end

!Функия реализующая метод Гаусса-3 для многочлена степени n

real\*8 function gausspoly(n,nn,a,b) !8

implicit none

real\*8 sum1,sum2,q,x,a,b !8

real\*8 h,g !4

dimension q(3),x(3)

integer n,nn,j,k

h=(b-a)/nn

sum2=0.0D00

x(1)=-0.774596669241483D00

x(2)=0.000000000000000D00

x(3)=0.774596669241483D00

q(1)=0.555555555555556D00

q(2)=0.888888888888889D00

q(3)=0.555555555555556D00

do j=1,3

sum1=0.0D00

do k=0,nn-1

sum1=sum1+h\*g((a+h\*(k)+a+h\*(k+1))/2.0D00+x(j)\*h/2.0D00,n) !8

!sum1=sum1+dble(h\*g(real(a)+h\*(2.\*k+real(x(j))+1.)/2.,n)) !4

enddo

sum2=sum2+q(j)\*sum1

enddo

gausspoly=sum2/2.0D00

return

end

!Функия реализующая метод Гаусса-3 для осцилирующей функции

real\*8 function gaussoscil(nn,a,b) !8

implicit none

real\*8 sum1,sum2,q,x,a,b !8

real\*8 h,f !4

dimension q(3),x(3)

integer nn,j,k

h=(b-a)/nn

sum2=0.0D00

x(1)=-0.774596669241483D00

x(2)=0.000000000000000D00

x(3)=0.774596669241483D00

q(1)=0.555555555555556D00

q(2)=0.888888888888889D00

q(3)=0.555555555555556D00

do j=1,3

sum1=0.0D00

do k=0,nn-1

sum1=sum1+h\*f((a+h\*k+a+h\*(k+1))/2.0D00+x(j)\*h/2.0D00) !8

!sum1=sum1+dble(h\*f(real(a)+h\*(2.\*k+real(x(j))+1.)/2.)) !4

enddo

sum2=sum2+q(j)\*sum1

enddo

gaussoscil=sum2/2.0D00

return

end

real\*8 function gaussoscil2()

implicit none

real\*8 sum1,sum2

real\*8 h,f,q,x

integer i,j,k

real\*8 w,pi

dimension w(42),h(41),q(3),x(3)

pi=3.14159265358979D00

x(1)=-0.774596669241483D00

x(2)=0.000000000000000D00

x(3)=0.774596669241483D00

q(1)=0.555555555555556D00

q(2)=0.888888888888889D00

q(3)=0.555555555555556D00

w(1)=0

w(2)=pi/40.0D0

w(3)=pi/40.0D0+1.0D0\*pi/60.0D0

w(4)=pi/40.0D0+2.0D0\*pi/60.0D0

w(5)=3.0D0\*pi/40.0D0

w(6)=5.0D0\*pi/40.0D0

w(7)=5.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(8)=5.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(9)=7.0D0\*pi/40.0D0

w(10)=9.0D0\*pi/40.0D0

w(11)=9.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(12)=9.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(13)=11.0D0\*pi/40.0D0

w(14)=13.0D0\*pi/40.0D0

w(15)=13.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(16)=13.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(17)=15.0D0\*pi/40.0D0

w(18)=17.0D0\*pi/40.0D0

w(19)=17.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(20)=17.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(21)=19.0D0\*pi/40.0D0

w(22)=21.0D0\*pi/40.0D0

w(23)=21.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(24)=21.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(25)=23.0D0\*pi/40.0D0

w(26)=25.0D0\*pi/40.0D0

w(27)=25.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(28)=25.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(29)=27.0D0\*pi/40.0D0

w(30)=29.0D0\*pi/40.0D0

w(31)=29.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(32)=29.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(33)=31.0D0\*pi/40.0D0

w(34)=33.0D0\*pi/40.0D0

w(35)=33.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(36)=33.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(37)=35.0D0\*pi/40.0D0

w(38)=37.0D0\*pi/40.0D0

w(39)=37.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(40)=37.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(41)=39.0D0\*pi/40.0D0

w(42)=40.0D0\*pi/40.0D0

do i=1,41

h(i)=w(i+1)-w(i)

enddo

sum1=0.0D00

sum2=0.0D00

do j=1,3

sum1=0.0D00

do k=1,41

sum1=sum1+h(k)\*f((w(k)+w(k+1))/2.0D00+x(j)\*h(k)/2.0D00) !8

enddo

sum2=sum2+q(j)\*sum1

enddo

gaussoscil2=sum2/2.0D00

return

end

real\*8 function gaussoscil3()

implicit none

real\*8 sum1,sum2

real\*8 h,f,q,x

integer i,j,k

real\*8 w,pi

dimension w(42),h(41),q(3),x(3)

pi=3.14159265358979D00

x(1)=-0.774596669241483D00

x(2)=0.000000000000000D00

x(3)=0.774596669241483D00

q(1)=0.555555555555556D00

q(2)=0.888888888888889D00

q(3)=0.555555555555556D00

w(1)=0

w(2)=1.0D0\*pi/60.0D0

w(3)=1.0D0\*pi/40.0D0

w(4)=3.0D0\*pi/40.0D0

w(5)=3.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(6)=3.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(7)=5.0D0\*pi/40.0D0

w(8)=7.0D0\*pi/40.0D0

w(9)=7.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(10)=7.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(11)=9.0D0\*pi/40.0D0

w(12)=11.0D0\*pi/40.0D0

w(13)=11.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(14)=11.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(15)=13.0D0\*pi/40.0D0

w(16)=15.0D0\*pi/40.0D0

w(17)=15.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(18)=15.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(19)=17.0D0\*pi/40.0D0

w(20)=19.0D0\*pi/40.0D0

w(21)=19.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(22)=19.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(23)=21.0D0\*pi/40.0D0

w(24)=23.0D0\*pi/40.0D0

w(25)=23.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(26)=23.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(27)=25.0D0\*pi/40.0D0

w(28)=27.0D0\*pi/40.0D0

w(29)=27.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(30)=27.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(31)=29.0D0\*pi/40.0D0

w(32)=31.0D0\*pi/40.0D0

w(33)=31.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(34)=31.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(35)=33.0D0\*pi/40.0D0

w(36)=35.0D0\*pi/40.0D0

w(37)=35.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(38)=35.0D0\*pi/40.0D0+2.0D0\*pi/60.0D0

w(39)=37.0D0\*pi/40.0D0

w(40)=39.0D0\*pi/40.0D0

w(41)=39.0D0\*pi/40.0D0+1.0D0\*pi/60.0D0

w(42)=40.0D0\*pi/40.0D0

do i=1,41

h(i)=w(i+1)-w(i)

enddo

sum1=0.0D00

sum2=0.0D00

do j=1,3

sum1=0.0D00

do k=1,41

sum1=sum1+h(k)\*f((w(k)+w(k+1))/2.0D00+x(j)\*h(k)/2.0D00) !8

enddo

sum2=sum2+q(j)\*sum1

enddo

gaussoscil3=sum2/2.0D00

return

end

program main

implicit none

integer m1,nn,m2,n

real\*8 a,b,simppoly,simposcil,gausspoly,gaussoscil !8

real\*8 gaussoscil2,gaussoscil3,simposcil2,simposcil3

open(10,file='simposcil2.txt',status='unknown',err=4)

write(10,300) 41,0.0D00,3.14159265358979D00,simposcil2()

close(10)

open(10,file='simposcil3.txt',status='unknown',err=4)

write(10,300) 41,0.0D00,3.14159265358979D00,simposcil3()

close(10)

open(10,file='gaussoscil2.txt',status='unknown',err=4)

write(10,300) 41,0.0D00,3.14159265358979D00,gaussoscil2()

close(10)

open(10,file='gaussoscil3.txt',status='unknown',err=4)

write(10,300) 41,0.0D00,3.14159265358979D00,gaussoscil3()

close(10)

1 print\*, 'Select method:'

print\*, '1) Method Simpson'

print\*, '2) Method Gaus-3'

print\*, '3) Exit'

read(\*,\*) m1

if(m1.eq.3) goto 10

if(m1.lt.1.or.m1.gt.3) goto 1

if(m1.eq.1)then

! Метод Симпсона

print\*, 'Method Simpson'

print\*, 'Enter quantity of segments:'

!количество сегментов разбиения отрезка

read(\*,\*) nn

print\*, 'Enter begin and end:'

read(\*,\*) a,b

2 print\*, 'Select:'

print\*, '1) Polynomial'

print\*, '2) Oscillating'

read(\*,\*) m2

if(m2.lt.1.or.m2.gt.2) goto 2

if(m2.eq.1)then

print\*, 'Enter degree of polynomial:'

read(\*,\*)n

open(10,file='simppoly.txt',status='unknown',err=4)

write(10,301) n

write(10,300) nn,a,b,simppoly(n,nn,a,b)

close(10)

endif

if(m2.eq.2)then

open(10,file='simposcil.txt',status='unknown',err=4)

write(10,300) nn,a,b,simposcil(nn,a,b)

close(10)

endif

endif

! Метод Гаусса-3

if(m1.eq.2)then

print\*, 'Method Gaus-3'

print\*, 'Enter quantity of segments:'

read(\*,\*) nn

print\*, 'Enter begin and end:'

read(\*,\*) a,b

5 print\*, 'Select:'

print\*,'1) Polynomial'

print\*, '2) Oscillating'

read(\*,\*) m2

if(m2.lt.1.or.m2.gt.2) goto 5

if(m2.eq.1)then

print\*, 'Enter degree of polynomial:'

read(\*,\*) n

open(10,file='gausspoly.txt',status='unknown',err=4)

write(10,301) n

write(10,300) nn,a,b,gausspoly(n,nn,a,b)

! DEBUG

write(\*,301) n

write(\*,300) nn,a,b,gausspoly(n,nn,a,b)

! END\_DEBUG

close(10)

endif

if(m2.eq.2)then

open(10,file='gaussoscil.txt',status='unknown',err=4)

write(10,300) nn,a,b,gaussoscil(nn,a,b)

! DEBUG

write(\*,300) nn,a,b,gaussoscil(nn,a,b)

! END\_DEBUG

close(10)

endif

endif

print\*, 'Completed.'

pause

goto 1

300 format('Quantity of segments = 'I8/'a = 'G23.16/'b = 'G23.16/

,'Integral = 'G23.16)

301 format('Degree of polynomial = 'I8)

4 print\*, 'Error writing file!'

goto 1

10 end