Aim : To find the numerical integration using Simpson’s 1/3 rule

Theory :

<u>Simpson’s 1/3 rule :</u>

It is an extension of the trapezoidal rule in which the integrand is approximated by a second-order polynomial.

<u>Find the Integration :</u>

One dimensional Integration:

The formula for the Simpson's method is :

$$\int_{a}^{b} f(x)dx \sim \frac{h}{3}[\ f(a)\ +\ f(b)\ +\ 4\sum_{odd} f(a+i*h)\ +\ 2\sum_{even} f(a+i*h)]$$

and $h=\frac{(b-a)}{n}$

Here were first we define the function and calculating the all summation and then putting them into the formula getting the answer. Here n is number of the intervals and this define also the error, larger the number of the intervals lesser the error.

Two dimensional Integration :

In 2 D integration we need just do integration first with one variable and then do it with other variable we just need to put the value of first Simpson's formula as we do second integration .

Program in FORTRAN95 , Output and Flow chart :

<u>One dimensional Integration :</u>

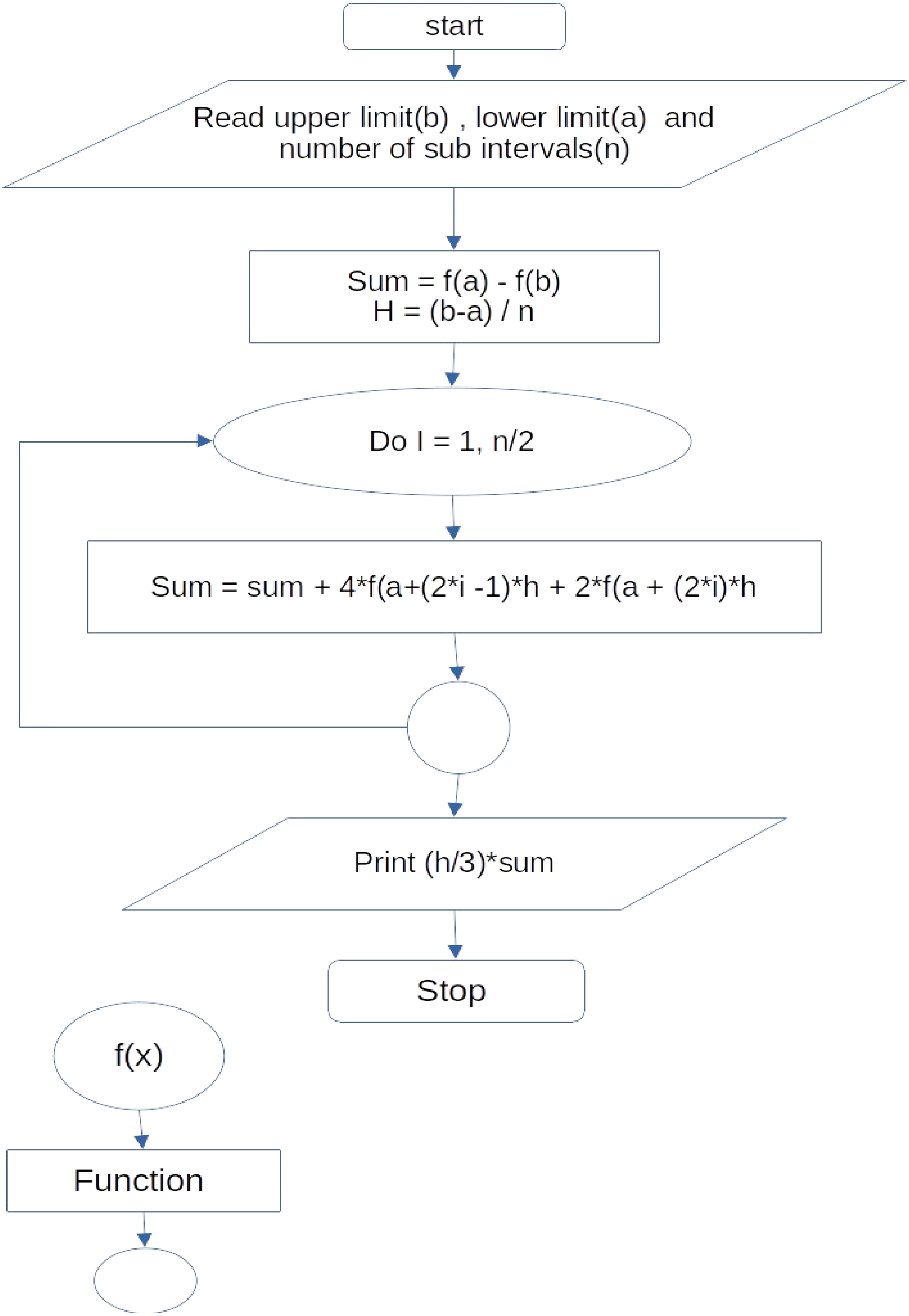
```
program Simpson_rule
   implicit none

   real func , sum , upper , lower , h
   integer i , d

! getting the input for upper limit and lower limit of integration

   print *, "Enter the upper limit of the integration :: "
   read(*,*) upper
   print *, "Enter the lower limit of the integration :: "
```

```
      read(*,*) lower
      print*, "Enter the number of iteration :: "
      read(*,*) d

   ! validating the number of iteration

      if(d <= 0) then
         print *, "Value of iteration is not correct"
         call Exit(1)
      endif

      if(mod(d,2) .ne. 0)then
         d = d+1
      endif
   ! defining the interval

      h = (upper - lower)/(real(d))

      sum =  func(lower) - func(upper)

   ! loop calculation the area of the all rectangle inside the limits
      do i = 1,d/2
         sum = sum + 4*func(lower + real(2*i-1)*h) + 2*func(lower + real(2*i)*h)
      enddo

      sum = sum*(h/3)

   ! print the result
      print*, "Value of the integration is ",sum

      stop
      end program

   ! defining the function which integration we interested  to find
   real function func(x)
      real x , value
      value = x**2
      func = value
      return
  end function
```

Output :

```
 Enter the upper limit of the integration ::
15
 Enter the lower limit of the integration ::
0
 Enter the number of iteration ::
1000
 Value of the integration is    1125.00024
```

Flow chart :


start
Read upper limit(b) , lower limit(a) and number of sub intervals(n)
Sum = f(a) - f(b)
H = (b-a) / n
Do I = 1, n/2
Sum = sum + 4*f(a+(2*i -1)*h + 2*f(a + (2*i)*h
Print (h/3)*sum
Stop
f(x)
Function

Two dimensional Integration :

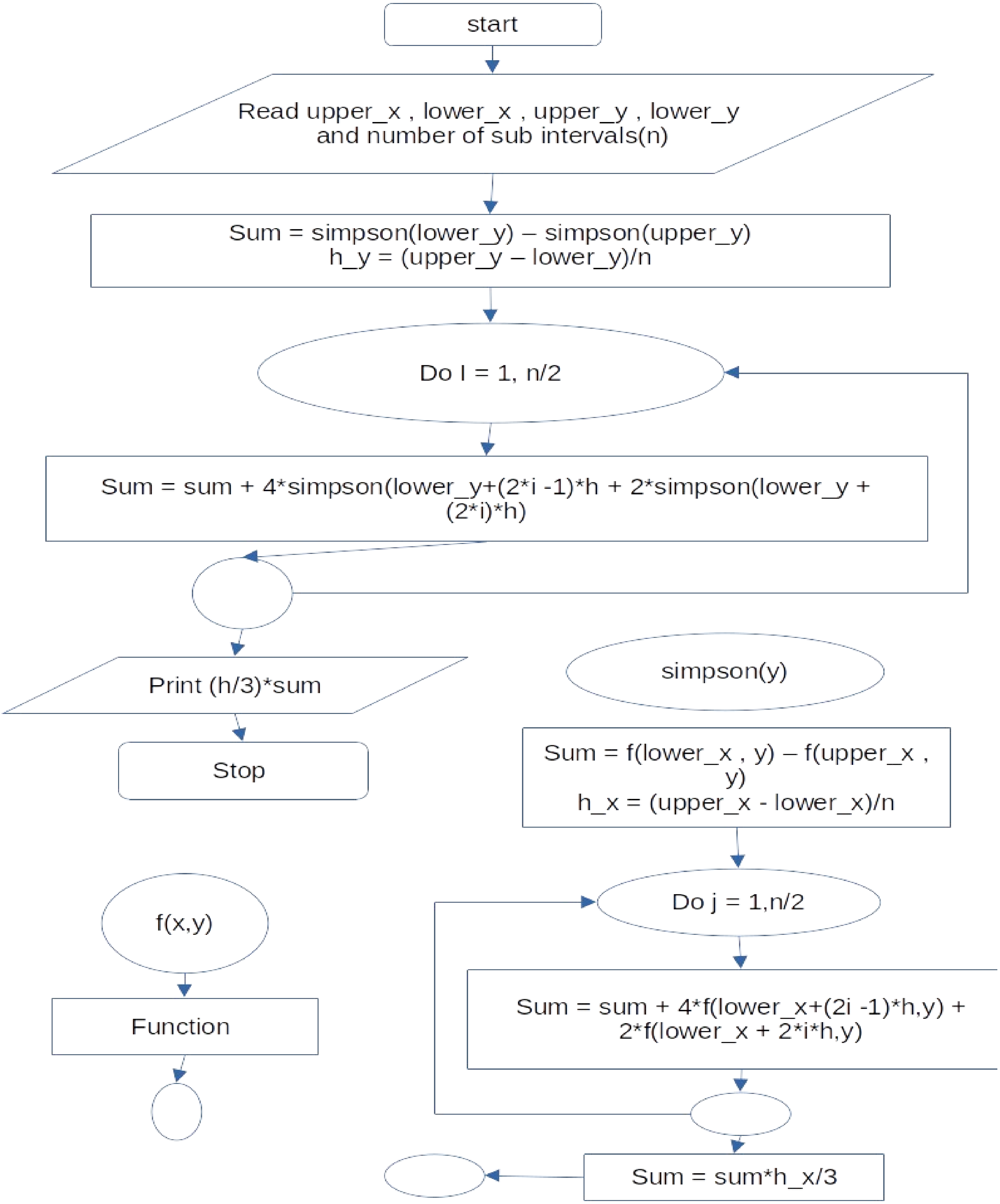
```
program simpson_rule
   implicit none

   real :: sum , upper_x , lower_x , upper_y , lower_y , h_y , h_x
   integer :: i , d_x , d_y

! getting the input for upper limit and lower limit of integration

   print *, "Enter the upper limit of x axis of the integration :: "
   read *, upper_x
   print *, "Enter the lower limit of x axis of the integration :: "
   read *, lower_x
   print *,"Enter the upper limit of y axis of the integration :: "
   read *, upper_y
   print *, "Enter the lower limit of y axis of the integration :: "
   read *, lower_y
   print*, "Enter the number of iteration of x axis :: "
   read *, d_x
   print *, "Enter the number of iteration of y axis :: "
   read *, d_y

! validating the number of iteration
   if(d_x <= 0 .or. d_y <=0) then
      print *, "Value of iteration is not correct"
      call Exit(1)
   endif

   if(mod(d_x,2) .ne. 0)then
      d_x = d_x +1
   endif

   if(mod(d_y,2) .ne. 0)then
      d_y = d_y + 1
   endif

! defining the interval

   h_y = (upper_y - lower_y)/(real(d_y))
   h_x = (upper_x - lower_x)/(real(d_x))

   sum =  simpson(lower_y) - simpson(upper_y)

! loop calculation the area of the all rectangle inside the limits
   do i = 1,d_y/2 ! integration with y axis and simpson do for x axis first
```

```
      sum = sum + 4*simpson(lower_y + real(2*i-1)*h_y) + 2*simpson(lower_y +
real(2*i)*h_y)
   enddo

   sum = sum*(h_y/3)

! print the result
   print*, "Value of the integration is ",sum

   stop
! defining the function which integration we interested to find
   contains

      real function func(x,y)
         real ,intent(in) :: x , y
         real :: value
         value = log(x+2*y)
         func = value
         return
      end function func

      real function simpson(t) ! integrated for value of x
         real x_sum , t
         integer j
         X_sum = func(lower_x,t) - func(upper_x,t)
         do j = 1,d_x/2
            x_sum = x_sum + 4*func(lower_x + (2*j-1)*h_x,t) + 2*func(lower_x + 2*j*h_x,t)
         enddo
         x_sum = x_sum*(h_x/3.0)
         simpson = x_sum
         return
      end function simpson
end program
```

Output :

```
 Enter the upper limit of x axis of the integration ::
2
 Enter the lower limit of x axis of the integration ::
1.4
 Enter the upper limit of y axis of the integration ::
1.5
 Enter the lower limit of y axis of the integration ::
1
 Enter the number of iteration of x axis ::
1000
 Enter the number of iteration of y axis ::
1000
 Value of the integration is   0.429554611
```

Flow Chart :


start
Read upper_x , lower_x , upper_y , lower_y and number of sub intervals(n)
Sum = simpson(lower_y) – simpson(upper_y)
h_y = (upper_y – lower_y)/n
Do I = 1, n/2
Sum = sum + 4*simpson(lower_y+(2*i -1)*h + 2*simpson(lower_y + (2*i)*h)
Print (h/3)*sum
Stop
simpson(y)
Sum = f(lower_x , y) – f(upper_x , y)
h_x = (upper_x - lower_x)/n
Do j = 1,n/2
Sum = sum + 4*f(lower_x+(2i -1)*h,y) + 2*f(lower_x + 2*i*h,y)
Sum = sum*h_x/3
f(x,y)
Function