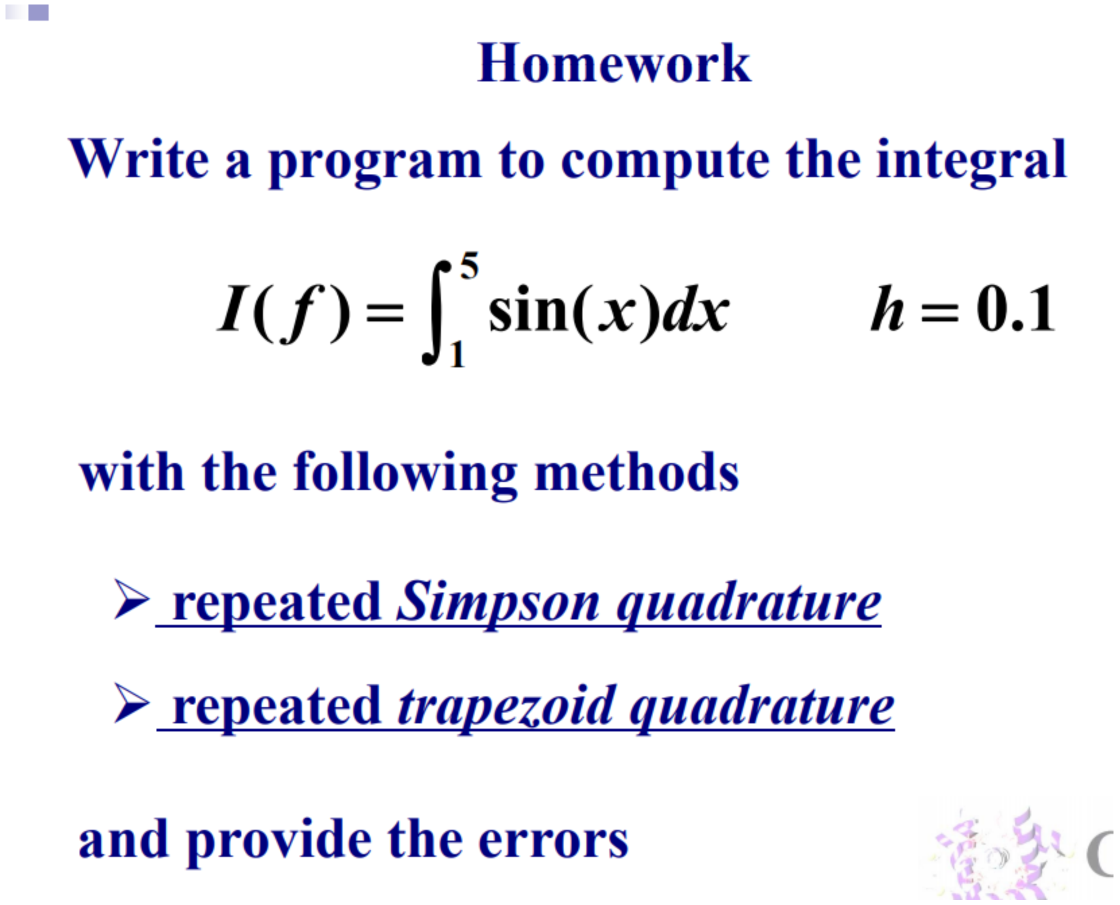
The third schoolwork of Computational Physics

万炫均 物理 1701 U201710170

Description of this chapter:

Integration is always a common problem when solving Physics problems. Howerer solving formulas by ourselves can be a really hard work when the formula to integrate can be very complex. And even some times there is no specific result of an integration. Thus we can turn to computer to help us calculate some integration in value.

- **Description of the problem**



Homework

Write a program to compute the integral

$$I(f)=\int_1^5 \sin(x)dx \qquad h=0.1$$

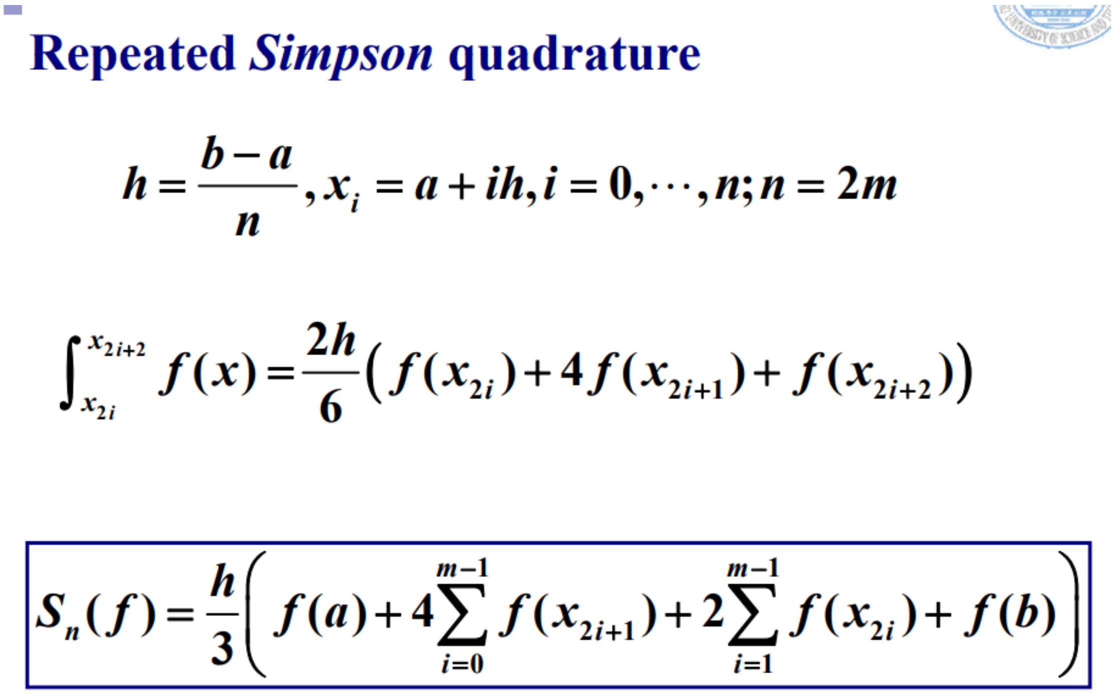
with the following methods

➢ repeated *Simpson quadrature*

➢ repeated *trapezoid quadrature*

and provide the errors

- **Formula to use**
  - Simpson quadrature



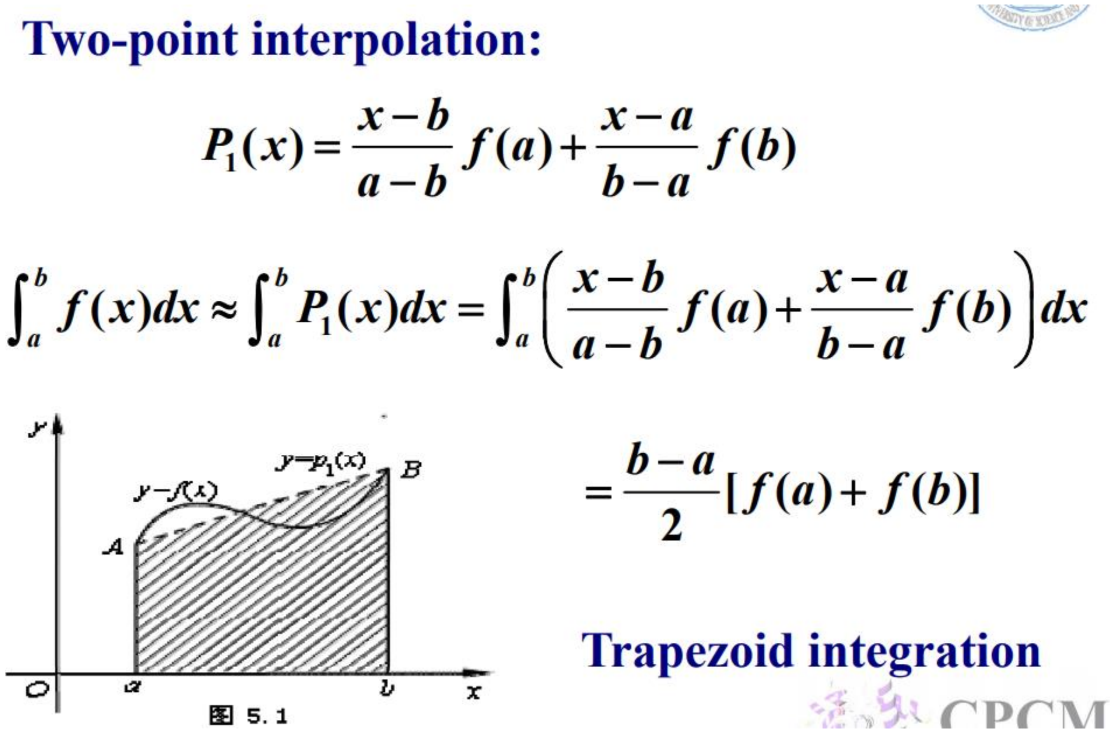
Repeated *Simpson* quadrature

$$h=\frac{b-a}{n},x_i=a+ih,i=0,\cdots,n;n=2m$$

$$\int_{x_{2i}}^{x_{2i+2}} f(x)=\frac{2h}{6}\left(f(x_{2i})+4f(x_{2i+1})+f(x_{2i+2})\right)$$

$$\boxed{S_n(f)=\frac{h}{3}\left(f(a)+4\sum_{i=0}^{m-1}f(x_{2i+1})+2\sum_{i=1}^{m-1}f(x_{2i})+f(b)\right)}$$

- Trapezoid quadrature



Two-point interpolation:

$$P_1(x)=\frac{x-b}{a-b}f(a)+\frac{x-a}{b-a}f(b)$$

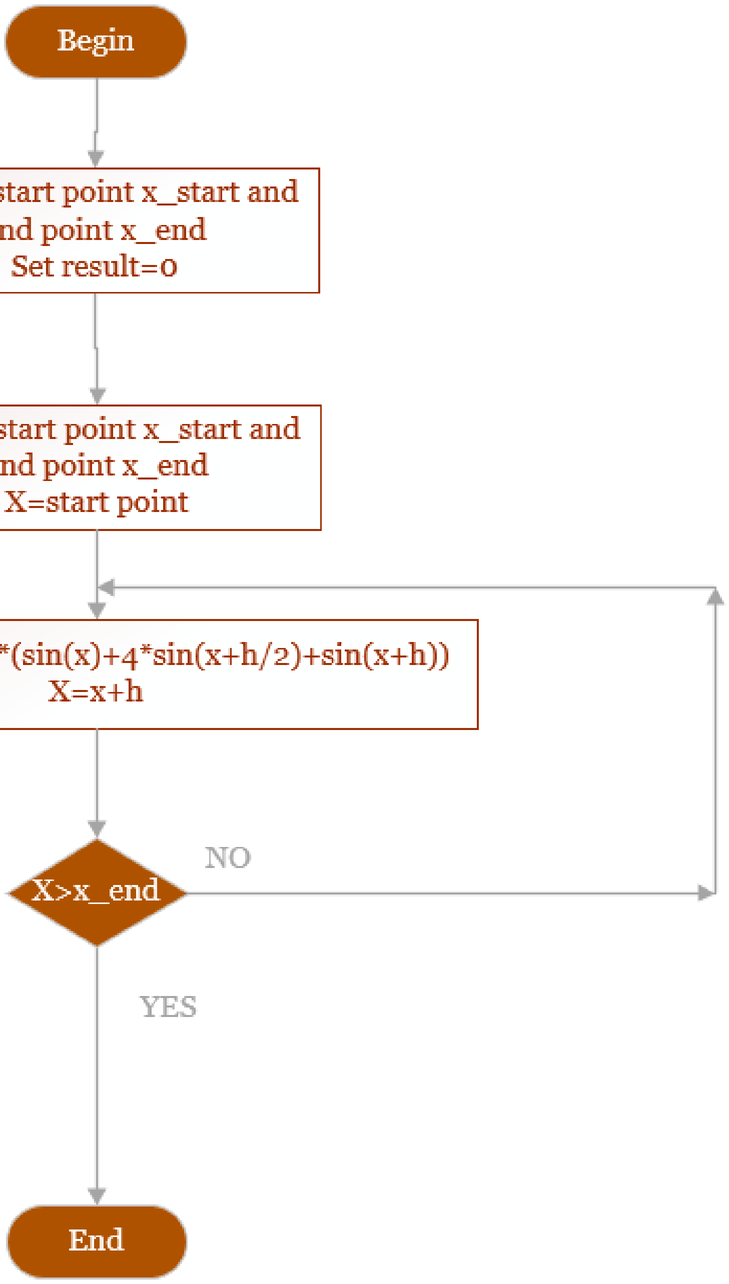
$$\int_a^b f(x)dx\approx\int_a^b P_1(x)dx=\int_a^b\left(\frac{x-b}{a-b}f(a)+\frac{x-a}{b-a}f(b)\right)dx$$

$$=\frac{b-a}{2}[f(a)+f(b)]$$

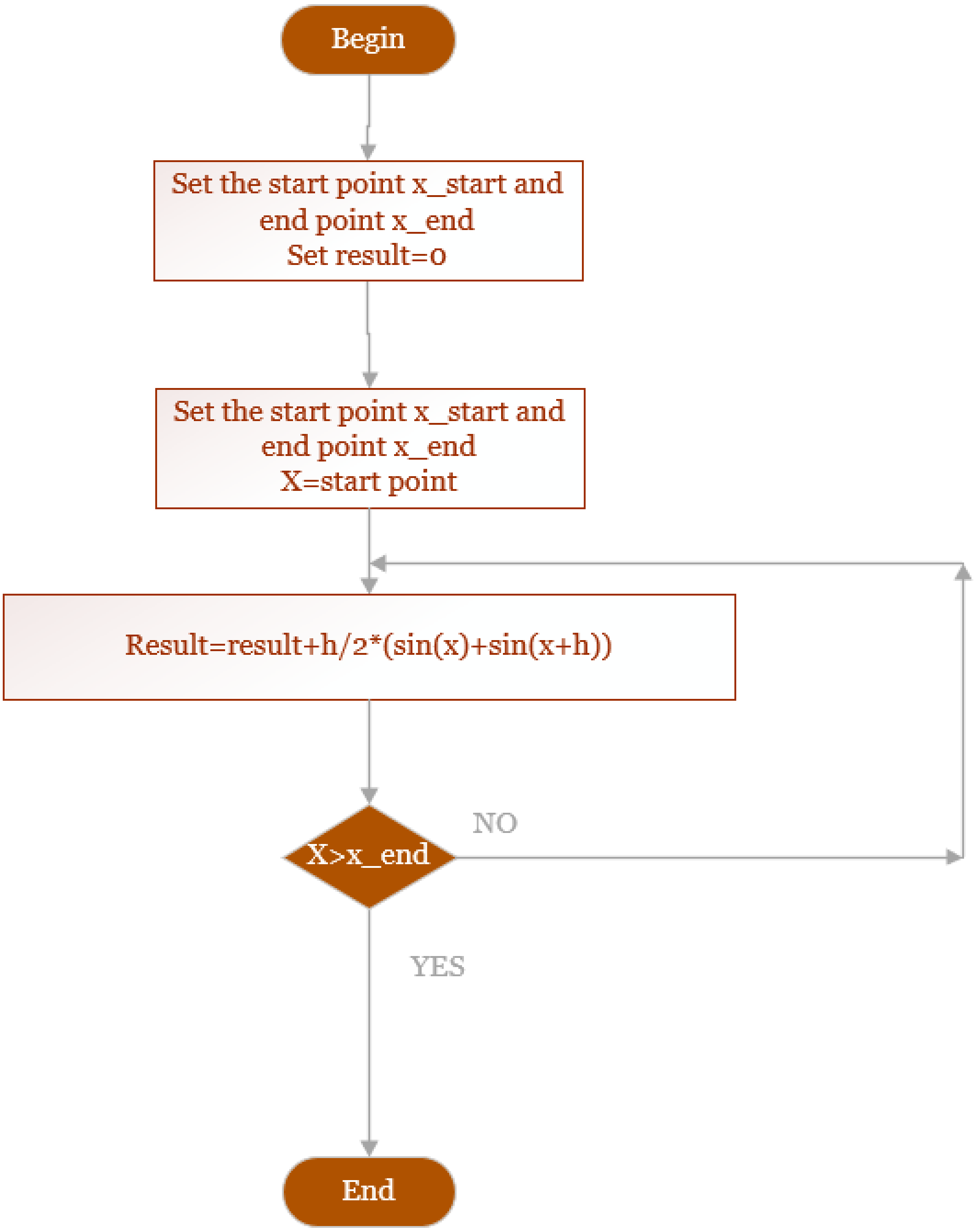
y
y=P1(x)
B
y=f(x)
A
O
a
b
x
图 5.1
CPCM

Trapezoid integration

- **Flow chart**
  - **Simpson Integrate**


Begin
Set the start point x_start and end point x_end
Set result=0
Set the start point x_start and end point x_end
X=start point
Result=result+h/6*(sin(x)+4*sin(x+h/2)+sin(x+h))
X=x+h
X>x_end
NO
YES
End

- **Trapezoid quadradate**


Begin
Set the start point x_start and end point x_end
Set result=0
Set the start point x_start and end point x_end
X=start point
Result=result+h/2*(sin(x)+sin(x+h))
X>x_end
NO
YES
End

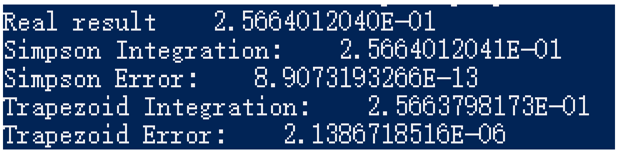
- **Source code**

```
program Main
    use Integrate
    real*8 :: simpson_result,trapezoid_result,real_result,f_origin
    real*8 :: step
    external f
    procedure(func),pointer ::  f_ptr=>null()

    f_ptr => f
    step =0.01
    call Simpson(f_ptr,dble(1.0),dble(5.0),step,simpson_result)
    call Trapezoid(f_ptr,dble(1.0),dble(5.0),step,trapezoid_result)

    real_result = f_origin(dble(5.0))-f_origin(dble(1.0))

    print "(a,es20.10)","Real result",real_result
    print "(a,es20.10)","Simpson Integration:",simpson_result
    print "(a,es20.10)","Simpson Error:",abs(simpson_result-real_result)
    print "(a,es20.10)","Trapezoid Integration:",trapezoid_result
    print "(a,es20.10)","Trapezoid Error:",abs(trapezoid_result-real_result)


end program Main

function f(x)
    real*8,intent(in) :: x
    real*8 :: f
    !TODO_add_body
    f=sin(x)
end function f

function f_origin(x)
    real*8 :: x
    real*8 :: f_origin
    !TODO_add_body

    f_origin=-cos(x)

end function f_origin
```

```
module Integrate
    abstract interface
        function func(x)
            real*8,intent(in) :: x
            real*8 :: func
        end function func
    end interface
contains
    subroutine Simpson(f_ptr, x_start,x_end,step,result)
        procedure(func),pointer,intent(in) ::  f_ptr
        real*8,intent(in)  :: x_start,x_end,step
        real*8,intent(out) :: result
        !Local vars
        real*8 :: x,h

        result = 0

        x=x_start
        h=step

        do while(.true.)
            if(x+h>x_end)then
                h=x_end-x
            end if

            result=result+h/6*(f_ptr(x)+4*f_ptr(x+h/2)+f_ptr(x+h))
            x=x+h

            if (x>=x_end)then
                exit
            end if
        end do
        !TODO_add_body
    end subroutine Simpson



    subroutine Trapezoid (f_ptr, x_start,x_end,step,result)
        procedure(func),pointer,intent(in) ::  f_ptr
        real*8,intent(in)  :: x_start,x_end,step
        real*8,intent(out) :: result
        !Local vars
        real*8 :: x,h

        result = 0

        x=x_start
        h=step

        do while(.true.)
            if(x+h>x_end)then
                h=x_end-x
            end if

            result=result+(f_ptr(x)+f_ptr(x+h))*h/2
            x=x+h
```

```
            if (x>=x_end)then
                exit
            end if
        end do
    end subroutine Trapezoid

end module Integrate
```

- **Example and Result**



```
Real result    2.5664012040E-01
Simpson Integration:    2.5664012041E-01
Simpson Error:    8.9073193266E-13
Trapezoid Integration:    2.5663798173E-01
Trapezoid Error:    2.1386718516E-06
```

- **Demo**

  Check the folder ”Integration” in the directory and follow the instruction to set up the matrices and vectors