Regla Trapeizodial

Martínez Navallez Daniel Isac

Octubre 2018

1 Introducción

En este artículo se verá la regla de cuadratura como un método para aproximar integrales definidas, por lo que es muy utilizado en análisis numérico.

El método del trapezoide tiene registro desde el año 50 a.C. en Babilonia como método para calcular la velocidad de Jupiter a lo largo de su eclíptica.

Por último se presentara un código en FORTRAN para el cálculo de la integral definida de cierta funcón utilizando el método del trapezoide.

2 Regla Trapezoidal

Utilizada para calcular integrales definidas:

$$\int_a^b f(x)dx$$

Con lo que se quiere calcular la región (área) que se encuentra debajo de una función indicada de la siguiente manera:

$$\int_a^b f(x)\,dx \approx \frac{\Delta x}{2}(f(x_0)+2f(x_1)+2f(x_2)+2f(x_3)+\cdots+2f(x_{n-1})+2f(x_n))$$

donde:

$$\Delta x = \frac{b-a}{2} \text{ y } x_i = a + i\Delta x$$

Para poder obtener una mejor aproximación a nuestra integral definida, es conveniente dividir el intervalo de integración de tal manera que se le aplique la regla del trapecio a cada subintervado y sumar los resultados.

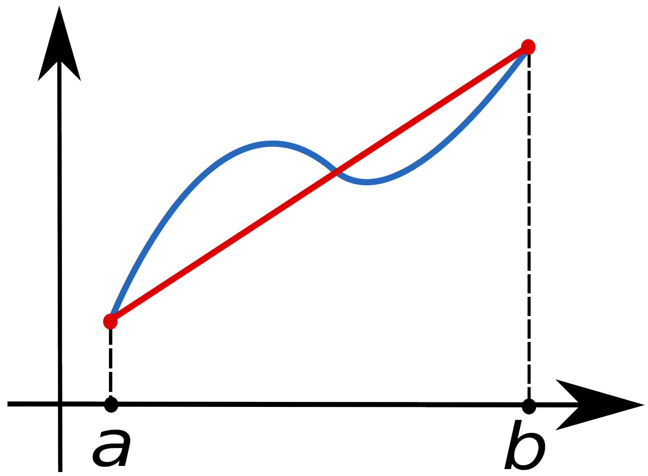

Figure 1: Figura 1

Tendremos que x_k pertenece al intervalo $[a, b]$

dado que $a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$ y $\Delta x_k = x_k - x_{k-1}$

Entonces:

$$\int_a^b f(x)dx \approx \sum_{k=1}^{N} \frac{f(x_{k-1}) + f(x_k)}{2} \Delta x_k$$

el método anterior se utiliza para los casos de rejilla no uniforme, en los casos de rejilla uniforme queda de la siguiente forma:

$$\Delta x_k = \Delta x = \frac{b-a}{N}$$

Para que la aproximación a la integral quede así:

$$\int_a^b f(x)dx \approx \frac{\Delta x}{2} \sum_{k=1}^{N} (f(x_{k-1}) + f(x_k))$$

$$= \frac{\Delta x}{2}(f(x_0) + 2f(x_1) + 2f(x_2) + 2f(x_3) + \cdots + 2f(x_{N-1}) + 2f(x_N))$$

$$= \frac{\Delta x}{2}\left(f(x_0) + 2\sum_{k=1}^{N-1} f(x_k) + f(x_N)\right)$$

Para este caso se requiere un menor número de evaluciones de la función a calcular.

3 Código en FORTRAN

```
PROGRAM TRAPEZOIDE

  IMPLICIT NONE
  REAL :: a,b

  PRINT*, "[a,b]"
  READ*,a,b

  call trapezoid_integration(a,b)

  contains

    subroutine trapezoid_integration(a,b)
      IMPLICIT NONE
      REAL ::  a,b
      REAL :: integral,u,h,error,integralo, T
      INTEGER :: i,n

      integral = 0.0
      n=10
      error=2.0
      integralo=0.0


      DO WHILE(error>1.0)


      DO i=0,n
         u = a + ((b-a)*float(i)/float(n))

         IF ((i.eq.0).or.(i.eq.n)) then
            integral = integral+integrand(u)
         ELSE
            integral = integral+(2.0*integrand(u))
         END IF
      END DO


     error=abs(integral-integralo)/integralo
```

```
     integralo=integral


      n=n*2
   END DO

   h=(b-a)/(n)

   T=integral*(h/2.0)

    PRINT*,"error=",error
     WRITE(*,*) "Integral=",T
    end subroutine trapezoid_integration

    function integrand(x) result (value)
      IMPLICIT NONE
      REAL :: x
      REAL :: value

      IF (x .lt. 0.00001) THEN
         x = 0.00001
      END IF

      value = (x**4)*EXP(X)/((EXP(X)-1.0)**2)
    end function integrand

END PROGRAM TRAPEZOIDE
```

4 Conclusion

Se mostró uno método numérico para calcular la integral definida por medio de aproximaciones, muy útil al momento de aplicarla con funciones que no son integrables directamente; por supuesto que en el área de la programación resulta también de mucha utilidad ya que el cálculo requiere de muchas evaluaciones para que el resultado sea tan aproximado como se requiera y reduciendo el tiempo que se tarda en llegar a éste.

References