

# Analisis Numerico-Taller Integrales

Cristian Camilo Contreras Borja

Kevin Andres Garzon Ospina

## Ejercicio 7.5



7.5 De una función  $f$ , conocemos la siguiente información

$$f(x) = ?$$

$x$	$f(x)$
0	3.592
0.2	3.110
0.4	3.017
0.6	2.865
0.8	2.658

Tabla 7.5

- a.) Aproximar  $\int_0^{0.8} f(x) dx$  usando regla del Trapecio.
- b.) Aproximar  $\int_0^{0.8} f(x) dx$  usando regla del Simpson.
- c.) Aproximar  $\int_0^{0.8} f(x) dx$  usando Romberg (interpolar con polinomios de grado 2).

### A) Trapecio

```
#Punto A
#Metodo de trapecios
def integral_trapecios(funcion,a,b,p):
    r = (b-a)/p
    s = 0
    for i in range(1,p):
        s = s+funcion(a+i*r)
    t = r/2*(funcion(a)+2*s+funcion(b))

    return t
```

```

Trapecios :
1  2.5000000000000009
2  2.4568000000000065
3  2.4329272976680447
4  2.4234000000000006
5  2.4187845120000006
6  2.4162227709190733
7  2.414659558517291
8  2.4136375000000063
9  2.412933380751585
10 2.4124280320000056
11 2.412053220408448
12 2.411767626886151
13 2.411545057946156
14 2.411368263223663
15 2.4112255097503494

```

## B) Simpson

```

#Punto B
#Metodo de simpson
def integral_simpson(funcion,a,b,p):
    r = (b-a)/p
    y = 0
    x = a
    for i in range(1, p):
        y = y+2*(i%2+1)* funcion(x+i*r)
    y = r/3 * (funcion(a)+y+funcion(b))
    return y

```

```

Simpson :
1  1.6666666666666723
2  2.4424000000000063
3  2.1672318244170143
4  2.4122666666666728
5  2.2659810986666717
6  2.410654595336083
7  2.3077049007358097
8  2.4103833333333387
9  2.3307154668157004
10 2.410309205333339
11 2.3452937549803123
12 2.4102825788751776
13 2.3553565024567353
14 2.4102711647924533
15 2.362720183659814

```

## C) Romberg

```
#Punto C
romberg= 2
print('\n \n Romberg Segundo Grado :', integrate.romberg(funcion, a, b, divmax=romberg))
```

Romberg Segundo Grado : 2.410257777777784

## Ejercicio Propuesto

\*  $\int_{-1}^1 (1 + \sin e^{3x}) dx$   $\begin{cases} \text{Trapecio compuesta} \\ \text{Simpson} \end{cases}$  \* Adaptativas \*

- **Metodo Simpson Compuesto**

```
#Metodo de simpson -----
simpson = function(f, a, b, n) {
  Ig = integrate(f,a,b)
  res = Ig$value
  if (n%%2 != 0) stop ("simpson: n es par!")
  k = (b-a)/n
  x1 = seq(1, n-1, by = 2)
  x2 = seq(2, n-2, by = 2)
  y = f(a+(0:n)*k)
  abs(k/3 * ( f(a) + f(b) + 4*sum (y[x1]) + 2*sum(sum (y [x2]) ) ))
  op = abs(k/3 * ( f(a) + f(b) + 4*sum (y[x1]) + 2*sum (sum (y[x2]) ) ))
  error = abs( res- op)
  cat("Integral metodo de simpson: ",op, "\n")
  cat("Error metodo de Simpson: ",error, "\n")
}
```

```
Integral metodo de Simpson: 2.599276
Error metodo de Simpson: 0.09846672
```

- **Metodo Trapecio Compuesto**

```
#Metodo del trapecio
trapezoid = function(f, a, b, n) {
  Ig = integrate(f,a,b)
  res = Ig$value
  k = (b-a)/n
  x = seq(a, b, by=k)
  y = f(x)
  s = k * (abs(y[1]/2) + abs( sum(y[2:n])) + abs (y [n+1]/2))
  op = k * (abs(y[1]/2) + abs( sum(y[2:n])) + abs (y [n+1]/2))
  cat(" Integral metodo del trapecio: ",op,"\n")
  error = abs( res - op)
  cat(" Error metodo del trapecio: ",error,"\n")
}
```

Integral metodo del trapecio: 2.394406  
Error metodo del trapecio: 0.1064027

## Referencias

- <http://blog.espol.edu.ec/analisisnumerico/regla-de-simpson-1-3/>
- <https://github.com/xDavidPM/Analisis-numerico-python/blob/master/trapecio.py>
- <https://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.romberg.html>
- [http://www.dma.uvigo.es/~aurea/Tema2\\_Lino.pdf](http://www.dma.uvigo.es/~aurea/Tema2_Lino.pdf)
- [http://campus.usal.es/~mpg/Personales/PersonalMAGL/Docencia/CalculoCATema5cTeoria\(09-10\).pdf](http://campus.usal.es/~mpg/Personales/PersonalMAGL/Docencia/CalculoCATema5cTeoria(09-10).pdf)