Pauta ayudantía 4

Algoritmos y complejidad

Diego Quezada; diego.quezadac@sansano.usm.cl

Kevin Lagos; kevin.lagos@sansano.usm.cl

Ejercicio 1

Obtenga la expresión del error para la regla del trapecio en un intervalo $[x_0, x_1]$.

Solución

Sabemos que la regla del trapecio utiliza un polinomio de grado 1. Por *Lagrange* lo obtenemos facilmente:

$$P(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1)$$

Sabemos que al interpolar $f(x)$ en $[x_0, x_1]$ el error vendrá dado por:

$$f(x) - P(x) = \frac{(x - x_0)(x - x_1)}{2!} f^{(2)}(c)$$

Así, tenemos que:

$$f(x) = \frac{x - x_1}{x_0 - x_1} f(x_0) + \frac{x - x_0}{x_1 - x_0} f(x_1) + \frac{(x - x_0)(x - x_1)}{2!} f^{(2)}(c)$$

Integrando esta última ecuación:

$$\int_{x_0}^{x_1} f(x) = \int_{x_0}^{x_1} \frac{x - x_1}{x_0 - x_1} f(x_0) dx + \int_{x_0}^{x_1} \frac{x - x_0}{x_1 - x_0} f(x_1) dx + \int_{x_0}^{x_1} \frac{(x - x_0)(x - x_1)}{2!} f^{(2)}(c) dx$$

Sabemos que la última integral es la que nos indica el error en la integración. Enfocaremos nuestro trabajo en ella:

$$\int_{x_0}^{x_1} \frac{(x - x_0)(x - x_1)}{2!} f^{(2)}(c) = \frac{1}{2} f^{(2)}(c) \int_{x_0}^{x_1} (x - x_0)(x - x_1) dx$$

Luego:

$$\frac{1}{2} f^{(2)}(c) \int_{x_0}^{x_1} x^2 - (x_0 + x_1)x + x_0 x_1 \, dx = \frac{1}{2} f^{(2)}(c) (\frac{x^3}{3} - (x_0 + x_1)\frac{x^2}{2} + x_0 x_1 x)\Big|_{x_0}^{x_1}$$

Al evaluar en los límites de integración y restar tenemos

$$\frac{1}{2} f^{(2)}(c) (\frac{x_1^3}{3} - (x_0 + x_1)\frac{x_1^2}{2} + x_0 x_1^2 - \frac{x_0^3}{3} + (x_0 + x_1)\frac{x_0^2}{2} - x_0^2 x_1)$$

Agrupando los términos cúbicos:

$$\frac{1}{2} f^{(2)}(c) (\frac{x_1^3}{3} - \frac{x_1^3}{2} + x_0 x_1^2 - \frac{x_0 x_1^2}{2} + \frac{x_0^2 x_1}{2} - x_0^2 x_1 - \frac{x_0^3}{3} + \frac{x_0^3}{2})$$

Sumamos:

$$\frac{1}{2} f^{(2)}(c) (-\frac{x_1^3}{6} + \frac{x_0 x_1^2}{2} - \frac{x_0^2 x_1}{2} + \frac{x_0^3}{6})$$

Si ordenamos y factorizamos por $1/6$ estamos listos:

$$\frac{1}{12} f^{(2)}(c) (x_0^3 - 3x_0^2 x_1 + 3x_0 x_1^2 - x_1^3) = -\frac{1}{12} f^{(2)}(c) (x_1 - x_0)^2 = -\frac{1}{12} f^{(2)}(c) \cdot h^2$$

Ejercicio 2

Obtenga la regla de cuadratura Gaussiana para el cálculo de $\int_{-1}^{1} f(x)dx$ con dos puntos de cuadratura.

Solución

Buscamos valores w_1, x_1, w_2, x_2 tal que:

$$\begin{aligned} w_1 + w_2 &= \int_{-1}^{1} dx = 2 \\ w_1x_1 + w_2x_2 &= \int_{-1}^{1} xdx = 0 \\ w_1x_1^2 + w_2x_2^2 &= \int_{-1}^{1} x^2dx = \frac{2}{3} \\ w_1x_1^2 + w_2x_2^2 &= \int_{-1}^{1} x^3dx = 0 \end{aligned}$$

De la primera ecuación obtenemos $w_2 = 2 - w_1$ facilmente. Al ingresar esto en la segunda ecuación obtenemos:

$$w_1x_1 = (2 - w_1)x_2 = 0 \to x_2 = -\frac{w_1x_1}{2 - w_1}$$

Utilizemos ahora nuestras expresiones para w_2 y x_2 para ingresarlas en la tercera ecuación:

$$w_1x_1^2 + (2 - w_1)(\frac{-w_1x_1}{2 - w_1})^2 = \frac{2}{3} \to x_1^2(w_1 + \frac{w_1^2}{2 - w_1}) = x_1^2(\frac{2w_1}{2 - w_1}) = \frac{2}{3}$$

De esta última ecuación obtenemos $x_1^2 = \frac{2 - w_1}{3w_1}$.

Solo nos queda ingresar todo a la última ecuación:

$$w_1x_1\frac{2 - w_1}{3w_1} + (2 - w_1)(-\frac{w_1x_1}{2 - w_1})^3 = x_1\frac{2 - w_1}{3} - \frac{w_1^3x_1}{(2 - w_1)^2} \cdot \frac{2 - w_1}{3w_1} = 0$$

Arreglando un poco la última expresión obtenemos:

$$x_1\frac{2 - w_1}{3} - \frac{w_1^2x_1}{3(2 - w_1)} = x_1(2 - w_1)^2 - w_1^2x_1 = x_1(4 - 4w_1 + w_1^2 - w_1^2) = x_1(4 - 4w_1) = 0$$

De esta última expresión obtenemos $w_1 = 1$.

Como $w_2 = 2 - w_1$ obtenemos $w_2 = 1$.

Como $x_1^2 = \frac{2 - w_1}{3w_1}$ obtenemos $x_1^2 = \frac{1}{3}$ lo que implica $x_1 = \sqrt{\frac{1}{3}}$.

Como $x_2 = -\frac{w_1x_1}{2-w_1}$ obtenemos $x_2 = -\sqrt{\frac{1}{3}}$.

Por lo que obtenemos la siguiente fórmula de cuadratura Gaussiana para cualquier función $f(x)$:

$$\int_{-1}^{1} f(x)dx = f(\sqrt{\frac{1}{3}}) + f(-\sqrt{\frac{1}{3}})$$

Ejercicio 3

Solucione con lo obtenido en 2: $\int_{-1}^{1} \frac{dx}{3+x}$:

Solución

Simplemente debemos evaluar:

$$\int_{-1}^{1} \frac{dx}{3+x} = \frac{1}{3+\sqrt{\frac{1}{3}}} + \frac{1}{3-\sqrt{\frac{1}{3}}} \approx 0.6923076923076923$$

El valor fue obtenido con Python.

Sabemos que el resultado real de la integral pedida es $\ln(2)$. Si calculamos el error con los valores de Python obtenemos un error de 0.0008394882522529956.

Notar: La fórmula encontrada en el ejercicio 2 es para "cualquier" función . En el siguiente enlace puede encontrar los valores de w_i y x_i para más puntos de cuadratura: https://pomax.github.io/bezierinfo/legendre-gauss.html.

Al utilizar 5 puntos de cuadratura obtenemos: 0.6931471578530402 con un error: $2.270690513395124 \cdot 10^{-8}$.

```
import numpy as np

def f(x):
    return 1/(3 + x)

w1 = 0.5688888888888889
w2 = 0.4786286704993665
w3 = 0.4786286704993665
w4 = 0.2369268850561891
w5 = 0.2369268850561891
x1 = 0
x2 = -0.5384693101056831
x3 = 0.5384693101056831
x4 = -0.9061798459386640
x5 = 0.9061798459386640

approx = w1*f(x1) + w2*f(x2) +w3*f(x3) +w4*f(x4) +w5*f(x5)

print(approx)
print(np.log(2) - approx)
```

Ejercicio 4

Programe la regla del Rectángulo, Trapecio y Simpson:

Solución

```
import numpy as np
import scipy.integrate as integrate

def rectangle_rule(f,a,b,n): # n + 1 puntos
    h = (b - a) / n
    suma_imagenes = 0
    x = a
    for i in range(0, n + 1): # i desde 0 hasta n
        suma_imagenes = suma_imagenes + f(x)
        x = x + h # siguiente x
    return h * suma_imagenes

def trapezoidal_rule(f,a,b,n): # n + 1 puntos
    h = (b - a) / n
    suma_imagenes = 0
    x = a
    for i in range(0, n + 1): # i desde 0 hasta n
        if (i == 0 or i == n): # f(a) y f(b)
            suma_imagenes = suma_imagenes + f(x)
        else:
            suma_imagenes = suma_imagenes + 2 * f(x)
        x = x + h # siguiente x
    return (h/2) * suma_imagenes

def simpson_rule(f,a,b,n): # n + 1 puntos
    h = (b - a) / n
    suma_imagenes = 0
    x = a
    for i in range(0, n + 1): # i desde 0 hasta n
        if (i == 0 or i == n): # f(a) y f(b)
            suma_imagenes = suma_imagenes + f(x)
        else:
            if(i % 2 == 0):
                suma_imagenes = suma_imagenes + 2 * f(x)
            else:
                suma_imagenes = suma_imagenes + 4 * f(x)
        x = x + h # siguiente x
    return (h/3) * suma_imagenes

f,a,b,n = (np.exp, 1, 1.5, 100)
rectangle = rectangle_rule(f, a, b, n)
trapezoidal = trapezoidal_rule(f, a, b, n)
simpson = simpson_rule(f, a, b, n)
real_value = integrate.quad(f, a, b)[0] # retorna una tupla (value, error)

print("Rectangle rule error: " + str(abs(real_value - rectangle)))
print("Trapezoidal rule error: " + str(abs(real_value - trapezoidal)))
print("Simpson rule error: " + str(abs(real_value - simpson)))
```