

Tema 2 - Operaciones matemáticas básicas

Integración numérica II

M. en C. Gustavo Contreras Mayén

27 de septiembre de 2014

1 Reglas de Simpson

- Regla de $1/3$ de Simpson
- Regla compuesta de $1/3$ de Simpson
- Regla de $3/8$ de Simpson

1 Reglas de Simpson

- Regla de $1/3$ de Simpson
- Regla compuesta de $1/3$ de Simpson
- Regla de $3/8$ de Simpson

2 Integración de Romberg

1 Reglas de Simpson

- Regla de $1/3$ de Simpson
- Regla compuesta de $1/3$ de Simpson
- Regla de $3/8$ de Simpson

2 Integración de Romberg

3 Librería Scipy

- Integración (`scipy.integrate`)

1 Reglas de Simpson

- Regla de $1/3$ de Simpson
- Regla compuesta de $1/3$ de Simpson
- Regla de $3/8$ de Simpson

2 Integración de Romberg

3 Librería Scipy

- Integración (`scipy.integrate`)

4 Cuadraturas Gaussianas

- Polinomios ortogonales
- Deduciendo las abscisas nodales y los pesos

1 Reglas de Simpson

- Regla de $1/3$ de Simpson
- Regla compuesta de $1/3$ de Simpson
- Regla de $3/8$ de Simpson

2 Integración de Romberg

3 Librería Scipy

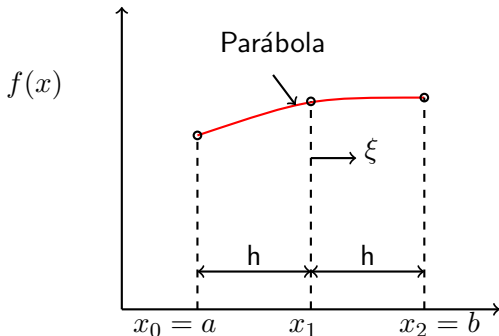
- Integración (`scipy.integrate`)

4 Cuadraturas Gaussianas

- Polinomios ortogonales
- Deduciendo las abscisas nodales y los pesos

Regla de 1/3 de Simpson

La regla de 1/3 de Simpson se obtiene de las fórmulas de Newton-Cotes con $n = 2$, es decir, haciendo una interpolación con una parábola a través de tres nodos, como se muestra en la siguiente figura:

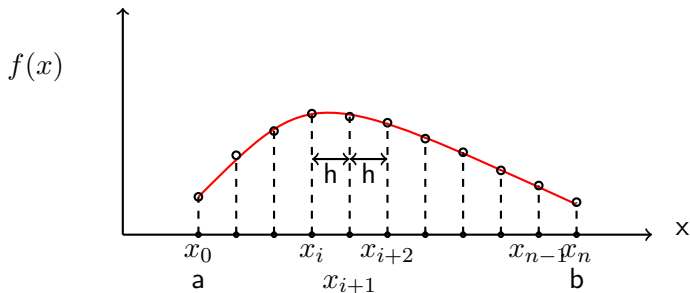


El área debajo de la curva representa una aproximación a la integral $\int_a^b f(x)$:

$$I = \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \frac{h}{3}$$

Regla compuesta de 1/3 de Simpson

Para obtener la regla compuesta de 1/3 de Simpson, se divide el intervalo de integración $[a, b]$ en n bloques (n par) de ancho $h = (b - a)/n$



Aplicando la fórmula anterior a dos bloques adyacentes, tenemos:

$$\int_{x_i}^{x_{i+2}} f(x)dx \simeq [f(x_i) + 4f(x_{i+1}) + f(x_{i+2})] \frac{h}{3}$$

sustituyendo la ecuación a todo el intervalo

$$\int_a^b f(x)dx = \int_{x_0}^{x_m} f(x)dx = \sum_{i=0,2,\dots}^n \left[\int_{x_i}^{x_{i+2}} f(x)dx \right]$$

Por lo que

$$\int_a^b f(x)dx \simeq I = [f(x_0) + 4f(x_1) + 2f(x_2) + \dots \\ \dots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)] \frac{h}{3}$$

es quizás el método más conocido de integración numérica. Aunque su reputación es algo inmerecido, ya que la regla del trapecio es más robusta, y la integración de Romberg es más eficiente.

El error en la regla de 1/3 de Simpson

El error en la regla compuesta de Simpson viene dado por:

$$E = \frac{(b-a)h^4}{180} f^{(4)}(\xi)$$

de donde inferimos que la integral obtenida por el método, es exacta si el polinomio es de grado tres o menor.

Regla de 3/8 de Simpson

La regla de 1/3 de Simpson necesita que el número de bloques n sea par.

Si la condición no se cumple, podemos integrar sobre los primeros (o últimos) tres bloques con la regla de 3/8 de Simpson:

$$I = [f(x_0) + 3f(x_1) + 3f(x_2) + f(x_3)] \frac{3h}{8}$$

y aplicar la regla de 1/3 de Simpson en los bloques restantes.

Ejemplo

Estimar la integral $\int_0^{2.5} f(x)dx$ a partir de los siguientes datos:

x	0	0.5	1.0	1.5	2.0	2.5
$f(x)$	1.5000	2.0000	2.0000	1.6364	1.25000	0.9565

Usaremos las reglas de Simpson:

- ➊ Dado que el número de bloques es impar, calculamos la integral sobre los primeros tres bloques con la regla $3/8$ de Simpson.

Usaremos las reglas de Simpson:

- ➊ Dado que el número de bloques es impar, calculamos la integral sobre los primeros tres bloques con la regla $3/8$ de Simpson.
- ➋ Usamos la regla de $1/3$ de Simpson en los dos últimos bloques.

Usaremos las reglas de Simpson:

- ➊ Dado que el número de bloques es impar, calculamos la integral sobre los primeros tres bloques con la regla 3/8 de Simpson.
- ➋ Usamos la regla de 1/3 de Simpson en los dos últimos bloques.

$$I = [f(0) + 3f(0.5) + 3f(1.0) + f(1.5)] \frac{3(0.5)}{8}$$

Usaremos las reglas de Simpson:

- ➊ Dado que el número de bloques es impar, calculamos la integral sobre los primeros tres bloques con la regla 3/8 de Simpson.
- ➋ Usamos la regla de 1/3 de Simpson en los dos últimos bloques.

$$I = [f(0) + 3f(0.5) + 3f(1.0) + f(1.5)] \frac{3(0.5)}{8} \\ + [f(1.5) + 4f(2.0) + f(2.5)] \frac{0.5}{3}$$

Usaremos las reglas de Simpson:

- ➊ Dado que el número de bloques es impar, calculamos la integral sobre los primeros tres bloques con la regla 3/8 de Simpson.
- ➋ Usamos la regla de 1/3 de Simpson en los dos últimos bloques.

$$\begin{aligned} I &= [f(0) + 3f(0.5) + 3f(1.0) + f(1.5)] \frac{3(0.5)}{8} \\ &\quad + [f(1.5) + 4f(2.0) + f(2.5)] \frac{0.5}{3} \\ &= 2.8381 + 1.2655 = 4.1036 \end{aligned}$$

Evalúa la integral

$$\int_{-1}^1 \cos(2 \cos^{-1} x) dx$$

con la regla de Simpson de $1/3$ usando 2, 4 y 6 bloques. Explica tus resultados.

1 Reglas de Simpson

- Regla de $1/3$ de Simpson
- Regla compuesta de $1/3$ de Simpson
- Regla de $3/8$ de Simpson

2 Integración de Romberg

3 Librería Scipy

- Integración (`scipy.integrate`)

4 Cuadraturas Gaussianas

- Polinomios ortogonales
- Deduciendo las abscisas nodales y los pesos

Integración de Romberg

La integración de Romberg combina la regla del trapecio con la extrapolación de Richardson. Usemos la siguiente notación:

$$R_{i,1} = I_i$$

donde I_i representa el valor aproximado de $\int_a^b f(x)dx$ calculado con la regla recursiva del trapecio, usando 2^{i-1} bloques.

Recordemos que el error en esta aproximación es $E = c_1 h^2 + c_2 h^4 + \dots$ donde

$$h = \frac{b-a}{2^{i-1}}$$

es el ancho del bloque.

La integración de Romberg inicia con el cálculo de $R_{1,1} = I_1$ (un bloque) y $R_{2,1} = I_2$ (dos bloques) a partir de la regla del trapecio.

El término dominante $c_1 h^2$ es entonces eliminado por la extrapolación de Richardson. Usando $p = 2$ (el exponente en el término dominante) e indicando el resultado por $R_{2,2}$, tenemos:

$$R_{2,2} = \frac{2^2 R_{2,1} - R_{1,1}}{2^2 - 1} = \frac{4}{3} R_{2,1} - \frac{1}{3} R_{1,1}$$

Es conveniente guardar los resultados en un arreglo con la forma

$$\begin{bmatrix} R_{1,1} \\ R_{2,1} \quad R_{2,2} \end{bmatrix}$$

El siguiente paso es calcular $R_{3,1} = I_3$ (cuatro bloques) y repetir la extrapolación de Richardson con $R_{2,1}$ y $R_{3,1}$, guardando los resultados como $R_{3,2}$:

$$R_{3,2} = \frac{4}{3}R_{3,1} - \frac{1}{3}R_{2,1}$$

Los elementos del arreglo R calculados hasta el momento son:

$$\begin{bmatrix} R_{1,1} \\ R_{2,1} & R_{2,2} \\ R_{3,1} & R_{3,2} \end{bmatrix}$$

Los elementos de la segunda columna tienen un error del orden $c_2 h^4$, el cual puede ser eliminado con la extrapolación de Richardson.

Usando $p = 4$, obtenemos:

$$R_{3,3} = \frac{2^4 R_{3,2} - R_{2,2}}{2^4 - 1} = \frac{16}{15} R_{3,2} - \frac{1}{15} R_{2,2}$$

El resultado tiene ahora un error del orden $O(h^6)$. El arreglo se ha expandido ahora como

$$\begin{bmatrix} R_{1,1} \\ R_{2,1} & R_{2,2} \\ R_{3,1} & R_{3,2} & R_{3,3} \end{bmatrix}$$

Luego de otra ronda de cálculos, se tiene

$$\begin{bmatrix} R_{1,1} & & & \\ R_{2,1} & R_{2,2} & & \\ R_{3,1} & R_{3,2} & R_{3,3} & \\ R_{4,1} & R_{4,2} & R_{4,3} & R_{4,4} \end{bmatrix}$$

donde el error en $R_{4,4}$ es del orden de $O(h^8)$.

Nótese que la estimación con mayor precisión es siempre el último término de la diagonal.

Este proceso continua hasta que la diferencia entre dos términos sucesivos de la diagonal son lo suficientemente pequeños.

La fórmula general para la extrapolación es:

$$R_{i,j} = \frac{4^{j-1}R_{i,j-1} - R_{i-1,j-1}}{4^{j-1} - 1} \quad i > 1, \quad j = 2, 3, \dots, i$$

Esquemáticamente lo que tenemos es:

$$\begin{array}{ccccc}
 R_{i-1,j-1} & & & & \\
 & \searrow & & & \\
 & & \alpha & & \\
 R_{i,j-1} & \rightarrow & \beta & \xrightarrow{\quad} & R_{i,j}
 \end{array}$$

Donde los multiplicadores α y β dependen de j de la siguiente manera:

j	2	3	4	5	6
α	$-1/3$	$-1/15$	$-1/63$	$-1/255$	$-1/1023$
β	$4/3$	$16/15$	$64/63$	$256/255$	$1024/1023$

El arreglo triangular es conveniente para manipularlo computacionalmente hablando. La aplicación del algoritmo de Romberg puede llevarse dentro de una matriz de una dimensión.

Luego de la primera extrapolación, $R_{1,1}$ ya no se ocupa de nuevo, por lo que podemos re-emplazarla con $R_{2,2}$, por tanto, tenemos en el arreglo

$$\begin{bmatrix} R'_1 = R_{2,2} \\ R'_2 = R_{2,1} \end{bmatrix}$$

En la segunda extrapolación, $R_{3,2}$ sobre-escribe a $R_{2,1}$ y $R_{3,3}$ re-emplaza a $R_{2,2}$, entonces el arreglo queda

$$\begin{bmatrix} R'_1 = R_{3,3} \\ R'_2 = R_{3,2} \\ R'_3 = R_{3,1} \end{bmatrix}$$

Y así podemos continuar. R'_1 contiene siempre el mejor resultado. La fórmula de extrapolación para el k -ésima vuelta, es:

$$R'_j = \frac{4^{k-j} R'_{j+1} - R'_j}{4^{k-j} - 1}, \quad j = k - 1, k - 2, \dots, 1$$

Ejemplo

Usando la integración de Romberg, evalúa

$$\int_0^{\pi} f(x) dx$$

donde $f(x) = \sin(x)$

Primera parte: Regla del trapecio recursiva

$$R_{1,1} = I(\pi) = \frac{\pi}{2}[f(0) + f(\pi)] = 0$$

Primera parte: Regla del trapecio recursiva

$$R_{1,1} = I(\pi) = \frac{\pi}{2}[f(0) + f(\pi)] = 0$$

$$R_{2,1} = I\left(\frac{\pi}{2}\right) = \frac{1}{2}I(\pi) + \frac{\pi}{2}f\left(\frac{\pi}{2}\right) = 1.5708$$

Primera parte: Regla del trapecio recursiva

$$R_{1,1} = I(\pi) = \frac{\pi}{2}[f(0) + f(\pi)] = 0$$

$$R_{2,1} = I\left(\frac{\pi}{2}\right) = \frac{1}{2}I(\pi) + \frac{\pi}{2}f\left(\frac{\pi}{2}\right) = 1.5708$$

$$R_{3,1} = I\left(\frac{\pi}{4}\right) = \frac{1}{2}I\left(\frac{\pi}{2}\right) + \frac{\pi}{4}\left[f\left(\frac{\pi}{4}\right) + f\left(\frac{3\pi}{4}\right)\right] = 1.8961$$

Primera parte: Regla del trapecio recursiva

$$R_{1,1} = I(\pi) = \frac{\pi}{2}[f(0) + f(\pi)] = 0$$

$$R_{2,1} = I\left(\frac{\pi}{2}\right) = \frac{1}{2}I(\pi) + \frac{\pi}{2}f\left(\frac{\pi}{2}\right) = 1.5708$$

$$R_{3,1} = I\left(\frac{\pi}{4}\right) = \frac{1}{2}I\left(\frac{\pi}{2}\right) + \frac{\pi}{4}\left[f\left(\frac{\pi}{4}\right) + f\left(\frac{3\pi}{4}\right)\right] = 1.8961$$

$$R_{4,1} = I\left(\frac{\pi}{8}\right) = \frac{1}{2}I\left(\frac{\pi}{4}\right) + \frac{\pi}{8}\left[f\left(\frac{\pi}{8}\right) + f\left(\frac{3\pi}{8}\right) + f\left(\frac{5\pi}{8}\right) + f\left(\frac{7\pi}{8}\right)\right] = 1.9742$$

Segunda parte: Extrapolación de Richardson

Usando las fórmulas de extrapolación, construimos la siguiente tabla:

$$\begin{bmatrix} R_{1,1} \\ R_{2,1} & R_{2,2} \\ R_{3,1} & R_{3,2} & R_{3,3} \\ R_{4,1} & R_{4,2} & R_{4,3} & R_{4,4} \end{bmatrix} = \begin{bmatrix} 0 \\ 1.5708 & 2.0944 \\ 1.8961 & 2.0046 & 1.9986 \\ 1.9742 & 2.0003 & 2.0000 & 2.0000 \end{bmatrix}$$

De acuerdo al procedimiento, vemos que converge, por tanto $\int_0^\pi \sin(x)dx = R_{4,4} = 2.0000$, que es el resultado exacto.

Usando la integración de Romberg, evalúa la siguiente integral:

$$\int_0^{\sqrt{\pi}} 2x^2 \cos(x^2) dx$$

Aquí hay dos caminos:

- 1 Elaborar un código completo para resolver el problema.

Usando la integración de Romberg, evalúa la siguiente integral:

$$\int_0^{\sqrt{\pi}} 2x^2 \cos(x^2) dx$$

Aquí hay dos caminos:

- 1 Elaborar un código completo para resolver el problema.
- 2 Apoyarnos en las ventajas que nos da Python.

Camino 1

```
1 from numpy import zeros
2 from math import *
3
4 def trapecio(f,a,b,lviejo,k):
5     if k == 1: lnueva = (f(a) + f(b))*(b - a)
6         /2.0
7     else:
8         n = 2**((k - 2))
9         h = (b - a)/n
10        x = a + h/2.0
11        sum = 0.0
12        for i in range(n):
13            sum = sum + f(x)
14            x = x + h
15        lnueva = (lviejo + h*sum)/2.0
16    return lnueva
```



```

1 def romberg(f,a,b,tol=1.0e-6):
2     def richardson(r,k):
3         for j in range(k-1,0,-1):
4             const = 4.0**(k-j)
5             r[j] = (const*r[j+1] - r[j])/(
6                 const - 1.0)
7
8         return r
9
10    r = zeros(21)
11    r[1] = trapecio(f,a,b,0.0,1)
12    r_viejo = r[1]
13    for k in range(2,21):
14        r[k] = trapecio(f,a,b,r[k-1],k)
15        r = richardson(r,k)
16        if abs(r[1]-r_viejo) < tol*max(abs(r
17            [1]),1.0):
18            return r[1], 2**(k-1)
19        r_viejo = r[1]
20
21    print 'La integracion de Romberg no

```

```
1 def f(x): return 2.0*(x**2)*cos(x**2)
2
3 I,n = romberg(f,0,sqrt(pi))
4 print 'Integral = ', I
5 print ' nBloques = ',n
```

El código anterior nos daría el resultado esperado. Veamos ahora la segunda manera de resolver el problema, explotando al máximo Python.

1 Reglas de Simpson

- Regla de $1/3$ de Simpson
- Regla compuesta de $1/3$ de Simpson
- Regla de $3/8$ de Simpson

2 Integración de Romberg

3 Librería Scipy

- Integración (`scipy.integrate`)

4 Cuadraturas Gaussianas

- Polinomios ortogonales
- Deduciendo las abscisas nodales y los pesos

Usando la función `romberg` de `scipy`

En la presentación anterior ya hemos presentado de la librería `scipy` el uso de la función `quad` para integrar una función.

De la misma librería, tomamos la función `romberg`, que realiza una integración con la técnica de Romberg.

Integración (`scipy.integrate`)

El subpaquete `scipy.integrate` proporciona varias técnicas de integración.

<code>quad</code>	Integración en general.
<code>dblquad</code>	Integración doble en general.
<code>tplquad</code>	Integración triple en general.
<code>fixed-quad</code>	Integración de $f(x)$ usando cuadraturas gaussianas de orden n .
<code>quadrature</code>	Integra con tolerancia dada usando cuadratura gaussiana.
<code>romberg</code>	Integra una función mediante la integración de Romberg.

scipy.integrate.romberg

```
scipy.integrate.romberg(function, a, b, show=False)
```

Es la integración de Romberg de una función.

Devuelve la integral de una función (función de una variable) en el intervalo $[a, b]$.

Si `show` es 1, se muestra el arreglo triangular de resultados intermedios.

Código con scipy

```
1 from scipy import *
2 from scipy.integrate import romberg
3
4
5 def f(x): return 2.0*(x**2)*cos(x**2)
6
7 resultado = romberg(f,0,sqrt(pi),show=True)
```

Tabla de resultados

Steps	StepSize	Results							
1	1.772454	-5.568328							
2	0.886227	-1.799813	-0.543642						
4	0.443113	-1.034769	-0.779755	-0.795496					
8	0.221557	-0.925214	-0.888695	-0.895958	-0.897553				
16	0.110778	-0.902166	-0.894484	-0.894870	-0.894852	-0.894842			
32	0.055389	-0.896649	-0.894810	-0.894832	-0.894831	-0.894831	-0.894831		
64	0.027695	-0.895285	-0.894830	-0.894831	-0.894831	-0.894831	-0.894831	-0.894831	-0.894831
128	0.013847	-0.894945	-0.894831	-0.894831	-0.894831	-0.894831	-0.894831	-0.894831	-0.894831

The final result is -0.894831469484 after 129 function evaluations.

Ejercicio de clase

Evalúa la siguiente integral con el procedimiento de Romberg:

$$\int_0^{\frac{\pi}{4}} \frac{dx}{\sqrt{\sin x}}$$

Vemos que la integral es impropia, por lo que hay que manejarla de tal manera que se remueva la singularidad, en este caso, mediante un cambio de variable, para luego usar `scipy.integrate.romberg` con los respectivos límites de integración.

Evalúa la integral

$$\int_0^2 (x^5 + 3x^3 - 2)dx$$

por el método de integración de Romberg.

1 Reglas de Simpson

- Regla de $1/3$ de Simpson
- Regla compuesta de $1/3$ de Simpson
- Regla de $3/8$ de Simpson

2 Integración de Romberg

3 Librería Scipy

- Integración (`scipy.integrate`)

4 Cuadraturas Gaussianas

- Polinomios ortogonales
- Deduciendo las abscisas nodales y los pesos

Hemos visto que las fórmulas de Newton-Cotes para aproximar la integral

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

trabajan muy bien si $f(x)$ es una función suave, como los polinomios.

También aplica para las cuadraturas Gaussianas, ya que son buenas para estimar integrales de la forma:

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

donde $w(x)$ se denomina *función de ponderación* que puede contener singularidades, siempre y cuando sean integrables.

Un ejemplo de este tipo, es la integral

$$\int_0^1 (1+x^2) \ln x dx$$

Para cuando tenemos límites de integración infinitos

$$\int_0^{\infty} \exp(-x) \sin x dx$$

éstos se pueden reacomodar para calcular la integral.

Fórmulas de integración Gaussianas

Las fórmulas de integración Gaussianas tiene la misma forma de las reglas de Newton-Cotes:

$$I = \sum_{i=0}^n A_i f(x_i)$$

donde I representa la aproximación al valor de la integral, la diferencia radica en la forma en que se determinan los pesos A_i y abscisas nodales x_i .

En la integración de Newton-Cotes los nodos se espacian uniformemente en (a, b) , es decir, estaban ya predeterminadas sus ubicaciones. En la cuadratura de Gauss, se eligen los nodos y los pesos de modo que la ecuación para I , se obtiene la integral exacta si $f(x)$ es un polinomio de grado $2n + 1$ o menor, es decir:

$$\int_a^b w(x)P_m(x)dx = \sum_{i=0}^n A_i P_m(x_i), \quad m \leq 2n + 1$$

$$\int_a^b w(x)P_m(x)dx = \sum_{i=0}^n A_i P_m(x_i), \quad m \leq 2n + 1$$

Una manera de determinar los pesos y las abscisas es sustituir $P_0(x) = 1$, $P_1(x) = x, \dots$, $P_{2n+1}(x) = x^{2n+1}$ en la ecuación anterior y resolver el sistema de $2n + 2$ ecuaciones

$$\int_a^b w(x)x^j dx = \sum_{i=0}^n A_i x_i^j, \quad j = 0, 1, \dots, 2n + 1$$

para las incógnitas A_i y x_i .

Como ejemplo veamos: sea $w(x) = \exp(-x)$, $a = 0$, $b = \infty$ y $n = 1$. Las cuatro ecuaciones que determinan x_0 , x_1 , A_0 y A_1 son:

$$\int_0^{\infty} \exp(-x) dx = A_0 + A_1$$

Como ejemplo veamos: sea $w(x) = \exp(-x)$, $a = 0$, $b = \infty$ y $n = 1$. Las cuatro ecuaciones que determinan x_0 , x_1 , A_0 y A_1 son:

$$\int_0^{\infty} \exp(-x) dx = A_0 + A_1$$
$$\int_0^{\infty} \exp(-x) x dx = A_0 x_0 + A_1 x_1$$

Como ejemplo veamos: sea $w(x) = \exp(-x)$, $a = 0$, $b = \infty$ y $n = 1$. Las cuatro ecuaciones que determinan x_0 , x_1 , A_0 y A_1 son:

$$\begin{aligned}\int_0^{\infty} \exp(-x) dx &= A_0 + A_1 \\ \int_0^{\infty} \exp(-x) x dx &= A_0 x_0 + A_1 x_1 \\ \int_0^{\infty} \exp(-x) x^2 dx &= A_0 x_0^2 + A_1 x_1^2\end{aligned}$$

Como ejemplo veamos: sea $w(x) = \exp(-x)$, $a = 0$, $b = \infty$ y $n = 1$. Las cuatro ecuaciones que determinan x_0 , x_1 , A_0 y A_1 son:

$$\begin{aligned}\int_0^{\infty} \exp(-x) dx &= A_0 + A_1 \\ \int_0^{\infty} \exp(-x) x dx &= A_0 x_0 + A_1 x_1 \\ \int_0^{\infty} \exp(-x) x^2 dx &= A_0 x_0^2 + A_1 x_1^2 \\ \int_0^{\infty} \exp(-x) x^3 dx &= A_0 x_0^3 + A_1 x_1^3\end{aligned}$$

Evaluando las integrales, obtenemos

$$A_0 + A_1 = 1$$

$$A_0x_0 + A_1x_1 = 1$$

$$A_0x_0^2 + A_1x_1^2 = 2$$

$$A_0x_0^3 + A_1x_1^3 = 6$$

Cuya solución es:

$$x_0 = 2 - \sqrt{2} \quad A_0 \frac{\sqrt{2} + 1}{2\sqrt{2}}$$

$$x_1 = 2 + \sqrt{2} \quad A_1 \frac{\sqrt{2} - 1}{2\sqrt{2}}$$

Por tanto la fórmula de integración obtenida es:

$$\int_0^{\infty} \exp(-x) f(x) dx \simeq \frac{1}{2\sqrt{2}} \left[\left(\sqrt{2} + 1 \right) f \left(2 - \sqrt{2} \right) + \left(\sqrt{2} - 1 \right) f \left(2 + \sqrt{2} \right) \right]$$

Debido a la no linealidad de las ecuaciones, este enfoque no va a funcionar bien para valores grandes de n .

Hay métodos prácticos para calcular x_i y A_i que requieren un cierto conocimiento de los polinomios ortogonales y su relación con la cuadratura de Gauss. Hay, sin embargo, varias fórmulas "clásicas" de integración Gaussianas para las cuales, las abscisas y pesos han sido calculados con gran precisión y tabulados.

Estas fórmulas se pueden utilizar sin conocer la teoría detrás de ellas, ya que todo lo que uno necesita para la integración de Gauss son los valores de x_i y A_i .

Polinomios ortogonales

Los polinomios ortogonales se utilizan en muchas áreas de la matemática y del análisis numérico; se han estudiado a fondo y muchas de sus propiedades ya son conocidas.

Los polinomios $\varphi_n(x)$, con $n = 0, 1, 2, \dots$ (n es el grado del polinomio) se dice que forman un conjunto ortogonal en el intervalo (a, b) con respecto a la función de ponderación $w(x)$ si

$$\int_a^b w(x) \varphi_m(x) \varphi_n(x) dx = 0, \quad m \neq n$$

El conjunto se determina, con excepción de un factor constante, por la elección de la función de ponderación y los límites de integración. Es decir, cada conjunto de polinomios ortogonales se asocia con ciertos $w(x)$, a y b . El factor constante se especifica de manera estandarizada.

A continuación se enlistan algunos de los polinomios ortogonales clásicos, la última columna indica la estandarización usada.

Polinomios ortogonales

Nombre	Símbolo	a	b	$w(x)$	$\int_a^b [\varphi_n(x)]^2 dx$
Legendre	$p_n(x)$	-1	1	1	$2/(2n+1)$
Chebyshev	$T_n(x)$	-1	1	$(1-x^2)^{-1/2}$	$\pi/2 \quad (n > 0)$
Laguerre	$L_n(x)$	0	∞	e^{-x}	1
Hermite	$H_n(x)$	$-\infty$	∞	e^{x^2}	$\sqrt{\pi} 2^n n!$

Los polinomios ortogonales cumplen relaciones de recurrencia de la forma

$$a_n \varphi_{n+1}(x) = (b_n + c_n x) \varphi_n(x) - d_n \varphi_{n-1}(x)$$

Si los dos primeros polinomios del conjunto se conocen, los otros elementos del conjunto pueden calcularse de la ecuación anterior. Los coeficientes en la fórmula de recurrencia, junto con $\varphi_0(x)$ y $\varphi_1(x)$ son:

Nombre	$\varphi_0(x)$	$\varphi_1(x)$	a_n	b_n	c_n	d_n
Legendre	1	x	$n + 1$	0	$2n + 1$	n
Chebyshev	1	x	1	0	2	1
Laguerre	1	$1 - x$	$n + 1$	$2n + 1$	-1	n
Hermite	1	$2x$	1	0	2	2

Otra manera para obtener los polinomios

Los polinomios ortogonales clásicos también se pueden obtener de las fórmulas:

$$p_n(x) = \frac{(-1)^n}{2^n n!} \frac{d^n}{dx^n} [(1 - x^2)^n]$$

$$T_n(x) = \cos(n \cos^{-1} x), \quad n > 0$$

$$L_n(x) = \frac{e^x}{n!} \frac{d^n}{dx^n} (x^n e^{-x})$$

$$H_n(x) = (-1)^n e^{x^2} \frac{d^n}{dx^n} (e^{-x^2})$$

Derivadas de los polinomios ortogonales

Las derivadas de los polinomios anteriores se pueden calcular de:

$$(1 - x^2)p'_n(x) = n[-xp_n(x) + p_{n-1}(x)]$$

$$(1 - x^2)T'_n(x) = n[-xT_n(x) + n p T n - 1(x)]$$

$$xL'_n(x) = n[L_n(x) - L_{n-1}(x)]$$

$$H'_n(x) = 2nH_{n-1}(x)$$

Propiedades de los polinomios

Algunas propiedades los polinomios ortogonales que son relevantes para la preceso de integración Gaussiana son:

- $\varphi(x)$ tiene n distintos ceros en el intervalo (a, b)
- Los ceros de $\varphi_n(x)$ están entre los ceros de $\varphi_{n+1}(x)$
- Cualquier polinomio $P_n(x)$ de grado n puede expresarse de la forma:

$$P_n(x) = \sum_{i=0}^n c_i \varphi_i(x)$$

- Se deduce de la ecuación anterior y de la propiedad de ortogonalidad que:

$$\int_a^b w(x) P_n(x) \varphi_{n+m}(x) dx = 0, \quad m \geq 0$$

Deduciendo las abscisas nodales y los pesos

Hay dos teoremas importantes que son de gran utilidad para apoyarnos y tomar sus resultados para la integración Gaussiana, la demostración es relativamente sencilla, pero no los demostraremos aquí, puede ser un buen ejercicio fuera de clase.

Teorema

Las abscisas nodales x_0, x_1, \dots, x_n son los ceros del polinomio $\varphi_{n+1}(x)$ que pertenece al conjunto ortogonal definido por

$$\int_a^b w(x) \varphi_m(x) \varphi_n(x) dx = 0, \quad m \neq n$$

Teorema

$$A_i = \int_a^b w(x) \mathcal{L}_i(x) dx, \quad i = 0, 1, \dots, n$$

donde $\mathcal{L}_i(x)$ son las funciones cardinales de Lagrange que abarcan los nodos x_0, x_1, \dots, x_n .

No es difícil calcular los ceros x_i , $i = 0, 1, \dots, n$ de un polinomio $\varphi_{n+1}(x)$ que pertenece a un conjunto ortogonal, podemos usar alguno de los métodos discutidos en la parte de cálculo de raíces.

Una vez conocidos los ceros, los pesos A_i , $i = 0, 1, \dots, n$ pueden calcularse de la ecuación anterior.

Fórmulas para calcular los pesos

Se puede demostrar que los pesos se pueden calcular a partir de:

$$\text{Gauss-Legendre } A_i = \frac{2}{(1 - x_i^2) [P'_{n+1}(x_i)]^2}$$

$$\text{Gauss-Laguerre } A_i = \frac{1}{x_i [L'_{n+1}(x_i)]^2}$$

$$\text{Gauss-Hermite } A_i = \frac{2^{n+2}(n+1)!\sqrt{\pi}}{[H'_{n+1}(x_i)]^2}$$

Abscisas y pesos para cuadraturas gaussianas

Vamos a mencionar la expresión para algunas fórmulas de integración por cuadraturas gaussianas.

La tabla de abscisas y pesos que se presenta a continuación, cubre para $n = 1$ a 5, y se redondea a seis decimales. Las operaciones con estos valores, se considera que funcionan bien si se hacen las cuentas a mano, en caso de requerir una mayor precisión o incluir un número mayor de nodos, será necesario usar la computadora.

Error en la cuadratura gaussiana

El error debido al truncamiento

$$E = \int_a^b w(x) f(x) dx - \sum_{i=0}^n A_i f(x_i)$$

es de la forma $E = K(n) f^{2n+2}(c)$, donde $a < c < b$, el valor de c no se conoce, solamente los extremos.

La expresión para $K(n)$ depende de la cuadratura en particular que se esté utilizando. Si las derivadas de $f(x)$ se pueden evaluar, el error de las fórmulas es útil para estimar el error en el intervalo.

Cuadratura de Gauss-Legendre

$$\int_{-1}^1 f(\xi) d\xi \approx \sum_{i=0}^n A_i f(\xi_i)$$

$\pm \xi_i$	A_i	$\pm \xi_{i1}$	A_i
$n = 1$		$n = 4$	
0.577350	1.000000	0.000000	0.568889
$n = 2$		0.538469	0.478629
0.000000	0.888889	0.906180	0.236927
0.774597	0.555556	$n = 5$	
$n = 3$		0.238619	0.467914
0.339981	0.652145	0.661209	0.360762
0.861136	0.347855	0.932470	0.171324

La cuadratura de Gauss-Legendre es la más utilizada. Nótese que los nodos están colocados simétricamente sobre $\xi = 0$, y los pesos asociados al par de nodos simétricos, son iguales, i.e. para $n = 1$, tenemos que $\xi_0 = -\xi_1$ y $A_0 = A_1$. El error de truncamiento está dado por:

$$E = \frac{2^{2n+3}[(n+1)!]^4}{(2n+3)[(2n+2)!]^3} f^{2n+2}(c), \quad -1 < c < 1$$

Para usar la cuadratura de Gauss-Legendre en la integral $\int_a^b f(x)dx$, primero hay que "mapear" el intervalo de integración (a, b) al intervalo "estándar" $(-1, 1)$, para ello, usamos la transformación

$$x = \frac{b+a}{2} + \frac{b-a}{2}\xi$$

Ahora $dx = d\xi(b-a)/2$, y la cuadratura toma la expresión

$$\int_a^b f(x)dx \approx \frac{b-a}{2} \sum_{i=0}^n A_i f(x_i)$$

El error por truncamiento, se expresa como

$$E = \frac{(b-a)^{2n+3}[(n+1)!]^4}{(2n+3)[(2n+2)!]^3} f^{(2n+2)}(c), \quad a < c < b$$

Ejercicio para el examen

Te pedimos que entregues una lista con los pesos y nodos para las siguientes cuadraturas:

1 Gauss-Chebyshev

$$\int_{-1}^1 (1 - x^2)^{-1/2} f(x) \approx \frac{\pi}{n + 1} \sum_{i=0}^n f(x_i)$$

2 Gauss-Laguerre

$$\int_0^{\infty} e^{-x} f(x) dx \approx \sum_{i=0}^n A_i f(x_i)$$

Ejercicio para el examen

1 Gauss-Hermite

$$\int_{-\infty}^{\infty} e^{-x^2} f(x) dx \approx \sum_{i=0}^n A_i f(x_i)$$

¿Aquí termina todo respecto a la integración numérica?

Como sabemos de los cursos de cálculo, podemos considerar ahora un proceso de integración para calcular integrales de funciones con dos y hasta tres variables, recurriendo a las técnicas mostradas.

¿Cómo construimos un algoritmo para ello?

El proceso no es complicado pero requiere una revisión cuidadosa, con Python podemos hacer el proceso más sencillo para calcular una integral doble o triple, pero no está demás que te las ingenies para desarrollar un código!!