Tema 2 - Operaciones matemáticas básicas

Integración numérica II

M. en C. Gustavo Contreras Mayén

4 de octubre de 2012

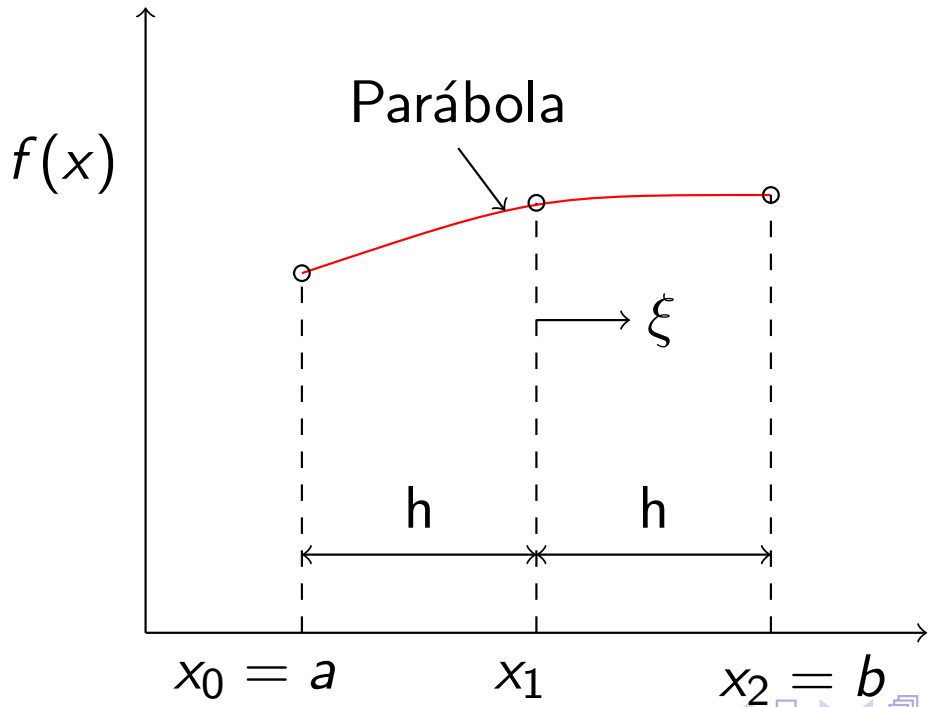
Contenido

Contenido

Contenido

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:

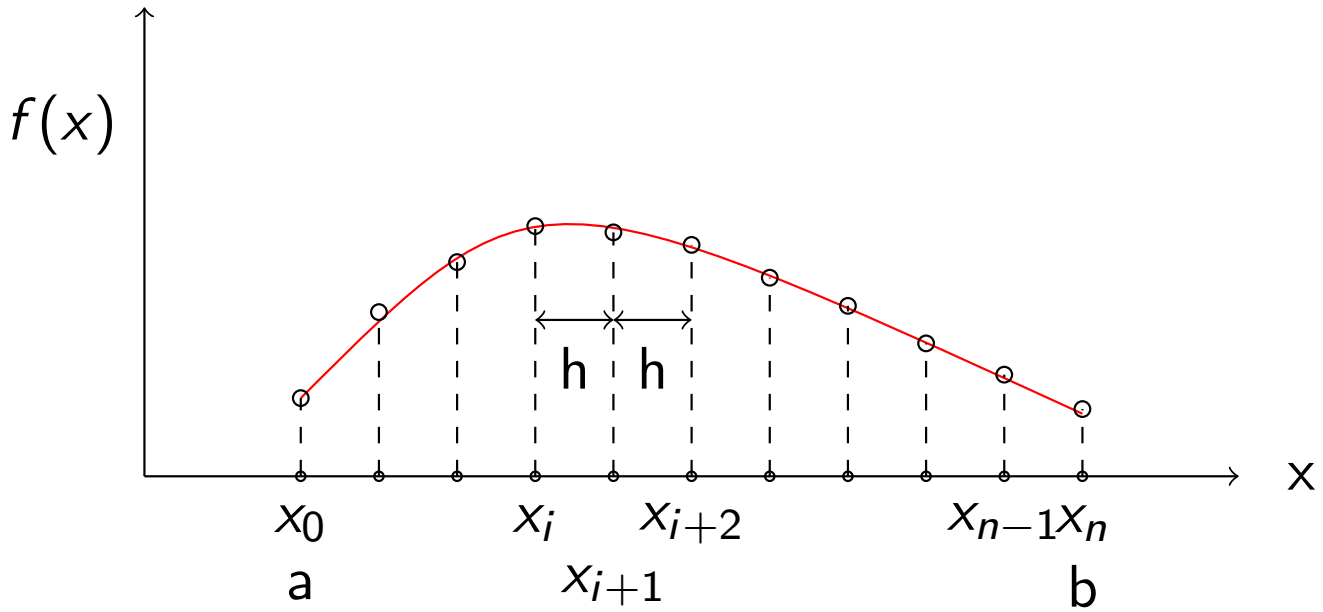


El área debajo de la curva representa una aproximación a la integral $\int_a^b f(x)$:

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

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

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

Por lo que

$$\int_a^b f(x)dx \simeq I = [f(x_0) + 4f(x_1) + 2f(x_2) + \ldots$$

$$\ldots + 2f(x_{n-2}) + 4f(x_{n-1}) + f(x_n)] \frac{h}{3}$$

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

Solución

Usaremos las reglas de Simpson:

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

Solución

Usaremos las reglas de Simpson:

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

Solución

Usaremos las reglas de Simpson:

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

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

Solución

Usaremos las reglas de Simpson:

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

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

Solución

Usaremos las reglas de Simpson:

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

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

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 $E = c_1 h^2 + c_2 h^4 + \ldots$ 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

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

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:

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

Esquemáticamente lo que tenemos es:

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

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:

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

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

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

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

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

Primera parte: Regla del trapecio recursiva

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

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

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

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

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.

Ejercicio

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.

Ejercicio

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,Iviejo,k):
5      if k == 1:Inueva = (f(a) + f(b))*(b - a)/2.0
6      else:
7          n = 2**(k - 2)
8          h = (b - a)/n
9          x = a + h/2.0
10         sum = 0.0
11         for i in range(n):
12             sum = sum + f(x)
13             x = x + h
14         Inueva = (Iviejo + h*sum)/2.0
15     return Inueva
```

```
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])/(const -
                   1.0)
6          return r
7
8      r = zeros(21)
9      r[1] = trapecio(f,a,b,0.0,1)
10     r_viejo = r[1]
11     for k in range(2,21):
12         r[k] = trapecio(f,a,b,r[k-1],k)
13         r = richardson(r,k)
14         if abs(r[1]-r_viejo) < tol*max(abs(r[1])
               ,1.0):
15             return r[1], 2**(k-1)
16             r_viejo = r[1]
17     print 'La integracion de Romberg no converge'
```

```
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.

Librería Scipy

SciPy es un conjunto de algoritmos matemáticos y funciones de conveniencia construidos en la extensión NumPy para Python.

Se agrega un poder significativo a la sesión interactiva de Python mediante la exposición del usuario a comandos de alto nivel y clases para la manipulación y visualización de datos.

Organización de Scipy

SciPy está organizada en sub-paquetes que cubren diferentes áreas de computación científica. Estos se resumen en la siguiente tabla:

Subpaquete	Descripción
cluster	Algortimos para clusters
constants	Constantes físicas y matemáticas
fftpack	Rutinas para la Transformada Rápida de Fourier
integrate	Integración y EDO
interpolate	Interpolación y uso de splines
io	Rutinas de entrada y salida

Subpaquete	Descripción
linalg	Algebra lineal
ndimage	Procesamiento N-dimensional de imagenes
odr	Regresión de distancias ortogonales
optimize	Optimizació y rutinas para encontrar raíces
signal	Procesamiento de señales
sparse	Matrices sparse y rutinas asociadas
spatial	Estructura de datos espaciales
special	Funciones especiales
stats	Distribuciones estadísticas
weave	Integración con C/C++

Integración (scipy.integrate)

El subpaquete scipy.integrate proporciona varias técnicas de integración.

quad	Integración en general.
dblquad	Integración doble en general.
tplquad	Integración triple en general.
fixed-quad	Integración de f(x) usando cuadraturas gaussianas de orden n.
quadrature	Integra con tolerancia dada usando cuadratura gaussiana.
romberg	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
128	0.013847	-0.894945	-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.