

# Laboratorio 5: Integración numérica

El objetivo de este laboratorio es aprender técnicas de integración numérica, así como el uso correcto de funciones como parámetros.

En muchas ocasiones resulta sumamente útil poder pasar una función como parámetro. Un caso típico es cuando se quiere escribir un programa para calcular la integral de una función cualquiera mediante una regla de integración numérica particular. Así, por ejemplo, si se escribe un programa Matlab llamado *trap* que calcula la aproximación de la integral de una función dada en un intervalo genérico  $[a, b]$  por la regla de los trapecios con  $N$  subintervalos, uno querrá poder llamar a ese programa con sentencias como las siguientes:

```
trap(f,0,1,10) % Calcula la integral de f(x) en [0,1] con 10 subintervalos.
trap(g,-1,1,16) % Calcula la integral de g(x) en [-1,1] con 16 subintervalos.
trap('sin',0,pi,8) % Calcula la integral de sen(x) en [0,pi] con 8 subintervalos.
```

Para poder hacer esto, el programa *trap* debe recibir la función pasada como parámetro ( $f(x)$ ,  $g(x)$ , o  $\sin(x)$ , respectivamente) en una variable alfabética (es decir una variable cuyos valores son nombres y no números) y, cada vez que haga falta evaluar esa función, debe utilizarse el comando *feval*, cuya sintaxis es como en el ejemplo siguiente:

```
function int=trap(funcnt,a,b,N)
...
y=feval(funcnt,a)
```

En este ejemplo, *funcnt* es la variable alfabética que contiene el nombre de la función que se quiere evaluar, *a* es el valor de la variable donde se quiere evaluar la función e y devuelve el valor calculado.

Por otra parte, debe haber un programa *function* (*f.m* en el primer ejemplo o *g.m* en el segundo) en el que se defina la función que se quiere integrar. Esto no es necesario cuando se trata de una de las funciones de biblioteca de *Matlab*, como en el caso de  $\sin(x)$  en el tercer ejemplo.

Alternativamente, cuando una función  $f(x)$  es muy simple y se quiere evitar tener que hacer un programa *f.m* para evaluarla, puede utilizarse el comando *inline* como en el siguiente ejemplo:

```
f = inline('exp(-x.^2)'); % Define la funcion f(x)=exp(-x^2).
trap(f,0,1,20) % Calcula la integral de f(x) en [0,1] con 20 subintervalos.
```

Paralelo a esto, el programa matlab posee otras funciones para poder calcular integrales mediante métodos basados en curvas de interpolación (polinomios principalmente), como por ejemplo la regla de Simpson entre otras.

Esta última se utiliza de la siguiente forma:

```
simp(f,0,1,20) % Calcula la integral de f(x) en [0,1] con 20 subintervalos.
```

1. a) Transcriba en un archivo *trap.m* el siguiente programa que calcula la aproximación de la integral de una función dada en un intervalo genérico  $[a, b]$  por la regla de los trapecios con  $N$  subintervalos:

```
function int=trap(funcnt,a,b,N)
h=(b-a)/N;
x=a+h*(1:N-1);
int=h*((feval(funcnt,a)+feval(funcnt,b))/2+sum(feval(funcnt,x))));
```

- b) Pruebe el programa anterior con las siguientes integrales y  $N = 10, 20, 40, 80$ .

$$\int_0^3 x^2 dx; \quad \int_{-1}^1 e^{-x^2} dx; \quad \int_1^2 \log(x) dx; \quad \int_0^1 \sqrt{x} dx$$

Para ello ingrese cuando sea posible el integrando como una función de biblioteca y, cuando no lo sea, utilice el comando *inline*. Imprima los resultados en formato largo (format long).

- c) Haga un programa semejante para la regla de Simpson y pruébelo con las mismas funciones.
2. El fin de este ejercicio es verificar experimentalmente que el error del método de los trapecios aplicado a un integrando con derivadas acotadas en el intervalo de integración es de orden  $O(h^2)$ .
  - a) Haga un programa que calcule la integral  $\int_0^1 e^{-x} dx$  por la regla de los trapecios con  $N = 10, 20, 30, \dots, 100$  subintervalos y almacene los errores respectivos.  
**Sugerencia:** para calcular el error utilice el valor verdadero de la integral con el comando *int*, que en este caso puede calcularse exactamente (de no estar implementada la función calcule el valor manualmente).
  - b) Grafique en escala logarítmica estos errores versus  $N$  y la función  $f(N) = (1/N)^2$ .  
**Sugerencia:** para graficar en escala logarítmica utilice el comando *loglog* en lugar de *plot*; la sintaxis de ambos comandos es la misma.
  - c) Explique por qué el gráfico anterior muestra que el error de la regla de los trapecios es, en este caso  $O(h^2)$ .
3. El fin de este ejercicio es verificar experimentalmente que el error del método de los trapecios aplicado a un integrando con derivadas no acotadas en el intervalo de integración no es de orden  $O(h^2)$ .
  - a) Repita el ejercicio anterior con la integral  $\int_0^1 \sqrt{x} dx$ .
  - b) Para estimar el orden de convergencia del método en este caso, determine por cuadrados mínimos las constantes  $C$  y  $\alpha$  que mejor ajustan los errores mediante el modelo

$$error = Ch^\alpha$$

- c) Grafique en escala logarítmica los errores versus  $N$  y la función  $f(N) = (1/N)^\alpha$ .