Universidad de los Andes
Departamento de Ingeniería Eléctrica y Electrónica
IELE 2009L – Laboratorio de computación científica para IEE
Gerardo Andrés Riaño Briceño (201112388)

Integración y derivación numérica

1. Integración

Utilizando las funciones *integral2(), integral3()* y *triplequad()* se resuelven las integrales definidas propuestas en el enunciado. El número de cifras significativas de cada resultado se determina a partir de la desigualdad,

$$\epsilon_{rel} \leq 5 \cdot 10^{-t}$$

Donde ϵ_{rel} es el error relativo y t el número de cifras significativas. Teniendo en cuenta que las funciones de Matlab calculan el resultado con un error relativo de $1 \cdot 10^{-6}$ por defecto, los resultados tienen al menos 6 cifras significativas. Los resultados obtenidos se muestran a continuación:

Integral f1 = 1.644934
Integral f2 (integral3) = 1.333333
Integral f2 (triplequad) = 1.333333
Integral f3 = 83.974026

¿A qué hacen referencia los parámetros *AbsTol* y *RelTol* de las funciones?

Hacen referencia al error absoluto y relativo que se obtiene como resultado del cálculo con el método numérico. Los valores por defecto de estos parámetros son $1 \cdot 10^{-6}$ para el error relativo y $1 \cdot 10^{-10}$ para el error absoluto. Ambos parámetros sirven como criterio de parada para el método numérico, ya que se busca que,

$$\left|\hat{I} - I\right| \leq \max\left(RelTol, AbsTol \cdot \left|\hat{I}\right|\right)$$

$$\hat{I}: valor\ estimado\ de\ la\ integral$$
$$I: valor\ real\ de\ la\ integral$$

Cuando la diferencia $|\hat{I} - I|$ es suficientemente pequeña, la convergencia del algoritmo puede estar determinada por el error absoluto. En el caso de la función *triplequad(),* la convergencia está determinada únicamente por el valor del error relativo.

Con el fin de verificar la incidencia de los parámetros en los resultados finales, se hacen variaciones a los valores del error relativo y absoluto. Para las funciones *integral(),* se fija uno de los errores en cero, y se hacen variaciones al valor que no está fijo. En el caso de la función *triplequad()* solo se hacen variaciones al error relativo. Los resultados se muestran en la Tabla 1 y la Tabla 2.

Tabla 1. Resultados para distintos valores de error absoluto.

Error absoluto	1	1.00E-01	1.00E-02	1.00E-03	1.00E-04	1.00E-05
f1	1.645091	1.645091	1.645091	1.644973	1.644937	1.644934
f2 (*integral)*	1.333333	1.333333	1.333333	1.333333	1.333333	1.333333
f2 (*triplequad)*	1.333333	1.333333	1.333333	1.333333	1.333333	1.333333
f3	83.974016	83.974027	83.974026	83.974026	83.974026	83.974026

Tabla 2. Resultados para distintos valores de error relativo.

Error relativo	1	1.00E-01	1.00E-02	1.00E-03	1.00E-04	1.00E-05
f1	1.645091	1.645091	1.645091	1.645091	1.644944	1.644935
f2 (*integral)*	1.333333	1.333333	1.333333	1.333333	1.333333	1.333333
f2 (*triplequad)*	1.333333	1.333333	1.333333	1.333333	1.333333	1.333333
f3	83.974016	83.974016	83.974016	83.974027	83.974026	83.974026

Al variar dichos parámetros se comprueba que el número de cifras significativas de la solución es menor cuando la tolerancia admisible es mayor. Además, se comprueba que la desigualdad planteada anteriormente se satisface en todos los casos, y que en algunos casos la tolerancia puede proporcionar más cifras significativas que las mínimas calculadas. Por ejemplo, en la Tabla 2 el número de cifras significativas es el mismo para errores relativos de 1, 1E-01 y 1E-02.

¿Qué métodos usan estos procedimientos?

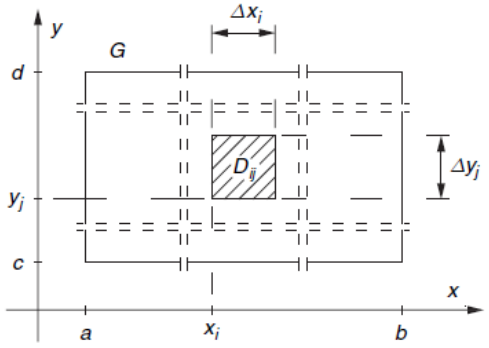

Figura 1. Región de integración.

La función *integral2(), por defecto,* resuelve la integral de dos variables asumiendo que la región de integración puede dividirse en $n \times m$ celdas de área $\Delta x \cdot \Delta y$ (ver Figura 1), con $n = \frac{x_{max} - x_{min}}{\Delta x}$ y $m = \frac{y_{max} - y_{min}}{\Delta y}$, si los tamaños de paso Δx y Δy se fijan constantes. De modo que el valor de la integral en cada celda es igual a

$$\widehat{I_i} = \Delta x_i \cdot \Delta y_i \cdot f_i(x, y)$$

y el valor de la integral es $\hat{I} = \sum_{i=1}^{n} \widehat{I_i}$. La función *integral3()* se base en esta función y en la función *integral()* para hacer el cálculo de la integral triple. Por otro lado, la función *triplequad()* usa un principio similar, solo que aproxima la región de integración a celdas espaciadas de manera óptima, con la función de cuadratura. Cuando la función es discontinua en uno de los puntos dentro del intervalo de integración, las funciones utilizan un método iterativo, mediante el cual se calcula el valor de la integral interna, con la función integral, el resultado se utiliza luego para resolver la integral externa. El proceso se lleva a cabo iterativamente hasta alcanzar

un criterio de convergencia. De este modo, no solo es posible evitar problemas por discontinuidad de la función, también es posible evaluar la integral cuando los límites de integración son indefinidos.

2. Derivación

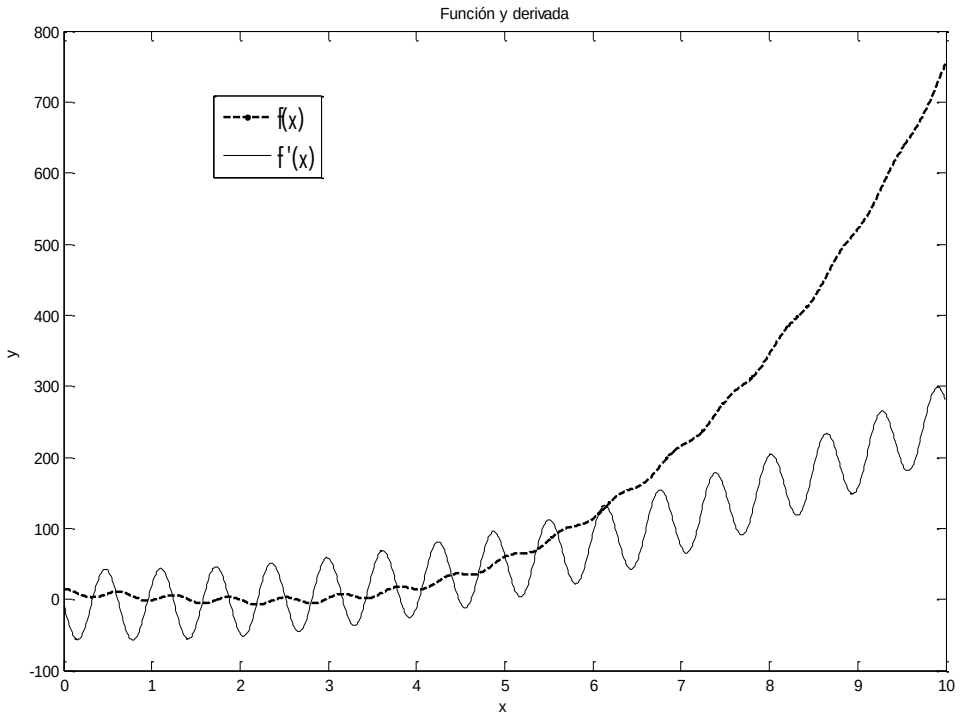

Figura 2. Función y derivada evaluadas en el intervalo (0, 10)

Al disminuir el tamaño de paso Δx disminuye significativamente el error del resultado numérico, como puede observarse en la Figura 3 y Figura 4. También, es posible apreciar que el error es mayor cuando la función deja de ser suave, i.e., donde la pendiente es más pronunciada.

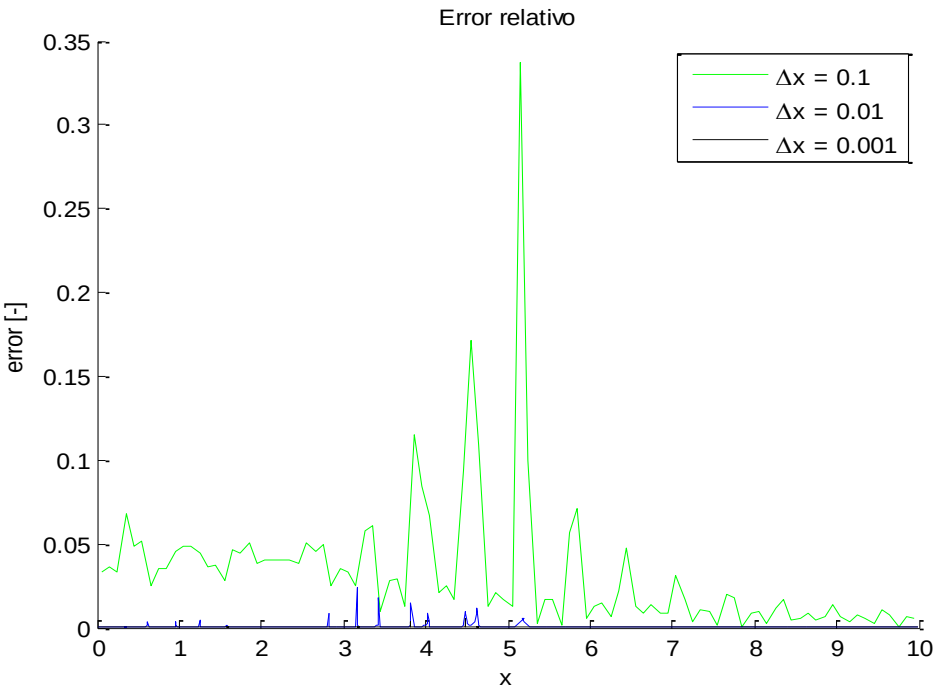

Figura 3. Error relativo derivación numérica.

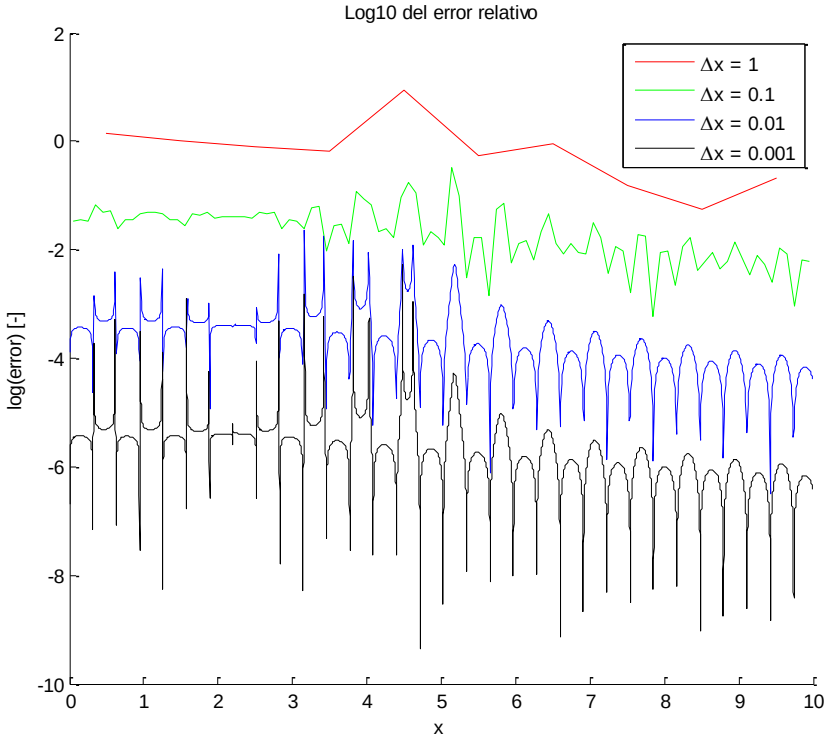

Figura 4. Log10 del error relativo.

Anexos

Código integración

```
function [r1, r21, r22, r3] = integracion(RelTol, AbsTol)

    f1 = @(x,y) 1 ./ (1 - x.*y); % Función 1
    f2 = @(x,y,z) 64.*x.*y.*z.*(1 - x).^2; % Función 2
    f3 = @(x,y) x.^2.*y; % Función 3

    % Integral de función 1
    r1 = integral2(f1, 0,1,0,1, 'RelTol', RelTol, 'AbsTol', AbsTol);
    % Integral de función 2
    r21 = integral3(f2, 0,1,0,1,0,1, 'RelTol', RelTol, 'AbsTol', AbsTol);
    r22 = triplequad(f2, 0,1,0,1,0,1,RelTol);
    % Integral de función 3
    r3 = integral2(f3, 1,2, @(x) x.^2, @(x) x.^4, 'RelTol', RelTol,
'AbsTol', AbsTol);

end
```

Script principal integración

```
err = [1, 1e-1, 1e-2, 1e-3, 1e-4, 1e-5];

for i = err
   [r1, r21, r22, r3] = integracion(i, 0); % Variacion RelTol y AbsTol
    fprintf('%.6f\n', r1);
    fprintf('%.6f\n', r21);
    fprintf('%.6f\n', r22);
    fprintf('%.6f\n', r3);
    fprintf('%s\n', '');
end
```

Código derivación

```
function [t,d] = derivada(h)

    % h: tamaño de paso
    x = 0:h:10;
    y = 5*cos(10*x) + x.^3 - 2*x.^2 - 6*x + 10;
    d = diff(y)./diff(x);

% Se ajusta el vector x, ya que la derivada queda evaluada en el medio
del intervalo

    t = (x + h/2); t(length(t)) = [];

end
```

Script principal derivación

```
%clear all; clc;
syms x;

[x1,d1] = derivada(1);
[x2,d2] = derivada(0.1);
[x3,d3] = derivada(0.01);
[x4,d4] = derivada(0.001);

f1 = double(subs(3*x.^2 - 50*sin(10*x) - 4*x - 6, x, x1));
f2 = double(subs(3*x.^2 - 50*sin(10*x) - 4*x - 6, x, x2));
f3 = double(subs(3*x.^2 - 50*sin(10*x) - 4*x - 6, x, x3));
f4 = double(subs(3*x.^2 - 50*sin(10*x) - 4*x - 6, x, x4));

errrel1 = abs(f1-d1)./f1; errrel2 = abs(f2-d2)./f2;
errrel3 = abs(f3-d3)./f3; errrel4 = abs(f4-d4)./f4;
errabs1 = f1-d1; errabs2 = f2-d2;
errabs3 = f3-d3; errabs4 = f4-d4;

figure; hold on;
plot(x1, log10(abs(errrel1)), 'r'); plot(x2, log10(abs(errrel2)), 'g');
plot(x3, log10(abs(errrel3)), 'b'); plot(x4, log10(abs(errrel4)), 'k');
```