

Tema 5

Derivación e Integración numérica

RESUMEN TEÓRICO

5.1. Efecto de los errores de redondeo en la derivación numérica

Para calcular una aproximación numérica de la derivada:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

podemos elegir una sucesión $\{h_k\} \rightarrow 0$ y calcular los términos de la sucesión $\{D_k\}$ dados por

$$D_k = \frac{f(x+h_k) - f(x)}{h_k} \quad \text{para } k = 1, 2, \dots$$

La pregunta que nos planteamos es ¿qué valor de h_k hay que elegir para asegurar que D_k es una buena aproximación a la derivada $f'(x)$?

Ejemplo 5.1. Para la función $f(x) = e^x$ calcular $f'(1)$ y el error cometido .

Los valores obtenidos se muestran en la siguiente tabla.

h_k	$D_k = \frac{f(1+h_k) - f(1)}{h_k}$	$error = D_k - f'(1) $
$h_1 = 0.1$	2.85884195487388	0.14056012641483
$h_2 = 0.01$	2.73191865578708	0.01363682732803
$h_3 = 0.001$	2.71964142253278	0.00135959407374
$h_4 = 0.0001$	2.71841774707848	0.00013591861944
$h_5 = 0.00001$	2.71829541991231	0.00001359145326
$h_6 = 10^{-6}$	2.71828318698653	0.00000135852748
$h_7 = 10^{-7}$	2.71828196396484	0.00000013550580
$h_8 = 10^{-8}$	2.71828177744737	0.00000005101167
$h_9 = 10^{-9}$	2.71828159981169	0.00000022864736
$h_{10} = 10^{-10}$	2.71827893527643	0.00000289318262

Como se puede apreciar los valores del error $|D_k - f'(1)|$ disminuyen hasta $h_k = 10^{-8}$ y luego aumentan. Este fenómeno es debido a que al hacer la diferencia $f(1+h_k) - f(1)$ aparece el problema de la pérdida de cifras significativas debido a la substracción de dos cantidades que son casi iguales.

5.2. Fórmulas de los n -puntos. Error de truncamiento.

El siguiente teorema utiliza el polinomio interpolador de una función f para obtener fórmulas de aproximación a la derivada de una función f .

Teorema 5.1 (fórmula de n puntos). *Sea f una función de clase $C^{n+1}[a, b]$ y $\{x_1, x_2, \dots, x_n\}$ n puntos distintos de dicho intervalo. Si llamamos $L_i(x)$ a los correspondientes polinomios elementales de Lagrange de grado $n - 1$, entonces existe un punto $\alpha \in [a, b]$ tal que*

$$f'(x_k) = \sum_{i=1}^n f(x_i) L'_i(x_k) + \frac{f^{(n)}(\alpha)}{n!} \prod_{\substack{i=1 \\ i \neq k}}^n (x_k - x_i).$$

Las fórmulas más utilizadas son las que emplean tres y cinco puntos de evaluación. Las siguientes fórmulas se obtienen de las fórmulas de 3 puntos:

- Si tomamos nodos equidistantes para aproximar $f'(x)$ $x_1 = x$, $x_2 = x + h$ y $x_3 = x + 2h$, con $h > 0$ de acuerdo a la fórmula anterior tenemos

$$f'(x) = \frac{-3f(x) + 4f(x+h) - f(x+2h)}{2h} + \frac{h^2}{3} f^{(3)}(\alpha),$$

que se conoce como **fórmula progresiva de tres puntos**.

- Si los nodos son $x_1 = x - 2h$, $x_2 = x - h$ y $x_3 = x$, con $h > 0$ de acuerdo a la fórmula anterior tenemos

$$f'(x) = \frac{f(x-2h) - 4f(x-h) + 3f(x)}{2h} + \frac{h^2}{3} f^{(3)}(\alpha),$$

que se conoce como **fórmula regresiva de tres puntos**.

- Si tomamos $x_1 = x - h$, $x_2 = x$ y $x_3 = x + h$ para aproximar $f'(x)$, nos queda lo que se conoce como **fórmula centrada de tres puntos**:

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} - \frac{h^2}{6} f^{(3)}(\alpha).$$

Observemos que la cota del error en este último caso es aproximadamente la mitad que en los otros dos casos, además esta fórmula necesita menos evaluaciones de f que la anterior.

El error de truncamiento, si $f^{(3)}(\alpha)$ no cambia muy rápidamente, tiende a cero a la misma velocidad que h^2 . Ahora bien de acuerdo al ejemplo anterior no es aconsejable elegir h demasiado pequeño, por lo que sería útil disponer de fórmulas que aproximen $f'(x)$ y que tengan un error de truncamiento de orden mayor al dado.

A continuación damos un programa de la fórmula centrada de tres puntos

```
function D=difc3p(f,x,h)
%      Datos de entrada
% - f es la función, introducida como una cadena de caracteres 'f'
% - x es el punto en el que se deriva
% - h es el incremento tomado
%      Datos de salida
% - D es la aproximación a la derivada mediante la formula
%      centrada de tres puntos
%
D(1,1)=(feval(f,x+h)-feval(f,x-h))/(2*h);
```

5.2.1. Extrapolación de Richardson.

Aplicando el teorema de Taylor, sabemos que si f es de clase $C^{2n+3}[a, b]$, entonces para $x \in (a, b)$ se tiene

$$f(x+h) = f(x) + f'(x)h + \frac{f''(x)}{2!}h^2 + \frac{f'''(x)}{3!}h^3 + \dots + \frac{f^{2n+2}(x)}{(2n+2)!}h^{2n+2} + \frac{f^{2n+3}(\alpha_1)}{(2n+3)!}h^{2n+3},$$

$$f(x-h) = f(x) - f'(x)h + \frac{f''(x)}{2!}h^2 - \frac{f'''(x)}{3!}h^3 + \dots + \frac{f^{2n+2}(x)}{(2n+2)!}h^{2n+2} - \frac{f^{2n+3}(\alpha_2)}{(2n+3)!}h^{2n+3},$$

y por tanto al restar a la primera ecuación la segunda y dividir por $2h$, resulta

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{f'''(x)}{3!}h^2 + \dots + \frac{f^{2n+1}(x)}{(2n+1)!}h^{2n} + \frac{f^{2n+3}(\alpha_1) + f^{2n+3}(\alpha_2)}{2(2n+3)!}h^{2n+2},$$

de donde se obtiene que

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \sum_{i=1}^n k_i h^{2i} + k_{n+1}(h)h^{2n+2},$$

con k_1, k_2, \dots, k_n independientes de h .

La extrapolación de Richardson se basa en la dependencia de h de la función error y con ello llegar a fórmulas de orden alto partiendo de fórmulas de orden bajo.

Si denotamos $D_k(h)$ una aproximación de orden $\mathcal{O}(h^{2k})$ a $f'(x)$ entonces se tendría

$$f'(x) = D_k(h) + c_1 h^{2k} + c_2 h^{2k+2} + \dots, \quad \text{para } k = 1, 2, 3, \dots, \quad (5.1)$$

siendo $D_1(h) = \frac{f(x+h) - f(x-h)}{2h}$. Considerando ahora $h/2$ en lugar de h se tiene

$$f'(x) = D_k(h/2) + \frac{c_1}{4^k} h^{2k} + \frac{c_2}{4^{k+1}} h^{2k+2} + \dots. \quad (5.2)$$

Si multiplicamos la ecuación (5.2) por 4^k y la restamos de la ecuación (5.1) resulta

$$f'(x) = \frac{4^k D_k(h/2) - D_k(h)}{4^k - 1} + \mathcal{O}(h^{2k+2}).$$

Por tanto, si denotamos $D_{k+1}(h) = \frac{4^k D_k(h/2) - D_k(h)}{4^k - 1}$, se cumple que

$$f'(x) = D_{k+1}(h) + \mathcal{O}(h^{2k+2}).$$

En resumen tenemos el siguiente algoritmo que aplica la extrapolación de Richardson a la fórmula centrada de tres puntos

$$D_1(h) := \frac{f(x+h) - f(x-h)}{2h},$$

$$D_{j+1}(h) := \frac{4^j D_j(h/2) - D_j(h)}{4^j - 1} = D_j(h/2) + \frac{D_j(h/2) - D_j(h)}{4^j - 1}, \quad \text{para } j \geq 1,$$

el cual se puede implementar en forma de tabla triangular como sigue

$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$
$D_1(h) = \frac{f(x+h) - f(x-h)}{2h}$		
$D_1(h/2) = \frac{f(x+h/2) - f(x-h/2)}{h}$	$D_2(h) = \frac{4D_1(h/2) - D_1(h)}{4-1}$	
$D_1(h/2^2) = \frac{f(x+h/2^2) - f(x-h/2^2)}{h/2}$	$D_2(h/2) = \frac{4D_1(h/2^2) - D_1(h/2)}{4-1}$	$D_3(h) = \frac{4^2 D_2(h/2) - D_2(h)}{4^2 - 1}$

El primer elemento $D(1, 1)$ de la tabla anterior lo podemos calcular mediante el programa difc3p.m, y los restantes elementos de dicha columna $D(j+1, 1) = D_1(h/2^j)$, para $j \geq 1$, se calculan aplicando el mismo programa con incremento $h/2^j$.

Los restantes elementos de dicha tabla $D(j+1, k+1) = D_{k+1}(h/2^{j-k})$, donde $j \geq 1$, $1 \leq k \leq j$, los calcularemos mediante el algoritmo anterior expresando

$$D(j+1, k+1) = \frac{4^k D(j+1, k) - D(j, k)}{4^k - 1} = D(j+1, k) + \frac{D(j+1, k) - D(j, k)}{4^k - 1}.$$

El siguiente programa en Matlab, basado en el esquema anterior, nos da una aproximación numérica a $f'(x)$ mediante un elemento de la diagonal $f'(x) \approx D(n, n)$.

```

function [D,err,n]=difrichardson(f,x,h,tol,nmax)
%           Datos de entrada
% - f es la función, introducida como una cadena de caracteres 'f'
% - x es el punto en el que se deriva
% - h es el incremento inicial
% - tol es la tolerancia para el error
% - nmax numero máximo de iteraciones
%           Datos de salida
% - D es la matriz con las aproximaciones a la derivada
% - err es la cota del error
% - n es la coordenada de la "mejor aproximación"
%
err=1; j=1;
% Construye el primer elemento de la tabla
D(1,1)=difc3p(f,x,h);
while err>tol & j<nmax
    % Construye el primer elemento de la fila j+1
    D(j+1,1)=difc3p(f,x,h/2^j);
    % Construye los restantes elementos de la fila j+1
    for k=1:j
        D(j+1,k+1)=D(j+1,k)+(D(j+1,k)-D(j,k))/(4^k-1);
    end
    % Calcula el error como la diferencia entre dos elementos de la diagonal
    err=abs(D(j+1,j+1)-D(j,j));
    j=j+1;
end
[n,n]=size(D);

```

Nota 5.1. En el caso particular de $k = 2$ se obtiene la fórmula centrada de 5 puntos que es de orden $\mathcal{O}(h^4)$

$$\begin{aligned}
 f'(x) &\approx D(2,2) = D_2(h) = \frac{4D_1(h/2) - D_1(h)}{4-1} = \frac{4\left(\frac{f(x+h/2)-f(x-h/2)}{h}\right) - \left(\frac{f(x+h)-f(x-h)}{2h}\right)}{4-1} \\
 &= \frac{-f(x+h) + 8f(x+h/2) - 8f(x-h/2) + f(x-h)}{6h}
 \end{aligned}$$

5.2.2. Análisis del error en la diferenciación numérica

Supongamos que hacemos los cálculos con un computador

$$f(x_0 - h) = y_{-1} + e_{-1}, \quad f(x_0 + h) = y_1 + e_1,$$

donde y_{-1} , y_1 son los valores obtenidos con el computador y e_{-1} , e_1 son los errores de redondeo.

Si usamos la fórmula centrada de tres puntos $f'(x_0) = \frac{y_1 - y_{-1}}{2h}$ entonces

$$f'(x_0) = \frac{y_1 - y_{-1}}{2h} + E(f, h),$$

donde $E(f, h)$ es el error cometido al sumar el error de truncamiento $E_{\text{trunc}}(f, h)$, y el error de redondeo $E_{\text{red}}(f, h)$, dados por

$$E_{\text{trunc}}(f, h) = \frac{h^2 f^{(3)}(\alpha)}{6} \quad \text{y} \quad E_{\text{red}}(f, h) = \frac{e_1 + e_{-1}}{2h}.$$

Si además se cumple que

- 1) $|e_{-1}| \leq \epsilon, |e_1| \leq \epsilon,$
- 2) $M = \max \{|f^{(3)}(x)|/a \leq x \leq b\},$

entonces

$$|E(f, h)| \leq \frac{\epsilon}{h} + \frac{Mh^2}{6},$$

y el valor de h que minimiza la expresión del miembro derecho de esta última desigualdad es el llamado **incremento óptimo**

$$h = \left(\frac{3\epsilon}{M}\right)^{\frac{1}{3}}.$$

5.2.3. Derivadas de orden superior

Para obtener fórmulas de diferencias centradas que aproximen las derivadas de orden superior se puede emplear el Teorema de Taylor.

$$\begin{aligned} f(x+h) &= f(x) + f'(x)h + \frac{f''(x)}{2!}h^2 + \frac{f'''(x)}{3!}h^3 + \frac{f^{(4)}(\alpha_1)}{4!}h^4, \\ f(x-h) &= f(x) - f'(x)h + \frac{f''(x)}{2!}h^2 - \frac{f'''(x)}{3!}h^3 + \frac{f^{(4)}(\alpha_2)}{4!}h^4, \end{aligned}$$

de donde obtenemos

$$f(x+h) + f(x-h) = 2f(x) + f''(x)h^2 + \frac{h^4}{4!} [f^{(4)}(\alpha_1) + f^{(4)}(\alpha_2)],$$

lo que nos permite enunciar el siguiente teorema.

Teorema 5.2. (fórmula centrada de tres puntos para el calculo de f'')

Supongamos que $f \in C^4[a, b]$ y que $x-h, x, x+h \in [a, b]$. Entonces

$$f''(x) = \frac{f(x+h) - 2f(x) + f(x-h)}{h^2} + E(f, h).$$

Si $f(x+h) = y_1 + e_1, f(x-h) = y_{-1} + e_{-1}, f(x) = y_0 + e_0$ donde y_1, y_{-1}, y_0 son los valores obtenidos por el computador y e_1, e_{-1}, e_0 son los errores que se cometen al calcular los valores de f , incluyendo los errores de medida y redondeo, entonces

$$E(f, h) = \frac{e_1 - 2e_0 + e_{-1}}{h^2} - \frac{h^2 f^{(4)}(\alpha)}{12},$$

para algún valor $\alpha \in [x - h, x + h]$.

Si suponemos que cada error e_0, e_{-1}, e_1 es de tamaño menor o igual que ϵ y que $|f^{(4)}(x)| \leq M$, entonces obtenemos la siguiente cota de error

$$|E(f, h)| \leq \frac{4\epsilon}{h^2} + \frac{Mh^2}{12},$$

siendo mínima dicha cota cuando

$$h = \left(\frac{48\epsilon}{M}\right)^{\frac{1}{4}}.$$

Para obtener fórmulas para la derivada segunda $f''(x)$ de orden $\mathcal{O}(h^4)$, $\mathcal{O}(h^6)$, ... se aplica el método de extrapolación de Richardson.

Análogamente si $f \in \mathcal{C}^5[a, b]$ y $x - 2h, x - h, x, x + h, x + 2h \in [a, b]$ se tiene la fórmula de cinco puntos para la derivada tercera

$$f'''(x) = \frac{f(x + 2h) - 2f(x + h) + 2f(x - h) - f(x - 2h)}{2h^3} + k_1h^2 + k_2h^4 + k_3h^6 + \dots$$

Por tanto se trata de una fórmula de orden $\mathcal{O}(h^2)$, para obtener fórmulas de orden superior $\mathcal{O}(h^4)$, $\mathcal{O}(h^6)$, ... se aplica el método de extrapolación de Richardson.

5.3. Integración numérica

Nuestro objetivo es aproximar $\int_a^b f(x)dx$ evaluando $f(x)$ en un número finito de puntos.

Definición 5.1. Supongamos que $a = x_0 < x_1 < \dots < x_n = b$. Llamamos **fórmula de integración numérica o de cuadratura** a una del tipo

$$\int_a^b f(x)dx = Q[f] + E[f]$$

donde

$$Q[f] = \sum_{k=0}^n \omega_k f(x_k) = \omega_0 f(x_0) + \omega_1 f(x_1) + \dots + \omega_n f(x_n).$$

El término $E[f]$ se llama **error de truncamiento** de la fórmula, los valores $\{x_k\}$ **nodos de integración** y los valores $\{\omega_k\}$ se llaman **pesos de la fórmula**

La obtención de las fórmulas de cuadratura puede hacerse a partir de polinomios de interpolación.

Recordemos que existe un único polinomio $p(x)$ de grado menor o igual que n que pasa por los $n + 1$ nodos dados $\{(x_k, y_k)\}_{k=0}^n$.

Las fórmulas de cuadratura se llaman **Fórmulas de Newton-Côtes** cuando aproximamos $\int_a^b f(x)dx \approx \int_a^b p(x)dx$ siendo $p(x)$ el polinomio interpolador con nodos igualmente espaciados.

Ejemplos particulares de fórmulas de Newton-Côtes:

- Para $n = 1$ (Regla del trapecio)

$$\int_{x_0}^{x_1} f(x)dx \approx \int_{x_0}^{x_1} p_1(x)dx = \int_{x_0}^{x_1} \left[f(x_0) \frac{x - x_1}{x_0 - x_1} + f(x_1) \frac{x - x_0}{x_1 - x_0} \right] dx.$$

Denotando $h = x_1 - x_0$ y desarrollando resulta el siguiente valor en el que ya incluimos el error

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}[f(x_0) + f(x_1)] - \frac{h^3}{12}f^{(2)}(\alpha), \quad x_0 < \alpha < x_1.$$

- Para $n = 2$ (Regla de Simpson)

$$\int_{x_0}^{x_2} f(x)dx \approx \int_{x_0}^{x_2} p_2(x)dx = \int_{x_0}^{x_2} \left[f(x_0) \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} + f(x_1) \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} + f(x_2) \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} \right] dx.$$

Denotando $h = x_1 - x_0 = x_2 - x_1$ y desarrollando resulta el siguiente valor en el que ya incluimos el error

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}[f(x_0) + 4f(x_1) + f(x_2)] - \frac{h^5}{90}f^{(4)}(\alpha), \quad x_0 < \alpha < x_2.$$

5.3.1. Fórmulas compuestas de cuadratura

Teorema 5.3. (Regla del trapecio compuesta) Sea $f \in C^2[a, b]$, entonces existe un punto $\alpha \in [a, b]$ tal que la regla del trapecio compuesta, que denotaremos por $T(f, h)$, para los nodos $x_k = a + kh$ ($k = 0, 1, \dots, n$), siendo $h = (b - a)/n$, puede expresarse como

$$\int_a^b f(x)dx = T(f, h) - \frac{b - a}{12}h^2 f^{(2)}(\alpha),$$

donde

$$T(f, h) = \frac{h}{2} \left[f(a) + 2 \sum_{k=1}^{n-1} f(x_k) + f(b) \right].$$

El siguiente programa en Matlab implementa el método de los trapecios


```

function T=trapecio(f,a,b,n)
%       Datos de entrada
% - f es el integrando, dado como una cadena de caracteres 'f'
% - a y b son los extremos inferior y superior del intervalo de
%   integración
% - n es el numero de subintervalos
%       Datos de salida
% - T es la aproximación obtenida con la regla compuesta del trapecio
h=(b-a)/n;
% nodos interiores
xint=a+h*[1:n-1];
% valores de f en los nodos interiores
fint=feval(f,xint);
% valor aproximado de la integral mediante el método del trapecio
T=h*(feval(f,a)+feval(f,b)+2*sum(fint))/2;

```

Teorema 5.4. (Regla de Simpson compuesta) Sea $f \in C^4[a, b]$, entonces existe un punto $\alpha \in [a, b]$ tal que la regla compuesta de Simpson para los nodos $x_k = a + kh$ ($k = 0, 1, \dots, n$), siendo $n = 2m$ par y $h = (b - a)/n$, puede expresarse como

$$\int_a^b f(x)dx = \frac{h}{3} \left[f(a) + 2 \sum_{k=1}^{m-1} f(x_{2k}) + 4 \sum_{k=1}^m f(x_{2k-1}) + f(b) \right] - \frac{b-a}{180} h^4 f^{(4)}(\alpha).$$

Nota 5.2. Como se observa los términos del error $E[f]$, para las reglas compuestas del trapecio y de Simpson son de orden $\mathcal{O}(h^2)$ y $\mathcal{O}(h^4)$ respectivamente. Por tanto el error en la fórmula de Simpson tiende a cero más rápidamente que el error de la regla del trapecio cuando $h \rightarrow 0$.

Además cuando se conoce $f^{(2)}(x)$ y $f^{(4)}(x)$ dichas fórmulas de error nos permiten estimar el número de subintervalos necesarios para alcanzar la precisión deseada.

A continuación implementamos el método de Simpson en el siguiente programa

```
function S=simpson(f,a,b,m)
%       Datos de entrada
% - f es el integrando, dado como una cadena de caracteres 'f'
% - a y b son los extremos inferior y superior del intervalo de
%   integración
% - 2*m es el numero de subintervalos
%       Datos de salida
% - S es la aproximación obtenida con la regla compuesta de Simpson
h=(b-a)/(2*m);
% nodos impares
ximp=a+h*[1:2:2*m-1];
% valores de f en los nodos impares
fimp=feval(f,ximp);
% nodos pares
xpar=a+h*[2:2:2*m-2];
% valores de f en los nodos pares
fpar=feval(f,xpar);
% valor aproximado de la integral mediante Simpson
S=h*(feval(f,a)+feval(f,b)+4*sum(fimp)+2*sum(fpar))/3;
```

5.3.2. Integración de Romberg

El proceso secuencial de tomar dos subintervalos, luego cuatro, luego ocho, etc. mediante la regla compuesta del trapecio para determinar una aproximación de la integral genera una sucesión de aproximaciones $\{T(f, h_J)\}$ donde $h_J = \frac{b-a}{2^{J-1}}$ que presentan el inconveniente de que al doblar el número de subintervalos, prácticamente se dobla el número de evaluaciones de la función. El método de integración de Romberg consiste en aplicar el método de extrapolación de Richardson a las aproximaciones de la integración obtenidas por la regla compuesta del trapecio para 2^{J-1} intervalos, $J \geq 1$. Este método consigue eliminar las evaluaciones y sumas redundantes.

Es posible probar (aunque no es fácil) que si $f \in C^\infty[a, b]$ entonces podemos escribir la regla compuesta del trapecio en la forma:

$$\int_a^b f(x) = R(J, 1) + \sum_{i=1}^{\infty} K_i h_J^{2i},$$

donde K_i para cada i es independiente de h_k , $h_J = \frac{b-a}{2^{J-1}}$ y $R(J, 1)$ es la aproximación dada por el método del trapecio para el incremento h_J , esto es, para 2^{J-1} subintervalos. Así:

$$R(1, 1) = T(f, h_1) = \frac{h_1}{2}[f(a) + f(b)] = \frac{b-a}{2}[f(a) + f(b)],$$

$$R(J, 1) = T(f, h_{J-1}) = \frac{1}{2} \left[R(J-1, 1) + h_{J-1} \sum_{i=1}^{2^{J-2}} f(a + (2i-1)h_{J-1}) \right], \quad J \geq 2,$$

Aplicando el método de extrapolación de Richardson obtenemos, para cada $J = 2, 3, \dots, n$ y $k = 2, \dots, J$ una fórmula de aproximación $\mathcal{O}(h_J^{2J})$ definida por

$$R(J, k) = R(J, k - 1) + \frac{R(J, k - 1) - R(J - 1, k - 1)}{4^{k-1} - 1}$$

El cálculo de los valores $R(J, k)$ se suele representar en forma de tabla.

J	$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^8)$	\dots	$\mathcal{O}(h^{2n})$
1	$R(1, 1)$					
2	$R(2, 1)$	$R(2, 2)$				
3	$R(3, 1)$	$R(3, 2)$	$R(3, 3)$			
4	$R(4, 1)$	$R(4, 2)$	$R(4, 3)$	$R(4, 4)$		
\vdots	\vdots	\vdots	\vdots	\vdots	\ddots	
n	$R(n, 1)$	$R(n, 2)$	$R(n, 3)$	$R(n, 4)$	\dots	$R(n, n)$

El método de extrapolación de Richardson podemos aplicarlo también usando la regla compuesta del trapecio y comenzando con un número cualquiera de subdivisiones del intervalo inicial. El siguiente programa implementa el método de Romberg partiendo de un número concreto de subintervalos.

```
function [R,aprox,err,h,J]=romber(f,a,b,n,tol,n0)
% Datos de entrada
% - f es el integrando, dado como una cadena de caracteres 'f'
% - a y b son los extremos inferior y superior del intervalo de
% integración
% - n es el número máximo de filas de la tabla
% - tol es la tolerancia
% - n0 es el numero de subintervalos inicial
% Datos de salida
% - R es la matriz con las aproximaciones a la integral
% - aprox es la aproximación a la integral
% - err es una estimación del error
% - h es el menor de los incrementos usados
% - J es el numero de subdivisiones realizadas
err=1;
J=0;
R(1,1)=trapecio(f,a,b,n0);
while ((err>tol)&(J+1<n))
    J=J+1;
    R(J+1,1)=trapecio(f,a,b,n0*2^J);
    %Construye los restantes elementos de la fila J+1
    for K=1:J
        R(J+1,K+1)=R(J+1,K)+(R(J+1,K)-R(J,K))/(4^K-1);
    end
    err=abs(R(J,J)-R(J+1,K+1));
end
end
```

```
aprox=R(J+1,J+1);  
h=(b-a)/(n0*2^J);  
J;
```

Comandos de Matlab:

- `Q=quad('f',a,b)`

aproxima la integral de $f(x)$ en el intervalo $[a,b]$ usando el método de Simpson con un error menor que 10^{-6} . 'f' es una cadena de caracteres que contiene el nombre de la función.

- `Q=quad('f',a,b,tol)`

aproxima la integral de $f(x)$ en el intervalo $[a,b]$ usando el método de Simpson con un error menor que `tol`. 'f' es una cadena de caracteres que contiene el nombre de la función.

Referencias básicas: Burden y Faires [1985, Cap. 4]; Kincaid y Cheney [1994, Cap. 7]; Henrici [1972, Cap. 12 y 13]; Henrici [1982, 6.1-6.3]; Mathews y Fink [2000, Cap. 6 y 7]; Nakamura [1997, Cap. 5]; Stoer y Bulirsch [1980, 3.1- 3.5];

Ejercicios Resueltos Tema 5

PROBLEMA 5-1: Se trata de calcular $f'(1)$ para la función $f(x) = e^x$. Para ello:

- (1) Evaluar $f'(1)$ mediante la fórmula centrada de dos puntos

$$f'(x) = \frac{f(x+h) - f(x-h)}{2h} + \mathcal{O}(h^2).$$

para distintos valores de h y calcular el incremento óptimo h teniendo en cuenta los errores de truncamiento y redondeo.

- (2) Evaluar $f'(1)$ aplicando tres veces la técnica de extrapolación de Richardson partiendo de la fórmula centrada de dos puntos del apartado (1). Utilizar un incremento inicial $h = 0.1$.
- (3) Evaluar $f'(1)$ aplicando la técnica de extrapolación de Richardson a la fórmula centrada de dos puntos del apartado (1), hasta que dos aproximaciones consecutivas difieran en menos de 10^{-13} . Utilizar un incremento inicial $h = 0.1$.

SOLUCIÓN PROBLEMA 5-1:

Apartado (1)

El siguiente programa nos da una matriz, cuya primera columna es la aproximación a la derivada y la segunda el error cometido, el paso óptimo, la aproximación a la derivada con dicho paso y el error cometido.

```
format long
for n=1:16
    h=10^(-n);
    A(n,1)=diffc3p('exp',1,h);
    A(n,2)=exp(1)-A(n,1);
end
A
% Calculo de M, máximo discreto de |f'''(x)| (en nuestro caso
% |f'''(x)|=exp(x)) en [1-0.1,1+0.1]
M=0;
for x=1-0.1:0.001:1+0.1
    if abs(exp(x))>M
        M= abs(exp(x));
    end
end
hoptimo=(3*eps/M)^(1/3)
'La aproximación para hoptimo es'
Aprox=(exp(1+hoptimo)-exp(1-hoptimo))/(2*hoptimo)
'con error'
exp(1)-Aprox
```

RESULTADO

▷ Ver tabla 5.1.

El paso óptimo es $h_{optimo} = 6.052654025886100e-06$.

La aproximación para h_{optimo} es 2.71828182851247 con error $-5.342792874785118e-011$.

◁

h	$\frac{f(1+h)-f(1-h)}{2h}$	$Error(h)$
10^{-1}	2.72281456394742	-0.00453273548837
10^{-2}	2.71832713338271	-0.00004530492367
10^{-3}	2.71828228150572	-0.00000045304668
10^{-4}	2.71828183298961	-0.00000000453057
10^{-5}	2.71828182851763	-0.00000000005859
10^{-6}	2.71828182829559	0.00000000016346
10^{-7}	2.71828182851763	-0.00000000005859
10^{-8}	2.71828182185629	0.00000000660275
10^{-9}	2.71828182185629	0.00000000660275
10^{-10}	2.71828115572248	0.00000067273657
10^{-11}	2.71829225795273	-0.00001042949368
10^{-12}	2.71849209809716	-0.00021026963811
10^{-13}	2.71782596428238	0.00045586417666
10^{-14}	2.70894418008538	0.00933764837366
10^{-15}	2.88657986402541	-0.16829803556636
10^{-16}	2.22044604925031	0.49783577920873

Cuadro 5.1: Resultados del problema 5-1 apartado (1)

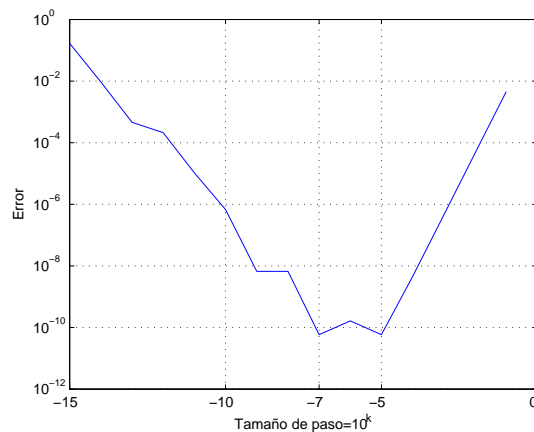


Figura 5.1: Gráfica del error cometido frente al tamaño de paso.

Apartado (2)

El siguiente programa nos da la matriz de Richardson y el error cometido con respecto a la aproximación que ocupa la posición (4,4) en dicha matriz.

```
format long
% Aplicación de Richardson
D=difrichardson('exp',1,0.1,0,4)
'error con respecto a la mejor aproximación'
exp(1)-D(4,4)
```

RESULTADO

▷ Ver tabla 5.2

h	$D_1(h)$	$D_2(h)$	$D_3(h)$	$D_4(h)$
0.1	2.72281456394742			
0.05	2.71941458747318	2.71828126198177		
0.025	2.71856499166488	2.71828179306212	2.71828182846747	
0.0125	2.71835261760135	2.71828182624684	2.71828182845915	2.71828182845902

Cuadro 5.2: Resultado problema 5-1 apartado (2)

el error con respecto a la mejor aproximación es 2.531308496145357e-014. ◀

Apartado (3)

El siguiente programa construye la matriz de Richardson hasta que la diferencia entre dos elementos de la diagonal en dicha matriz sea menor que 10^{-13} .

```
format long
% Aplicación de Richardson
D=difrichardson('exp',1,0.1,10^(-13),100)
```

RESULTADO

▷ Ver tabla 5.3. ◀

$h = 0,1$	$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^8)$	$\mathcal{O}(h^6)$
h	2.72281456394742				
$h/2$	2.71941458747318	2.71828126198177			
$h/2^2$	2.71856499166488	2.71828179306212	2.71828182846747		
$h/2^3$	2.71835261760135	2.71828182624684	2.71828182845915	2.71828182845902	
$h/2^4$	2.71829952564097	2.71828182832085	2.71828182845912	2.71828182845911	2.71828182845912

Cuadro 5.3: Resultado problema 5-1 apartado (3)

PROBLEMA 5-2: Aplicar las fórmulas del trapecio y de Simpson tomando 2, 4 y 6 subintervalos para calcular

$$\int_0^{\pi/4} \text{sen}(x) dx = 1 - (\sqrt{2}/2).$$

Imprimir en una tabla las aproximaciones y los errores correspondientes.

SOLUCIÓN PROBLEMA 5-2:

El siguiente programa genera una matriz con las soluciones.

```
format long
for i=1:3
Ap(i,1)=trapecio('sin',0,pi/4,2*i);
Ap(i,2)=simpson('sin',0,pi/4,i);
Ap(i,3)=abs(Ap(i,1)-1+sqrt(2)/2);
Ap(i,4)=abs(Ap(i,2)-1+sqrt(2)/2);
end
Ap
```

RESULTADO

▷ Los resultados figuran en la tabla 5.4. ◁

n	$T(f, n)$	$S(f, n)$	$E_T(f, n)$	$E_S(f, n)$
n=2	0.28911952428854	0.29293263783975	0.00377369452492	0.00003941902630
n=4	0.29195161745926	0.29289564851617	0.00094160135419	0.00000242970272
n=6	0.29247487881452	0.29289369752943	0.00041833999893	0.00000047871598

Cuadro 5.4: Resultados problema 5-2

PROBLEMA 5-3: Se trata de calcular la integral

$$\int_0^1 \text{sen}(x^2) dx,$$

por distintos métodos de integración compuesta y utilizando Romberg.

- (1) Mediante la regla de trapecios compuesta calcula la integral con el siguiente criterio de paro: Comenzar subdividiendo el intervalo en dos partes, tres partes, etc. Parar cuando el valor absoluto de dos aproximaciones consecutivas sea menor que $TOL = 10^{-6}$. ¿Cuántas subdivisiones se necesitan?. Imprimir la aproximación.
- (2) Mediante la regla de Simpson compuesta calcula la integral con el siguiente criterio de paro: Comenzar subdividiendo el intervalo en dos partes, cuatro partes, etc. Parar cuando el valor absoluto de dos aproximaciones consecutivas sea menor que $TOL = 10^{-6}$. ¿Cuántas subdivisiones se necesitan?. Imprimir la aproximación.

- (3) Emplear integración de Romberg usando la regla compuesta del trapecio, comenzando con $h = 0,1$ y parando cuando dos aproximaciones consecutivas de la diagonal difieran en valor absoluto menos de 10^{-6} . Imprimir la aproximación y la tabla. ¿Y para la tolerancia 10^{-14} ?

SOLUCIÓN PROBLEMA 5-3:

Apartado (1)

Creamos el fichero de función del integrando

```
function y=fprob3tema5(x)
y=sin(x.^2);
```

y a continuación generamos el siguiente fichero de comando que resuelve el apartado (1)

```
format long
i=1;
errf=1;
while errf>=10^(-6)
    i=i+1;
    AprT(i)=trapecio('fprob3tema5',0,1,i);
    err(i)=errf;
    if i>2
        errf=abs(AprT(i)-AprT(i-1));
        err(i)=errf;
    end
end
'aproximación al error absoluto'
errf
'aproximación anterior a la integral'
AprT(i-1)
'aproximación final a la integral'
AprT(i)
'alcanzada tomando el siguiente numero de subintervalos'
i
```

RESULTADO

▷ Para 56 subintervalos la aproximación es 0.31029701880598 y para 57 subintervalos la aproximación es 0.31029601995942 ◁

Apartado (2)

```

format long
i=1;
errf=1;
while errf>=10(-6)
    i=i+1;
    AprT(i)=simpson('fprob3tema5',0,1,i);
    err(i)=errf;
    if i>2
        errf=abs(AprT(i)-AprT(i-1));
        err(i)=errf;
    end
end
'aproximación al error absoluto'
errf
'aproximación anterior a la integral'
AprT(i-1)
'aproximación final a la integral'
AprT(i)
'alcanzada tomando un numero de subintervalos igual a'
2*i

```

RESULTADO

▷ Para 14 subintervalos la aproximación es 0.31026620880279, y para 16 subintervalos la aproximación es 0.31026707591900 ◁

Apartado (3)

Para la tolerancia 10^{-6} ejecutamos los siguientes comandos:

```

format long
%para h=0.1 hemos de considerar n0=10
[R,aprox,err,h,J]=romber('fprob3tema5',0,1,100,10(-6),10)

```

Y para la tolerancia 10^{-14} los siguientes:

```

format long
%para h=0.1 hemos de considerar n0=10
[R,aprox,err,h,J]=romber('fprob3tema5',0,1,100,10(-14),10)

```

RESULTADO

▷ Los resultados para $TOL = 10^{-6}$ y $TOL = 10^{-14}$ aparecen en las tablas 5.5 y 5.6, respectivamente. ◁

h	$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$
0.1	0.31117081121703		
0.05	0.31049355290331	0.31026780013207	
0.025	0.31032459103767	0.31026827041579	0.31026830176803

Cuadro 5.5: Resultados problema 5-3 apartado (3) para $TOL = 10^{-6}$.

h	$\mathcal{O}(h^2)$	$\mathcal{O}(h^4)$	$\mathcal{O}(h^6)$	$\mathcal{O}(h^8)$	$\mathcal{O}(h^{10})$
0.1	0.31117081121703				
0.05	0.31049355290331	0.31026780013207			
0.025	0.31032459103767	0.31026827041579	0.31026830176803		
0.0125	0.31028237258490	0.31026829976731	0.31026830172408	0.31026830172339	
0.00625	0.31027181934708	0.31026830160114	0.31026830172339	0.31026830172338	0.31026830172338

Cuadro 5.6: Resultados problema 5-3 apartado (3) para $TOL = 10^{-14}$.

PROBLEMA 5-4: Se trata de calcular el área formada por la curva $y = f(x) = x^7 + 5x^6 + x^3 - 1$ y el eje OX en $[0, 2]$. Para ello calcular las raíces de la ecuación polinómica en $[0, 2]$ por el método de Newton–Raphson (tomar una primera aproximación gráfica). Las integrales que se tengan que calcular deben hacerse por el método de Romberg hasta que dos aproximaciones diagonales difieran en menos de 10^{-12} .

SOLUCIÓN PROBLEMA 5-4:

Definición de $f(x)$ y $f'(x)$ en los ficheros `fprob4tema5.m` y `dfprob4tema5.m`

```
%Definición de f(x) del problema 4 del tema 5
function y=fprob4tema5(x)
y=x.^7+5*x.^6+x.^3-1;

% Definición de la derivada de f(x) del problema 4 tema 5
function y=dfprob4tema5(x)
y=7*x.^6+30*x.^5+3*x.^2;

%Dibujos de y=fprob4tema5(x) en [0,2]
x=linspace(0,2,1000); plot(x,fprob4tema5(x)),pause
%Ampliación a [0,1]
x=linspace(0,1,1000); plot(x,fprob4tema5(x)),pause
%Sucesivas ampliaciones
x=linspace(0.65,.75,1000); subplot(221),plot(x,fprob4tema5(x))
x=linspace(0.69,.71,1000);
subplot(222),plot(x,fprob4tema5(x)),axis([0.69,0.71,-Inf,Inf])
x=linspace(0.695,.7,1000);
subplot(223),plot(x,fprob4tema5(x)),axis([0.695,0.7,-Inf,Inf])
x=linspace(0.698,.699);
subplot(224),plot(x,fprob4tema5(x)),axis([0.698,0.699,-Inf,Inf])
```

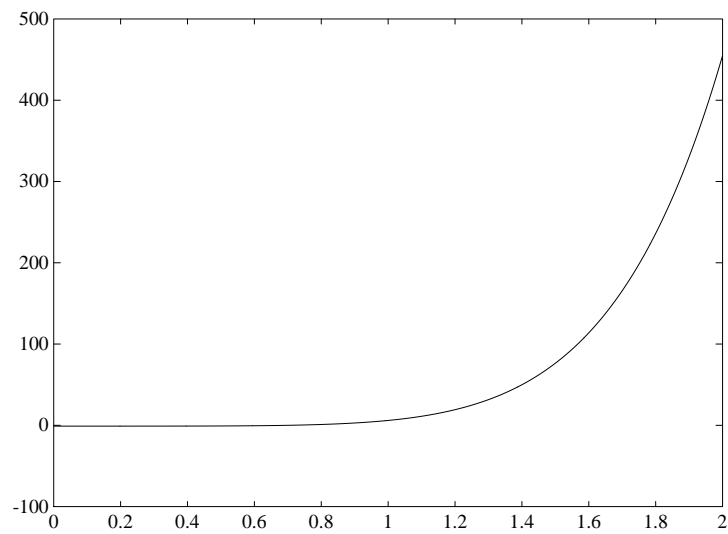


Figura 5.2: Representación gráfica de $y = f(x) = x^7 + 5x^6 + x^3 - 1$ en $[0, 2]$.

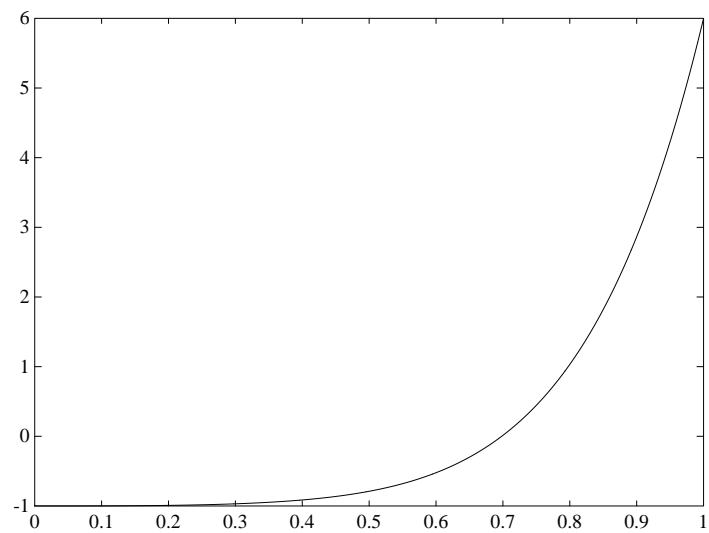


Figura 5.3: Representación gráfica de $y = f(x) = x^7 + 5x^6 + x^3 - 1$ en $[0, 1]$.

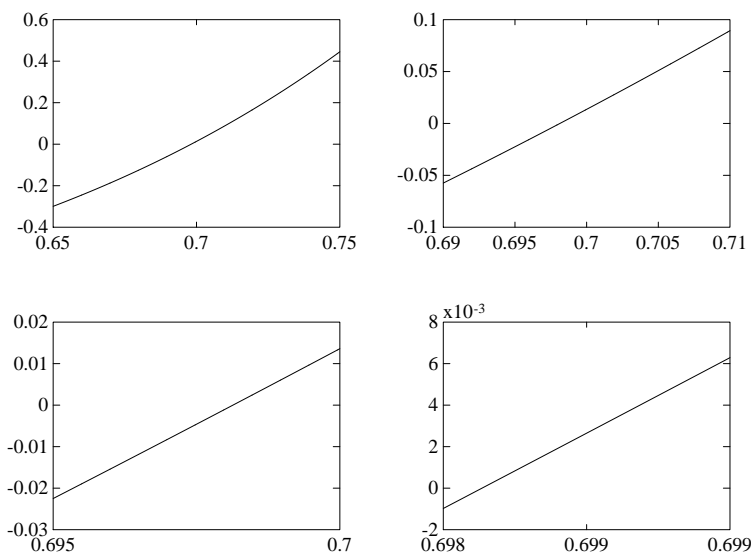


Figura 5.4: Sucesivas ampliaciones de las gráficas anteriores.

▷ Los resultados aparecen en las figuras: 5.2, 5.3 y 5.4, respectivamente
 Por tanto, una aproximación inicial a la raíz en $[0, 2]$ es 0.6985. ◁

```
format long
x=.6985; %después del análisis gráfico (primera aprox)
y=x-fprob4tema5(x)/dfprob4tema5(x);
while abs(x-y)>10^-14
    x=y;
    y=x-fprob4tema5(x)/dfprob4tema5(x);
end
'La raíz es ',
y
```

RESULTADO

▷ La raíz es 0.69813496445960.

En consecuencia, la función dada es negativa para $x \in [0, 0,69813496445960)$ y positiva para $x \in (0,69813496445960, 2]$, por tanto el área viene dada por

$$A = - \int_0^{0,698134964459} f(x)dx + \int_{0,698134964459}^2 f(x)dx.$$

◁

Al ejecutar el siguiente programa

```
format long
%comenzamos con n0=10 subintervalos
[R,aprox1,err,h]=romber('fprob4tema5',0,0.698134964459,100,10^(-12),10)
[R,aprox2,err,h]=romber('fprob4tema5',0.698134964459,2,100,10^(-12),10)
```

RESULTADO

▷ se obtienen los valores -0.57395715491424 para la integral de $f(x)$ entre 0 y 0.698134964459 y 1.260025285834857e+002 para la integral de $f(x)$ entre 0.698134964459 y 2

- Utilizando la instrucción `quad`, se obtienen valores muy parecidos:

```
» quad('fprob4tema5',0,0.698134964459,10^(-12))
-0.57395715491423
```

```
» quad('fprob4tema5',0.698134964459,2,10^(-12))
1.260025285834857e+002
```

El área pedida es:

$$A = - \int_0^{0.698134964459} f(x)dx + \int_{0.698134964459}^2 f(x)dx = 1,265764857383999e + 002.$$

◁