1 Cálculo Analítico y Simbólico

En esta práctica se analizan los métodos de derivación e integración numérica, mostrando la inestabilidad de la derivación.

1.1 Derivación

Sea f(x) una función real, definida en un intervalo [a,b] y derivable en un punto c de [a,b]. Vamos a considerar fórmulas del tipo $f'(c)=\sum_{i=0}^{n}\alpha_i f(x_i)+R(f)$ que expresan que se va a usar $\sum_{i=0}^{n}\alpha_i f(x_i)$ como valor aproximado de $f'(c)$. Utilizamos MATLAB para obtener la derivada exacta y la aproximación numérica de utilizando $f(x)=x\cos(x)$ en $x=\pi/2$ utilizando la expresión:

$$f'(x)=\frac{f(x+h)-f(x)}{h}-\frac{h}{2}f''(\zeta) \text{ con } \zeta\in(x,x+h)$$

```
»% Definición de variables y funciones simbólicas
»syms x real; fs=x*cos(x); dfs=diff(fs),subs(dfs,pi/2) % Derivación analítica
»h=0.1;F=subs(fs,pi/2+[0,h]);diff(F)/h % Derivación numérica
```

La evaluacion numérica también se podría haber realizado mediante

```
»fn=inline(vectorize(fs));
»h=0.1;F=fn(pi/2+[0,h]) ;diff(F)/h
```

Ejercicios recomendados:

1. Calcular la derivada numérica y simbólica de la función $f(x)=x\cos(x)$ en el intervalo [0,2]. Representar el error cometido tomando h=0.025 y h=0.05.
2. Calcular los errores cometidos en 0, 1 y 2, para valores de paso de 10^{-k} variando k desde 0 hasta 20. Representarla evolución del error en función de log(*h*).
3. Utilizar las siguientes fórmulas de tres y cuatro puntos para calcular los errores cometidos el intervalo [0,2]. Representar el error cometido tomando h=0.2. ¿Cuál es mejor?

$$f'(x_0)\approx\frac{1}{2h}\left[-3f(x_0)+4f(x_1)-f(x_2)\right],\quad f'(x_0)\approx\frac{1}{6h}\left[-2f(x_{-1})-3f(x_0)+6f(x_1)-f(x_2)\right],$$

$$f'(x_0)\approx\frac{1}{2h}\left[-f(x_{-1})+f(x_1)\right],\qquad f'(x_0)\approx\frac{1}{6h}\left[-11f(x_0)+18f(x_1)-9f(x_2)+2f(x_3)\right] \quad \text{con}$$

$x_k=x_0+kh$

1.2 Integración compuesta

MATLAB permite utilizar integración simbólica para encontrar las primitivas analíticas, si existen, de muchas funciones. Asimismo hay instrucciones específicas para polinomios. Tres alternativas para calcular con MATLAB $\int_0^1 \frac{1}{\cos(x)}dx$, utilizando integración simbólica son

```
» a=0;b=1;
» syms x real; fs=1/cos(x); F=int(fs);subs(F,b)-subs(F,a)
» F=int('1/cos(x)');subs(F,b)-subs(F,a)
» int(fs,a,b)
```

Para realizar el mismo cálculo utilizando integración numérica, se tiene

```
» fn=inline(vectorize(fs));
» quad(fn,a,b,1e-6)
» quadl(fn,a,b,1e-6)
```

Una fórmula usual de integración numérica es la fórmula del trapecio compuesta $\int_a^b f(x)dx = \frac{h}{2}\left[f(a)+2\sum_{k=1}^{n-1} f(x_k)+ f(b)\right]-\frac{h^2}{12}(b-a)f''(\zeta)$ con $h=\frac{b-a}{n}$, y donde $x_k = a+k\times h$ y $\zeta \in [a,b]$

La evaluación de la integral anterior con 10 intervalos se realiza en MATLAb mediante

```
» n=10;X=linspace(a,b,n+1);h=X(2)-X(1);
» F=fn(X);valor=(sum(2*F)-F(1)-F(end))*h/2
```

Las fórmula de integración numérica no presentan la inestabilidad vista en las de derivación. Analizando la formula anterior para números crecientes de intervalos (y utilizando la técnica de Romberg para realizar menos evaluaciones), se tiene

```
a=0;b=1;h=(b-a)/1;
X=[a,b];
F=fn(X);valor=(2*sum(F)-F(1)-F(end))*h/2
N=1:20;
for k=N
    n=2^k;h=h/2;X=a+h:2*h:b-h;
    F=fn(X);valor=valor/2+sum(F)*h;
    disp(['Equiespaciado:',num2str(h),' Integral:',num2str(valor)]);
end
```

Otra fórmula habitual para aproximar la integral es la Regla de Simpson:

$$\int_a^b f(x)dx = \frac{h}{3}\left[f(a)+4\sum_{k=1}^{n} f(x_{2k-1})+2\sum_{k=1}^{n-1} f(x_{2k})+ f(b)\right]-\frac{h^4}{180} f^{iv}(\zeta) \quad \text{con} \quad h=\frac{b-a}{2n} \quad \text{donde}$$

$x_k = a+k\times h$ y $\zeta \in [a,b]$

Ejercicios recomendados:

4. Representar la evolución del error de la regla del trapecio compuesta en función del número de intervalos 2^k con k desde 1 hasta 10, cuando se considera la función $f(x)=\frac{\sin(x)}{x}$ en [1,5]. Utilizar gráficas logarítmicas en ambos ejes.

5. Evaluar la integral $\int_0^1 \frac{1}{\cos(x)} dx$ mediante la regla de Simpson con 10 intervalos.

1.3 Integración múltiple

Las integrales dobles se resuelven simbólicamente anidándolas, p.e., $\int_0^1 \int_0^{1-x} \left(x^2 + y^2\right) dydx$

```
syms x y real;
fs2=x^2+y^2; F=int(int(fs2,'y',0,1-x),'x',0,1)
```

Para evaluarlas de forma numérica se utilizan las instrucciones *quad2d*, *dblquad*, *triplequad*, siempre que los límites de integración sean simples.

```
» Q = quad2d(@(x,y) x.^2+y.^2,0,1,0,1)
```

En caso que los límites sean complejos, se debe acudir a los métodos estadísticos de Montecarlo.

2 Anexo: Instrucciones necesarias

2.1 Derivación

diff(funsym,'x',n)	Calcula la derivada n-sima de la función simbólica "funsym" respecto a la variable x
diff(v)	Calcula las diferencias finitas adelantadas del vector v, esto es, w=∇(v).
gradient(F,h)	Calcula el gradiente de la función dada por los valores de F, con puntos equiespaciados a distancia *h*.
del2(F,h)	Calcula la laplaciana de la función dada por los valores de F, con puntos equiespaciados a distancia *h*.
jacobian(symV,var)	Calcula la matriz Jacobiana de la función vectorial dada por symV con respecto a las variables simbólicas *var*.
polyder(v)	Deriva el polinomio cuyos coeficientes vienen dados por el vector v
polyder(v,w)	Deriva el polinomio producto de v*w
[q,d]=polyder(v,w)	Deriva el polinomio cociente de v/ w

2.2 Integración

int(funsym)	Calcula la integral indefinida (primitiva) de la función simbólica "funsym"
int(funsym,a,b)	Calcula la integral definida (valor) de la función simbólica "funsym" en el intervalo [a,b]
quad(fun,a,b,tol)	Calcula la integral definida (valor) de la función "fun" en el intervalo [a,b] utilizando integración adaptativa de Simpson con error menor que "tol"
quadl(fun,a,b,tol)	Idem anterior, pero con fórmulas de Gauss-Lobato de orden superior
quad(@funfile,a,b,tol)	Idem con la función definida en el fichero funfile.m"
dblquad(fun,xa,xb,ya,yb,tol)	Calcula la integral definida doble (valor) de la función "fun" en el intervalo [xa,xb]x[ya,yb] utilizando integración adaptativa de Simpson con error menor que "tol"
polyint(v)	Integra el polinomio cuyos coeficientes vienen dados por el vector v
quad2d(fun,a,b,c,d)	Evalúa numéricamente una integral doble sobre una región plana
triplequad(fun,xmin,xmax,ymin, ymax,zmin,zmax,tol,method)	Evalúa numéricamente una integral triple sobre un recinto prismático

2.3 Gráficos logarítmicos y semilogarítmicos

loglog(X,Y)	Realiza los mismos gráficos que plot(X,Y), pero con escala logarítmica en ambos ejes.
semilogx(X,Y)	Realiza los mismos gráficos que plot(X,Y) , pero con escala logarítmica en el eje x, y escala normal en el eje y (escala semilogarítmica
semilogy(X,Y)	Realiza los mismos gráficos que plot(X,Y), pero con escala logarítmica en el eje y, y escala normal en el eje x (escala semilogarítmica)