Lino Mediavilla
23/01/2017
Mecánica Computacional – Módulo 1
Prof. Juan Baus

Deber # 1

Para el presente análisis, se escogió la función $f(x) = 3x^9$, y se aproximaron las derivadas con pasos de 0.5 y 0.0001. Se utilizaron fórmulas con un error de truncación de orden $O(h^2)$ (Chapra, 2008. p. 525-529). Estas fórmulas están diseñadas para incluir más términos de la expansión de Taylor de la función. A continuación se presentan las tablas resultantes del script que se programó en MATLAB. Se incluyen los errores porcentuales de cada aproximación con respecto a los valores exactos de las derivadas de la función escogida en el punto $x = 3$. Se hace énfasis en los resultados para el valor más pequeño de h.

Resultados

f(x) = 3*x^9

Para x = 3

Primera_Derivada_Exacta = 177147

Segunda_Derivada_Exacta = 472392

Diferencias_Centradas =

h	Primera_Derivada	Error_Primera_Derivada	Segunda_Derivada	Error_Segunda_Derivada
0.5	2.25e+05	27.015	1.4903e+06	215.49
0.0001	1.7715e+05	1.0372e-06	4.725e+05	0.023334

Hacia_Adelante =

h	Primera_Derivada	Error_Primera_Derivada	Segunda_Derivada	Error_Segunda_Derivada
0.5	-17791	110.04	-7.538e+05	259.57
0.0001	1.7715e+05	2.074e-06	4.7239e+05	9.4086e-06

Hacia_Atras =

h	Primera_Derivada	Error_Primera_Derivada	Segunda_Derivada	Error_Segunda_Derivada
0.5	1.3291e+05	24.974	2.6762e+05	43.347
0.0001	1.7715e+05	2.0734e-06	4.7239e+05	1.6719e-07

Código (Implementado en MATLAB R2013b)

```
% ********** Deber 01 de Mecánica Computacional - Módulo 1 *********
clear
clc
% --------------- Derivadas exactas por método analítico -------------

%Creación de la función
syms x;
f(x) = 3*x^9
%Obtención de las derivadas exactas de la función y
%transformación a funciones anónimas.
df_Ex = matlabFunction(diff(f,x)); %primera derivada
d2f_Ex = matlabFunction(diff(diff(f(x),x),x)); %segunda derivada
% ---------------------------------------------------------------------

% --------- Derivadas aproximadas por diferencias centradas -----------

fun = matlabFunction(f(x)); %transformación de f(x) a función anónima

% para almacenar las aproximaciones de la 1era derivada
dfcent = zeros(2,1);
dfsup = zeros (2,1);
dfinf = zeros(2,1);

% para almacenar las aproximaciones de la 2da derivada
d2fcent = zeros(2,1);
d2fsup = zeros (2,1);
d2finf = zeros(2,1);

h = [0.50; 0.0001]; % valores que serán tomados como el paso
X = 3; % punto donde se evaluarán las derivadas para compararlas

%para almacenar los errores porcentuales (1era derivada)
e1_cent = zeros(2,1);
e1_sup = zeros(2,1);
e1_inf = zeros(2,1);

%para almacenar los errores porcentuales (2da derivada)
e2_cent = zeros(2,1);
e2_sup = zeros(2,1);
e2_inf = zeros(2,1);

for i =1:2
  %1era derivada
  dfsup(i,1)= (-fun(X+2*h(i))+4*fun(X+h(i))-3*fun(X)) / (2*h(i));  %forward
  dfinf(i,1)= (3*fun(X)-4*fun(X-h(i))+fun(X-2*h(i))) / (2*h(i));   %backward
  dfcent(i,1) = (fun(X+h(i))-fun(X-h(i))) / (2*h(i));              %centered

  %2da derivada
  d2fsup(i,1)= (-fun(X+3*h(i))+4*fun(X+2*h(i))-5*fun(X+h(i))...    %forward
                  +2*fun(X)) / ((h(i))^2);
  d2finf(i,1)= (2*fun(X)-5*fun(X-h(i))+4*fun(X-2*h(i))...          %backward
                  -fun(X-3*h(i))) / ((h(i))^2);
  d2fcent(i,1) = (fun(X+2*h(i,1))-2*fun(X+h(i,1))+fun(X))...       %centered
                  / ((h(i,1))^2);



  %Error porcentual - 1era derivada
  e1_cent(i,1) = 100*abs(df_Ex(X)-dfcent(i,1)) ...
                          / df_Ex(X);
  e1_sup(i,1) =  100*abs(df_Ex(X)-dfsup(i,1)) ...
```

```
                          / df_Ex(X);
  e1_inf(i,1) =  100*abs(df_Ex(X)-dfinf(i,1)) ...
                          / df_Ex(X);


  %Error porcentual - 2da derivada
  e2_cent(i,1) = 100*abs(d2f_Ex(X)-d2fcent(i,1)) ...
                          / d2f_Ex(X);
  e2_sup(i,1) =  100*abs(d2f_Ex(X)-d2fsup(i,1)) ...
                          / d2f_Ex(X);
  e2_inf(i,1) =  100*abs(d2f_Ex(X)-d2finf(i,1)) ...
                          / d2f_Ex(X);
end

%RESULTADOS
display('Para x = 3');

%Resultado por diferenciación simbólica (Derivada exacta)
Primera_Derivada_Exacta = df_Ex(X)
Segunda_Derivada_Exacta = d2f_Ex(X)

%Resultados de diferencias centradas
Diferencias_Centradas = table(h, dfcent(:,1),e1_cent,d2fcent(:,1),e2_cent);
Diferencias_Centradas.Properties.VariableNames =
{'h','Primera_Derivada','Error_Primera_Derivada','Segunda_Derivada','Error_Segunda_Derivada'
}

%Resultados de diferencias hacia adelante
Hacia_Adelante = table(h, dfsup(:,1),e1_sup, d2fsup(:,1),e2_sup);
Hacia_Adelante.Properties.VariableNames =
{'h','Primera_Derivada','Error_Primera_Derivada','Segunda_Derivada','Error_Segunda_Derivada'
}

%Resultados de diferencias hacia atrás
Hacia_Atras = table(h, dfinf(:,1),e1_inf, d2finf(:,1),e2_inf);
Hacia_Atras.Properties.VariableNames =
{'h','Primera_Derivada','Error_Primera_Derivada','Segunda_Derivada','Error_Segunda_Derivada'
}

% ------------------------------------------------------------------------
```

Referencias

Chapra, S. C. (2008). *Applied numerical methods with MATLAB for engineers and scientists*. Boston: McGraw-Hill Higher Education.