Sexta Entrega - Métodos Numéricos

$Profesor - JobFlores.$

Autor - Fabricio Aldecosea

Fecha de entrega - Jueves 3 de Junio de 2021

Contents

Algoritmos a utilizar en los ejercicios de la entrega

Función: Secant Methods

```
function [State, x, IterationCounter, X] = secant(PreviousValue, x, IterationMaximum, ErrorMinimum)

  State = 'Success';
  IterationCounter = 0;
  X = [x];
  while (true)

    TemporaryValue = x ;
    x = x - funcion(x) * (PreviousValue - x) / ( funcion(PreviousValue) - funcion(x) );

    PreviousValue = TemporaryValue;

    if (funcion(x) == 0)
      return;
    endif
    X = [X,x];
    Ei = abs(funcion(x) - funcion(PreviousValue));
    Ed = abs(x - PreviousValue);

    IterationCounter = IterationCounter + 1;


    if (max(Ei,Ed) <= ErrorMinimum)
      return;
    elseif (IterationCounter == IterationMaximum)
      State = 'Fail';
      return;
    endif
  endwhile

endfunction
```

Función: Newton Raphson

```
function [State, x, IterationCounter, X] = newtonraphson(x, IterationMaximum, ErrorMinimum)

  State = 'Success';
  IterationCounter = 0;
  X = [x];
  while (true)

    PreviousValue = x;

    x = x - funcion(x) / funcionDerivada(x);
    X = [X,x];
    if (funcion(x) == 0)
      return;
    elseif (funcionDerivada(x) == 0)
      State = 'Fail';
      return;
    endif

    Ei = abs(funcion(x) - funcion(PreviousValue));
    Ed = abs(x - PreviousValue);

    IterationCounter = IterationCounter + 1;

    if (max(Ei,Ed) <= ErrorMinimum)
      return;
    elseif (IterationCounter == IterationMaximum)
      State = 'Fail';
      return;
    endif
  endwhile


endfunction
```

Función: Bisection

```
function [State, x, IterationCounter, X] = bisection(XL, XU, IterationMaximum, ErrorMinimum)
  x = XL;
  State = 'Success';
```

```
  IterationCounter = 0;
  X = [x];
  while (true)
    PreviousValue = x;
    x = (XL + XU) / 2;
    X = [X,x];
    Evaluation = funcion(x); # define function

    if (Evaluation == 0)
      return;
    elseif (Evaluation * funcion(XL) > 0)
      XL = x;
    elseif (Evaluation * funcion(XU) > 0)
      XU = x;
    endif

    Ei = abs(funcion(x) - funcion(PreviousValue));
    Ed = abs(x - PreviousValue);

    IterationCounter = IterationCounter + 1;



    if (max(Ei,Ed) <= ErrorMinimum)
      return;
    elseif (IterationCounter == IterationMaximum)
      State = 'Fail';
      return;
    endif
  endwhile

endfunction
```

```
clc
```

Problema 1. Considere a la función

$f(t)=(et-2t)e-t2-1.5$ Encuentre las raíces de la función usando los métodos de Bisección, Newton y Secante.

Compare el desempeño de los métodos. Utilize como condición de "exito"

$max\{Ei,Ed\}<10-8$

donde $Ei=f(tn+1)-f(tn)$, $Ed=tn+1-tn$.

Corrobore los órdenes de convergencia de los diferentes algoritmos.

```
% Problema 2. Utilizando el método de Newton aproxime la raíz de la función
% f(x1,x2)=(x21-x1x2+x22-3x1+x2-x1x2)
% partiendo de la condición inicial (x1,x2)=(-1.5,1.5). La condición de éxito es la misma que en el problema 1.
%
```

Comparacion de orden de convergencia entre los metodos de [Newton-Raphson,Secant,Bisection]

Tiempo de ejecucion de newton

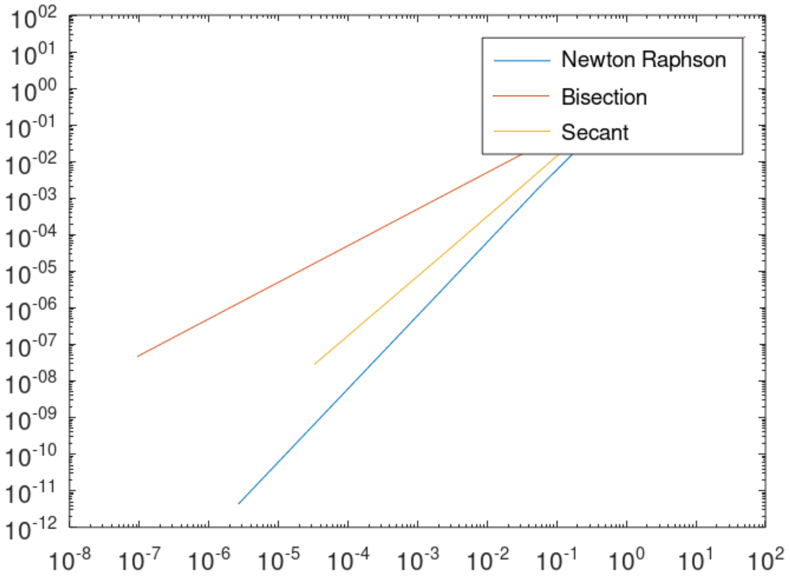
```
tic;
[State, x, IterationCounter, X] = newtonraphson(0,100,10e-8); toc
```

```
   Elapsed time is 0.00101209 seconds.
```

```
hnr = abs(diff(X));
[m,nr] = size(hnr);
%
% Tiempo de ejecucion de bisection
tic;
[State, x, IterationCounter, X] = bisection(0,100,100,10e-8); toc
%
bis = abs(diff(X));
[m,nb] = size(bis);
% Tiempo de ejecucion de secant
tic;
[State, x, IteratorCount, X] = secant(-100,100,100,10e-8); toc
hse = abs(diff(X));
[m,nse] = size(hse);

loglog(hnr(1:nr-1),hnr(2:nr),bis(1:nb-1),bis(2:nb),hse(1:nse-1),hse(2:nse));
legend('Newton Raphson','Bisection','Secant');
```

```
   Elapsed time is 0.00371099 seconds.
   Elapsed time is 0.011174 seconds.
```



REFERENCIAS

```
% [1]
% [2]
% [3]  Subject Recommended Book - Numerical Methods And Modelling Book (Khoury y Harder 2016)
% [4]  Newton method for logistic regression - https://codatalicious.medium.com/newtons-method-for-logistic-regression-4a'
```

Published with GNU Octave 6.1.0