ITESO
Universidad Jesuita
de Guadalajara

Ingeniería en Seguridad Informática y Redes

Ingeniería en Sistemas Computacionales

Algoritmos de Solución Numérica

<u>Rapshon Newton mejorado</u>

IS727272 - Marco Ricardo Cordero Hernández

SI727576 - Edgar Guzmán Claustro

Jal., 26 de septiembre de 2023

Índice

Contexto

Como segundo método numérico analizado, y retrocediendo del campo de las integrales revisado con anterioridad, en esta ocasión se propone la implementación y visualización de una famosa herramienta para la aproximación de raíces reales para funciones algebráicas y trascendentes: *el método Newton-Raphson*.

La forma en que opera este análisis numérico cuenta con múltiples etapas, una de ellas es el cálculo de la derivada de la función de la cual se desea saber alguna de sus raíces (ULPGC, s.f.). Entre otros, el objetivo de este ejercicio es el de dotar al método rudimentario con las bondades que ofrece la automatización, como lo es la obtención automática sin la intervención humana para obtener el cálculo del paso descrito.

La fórmula relevante para este método numérico (Lizarralde, 2003) es la siguiente:

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

Nota: Esta no es la única fórmula necesaria para obtener el resultado final.
La inferencia que resulta de la obtención de este elemento resulta en la necesidad de una implementación de un método iterativo, para lo cual, nuevamente se hará uso de la herramienta *Matlab* (The MathWorks Inc., 2023), software que hará mucho más fácil la visualización de los resultados de manera analítica y gráfica para cada una de sus iteraciones.

Antes de introducir formalmente el desarrollo de esta actividad, se ha de notar que el código, aunque es funcional y arroja resultados correctos, no cuenta con comprobaciones relacionadas a este método como lo son la existencia de la raíz en un rango dado, su unicidad, su concavidad, entre otras. No obstante, y como ya se ha hecho mención, lo obtenido sobrepasa las expectativas iniciales de la actividad y presenta unos datos formateados y en regla acorde a las rúbricas establecidas y el material revisado en sesiones anteriores.

Resultados

Datos de entrada

Estos valores son ingresados por el usuario. Estos se componen por la función original *f*, el valor inicial de la aproximación *X0* y el porcentaje de error *tol*. Para este ejemplo se omite el ingreso de la derivada por parte del usuario, *para calcularla de forma automática*.

Método Raphson Newton

```
function rn = RaphsoN(f,x0,tol)
%{
   f   = Función a evaluar
   x0  = Punto inicial
   tol = Tolerancia
%}

% Variables locales
rn = [];                 % Arreglo donde se almacenan las aproximaciones
syms x;                  % Variable simbólica
i = 1;                   % Contador auxiliar para iteraciones
fx(i) = x0;              % Arreglo que guarda el punto de partida
f1 = subs(f,x,fx(i));    % Evaluación de la función en base a los puntos de
partida
der = diff(f);           % Derivada
df = subs(der,x,fx(i)); % Evaluación de la derivada
ea(1) = 100;             % Error de la aproximación

% Obtener valores del método mientras que no se cumpla el error
while abs(ea(i)) >= tol
   fx(i+1) = fx(i) - f1/df;     % Se guarda la división menos la diferencia
   f1 = subs(f,x,fx(i+1));      % Evaluación con el nuevo valor de i en
función
   df = subs(der,x,fx(i+1));    % Evaluación con el nuevo valor de i en
derivada
   ea(i+1) = abs((fx(i+1)-fx(i))/fx(i+1)*100); % Cálculo del error
aproximado
   i = i + 1;                   % Siguiente iteración
end

% Muestra y almacenamiento de resultados del método
fprintf('Inter\t\tAprox\n');
for j = 1:i
   fprintf('%d\t\t%1.9f\n', j-1, fx(j));
   rn(end+1) = fx(j);  % Almacenar resultado de iteración
end
```

Gráfica

```
% Restablecer entorno
clear, clc, close all

% Definición de parámetros
syms x;                                  % Variable simbólica
```

```
func = input('Ingrese la función: ');     % Función a evaluar
x0 = input('Ingrese el valor incial: '); % Valor inicial de la aproximación
tol = input('Ingrese la tolerancia: ');  % Porcentaje de tolerancia
range = x0-5:0.1:x0+5;                   % Rango en el eje X
y_sub = subs(func, x, range);            % Rango en el eje Y

% Configuración de la gráfica
figure('name','Método Newton Raphson', ...
   'NumberTitle','off');                 % Título de ventana
title('Método Newton Raphson');          % Título de la gráfica
grid on;
hold on;
xlabel('x'), ylabel('y');                % Etiquetas de ejes
plot(range,y_sub);                       % Graficar límites

% Llamada a función y gráfica
pts = RaphsoN(func,x0,tol);              % Llamada de función

% Impresión de los puntos en la gráfica de la función
for p = 1:length(pts) - 1
   plot(pts(p), ...
        subs(func, x, pts(p)), 'r*');    % Graficar puntos
end

plot(pts(end),subs(func, x, ...
   pts(end)),'b*');                      % Gráfica del resultado
text(pts(end), subs(func,x,pts(end)+2), ...
   'Resultado','FontSize',8);            % Texto del resultado
plot([0 0], ylim, 'k-')                  % Eje de abscisas
plot(xlim, [0 0], 'k-')                  % Eje de ordenadas
```

Casos de prueba y resultados

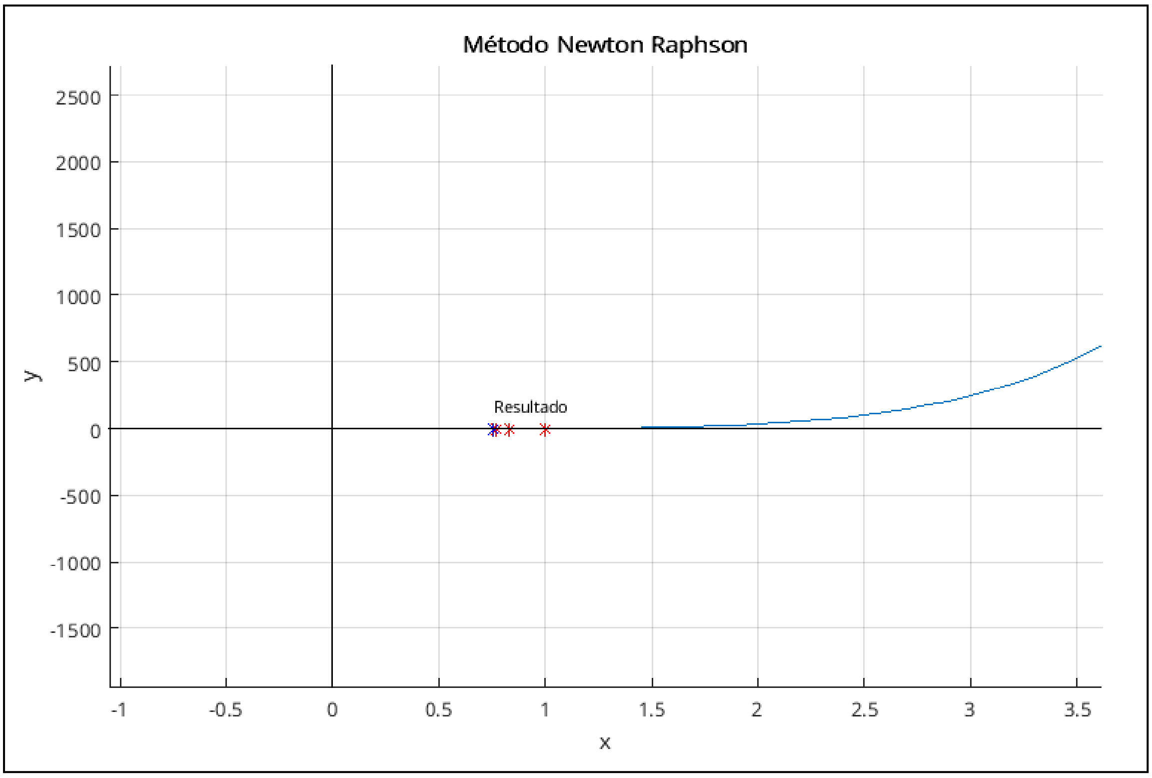
Caso 1

Probando para $x^5 + x - 1$ con un valor inicial de 1 y tolerancia de 0.00001

```
Ingrese la función: x.^5+x-1
Ingrese el valor incial: 1
Ingrese la tolerancia: 0.00001
```

Resultado

Inter	Aprox
0	1.000000000
1	0.833333333
2	0.764382116
3	0.755024867
4	0.754877702
5	0.754877666



Comprobación

Utilizando una herramienta en línea, se pueden generar comprobaciones del método.

Función	Valor inicial x0	Tolerancia deseada
x^5+x-1	1	0.00001

Iteraciones

Paso	x	F (X)	|x(i) - x(i-1)|
x1	0.8333	0.23521	0.16667
x2	0.7644	0.02533	0.06895
x3	0.7550	0.00039	0.00936
x4	0.7549	0.00000	0.00015
x5	0.7549	0.00000	0.00000

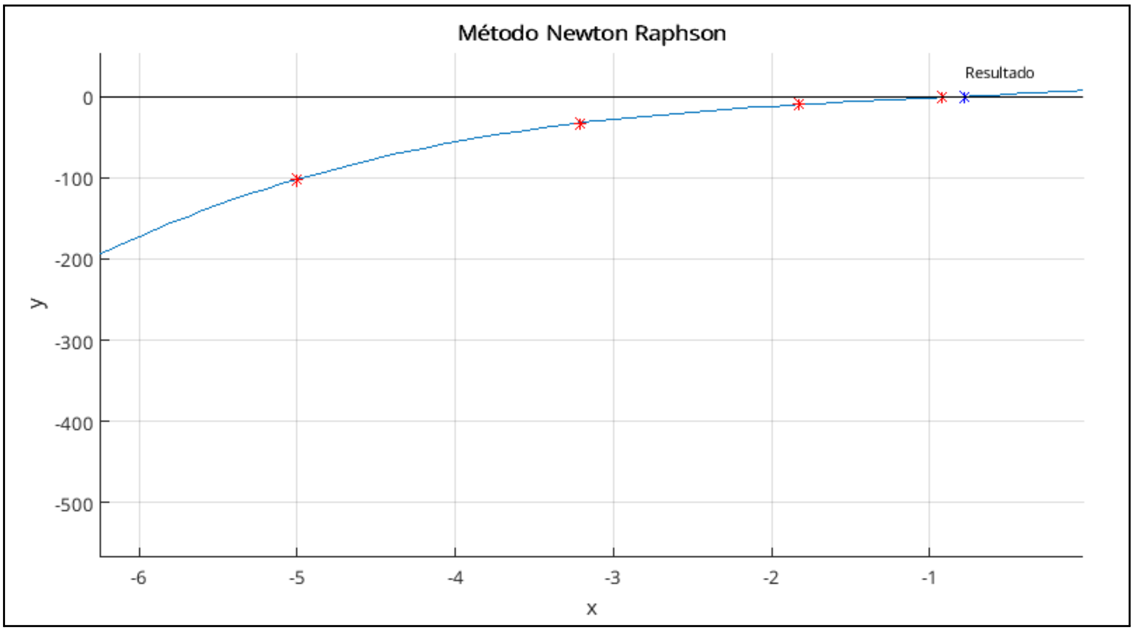
Caso 2

Probando para $x^3 + 3x^2 + 12x + 8$ con un valor inicial de -5 y tolerancia de 0.0001

```
Ingrese la función: x^3+3*x^2+12*x+8
Ingrese el valor incial: -5
Ingrese la tolerancia: 0.0001
```

Resultado

```
Inter        Aprox
0          -5.000000000
1          -3.210526316
2          -1.828560689
3          -0.922025233
4          -0.778122101
5          -0.778977409
6          -0.778977463
```



Comprobación

Utilizando la misma herramienta en línea, se pueden generar comprobaciones del método.

Función	Valor inicial x0	Tolerancia deseada
x^3+3*x^2+12*x+8	-5	0.0001

Iteraciones

Paso	x	F (X)	|x(i) - x(i-1)|
x1	-3.2105	-32.69631	1.78947
x2	-1.8286	-10.02586	1.38197
x3	-0.9220	-1.29775	0.90654
x4	-0.7781	0.00782	0.14390
x5	-0.7790	0.00000	0.00086
x6	-0.7790	0.00000	0.00000

Conclusiones

Guzmán Claustro, Edgar

La realización de este método numérico dentro de matlab, me deja como aprendizaje el conocer la importancia que tiene la herramienta y el potencial que se puede obtener de ella. En ese caso solo fue la realización de un simple método numérico, el cual cualquier persona podría calcular manualmente. El hecho de automatizar el proceso hace que la magia ocurra, pues a unos simples clicks se tiene el resultado completo sin necesidad de esperar.

Como comentarios finales, me divertí realizando esta actividad y fue de gran ayuda para repasar la sintaxis del lenguaje.

Cordero Hernández, Marco R.

Es bien sabido que la práctica hace al maestro, es por ello que la realización de este ejercicio, aún siendo sumamente similar a los dos anteriores en su más pura esencia, ha resultado benéfica tanto para el fin pedagógico de desarrollar habilidades pertinentes al curso en relación de la implantación de procesos y algoritmos automatizados, así como un manejo fundamentalmente sólido del software matemático en cuestión, el cual se ha venido manipulando desde hace ya algunas demostraciones y desarrollos.

A pesar de intuir que ya se contarían con las habilidades desarrolladas al grado de construir soluciones “al vuelo” sin siquiera realizar comprobaciones, el inminente baño de humildad eventualmente llega para demostrar que por más capaces que las personas pueden llegar a creer que son, en realidad siempre habrá cabida para la mejora, algo que sin duda alguna pudo haber sido comprobado en este ejercicio, el cual, por cierto, ha resultado sumamente interesante en cuestión de su desarrollo.

Bibliografía

Universidad de Las Palmas de Gran Canaria. (s.f.). *Métodos Numéricos 3: Raíces de ecuaciones: Métodos de Newton-Raphson y de la secante*. Recuperado de https://estadistica-dma.ulpgc.es/FCC/05-3-Raices-de-Ecuaciones-2.html.

Lizarralde, F. A. (2003). *METODO DE NEWTON-RAPHSON*. Recuperado de http://www3.fi.mdp.edu.ar/analisis/temas/no_lineales_1/newtonRaphson.

The MathWorks Inc. (2022). MATLAB version: 9.14.0 (R2023a), Natick, Massachusetts: The MathWorks Inc. https://www.mathworks.com.

Timur: miembro planetcalc. (s.f.). *Calculadora en línea: método de Newton*. Recuperado de https://es.planetcalc.com/7748/.