



Instituto Tecnológico de Costa Rica

Escuela de Escuela de Ciencia e Ingeniería de los Materiales

CM 3207 Métodos Numéricos para Ingeniería

Tarea Programada 1
Métodos numéricos Abiertos y Cerrados

Profesor: Ing. Adrián Quesada Martínez

Estudiante:

Acuña Chacón 2015183035

Fecha: 23 de marzo 2021

I semestre, 2021

Resumen

El presente trabajo consiste en aplicar métodos numéricos los cuales pueden ser útiles para resolver problemas de la vida cotidiana. Se deben de aplicar los métodos numéricos vistos en clase, los cuales son métodos de aproximación de funciones abiertos y cerrados. Los métodos de bisección y falsa posición son métodos cerrados y los métodos de Punto fijo y Newton-Raphson, son los métodos abiertos. Dos de dichos métodos deben ser implementados en el programa MatLab, en el cual se deben realizar distintos algoritmos en dicho software para la implementación de los métodos.

Pseudocodigos y explicación algoritmo

- Procedimiento del algoritmo de bisección:

En este algoritmo procedimos primero a pedir al usuario los valores de entrada establecidos (función, valor inicial, valor final y un criterio de paro independientemente de cual sea).

Como se explica en Chapra (2007), la bisección es un tipo de búsqueda incremental en el que el intervalo se divide siempre a la mitad. Utilizamos un X_n y evaluamos la función en ese valor con el fin de comparar, como se expresa en el algoritmo. X_n se calcula como la suma de los dos puntos de límites del intervalo de aproximación. Seguidamente, calculamos el error, el cual corresponde al error absoluto o aproximado.

En adición, definimos la variable iter, que define el número de iteración con el que trabajamos. Para continuar, utilizamos un ciclo while, con el fin de continuar el análisis hasta que se cumplan las condiciones definidas, que en este caso es hasta que las iteraciones alcancen el máximo indicado o que el error sea menor o igual al

definido como permitido. (También se definen un par de matrices que funcionaran para guardar los valores en una tabla)

Seguidamente, evaluamos la función en los límites; y además en la variable X_r . Con estos datos averiguamos cual es el extremo, que debe ser reemplazado por X_r , por medio del comando if else. Una vez termine el if-else, actualizamos estos datos con las fórmulas descritas anteriormente. Para finalizar, se inserta en una tabla.

- Procedimiento del algoritmo de Newton-Raphson

Para este método se le solicita al usuario que ingrese los valores de la función f con la que se trabajará en términos de x , así como el valor inicial x_i , el error que se desea o el número de iteraciones máximas que se desean realizar. Posteriormente, df será la variable que almacene el valor de la derivada de la función ingresada, y se define la variable encargada de llevar un conteo de las iteraciones que se realizan a lo largo del programa, i la cual comienza en 1 debido a que el conteo solo se llevará a cabo dentro del for, y antes de este ciclo se realiza la primera iteración. También se colocan dos Matrices donde se guardarán las tablas a posterior. Entonces se procede a calcular el valor x_1 , el error de aproximación.

Una vez que se da inicio al ciclo for, este se encargará de la segunda iteración en adelante, llevando a cabo el mismo procedimiento que se realizó para la primera iteración, se calculan los nuevos valores iniciales " x_0 " (que serán los x_1 de la iteración anterior), los de aproximaciones x_1 y errores de aproximación "error" según el método Newton-Raphson. El conteo de iteraciones "iter" aumenta cada vez que se cumple el ciclo. El ciclo for llega a su fin cuando el máximo de iteraciones ha sido sobrepasado o el error ha llegado a ser el que se buscaba para la aproximación.

Para finalizar, se inserta en una tabla.

Análisis del Procedimiento en Matlab

A pesar de que a partir de paros en el código se podía observar el funcionamiento del código como se puede observar:

```

11 CHECK%CONDICION en la que uso CHECK
    xu=xr;%sustituyo el valor por la raiz
    fu=fr;%asigno la evaluacion para proseguir la iteracion
else
    xl=xr;%sustituyo el valor por la raiz
    fl=fr;%asigno la evaluacion para proseguir la iteracion
end
F=[iter xl xu xr fl fr ff ea];%definicion de la matriz con los resultados
R=[T;F];%definicion de la matriz para la tabla
end

```

R: 1x8 sym =
[1, 2, 5/2, 5/2, 22.39, 23.755, 531.87445, 3/5]

variableNames',{'iter' 'xl' 'xu' 'xr' 'f(xl)' 'f(xr)' 'f(xl)*f(xr)' 'Ea'});
;En la linea se hace la conversion y esta importa la tabla

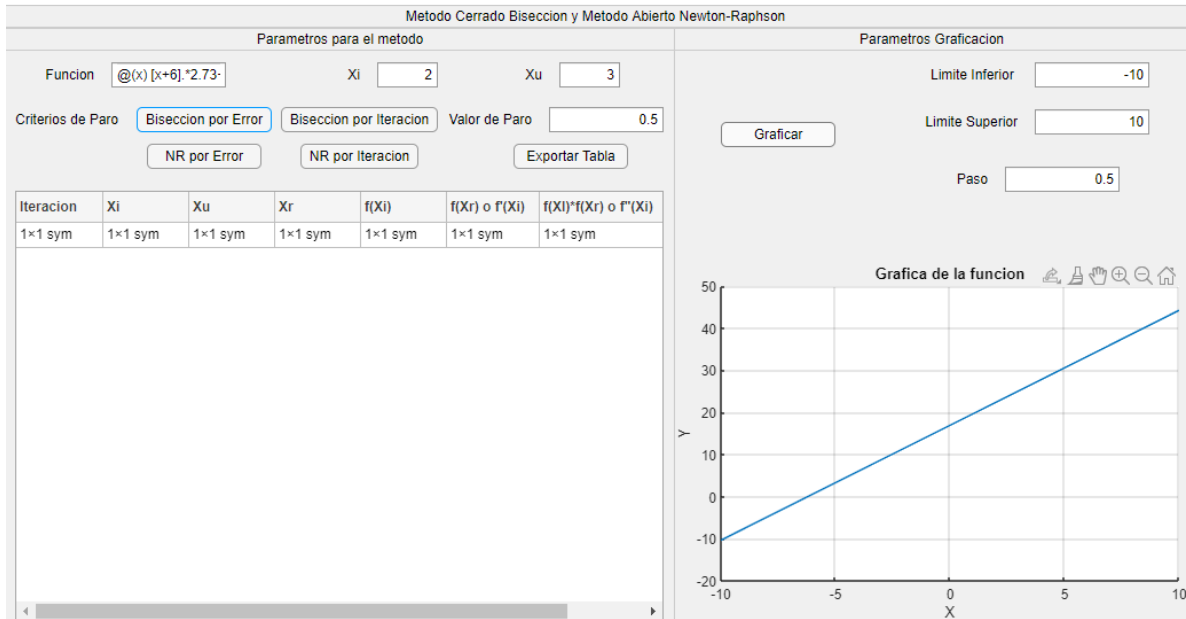
no pude arreglar este problema en la tabla de la app designer

Iteracion	Xl	Xu	Xr	f(Xl)	f(Xr)	f(Xl)*f(Xr)	Ea
1×1 sym	1×1 sym	1×1 sym	1×1 sym	1×1 sym	1×1 sym	1×1 sym	1×1 sy

Además, tuve que optar por usar botones en vez de la opción dropdown button, ya que no logre que el código corriera con esas opciones a pesar de que invertí bastante tiempo para utilizarlo así. El análisis de ambos ejercicios del Chapra no los pude adjuntar por el motivo de las tablas, no podía mostrar los valores obtenidos en pantalla.

Los códigos de la gráfica y el botón de exportar a Excel los explique en el código, aunque ya estos dos códigos fueron vistos en clase.

Diseño final de mi app designer corriendo:



Funcionamiento de la tabla en Excel(Bisección):

	A	B	C	D	E	F	G	H
1	iter	xl	xu	xr	f(xl)	f(xr)	f(xl)*f(xr)	Ea
2								
3								
4								
5								
6								

BIBLIOGRAFIA

[1]S Chapra y RP Canale. Métodos Numéricos para Ingenieros, 6ª ed. McGraw-Hill. M México, 2011.