MÉTODOS NUMÉRICOS.

TAREA 10. PSEUDOCÓDIGO PARA RESOLVER LAS RAÍCES DE UNA FUNCIÓN TRIGONOMÉTRICA POR EL MÉTODO DE BISECCIÓN..

,

Adolfo Hernández Ramírez (427560)

Correo: a.hernandezramirez3@ugto.mx.

Licenciatura Ingeniera Química Sustentable. Universidad de Guanajuato. División de Ciencias e Ingenierías. Campus León. Loma del Bosque 103, Lomas del Campestre. León, Gto, México.

Pseudocódigo.

1. Declaramos la función trigonométrica.

   - Float fx (float x) {return sin(10*x) - cos(3*x)}.

2. Declarar las variables flotantes error_max = 0.001, a_0 = 3, b_0 = 5, inc = 0.2.
3. Declarar las variables enteras i =0, N, intervalos =11.
4. Declarar los siguientes arreglos: float k[intervalos], float a[intervalos], float b[intervalos].
5. Evitar los desbordamientos de arreglos. Si N > intervalos entonces N = intervalos.
6. Determinar los pequeños intervalos a evaluar con un ciclo for i=0 hasta i < N.

   - a[i] = a_0 + i * inc;
   - b[i] = a[i] + inc;
   - k[i] = 0;

7. Con un ciclo for evaluar las funciones para cada subintervalo. For i =0 hasta i < N.

   Float fa = fx(a[i]).
   Float fb = fx(b[i]).

7.1. Si fa * fb > 0 entonces no hay cambios de signos. Dar un mensaje de salida para estos casos y continuar con el ciclo for.

8. Aplicar el método de bisección para cada subintervalo.

8.1. Declarar las siguientes variables flotantes: float a_new = a[i], float b_new = b[i]; float k_new = 0, float fk_new, float error = 0.

Donde es importante mencionar que a lo que se refiere cada variable como nueva, es porque para cada subintervalo se toman los valores de a, b y k. Y como este código evalúa de intervalo en intervalo, es por ello que estos valores cambian y en cierto modo nos recorremos en subintervalo por subintervalo y evaluamos si se encuentra una raíz en ese intervalo.

8.2. Declarar las variables enteras: iteraciones = 0.

9. Mientras error > error_max. Determinar:

9.1. k_new = (b_new + a_new) / 2.

9.2. Declarar que fk_new = fx(k_new).

9.3. Si fa * fk_new < 0. entonces b_new = k_new y fb = fk_new. Si no, entonces a_new = k_new y fa = fk_new.

9.4. Determinar el error:

Error = fabs(b_new – a_new).

9.5. Iteracciones ++.

9.6. Declarar que k[i] = k_new.

9.7. Imprimir los resultados en forma de tabla, que muestre las iteraciones, a[i], b[i], k[i], fx(k[i]), error.

10. Imprimir las raices encontradas.

10.1. for i =0 hasta i < N.

10.1.1. Si k[i] != 0, entonces se imprimen los resultados de k[i] y fx(k[i]).