

Curso de Física Computacional

Método de bisección

M. en C. Gustavo Contreras Mayén.

1. Descripción.

El siguiente programa encuentra una raíz de una ecuación mediante el método de bisección. Antes de ejecutar el programa, hay que definir la ecuación a resolver en la función **FUN**, la cual tiene un polinomio cúbico de ejemplo. Cuando el programa se ejecuta, se pide la entrada. Después de que el contador de iteración se inicializa en cero, se encuentran valores de y para $x = A$ y $y = C$ llamando a **FUN**, la cual contiene la función a resolver: **YA** y **YC** son los valores de la función en $x = A$ y $x = C$, respectivamente. Sin embargo, si $F = 0$ para $x = A$ o $x = B$, el programa se detiene.

Si el producto $YA * YC$ es positivo, el programa se va a la etiqueta 200 y se detiene después de imprimir un mensaje. Esto se debe a que no hay una raíz a encontrar cuando los signos de la función en los dos puntos extremos son iguales. Si el producto tiene un signo negativo en la etiqueta 50, el programa pasa a la etiqueta 60, donde el contador de iteraciones **IT** se incrementa en uno. El punto medio **B** se calcula en la etiqueta 70.

Se encuentra el valor de la función para $X = B$ y se guarda en **YB**. En la etiqueta 90 se determina en cuál de los dos intervalos $[A, B]$ o $[B, C]$ se encuentra la raíz. Si el producto $YA * YB$ no es positivo, el intervalo $[A, B]$ contiene la raíz, en caso contrario, el intervalo $[B, C]$ la contiene. De cualquier manera, el valor de **C** o **A** se actualiza en la etiqueta 110 y el programa regresa a la etiqueta 60 para el siguiente paso de iteración.

2. Variables

A, C: valores de x en los puntos extremos actuales.

EP: tolerancia.

IL: máximo número de pasos de iteración permitido.

IT: contador de pasos de iteración.

YA, YC: valores de la función en dos puntos extremos actuales.

F: valor de la función en x .

3. Código.

```
PROGRAM biseccion
16      PRINT *, 'Cota inferior: A ? '
        READ *, A
        PRINT *, 'Cota superior: B ? '
        READ *, B
        PRINT *, 'Tolerancia: EP ? '
        READ *, EP
        PRINT *, 'Límite de iteraciones ?'
```

```

        READ *, IL
        PRINT *, 'IT, A X1 B F(A) F(X1) F(B) ABS(F(B)-F(A))/2'

        IT=0
30      YA = FUN(A)
        YB = FUN(B)

50      IF (YA*YB .GT. 0) GOTO 200
60      IT = IT+1
70      X1=(A+B)/2
        YX1 = FUN(X1)
        PRINT 80, IT, A, X1, B, YA, YX1, YB, abs(YB-YA)/2
80      FORMAT (I4,3F9.4, 1X, 1P3E10.2, 2X, 1PE10.3)

        IF (IT .GT. IL) THEN
            PRINT *, 'Se ha excedido el limite de iteraciones'
            GOTO 205
        END IF

71      IF (abs(X1-A) .LT. EP) THEN
            PRINT *, 'Se ha satisfecho la tolerancia'
            GOTO 205
        END IF

90      IF (YA*YX1 .LE. 0) THEN
            B = X1
            YB = YX1
        ELSE
110             A = X1
            YA = YX1
        END IF

        GOTO 60

200     PRINT *, 'YA*YB es positivo'
        GOTO 210
205     PRINT *
        PRINT *, 'Resultado final: Raíz aproximada = ', X1
        PRINT *
        PRINT *
210     PRINT *, 'Presiona 1 para Continuar o 0 Para Terminar'
        READ *, KS
        IF (KS .EQ. 1) GOTO 16
        STOP
END PROGRAM biseccion
!+++++
FUNCTION FUN(X)
    FUN = X*X*X-3*X*X-X+3
    RETURN
END

```