Programa de Métodos Numéricos: Bisección y Secante

Este programa está diseñado para resolver problemas de **búsqueda de raíces** en ecuaciones no lineales, utilizando dos métodos numéricos conocidos: el *método de bisección* y el *método de la secante*. Ambos métodos permiten encontrar aproximaciones sucesivas a la raíz de una función en un intervalo determinado, lo cual es útil en situaciones donde una solución analítica exacta no es posible o es difícil de obtener.

Ahora, para sus funciones Definidas, definimos cuatro funciones matemáticas que el usuario puede elegir para resolver, cada una con características diferentes. Estas funciones son evaluadas mediante los dos métodos numéricos, y el usuario puede elegir cuál de ellas utilizara

1. $f(x) = x^2 cos(x) - 2x$
2. $f(x) = (6 - \frac{2}{x^2})(\frac{e^{2+x}}{4}) + 1$
3. $f(x) = x^3 - 3sen(x^2) + 1$
4. $f(x) = x^3 + 6x^2 + 9.4x + 2.5$

Para la implementación del ***Método de Bisección*** se implementó un enfoque iterativo que utiliza un intervalo [a, b] donde la función cambia de signo (es decir, (f(a) y f(b) tienen signos opuestos) para localizar la raíz. El programa simula una especie de hoja de cálculo, donde el usuario debe introducir los siguientes parámetros:

Su intervalo inicial [a, b]: que son sus valores iniciales entre los cuales se buscará la raíz. *Número máximo de iteraciones*, donde es el límite de pasos que realizará el algoritmo y su *Tolerancia de error que* es su precisión deseada para la aproximación de la raíz.

Ahora bien, en cada iteración, el programa divide el intervalo a la mitad (calculando el punto medio (p) y verifica si la raíz se encuentra en alguna de las mitades. Dependiendo de dónde se encuentre el cambio de signo, el programa ajusta el intervalo y repite el proceso hasta que la diferencia entre los límites del intervalo sea menor que la tolerancia especificada o hasta que se alcance el número máximo de iteraciones. Este método es particularmente robusto y garantiza convergencia siempre que la función sea continua en el intervalo dado y cambie de signo.

Para la implementación del **Método de la Secante**, es otro método numérico para encontrar raíces, pero, a diferencia del método de bisección, no requiere que la función cambie de signo en el intervalo. En lugar de dividir el intervalo, utiliza dos aproximaciones iniciales X_0 y X_1 proporcionadas por el usuario.

La fórmula principal utilizada en este método está basada en la pendiente entre los dos puntos iniciales

$$x_2 = x_1 - f(x_1)\frac{(x_1 - x_0)}{f(x_1) - f(x_0)}$$

Esta fórmula aproxima la raíz utilizando la secante de los puntos X_0 y X_1. Después de calcular \(x_2\), se repite el procedimiento usando X_1 y X_2 como nuevas aproximaciones, y así sucesivamente hasta alcanzar la tolerancia especificada o el número máximo de iteraciones. Este método puede ser más rápido que el de bisección, pero no *garantiza convergencia* si los valores iniciales no están bien elegidos o si la función no es lo suficientemente suave.

Para su **simulación del proceso** ambos métodos implementados en este programa muestran los resultados de cada iteración en una tabla, lo que facilita el seguimiento del progreso hacia la raíz. En cada iteración se muestra el valor de la aproximación actual, el valor de la función en ese punto y el error relativo o absoluto según el método.

Así que podemos determinar que este programa simplifica enormemente la tarea de resolver ecuaciones no lineales mediante los métodos numéricos de **bisección** y **secante**, haciendo el proceso mucho más fácil que si se realizara manualmente. Al permitir la introducción de parámetros como el intervalo inicial, el número máximo de iteraciones y la tolerancia de error, el programa es altamente flexible y útil para una amplia variedad de problemas de búsqueda de raíces.

Los resultados son presentados de forma clara, lo que permite a los usuarios ajustar sus parámetros y realizar cálculos precisos sin necesidad de recurrir a cálculos manuales tediosos. Este código demuestra cómo los métodos numéricos pueden ser implementados de manera eficiente para resolver problemas complejos.