Universidad Alfonso X el Sabio

Lucía Mielgo Torres, Martina García González

Grado en Ingeniería Matemática

AMPLIACIÓN DE LOS MÉTODOS NUMÉRICOS

Taller 2

28 de enero de 2025

Seguidamente debe de estar el abstract donde se resuma el trabajo realizado. El abstract no debe de contener más de 150 palabras

1. Introducción

1. Implemente una función que entregue una solución de la ecuación $f(x) = 0$ por el método de la bisección para un error dado. La funcioón tendrá como variables la función $f(x)$, los parámetros iniciales $x1$ y $x2$, y el error total ϵ.

2. Determine el valor de las racies de la función $f(x) = -\frac{1}{10}x^2 + 3$ por el método de la bisección y el número de iteraciones necesarias para obtener errores de $\epsilon = 0{,}1$; $\epsilon = 0{,}001$ y $\epsilon = 0{,}00001$

3. Implemente un fución que entregue una solución de la ecuación $f(x) = 0$ por el método de Newton-Raphson para un error dado. La función debe de aceptar de variables, la función $f(x)$, la derivada $f'(x)$, la semilla inicial p_0 y el error ϵ.

4. Determine el valor de la raíz de la función $f(x) = e^{-x} - \sin(x)$ por el método de Newton- Raphson y el número de iteraciones para los siguientes casos:

    a) Semilla inicial $P_0 = 0{,}2$ y errores de 0.1, 0.001 y 0.00001

    b) Semilla inicial $P_0 = 0{,}4$ y errores de 0.1, 0.001 y 0.00001

    c) Semilla inicial $P_0 = 10$ y errores de 0.1, 0.001 y 0.00001

    d) Semilla inicial $P_0 = 25$ y errores de 0.1, 0.001 y 0.00001

2. Métodos y modelos matemáticos

El **método de la bisección** se utiliza para encontrar una raíz c de la función $f(x)$, donde $f(c) = 0$, en un intervalo $[a, b]$. El método se basa en los siguientes pasos:

1. Dado un intervalo $[a, b]$, se asume que $f(a) \cdot f(b) < 0$, lo que implica que existe al menos una raíz en el intervalo.

2. Se calcula el punto medio del intervalo:

$$m = \frac{a + b}{2}$$

3. Se evalúa el valor de la función en m. Si $f(m) = 0$, entonces m es la raíz. De lo contrario, se procede de la siguiente manera:

   - Si $f(a) \cdot f(m) < 0$, entonces la raíz se encuentra en el intervalo $[a, m]$, y se actualiza $b = m$.

   - Si $f(m) \cdot f(b) < 0$, entonces la raíz se encuentra en el intervalo $[m, b]$, y se actualiza $a = m$.

4. El proceso se repite hasta que el intervalo $[a, b]$ sea lo suficientemente pequeño para considerar que el punto medio m es una aproximación suficientemente buena de la raíz.

El algoritmo puede detenerse cuando:

$$|b - a| < \epsilon$$

donde ϵ es una tolerancia pequeña definida previamente.

El **método de Newton-Raphson** es una forma iterativa de encontrar una raíz de la ecuación $f(x) = 0$. A diferencia del método de la bisección, es un método abierto, ya que no requiere un intervalo inicial donde la función cambie de signo. El proceso de iteración se basa en la siguiente fórmula:

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

Dada una estimación inicial x_0, se utiliza esta fórmula para obtener una mejor aproximación

x_1, luego x_2, y así sucesivamente, hasta que la diferencia entre x_n y x_{n+1} sea suficientemente pequeña.

El método converge rápidamente si la función $f(x)$ es suave y la aproximación inicial está cerca de la raíz.

3. Resultados

Ejercicio 2

En el ejercicio 2 mediante el método de la bisección, con un error de 0,1, la raíz calculada es 5.53125, tras 6 iteraciones. con un error de 0,001, la raíz calculada es 5.477783203125, tras 13 iteraciones. Con un error de 0,00001, la raíz calculada es 5.477228164672852, tras 20 iteraciones.

Ejercicio 3

a)$P_0 = 0{,}2$

	Error 1 = 0.1	**Error 2 = 0.001**	**Error 3 = 0.00001**
Iteración	2	3	4
Raíz	0.5877953224842856	0.5885325264708369	0.5885327439818422

b) $P_0 = 0{,}4$

	Error 1 = 0.1	**Error 2 = 0.001**	**Error 3 = 0.00001**
Iteración	2	3	4
Raíz	0.5884755925858224	0.5885327426745202	0.5885327439818611

c)$P_0 = 10$

	Error 1 = 0.1	**Error 2 = 0.001**	**Error 3 = 0.00001**
Iteración	2	3	4
Raíz	9.42482844904054	9.424697254739158	9.424697254738522

d)$P_0 = 25$

	Error 1 = 0.1	**Error 2 = 0.001**	**Error 3 = 0.00001**
Iteración	2	2	3
Raíz	25.132741228569152	25.132741228569152	25.132741228730506

4. Discusión

Como podemos ver en los resultados, el método de la **bisección** converge más lentamente hacia la raíz que el método de **Newton-Raphson**. Además, otra desventaja del método de la bisección es que requiere que la función cambie de signo dentro del intervalo para garantizar la existencia de la raíz, mientras que el método de Newton no tiene esta limitación.

En el método de Newton-Raphson, **no es necesario que la función cambie de signo**. Este método es abierto y se basa en la derivada de la función para iterar hacia la raíz, partiendo de una aproximación inicial. Esto le permite converger más rápido cuando la función es suave y la aproximación inicial es cercana a la raíz. Sin embargo, puede fallar si la estimación inicial no está cerca de la raíz o si la derivada de la función es cero o cambia abruptamente.

5. Conclusiones

En conclusión, el método de bisección es más robusto pero lento, mientras que el método de Newton-Raphson es más rápido, aunque depende de una buena estimación inicial para converger.

6. Referencias

Las referencias deben de estar bien identificadas en el texto cuando corresponda. Además deberan de seguir el formato APA.

7. Anexos

Las funciones utilizadas para realizar los ejercicios son las siguientes. No se ha incluido las funciones de incio de función y su derivada.

Código del método de la bisección

```
def biseccion(a, b, e, f):
    # a es el punto inicial, b es el punto final, e es el error y f es la función
    if f(a)*f(b) >= 0: # Si f(a)*f(b) es mayor o igual a cero, (a,b) no es un intervalo
        print("No hay raiz en el intervalo")
        return None
    else:
```

```
        iteracion = 0
        while b - a > e: #Calulamos nuestro error actual y lo comparamos con el error má
            c = (a + b) / 2  # c es el punto medio
            if f(c) * f(a) < 0: # Si  se cumple, la raíz está en nuestro intervalo izqui
                b = c  # renombramos c como b para tener nuestro nuevo intervalo (a,b)
            else: # Si no está en el intervalo izquierdo está en el derecho,
                a = c #Por lo tanto c es nuestra nueva a

            iteracion += 1  # vamos calculando el número de iteraciones
        print("Iteracion:", iteracion)
    print("Root:", c)
```

Código del método de Newton-Rapson

```
def NewtonR(f2, df2, x0, e):
    iteracion = 0
    error = float('inf')  # Inicializar el error como un valor grande

    while error > e:
        xn = x0 - (f2(x0) / df2(x0))
        error = abs(xn - x0)
        x0 = xn
        iteracion += 1
    print("Iteración:", iteracion)
    print("Raíz:", x0)
```