



EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



INSTITUTO TECNOLÓGICO DE COMITAN

Alumnos:

- Ruedas Velasco Pedro Eduardo__19700073.
- Hernández Méndez Levi Magdiel__19700039.
- Molina Cifuentes Adriel David__19700061.
- Panti Ordoñez Sergio Ismael__19700065.

Docente: Vera Guillen José Flavio.

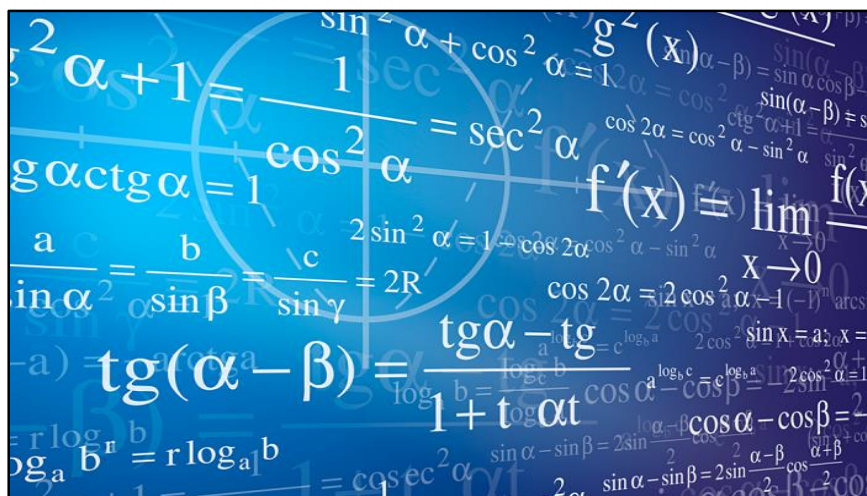
Materia: Métodos Numéricos.

Semestre: Cuarto

Grupo: "A"

Actividad: Mapa Conceptual "Métodos de solución de ecuaciones".

Comitán de Domínguez Chiapas, a 12 de Abril del 2021.





“Método de Bisección”.

Este método se utiliza para encontrar una aproximación de la raíz de una función a partir de un intervalo inicial. El método va iterando, dividiendo el intervalo a la mitad hasta encontrar un valor que se acerque lo suficiente a cero y, finalmente, obtener un aproximado de la raíz.

Método

El método de bisección consiste en tres pasos:

1. El usuario determina un grado de exactitud y elige los valores límites del intervalo, x_1 y x_2 , con la intención de que entre éstos se encuentre la raíz. Para comprobarlo se verifica que $f(x_1)f(x_2) < 0$. De resultar lo contrario, se introducirán nuevos valores hasta que se cumpla la condición y así poder continuar.

2. Se determina una aproximación de la raíz con el siguiente cálculo:

$$x_r = (x_1 + x_2) / 2$$

3. Se realizan distintas evaluaciones para determinar en qué subintervalo se encuentra la raíz:

Si $f(x_1) f(x_r) < 0$, entonces la raíz está en el inferior, por tanto, $x_2 = x_r$. Se repite el paso 2.

Si $f(x_1) f(x_r) > 0$, se sitúa en el subintervalo superior, por tanto, $x_1 = x_r$. Se repite el paso 2.

Si $f(x_1) f(x_r)$ es menor o igual al grado de exactitud, el aproximado de la raíz se encuentra en x_r . Termina el método.



"CLASE PRINCIPAL".

class Biseccion2{

/*****

Miembros datos

***/**

double a, b, tol, p;

int ni;

/******

Constructor

***/**

public Biseccion2(double a1, double b1, double tol1,

int ni1)

{

a = a1; b = b1; tol = tol1; ni = ni1;

}

/******

Función de la que se calculará la raíz

***/**

private double f(double x){

// double r = x - Math.pow(2,-x);

double r = Math.exp(x) - x*x + 3*x -2;

// double r = 2*x*Math.cos(2*x) - (x + 1)*(x + 1);

// double r = x*Math.cos(x)-2*x*x + 3*x -1;

return r;

}



```
public double calcularRaiz(){  
    double p=a;  
    int i=1;  
    double eps=1;  
    while(f(p)!=0 & i<=ni & eps>tol){  
        double pa = p;  
        p = (a+b)/2;  
        if(f(p)*f(a)>0)  
            a=p;  
        else  
            if(f(p)*f(b)>0)  
                b=p;  
        i = i + 1;  
        eps = Math.abs(p-pa)/p;  
    }  
    return p;  
}  
}
```

"CLASE MAIN"

```
class BiseccionMain2{  
    public static void main(String[] args){  
        Biseccion2 c = new Biseccion2(0,1,1e-5,20);  
        double raiz = c.calcularRaiz();  
        System.out.println("La raíz es: "+raiz);  
    }  
}
```

“Método de Falsa Posición”.

Este método, también conocido como método de interpolación lineal, es un método iterativo de resolución numérica de ecuaciones no lineales. El método combina el método de bisección y el método de la secante.

En este método el nuevo valor de prueba se encuentra por interpolación lineal entre dichos límites, tal como se muestra en la figura de arriba. Para repetir el proceso se debe ubicar el segmento donde se encuentra la solución y repetir el proceso en ese nuevo segmento, prosiguiendo de esa manera hasta que se encuentra la solución, es decir hasta que el valor de la función sea cercano a cero o hasta que los dos últimos valores de prueba sean aproximadamente iguales.

“Código de Falsa Posición”.

```
public class Falsa_Posicion {  
  
    public static double f(double x){  
  
        return -(x*x)-(2*x)+1;  
  
    }  
  
    public void posicion(){  
  
        double a, b,c;  
  
        a=Integer.parseInt(JOptionPane.showInputDialog("Digite un numero "));  
        b=Integer.parseInt(JOptionPane.showInputDialog("Digite un numero "));  
        if (f(a)*f(b)>0){  
  
            System.out.println("no se cumple que f(a)*f(b)>0, Intente con otro Intervalo");  
  
            return;  
  
        }  
  
        while(1>0){  
  
            c=((f(b)*a)-(f(a)*b))/(f(b)-f(a));  
  
            if(f(c)<=1e-10){  
  
                System.out.println("La raiz aproximada es r= "+c);  
  
                System.out.println("El valor de f en r es= "+f(c));  
  
                return;  
  
            }  
  
        }  
  
    }  
}
```



```
}  
if(f(a)*f(b)<0){  
    b=c;  
}else{  
    a=c;  
}  
}  
}  
}
```

```
public static void main(String[] args){
```

```
    Falsa_Posicion met=new Falsa_Posicion();  
    met.posicion();
```