

Ensenada, Baja California a 05 de noviembre del 2020



Practica No.6

Métodos de Newton-Rhapson y de la secante

Profesor: Roilhi Frajo Ibarra Hernández

Alumno: Fabian Diaz Fajardo

Grupo: 021

Desarrollo de la práctica.

- Desarrollar y codificar en C los métodos de Newton-Raphson y de la secante
- Encontrar las ventajas y desventajas de los métodos abiertos de bisección para el cálculo de las raíces.

Función 1.

Método de la secante

```
#include <stdio.h>
#include <math.h>
float f(float x)
{
    float y;
    y= (8*sin(x)*exp(-x))-1;
    return y;
}
float secante(float x0,float x1, int iter)
{
    int i=1;
    float xi, ximas1,ximenos1,erp;
    float fprima;
    printf("i \t Xi \t Xi-1 \t Xi+1 \t ~f'(x) \t Err\n\n");
    xi=x1;
    ximenos1=x0;
    do
    {
        ximas1= xi - (f(xi)*(ximenos1-xi))/(f(ximenos1)-f(xi));
        erp= fabs((ximas1-xi)/ximas1)*100;
        fprima= (f(ximenos1)-f(xi))/(ximenos1-xi);
        printf("%i \t %.8f \t %.8f \t %.8f \t %.8f \t %.8f\n",i,xi,ximenos1,ximas1,fprima,erp);
        ximenos1=xi;
        xi=ximas1;
        i++;
    }while(i<=iter);
    return ximas1;
}
int main ()
{
    float x0=0.3, x1=0.5,raiz;
    int iter=5;
    raiz=secante(x0,x1,iter);
    return 0;
}
```

i	Xi	Xi-1	Xi+1	~f' (x)	Err
1	0.50000000	0.30000001	0.03858230	2.87438107	1195.93127441
2	0.03858230	0.50000000	0.19844407	4.39815712	80.55759430
3	0.19844407	0.03858230	0.15138994	6.23270893	31.08140755
4	0.15138994	0.19844407	0.14459632	5.44636822	4.69833374
5	0.14459632	0.15138994	0.14501795	5.80675316	0.29074219

Método de newton-Raphson.

```
#include <stdio.h>
#include <math.h>
float f(float x)
{
    float y;
    y= (8*sin(x)*exp(-x))-1;
    return y;
}
float fprima(float x)
{
    float y;
    y = (8*exp(-x)*cos(x))-(8*exp(-x)*sin(x));
    return y;
}
float newtonRaphson(float X0,int iter)
{
    int i=0;
    float xi, ximasuno,erp;
    xi=X0;
    printf("i \t Xi \t\t Xi+1 \t\t Err\n");
    do
    {
        ximasuno= xi - f(xi)/fprima(xi);
        erp= fabs((ximasuno-xi)/ximasuno)*100;
        printf("%i \t %.8f \t %.8f \t %.8f \n",i,xi,ximasuno,erp);
        i++;
        xi=ximasuno;
    }while(i<iter);
    return ximasuno;
}
int main ()
{
    float X0=0.3;
    int iter=5;
    float raiz;
    raiz=newtonRaphson(X0,iter);
    printf("\n\n");
    return 0;
}
```

i	Xi	Xi+1	Err
0	0.30000001	0.10784370	178.18038940
1	0.10784370	0.14348660	24.84058189
2	0.14348660	0.14501208	1.05196512
3	0.14501208	0.14501481	0.00188044
4	0.14501481	0.14501481	0.00000000

Función 2.

Método de la secante.

```
#include <stdio.h>
```

```
#include <math.h>
```

```
float f(float x)
```

```
{
```

```
    float y;
```

```
    y= 2*x*x*x-11*x*x+17.7*x-5;
```

```
    return y;
```

```
}
```

```
float secante(float x0,float x1, int iter)
```

```
{
```

```
    int i=1;
```

```
    float xi, ximasuno,ximenosuno,erp;
```

```
    float fprima;
```

```
    printf("i \t Xi \t Xi-1 \t Xi+1 \t ~=f'(x) \t Err\n");
```

```
    xi=x1;
```

```
    ximenosuno=x0;
```

```
    do
```

```
    {
```

```
        ximasuno= xi - (f(xi)*(ximenosuno-xi))/(f(ximenosuno)-f(xi));
```

```
        erp= fabs((ximasuno-xi)/ximasuno)*100;
```

```
        fprima= (f(ximenosuno)-f(xi))/(ximenosuno-xi);
```

```
        printf("%i \t %.8f \t %.8f \t %.8f \t %.8f\n",i,xi,ximenosuno,ximasuno,fprima,erp);
```

```
        ximenosuno=xi;
```

```
        xi=ximasuno;
```

```
        i++;
```

```
    }while(i<=iter);
```

```
    return ximasuno;
```

```
}
```

```
int main ()
```

```
{
```

```
    float x0=0.3, x1=0.5,raiz;
```

```
    int iter=5;
```

```
    raiz=secante(x0,x1,iter);
```

```
    return 0;
```

```
}
```

i	Xi	Xi-1	Xi+1	~=f' (x)	Err
1	0.50000000	0.30000001	0.36336032	9.88000011	37.60446167
2	0.36336032	0.50000000	0.35531259	9.33045864	2.26497173
3	0.35531259	0.36336032	0.35625592	10.56937695	0.26479080
4	0.35625592	0.35531259	0.35625035	10.63223362	0.00156436
5	0.35625035	0.35625592	0.35625032	10.59839630	0.00000837

Método de newton-Raphson.

```
#include <stdio.h>
#include <math.h>
float f(float x)
{
    float y;
    y= 2*x*x*x-11*x*x+17.7*x-5;
    return y;
}
float fprima(float x)
{
    float y;
    y=6*x*x-22*x+17.7;
    return y;
}
float newtonRaphson(float X0,int iter)
{
    int i=0;
    float xi, ximasuno,erp;
    xi=X0;
    printf("i \t Xi \t Xi+1 \t Err\n");
    do
    {
        ximasuno= xi - f(xi)/fprima(xi);
        erp= fabs((ximasuno-xi)/ximasuno)*100;
        printf("%i \t %.8f \t %.8f \t %.8f \n",i,xi,ximasuno,erp);
        i++;
        xi=ximasuno;
    }while(i<iter);
    return ximasuno;
}
int main ()
{
    float X0=0.3;
    int iter=5;
    float raiz;
    raiz=newtonRaphson(X0,iter);
    printf("\n\n");
    return 0;
}
```

i	Xi	Xi+1	Err
0	0.30000001	0.35378006	15.20154953
1	0.35378006	0.35624525	0.69199193
2	0.35624525	0.35625032	0.00142214
3	0.35625032	0.35625032	0.00000000
4	0.35625032	0.35625032	0.00000000

Comparar el error absoluto para 5 iteraciones o más si es posible entre los dos métodos para las dos funciones.

$$F1(x)=8\sin(x)e^{-x}-1y$$

Método de Newton-Raphson	Regla de la Secante
178.18	1195.93
24.84	80.55
1.05	31.08
0.001	4.69
0.000	0.29

$$F2(x)=f(x)=2*3-11x^2+17.7x-5$$

Método de Newton-Raphson	Regla de la Secante
1.92	15.20
21.81	0.6919
58.64	0.001
54.06	0.000
337.49	0.000