# INSTITUTO POLITÉCNICO NACIONAL

## UNIDAD PROFESIONAL INTERDISCIPLINARIA DE BIOTECNOLOGÍA

### METODOS NUMERICOS

**Reporte y Tarea 3**

**PUNTO FIJO Y NEWTON RAPHSON**

**1er Parcial**

## GRUPO: 4FM4


**INTEGRANTES:**

**ALVARES ZEPEDA NALLELY ALITZEL**

**FLORES MOSQUEDA EDUARDO**

**GARCIA CRUZ JOSUE**

**MARTINEZ MONTEJO H. DAVID**

**MENDOZA OROZCO MARICRUZ**


**ENTREGA: 09/09/2021**


**PROFESORES:**

**GRANADOS HERNANDEZ JESUS**

**FLORES NUÑEZ JOSE IGNACIO**

# Reporte

## Introducción

Los métodos abiertos utilizan una fórmula para predecir la raíz. Esta fórmula puede desarrollarse como una iteración simple de punto fijo (también llamada iteración de un punto o sustitución sucesiva o método de punto fijo).

Las raíces multiples

Las raíces múltiples son determinados de ecuaciones polinómicas que tienen la forma general:fx = a0 + a1x + a2x2 + ... + anxn
Donde n es el grado del polinomio y son los coeficientes. Las raíces de los polinomios pueden ser reales y / o complejos, y cumplir con las tres reglas.

En una ecuación de grado n, hay n raíces reales o complejas. Cabe señalar que las raíces no son necesariamente diferentes. Si n es impar hay al menos una raíz real.Si hay raíces complejas, estas se encuentran en pares conjugados.

Método de Punto Fijo

El método de punto fijo o de aproximaciones sucesivas es, junto con el de la Bisección, uno de los primeros métodos que se utilizaron para resolver ecuaciones algebraicas y trascendentes. No obstante que en la actualidad existen otros métodos más eficientes, el de punto fijo se considera el más simple en sus principios y en él se pueden apreciar claramente todas las características de un método de aproximaciones sucesivas.

El método de Newton

Es un método abierto, en el sentido de que no está garantizada su convergencia global. La única manera de alcanzar la convergencia es seleccionar un valor inicial lo suficientemente cercano a la raíz buscada. Así, se ha de comenzar la iteración con un valor razonablemente cercano al cero (denominado punto de arranque o valor supuesto). La relativa cercanía del punto inicial a la raíz depende mucho de la naturaleza de la propia función; si ésta presenta múltiples puntos de inflexión o pendientes grandes en el entorno de la raíz, entonces las probabilidades de que el algoritmo diverge aumentan, lo cual exige seleccionar un valor supuesto cercano a la raíz. Una vez que se ha hecho esto, el método linealiza la función por la recta tangente en ese valor supuesto. La abscisa en el origen de dicha recta será, según el método, una mejor aproximación de la raíz que el valor anterior. Se realizarán sucesivas iteraciones hasta que el método haya convergido lo suficiente

## Objetivo:

Aplicar el método de Newton Raphson para obtener raíces con una aproximación en cifras esperadas. Así como aplicar el método de punto fijo para la obtención de raíces de funciones de forma f(x)=0.

Determinar los errores y la forma de convergencia para el método de punto fijo y Newton Raphson.

# Contenido:

## EJEMPLO 1

**Para la siguiente función, obtén la raíz más cercana a 0, con un error absoluto <= 0.01**

$$f(x) = x^2 - x - 4$$

Despejes:

X0=-1

1. $x = \sqrt{x+4}$ Apto
2. $x = x^2 - 4$ No apto
3. $x = \frac{4}{x-1}$ Mas Apto

| it | xi | error |
|---|---|---|
| 1 | -2 | 1 |
| 2 | -1.3333 | 0.6667 |
| 3 | -1.7143 | 0.3810 |
| 4 | -1.4737 | 0.2406 |
| 5 | -1.6170 | 0.1433 |
| 6 | -1.5285 | 0.0886 |
| 7 | -1.5820 | 0.0535 |
| 8 | -1.5492 | 0.0328 |
| 9 | -1.5691 | 0.0199 |
| 10 | -1.5569 | 0.0122 |
| 11 | -1.5644 | 0.0074 |

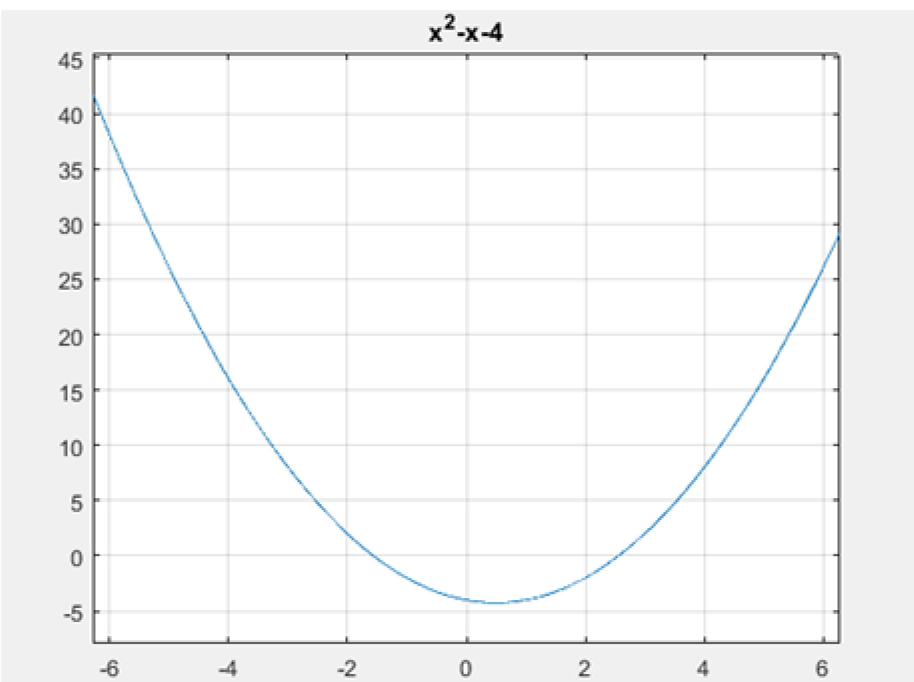
La raíz más cercana a 0 es -1.5644 con un error absoluto de 0.0074.

# Código:

```
clc, clear, close all
%Punto fijo
%Ejemplo 1
f=inline('x^2-x-4');
ezplot(f)
grid on
x0=-1;%Valor inicial
%Despeje 2
g=inline('4/(x-1)');
syms x
k=subs(diff(g(x),1),x0);
k=vpa(k,5)%Apto
%Iteracion 1
x1=g(x0)
er=abs(x1-x0)
%Iteracion 2
x2=g(x1)
er=abs(x2-x1)
%Iteracion 3
x3=g(x2)
er=abs(x3-x2)
%Iteracion 4
x4=g(x3)
er=abs(x4-x3)
%Iteracion 5
x5=g(x4)
er=abs(x5-x4)
%Iteracion 6
x6=g(x5)
er=abs(x6-x5)
```

```
%Iteracion 7
x7=g(x6)
er=abs(x7-x6)
%Iteracion 8
x8=g(x7)
er=abs(x8-x7)
%Iteracion 9
x9=g(x8)
er=abs(x9-x8)
%Iteracion 10
x10=g(x9)
er=abs(x10-x9)
%Iteracion 11
x11=g(x10)
er=abs(x11-x10)
```

Grafica:




## EJEMPLO 2

**Para la siguiente función, obtén la raíz positiva, con un error absoluto <= 0.01**

$$f(x) = x^2 - 2x - 3$$

Despejes:

1. $x = \frac{x^2-3}{2}$ No apto X0=3

2. $x = \frac{3}{x-2}$ No apto

3. $x = \sqrt{2x+3}$ Apto

| it | xi | error |
|---|---|---|
| 1 | 3.3166 | 0.6834 |
| 2 | 3.1037 | 0.2129 |
| 3 | 3.0344 | 0.0694 |
| 4 | 3.0114 | 0.0229 |
| 5 | 3.0038 | 0.0076 |

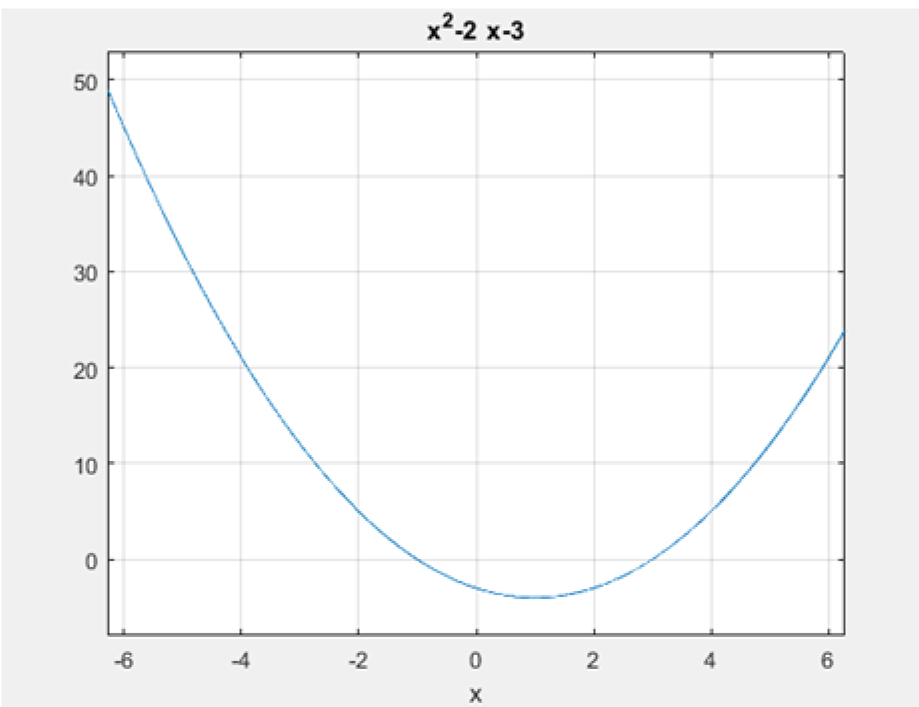
La raíz más cercana a 3.0038 con un error absoluto de 0.0076.

Código:

```
clc, clear, close all
%Punto fijo
%Ejemplo 2
f=inline('x^2-2*x-3');
ezplot(f)
grid on
x0=4;%Valor inicial
g=inline('sqrt(2*x+3)');
syms x
k=subs(diff(g(x),1),x0);
k=vpa(k,5)%Apto
%Iteracion 1
x1=g(x0)
```

```
er=abs(x1-x0)
%Iteracion 2
x2=g(x1)
er=abs(x2-x1)
%Iteracion 3
x3=g(x2)
er=abs(x3-x2)
%Iteracion 4
x4=g(x3)
er=abs(x4-x3)
%Iteracion 5
x5=g(x4)
er=abs(x5-x4)
```

Grafica:



## EJEMPLO 3

**Encuentra la raíz positiva más cercana a 1 con un error absoluto <= 0.01**

$$f(x) = cos(x) - 2^{-x}$$

Despejes:

$x{=}\cos(x){-}2^{-x}$

| it | xi | error |
|---|---|---|
| 1 | 1.0403 | 0.0403 |
| 2 | 1.0600 | 0.0197 |
| 3 | 1.0693 | 0.0092 |

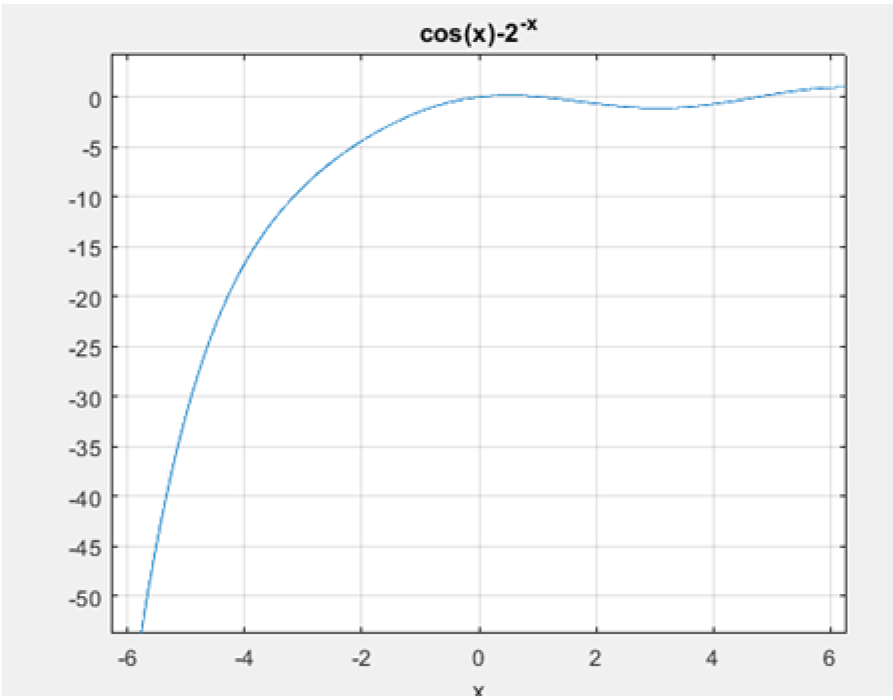
La raíz más cercana a 0 es 1.0693 con un error absoluto de 0.0092

Código:

```
clc, clear, close all

%Punto fijo

%Ejemplo 3

f=inline('cos(x)-2^(-x)');

ezplot(f)

grid on

x0=1;%Valor inicial

%Despeje 1

g=inline('cos(x)-2^(-x)+x');

syms x

k=subs(diff(g(x),1),x0);

k=vpa(k,5)%Apto

%Iteracion 1

x1=g(x0)

er=abs(x1-x0)

%Iteracion 2

x2=g(x1)

er=abs(x2-x1)
```

```
%Iteracion 3
x3=g(x2)
er=abs(x3-x2)
```

Grafica:




# NEWTON - RAPHSON

## EJEMPLO 1

**Para la siguiente función, obtén su raíz con error absoluto <= 0.0001**

$$f(x) = ln(x) - 4.5$$

| it | xi | error |
|---|---|---|
| 1 | 31.974 | 21.974 |
| 2 | 65.07 | 33.096 |
| 3 | 86.187 | 21.118 |
| 4 | 89.934 | 3.747 |

| | | |
|---|---|---|
| 5 | 90.017 | 0.082606 |
| 6 | 90.01713 | 0.00003794875 |

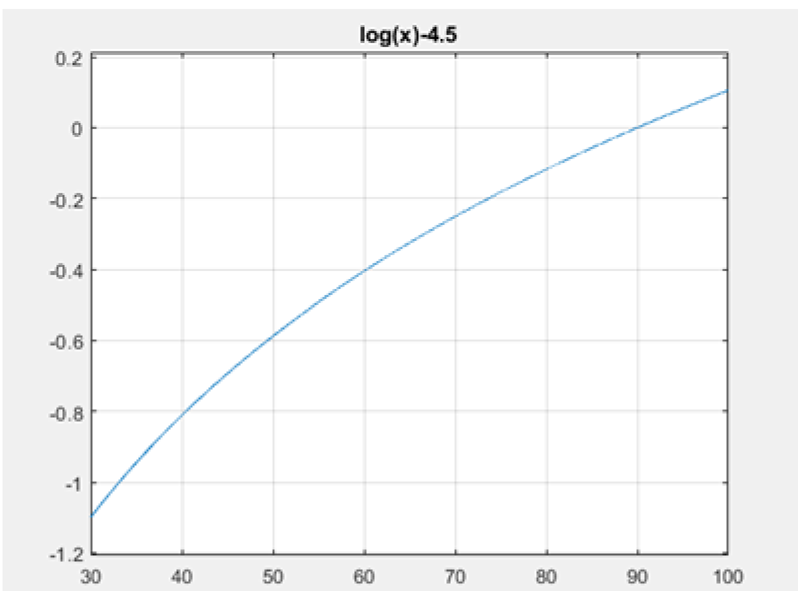
La raíz más cercana a 0 es 90.01713 con un error absoluto de 0.00003794875

Código:

```
clc, clear, close all
%Newton-Raphson
%Ejemplo 1
f=inline('log(x)-4.5');
ezplot(f,[30,100])
grid on
x0=10;
syms x
%Iteracion 1
x1=x0-f(x0)/subs(diff(f(x),1),x0);
x1=vpa(x1,5)
er=abs(x1-x0);
er=vpa(er,5)
%Iteracion 2
x2=x1-f(x1)/subs(diff(f(x),1),x1);
x2=vpa(x2,5)
er=abs(x2-x1);
er=vpa(er,5)
%Iteracion 3
x3=x2-f(x2)/subs(diff(f(x),1),x2);
x3=vpa(x3,5)
er=abs(x3-x2);
er=vpa(er,5)
%Iteracion 4
x4=x3-f(x3)/subs(diff(f(x),1),x3);
x4=vpa(x4,5)
er=abs(x4-x3);
```

```
er=vpa(er,5)
%Iteracion 5
x5=x4-f(x4)/subs(diff(f(x),1),x4);
x5=vpa(x5,5)
er=abs(x5-x4);
er=vpa(er,5)
%Iteracion 6
x6=x5-f(x5)/subs(diff(f(x),1),x5);
x6=vpa(x6,7)
er=abs(x6-x5);
er=vpa(er,7)
```

Grafica:

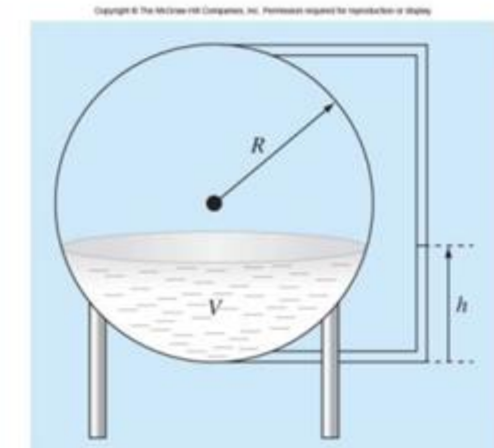



# TAREA:

## PROBLEMA 1

Suponga que está diseñado un tanque esférico (vea la figura abajo) para almacenar agua para un poblado pequeño en un país en desarrollo. El volumen de líquido que puede contener se calcula con:

$$V = \pi h^2 \frac{[3R - h]}{3}$$




Donde V = volumen [$m^3$], h = profundidad del agua en el tanque [m], R = radio del tanque [m].

Si R = 3 m, ¿a qué profundidad debe llenarse el tanque de modo que contenga 30 $m^3$?

Aplica el Método de Newton-Raphson para encontrar dicha aproximación, con un valor inicial de x0 = 0.5, con un error de 0.0001.

Solucion:

Sustituimos x por h:

x=h;

Y la funcion para encontrar el valor de x donde la funcion es igual a 30, despejamos para que la funcion quede igual a 0:

$$30 = \pi h^2 (\frac{(3*3)-h}{3})$$

$$0 = \pi h^2 \left(\frac{9-h}{3}\right) - 30$$

| it | xi | error |
|---|---|---|
| 1 | 3.7149 | 3.2149 |
| 2 | 1.9758 | 1.7391 |
| 3 | 2.0272 | 0.051428 |

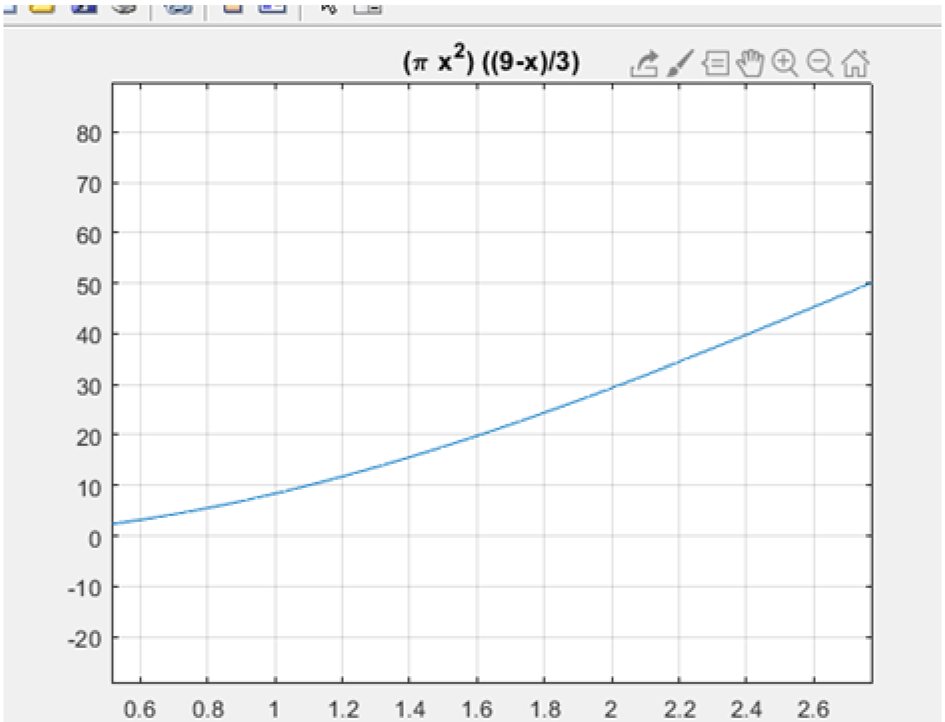
| | | |
|---|---|---|
| 4 | 2.0269 | 0.00033071 |
| 5 | 2.026906 | 0.0000000132125 |

La profundidad del tanque es a 2.026906 m para tener un volumen de 30 $m^3$

Código:

```
clc, clear, close all

%Newton-Raphson

%Problema 1

f=inline('((pi*x^2)*((9-x)/3))-30');

ezplot(f)

grid on

x0=0.5;

syms x

%Iteracion 1

x1=x0-f(x0)/subs(diff(f(x),1),x0);

x1=vpa(x1,5)

er=abs(x1-x0);

er=vpa(er,5)

%Iteracion 2

x2=x1-f(x1)/subs(diff(f(x),1),x1);

x2=vpa(x2,5)

er=abs(x2-x1);

er=vpa(er,5)

%Iteracion 3

x3=x2-f(x2)/subs(diff(f(x),1),x2);

x3=vpa(x3,5)

er=abs(x3-x2);

er=vpa(er,5)

%Iteracion 4

x4=x3-f(x3)/subs(diff(f(x),1),x3);

x4=vpa(x4,5)

er=abs(x4-x3);
```

```
er=vpa(er,5)
%Iteracion 5
x5=x4-f(x4)/subs(diff(f(x),1),x4);
x5=vpa(x5,7)
er=abs(x5-x4);
er=vpa(er,9)
```




## EJERCICIO 2

Aplicando el Método de Newton-Raphson, encontrar una raíz próxima a x0 = 0.6, con una precisión de $10^{-5}$ para la función:

$$f(x) = \ln(x^2 + 1) - e^{x/2}\cos(\pi x)$$

| it | xi | error |
|---|---|---|
| 1 | 0.45859 | 0.14141 |
| 2 | 0.45256 | 0.0060287 |
| 3 | 0.45253 | 0.00002649 |

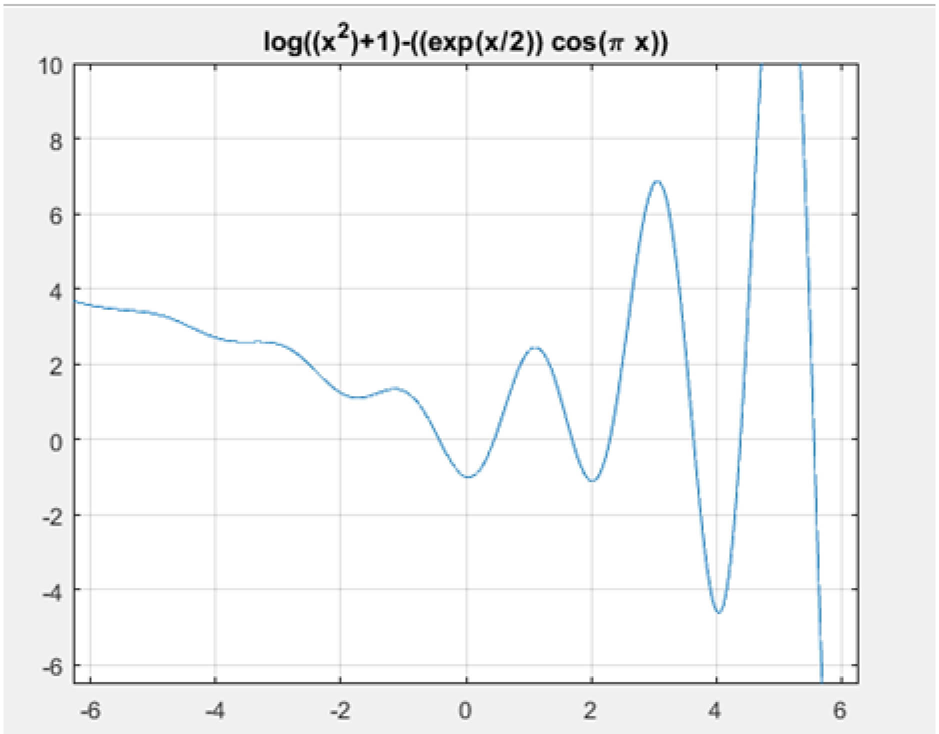
| | | |
|---|---|---|
| 4 | 0.4525311 | $5.226x10^{-10}$ |

La raiz mas cercana a 0 es 0.4525311 con un error absoluto de $5.226x10^{-10}$

## CÓDIGO

```
clc, clear, close all
%Newton-Raphson
%Problema 1
f=inline('log((x^2)+1)-((exp(x/2))*cos(pi*x))');
ezplot(f)
grid on
x0=0.6;
syms x
%Iteracion 1
x1=x0-f(x0)/subs(diff(f(x),1),x0);
x1=vpa(x1,5)
er=abs(x1-x0);
er=vpa(er,5)
%Iteracion 2
x2=x1-f(x1)/subs(diff(f(x),1),x1);
x2=vpa(x2,5)
er=abs(x2-x1);
er=vpa(er,5)
%Iteracion 3
x3=x2-f(x2)/subs(diff(f(x),1),x2);
x3=vpa(x3,5)
er=abs(x3-x2);
er=vpa(er,5)
%Iteracion 4
x4=x3-f(x3)/subs(diff(f(x),1),x3);
x4=vpa(x4,7)
er=abs(x4-x3);
er=vpa(er,9)
```

# GRÁFICA




### EJERCICIO 3

Considérese el polinomio $p(x) = x^4 + 3x^3 - 2$ del que queremos **obtener su raíz cercana a x0=1**. Por medio del método de <u>Punto Fijo</u> obtén su raíz con un **error menor a 0.001**. Calcule todas las iteraciones necesarias.

# CÓDIGO

```
clc,clear, close all

%punto fijo

%problema 3

f=inline('x^4+3*x^3-2');

ezplot(f)

grid on

x0=1; %valor inicial

%Despeje 1

g=inline('((x^4)+(3*x^3)-(2)+(x^70))^(1/70)');

syms x
```

```
a=abs(subs(diff(g(x),1),x0));
a=vpa(a,5)
%it 1
x1=g(x0)
ea=abs(x1-x0)
%it2
x2=g(x1)
ea=abs(x2-x1)
%it3
x3=g(x2)
ea=abs(x3-x2);
%it4
x4=g(x3)
ea=abs(x4-x3)
%it5
x5=g(x4)
ea=abs(x5-x4)
%it6
x6=g(x5)
ea=abs(x6-x5)
%it7
x7=g(x6)
ea=abs(x7-x6)
%it8
x8=g(x7)
ea=abs(x8-x7)
%it9
x9=g(x8)
ea=abs(x9-x8)
%it10
x10=g(x9)
ea=abs(x10-x9)
```

```
%it11
x11=g(x10)
ea=abs(x11-x10)
%it12
x12=g(x11)
ea=abs(x12-x11)
%it13
x13=g(x12)
ea=abs(x13-x12)
%it4
x14=g(x13)
ea=abs(x14-x13)
%it15
x15=g(x14)
ea=abs(x15-x14)
%it16
x16=g(x15)
ea=abs(x16-x15)
%it17
x17=g(x16)
ea=abs(x17-x16)
```

## RESULTADOS

| it | xi | Error |
|---|---|---|
| 1 | 1.0158 | 0.0158 |
| 2 | 1.0239 | 0.0080 |
| 3 | 1.0293 | 0.0060 |
| 4 | 1.0333 | 0.0041 |
| 5 | 1.0366 | 0.0033 |
| 6 | 1.0393 | 0.0027 |
| 7 | 1.0417 | 0.0023 |
| 8 | 1.0437 | 0.0021 |
| 9 | 1.0455 | 0.0018 |
| 10 | 1.0472 | 0.0016 |
| 11 | 1.0487 | 0.0015 |
| 12 | 1.0500 | 0.0014 |

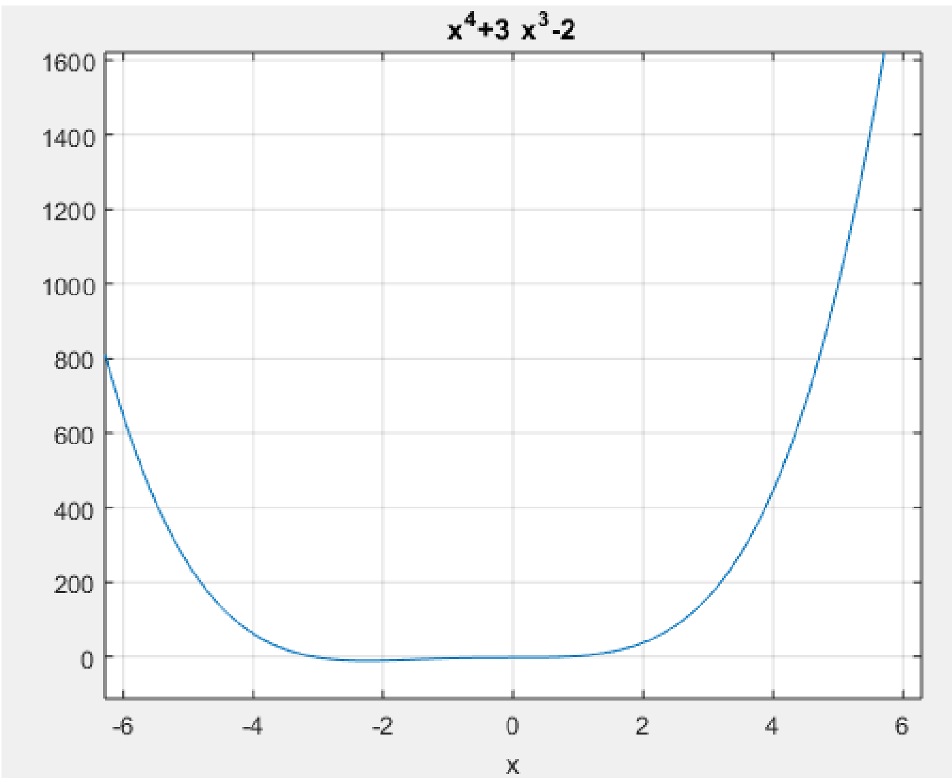
| | | |
|---|---|---|
| 13 | 1.0513 | 0.0013 |
| 14 | 1.0525 | 0.0012 |
| 15 | 1.0536 | 0.0011 |
| 16 | 1.0546 | 0.0010 |
| 17 | 1.0556 | 9.7209e-04 |

Iteraciones necesarias: 17

Raiz obtenida: 1.0556

Error: 9.7209e-04

GRÁFICO




## EJERCICIO 4

La concentración de una bacteria contaminante en un líquido disminuye a lo largo del tiempo según la expresión:

$$c(t) = 80\, e^{-2t} + 20\, e^{-0.5t},$$

siendo ***t*** el tiempo en horas. Por medio del métodos de **Newton-Raphson** determina el tiempo necesario que se necesita para que el número de bacterias se reduzca a 7. Con un error menor a 0.0001. Calcular todas las iteraciones necesarias.

## CÓDIGO

```
clear all, close all, clc
%Newton-R Problema 4
c=inline('(80*exp(-2*x))+(20*exp(-0.5*x))-7');
ezplot(c);
grid on
x0=0%Valor inicial
syms x
%It1
x1=x0-c(x0)/subs((diff(c(x),1)),x0);
x1=vpa(x1,5)
er=abs(x1-x0);
er=vpa(er,5)
%It2
x2=x1-c(x1)/subs((diff(c(x),1)),x1);
x2=vpa(x2,5)
er=abs(x2-x1);
er=vpa(er,5)
%It3
x3=x2-c(x2)/subs((diff(c(x),1)),x2);
x3=vpa(x3,5)
er=abs(x3-x2);
er=vpa(er,5)
%It4
x4=x3-c(x3)/subs((diff(c(x),1)),x3);
```

```
x4=vpa(x4,5)
er=abs(x4-x3);
er=vpa(er,5)
%It5
x5=x4-c(x4)/subs((diff(c(x),1)),x4);
x5=vpa(x5,5)
er=abs(x5-x4);
er=vpa(er,5)
%It6
x6=x5-c(x5)/subs((diff(c(x),1)),x5);
x6=vpa(x6,5)
er=abs(x6-x5);
er=vpa(er,5)
%It7
x7=x6-c(x6)/subs((diff(c(x),1)),x6);
x7=vpa(x7,5)
er=abs(x7-x6);
er=vpa(er,5)
%It8
x8=x7-c(x7)/subs((diff(c(x),1)),x7);
x8=vpa(x8,5)
er=abs(x8-x7);
er=vpa(er,5)
```

## RESULTADOS

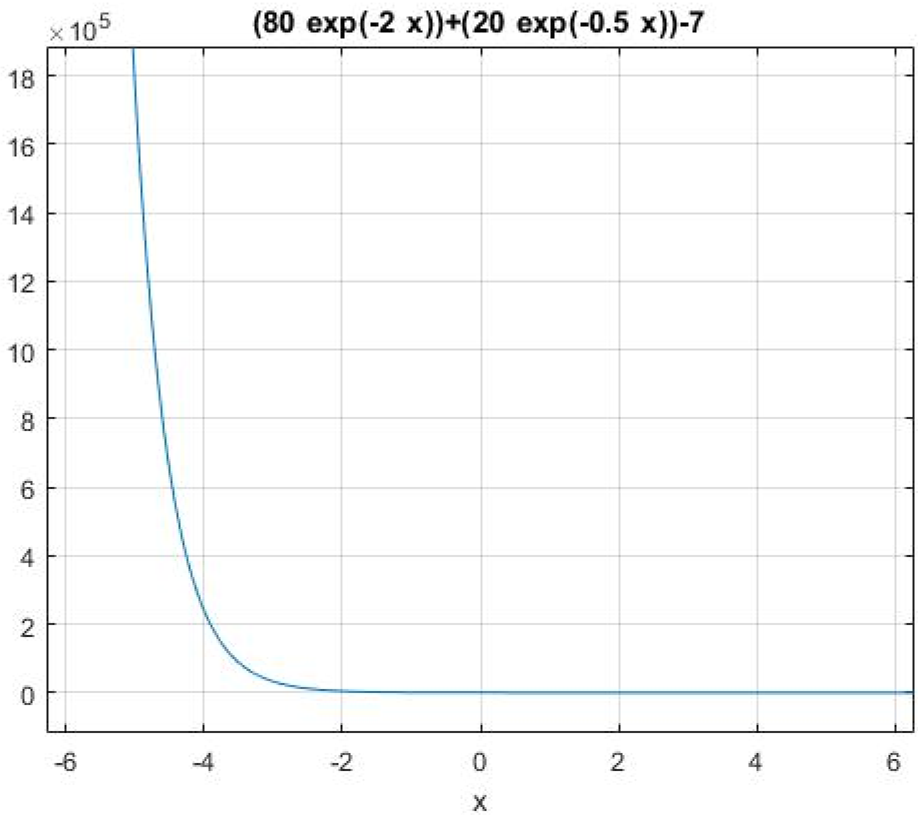
| it | xi | error |
|---|---|---|
| 1 | 0.54706 | 0.54706 |
| 2 | 1.1191 | 0.57209 |
| 3 | 1.6882 | 0.56901 |
| 4 | 2.1318 | 0.44362 |
| 5 | 2.3098 | 0.17807 |
| 6 | 2.3289 | 0.01906 |
| 7 | 2.3291 | 0.0001835 |
| 8 | 2.3291 | 1.6685E-8 |

Tiempo necesario: 2.3291 horas

Error obtenido: 1.6685E-8=0.000000016685

Iteraciones calculadas: 8

GRÁFICA




## EJERCICIO 5

Por medio del método de Punto Fijo determina la raíz real máxima de la siguiente función:

$$f(x) = x^3 - 6x^2 + 11x - 6.1$$

El valor inicial $x0 = 3$. Con un error menor a $10^{-3}$

## Código

```
clc,clear,close all
% Punto fijo
% Problema 5
syms x;
f=inline('x^3-6*x^2+11*x-6.1');
ezplot(f)
grid on
x0=3; % Valor inicial
% Despeje 1
g=inline('6.1/(x^2-6*x+11)')
k=subs(diff(g(x),1),x0);
k=vpa(k,4) % Apto
% It 1
x1=g(x0)
er=abs(x1-x0)
% It2
x2=g(x1)
er=abs(x2-x1)
% It 3
x3=g(x2)
er=abs(x3-x2)
```

## Resultados

**Tabla . "Concentración de datos obtenidos para el ejercicio 5"**

| it | xi | error |
|---|---|---|
| 1 | 3.0500 | 0.0500 |
| 2 | 3.0462 | 0.0038 |

| | | |
|---|---|---|
| 3 | 3.0467 | 0.00055728 |

El valor de la raíz obtenida es de 3.0467 y su error es de 0.00055728 menor que el $10^{-3}$ indicado.

## Gráfica

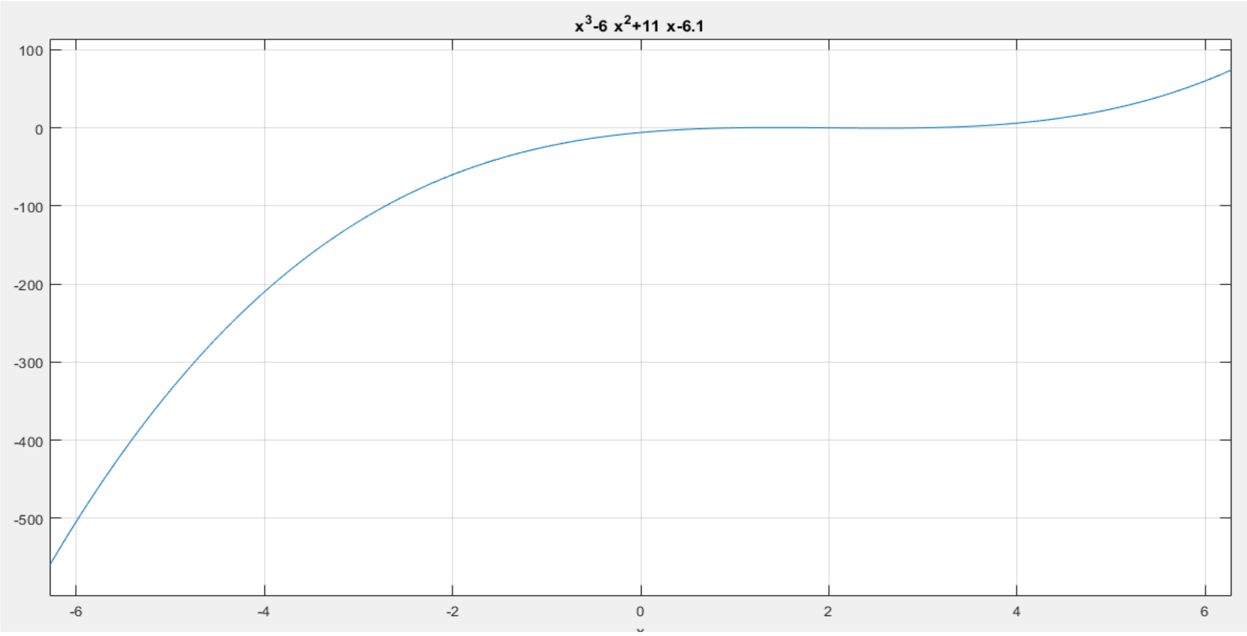



Figura ."Gráfica del comportamiento de la función dada para el ejercicio 5"

**EJERCICIO 6**

Para resolver la ecuación **x + log(x) = 0** usando el método de punto fijo, se proponen las siguientes fórmulas:

$$\text{(a)}\ x_{n+1} = -\log(x_n)$$
$$\text{(b)}\ x_{n+1} = e^{-x_n}$$
$$\text{(c)}\ x_{n+1} = \frac{x_n + e^{-x_n}}{2}$$

Si se sabe que la raíz está cerca de **x0 = 0.5**, ¿cuáles de las formulas anteriores se pueden usar?, ¿cuál es mejor?,

Experimente con las 3 formulas y discuta los resultados. **Error 0.001**

## Código

### Formula a)

```
clc, clear, close all

%Punto fijo

%ejercicio 6

f=inline('x+log10(x)');

ezplot(f)

grid on

x0=0.5; %Valor inicial

%Formula a

g=inline('-log10(x)');

syms x

k=subs(diff(g(x),1),x0);

k=vpa(k,4) k =-0.8686
```

### Formula b)

```
clc, clear, close all

%Punto fijo

%ejercicio 6

f=inline('x+log10(x)');

ezplot(f)
```

```
grid on
x0=0.5; %Valor inicial
%Formula b
g=inline('exp(x)^(-x)');
syms x
k=subs(diff(g(x),1),x0);
k=vpa(k,4)  k =-0.7788
```

Formula c)

```
clc, clear, close all
%Punto fijo
%ejercicio 6
f=inline('x+log10(x)');
ezplot(f)
grid on
x0=0.5; %Valor inicial
%Formula c
g=inline('(x+exp(x)^(-x))/2');
syms x
k=subs(diff(g(x),1),x0);
k=vpa(k,4) k =0.1106
%Itearción 1
x1=g(x0)
er=abs(x1-x0);
er=vpa(er,4)
%Itearción 2
x2=g(x1)
er=abs(x2-x1);
er=vpa(er,4)
%Iteracion 3
x3=g(x2)
er=abs(x3-x2);
```

```
er=vpa(er,4)
```

## Resultados

**Tabla . "Concentración de datos obtenidos para el ejercicio 6"**

| it | xi | error |
|---|---|---|
| 1 | 0.6394 | 0.1394 |
| 2 | 0.6519 | 0.01251 |
| 3 | 0.6528 | 0.0009316 |

La formula que mas se acerca a la raiz requerida x0=0.5 es la formula **c** ya que esta arrojó un resultado positivo y las anteriores uno negativo dando así que la **raíz cercana** fue de 0.6528 con un **error** de 0.0009316

## Gráfica

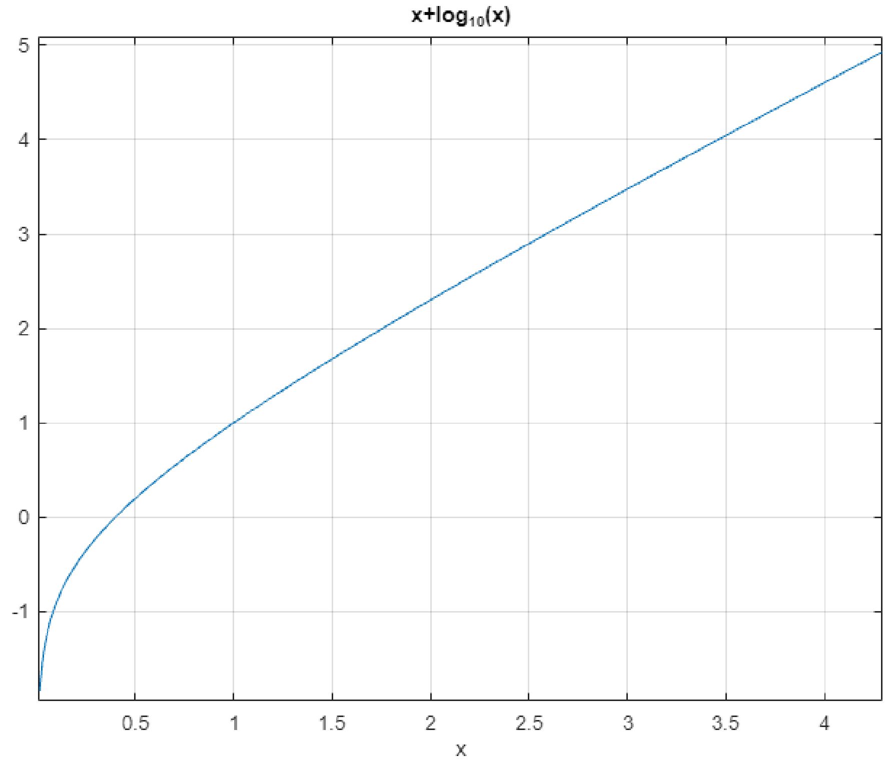



Figura ."Gráfica del comportamiento de la función dada para el ejercicio 6"

## CONCLUSIONES

- Se aplicaron los diferentes métodos (Newton-Rhapson, Punto Fijo) para obtener aproximaciones para las cifras esperadas, así como sus errores
- El método de punto fijo busca hallar las raíces en funciones de la forma , a través de aproximaciones sucesivas que convergen a la solución de la ecuación.
- En el método de Newton Raphson, cuando se tiene un máximo o mínimo local la tendencia del método será oscilar alrededor del máximo o mínimo local, de enfocándose de la raíz y perdiendo el efecto de búsqueda de raíces del método.
- Se obtuvieron las gráficas de los diferentes métodos para determinar su forma de convergencia
- Se puede afirmar en el método de punto fijo este tiene definidas las etapas de su procedimiento, lo cual evita confusiones en sus pasos y su complejidad radica principalmente en que no ocurra la convergencia con el despeje realizado y tener que determinar otros posibles despejes que permitan continuar con su ejecución.

## BIBLIOGRAFIA


- Tjalling J. Ypma, Historical development of the Newton-Raphson method, SIAM Review 37 (4), 531–551, 1995.
- P. Deuflhard, Newton Methods for Nonlinear Problems. Affine Invariance and Adaptive Algorithms.Springer Series in Computational Mathematics, Vol. 35. Springer, Berlín, 2004
- Lizarralde F. (2003). METODO DE NEWTON-RAPHSON. 18/03/2021, de Facultad de Ingeniería UNMDP Sitio web: http://www3.fi.mdp.edu.ar/analisis/temas/no_lineales_1/newtonRaphson.htm#:~:text=El%20m%C3%A9todo%20de%20Newton%2DRaphson,alrededor%20de%20la%20estimaci%C3%B3n%20inicial.