Pontificia Universidad Javeriana

Taller N. 1 Análisis numérico

Juan Felipe Vanegas

Diego Mateus Cruz

1. Evaluar el valor de un polinomio es una tarea que involucra para la maquina realizar un número de operaciones la cual debe ser mínimas. Para cada uno de los siguientes polinomios,
   a. Hallar P(x) en el valor indicado y el número mínimo de operaciones para realizarlo
   b. Demuestre que el número mínimo de multiplicaciones es n siendo n el grado del polinomio

$P(x) = 2x^4 - 3x^2 + 3x - 4 \qquad \text{en } x_0 = -2$

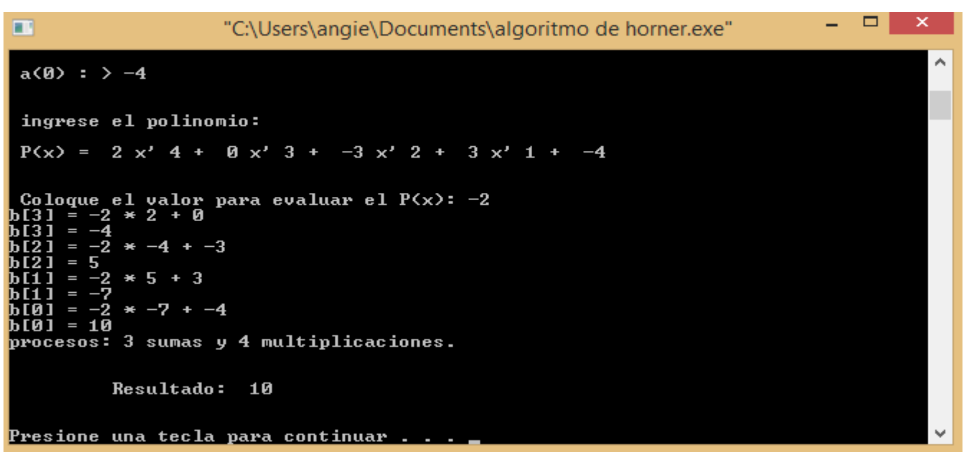
$P(x) = 7x^5 + 6x^4 - 6x^3 + 3x - 4 \qquad \text{en } x_0 = 3$

$P(x) = -5x^6 + 3x^4 + 2x^2 - 4x \qquad \text{en } x_0 = -1$

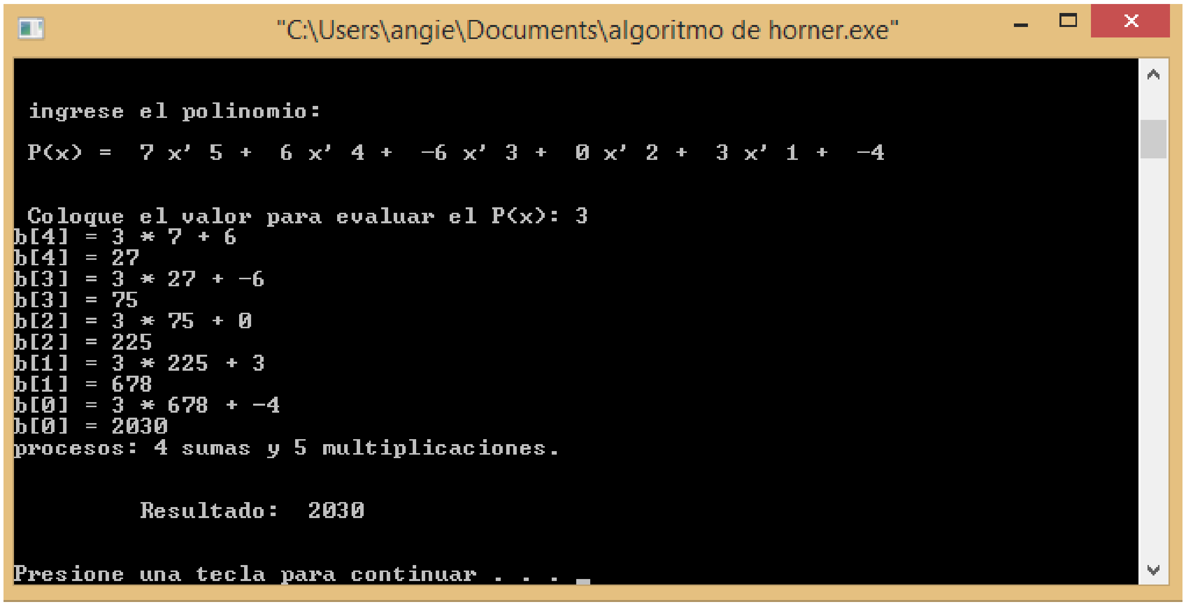
Solución:

a. Para encontrar el mínimo de operaciones para realizar cada polinomio (p(x)) se hizo uso del algoritmo de Horner, para ello se implementó un código en C++ el cuál hace de manera iterativa (ver archivo "Códigos.pdf")

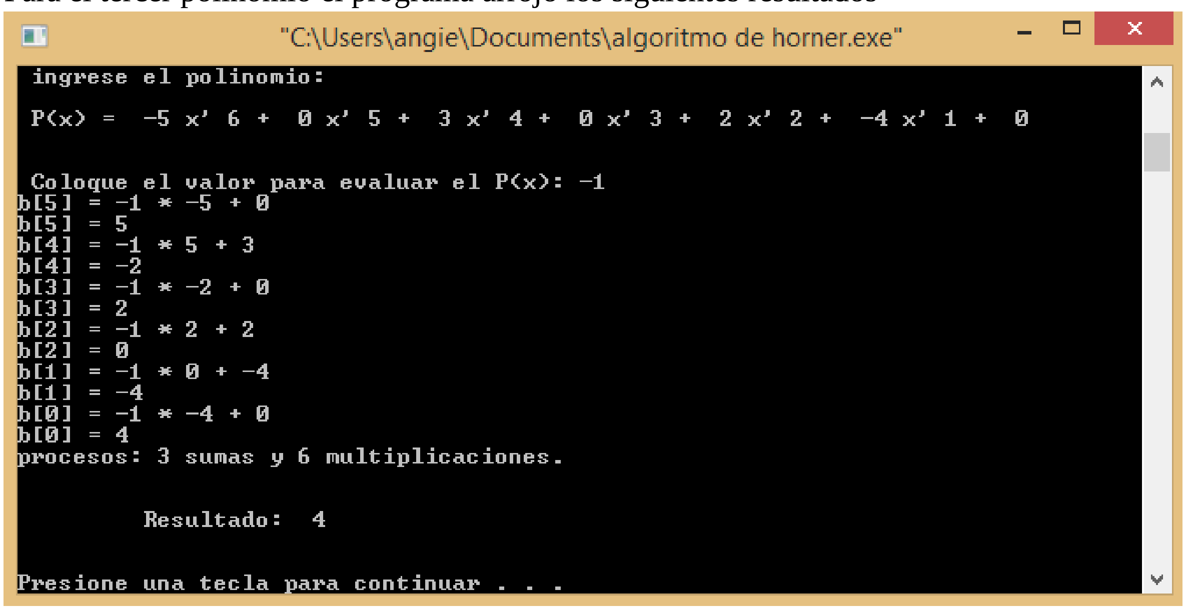
- Para el primer polinomio el programa arrojó los siguientes resultados



- Para el segundo polinomio el programa arrojó los siguientes resultados



- Para el tercer polinomio el programa arrojó los siguientes resultados



b. Demostración

2. La eficiencia de un algoritmo esta denotada por **T(n)**

6. Dado el siguiente algoritmo

```
Leer n
Mientras n>0 repita
    d ← mod(n,2)        Produce el residuo entero de la división n/2
    n ← fix(n/2)        Asigna el cociente entero de la división n/2
    Mostrar d
fin
```

a) Recorra el algoritmo con **n = 73**

b) Suponga que **T(n)** representa la cantidad de operaciones aritméticas de división que se realizan para resolver el problema de tamaño **n**. Encuentre **T(n)** y exprésela con la notación **O( )** Para obtener **T(n)** observe el hecho de que en cada ciclo el valor de **n** se reduce aproximadamente a la mitad.

Solución:

a. Recorrido con n = 73 arroja los siguientes datos

Valor entrada	**Valor salida**
73	1
36	0
18	0
9	1
4	0
2	0
1	1

El programa recibe un n como parámetro y de ahí hace un proceso iterativo para convertir el número ingresado en binario.

b.

$$T(n) = \begin{cases} O(1) & \text{si } n \leq 0 \\ O(\log n) & \text{si } n > 0 \end{cases}$$

3. Utilice R y el método de Newton para resolver el problema, muestre gráficamente cómo se comporta la convergencia a la solución

Ejemplo. Una partícula se mueve en el espacio con el vector de posición **R(t) = (2cos(t), sen(t), 0)**. Se requiere conocer el tiempo en el que el objeto se encuentra más cerca del punto **P(2, 1, 0).** Utilice el método de Newton con cuatro decimales de precisión.

Solución:

Para 2cos(x) con x0 = 2

```
-------------------------------------------------------------------------
--
x_k                   f(x_k)              Error est.
      n-------------------------------------------------------------------
---------
 1.545658487756334  0.050270383285824  0.454341512243666
 1.568298255699715  0.004996136994078  0.022639767943381
 1.570548315770197  0.000496022044314  0.002250060070482
 1.570771704239591  0.000049245110606  0.000223388469394
 1.570793882265769  0.000004889058256  0.000022178026177
-------------------------------------------------------------------------
--
k =  5  x =  1.57079388226577  f(x) =  4.88905825613663e-06  Error estima
do <=  -2.21780261774142e-05
```

Para sen(x) con x0 = 1

```
-------------------------------------------------------------------------
--
x_k                   f(x_k)              Error est.
      n-------------------------------------------------------------------
---------
-0.515859365449571 -0.493282566569917  1.515859365449571
 0.057443133387240  0.057411547617734  0.573302498836811
-0.000014338714536 -0.000014338714536  0.057457472101776
-0.000000000004607 -0.000000000004607  0.000014338709929
-------------------------------------------------------------------------
--
k =  4  x =  -4.60701481575388e-12  f(x) =  -4.60701481575388e-12  Error
estimado <=  -1.43387099290709e-05
```

4. Solución en R. Para la siguiente ecuación, utilice dos métodos diferentes, grafique las soluciones y comparar

Encuentre una intersección de las siguientes ecuaciones en coordenadas polares

r = 2+cos(3*t), r = 2 – e^t

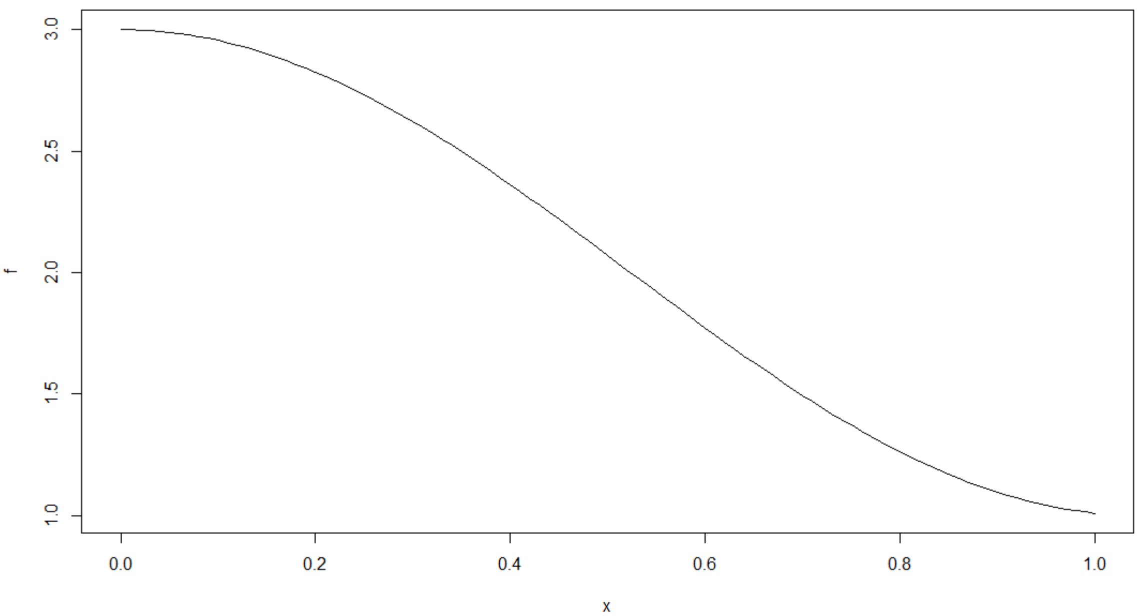
Solución:

Se hizo uso de tres códigos en R para la ejecución de los métodos iterativos de solución de raíces de **Newton y punto fijo,** además de utilizar un código para la gráfica en coordenadas polares de la intersección de las dos funciones**.**

Con método de newton

r = 2 + cos(3t)

```
--------------------------------------------------------------------------
--
x_k                     f(x_k)              Error est.
      n-------------------------------------------------------------------
---------
 1.166126520700921  1.062976120004705  0.666126520700921
 0.295723091981480  2.631609177587467  0.870403428719441
 1.373151647578543  1.441203364085181  1.077428555597063
 0.832268069790283  1.200773063934963  0.540883577788260
 1.468479966648404  1.697848271924218  0.636211896858121
 0.975161976663271  1.023260320697251  0.493317989985132
 2.818503896261333  1.434095379303633  1.843341919598062
 3.464363037845892  1.433308169723315  0.645859141584559
 2.967821027147968  1.132834907744077  0.496542010697924
 4.243279744245097  2.986668731610139  1.275458717097129
 10.968659837624365  2.080655644190435  6.725380093379269
--------------------------------------------------------------------------
--
Se alcanzó el máximo número de iteraciones.
k =  11 Estado: x =  10.9686598376244 Error estimado <=  -6.7253800933792
7
```

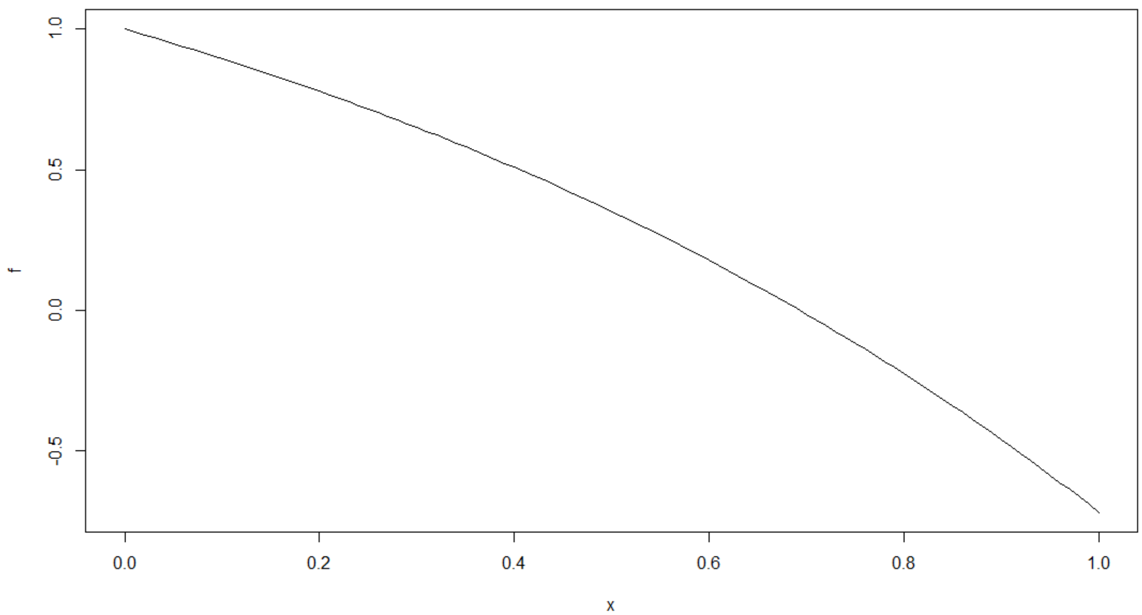


r = 2 - e^t

```
--------------------------------------------------------------------------
--
x_k                     f(x_k)              Error est.
      n-------------------------------------------------------------------
---------
 0.710935833917078 -0.035895627601176  0.210935833917078
 0.692973663351646  0.000347004310119  0.017962170565432
 0.693147304294062 -0.000000247468249  0.000173640942416
 0.693147180460961  0.000000000197970  0.000000123833102
```

```
 0.693147180560025 -0.000000000000159  0.000000000099064
```

```
--
k =  5  x =  0.693147180560025  f(x) =  -1.58539847916472e-13  Error esti
mado <=  -9.9064e-11
```



Con método de punto fijo

r = 2 + cos(3t)

```
x_ 1 =  2.0707372016677
x_ 2 =  2.99748242387438
x_ 3 =  1.0920083145008
x_ 4 =  1.00902242020347
x_ 5 =  1.00655086798591
x_ 6 =  1.00742547998645
x_ 7 =  1.00710973867758
x_ 8 =  1.00722293528734
x_ 9 =  1.00718225066828
x_ 10 =  1.00719686017209
x_ 11 =  1.00719161232048
x_ 12 =  1.00719349717162
x_ 13 =  1.0071928201684
x_ 14 =  1.00719306333162
x_ 15 =  1.00719297599278
x_ 16 =  1.00719300736289
x_ 17 =  1.00719299609546
x_ 18 =  1.00719300014247
x_ 19 =  1.00719299868887
x_ 20 =  1.00719299921097
x* es aproximadamente  1.00719299921097  con error menor que  1e-09
```

r = 2 – e^t

```
x_ 1 =  0.351278729299872
x_ 2 =  0.579116687466003
x_ 3 =  0.215538502816883
x_ 4 =  0.75947025410675
x_ 5 =  -0.137143777430392
x_ 6 =  1.12815513449415
x_ 7 =  -1.08995069506946
x_ 8 =  1.66376692875373
x_ 9 =  -3.27915965474795
x_ 10 =  1.96234010917599
x_ 11 =  -5.11595971908799
x_ 12 =  1.9939997834521
x_ 13 =  -5.34485290922668
x_ 14 =  1.99522734683304
x_ 15 =  -5.3538747179846
x_ 16 =  1.99527021114961
x_ 17 =  -5.35418994355443
x_ 18 =  1.99527170186503
x_ 19 =  -5.35420090656694
x_ 20 =  1.99527175370114
x_ 21 =  -5.3542012877801
x_ 22 =  1.99527175550361
x_ 23 =  -5.35420130103582
x_ 24 =  1.99527175556628
x_ 25 =  -5.35420130149675
x_ 26 =  1.99527175556846
x_ 27 =  -5.35420130151278
x_ 28 =  1.99527175556854
x_ 29 =  -5.35420130151334
x_ 30 =  1.99527175556854
x_ 31 =  -5.35420130151336
x_ 32 =  1.99527175556854
x_ 33 =  -5.35420130151336
x_ 34 =  1.99527175556854
x_ 35 =  -5.35420130151336
x_ 36 =  1.99527175556854
x_ 37 =  -5.35420130151336
x_ 38 =  1.99527175556854
x_ 39 =  -5.35420130151336
x_ 40 =  1.99527175556854
x_ 41 =  -5.35420130151336
x_ 42 =  1.99527175556854
x_ 43 =  -5.35420130151336
x_ 44 =  1.99527175556854
x_ 45 =  -5.35420130151336
x_ 46 =  1.99527175556854
x_ 47 =  -5.35420130151336
x_ 48 =  1.99527175556854
x_ 49 =  -5.35420130151336
x_ 50 =  1.99527175556854
x_ 51 =  -5.35420130151336
x_ 52 =  1.99527175556854
x_ 53 =  -5.35420130151336
x_ 54 =  1.99527175556854
x_ 55 =  -5.35420130151336
x_ 56 =  1.99527175556854
x_ 57 =  -5.35420130151336
```

```
x_ 58 =  1.99527175556854
x_ 59 =  -5.35420130151336
x_ 60 =  1.99527175556854
x_ 61 =  -5.35420130151336
x_ 62 =  1.99527175556854
x_ 63 =  -5.35420130151336
x_ 64 =  1.99527175556854
x_ 65 =  -5.35420130151336
x_ 66 =  1.99527175556854
x_ 67 =  -5.35420130151336
x_ 68 =  1.99527175556854
x_ 69 =  -5.35420130151336
x_ 70 =  1.99527175556854
x_ 71 =  -5.35420130151336
x_ 72 =  1.99527175556854
x_ 73 =  -5.35420130151336
x_ 74 =  1.99527175556854
x_ 75 =  -5.35420130151336
x_ 76 =  1.99527175556854
x_ 77 =  -5.35420130151336
x_ 78 =  1.99527175556854
x_ 79 =  -5.35420130151336
x_ 80 =  1.99527175556854
x_ 81 =  -5.35420130151336
x_ 82 =  1.99527175556854
x_ 83 =  -5.35420130151336
x_ 84 =  1.99527175556854
x_ 85 =  -5.35420130151336
x_ 86 =  1.99527175556854
x_ 87 =  -5.35420130151336
x_ 88 =  1.99527175556854
x_ 89 =  -5.35420130151336
x_ 90 =  1.99527175556854
x_ 91 =  -5.35420130151336
x_ 92 =  1.99527175556854
x_ 93 =  -5.35420130151336
x_ 94 =  1.99527175556854
x_ 95 =  -5.35420130151336
x_ 96 =  1.99527175556854
x_ 97 =  -5.35420130151336
x_ 98 =  1.99527175556854
x_ 99 =  -5.35420130151336
x_ 100 =  1.99527175556854
No hubo convergencia
```

Gráfico de intersectas de las dos funciones

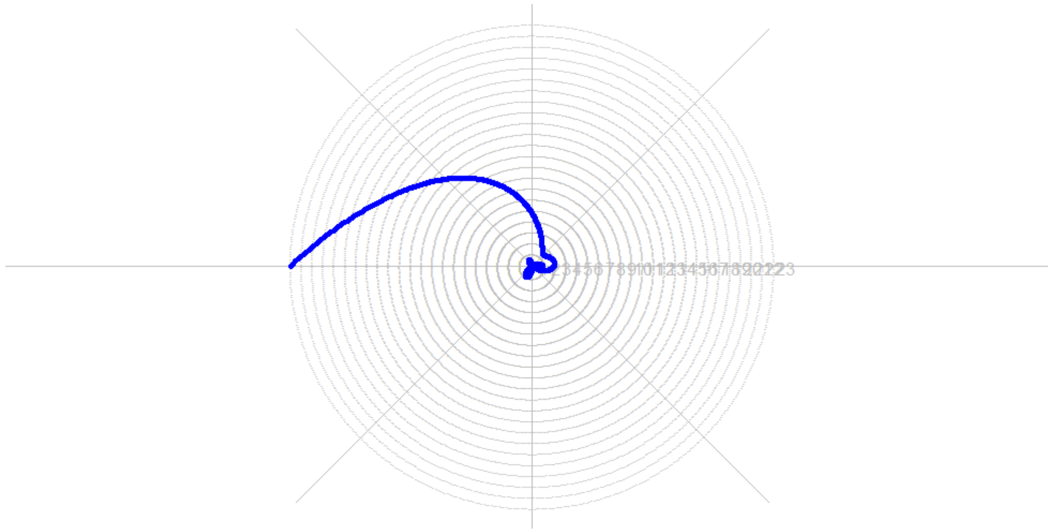
La ecuación se obtuvo igualando ambas funciones de r de la siguiente manera

2+ cos(3t) = 2 - e^t

2 – 2 + cos(3t) + e^t = 0

r =Cos(3t) + e^t = 0

Y se obtuvo la siguiente gráfica:



5. El número épsilon de una maquina el cual se denota como e, es la distancia entre el 1 y el menor número de punto flotante mayor que 1. Para el punto flotante de precisión doble se tiene que: $$ \varepsilon_{maq} = 2^{-52} $$

Teniendo en cuenta lo anterior, encuentre el error de redondeo para x = 0.4

Solución:

Se implementó un código en R el cuál calcula el valor de epsilón con el error de redondeo en x = 0.4 (ver archivo "Codigos.pdf")

Esto da como resultado:

```
Epsilon de la maquina en forma hexadecimal =  2.220446e-16
Epsilon de la maquina en forma binaria = 2^- 52
El error acumulado es  =  401
```

6. Resolver los ejercicios 13,14

13. Encuentre una fórmula iterativa de convergencia cuadrática y defina un intervalo de convergencia apropiado para calcular la raíz real n-ésima de un número real. El algoritmo solamente debe incluir operaciones aritméticas elementales.

14. El siguiente es un procedimiento intuitivo para calcular una raíz real positiva de la ecuación **f(x) = 0** en un intervalo **[a, b]** con precisión **E**:

A partir de **x = a** evalúe **f(x)** incrementando **x** en un valor **d**. Inicialmente **d = (b – a)/10** Cuando **f** cambie de signo, retroceda **x** al punto anterior **x – d**, reduzca **d** al valor **d/10** y evalúe nuevamente **f** hasta que cambie de signo Repita este procedimiento hasta que **d** sea menor que **E**.

a) De manera formal escriba las condiciones necesarias para que la raíz exista, sea única y pueda ser calculada.
b) Indique el orden de convergencia y estime el factor de convergencia del método.
c) Describa el procedimiento anterior en notación algorítmica, o en MATLAB o en Python

Solución

14. Se implementó un código en Python que sigue paso a paso las condiciones dadas (ver archivo "Codigos.pdf")

Se hizo el procedimiento con los valores

a= 0

b=4

E =0.0001

Y dio el siguiente resultado

```
Por favor introducir valor de limite innferior(a): 0
Por favor introducir valor de limite superior(b): 4
Por favpr ingrese la precision (E): 0.0001
X1:  4.3999999999999995 b:  4.0
La funcion ha superado el limite superior establecido en el rango
La raiz es : 4.3999999999999995
```