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

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

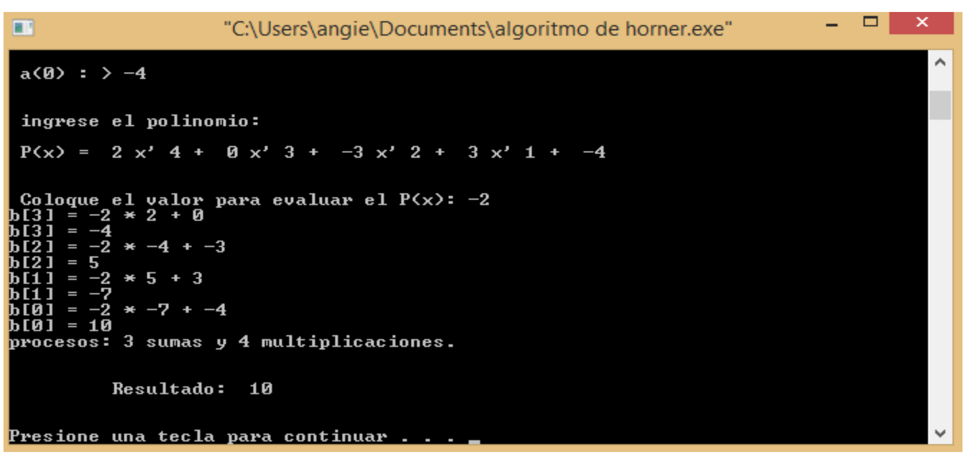
$$P(x) = 7x^5 + 6x^4 - 6x^3 + 3x - 4 \qquad \text{en } x_0 = 3$$

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

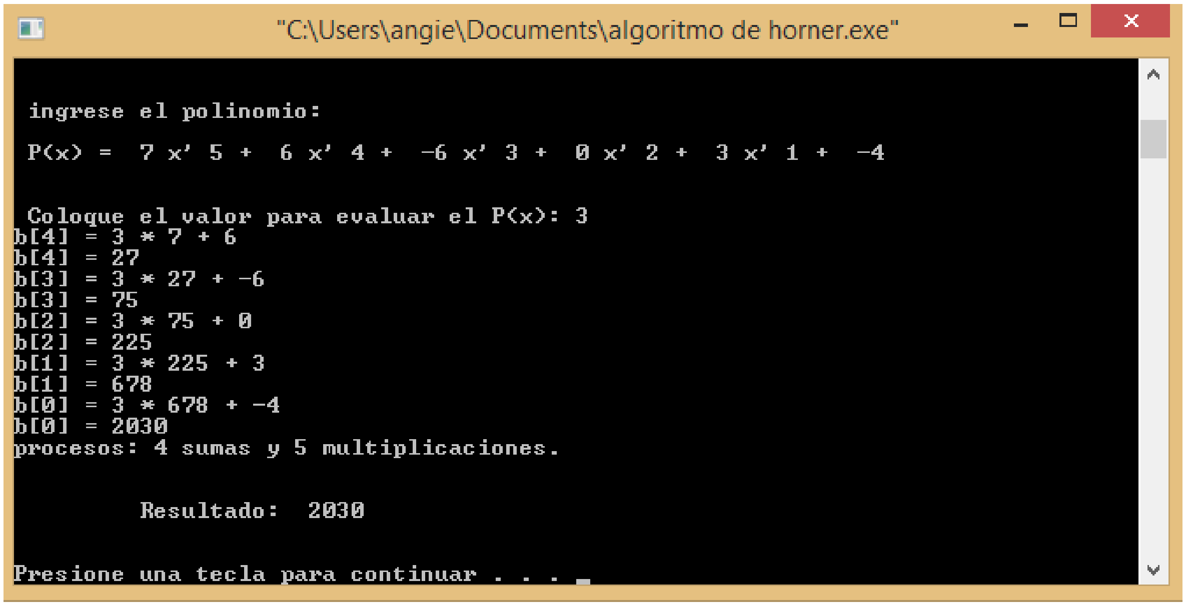
Solución:

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")

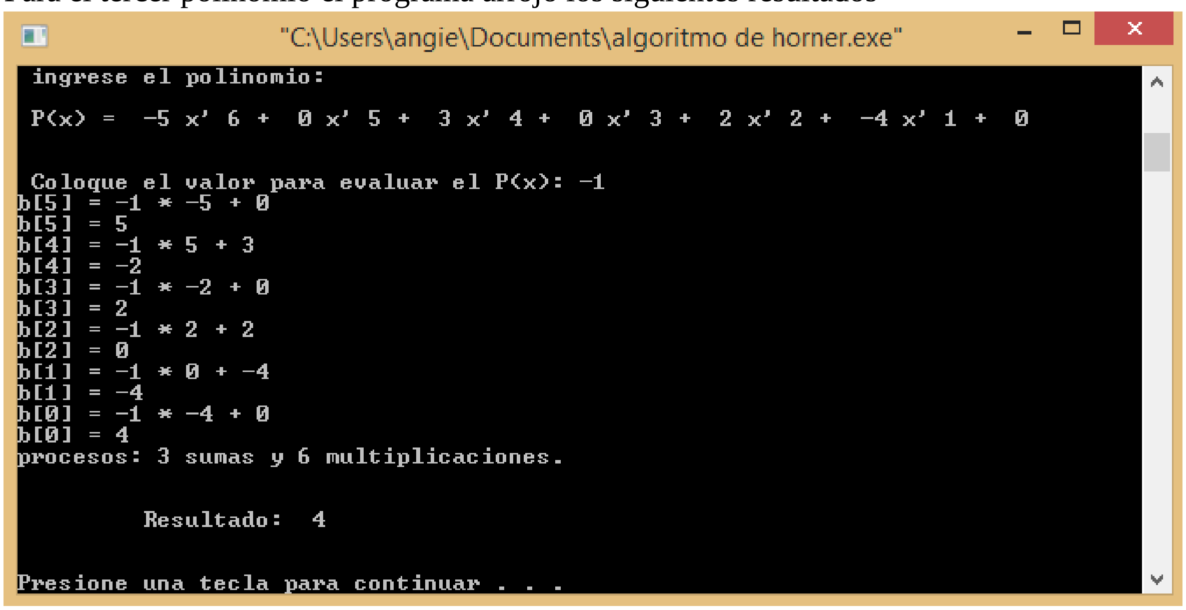
- Para el primer polinomio el programa arrojó los siguientes resultados



- 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**

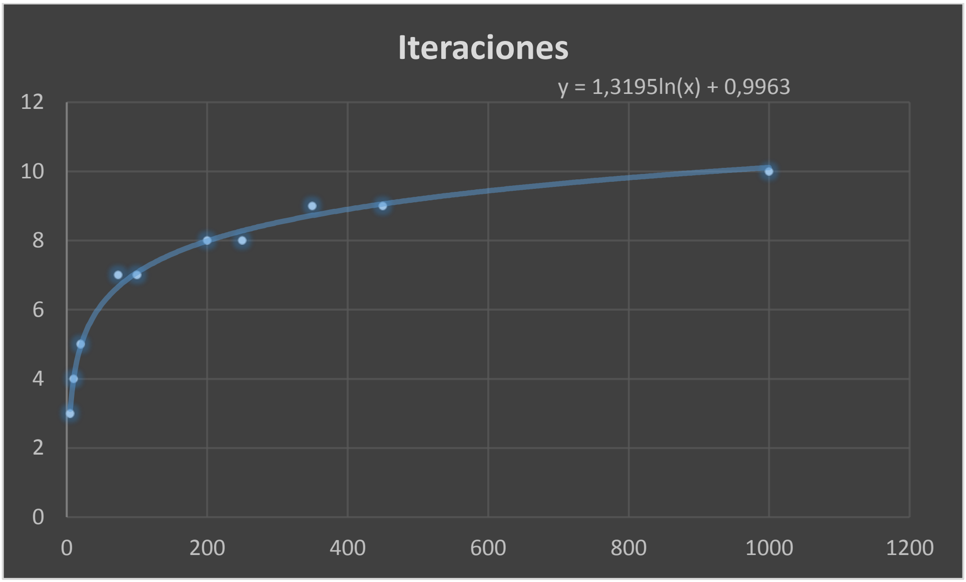
b) Suponga que **T(n)** representa la cantidad de operaciones aritméticas de división que se realizan para resolver el problema de tamaño **n**. Encuentre **T(n)** y exprésela con la notación **O( )** Para obtener **T(n)** observe el hecho de que en cada ciclo el valor de **n** se reduce aproximadamente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

Solución:

a. Recorrido con n = 73 arroja los siguientes datos

n	iteraciones
5	3
10	4
20	5
73	7
100	7
200	8
250	8
350	9
450	9
1000	10

El programa recibe un n como parámetro, imprime unos y ceros. Sin embargo, cuando se analiza el número de operaciones que realiza al recibir un n como parámetro (siendo n cualquier entero) y da la siguiente gráfica.



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 solucionar este punto, se utilizó formula de distancia entre un punto y una función parametrizada, de ahí, dicha función fue evaluada con método de newton

Se tiene:

F(x) = √((2cos(t)-2) ^**2** + (sen(t)-1) ^**2**)

= √((3cost) ^**2** + 6 – 8cos(t) – 2sin(t))

Con x0= 0

```
x_k                    f(x_k)              Error est.
 0.948126237341157  0.364920022310591  0.948126237341157
 0.748919731098974  0.016494936147609  0.199206506242183
 0.739014852660591 -0.000117620556204  0.009904878438383
 0.739085481446337  0.000000582803930  0.000070628785746
-------------------------------------------------------------------------
--

Error estimado <=  -7.06287857458e-05
Tiempo estimado. 0.739085481446337
```

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

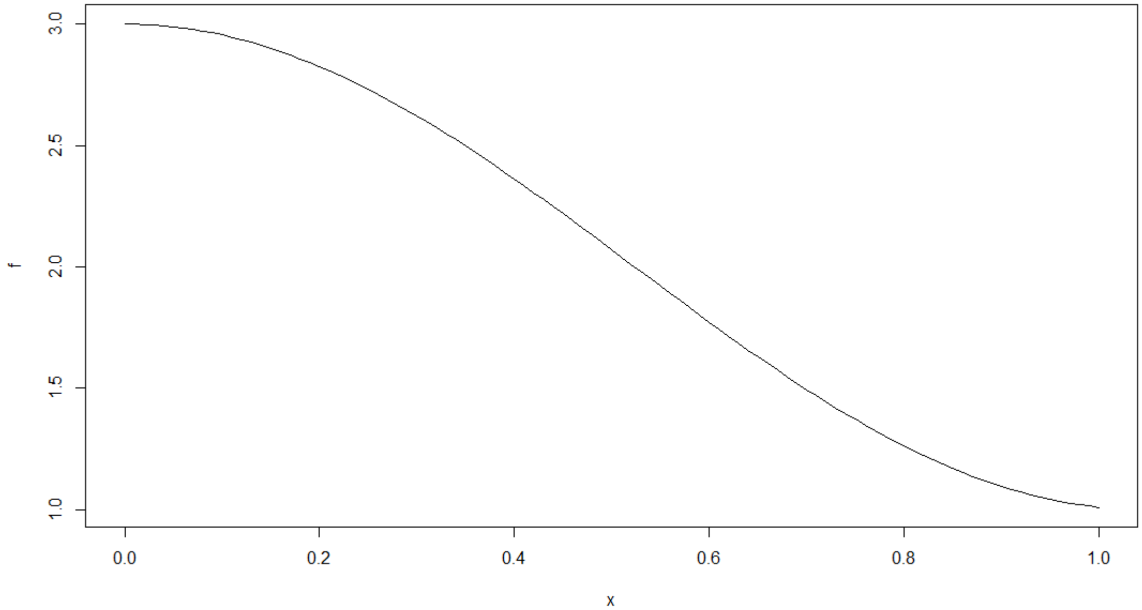
Solución:

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

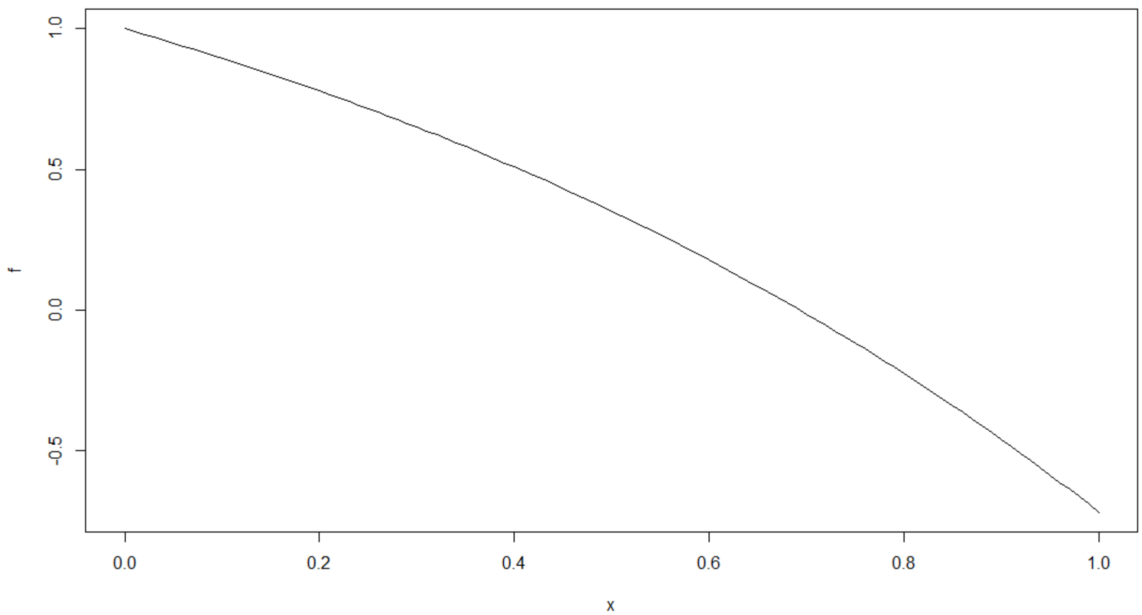
r = 2 + cos(3t)

```
-------------------------------------------------------------------------
--
x_k                    f(x_k)              Error est.
      n-------------------------------------------------------------------
---------
 2.819467766380655  1.431713700444720  1.819467766380656
 3.498190428256874  1.519694484936670  0.678722661876218
 2.846371807678037  1.367223385350123  0.651818620578837
 3.462114480557541  1.427763144931307  0.615742672879504
 2.785264631348492  1.518984841715655  0.676849849209049
 3.469354588061698  1.445709306365542  0.684089956713205
 2.818264998718306  1.434686416675519  0.651089589343391
 3.464390319871645  1.433375607046068  0.646125321153338
 2.818855334894181  1.433226441406389  0.645534984977463
 3.464323141639717  1.433209558759126  0.645467806745536
 2.967815292811145  1.132843475329179  0.496507848828572
-------------------------------------------------------------------------
--
Se alcanzó el máximo número de iteraciones.
k =  11 Estado: x =  2.96781529281115 Error estimado <=  0.49650784882857
2
```



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

Se aplicó el siguiente procedimiento
F(t) = cos(3t) + e ^t = 0
e ^t = -cos(3t)
t =ln(-cos(3t))
g(x) = ln(-cos(3t))
g’(x) = -3sen(3t)/cos(3t) y se evaluó con esta expresión, lo cuál generó los siguientes res ultados con un intervalo de -1 a 0 y p.medio = - 0.5

```
 x_ 1 =  0.427639629222833
x_ 2 =  -10.1316208660407
x_ 3 =  -4.89606416089471
x_ 4 =  -4.88188171571945
x_ 5 =  -5.38249366073182
x_ 6 =  1.4105744524602
x_ 7 =  -5.75310913948251
x_ 8 =  154.363687076245
x_ 9 =  -9.91354978294327
x_ 10 =  28.6088360054706
x_ 11 =  -4.70845458062871
x_ 12 =  254.156573577509
x_ 13 =  4.08445265442287
x_ 14 =  0.970957506668106
x_ 15 =  0.698381200151447
x_ 16 =  5.18718207378698
x_ 17 =  0.442417173396673
x_ 18 =  -12.0735485653858
x_ 19 =  -32.39970173477
x_ 20 =  -0.577790359123896
No hubo convergencia
```

La función ingresada no tiene convergencia, ya que no existe ningun punto medio que satisfaga la igualdad -1 <= g'(x) <= 1

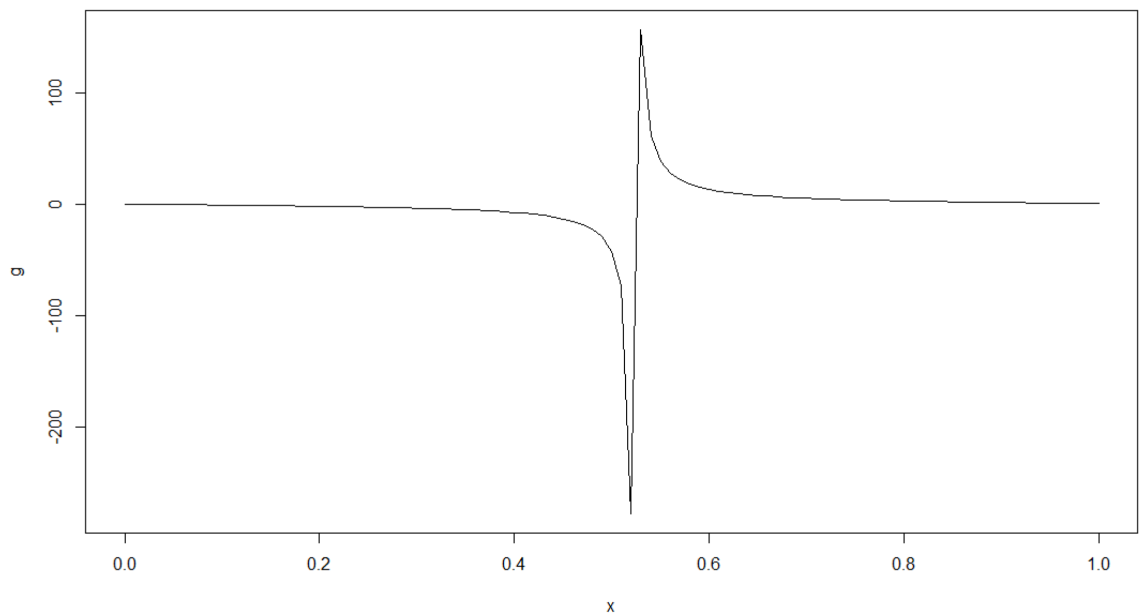


Gráfico de intersectas de las dos funciones

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

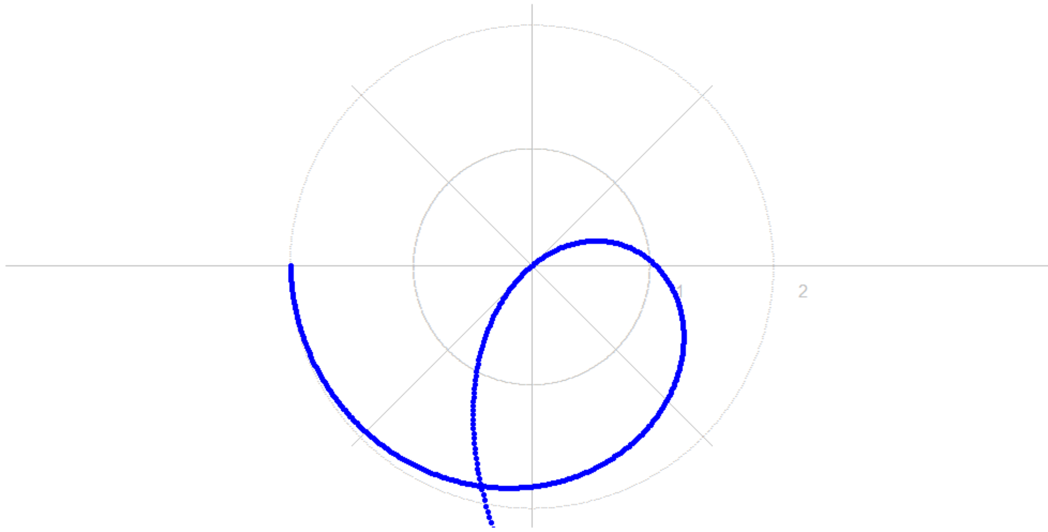
2+ cos(3t) = 2 - e^t

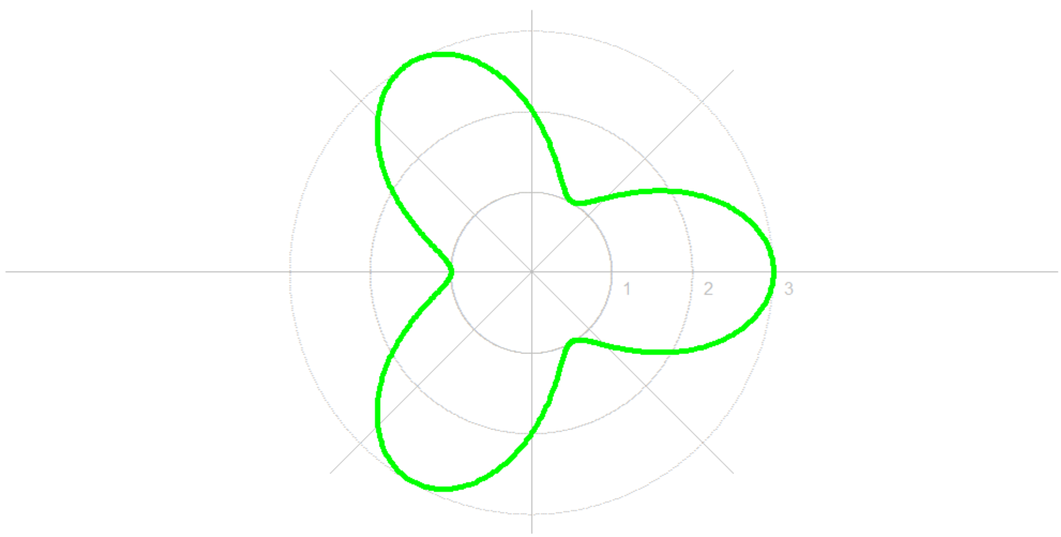
2 – 2 + cos(3t) + e^t = 0

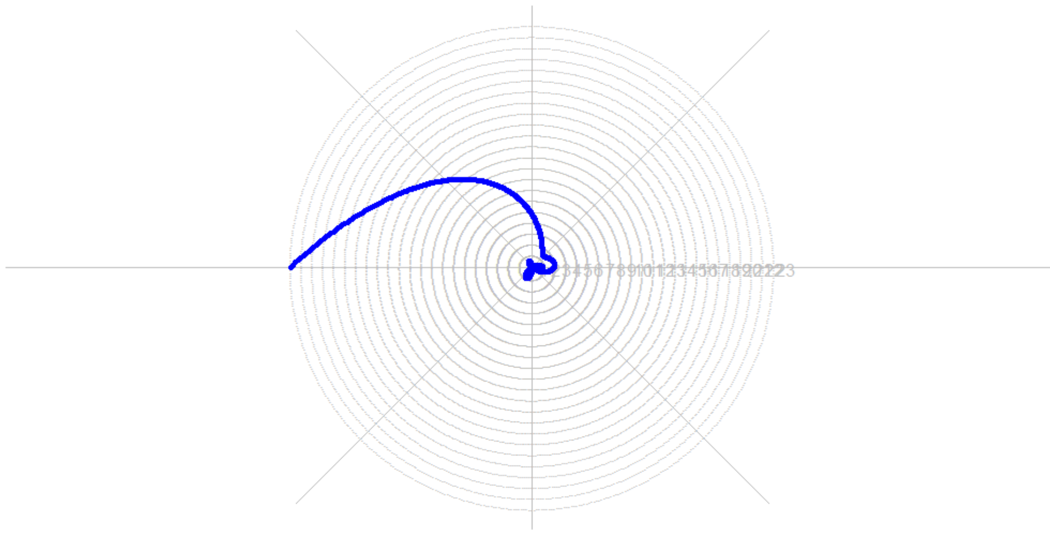
r =Cos(3t) + e^t = 0

Y se obtuvieron las siguientes gráficas:

Para 2- e^t


2


1
2
3



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.

a. Las condiciones para que la raíz exista, sea única y pueda ser calculada son

- La función debe ser continua en un intervalo de a,b
- X debe existir en dicho intervalo
- Debe haber un cambio de signo cuando se evalúa el valor a y b en la función
- La pendiente debe ser menor a 1 para que exista raíz única

c. 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
```