Nombre

David Sarmiento Santamaria y Natalia Villate Obando

Fecha: 11 de febrero del 2019

Análisis Numérico. Taller 1

Punto 1. Haller el valor de P(x) (Polinomio) y el numero mínimo de operaciones.

Para la realización de este punto se tomo como base del desarrollo el método de Horner, el cual ofrece un algoritmo para evaluar de forma más eficiente funciones polinómicas de manera monomial, a partir de un numero real dado que en este caso será llamado x_0.

A partir de un polinomio P(x)

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots a_n x^n$$

En donde se quiere evaluar un valor especifico para x, definido como x_0. Se define una nueva expresión/secuencia de constantes para después poder sustituir iterativamente el valor de b en la expresión.

$$b_n \coloneqq a_n$$

$$b_{n-1} \coloneqq a_{n-1} + b_n x_0$$

$$b_0 \coloneqq a_0 + b_1 x_0$$

a. Valor del Polinomio y número mínimo de operaciones
   1. $P(x) = 2x^4 - 3x^2 + 3x - 4 \qquad x_0 = -2$

      El valor del polinomio de grado 4 es P (-2) = 10, el valor de multiplicaciones usados para resolver el polinomio es el mismo valor que el grado es decir en este caso 4.

   2. $P(x) = 7x^5 + 6x^4 - 6x^3 + 3x - 4 \qquad x_0 = 3$

      El valor del polinomio de grado 5 es P (3) = 2030, el valor de multiplicaciones usados para resolver el polinomio es el mismo valor que el grado es decir en este caso 5.

   3. $P(x) = -5x^6 + 3x^4 + 2x^2 - 4x \qquad x_0 = -1$

El valor del polinomio de grado 6 es P (-1) = 4, el valor de multiplicaciones usados para resolver el polinomio es el mismo valor que el grado es decir en este caso 6.

b. Demostración por medio de inducción matemática

$$P(x) = a_0 + a_1 x + a_2 x^2 + \cdots a_n x^n$$

Nueva secuencia de constantes

$$b_n \coloneqq a_n$$
$$b_{n-1} \coloneqq a_{n-1} + b_n x_0$$
$$b_0 \coloneqq a_0 + b_1 x_0$$

El polinomio entonces se puede escribir de la siguiente manera

$$P(x) = a_0 + x_0(a_1 + x_0(a_2 + \cdots + x_0(a_{n-1} + b_0 x_0) \ldots))$$
$$= a_0 + x_0\big(a_1 + x_0(a_2 + \cdots + x_0(b_{n-1}) \ldots)\big)$$
$$= a_0 + x_0(b_{1)}$$
$$= b_0$$

Así resulta después de sustituir iterativamente la b_i en la expresión.

Punto 2. Eficiencia de un algoritmo.

Para el desarrollo de este ejercicio se tuvo totalmente en cuenta el algoritmo dado por el enunciado.

```
Leer n
Mientras n > 0 repita
       d ← mod (n,2)
       n ← fix (n/2)
      Mostrar d
Fin
```

a. Evaluar el algoritmo con n =73

Evaluando el algoritmo con n= 73. Se tiene un output: 1001001. Con un total de 7 iteraciones.

Donde:

n(inicial): valor inicial a descomponer.
n: cociente entero de la división n/2
d: residuo de la división n/2

#iteración	n	d
0	36	1
1	18	0
2	9	0
3	4	1
4	2	0
5	1	0
6	0	1

Tabla1. Tabla de valores para n (inicial) = 73

b) Encontrar T(n) y expresarlo en notación de O(n)

T(n) → (cantidad de operaciones aritméticas de división)

•Se tiene que, “n” se divide entre “2” dentro de cada ciclo.
Dando: T (log(n))

•Sin embargo, no solo se divide la variable “n” dentro de cada iteración, sino que, adicionalmente se calcula su modulo. Dicho de otra manera, se calculan dos operaciones aritméticas de división.
Dando: T (2log (n))

•Se pide que se exprese a manera de notación “O (n)”, en la cual se omiten los valores constantes debido a que el foco de interés es observar que tan rápido crece el algoritmo.
Dando: O (log(n))

Punto 3. Utilice Método de Newton para mostrar como se comporta la convergencia

Para la realización de este ejercicio debido a que se debe de realizar con el método de newton es necesario conocer la derivada de la ecuación inicial

$R(t) = (2\cos(t), sen(t), 0)$ par el punto p = (2,1,0).

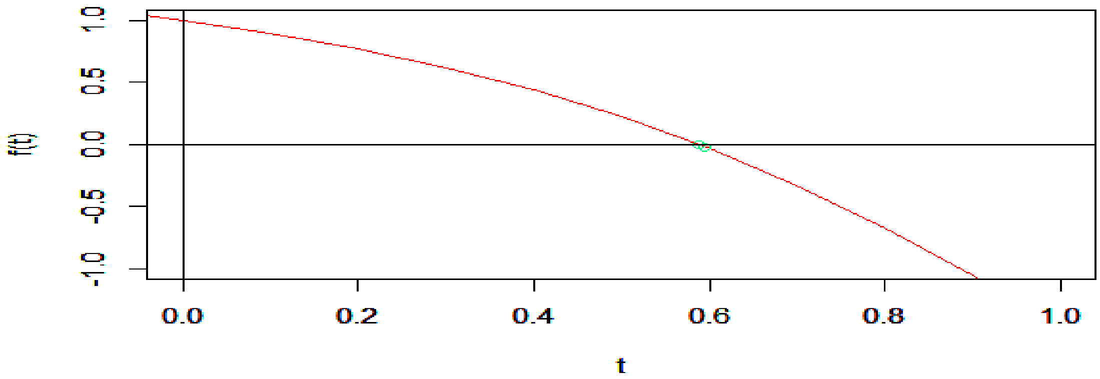
Para hallar la función a aplicar se encuentra la distancia del vector dado es decir:

$$d = \sqrt{(2\cos(t) - 2)^2 + (sen(t) - 1)^2}$$

$$f(t) = 3sen(t)\cos(t) - 4sen(t) + \cos(t)$$

Y para la aplicación del método de newton entonces es necesario encontrar la derivada

$$f'(x) = -\sin(t) - 4\cos(t) + 2\cos(2t)$$



El resultado obtenido es $t = 0.5872198$ con 4 iteraciones.

Punto 4. Utilice dos métodos diferentes, grafique y compare.

En primera instancia para poder resolver este ejercicio es fundamental conocer la nueva ecuación g(x) que surge de la intersección de las dos ecuaciones de r.

$$r = 2 + \cos(3t)$$

$$r = 2 - e^{t}$$

Al igual las ecuaciones obtenemos la siguiente función que es con la cual se procederá a realizar la bisección.

$$2 + \cos(3t) = 2 - e^{t}$$

Surge:

$$f(x) = \cos(3t) + e^{t}$$

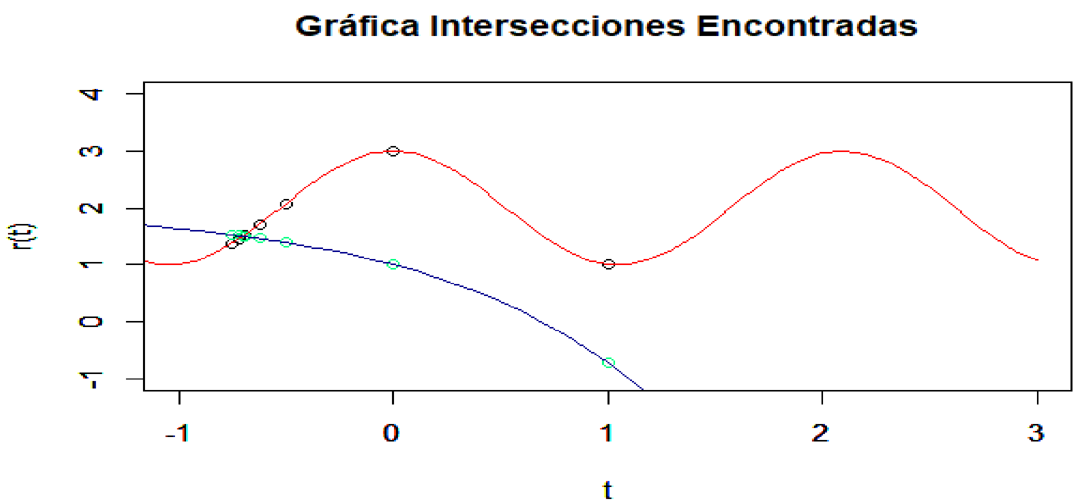
Método de Bisección

El método de Bisección es un algoritmo de búsqueda de raíces que divide el intervalo establecido en la mitad y va seleccionando el subintervalo que contiene la raíz.

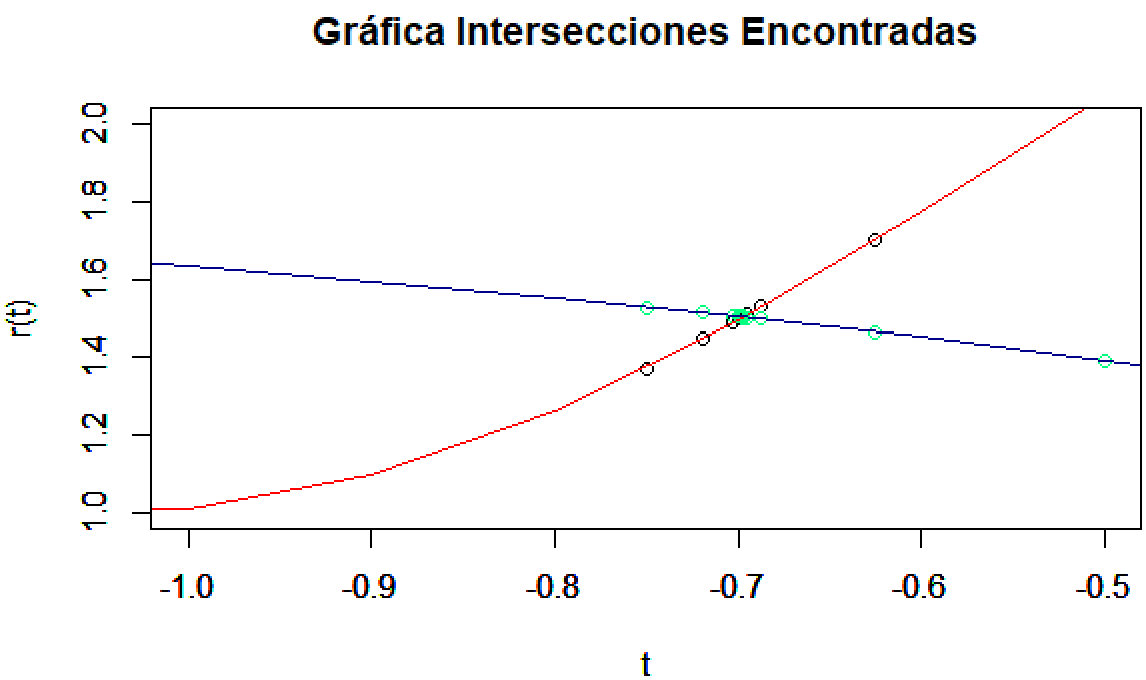
El intervalo que se decidió usar para verificar la intersección de las ecuaciones por medio de bisección fue de -1 al 3 [(-1,3)].

Las iteraciones que se realizaron fueron: 26

Con un resultado de: t = `-0.6973291`



La siguiente grafica muestra más claramente los puntos de intersección resultantes



Método de Newton

El método de Newton es un algoritmo para encontrar aproximaciones de los ceros o raíces de una función real.

Para el desarrollo del ejercicio con el uso de Newton fue necesario halla la derivada de la función encontrada a partir de las ecuaciones dadas:

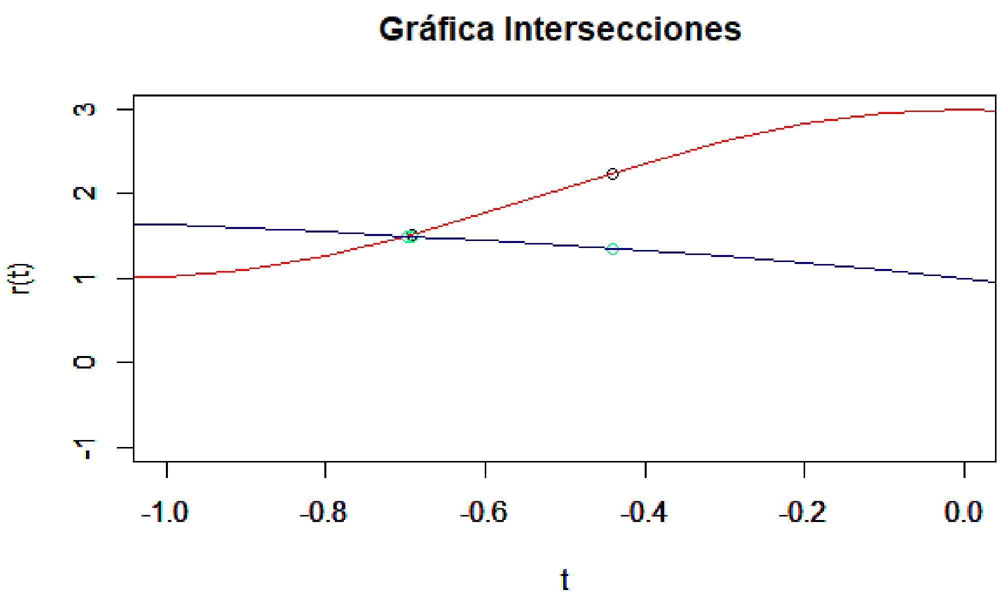
$$f(x) = \cos(3t) + e^t$$

$$f'(x) = -3\sin(3t) + e^t$$

El intervalo que se decidió usar para verificar la intersección de las ecuaciones por medio de bisección fue de -1 al 1 [(-1,1)].

Las iteraciones que se realizaron fueron: 6

Con un resultado de: *t* = -0.6973291



Punto 5. Error de Redondeo de x = 0.4

a. ¿Cómo se ajusta el número binario infinito a número finito en bits?
   El número real se escribe usando 32 bits; 1 para el signo, 23 del mantisa (es decir el coeficiente) y 8 bits para el exponente.

b. ¿Cuál es la diferencia entre redondeo y recorte?
   En el redondeo se descartan cifras del número teniendo en cuenta las cifras que siguen con el objetivo de conseguir/conocer un valor con mayor exactitud de la que se tiene en realidad. Mientras que el recorte consiste descartan cifras del número, pero sin tener en cuenta el valor de las cifras que le siguen

c. Indique el numero punto flotante de precisión asociado a x, para x = 0.4

Son un total de 25 decimales
0.4 = (0011111111011001100110011001100 1…..)$_2$

d. Error de Redondeo

$$\frac{|fl(x) - x|}{|x|} \leq \frac{1}{2} cmaq$$

= 8.237*10^{-7}

Punto 6. Formula iterativa de convergencia cuadrática para calcular la raíz n-ésima de un número real.

Para la realización de este punto fuer necesario aplicar el método de newton pensando en la fórmula:

$$x^2 = a$$

$$x = \sqrt[2]{a}$$

Por lo tanto, para cualquier raíz

$$x = \sqrt[b]{a}$$

$$x^b - a = 0$$

Para poder hacer esto solo se necesita una función de la forma dada por la anterior formula y donde x es el número cuya raíz cuadrada deseamos aproximar. Su respectiva derivada f'(x) seria, entonces aquella otra ecuación que se usa en el algoritmo de newton. El algoritmo desarrollado dio como resultado lo siguiente:

Ejemplo:

$a = 12 \ y \ b = \ 5$

```
x:  38.8 con:     1 Iteración
x:  31.04 con:    2 Iteración
x:  24.832 con:   3 Iteración
x:  19.86561 con: 4 Iteración
x:  15.8925 con:  5 Iteración
x:  12.71404 con: 6 Iteración
x:  10.17132 con: 7 Iteración
x:  8.137283 con: 8 Iteración
x:  6.510374 con: 9 Iteración
x:  5.209635 con: 10 Iteración
x:  4.170966 con: 11 Iteración
x:  3.344703 con: 12 Iteración
x:  2.694939 con: 13 Iteración
x:  2.201452 con: 14 Iteración
x:  1.863344 con: 15 Iteración
x:  1.68976 con:  16 Iteración
x:  1.64619 con:  17 Iteración
x:  1.643759 con:  18 Iteración
x:  1.643752 con:  19 Iteración
Iteraciones = 20
```

```
Resultado = 1.643752 es la raíz 5 del numero 12
```

Punto 7. Raíz real Positiva

a. Condiciones necesarias para que la raíz exista, sea única y pueda ser calculada.

Se entiende que una función posee una raíz si:

Sea f(x) una función real.
$f : S \subseteq \mathbb{R} \to S' \subseteq \mathbb{R}$

Además, f(x) debe de ser continua en un intervalo cerrado [a, b]

Una función **f es continua en un punto x_0** en el dominio de la función si:
$\forall \varepsilon > 0 \quad \exists \delta > 0$

tal que para toda **x** en el dominio de la función:

$\forall \varepsilon > 0 \quad \exists \delta > 0$

Teniendo el intervalo cerrado [a, b] se debe cumplir: $f(a) * f(b) < 0$

de cumplirse se tiene entonces que: ∃ n ∈ **R** & ∈ [a, b] tal que f (n) = 0

Se entiende que una raíz será única si:

∃ n ∈ **R** & ∈ [a, b] tal que f (n) = 0

$f'(x) \neq 0$ en todo (a, b).

b. Es fundamental en este ejemplo tener en cuenta la importancia del Epsylon, dependiendo del valor que se le otorgue a este, variará la cantidad de iteraciones que el algoritmo puede llegar a realizar.

•Dentro de un intervalo cerrado [a, b], donde el valor “d” que aumenta x después de cada iteración está denotado por (b-a) /10. Se puede observar entonces una relación entre el epsylon y “d”, donde la capacidad a la que puede crecer el “d” es directamente proporcional al epsylon.

Buscando ejemplificar esto, se puede plantear este algoritmo como:

teniendo, epsylon =

A medida que el epsylon sea más grande, el algoritmo convergerá cada vez más cerca a cero “0”.

c. Notación algorítmica

Variables de entrada:

a: Valor del intervalo cerrado a evaluar

b: Valor del intervalo cerrado a evaluar

Epsylon: El error que será tolerado, sirviendo como parámetro para detener el ciclo.

FuncionX: Al ser un algoritmo genérico, se ha denotado como “función” a la función a analizar, es decir, la función de donde se obtendrá la raíz buscada.