

*Juan Pablo Amorocho, Diego Fernando Trujillo, Jose Alejandro López, Nicolás Pérez Fonseca*

*Materia: Análisis Numérico*

*26/09/2020*

*Profesor: Eddy Herrera Daza*

# Taller 1

## 1.Número de Operaciones

### Ejercicio 1:

#### a)

Utilizar el método de Horner

R/Coeficientes del polinomio de ejemplo sacado del documento: -2, 0, -3, 3, -4

Soltando:

Número de operaciones: 8

Numero de sumas: 0.5

Numero de multiplicaciones: 4

#### b)

Evaluar en con , encontrar el error y compararlo con la derivada de la expresión equivalente de

R/El error de x sería igual a 5.1127708 con P(x) y con Q(x) sería igual a 5.1127708 dando el mismo valor para los dos siendo muy cercanos los dos tipos de funciones.

### Números Binarios

#### a)

Encuentre los primeros 15 bits en la representación binaria de π

R/Primeros 15 bits en binario de pi: 11.10110000111…

#### b)

Convertir los siguientes números binarios a base 10: 1010101; 1011.101; 10111.010101...; 111.1111...

R/Utilizando el código implementado se tiene

- 10 = 2

-1010101 = 85

-1011.101 = 11.5

- 10111.010101... = 23.21

-111.1111... = 7.15

#### c)

Convierta los siguientes números de base 10 a binaria: 11.25; 2 /3; 30.6; 99.9

R/Utilizando el código implementado se obtiene que

- 11.25= 1011.11001

-2 /3= 0.1000010

-30.6= 11110.110

- 99.9 = 1100011.1001

### Representación del Punto Flotante de los Números reales

#### Ejercicio 1:

¿Cómo se ajusta un número binario infinito en un número finito de bits?

R/Se pueden utilizar distintos métodos como el de redondeo o truncamiento para así representar un valor más pequeño para la aproximación finita de bits y asi tenerlo en valores más manejable para el proceso.

#### Ejercicio2:

¿Cuál es la diferencia entre redondeo y recorte?

R/En el redondeo simplemente se aproxima los valores en un lugar determinado para así seguir el bit siguiente ya sea para arriba o para abajo. Mientras que en el recorte se quitan bits en una posición determinada para mejorar el proceso.

#### Ejercicio 3:

¿Cómo se ajusta un numero binario finito en un numero finito de bits?

R/ Se intenta aproximar para que quede lo más parejo posible al poder ser un número tan grande para su uso por medio de método de acercamiento se tiende a utilizar una diferente precisión o forma de limitarla.

#### Ejercicio 4:

Indique el número de punto flotante (IEEE) de precisión doble asociado a x, el cual se denota como fl(x); para x(0.4)

R/ Representamos el entero en binario

0,4=100110011001100…

que también podría ser visto como

1,100110011001…

luego definimos que el exponente es -1 por lo tanto la característica es 1022 que equivale a expresarlo en forma binaria como

1111111110.

La matisa corresponde a los primeros 52 dígitos después de la coma (“,”) y se utiliza el método de redondeo para aproximar lo obtenido dando

10011001…10010

por lo que el final termino así entonces fue redondeado hacia arriba.

El número 0.4 en IEEE corresponde a:

#### Ejercicio 5:

Error de redondeo En el modelo de la aritmética de computadora IEEE, el error de redondeo relativo no es más de la mitad de la épsilon de maquina:

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

R/El error relativo corresponde a:

El error de la maquina seria:

#### Ejercicio 6:

Verificar si el tipo de datos básico de R y Python es de precisión doble IEEE y Revisar en R y Phython el formato long.

R/Python y R utilizan la aritmética de punto flotante IEEE-754 para el mapeo de los datos se usaría doble de precisión de IEEE-754

#### Ejercicio 7:

Encuentre la representación en número de máquina hexadecimal del número real 9.4

R/ Se utiliza la herramienta hexmode en R para encontrar el valor hexadecimal de un número real que en este caso seria 9.4 que se expresaría como 9.4 también.

#### Ejercicio 8:

Encuentre las dos raíces de la ecuación cuadrática Intente resolver el problema usando la arimética de precisión doble, tenga en cuenta la pérdida de significancia y debe contrarrestarla.

R/Utilizando wólfram Alpha se obtiene que: y .

Y al comparar ambos resultados con precisión doble al restarlos serian:

Por lo visto la diferencia al restarlo es grandísima por lo que pueden ser respuestas independientes.

#### Ejercicio 9:

Explique cómo calcular con mayor exactitud las raíces de la ecuación:

Donde b es un número mayor que 100

R/Se podría utilizar algún método iterativo para simplificar el trabajo como por ejemplo em método de bisección ya que lo podríamos limitar a una tolerancia fija para que el valor de b no sea tan demandante y si no fuera útil solo se detendría para tomarlo desde otro enfoque.

## 2.Raíces de una Ecuación

### Ejercicio 1:

Implemente en R o Python un algoritmo que le permita sumar únicamente los elementos de la sub matriz triangular superior o triangular inferior, dada la matriz cuadrada An. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notación O() con una gráfica que muestre su orden de convergencia.

Se implemento un algoritmo que suma la parte triangular inferior de una matriz incluyendo la diagonal. Como entrada de datos fue un arreglo con los siguientes valores:

El algoritmo creaba automáticamente las matrices con todos sus valores en 1 y contaba cuantas sumas se realizabas en cada matriz que también sería el resultado de la suma de los valores de la triangular inferior debido a que todos los valores son igual a 1. Se obtuvo los siguientes datos:

|  |  |
| --- | --- |
|  | Número de Sumas |
| 5 | 15 |
| 10 | 55 |
| 15 | 120 |
| 20 | 210 |
| 25 | 325 |
| 30 | 465 |
| 35 | 630 |
| 40 | 820 |
| 45 | 1035 |
| 50 | 1275 |

Gráfico, Gráfico de líneas

Descripción generada automáticamenteAl realizar la gráfica de los datos se obtiene lo siguiente:

Se puede observar que la función crece cuadráticamente, por tanto, la función f(n) tiene complejidad .

### Ejercicio 2:

Implemente en R o Python un algoritmo que le permita sumar los primeros números naturales al cuadrado. Imprima varias pruebas, para diferentes valores de n y exprese f(n) en notación O() con una gráfica que muestre su orden de convergencia.

Se resolvió este problema de dos formas, primero se implementó un algoritmo que simplemente hiciera la suma de los valores, se hicieron pruebas con los siguientes valores:

Se obtuvieron los siguientes resultados:

|  |  |
| --- | --- |
|  | Resultado de la suma |
| 10 | 385 |
| 20 | 2870 |
| 30 | 9455 |
| 40 | 22140 |
| 50 | 42925 |
| 60 | 73810 |
| 70 | 116795 |
| 80 | 173880 |
| 90 | 247065 |
| 100 | 338350 |

Gráfico, Gráfico de líneas

Descripción generada automáticamenteAl realizar la gráfica de los resultados se obtiene lo siguiente:

Se puede observar que la crece en función , quiere decir que, crece muy rápido cada vez que se aumenta . En cambio, si se analiza la complejidad del algoritmo es ya que siempre se van a ser sumas.

La segunda forma en la que se solucionamos el problema fue realizando la fórmula de la suma que es la siguiente:

Al realizar las pruebas con los mismos valores se obtiene el mismo resultado que con la primera forma que solucionamos. Pero algo a observar es que es mejor utilizar esta segunda forma ya que la complejidad del algoritmo es ya que no ha de hacer iteraciones, si no que obtiene el resultado de una vez, esto ahorra recursos para el computador y tiempo de ejecución.

### Ejercicio 3:

Para describir la trayectoria de un cohete se tiene el modelo:

Donde, y es la altura en [m] y t tiempo en [s]. El cohete está colocado verticalmente sobre la tierra. Utilizando dos métodos de solución de ecuación no lineal, encuentre la altura máxima que alcanza el cohete.

Para encontrar el punto más alto se tiene que derivar la función y con la derivada encontrar la raíz. Se hizo la derivada por medio de la aplicación de wolframalpha dando como resultado lo siguiente:

Primero se trató de encontrar la raíz por medio de la implementación del método de punto fijo y Newton-Rapson pero se encontró el problema que esta función también tiene una raíz en 0, por tanto, estos dos métodos siempre convergía hacia el punto 0.

Por tanto, se utilizó la función polyroot() que está disponible en R, esta función encuentra las raíces de una función por medio del método de Jenkins-Traub(1972), al pasarle como parámetro se obtuvo los siguientes valores:

* 0.00000000000000+0i
* 28.62220762329085-0i
* -28.62220762329085+0i

Para comprobar el valor también se utilizó la función uniroot() que también encuentra las raíces de una función, al pasarle como parámetro se obtuvo el siguiente valor:

* 28.62220762329078

Al hacer esto ya sabemos que el valor de la raíz es aproximadamente 28.622207, ya por último implementamos el algoritmo de Brent que realizamos en el reto 1 y nos arrojó el siguiente resultado:

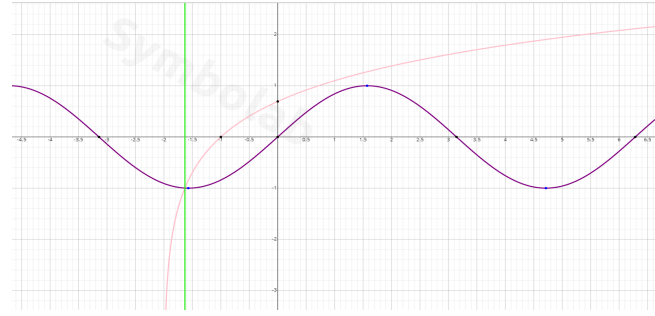
* 28.6222076267004 con una tolerancia de

Por tanto, se puede concluir que el valor de la raíz es de 28.62220762 que al remplazarlo en da:

En conclusión, la máxima altura del cohete es de 879.48076 metros en el segundo 28.62220762.

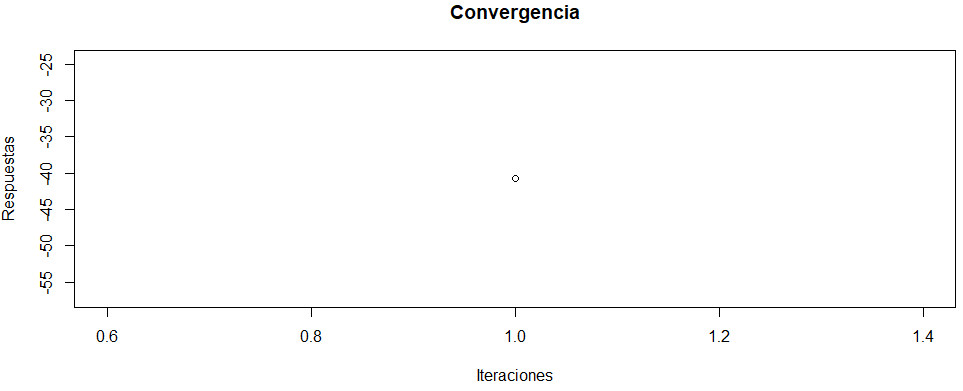
## 3.Convergencia de Métodos Iterativos

### Ejercicio 1:

Sean y dos funciones de valor real.

#### a)

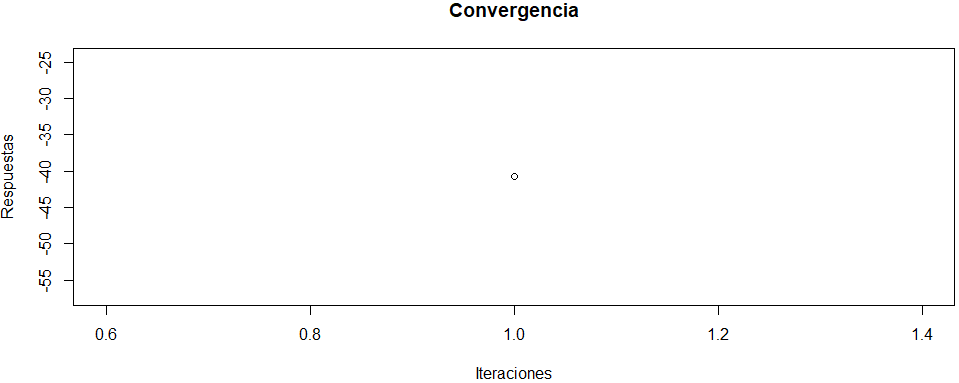
Utilice la siguiente formula recursiva con para determinar aproximadamente el punto de intersección.:

Aplicando la fórmula sin importar el punto de partida que tengo luego de la segunda iteración el algoritmo cae en indeterminación. Lo que sólo permite graficar un solo punto y no permite ver con claridad el comportamiento de este para determinar su complejidad mediante la notación de O grande.

Se supone que la complejidad de estos algoritmos debería ser cuadrática es decir .

#### b)

Aplicar el método iterativo siguiente con para encontrar el punto de intersección:

Así sea variando el error este algoritmo iterativo, por lo general converge cuando se aproxima a su respuesta. En este caso tampoco pasa de la segunda iteración pues ambas funciones se van acercando a cero con cada interacción con la fórmula mas no se acerca a la intersección de las dos curvas. Para este caso la complejidad de este podría ser: pero tampoco se puede estar seguro porque el comportamiento no se deja ver ya que solo permite graficar un solo punto antes de quedar indeterminada el cálculo de las imágenes en las funciones.

### Ejercicio 2:

Determine el valor de los coeficientes de a y b tal que: y con obtenga la respuesta con .

Tenemos el siguiente sistema de ecuaciones no lineales, porque tenemos exponenciales.

A continuación, se procede a resolver el sistema de ecuaciones:

### Newton

#### Ejercicio1:

Demuestre que tiene un cero de multiplicidad 2 en x=0:

Luego, para demostrar su multiplicidad debe poder escribirse así:

Tomamos a q(x) como;

Y a el primer producto como:

El exponente que nos indica la multiplicidad del cero lo vamos a llamar a y es equivalente

Sabiendo que la complicidad es 2 porque nos lo dicen desde el comienzo, podemos averiguar w

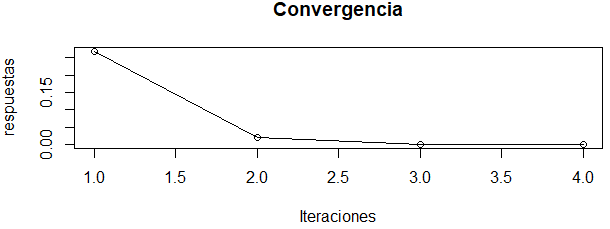
Reemplazando en la expresión general de la multiplicidad según el teorema tenemos:

Por lo tanto, como pudo ser reescrita de esta forma, se comprueba que su multiplicidad 2.

#### Ejercicio2:

Utilizando el método de Newton con verifique que converge a cero, pero no de forma cuadrática.

Luego de ejecutar el algoritmo, podemos ver como la convergencia de este a medida que pasan las iteraciones se acerca a cero, pero forma solamente media parábola como se puede ver en el gráfico que tenemos a continuación.

La salida del programa muestra a continuación la siguiente salida:



Con el error más alto permitido por R estándar, lo hace en 4 iteraciones, cada punto representa una iteración.

#### Ejercicio3:

¿Utilizando el método de Newton generalizado, mejora la tasa de rendimiento? explique su respuesta.

Al parecer, requiere una iteración menos en hallar la raíz de la función, más sin embargo requiere de la segunda derivada para llevarse a cabo. A continuación, la comparación lado a lado de las dos corriendo en igualdad de condiciones.

|  |  |
| --- | --- |
| Newton | Newton Generalizado |
|  |  |

Acá se puede ver cómo le toma una iteración menos en hallar la raíz de la función.

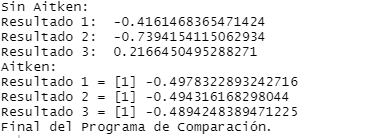
## 4.Convergencia acelerada

### Método de ∆2 Aitken

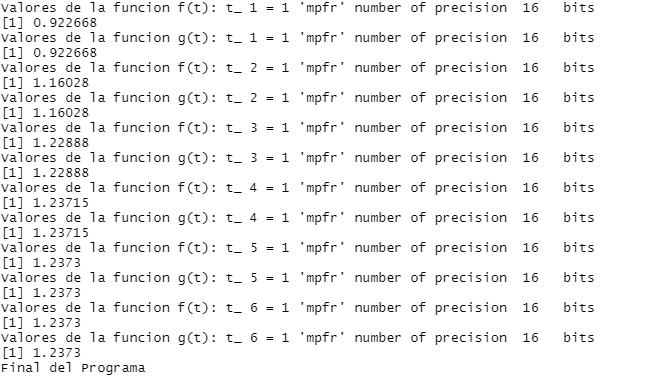
#### Ejercicio 1:

1. La función . En este caso, similar a con el seno, se encuentra fácilmente que O()

**2.** Para comparar la diferencia entre usar el método de Aitken, se hizo una iteración de punto fijo sobre la función  **.** Se compararon los primeros tres valores, y se puede ver la gran diferencia que cause esta aceleración:



**3.** Al usar el método de punto fijo con el método de Aitken (En este caso se usó el método de Steffensen, ya que da una respuesta, aunque se puede llegar a la misma conclusión usando Aitken), se puede ver que las dos funciones resultan con los mismos valores en las iteraciones.

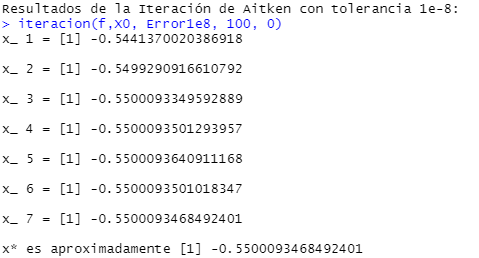


### Método de Steffensen

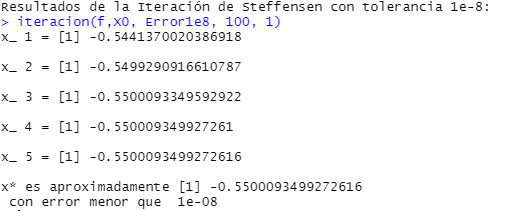
Se aplicó el método de Steffensen para la función y se comparó con el método de Aitken, resultando en las siguientes conclusiones:

#### Con Tolerancia:

##### Aitken:



##### Steffensen:



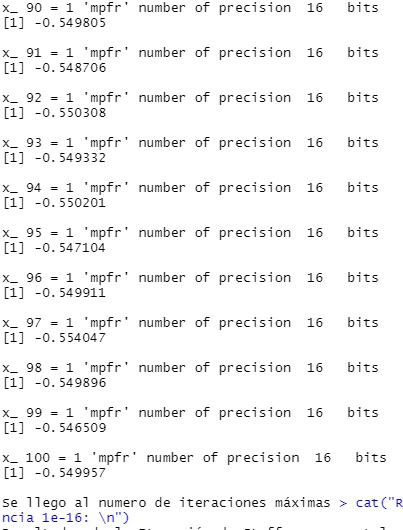
#### Conclusiones:

Para los dos métodos, se puedo encontrar una raíz. Claramente se puede ver que, para este nivel de tolerancia, se encontraron las raíces de manera rápida. Steffensen encuentra la respuesta de una manera más rápida, ya que, por su naturaleza, cambia el orden de convergencia de la función a una cuadrada.

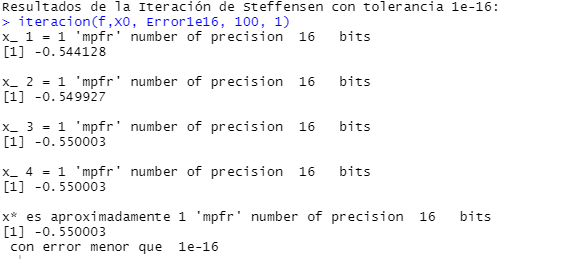
#### Con Tolerancia:

Con esta tolerancia, se tuvo que usar la librería Rmpfr para incrementar la precisión.

##### Aitken:



##### Steffensen:



#### Conclusiones:

Podemos ver dos resultados totalmente distintos para los dos métodos. Gracias una restricción mayor en términos de tolerancia, el método de Aitken no puede llegar a un resultado, ya que sobrepasa incluso las 100 iteraciones, y los datos por cada iteración resultan en un ciclo, nunca llegando a la raíz deseada. Steffensen, llega a la respuesta en cuatro iteraciones, significando que una mayor tolerancia no cambia en mucho el tiempo que le toma a Steffensen a llegar a su respuesta. Esta respuesta toma menos iteraciones que la anterior gracias a la mayor precisión usada en esta comparación.

# Referencias

|  |  |
| --- | --- |
| [1] | J. C. Gorostizaga, «EHU,» [En línea]. Available: http://www.ehu.eus/juancarlos.gorostizaga/mn11b/temas/horner.pdf. [Último acceso: 26 Septiembre 2020]. |
| [2] | Zator, «Zator Curso de C++,» 2016. [En línea]. Available: https://www.zator.com/Cpp/E2\_2\_4a.htm. [Último acceso: 26 Septiembre 2020]. |
| [3] | W. Alpha, «Wolfram Alpha,» Wolfram Alpha, 2020. [En línea]. Available: https://www.wolframalpha.com/input/?i=x%5E2%2B9%5E12+x%3D3+. [Último acceso: 26 Septiembre 2020]. |
| [4] | Desconocido, «Tarwi,» Universidad Agraria del Perú, 2008. [En línea]. Available: https://tarwi.lamolina.edu.pe/~fmendiburu/index-filer/academic/script\_numerico.htm. [Último acceso: 26 Septiembre 2020]. |