



FACULTAD DE INGENIERIA

Universidad de Buenos Aires

ANÁLISIS NUMÉRICO I

PRIMER CUATRIMESTRE DE 2021

Trabajo Práctico 1: Búsqueda de raíces

Alumno	Número de padrón	Email
BOCACCIO, Agustina	106393	abocaccio@fi.uba.ar
SINGER, Joaquín	105854	josinger@fi.uba.ar
LAZZARO, Melina	105931	mlazzaro@fi.uba.ar

Curso: Sassano

Corrector: VERA GUZMAN, Ramiro

Lenguaje de programación utilizado: python

26 de Mayo de 2021

Índice

1. Introducción	2
2. Objetivos	2
3. Especificaciones sobre el desarrollo del trabajo	2
3.1. Método para seres queridos	2
3.2. Hallar π por dos caminos	2
3.3. Búsqueda de raíces	3
3.3.1. Gráficos del orden de convergencia con ambas cotas	4
3.3.2. Gráficos de la constante asintótica con ambas cotas	5
4. Conclusiones	6
5. Referencias	7

1. Introducción

El presente informe sera utilizado para especificar y analizar lo realizado en el Trabajo Practico 1 de Análisis Numérico I, en el cual se estudiaron métodos de búsqueda de raíces y su aplicación en diferentes problemas planteados.

2. Objetivos

Se estudian distintos métodos de búsqueda de raíces desarrollando los algoritmos de **Newton Raphson**, **serie de Leibniz**, **Newton Raphson modificado**, **Método de la Secante** y **Bi-sección**. Estos métodos fueron utilizados para hallar π , hallar raíces de tres funciones determinadas y para ayudar a un ser querido. Se realizará una comparativa entre los mismos analizando ventajas y desventajas de cada uno, también calculando su orden de convergencia y constante asintótica.

3. Especificaciones sobre el desarrollo del trabajo

3.1. Método para seres queridos

Se busca aplicar uno de los métodos vistos en clase en algo de la vida cotidiana para ayudar a un ser querido. Se decidió utilizar el **método de la bisección** ya que es fue considerado como el mas sencillo de aplicar en lo cotidiano al ser el mas intuitivo y no necesitar hacer cálculos complejos de funciones y derivadas. Lo aplicamos para ayudar a un hermano/a pequeño a aprender a usar un diccionario y detallamos el proceso y las conclusiones en el archivo `TP1_ramiro_6_2021_05_26.ipynb`.

Este ejercicio permitió aplicar uno de los métodos aprendidos (en este caso el de la bisección), llevándolo a un plano mas cotidiano, reforzando el aspecto teórico aprendido durante las clases.

3.2. Hallar π por dos caminos

El objetivo de esta parte es encontrar π usando la **serie de Leibniz** y el método de búsqueda de raíces de **Newton-Raphson** en 64 y 32 bits, con la máxima precisión de la herramienta en el caso de Newton Raphson y con distinta cantidad de iteraciones en Leibniz.

Para Newton Raphson fue utilizada la función $\sin(x)$, que cumple con las condiciones del método ya que es continua al igual que su derivada, y la misma no se anula en π . La semilla elegida fue $x_0 = 3$ ya que el método además requiere una semilla que esté lo suficientemente cerca de la raíz.

Tipo	signo	mantisa	exponente	total	Emin	Emax	Bits prec	díg sign
Half	1	10	5	16	-14	+15	11	3.3
Single	1	23	8	32	-126	+127	24	7.2
Double	1	52	11	64	-1022	+1023	53	15.9

Figura 1: Precisión de los tipos de dato en python

Se puede observar que en 64 bits python nos asegura 15 dígitos significativos, y en el caso de 32 bits, 7 dígitos significativos.

Además, fue utilizada una calculadora casio **fx-570LA X** para obtener π con ambos métodos y el error indicado en su manual.

Frente a los resultados obtenidos, se puede obtener como conclusión que, de entre los métodos utilizados para hallar π en esta sección, el de Newton-Raphson en una arquitectura de 63 bits fue el mas óptimo, ya que se obtuvo una mejor aproximación, con un error menor y con menos iteraciones.

Todo lo mencionado, los resultados y la implementación de los métodos se encuentran detallados en el archivo entregado.

3.3. Búsqueda de raíces

En esta sección del trabajo se utilizaron los diferentes métodos aprendidos en clases para hallar las raíces de tres funciones y estudiar la convergencia de cada uno. Estos son:

- Método de Newton- Raphson
- Método de Newton-Raphson modificado
- Bisección
- Método de la secante

Para estudiar estos métodos y usarlos en la búsqueda de raíces, se comenzó por graficar las funciones en el intervalo que se pidió estudiar, y luego se desarrollaron los algoritmos de los métodos en Python.

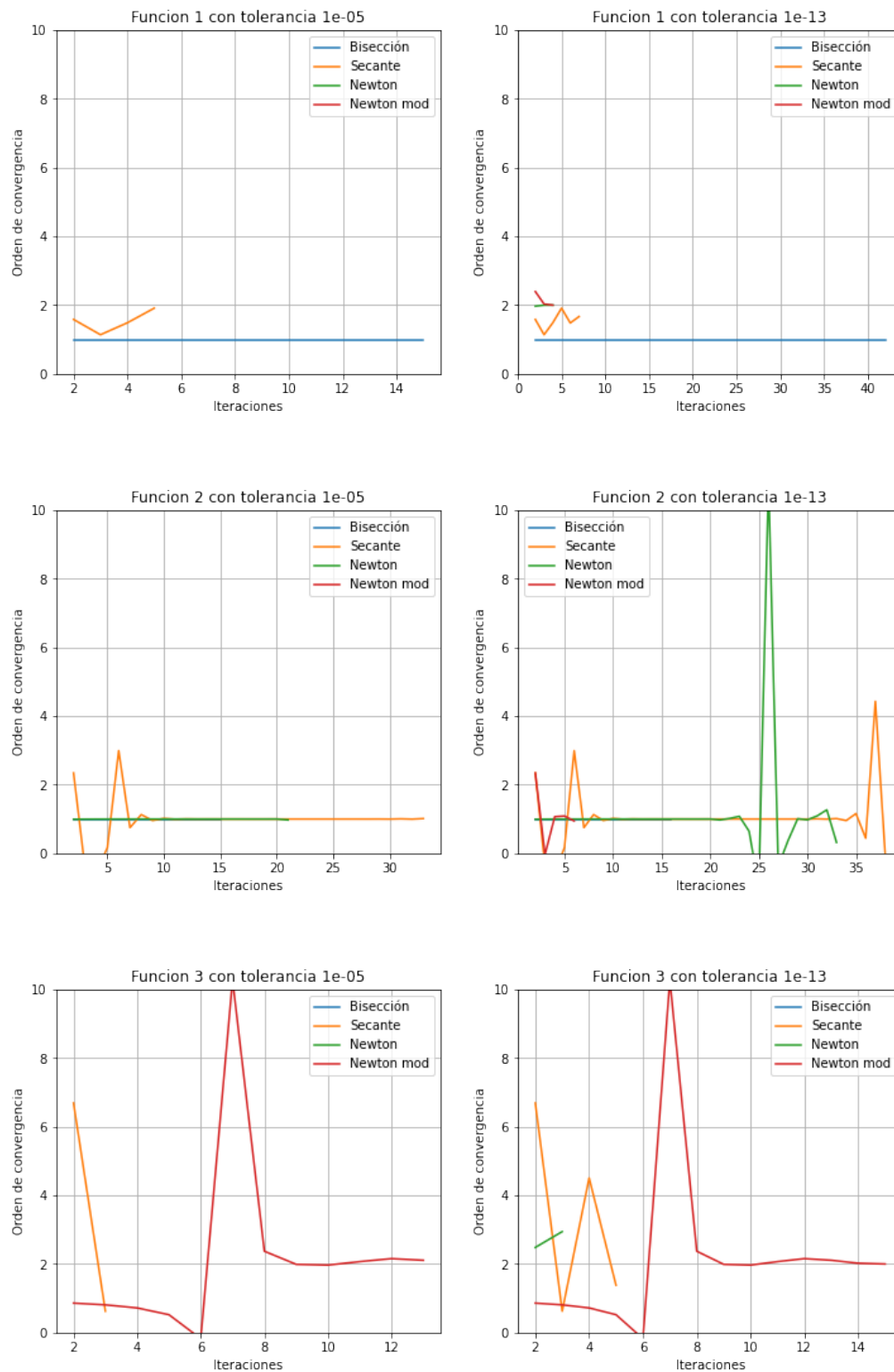
Una vez desarrollados los algoritmos de los métodos, fueron usados para hallar la raíz de cada una de las funciones en el intervalo solicitado. Como criterio de paro se todos los métodos se usaron las cotas de error $1 * 10^{-5}$ y $1 * 10^{-13}$, es decir que el modulo de la diferencia entre dos iteraciones sucesivas sea menor a ese valor. Además, se uso $x_0 = 1,0$ como semilla para Newton-Raphson y Newton-Raphson modificado, y a los extremos del intervalo (0 y 2) como semillas del método de la secante.

En la función 3, se observó que con la o las semillas utilizadas en el método de Newton-Raphson, el de Newton-Raphson modificado, y el de la secante, estos procedimientos no pudieron converger. El motivo de esto fue explicado en el archivo entregado, junto con el nuevo calculo con este método con mejores semillas (elegidas basada en el gráfico de la función en este caso, pero también se podría obtener una semilla mejor usando el método de la bisección primero y a partir de allí usar Newton-Raphson).

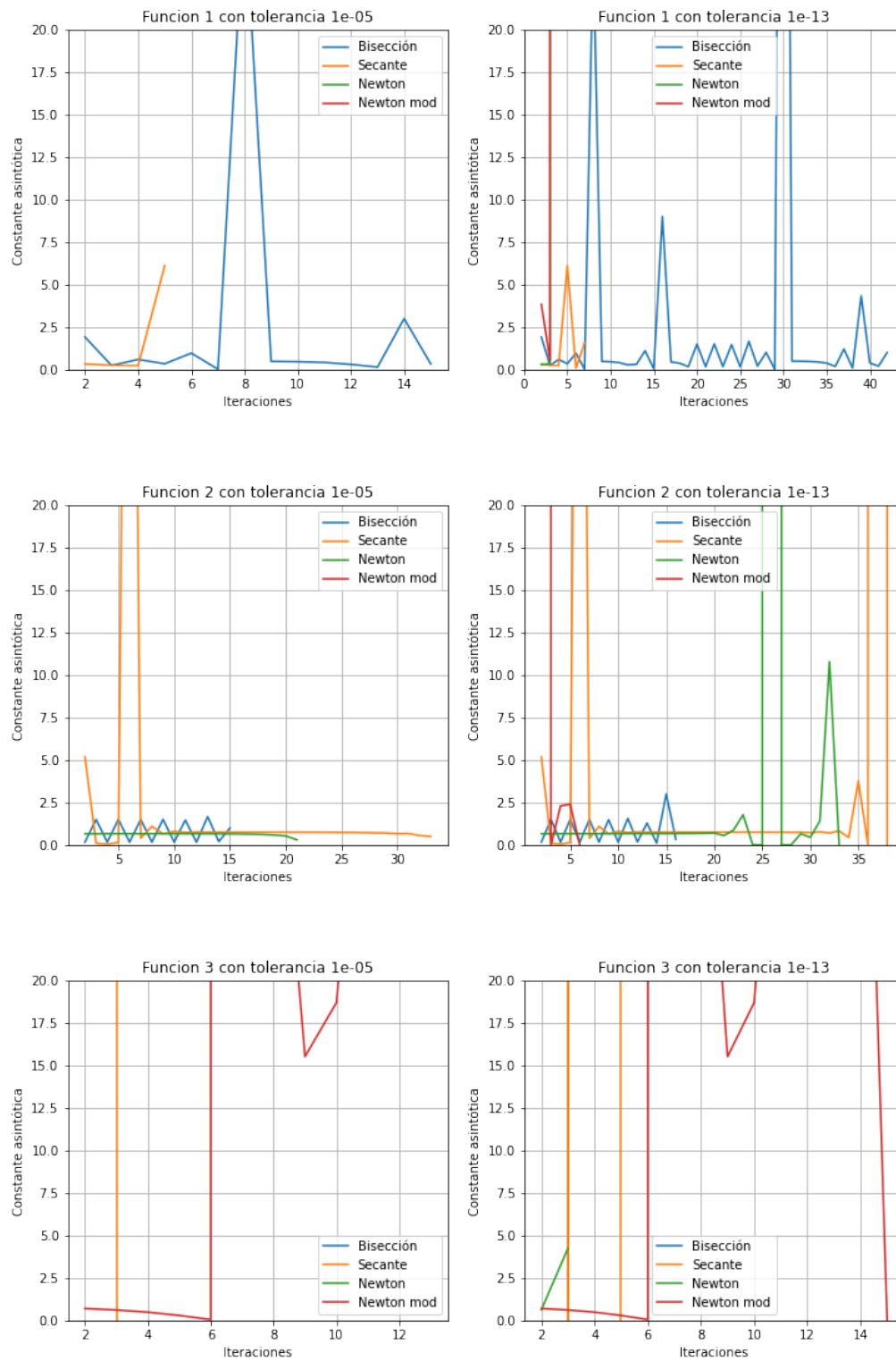
En la ultima parte de este enunciado fueron implementados algoritmos para hallar el orden de convergencia y la constante asintótica para cada método con las diferentes funciones y las distintas cota de error. Ver las secciones **Gráficos del orden de convergencia con ambas cotas** y **Gráficos de la constante asintótica con ambas cotas**.

En esta parte del trabajo se pudieron observar las ventajas y desventajas del uso de cada método de búsqueda de raíces para diferentes funciones. El método de Newton-Raphson es muy poderoso por ser el de mayor convergencia (cuadrática para raíces simples), sin embargo para usarlo se necesita la derivada de la función estudiada, tiene convergencia lineal para raíces dobles y necesita de una buena aproximación inicial. Esto ultimo se debe a que de no tener una buena semilla, el método no va a converger, lo cual lo diferencia del de la bisección, el cual, a pesar de hacerlo mas lento, (linealmente), siempre converge. El otro procedimiento estudiado, el de la secante, tiene mayor convergencia que el de la bisección, pero menor que el de Newton-Raphson, y como este, necesita una buena aproximación inicial (y en su caso, no solo una, sino dos semillas) para converger. Sin embargo, tiene la ventaja de que no precisa la derivada de la función que se esta analizando.

3.3.1. Gráficos del orden de convergencia con ambas cotas



3.3.2. Gráficos de la constante asintótica con ambas cotas



Los casos donde no se ven graficados algunos métodos es porque convergen demasiado rápido y no hay suficiente información para hacer los cálculos. No se pueden sacar conclusiones definitivas con respecto a la convergencia de cada método ya que teóricamente se calcula con infinitas iteraciones, cosa que es imposible para una computadora. Aumentar el número de iteraciones tampoco sería viable ya que llegará un momento en que la diferencia entre iteraciones será tan pequeña que harán underflow.

4. Conclusiones

Frente a las soluciones, aplicaciones y resultados obtenidos en el trabajo, se pueden determinar conclusiones basadas en los mismos.

En la primera parte del trabajo se pudo encontrar una forma de aplicar uno de los métodos aprendidos a un problema de la vida cotidiana, lo cual permitió darle un uso práctico y más cercano a lo estudiado, más allá del aspecto teórico del mismo. El uso de la bisección hizo que el ser querido solucionara de forma más eficiente su problema, ya que el niño/a hubiese tardado más en leer hoja por hoja el diccionario que utilizando el método que se le fue dado.

De la segunda parte del trabajo, en la cual se halló π mediante diferentes métodos, se puede concluir que el método de Newton-Raphson en una arquitectura de 64 bit fue el que obtuvo la mejor aproximación con un error más pequeño que Newton-Raphson de 32 bit y Leibniz (en la calculadora o con la implementación de 64 bit), y también con una menor cantidad de iteraciones. Esto se debe a que la arquitectura de 64 bit proporciona una mayor precisión numérica, lo cual permite tener un error menor, pero nunca cero, ya que como fue explicado en el trabajo, el número π no es una constante para la computadora, porque cada número en ella se representa mediante un número finito de dígitos, y al ser π irracional, su representación va a variar y va a tener un error determinado.

De la tercera parte se puede concluir que cada método de búsqueda de raíces tiene sus ventajas y desventajas, ya que si bien teóricamente Newton-Raphson tiene una convergencia mayor, tiene más requisitos y en algunos casos, como sucede en este caso con la función 3, se tendrá que probar con distintas semillas para que pueda converger y no siempre se realizarán menos iteraciones que en los demás métodos. Las constantes asintóticas y orden de convergencia calculados mostraron no ser consistentes y dependen fuertemente de las condiciones iniciales como la función y semillas, además del número de iteraciones.

5. Referencias

[1] Apuntes del curso Análisis numérico 1 - curso Sassano - Facultad de Ingeniería - Universidad de Buenos Aires - 2021.

[2] https://es.wikipedia.org/wiki/Serie_de_Leibniz