

Trabajo Práctico 1

[75.12 - 95.04] Análisis Numérico
Curso 5
Primer cuatrimestre de 2020

| | |
|------------------------|--------|
| De Angelis Riva, Lukas | 103784 |
| Gomez, Joaquín | 103735 |
| Grassano, Bruno | 103855 |
| Guillemi, Andrés | 104006 |
| Rodriguez, Ezequiel | 103976 |
| Romero, Adrián | 103371 |

Índice

| | |
|--|-----------|
| 1. Introducción | 2 |
| 2. Objetivos | 2 |
| 3. Gráficos de las funciones | 3 |
| 4. Búsqueda de raíces | 5 |
| 4.1. Bisección | 5 |
| 4.1.1. Primera función | 5 |
| 4.1.2. Segunda función | 6 |
| 4.1.3. Tercera función | 6 |
| 4.2. Newton-Raphson | 7 |
| 4.2.1. Primera función | 7 |
| 4.2.2. Segunda función | 7 |
| 4.2.3. Tercera función | 7 |
| 4.3. Newton-Raphson modificado | 8 |
| 4.3.1. Primera función | 8 |
| 4.3.2. Segunda función | 8 |
| 4.3.3. Tercera función | 8 |
| 4.4. Secante | 8 |
| 4.4.1. Primera función | 9 |
| 4.4.2. Segunda función | 9 |
| 4.4.3. Tercera función | 9 |
| 5. Comparación de resultados | 10 |
| 5.0.1. Primera función | 10 |
| 5.0.2. Segunda función | 12 |
| 5.0.3. Tercera función | 14 |
| 6. Conclusiones | 16 |

1. Introducción

En este trabajo práctico creamos un programa para la búsqueda de raíces de funciones a través de los siguientes métodos numéricos aprendidos en la materia:

- Bisección
- Newton-Raphson
- Newton-Raphson modificado
- Secante

Con este fin, primero graficamos las funciones en Geogebra para tener una idea sobre ellas, corroborando que los intervalos brindados en el enunciado eran válidos para los métodos. Luego, obtenemos las raíces y con la historia de los resultados pasamos a analizar los distintos métodos analizando los ordenes de convergencia, su conveniencia en tiempos de cómputo y error en el resultado.

Para verificar los resultados, calculamos las raíces mediante la biblioteca de Python ‘SciPy’. De esta manera pudimos corroborar el correcto funcionamiento de las funciones programadas.

2. Objetivos

El objetivo del trabajo práctico es obtener la raíz de las tres funciones dadas mediante los métodos numéricos indicados, para luego analizar el orden de convergencia P y la constante asintótica de cada método. λ

3. Gráficos de las funciones

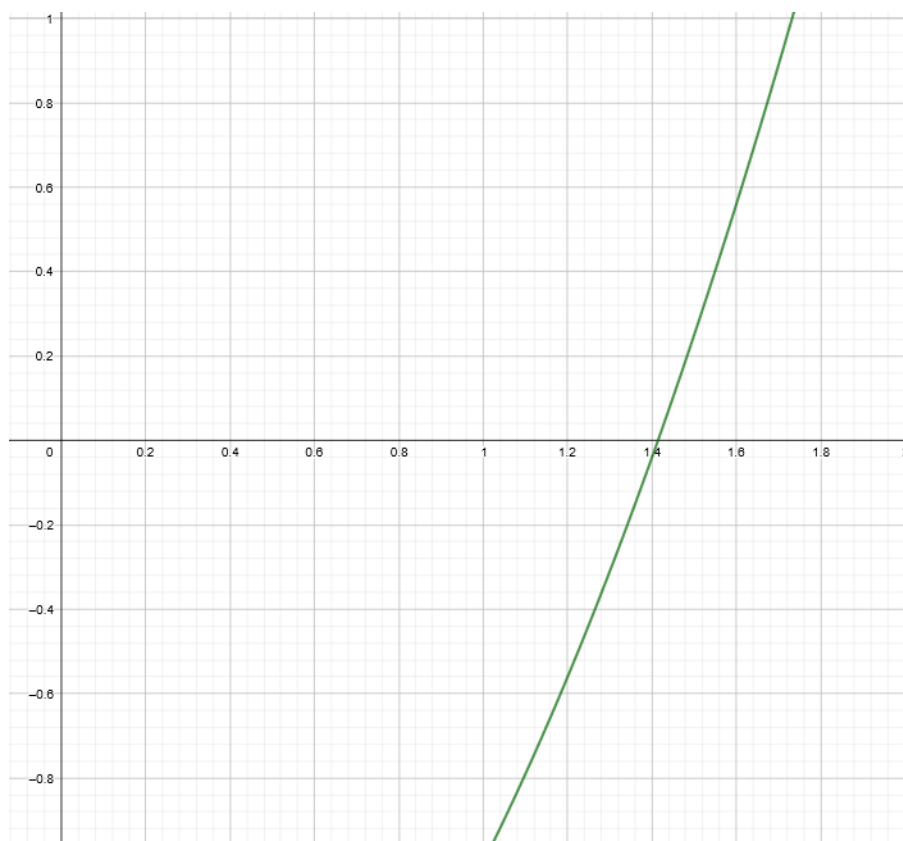


Figura 1: Gráfico correspondiente a la primera función en el intervalo $[0; 2]$.

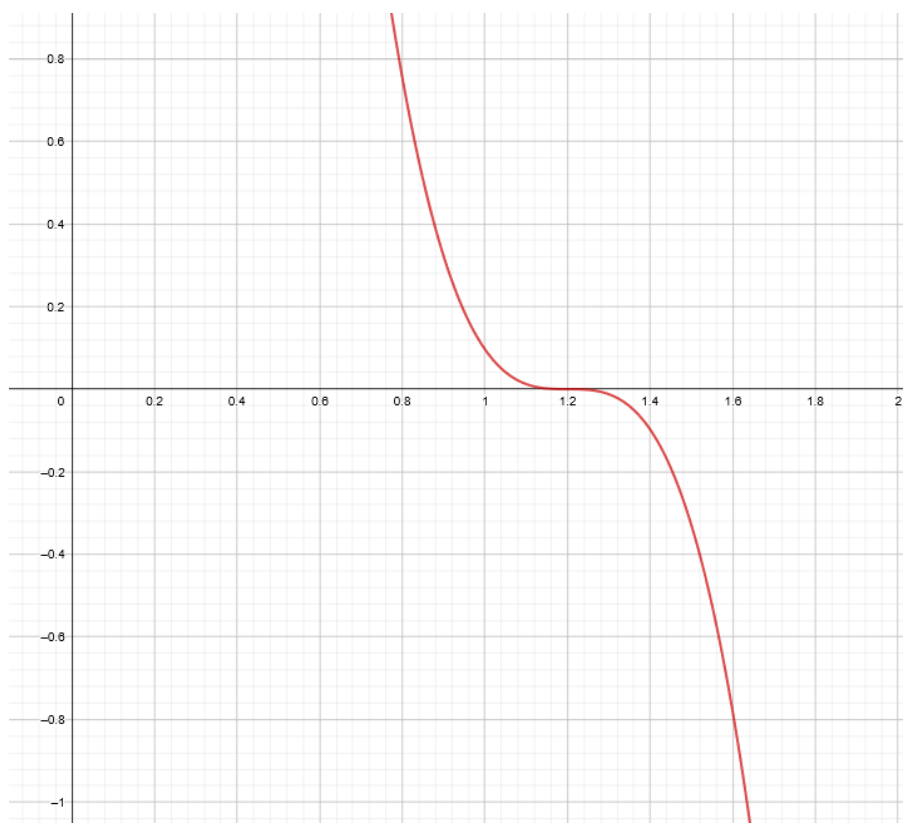


Figura 2: Gráfico correspondiente a la segunda función en el intervalo $[0;2]$.

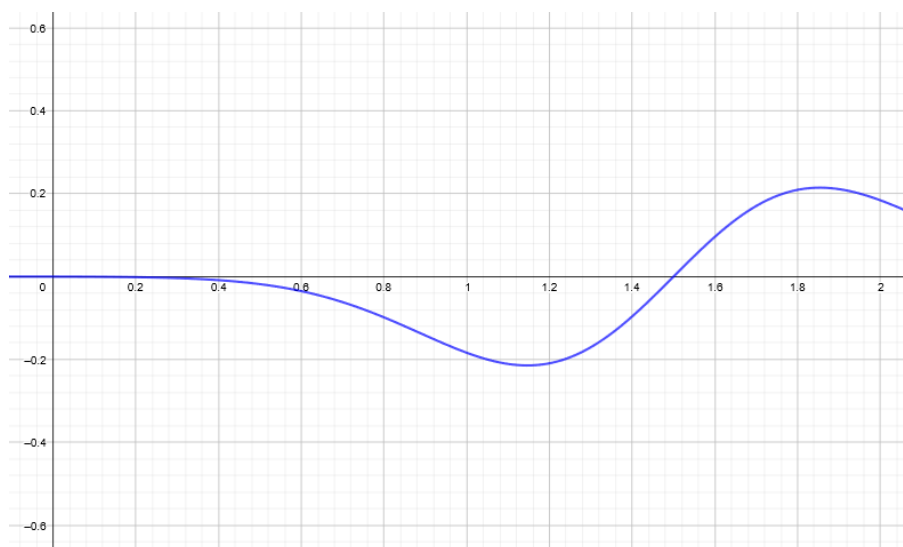


Figura 3: Gráfico correspondiente a la tercera función en el intervalo $[0;2]$.

4. Búsqueda de raíces

4.1. Bisección

El método de bisección halla la raíz mediante la siguiente fórmula por métodos iterativos hasta que $|p_{n+1} - p_n| < \epsilon$, tomando los límites iniciales de intervalo a_n y b_n :

Sea $p_n = \frac{a_n + b_n}{2}$, entonces, en la siguiente iteración

$$a_{n+1} = \begin{cases} a_n & \text{si } f(a_n) \cdot f(r_n) < 0 \\ p_n & \text{si } f(a_n) \cdot f(r_n) > 0 \end{cases} \quad b_{n+1} = \begin{cases} b_n & \text{si } f(b_n) \cdot f(r_n) < 0 \\ p_n & \text{si } f(b_n) \cdot f(r_n) > 0 \end{cases}$$

Los resultados obtenidos fueron los siguientes:

Para las tres funciones, observamos que la cota $1 \cdot 10^{-5}$ se alcanza en la iteración 18

4.1.1. Primera función

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.0 |
| 1 | 1.5 |
| 2 | 1.25 |
| 3 | 1.375 |
| 4 | 1.4375 |
| 5 | 1.40625 |
| 6 | 1.421875 |
| 7 | 1.4140625 |
| 8 | 1.41796875 |
| 9 | 1.416015625 |
| 10 | 1.4150390625 |
| 11 | 1.41455078125 |
| 12 | 1.414306640625 |
| 13 | 1.4141845703125 |
| 14 | 1.41424560546875 |
| 15 | 1.414215087890625 |
| 16 | 1.4141998291015625 |
| 17 | 1.4142074584960938 |
| 18 | 1.4142112731933594 |
| 19 | 1.4142131805419922 |
| 20 | 1.4142141342163086 |
| 21 | 1.4142136573791504 |
| 22 | 1.4142134189605713 |

| Iteración | Resultado |
|-----------|--------------------|
| 23 | 1.4142135381698608 |
| 24 | 1.4142135977745056 |
| 25 | 1.4142135679721832 |
| 26 | 1.414213553071022 |
| 27 | 1.4142135605216026 |
| 28 | 1.414213564246893 |
| 29 | 1.4142135623842478 |
| 30 | 1.4142135614529252 |
| 31 | 1.4142135619185865 |
| 32 | 1.4142135621514171 |
| 33 | 1.4142135622678325 |
| 34 | 1.4142135623260401 |
| 35 | 1.414213562355144 |
| 36 | 1.4142135623696959 |
| 37 | 1.4142135623769718 |
| 38 | 1.4142135623733338 |
| 39 | 1.4142135623715149 |
| 40 | 1.4142135623724243 |
| 41 | 1.414213562372879 |
| 42 | 1.4142135623731065 |
| 43 | 1.4142135623729928 |
| 44 | 1.4142135623730496 |

4.1.2. Segunda función

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.0 |
| 1 | 1.5 |
| 2 | 1.25 |
| 3 | 1.125 |
| 4 | 1.1875 |
| 5 | 1.21875 |
| 6 | 1.203125 |
| 7 | 1.1953125 |
| 8 | 1.19921875 |
| 9 | 1.201171875 |
| 10 | 1.2001953125 |
| 11 | 1.19970703125 |
| 12 | 1.199951171875 |
| 13 | 1.2000732421875 |
| 14 | 1.20001220703125 |
| 15 | 1.199981689453125 |
| 16 | 1.1999969482421875 |
| 17 | 1.2000045776367188 |
| 18 | 1.2000083923339844 |
| 19 | 1.2000064849853516 |
| 20 | 1.2000055313110352 |
| 21 | 1.200005054473877 |
| 22 | 1.200005292892456 |

| Iteración | Resultado |
|-----------|--------------------|
| 23 | 1.2000051736831665 |
| 24 | 1.2000051140785217 |
| 25 | 1.2000051438808441 |
| 26 | 1.2000051587820053 |
| 27 | 1.2000051513314247 |
| 28 | 1.2000051476061344 |
| 29 | 1.2000051457434893 |
| 30 | 1.2000051466748118 |
| 31 | 1.2000051471404731 |
| 32 | 1.2000051473733038 |
| 33 | 1.200005147489719 |
| 34 | 1.2000051475479268 |
| 35 | 1.2000051475770306 |
| 36 | 1.2000051475624787 |
| 37 | 1.2000051475552027 |
| 38 | 1.2000051475588407 |
| 39 | 1.2000051475570217 |
| 40 | 1.2000051475561122 |
| 41 | 1.2000051475556575 |
| 42 | 1.2000051475558848 |
| 43 | 1.2000051475559985 |
| 44 | 1.2000051475559417 |

4.1.3. Tercera función

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.0 |
| 1 | 1.5 |
| 2 | 1.25 |
| 3 | 1.375 |
| 4 | 1.4375 |
| 5 | 1.46875 |
| 6 | 1.484375 |
| 7 | 1.4921875 |
| 8 | 1.49609375 |
| 9 | 1.498046875 |
| 10 | 1.4990234375 |
| 11 | 1.49951171875 |
| 12 | 1.499755859375 |
| 13 | 1.4998779296875 |
| 14 | 1.49993896484375 |
| 15 | 1.499969482421875 |
| 16 | 1.4999847412109375 |
| 17 | 1.4999923706054688 |
| 18 | 1.4999961853027344 |
| 19 | 1.4999980926513672 |
| 20 | 1.4999990463256836 |
| 21 | 1.4999995231628418 |
| 22 | 1.499999761581421 |

| Iteración | Resultado |
|-----------|--------------------|
| 23 | 1.4999998807907104 |
| 24 | 1.4999999403953552 |
| 25 | 1.4999999701976776 |
| 26 | 1.4999999850988388 |
| 27 | 1.4999999925494194 |
| 28 | 1.4999999962747097 |
| 29 | 1.4999999981373549 |
| 30 | 1.4999999990686774 |
| 31 | 1.4999999995343387 |
| 32 | 1.4999999997671694 |
| 33 | 1.4999999998835847 |
| 34 | 1.4999999999417923 |
| 35 | 1.4999999999708962 |
| 36 | 1.499999999985448 |
| 37 | 1.499999999992724 |
| 38 | 1.499999999996362 |
| 39 | 1.499999999998181 |
| 40 | 1.4999999999990905 |
| 41 | 1.4999999999995453 |
| 42 | 1.4999999999997726 |
| 43 | 1.4999999999998863 |
| 44 | 1.4999999999999432 |

4.2. Newton-Raphson

El método de Newton-Raphson halla la raíz a partir de una semilla inicial, iterando la siguiente sucesión $p_n = p_{n-1} - \frac{f(p_{n-1})}{f'(p_{n-1})}$ hasta que $p_n - p_{n-1} < \epsilon$

Para el cálculo de las derivadas, utilizamos el método 'Derivative' de la biblioteca 'SymPy'. De esa manera, logramos realizar un programa más escalable debido a que no es necesario conocer las derivadas de la función a evaluar.

Los resultados obtenidos fueron los siguientes:

4.2.1. Primera función

En la iteración 3 se llega a la cota $1 \cdot 10^{-5}$

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.0 |
| 1 | 1.5 |
| 2 | 1.4166666666666667 |
| 3 | 1.4142156862745099 |
| 4 | 1.4142135623746899 |
| 5 | 1.4142135623730951 |
| 6 | 1.414213562373095 |

4.2.2. Segunda función

Ambas cotas de error se alcanzan en la última iteración.

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.0 |
| 1 | 1.0670391061452542 |
| 2 | 1.1115008625849883 |
| 3 | 1.141056567216851 |
| 4 | 1.1607272681472196 |
| 5 | 1.173827768369476 |
| 6 | 1.182555936035766 |
| 7 | 1.1883723914181477 |
| 8 | 1.192249031513553 |
| 9 | 1.1948330257346942 |
| 10 | 1.1965554994363112 |
| 11 | 1.1977037321147688 |
| 12 | 1.1984691839006911 |
| 13 | 1.198979468821718 |
| 14 | 1.199319651930617 |

| Iteración | Resultado |
|-----------|--------------------|
| 15 | 1.1995464373959681 |
| 16 | 1.1996976269098787 |
| 17 | 1.1997984196033389 |
| 18 | 1.1998656232066638 |
| 19 | 1.1999104313717341 |
| 20 | 1.1999402942189605 |
| 21 | 1.1999602375275107 |
| 22 | 1.1999735424884255 |
| 23 | 1.199982724803287 |
| 24 | 1.1999885987867616 |
| 25 | 1.1999930940510006 |
| 26 | 1.1999971781007444 |
| 27 | 1.20002163874357 |
| 28 | 1.2000145671108424 |
| 29 | 1.2000118134555946 |
| 30 | 1.200009022124353 |

4.2.3. Tercera función

En la iteración 3 se llega a la cota $1 \cdot 10^{-5}$

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.3 |
| 1 | 1.5941176470588234 |
| 2 | 1.4928216542091288 |
| 3 | 1.5000029603439848 |
| 4 | 1.4999999999999998 |
| 5 | 1.5 |

4.3. Newton-Raphson modificado

El método de Newton-Raphson modificado le aplica el método de Newton-Raphson a la función $\mu = \frac{f(x)}{f'(x)}$, obteniendo la derivada con la función derivar (biblioteca 'SymPy') y halla la raíz a partir de una semilla inicial, iterando la siguiente sucesión $p_{n+1} = p_n - \frac{f(p_n) \cdot f'(p_n)}{f'(p_n)^2 - f(p_n) \cdot f''(p_n)}$ hasta que $p_n - p_{n-1} < \epsilon$

Los resultados obtenidos fueron los siguientes:

4.3.1. Primera función

En la iteración 3 se llega a la cota $1 \cdot 10^{-5}$

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.0 |
| 1 | 1.3333333333333333 |
| 2 | 1.4117647058823528 |
| 3 | 1.41421143847487 |
| 4 | 1.4142135623715002 |
| 5 | 1.414213562373095 |
| 6 | 1.4142135623730951 |

4.3.2. Segunda función

Ambas cotas de error se alcanzan en la última iteración.

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.0 |
| 1 | 1.198429561200951 |
| 2 | 1.1999999596424822 |

4.3.3. Tercera función

En la iteración 4 se llega a la cota $1 \cdot 10^{-5}$

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.3 |
| 1 | 1.403030303030303 |
| 2 | 1.4864315963929944 |
| 3 | 1.4999600913460671 |
| 4 | 1.499999999998983 |
| 5 | 1.5 |

4.4. Secante

El método de la secante halla la raíz a partir de dos semillas iniciales, iterando la siguiente sucesión $p_n = p_{n-1} - \frac{f(p_{n-1}) \cdot (p_{n-1} - p_{n-2})}{f(p_{n-1}) - f(p_{n-2})}$ hasta que $p_n - p_{n-1} < \text{cota de error}$

Los resultados obtenidos fueron los siguientes:

4.4.1. Primera función

En la iteración 4 se llega a la cota $1 * 10^{-5}$

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.0 |
| 1 | 1.3846153846153846 |
| 2 | 1.4193548387096775 |
| 3 | 1.4141592920353983 |
| 4 | 1.414213463902154 |
| 5 | 1.4142135623749845 |
| 6 | 1.4142135623730951 |
| 7 | 1.414213562373095 |

4.4.2. Segunda función

En la iteración 35 se llega a la cota $1 * 10^{-5}$

| Iteración | Resultado | Iteración | Resultado |
|-----------|--------------------|-----------|--------------------|
| 0 | 1.0 | 20 | 1.1993404372360148 |
| 1 | 1.0655021834061167 | 21 | 1.1995021126720737 |
| 2 | 1.0943398952179324 | 22 | 1.1996241562355223 |
| 3 | 1.1215850325280734 | 23 | 1.199716287429286 |
| 4 | 1.1404772887106263 | 24 | 1.1997858306793374 |
| 5 | 1.1551937245338697 | 25 | 1.1998383355364535 |
| 6 | 1.1661565358156876 | 26 | 1.1998779580183785 |
| 7 | 1.1744671484356637 | 27 | 1.1999079106133839 |
| 8 | 1.1807264667044532 | 28 | 1.1999304889638724 |
| 9 | 1.1854534082642374 | 29 | 1.1999475560634543 |
| 10 | 1.1890198994831873 | 30 | 1.1999603693570613 |
| 11 | 1.19171200301767 | 31 | 1.1999701623743182 |
| 12 | 1.1937438805214264 | 32 | 1.1999775472725775 |
| 13 | 1.195277583160919 | 33 | 1.1999837505871154 |
| 14 | 1.1964352549762844 | 34 | 1.1999875680114462 |
| 15 | 1.197309112780888 | 35 | 1.199991385435777 |
| 16 | 1.197968742660849 | 36 | 1.2000028377087697 |
| 17 | 1.1984666682043323 | 37 | 1.200025742254755 |
| 18 | 1.1988425329007377 | 38 | 1.200004315421414 |
| 19 | 1.1991262601453252 | 39 | 1.200004315421414 |

4.4.3. Tercera función

En la iteración 6 se llega a la cota $1 * 10^{-5}$

| Iteración | Resultado |
|-----------|--------------------|
| 0 | 1.0 |
| 1 | 1.394130271809281 |
| 2 | 1.8764930021801223 |
| 3 | 1.549246841462836 |
| 4 | 1.4523931358165483 |
| 5 | 1.5000153797084042 |
| 6 | 1.4999998599846303 |
| 7 | 1.5000000000000002 |
| 8 | 1.5 |

5. Comparación de resultados

5.0.1. Primera función

Usando la biblioteca 'SciPy', las raíces halladas fueron:

* Usando bisección: $1.4142135623715149 \pm 0.0000000000000001$

* Usando Newton Raphson: $1.414213562373095 \pm 0.0000000000000001$

Los resultados obtenidos con las funciones programadas son congruentes con estos últimos, viendo que la convergencia más rápida se encontró con los métodos de Newton-Raphson, luego el método de la secante y por último el método de bisección.

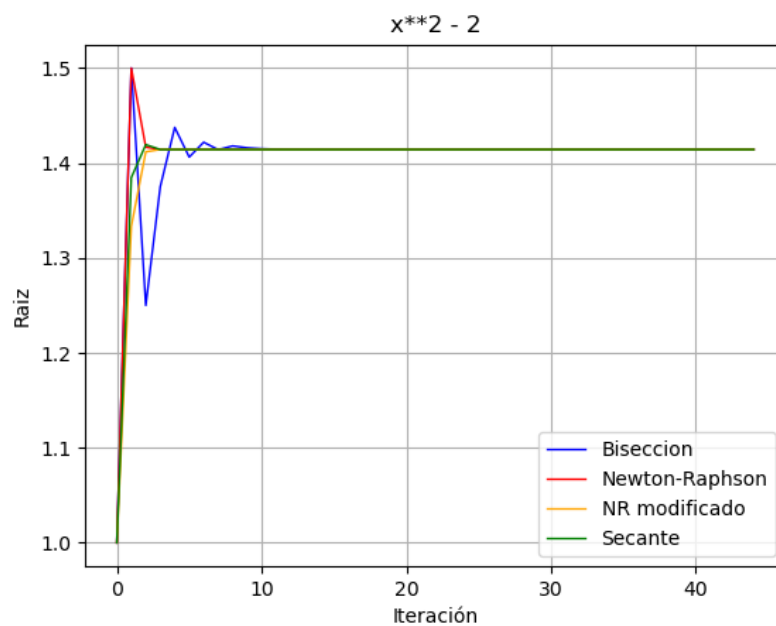
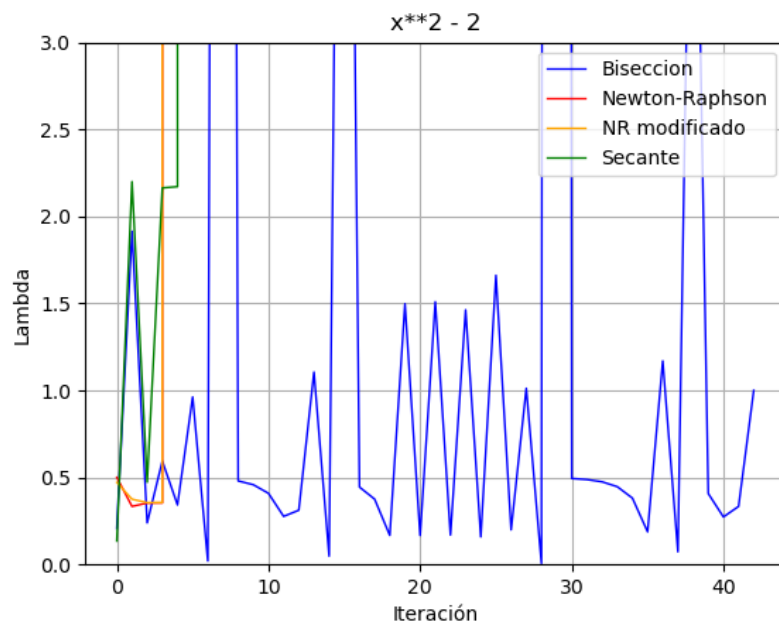
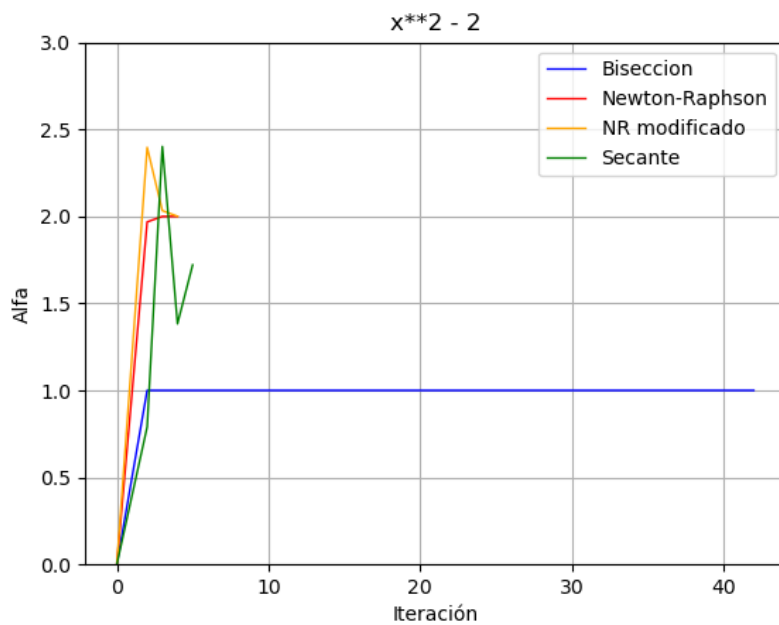


Figura 4: Valor de la raíz según iteración.

Figura 5: Valor de λ según iteración.Figura 6: Valor de P según iteración.

Podemos ver como los valores de P están de acuerdo con lo que se esperaba en base a la teoría, mostrando una convergencia lineal para la bisección, una cuadrática para los métodos Newton-Raphson y Newton-Raphson modificado y una convergencia supra-lineal para el método de la secante.

5.0.2. Segunda función

Usando la biblioteca 'SciPy', las raíces halladas fueron:

* Usando bisección: $1.2000045776367188 \pm 0.0000000000000001$

* Usando Newton Raphson: $1.2000066214794778 \pm 0.0000000000000001$

Los resultados obtenidos con las funciones programadas son congruentes con estos últimos para la tolerancia $1 * 10^{-5}$, no así para la cota de error $1 * 10^{-13}$, se estima que esto se debe a como está programado el método de la biblioteca. Para esta función, claramente el método de Newton-Raphson modificado fue el que mostró una convergencia más acelerada.

Analizando la diferencia en iteraciones para las dos cotas de error, se observa que prácticamente necesitaron la misma cantidad.

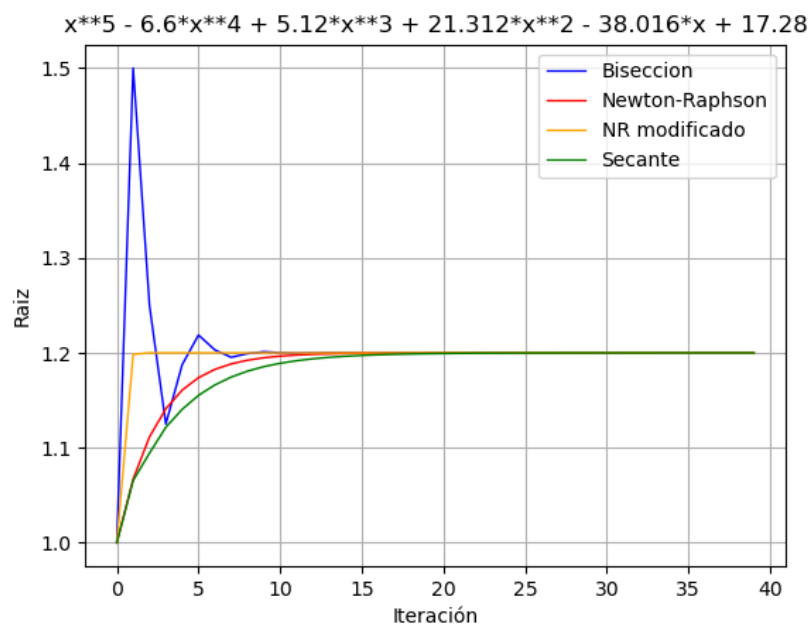


Figura 7: Valor de la raíz según iteración.

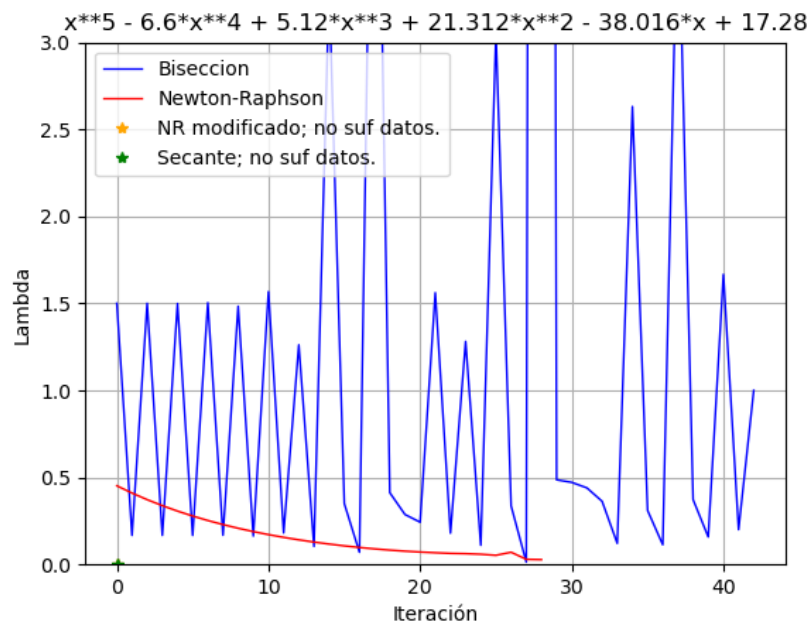
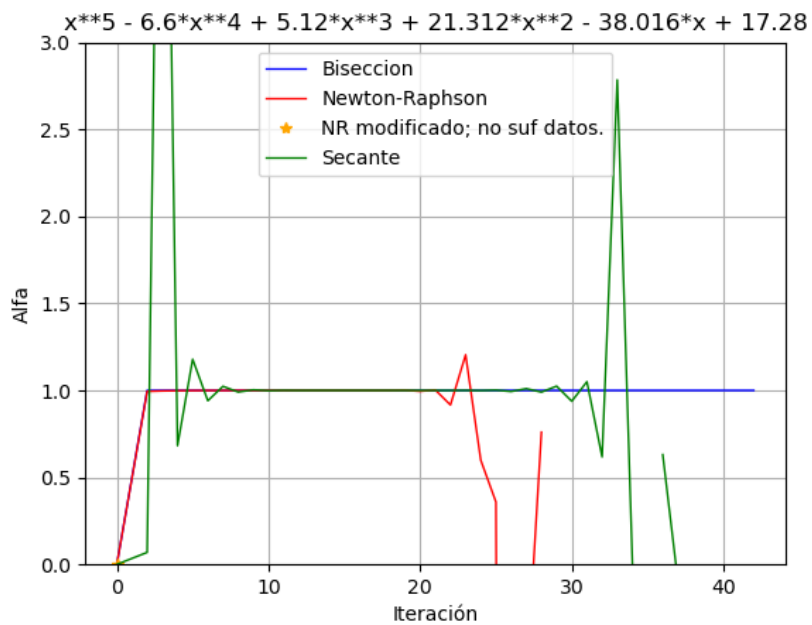
Figura 8: Valor de λ según iteración.

Figura 9: Valor de P según iteración.

5.0.3. Tercera función

Usando la biblioteca 'SciPy', las raíces halladas fueron:

* Usando bisección: $1.5000000000000000 \pm 0.0000000000000001$

* Usando Newton Raphson: $1.5000000000000000 \pm 0.0000000000000001$

Los resultados obtenidos con las funciones programadas son congruentes con estos últimos, viendo que la convergencia más rápida se encontró con los métodos de Newton-Raphson, luego el método de la secante y por último el método de bisección.

Algo que notamos en esta función para el método de bisección es que llega al valor correcto de la raíz en la segunda iteración, pero como el criterio de corte es la diferencia entre iteraciones, no sería correcto el corte en esa segunda iteración.

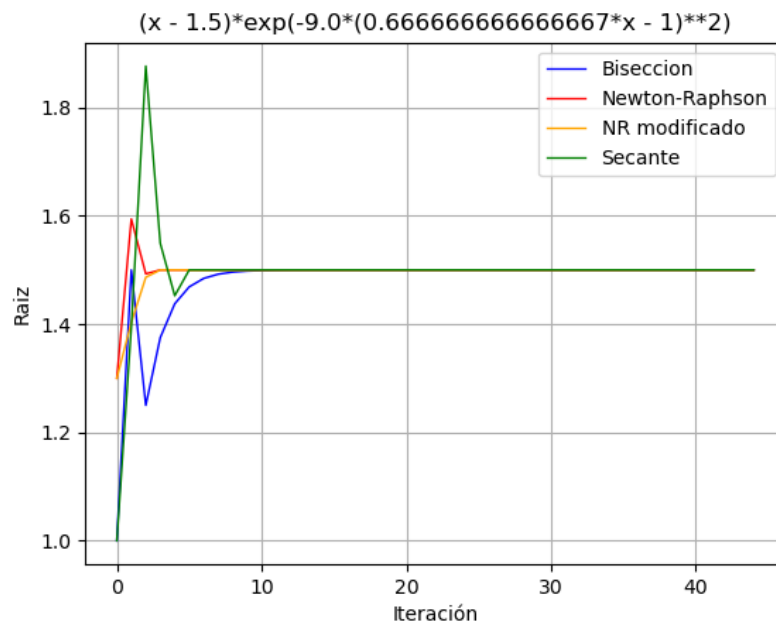
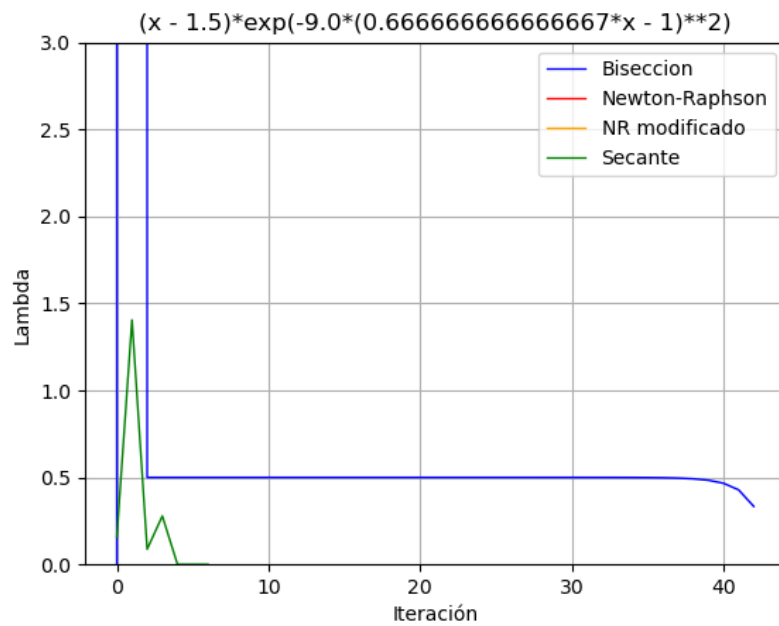
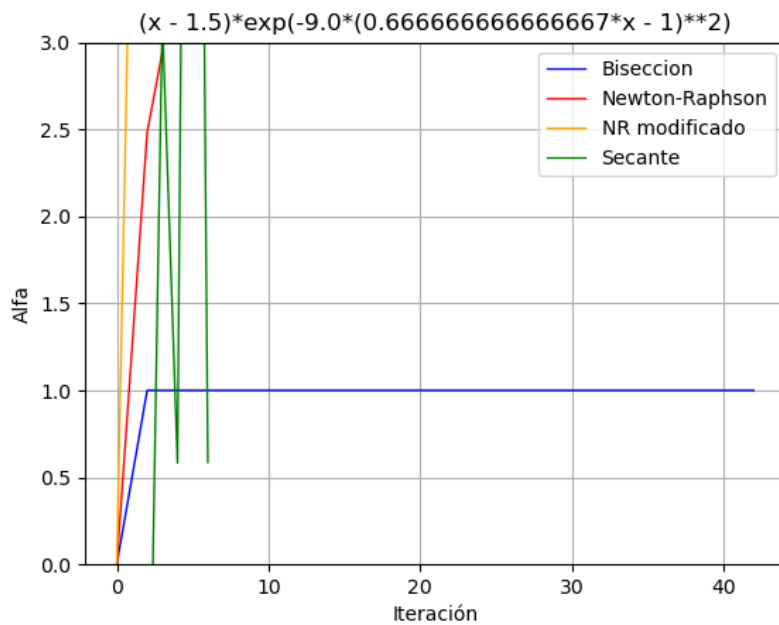


Figura 10: Valor de la raíz según iteración.

Figura 11: Valor de λ según iteración.Figura 12: Valor de P según iteración.

6. Conclusiones

A modo de conclusión de este trabajo práctico, se destaca que la convergencia al resultado con la tolerancia dada con menos iteraciones fue utilizando el método de Newton-Raphson modificado, de acuerdo a la teoría.

Algo interesante que observamos fue que el método de la bisección converge siempre en la misma cantidad de iteraciones para una cota de error dada, comprobando en la práctica uno de los teoremas del método.

Además, se encontraron dificultades para hallar el orden de convergencia P y la constante asintótica λ . Estimamos que una posible fuente de este inconveniente es el error de redondeo proveniente del uso de notación de punto flotante. Además, al trabajar con números muy cercanos entre si, el error tiende a aumentar al dividir por un número muy pequeño o multiplicando por uno muy grande, siendo estos cálculos necesarios para encontrar los valores que buscamos. Este error se podría llegar a reducir anidando las funciones, lo que provocaría una reducción en la cantidad de operaciones a realizar en algunos casos. Otra fuente de error es que obtenemos el orden de convergencia vía una estimación y no vía un cálculo directo, es decir, le agregamos una incerteza más a un resultado con ya mucho error

Sobre la función 3, con los métodos NR y NR modificado, elegimos modificar la semilla inicial a 1,3, ya que observamos que con la semilla dada era imposible la convergencia (mediante iteraciones encontramos que el intervalo de convergencia es aproximadamente $1,26 < \text{semilla} < 1,74$). Nos pareció la decisión más adecuada elegir una semilla arbitrariamente dentro del intervalo de convergencia hallado para de esa manera mostrar el funcionamiento completo de los métodos.