Universidad Nacional Autónoma de México

Escuela Nacional de Estudios Superiores

Unidad Juriquilla

Computación II

Practica 3

Comparación de métodos numéricos e ingenuos

Bruno Arturo López Pacheco

Dr. Ulises Olivares Pinto

No. De Cuenta 317347140

Licenciatura en Tecnología

Objetivo

Implementar los métodos de exploración exhaustiva, de la bisección, de Newton-Raphson, de la secante y el de Brent-Decker para realizar una comparación de tiempos de ejecución.

Marco teórico

Método de la bisección

Este método se usa para resolver ecuaciones y se basa en el Teorema de los Valores Intermedios, el cual dice que toda función continua f, en un intervalo cerrado *[a,b],* toma todos los valores que se hallan entre *f(a)* y *f(b),* de tal forma que la ecuación $f(x) = 0$ tiene una sola raíz que verifica si $f(a) \cdot f(b) < 0$

Consiste en partir de un intervalo $[x_0, x_1]$ tal que $f(a) \cdot f(b) < 0$, por lo que se sabe que existe al menos una raíz real. A partir de ese punto se va reduciendo el intervalo sucesivamente hasta hacerlo tan pequeño como se diga la precisión que se haya decidido usar.

El método consiste en:

- Debe existir seguridad sobre la continuidad de la función $f(x)$ en el intervalo $[a, b]$.
- Se verifica que $f(a) \cdot f(b) < 0$
- Se calcula el punto medio m del intervalo $[a, b]$ y se evalúa $f(m)$; si ese valor es igual a cero, es la raíz buscada.
- En caso de que no lo sea, se verifica si $f(m)$ tiene signo opuesto con $f(a)$ o con $f(b)$.
- Se define un nuevo intervalo $[a, b]$ siendo ahora $[a, m]$ o $[m, b]$ según se haya determinado en cuál de estos intervalos ocurre un cambio de signo,
- Con este nuevo intervalo se continúa sucesivamente encerrando la solución en un intervalo cada vez más pequeño hasta alcanzar la precisión deseada.

Este método es muy simple y al mismo tiempo muy robusto.

Método de Newton-Raphson

Es un procedimiento basado en la derivada para encontrar raíces de una función real de variable real que sea derivable y es muy útil para aproximar raíces de polinomios en los cuales los métodos conocidos no funcionan.

El método consiste en:

- Sea $f: [a, b] \rightarrow \mathrm{R}$ una función derivable, de la cual se sabe que tiene una raíz en el intervalo
- Se elige x_0 en el eje de las x, asumiendo que está cerca de la solución de $f(x) = 0$
- Se calcula la ecuación punto pendiente de la recta tangente en función de $\left(x_0, f(x_0)\right)$ a saber:

$$y - f(x_0) = f^{'}(x_0)(x - x_0)$$

- Esta recta debe intersecar al eje de las x en un punto x_1, más cercano a la raíz buscada.
- Así, el punto $\left(x_{1,} 0\right)$ satisface la ecuación anterior y sustituyendo queda:

$$0 - f(x_0) = f^{'}(x_0)(x_1 - x_0)$$

- Si $f^{'}(x_0) \neq 0$, entonces, despejando x_1 en la segunda ecuación, queda que: $x_1 = x_0 - \frac{f(x_0)}{f^{'}(x_0)}$.
- Se repite el mismo razonamiento para x_0, pero ahora comenzando en x_1, en ese caso se tiene $x_2 = x_0 - \frac{f(x_1)}{f^{'}(x_1)}$, que está más cerca de la raíz buscada que x_1
- Iterando cada vez con el número obtenido se construye una secuencia sobre: $x_0, x_1, \dots, x_n$ de números cada vez más próximos a la raíz, tales que:

$$x_{n+1} = x_n - \frac{f(x_n)}{f^{'}(x_n)}$$

La aproximación será mejor entre más términos se quieran calcular.

Método de la Secante

El método anterior (de Newton) requiere conocer el valor de la primera derivada de la función en un punto, por lo que, en algunos casos, resulta complicado. Para estos casos es más útil usar el método de la secante.

El método de la secante parte de dos puntos y estima la tangente por una aproximación de acuerdo con la expresión:

$$f^{'}(x_{\mathrm{n}}) = \frac{f(x_{\mathrm{n}}) - f(x_{\mathrm{n-1}})}{x_{\mathrm{n}} - x_{\mathrm{n-1}}}$$

Sustituyendo está nueva expresión en la ecuación del método de Newton se puede obtener la expresión del método de la secante:

$$x_{n+1} = x_n - \frac{f(x_n)(x_n - x_{n-1})}{f(x_n) - f(x_{n-1})}$$

Método de Brent-Dekker

El método de Dekker es un método que combina los métodos de la bisección y la secante para encontrar raíces de polinomios. Posteriormente, Brent retoma el trabajo de Dekker y propone algunas mejoras al método incorporando una interpolación cuadrática inversa. El algoritmo de la secante domina al inicio sobre el método de la bisección. Pero después que se aproxima la raíz, el método de la bisección domina.

La interpolación cuadrática inversa es similar al método de la secante, se basa en el cálculo de una línea recta que pasa por dos puntos iniciales. La intersección de esta línea en el eje de las x representa la nueva aproximación a la raíz.

Dados 3 puntos, es posible determinar una función cuadrática de x que pase por estos 3 puntos. Al igual que en el método de la secante, la intersección de la interpolación (función cuadrática) con el eje de las x es la nueva aproximación a la raíz.

Este método representa una mejora con respecto al método de la secante, no obstante, es posible que pueda fallar en el caso donde no existan raíces reales en una función.

El método consiste en:

- Se seleccionan dos puntos iniciales a y b (como en la bisección).

- Tal que la evaluación de la función f(a) y f(b) tengan signos opuestos. Se debe cumplir que $f(a)f(b) < 1$
- Si la función es continua en el intervalo [a, b] entonces, el teorema del valor intermedio garantiza que existe algún valor en el intervalo tal que la evaluación de la función es cero ($f(x) = 0$).
- Dados 3 puntos: $f(a)$, $f(b)$ y $f(c)$, el método de Brent primero ajusta a x como una función cuadrática de y.
- Se una el siguiente método de interpolación:

$$x = \frac{(y-f_b)(y-f_c)}{(f_a-f_b)(f_a-f_c)} a + \frac{(y-f_a)(y-f_c)}{(f_b-f_a)(f_b-f_c)} b + \frac{(y-f_a)(y-f_b)}{(f_c-f_a)(f_c-f_b)} \text{c}$$

- Para encontrar aproximaciones sucesivas a la raíz se establece a $y = 0$.
- Entonces se define la interpolación como: $x = b + \frac{P}{Q}$; donde

$$R = \frac{f(b)}{f(c)}, S = \frac{f(b)}{f(a)}, T = \frac{f(a)}{f(c)}$$

$$P = S[T(R-T)(c-b) - (1-R)(b-a)]$$

$$Q = (T-1)(R-1)(S-1)$$

- El punto en el que $y = 0$, es el nuevo b y el antiguo b se convierte a, pero si c y el nuevo b no tienen un cambio de signo, entonces se usa a en su lugar.

Metodología

Pseudocódigo del método de la bisección:

```
    Si (f(a)*f(b) < 0):
      Existe un cambio de signo
    Sino:
      No existen raíces reales en el intervalo
      Salir

    error = inf
    iteración=1

    mientras (error>=tolerancia o iteración==máximaIteración):
      c=(a+b) /2
      Si (f(a)*f(c) < 0):
        Hay una raíz en [a, c]
        a=a
        b=c
      Si (f(a)*f(c) > 0):
        Hay una raíz en [c, b]
        a=c
        b=b
      Si (f(a)*f(c) == 0):
        La raíz real se encuentra en: c
        break
      error=abs(b-a)
      iteración=iteración+1
```

Pseudocódigo del método de Newton-Raphson:

```
mientras (errorAbsoluto>=tolerancia o iteración==IteraciónMáxima):
      x1=x2
      Si (f prima(x1) ==0):
        "Error matemático"
        break
      x2=x1-(f(x1) /f prima(x1))
      error Absoluto=|x2-x1|
      iteración+=1

    Si (iteración>Iteración Máxima):
      "No converge a ninguna solución"
    Si No:
      "Solución:"x2
```

Pseudocódigo del método de la secante:

```
mientras (errorAbsoluto>=tolerancia o iteración==IteraciónMáxima):
      x0=x1
      x1=x2
      x2=x1-((f(x1) *(x1-x0))/(f(x1)-f(x0)))
      error Absoluto=|x2-x1|
      iteración+=1
    Si (iteración>Iteración Máxima):
      "No converge a ninguna solución"
    Si No:
      "Solución:”, x2
```

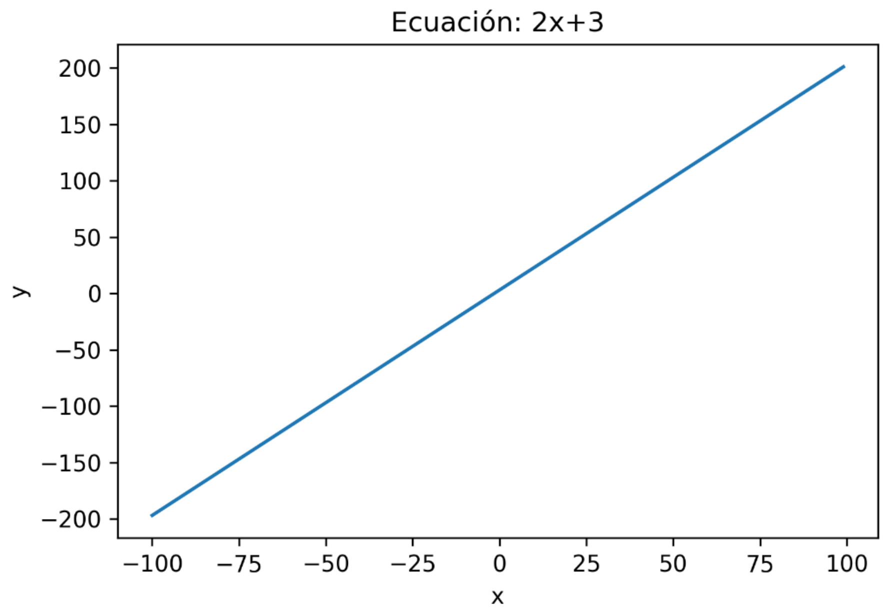
Pseudocódigo del método de Brent-Dekker:

```
Si  (f(a)*f(b)>=0):
      "No existe raíz en el rango proporcionado"
      regresar 0
    Si  (|f(a)|<|f(b)) |:
      a, b=b,a

    c=a
    mflag=True
    iteración=1
    error=np.inf

    mientras (error>=tolerancia o iteración==Iteración Máxima):
      Si  (f(a)! =f(c) and f(b)! =f(c)):
        sa=a*f(b)*f(c)*(1/(f(a)-f(b))) *(1/(f(a)-f(c)))
        sb=b*f(a)*f(c)*(1/(f(b)-f(a))) *(1/(f(b)-f(c)))
        sc=c*f(a)*f(b)*(1/(f(c)-f(a))) *(1/(f(c)-f(b)))
        s=sa + sb + sc
      Si No:
        s=b-f(b)*(b-a) *(1/(f(b)-f(a)))

      aux=(3*a+b)/4
      delta=0.00001
      Si  (s>aux and s<b) or (mflag==True and abs(s-b)>=(abs(b-
c)/2)) or (mflag==False and abs(s-b)>=(abs(c-
d)/2)) or (mflag==True and abs(b-
c)<abs(delta)) or (mflag==False and abs(c-d)<abs(delta)):
        s=(a+b)/2
        mflag=True
      Si No:
        mflag=False

      d=c
      c=b
      Si  (f(a)*f(s)<0):
        b=s
      Si No:
        a=s

      if (|f(a)|<|f(b)|):
        a, b=b, a
      iteración+=1
      error=|b-a|
    "Solución:", b
```

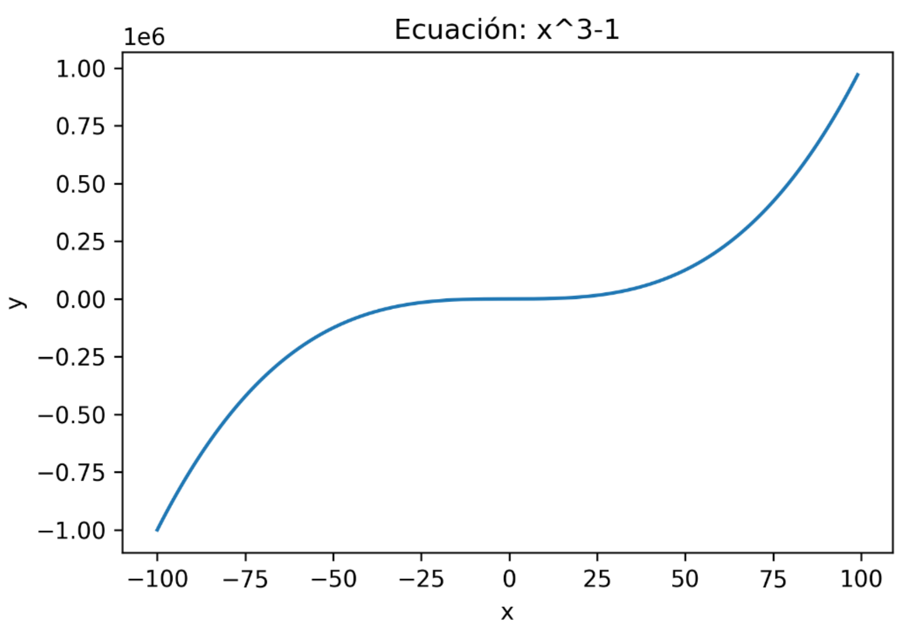
Pruebas para comparación

Al momento de comparar los tiempos entre métodos, se analizaron las respuestas de 5 ecuaciones específicas, las cuales son:

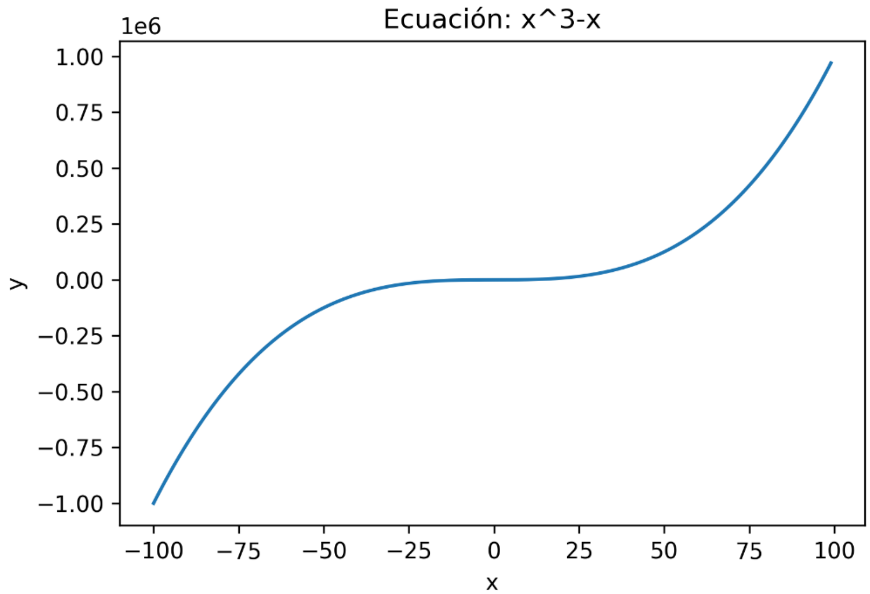
$f_{1(x)} = 2x + 3$ con raíz real de -1.5



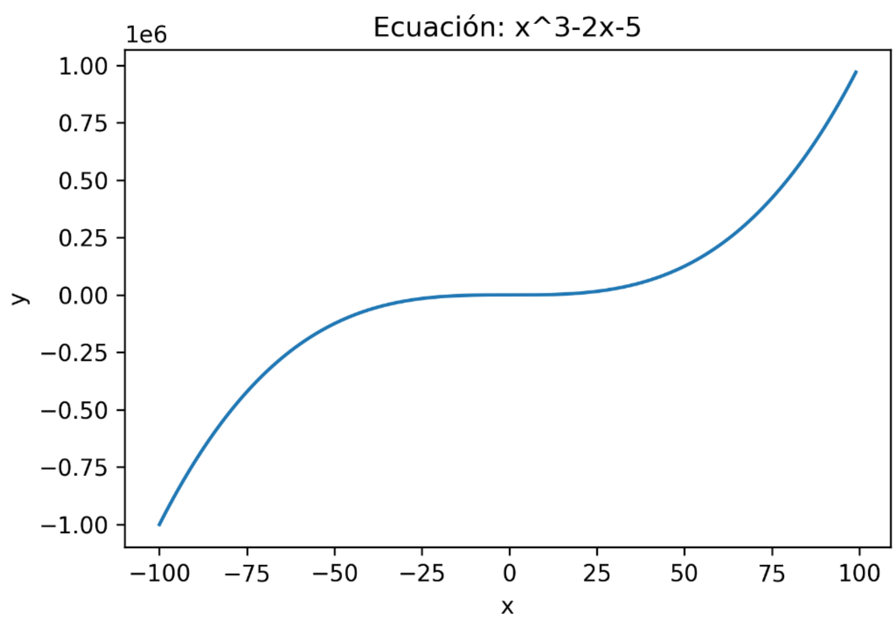
$f_2(x) = x^3 - 1$ con raíz real de 1



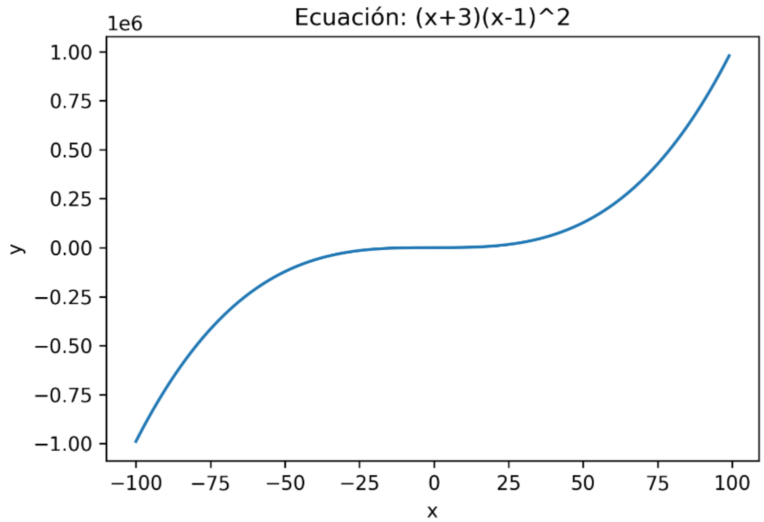
$f_3(x) = x^3 - x$ con 3 raíces reales; -1,0 y 1



$f_4(x) = x^3 - 2x - 5$ con una raíz real de aproximadamente 2.0945



$f_5(x) = (x+3)(x-1)^2$ con raíces reales de -3 y 1



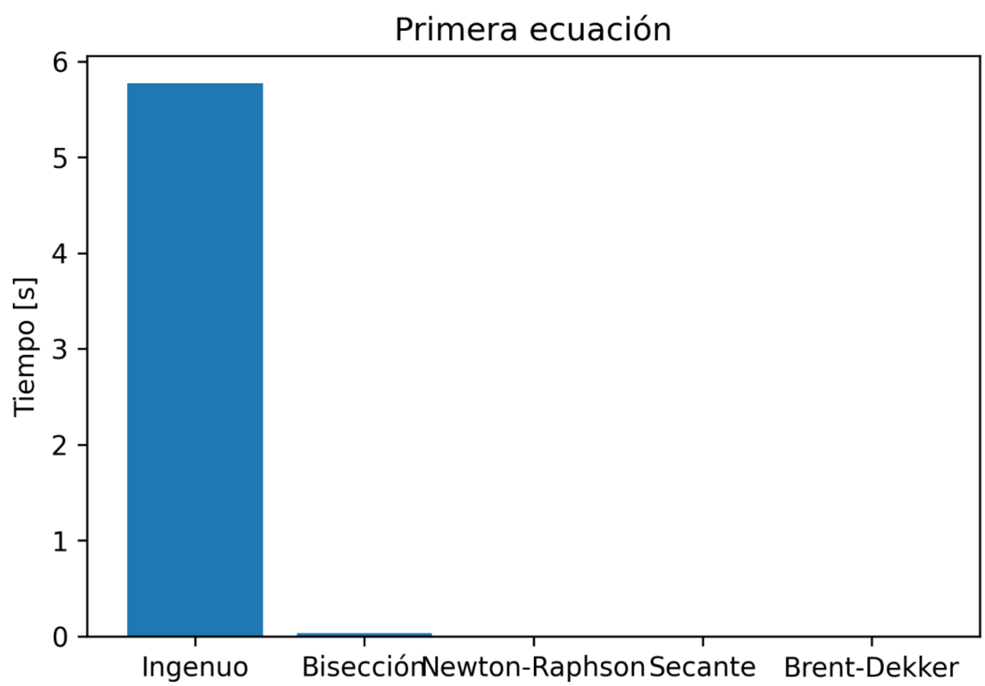
Durante este proceso de pruebas, se realizaron pruebas de tiempo haciendo uso de la librería time para mostrar el tiempo de ejecución.

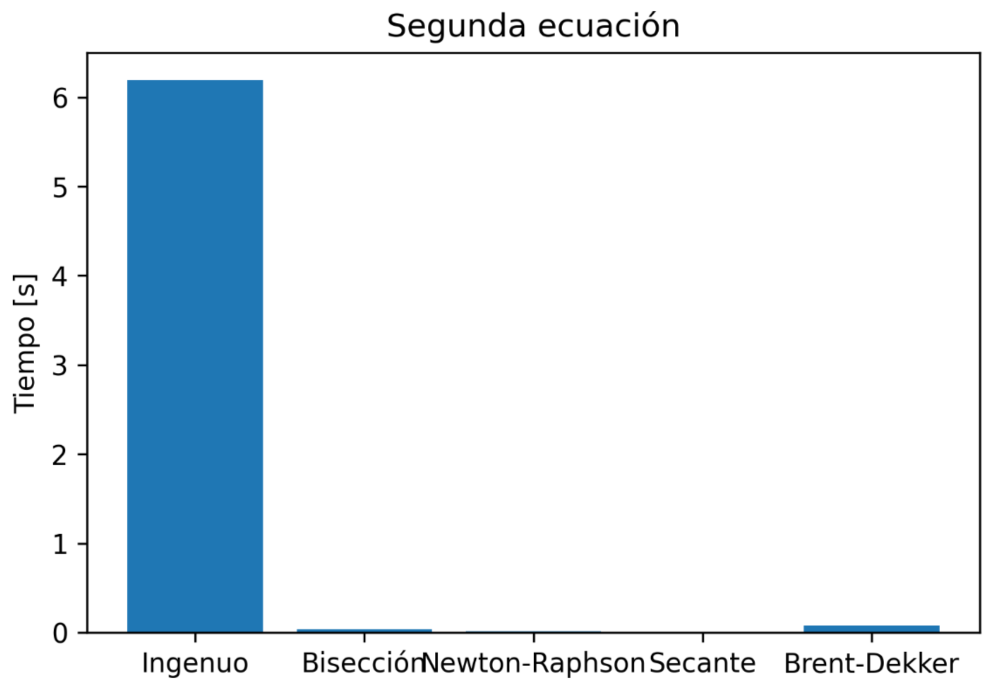
Resultados

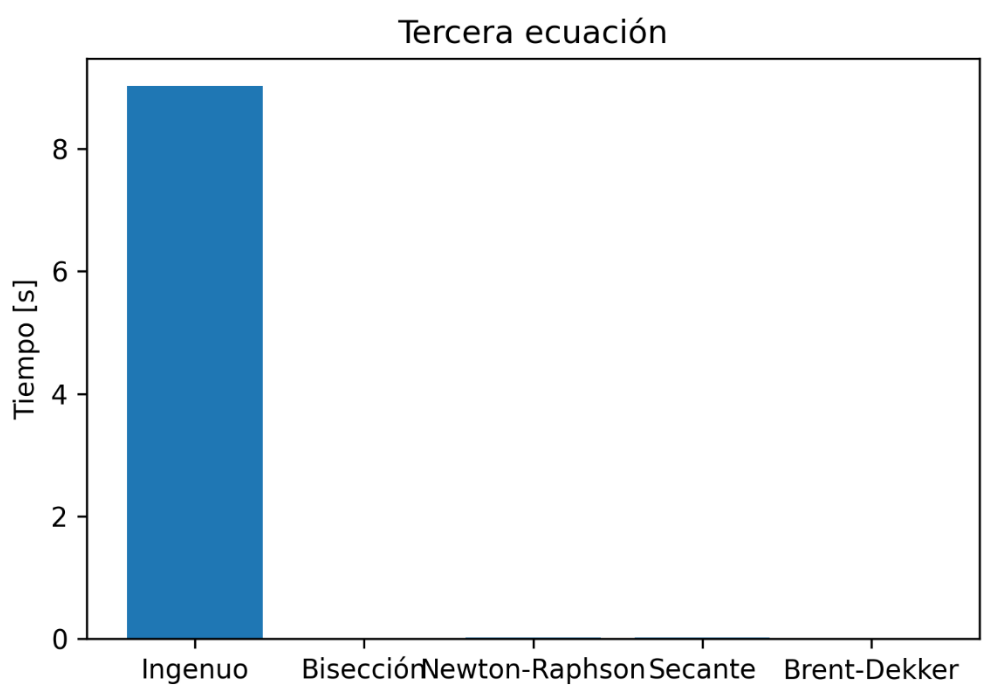
Existen varios puntos a tratar en los diferentes casos de las ecuaciones, como es el caso de el número de raíces que es capaz de calcular cada método, y que, debido a la implementación de los métodos, los métodos numéricos solo son capaces de calcular una raíz, pero el método exhaustivo es capaz de calcular todas las raíces reales.

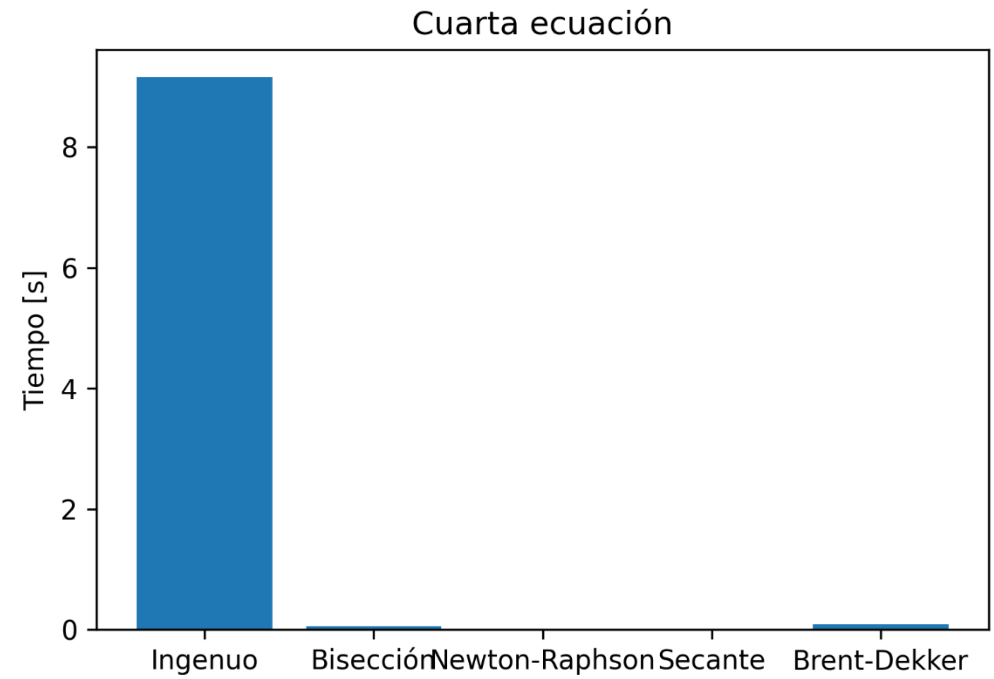
En el caso de la prueba 4, esta ecuación tiene una raíz que es un número irracional, por lo que, debido a la implementación del método exhaustivo, nunca se llega a encontrar un valor exacto de x en donde $f(x) = 0$.

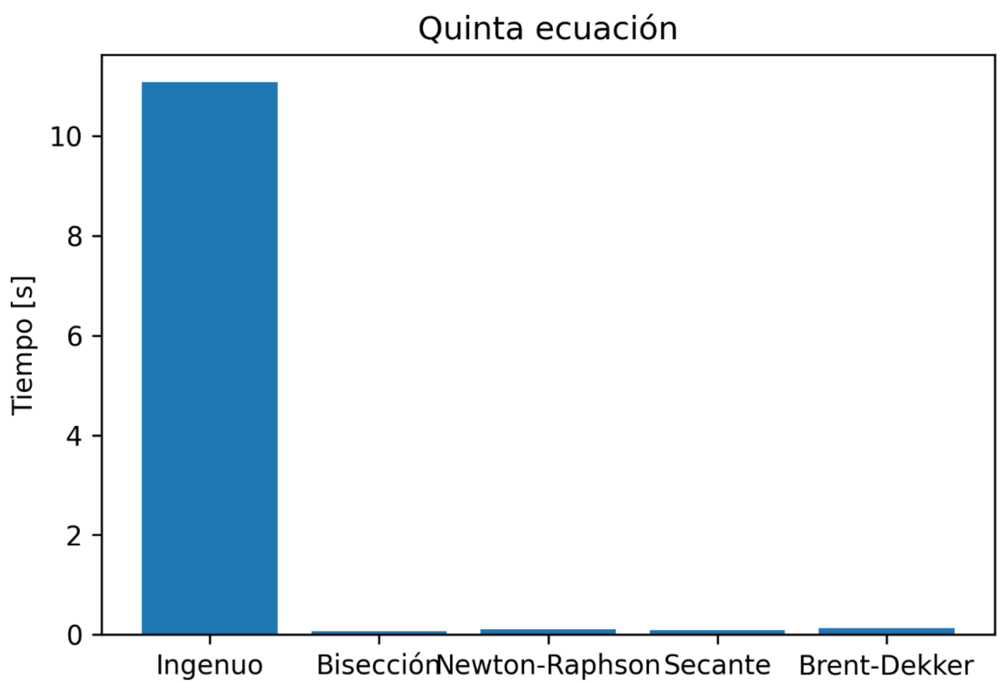
A continuación, se muestran las gráficas comparando los tiempos de ejecución en los 5 casos de prueba.



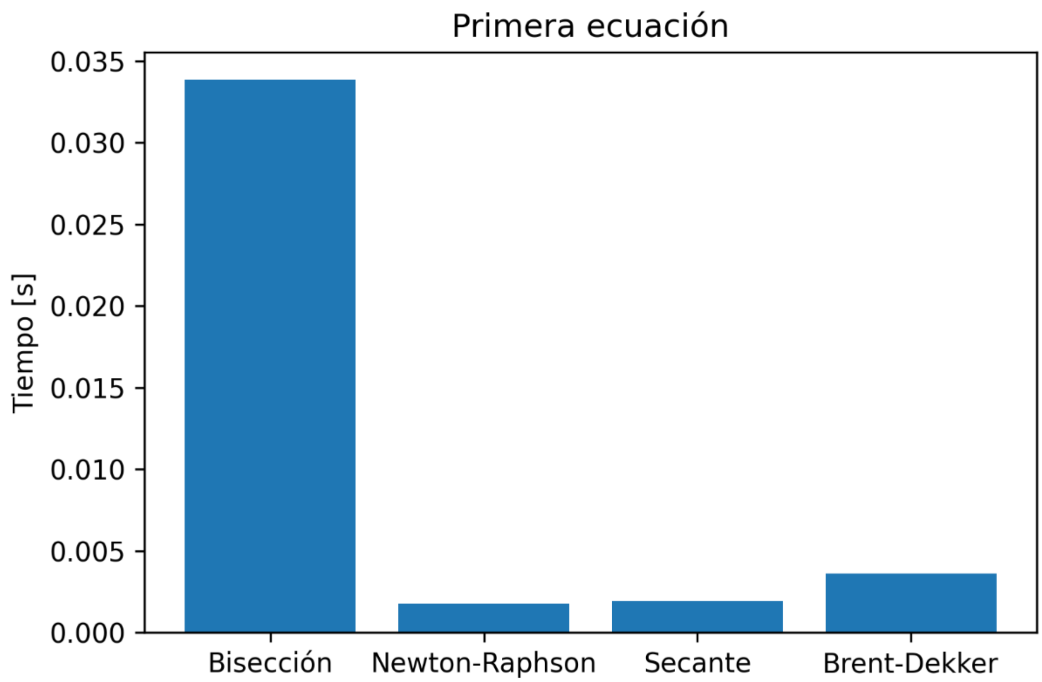

Segunda ecuación
Tiempo [s]
0
1
2
3
4
5
6
Ingenuo
Bisección
Newton-Raphson
Secante
Brent-Dekker

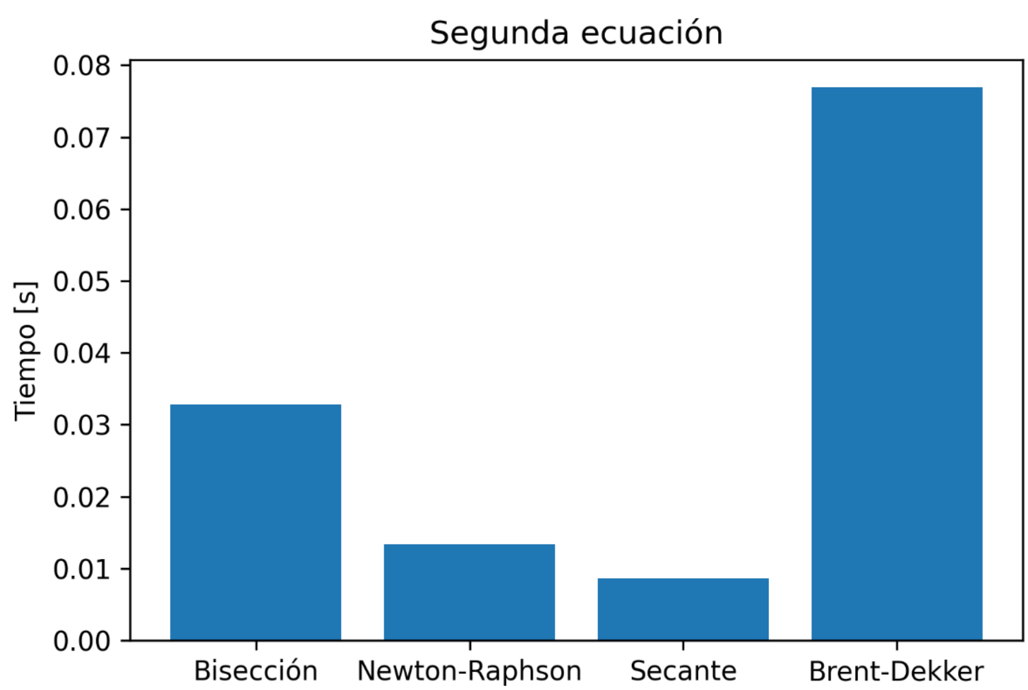

Tercera ecuación
Tiempo [s]
0
2
4
6
8
Ingenuo
Bisección
Newton-Raphson
Secante
Brent-Dekker

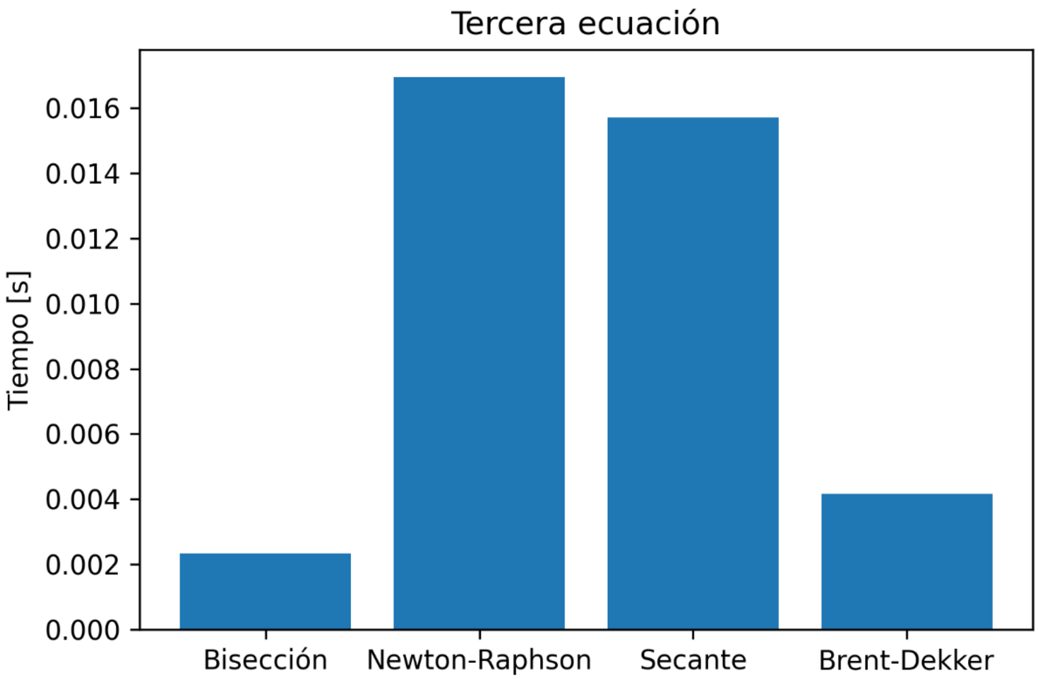

Cuarta ecuación
Tiempo [s]
0
2
4
6
8
Ingenuo
Bisección
Newton-Raphson
Secante
Brent-Dekker

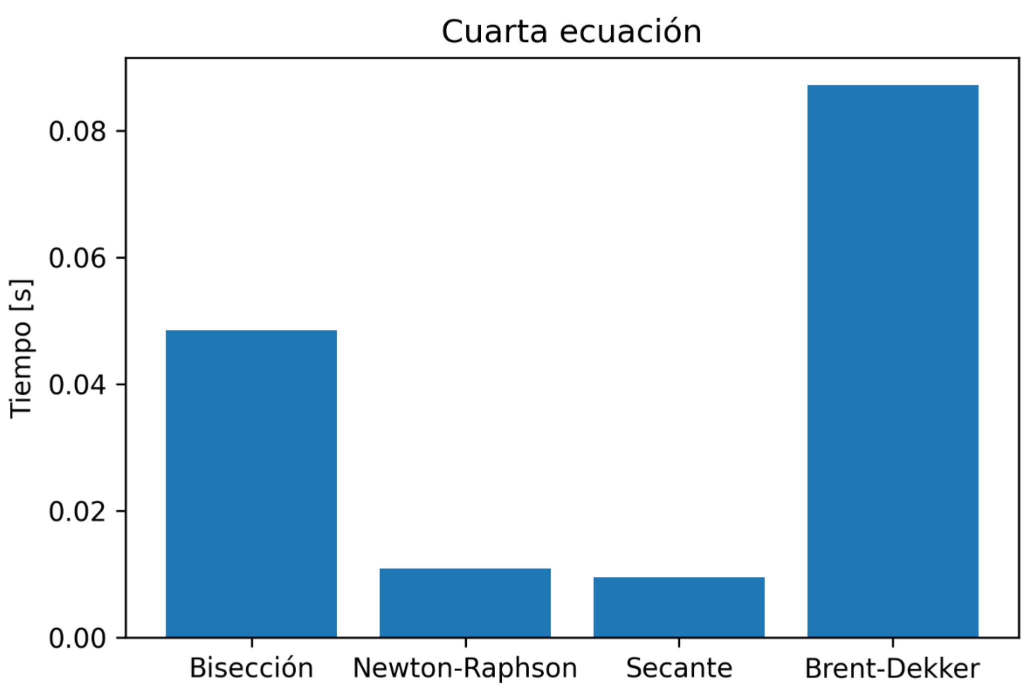

Quinta ecuación
Tiempo [s]
0
2
4
6
8
10
Ingenuo
Bisección
Newton-Raphson
Secante
Brent-Dekker

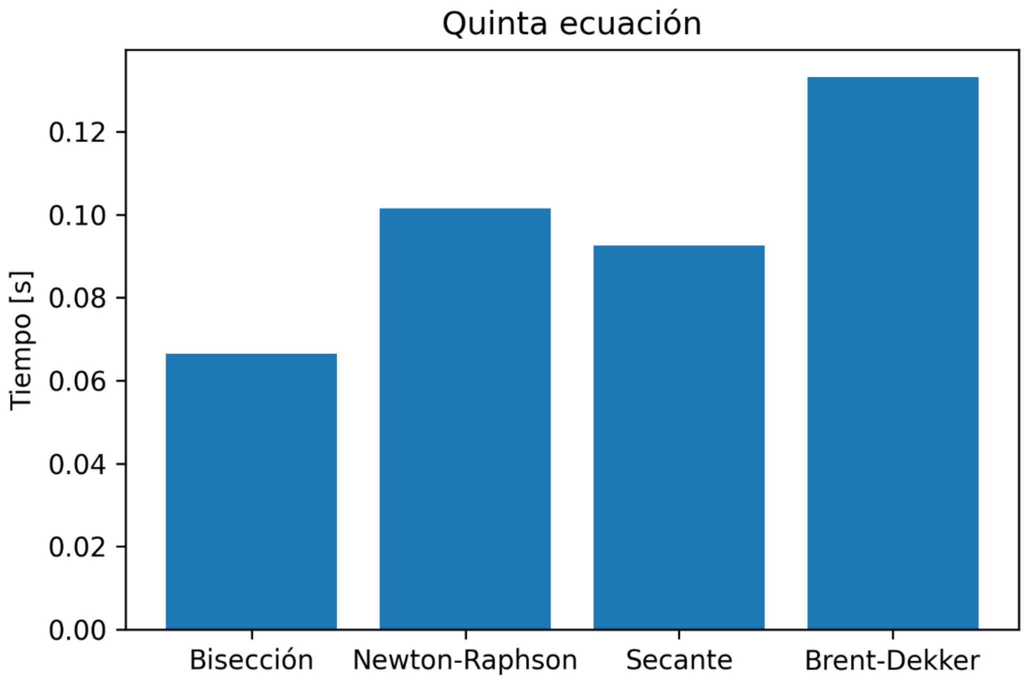
Es notable la gran diferencia entre los métodos numéricos y el método de búsqueda exhaustiva, por lo que ahora se muestran las diferencias de tiempo entre los métodos numéricos.






Tercera ecuación
Tiempo [s]
0.000
0.002
0.004
0.006
0.008
0.010
0.012
0.014
0.016
Bisección
Newton-Raphson
Secante
Brent-Dekker


Cuarta ecuación
Tiempo [s]
0.00
0.02
0.04
0.06
0.08
Bisección
Newton-Raphson
Secante
Brent-Dekker



Conclusión

Los métodos numéricos son de gran ayuda para conocer con cierta exactitud las raíces de un polinomio y son mucho mejores en rendimiento. Como se vio en las diferentes pruebas que se hicieron, hubo casos en los que solo los métodos numéricos pudieron obtener una solución para los polinomios.

Es incluso más visual ver el proceso de los métodos numéricos en programación ya que se pueden ver las iteraciones y como la aproximación va alcanzando la solución real. Sería una buena idea ir graficando la solución o la gráfica de la secante para el caso del método de Newton y el método de la secante, y así mostrar la aproximación de manera más didáctica.

Estos métodos numéricos tienen sus limitaciones, dependiendo del algoritmo en sí, o de la implementación que se les de al programar, pero cuando se tienen las condiciones necesarias, convergen a las soluciones de una manera muy rápida. E incluso, no sería necesario mucho poder computacional para poder hallar las raíces de un polinomio con estos métodos.

Referencias

Chapra, S., & Canale, R. (2011). METODOS NUMERICOS PARA INGENIEROS (Spanish Edition) (6th ed.). McGraw-Hill Interamericana de España S.L.