Métodos de búsqueda de raíces

Jhon James Angarita Cod. 2188139

Index Terms—Búsqueda de raíces, algoritmos, tolerancia, número de iteraciones.

1. Introduction

Si una ecuación algebráica o trascendente es relativamente complicada, no resulta posible por lo general hallar raíces exactas. Además, en algunos casos las ecuaciones tienen coeficientes conocidos sólo de forma aproximada, y por tanto, carece de sentido tratar de hallar las raíces exactas de la ecuación. Por consiguiente, adquieren particular importancia los procedimientos de cálculo aproximado de raíces de una ecuación así como la estimación de su grado de exactitud.
El problema consiste en encontrar los valores de la variable x que satisfacen la ecuación $f(x) = 0$, para una función f dada, que está definida y es continua en un cierto intervalo infinito o finito $a < x < b$. En ciertos casos se necesitará la existencia y continuidad de la primera derivada $f'(x)$.
Este trabajo presenta los algoritmos más comunes de búsquedas de raíces, haciendo analisis del costo computacional y número de iteraciones requeridas para llegar a una tolerancia estimada. Los algoritmos que se abordan en este trabajo son: Bisección, Newton-Rapshon, Secante y Interpolación Cradrática Inversa.

2. Métodos de Búsqueda de raíces

En esta sección analizaremos uno de los problemas básicos del anàlisis numárico: el problema de búsqueda de raíces. Se presentan los algoritmos mas cotidianos de busqueda de raíces con el fin de determinar el desempeño de cada uno de ellos.

2.1. Método de Biseccción

El método de bisección es un algoritmo de búsqueda de raíces que trabaja dividiendo el intervalo a la mitad y seleccionando el subintervalo que tiene la raíz. Es también conocido como Meéodo de Intervalo Medio dado a que se se basa en el teorema del valor intermedio. El proceso de separación de raíces comienza estableciendo los signos de la función $f(x)$ en los puntos extremos $x = a$ y $x = b$ de sus dominios de existencia. A continuación se determinan los signos de la función $f(x)$ para un número intermedio de puntos $x = \alpha_1, \alpha_2, \cdots, \alpha_k$ cuya elección depende de la peculiaridades de la función $f(x)$. Si se cumple que $f(\alpha_k) \cdot f(\alpha_{k+1}) < 0$, entonces, existe una raíz de la ecuación $f(x) = 0$ en el intervalo (α_k, α_{k+1}).
En el algoritmo 1 resume el método de Bisección. Note que como parametros de entrada tiene la funcion $f(x)$, el intervalo de busqueda $[a, b]$ y la tolerancia al error ϵ.

Algorithm 1: Método de Bisección

Input : función $f(x)$, intervalo $[a, b]$, tolerancia ϵ
Output: x_k
while $|x_k - \beta| > \epsilon$ **do**
  $m = a + \frac{b-a}{2}$;
  if $signo(f(a)) = signo(f(m))$ **then**
    $a = m$;
  else
    $b = m$;
  end
  $x_k = m$
end

El método termina cuando se cumple con alguna condición de paro, en este algortimo la condición de tolerancia al error se define como:

$$|x_k - \beta| < \epsilon \tag{1}$$

donde $|\cdot|$ es el valor absoluto y β es la raíz real. Después de cada iteración el intervalo se reduce a la mitad. Entonces, después de n iteraciones el intervalo original se ha reducido 2^n veces, por lo tanto, si el intervalo original es de tamaño $|a - b|$ y si el criterio de convergencia se rige por la ecuación (1), entonces se requieren n iteraciones donde n se calcula con la igualdad de la expresión:

$$\frac{|b - a|}{2^n} < \epsilon \tag{2}$$

Despejando de la ecuacion (2) se obtiene,

$$n = \frac{\log |a - b| - \log \epsilon}{\log 2} \tag{3}$$

2.2. Metodo de Newton-Raphson

El método de Newton-Raphson es un método abierto, en el sentido de que no está garantizada su convergencia global. La única manera de alcanzar la convergencia es seleccionar un valor inicial lo suficientemente cercano a la raíz buscada. Así, se ha de comenzar la iteración con un valor razonablemente cercano al cero (denominado punto de arranque o valor supuesto). Se requiere que las funciones sean diferenciables, y por tanto, continuas, para poder aplicar este método. El método de Newton-Raphson es convergente cuadráticamente, es decir, el error es aproximadamente al cuadrado del error anterior. Esto significa que el numero de cifras decimales correctas se duplica aproximadamente en

cada interacción.
El algoritmo 2 resume el método de Newton-Raphson. Observe que los parametros de entrada son la funcion $f(x)$ y punto de arranque x_0.

Algorithm 2: Método de Newton-Raphson

Input : función $f(x)$, punto inicial x_0
Output: x_k
while $|x_k - \beta| > \epsilon$ **do**
  $x_k = x_k + \frac{f(x)}{f'(x)}$;
end

2.3. Método de Secante

El método de la secante se puede pensar como una simplificación del método de Newton-Raphson. En lugar de tomar la derivada de la función cuya raíz se quiere encontrar, se aproxima por una recta secante a la curva, cuya pendiente es aproximadamente igual a la derivada en el punto inicial,

$$f'(x_k) = \frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}} \tag{4}$$

La principal diferencia con el método de Newton-Raphsaon es conocer dos puntos de la función para poder generar dicha recta.
Debido a que no podemos asegurar la convergencia tal como en el método de Newton-Raphson, hay que intentar con dos puntos de partida y analizar iteración a iteración el comportamiento de los resultados. También se pueden tener en cuenta ciertos criterios que utiliza el método para obtener la siguiente aproximación, por ejemplo, la pendiente de la recta secante no debe ser nula pues nunca intesectará con el eje x ni tendiendo a infinito (siempre y cuando la raíz se encuentre alejada a los puntos iniciales) ya que la convergencia puede llegar a ser lenta. De todas maneras, dependerá de la forma que tiene la función.

El algoritmo 3 resume el método de Newton-Raphson. Observe que los parametros de entrada son la funcion $f(x)$ y los dos puntos de arranque x_0, x_1.

Algorithm 3: Método de Secante

Input : función $f(x)$, punto inicial x_0, x_1
Output: x_k
while $|x_k - \beta| > \epsilon$ **do**
  $x_k = x_k + \frac{f(x)\cdot(x_k - x_{k-1})}{f(x_k) - f(x_{k-1})}$;
end

2.4. Interpolación Cuadrática Inversa

El método de interpolación cuadrática inversa usa tres puntos de arranque, en donde en cada iteración existen tres valores de soluciones aproximadas. Defina los puntos de arranque como a, b y c, con valores de función $f(a)$, $f(b)$ y $f(c)$, respectivamente.
El algoritmo 4 resume el método de interpolación cuadrática inversa, donde los parametros de entrada son la función $f(x)$, los tres puntos de arranque a, b, c y la tolerancia de parada.

Algorithm 4: Método de Brent

Input : función $f(x)$, puntos iniciales a, b, c, tolerancia ϵ
Output: x_k
while $|b - \beta| > \epsilon$ **do**
  $u = \frac{f(b)}{f(c)}$;
  $v = \frac{f(b)}{f(a)}$;
  $w = \frac{f(a)}{f(c)}$
  $p = v(w(u-w)(c-b) - (1-u)(b-a))$;
  $q = (w-1)(u-1)(v-1)$;
  $c = a$;
  $a = b$;
  $b = b + \frac{p}{q}$;
end

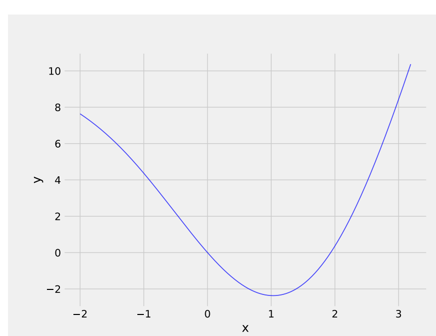
La tasa de convergencia de la interpolación cuadrática inversa para encontrar la raíz es $\gamma = 1.839$

3. Simulaciones y resultados

Se realizarón simulaciones con los distintos algoritmos planteados en este trabajo, a fin de determinar el desempeño de cada uno de estos.
La figura 1 ilustra una función objetivo la cual esta determinada por la siguiente ecuación,

$$f(x) = x^2 - 4\sin x \tag{5}$$

para todos los algoritmos se tomó un dominio que abarca $[1, 3]$, donde la raiz teórica se encuentra en $\beta = 1.93$. La figura 2 ilustra el error de cada algoritmo al buscar la raiz que hace la función $f(x) = 0$, este error se mide como el valor absoluto entre la diferencia de x_k y β, $|x_k - \beta|$. Note, el algoritmo más lento en converger es el método de Bisección, gastando para este caso particular 17 iteraciones. Mientras los algoritmos Newton-Raphson y Interpolación Cuadrática Inversa solo gastan 5 iteraciones, siendo rápidos en converger. La figura 3 presenta los deltas de cada algoritmo, donde es evidente que Bisección tiene una caida lenta, comparada



(a)

Figura 1. Funcion objetivo $f(x) = x^2 - 4\sin x$ para aplica los métodos de búsqueda de raíces.

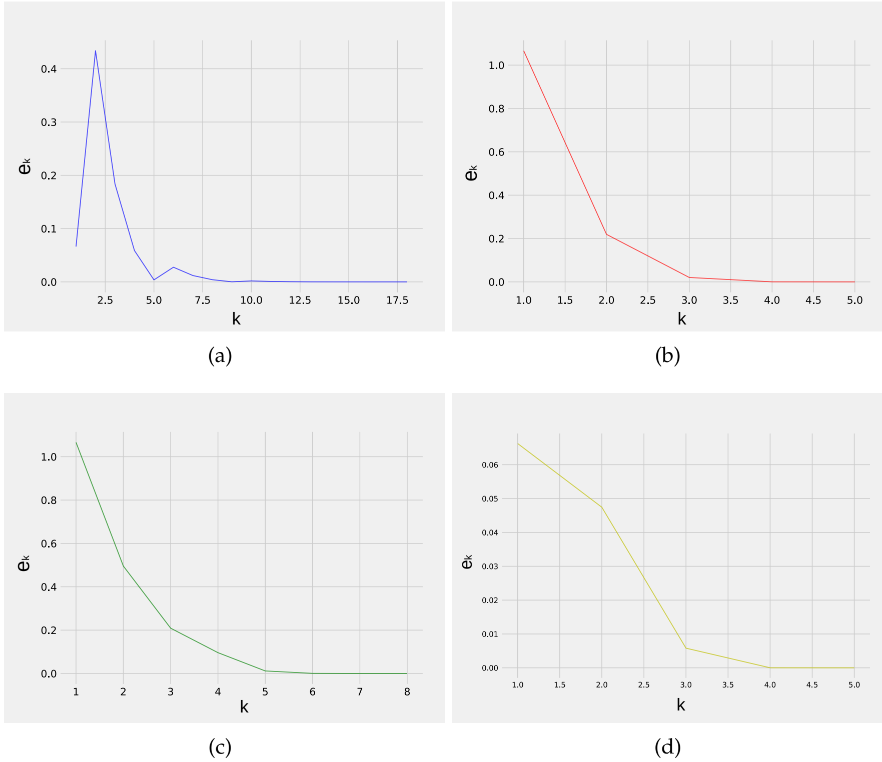
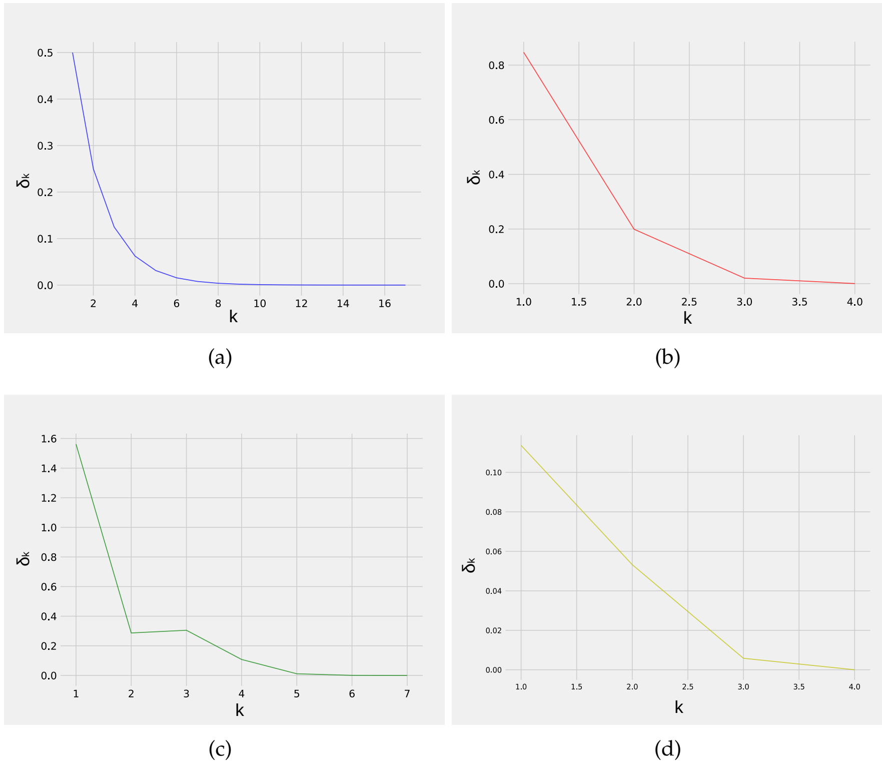


Figura 2. Análisis de error (a) Bisección (b) Newton-Raphson (c) Secante (d) Interpolación Cuadrática

con Interpolación Cuadrática.
La figura 6 presenta tiempo computacional empleado por cada algortimo al llegar a un tolerancia objetivo.

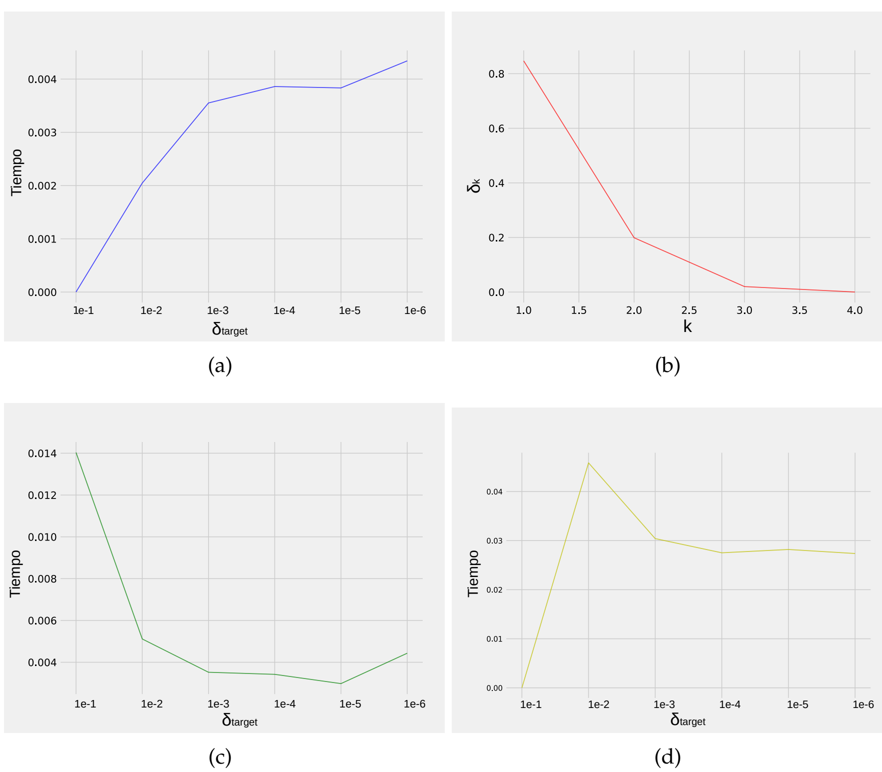
4. Conclusion

Todos los algoritmos presentados en este trabajo, son métodos que dadas unas condiciones iniciales pueden resolver el problema de búsqueda de raíces sin problemas. Sin embargo, estas condiciones iniciales deben ser bien planteados, de tal forma que los algoritmos converjan a un punto de búsqueda. El método de bisección es un método "de encierro" y tiene como desventaja que para poder aplicarlo se debe contar con un intervalo inicial, en donde $f(a) * f(b) < 0$. Este método requiere de menos pasos en un programa, sin embargo converge mas lentamente que el resto de los algoritmos planteados en este trabajo.
El método de Newton-Raphson algunas veces no converge, sino que oscila. Si la raíz es un punto de inflexión o si el valor inicial esta muy alejado de la raíz buscada. El principal inconveniente del método de Newton estriba en que requiere conocer el valor de la primera derivada de la función en el punto.
Los métodos de Newton-Raphson y Interpolción Cuadrática son los algoritmos mas rápidos en la búsqueda de la raíz.



(a) (b) (c) (d)

Figura 3. Análisis de tolerancia (a) Bisección (b) Newton-Raphson (c) Secante (d) Interpolación Cuadrática



(a) (b) (c) (d)

Figura 4. Análisis de tiempo (a) Bisección (b) Newton-Raphson (c) Secante (d) Interpolación Cuadrática