

Raíces de una función

```
clear;
close all;
% Global settings
warning('off'); % warnings about rank
```

El siguiente ejercicio contiene los siguientes codigos para su desarrollo

- [bisection.m](#)
- [newtonraphson.m](#)

3.1 Entendiendo las distintas metodologías

3.1.1 Newton-Raphson

El algoritmo de Newton-Raphson permite encontrar la raíz de una función partiendo desde un punto del dominio cercano a dicha raíz. Es un método **iterativo** y preciso que es útil sobre polinomios de grado *impar*. Desde una aproximación de **Taylor** de primer orden es posible obtener la siguiente sucesión que define el algoritmo. Es importante notar que dos valores consecutivos de la sucesión serán cada vez más similares en la medida en que se aproximen a la raíz de la función :

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

Con x_0 conocido. La figura adjunta en el enunciado explica de forma gráfica el método de estudio

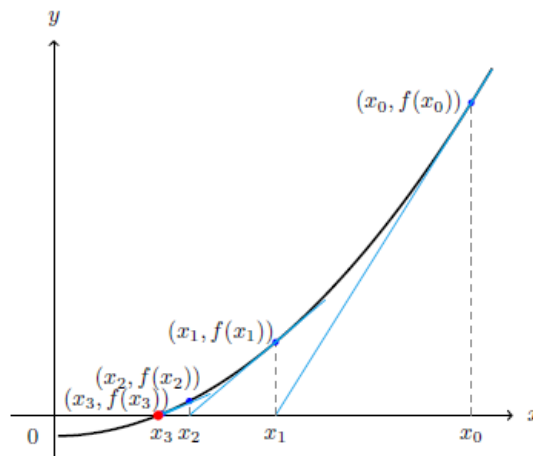


Figure 3: Descripción gráfica del método.

Formalmente sea $f(x)$ una función C^1 y continua, y x_r una raíz desconocida sobre $f(x)$. Asumamos que x_0 es un *guess* para x_r . A menos que hayamos tenido mucha suerte, $f(x_0)$ no es una raíz. Dado ese escenario queremos encontrar un x_1 tal que se mejore la performance de x_0 (esto es, estar más cerca de x_r que de x_0). Ahora bien, si suponemos que de todas maneras x_0 está lo suficientemente cerca de x_r , entonces podremos hacer una **aproximación lineal** de $f(x)$ alrededor de x_0 tal que buscaremos encontrar la intersección de esa

línea con el eje x . En síntesis, estaremos haciendo una aproximación lineal de $f(x)$ alrededor de x_0 que se escribe de la siguiente manera:

$$f(x) \approx f(x_0) + f'(x_0)(x - x_0)$$

Luego, como indicamos, encontraremos x_1 tal que $f(x_1) = 0$. Tomando esos valores y llevándolos a la aproximación lineal resulta la siguiente ecuación:

$$0 = f(x_0) + f'(x_0)(x - x_0)$$

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

En términos generales, el método **de Newton** computa la aplicación del guess, x_n , usando un guess anterior x_{n-1} y que está dado por la siguiente ecuación

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

Ahora bien el método **Newton-Raphson** aplica encontrar esa raíz pero iterando el método de Newton desde x_0 hasta que el error sea menor a una tolerancia esperada.

3.1.2 Algoritmo de Bisección

Basado en el teorema de los Valores Intermedios, el algoritmo de Bisección establece:

Sea

$f : [a, b] \rightarrow \mathbb{R}$ una función continua diferenciable C^1 donde todo valor entre $f(a)$ y $f(b)$ es imagen de al menos un valor del intervalo $[a, b]$. Luego, si $f(a)$ y $f(b)$ tienen signos opuestos ($f(a) \cdot f(b) < 0$), el teorema asegura la existencia de al menos un valor x del dominio de f tal que $f(x) = 0$.

Para programar el algoritmo de Bisección siga los siguientes pasos

1. Compute una tolerancia $\tau = 10^{-4}$ y defina un intervalo $[a, b]$ que contenga la raíz de la función.
2. Dada una función que cumpla con las características, compute $f(a)$ y $f(b)$
3. Compute $\bar{r} = \frac{a+b}{2}$ y luego obtenga $f(\bar{r})$. Si el signo de $f(\bar{r})$ es igual al signo $f(a)$, entonces el nuevo intervalo definido en el dominio de la función será $[\bar{r}, b]$ de otra forma el nuevo intervalo será $[a, \bar{r}]$.
4. Compute el error de aproximación $\epsilon = f(\bar{r})$
5. Mientras $\epsilon > \tau$ repita el proceso hasta que el algoritmo converja a la tolerancia deseada

Generalmente el algoritmo de bisección no produce una solución exacta a la ecuación $f(x)=0$. Ahora bien, se puede proveer de una estimación sobre el error absoluto. El teorema es el siguiente.

Teorema. Sea $f(x)$ una función continua entre $[a, b]$ tal que $f(a)f(b) < 0$. Luego de N iteraciones del método de bisección, sea $[X_n]$ el punto intermedio en el subintervalo n -ésimo $[a_n, b_n]$

$$X_n = \frac{a_n + b_n}{2}$$

Entonces, existe una solución exacta y verdadera de la ecuación $f(x) = 0$ en el subintervalo $[a_n, b_n]$ y el error absoluto es

$$\epsilon > [x^* - x_n] \leq \frac{b-a}{2^{n+1}}$$

Notemos que podemos reorganizar el límite del error de modo de poder notificar cuál es el número mínimo de iteraciones requeridas para garantizar que el error absoluto sea menor al descrito

$$\frac{b-a}{2^{n+1}} < \epsilon$$

$$\frac{b-a}{\epsilon} < 2^{n+1}$$

$$\ln \frac{b-a}{\epsilon} < n+1 \cdot \ln(2)$$

$$\frac{\ln \frac{b-a}{\epsilon}}{\ln(2)} - 1 < n$$

(a) Usando el método Newton-Raphson, encuentra la raíz de las siguientes funciones

$$f(x) = x^3 + x + 2$$

$$f(x) = x^5 + x^4 + x^3 + x^2 + 1$$

$$f(x) = \log(x) + \log(3x^3)$$

```
clear;
f = @(x) x^3 -x + 2;
x0 = 0
```

```
x0 = 0
```

```
newtonraphson(f,x0)
```

```
ans = -1.5214
```

```
x0 = 1;
F = { @(x) x.^3-x+2; @(x) x.^5+x.^4+x.^3+x.^2+1; @(x) log(x)+log(3*x.^3) } ;
newtonraphson(F{1},x0)
```

```
ans = -1.5214
```

```
newtonraphson(F{2},x0)
```

```
ans = -1.2499
```

```
newtonraphson(F{3},x0)
```

```
ans = 0.7598
```

(b) Obtenga la raíz de las funciones usando el algoritmo de Bisección. ¿Obtiene los mismos resultados?

Como podemos ver se obtienen los mismos resultados para ambos métodos numéricos de cálculo de raíces

```
% La funcion bisection calcula para un f e intervalo a,b, la raiz y error
F = {@(x) x.^3-x+2; @(x) x.^5+x.^4+x.^3+x.^2+1; @(x) log(x)+log(3*x.^3)} ;
a = -6;
b = 6;
bisection(F{1},a,b)
```

```
Solucion: -1.5214  Error: 2.6274e-46
ans = -1.5214
```

```
bisection(F{2},a,b)
```

```
Solucion: -1.2499 Error: 8.7581e-47
ans = -1.2499
```

```
% Para log definimos el intervalo entre 0 y +infinito pues en negativos se
% indefiniría y ya no se cumpliría el teorema
bisection(F{3},0,12)
```

```
Solucion: 0.75984   Error: 2.1895e-47
ans = 0.7598
```

(c) ¿Cuál algoritmo es más eficiente? Justifique brevemente

```
f = @(x) bisection(F{1},a,b); % handle to function
timeit(f)
```

[illegible]

```
Solucion: -1.5214 Error: 2.6274e-46
Solucion: -1.5214 Error: 2.6274e-46
Solucion: -1.5214 Error: 2.6274e-46
Solucion: -1.5214 Error: 2.6274e-46
Solucion: -1.5214 Error: 2.6274e-46
ans = 2.5035e-04
```

```
f = @( ) newtonraphson(F{1},x0); % handle to function
timeit(f)
```

```
ans = 0.0217
```

Si medimos la eficiencia en términos de tiempo en que se demora el método en calcular la raíz, el algoritmo de bisección se demora 0.0048 segundos, mientras que el algoritmo de Newton Raphson se demora 0.0270 segundos. Ahora bien, en programación la eficiencia no solo tiene que ver con el tiempo si no que de los parámetros con los cuales se construyen las funciones, y su capacidad de replicabilidad. Por ello, presentaré una breve discusión de las ventajas y desventajas de ambos métodos, haciendo alusión a sus potenciales respecto a eficiencia y eficacia (precisión).

1. La ventaja principal del método de Newton es que cuando se converge, esto ocurre rápidamente. Ahora bien, no asegura la existencia de esa convergencia, lo que evidentemente es una desventaja comparada con el método de bisección que garantizan la convergencia hacia una solución.
2. Además, el método de Newton requiere computar las derivadas de la función objetivo, algo que potencialmente puede ser una desventaja si la derivada es difícil de calcular.
3. Penúltimo el criterio de convergencia del método de Bisección y Newton es muy distinto: en el primero nosotros sabemos qué tan cerca estamos de la solución pues estamos computando la estimación dentro de intervalos que si o si contienen la solución; en el segundo, no sabemos realmente que tan cerca estamos de la solución.
4. Por último, computacionalmente es mucho menos preciso Newton que Bisección. Esto pues en el primero debemos indicar hasta que número de iteraciones se quiere seguir probando la convergencia, mientras que en método de bisección el criterio de *stopping* es qué tan pequeño es el error de estimación. Las iteraciones quedan sujetas a ello.

(d) Cree las funciones NR y BS que permite encontrar las raíces de una función con características indicadas mediante el método Newton-Raphson y Bisección respectivamente. Utilice las funciones para computar raíces de los incisos anteriores nuevamente.

Las funciones se crearon enseguida, dado que me es más fácil pensar en las funciones en términos generales y luego ver si su aplicación funciona para los casos particulares propuestos.

Solo agregar que en ambos métodos de igual manera debe existir una vigilancia del usuario sobre los **dominios de la función**. No es trivial dónde viven las funciones a evaluar, por lo que son métodos dependientes de las cotas de las funciones. Por lo mismo se crearon *warnings* ante ciertas propiedades que el usuario no podría preveer. Por ejemplo,

1. El número de iteraciones excede las que el programa se propuso. Para lograr converger se propusieron al menos 100 para Newton-Raphson, mientras que para Bisección eso no fue necesario.

2. En el caso de bisección la direccionalidad de las funciones era relevante, por lo que dado el teorema, era necesario asegurarse que las soluciones de la función vivieran dentro del intervalo. Lo positivo es que da igual si se elige un intervalo mayor al encontrado (por ejemplo que la solución viva entre $[a,b]$ pero se elija $[a+30,b+30]$) pues la estimación es tan consistente que eso no trae problemas.

3.2 Aplicación de tiro vertical

En física la notificación de Tiro Vertical describe un movimiento rectilíneo uniformemente acelerado, es decir, un movimiento que se caracteriza por tener aceleración constante. Este tipo de movimiento intenta estudiar la trayectoria de objetos que son lanzados verticalmente desde una cota inferior hacia una cota superior, es decir, hacia arriba. Una buena forma de comprender este movimiento es imaginar a una máquina “lanzapelotas” posicionada en 90° . La trayectoria del movimiento de la pelota puede ser explicada por las ecuaciones de un Tiro Vertical.

El propósito de este ejercicio es utilizar el algoritmo de Bisección y Newton-Raphson para hallar la solución al problema.

Contexto

Imagine que durante un día de partido de tenis, su rival no se presenta y usted decide practicar su saque con la ayuda de una máquina “lanzapelotas”. Para optimizar sus movimientos y energía, **desea saber el momento exacto en que la pelota alcanzar a la altura máxima en su recorrido**, de tal forma de poder aplicar un golpe limpio. Usted sabe que las ecuaciones de Tiro Vertical son las que mejor describen el movimiento de la pelota y están dadas por

$$h(t) = h_o + v_o \cdot t - \frac{1}{2} g \cdot t^2$$

$h(t)$: altura de la pelota al instante t

h_o : altura inicial $\rightarrow 50$ (cm)

v_o : velocidad inicial $\rightarrow 4.5 \left(\frac{m}{s} \right)$

g : aceleración de gravedad $\rightarrow 9.81 \left(\frac{m}{s^2} \right)$

(a) Mediante el algoritmo de Bisección y Newton-Raphson obtenga el tiempo t^* en el cual la pelota alcanza la altura máxima.

El lanzamiento de la pelota es una parábola cuya trayectoria se describe por $h(t)$. En ese sentido nosotros queremos saber cuál es la cúspide de esa trayectoria, o matemáticamente, queremos saber el valor de

t cuando $\frac{\partial h(t)}{\partial t} = 0$. Los pasos son

1. Encontrar y clasificar los extremos globales de la función (que realizaremos con método Newton-Raphson y Bisección)
2. Encontrar los puntos críticos computando la primera derivada y luego despejando t

```
f = @(t)50+4.5*t -0.5*9.81*t^2; % Expresamos el problema numericamente y con x=t
```

Si calculamos con el método de Newton Raphson obtendremos la menor raíz de la solución ($t = -2.7668$)

```
t0= 0 % guess
```

```
t0 = 0
```

```
newtonraphson(f,t0)
```

```
ans = -2.7668
```

Cuando calculamos con el método de bisección obtenemos la mayor raíz de la solución ($t=3.6843$)

```
bisection(f,0,8)
```

```
Solucion: 3.6843 Error: 8.7581e-47  
ans = 3.6843
```

Hasta aquí uno se podría asustar y pensar que hay un error. Pero, recordemos lo que indicamos al inicio de este ejercicio: ambos métodos son eficientes pero sensibles a los intervalos por el cuál nosotros los estudiamos, por lo que podremos muchas veces obtener raíces para máximo o mínimos, y no necesariamente serán los globales. Veamos si tomamos el método de bisección por el otro lado obtenemos la raíz -2.7668

```
bisection(f,-8,0)
```

```
Solucion: -2.7668 Error: 1.7516e-46  
ans = -2.7668
```

Lo mismo con Newton Raphson

```
newtonraphson(f,1)
```

```
ans = 3.6843
```

Lo que evidentemente evidencia es que **entre estas raíces existe el máximo de la función, y con ello podemos clasificar los extremos globales de la función**

El segundo paso que viene consiste en derivar la función

```
syms t  
f1 = str2sym(char(f));  
df = diff(f1);  
df = matlabFunction(df);  
% Obtener punto critico. Sabemos que newton raphson funcional igual, asi  
% que para simplificar ocupamos biseccion  
[t1] = bisection(df,0,8)
```

```
Solucion: 0.45872 Error: 2.1895e-47  
t1 = 0.4587
```

Entonces con la primera bisección encontramos los intervalos donde vive el máximo o mínimo global, y ahora con la segunda de estas obtuvimos el máximo.

(b) Obtenga la altura máxima que alcanza la pelota.

Tenemos que evaluar la función original y obtengamos el máximo

$$\max f = 50 + 4.5 \cdot t_1 - 0.5 \cdot 9.81 \cdot t_1^2;$$

Nuevamente, como nuestro punto crítico vive en el intervalo de los zeros que calculamos inicialmente podemos decir que en 0.4591 " es un punto de inflexión entre mínimos globales y máximos globales.

(c) ¿Cuál es la velocidad de la pelota en el instante t? ¿Por qué son válidos estos algoritmos para encontrar la solución al problema? Explique brevemente

La velocidad se obtiene así

$$v(t) = \frac{\partial h(t)}{\partial t} = -9.8t + 4.5$$

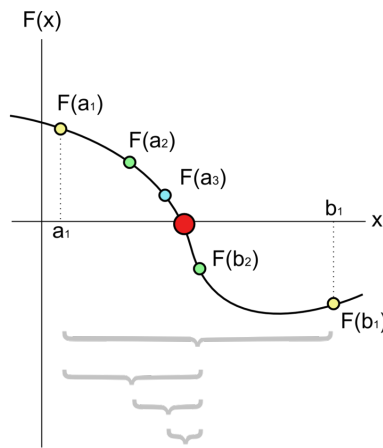
$$-9.8 \cdot t_1 + 4.5$$

$$\text{ans} = 0.0046$$

Notemos que la velocidad en ese punto es aproximadamente cero, lo que rectifica que nuestros cálculos son correctos dado que en un movimiento parabólico regido por MRU y MRUA, cuando se alcanza el máximo de distancia, el vector velocidad es igual a cero debido a que el cuerpo pierde toda su energía cinética que lo hizo desplazarse hacia la posición máxima. Luego, el cuerpo va a seguir su trayectoria pero a una velocidad dependiente de la aceleración. Formalmente cuando la velocidad en el eje y vale 0, obtenemos el tiempo t que tarda el cuerpo en llegar a dicha altura.

Ahora respecto a la validez de estos métodos para estos problemas, se pueden hacer dos acotaciones, una formal y otra sustantiva.

1. El movimiento descrito consiste en una función $f: \mathbb{R}^2 \rightarrow \mathbb{R}$, acotada, continua y diferenciable en todo su intervalo. En ese sentido tanto el método de Newton como de Bisección que se inspira en el **teorema del valor intermedio**, pueden ser matemáticamente aplicados.
2. En términos sustantivos, nosotros tenemos una trayectoria que se describe por una función de la cuál desconocemos sus puntos o raíces exactamente. Una forma de aproximarnos es partir conociendo sus valores extremos (Teorema de Weierstrass). Una vez conocido ese máximo y ese mínimo, por el teorema de la acotación (Bolzano) podemos establecer que en toda función continua cerrada, esta es acotada. El ejercicio será ir construyendo cada vez menores intervalos que nos lleven a un único valor c que está contenido en la cota. Gráficamente



En ese sentido, en esta búsqueda por ese $f(c) = 0$, estaremos **mapeando** todo el recorrido de la función. Con esa aproximación, luego, podemos conocer cualquier momento de nuestra distribución, tal como los puntos máximos, la velocidad (varianza) u otros.

(d) Imagine que ahora desea estudiar el movimiento denominado caída libre, ¿Podría utilizar las funciones NR y BS para estudiar la trayectoria del objeto de estudio? No es necesario que haga cálculos. Explique brevemente la lógica que hay detrás de los movimientos estudiados.

Anteriormente se entregó una intuición sobre el estudio de funciones de las cuales tenemos información general, pero no sabemos cómo se comporta en cada uno de los puntos. Un punto de partida es el teorema de los valores extremos absolutos, para luego pasar al teorema de la acotación, ambos componentes esenciales del teorema del valor intermedio (de hecho, este es un resultado de estos). Una vez reconocido eso entendremos que **si se puede tomar otro movimiento que se comporta como una subsucesión de cada punto de la sucesión general (la función del movimiento parabólico), entonces podremos también aplicar esta metodología.**

En términos simples, la caída libre que va en la misma dirección que el proyectil en la parábola se comporta de igual manera que un movimiento parabólico, solo que comprende la trayectoria desde que la altura es máxima (y velocidad cero), hasta cuando el proyectil toca el piso. En condiciones ideales donde estos proyectiles son lanzados al aire (y no hay resistencia) podemos notar que

1. La caída libre y el lanzamiento del proyectil desde una misma altura tardan lo mismo en llegar al suelo
2. Tanto en esta parábola como semiparábola, la masa es independiente de la trayectoria.

Como resultante la función que describe el tiempo que tarda en llegar a la altura máxima el cuerpo, es el mismo que tarda en llegar desde el punto máximo hacia el suelo, y por consiguiente, incluso basta con calcular una semiparábola de caída libre y obtendremos el mismo resultado respecto a los momentos de la función. La única diferencia es **el intervalo de análisis, algo que recalcamos bastante al inicio**. Así, en el caso de la parábola tomamos todo el intervalo comprendido entre sus raíces, mientras que en la semiparábola, la cota inferior corresponderá a la altura dada del problema (que debe coincidir con la máxima de la parábola completa).

En síntesis, si la caída libre sigue la misma trayectoria de lanzamiento que el movimiento parabólico, podemos llegar a los mismo resultados. Cosa distinta es si la caída libre es totalmente independiente, pues en ese caso,

tendremos que evaluar la función objetivo nuevamente con la misma rigurosidad con que lo hicimos para el caso del movimiento parabólico.

Apéndice

1. Resolvamos x en los números reales

$$-4.9x^2 + 4.5x + 50 = 0$$

$$x^2 - \frac{45}{49}x - \frac{500}{49} = 0$$

$$\left(x - \frac{45}{98}\right)^2 = \frac{100025}{9604}$$

2. Tomemos el cuadrado por ambos lados

$$x_1^* = \frac{45}{98} + \frac{5\sqrt{4001}}{98} \approx 3.6843 \quad x_2^* = \frac{45}{98} - \frac{5\sqrt{4001}}{98} \approx -2.7668$$