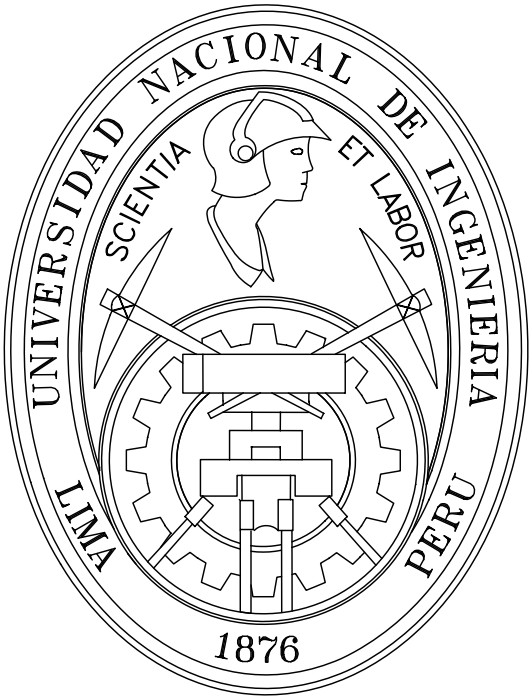
UNIVERSIDAD NACIONAL DE INGENIERIA

FACULTAD DE INGENIERIA MECANICA

Departamento Académico de Ciencias Básicas, Humanidades y cursos complementarios



METODOS NUMERICOS (MB –536)

SOLUCION DE SISTEMAS DE ECUACIONES NO LINEALES

Profesores:

- **Garrido Juárez, Rosa**
- **Castro Salguero, Robert**
  Obregón Ramos, Máximo

2009-1

ECUACION NO LINEAL

INTRODUCCION.

Una función lineal es aquella que satisface las siguientes propiedades.

- Aditividad: $f(x+y) = f(x) + f(y)$
- Homogeneidad: $f(\alpha x) = \alpha f(x)$

Conocido también como principio de superposición.

Una función no lineal, es aquella que no cumple dicho principio.

Ejemplos de algunas funciones no lineales:

- $f(x)=x^2+1$ Parábola
- $\frac{1}{\sqrt{\lambda}} = -2\log\left(\frac{k/D}{3.7} + \frac{2.51}{\mathrm{Re}\sqrt{\lambda}}\right)$, donde $f(x)=g(\lambda)$ Función de Colebrook-White

Las raíces de muchas de estas funciones no pueden calcularse analíticamente. Pero si se podrá resolver, usando métodos iterativos con ciertos criterios de parada.

SOLUCION DE ECUACION NO LINEAL

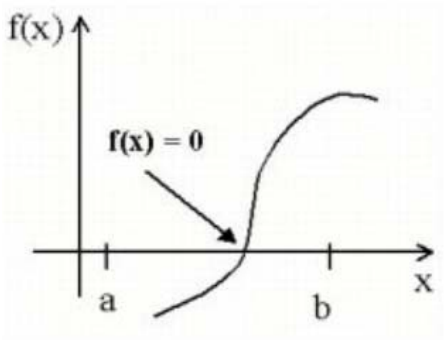


Figura Solución de una función no lineal

Los diversos métodos las podemos clasificar en 2 tipos:

Métodos Cerrados: Se caracterizan porque se necesita de 2 valores iniciales para la raíz, además se aprovecha de que en estos dos extremos la función cambia de signo. Dentro de ellas tenemos:

- Método gráfico
- Método de bisección
- Método de la falsa posición
- Método de búsqueda por incrementos

Métodos Abiertos: Se caracterizan porque se necesita normalmente un valor de partida para la raíz, normalmente los métodos anteriores tienen la ventaja de encontrar por lo menos una solución y siempre converge, en cambio en estos métodos se presentaran casos en que

no converge a la solución, pero cuando converge normalmente son más rápidos que los métodos cerrados. Dentro de ellas tenemos:

- Método de Punto fijo
- Método de Newton-Raphson
- Método de la secante

Nota: Cabe destacar que algunos autores designan a algunos de estos métodos con otros nombres.

Criterios de parada para los algoritmos

Considerando que se desea hallar la solución de la función $f(x)$, entonces se tiene los siguientes criterios de paro para estimar el error en cada iteración.

- Error absoluto: $\xi_a = |x_{nuevo} - x_{anterior}|$
- Error relativo: $\xi_r = \dfrac{|x_{nuevo} - x_{anterior}|}{x_{nuevo}}$
- Error como la función: $\xi_f = |f(x_{nuevo})|$
- Máximo número de iteraciones: Usado más en los métodos donde puede suceder la divergencia.

MÉTODOS GRÁFICO

Desde hace mucho tiempo este método era uno de los preferidos para poder hallar la solución, se tabulaban diferentes puntos en un gráfico hecho a mano, luego visualmente se realizaba un ajuste y se aproximaba las soluciones posibles. Hoy en día con las herramientas informáticas se disponen de varios software que te permiten graficar cualquier función matemática e inclusive te permite hacer ampliaciones en la región deseada y puedes obtener grandes precisiones, como por ejemplo el Autocad, Matlab, etc.

Ejm:

Considere la siguiente función para calcular la velocidad del paracaidista

$$v = \frac{gm}{c}\left(1 - e^{-(c/m)t}\right)$$

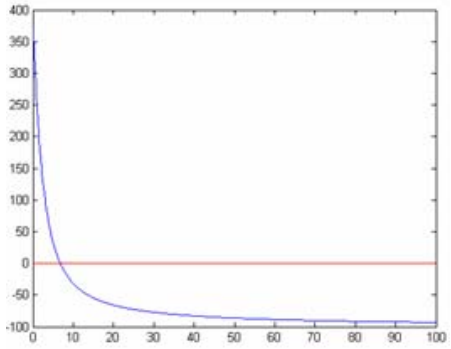
Donde:

v = velocidad g = gravedad t = tiempo c = coeficiente de arrastre m = masa

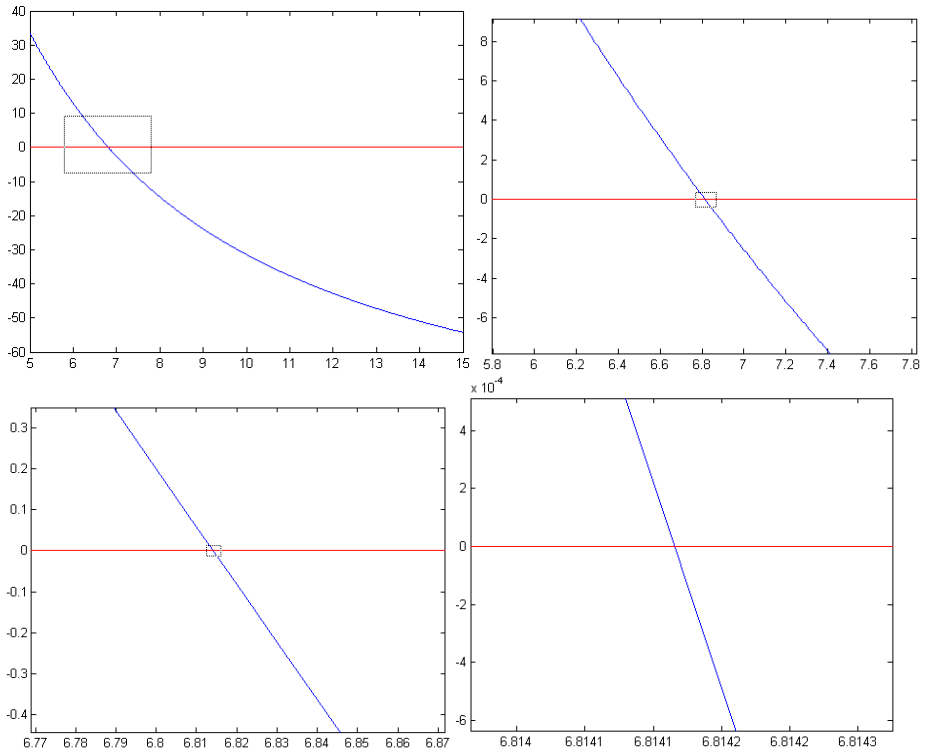
Ahora imagine que se desea hallar c para v=100 m/s, g=9.81 m/s², t=50 s, m=70 kg.

Usando el Matlab revisaremos el comportamiento del grafico en un rango relativamente grande [0 100], con las siguientes instrucciones:

```
ci=0;cf=100;np=1000;
h=(cf-ci)/np;c=ci:h:cf;
g=9.81;m=70;t=50;v=100;
y=g*m*(1-exp(-c*t/m))./c-v;
plot(c,y)
hold on
plot([c(1) c(end)],[0 0],'r')
```



Claramente podemos notar visualmente que la solución se encuentra cercano a 10, entonces vuelvo a graficar con un rango cercano a la solución por ejemplo de 5 a 15. Además el Matlab tiene un icono comando para poder realizar un zoom en cualquier región deseada, y podemos repetir este último paso varias veces, obteniendo lo siguiente:



Se puede notar que la respuesta es 6.814, con 3 decimales exactos. Si seguimos los pasos anteriores podemos obtener mayor precisión. La desventaja de este método es que depende de la visión humana y por lo tanto no es automático, es decir no se dispone de un programa que te entregue la respuesta directamente con una cierta precisión, ahora imagina que te pidan graficar c en función de t, con este método seria muy tedioso.

MÉTODOS DE BISECCION.

Conocido también como corte binario, de partición de intervalos o de Bolzano, es un tipo de búsqueda incremental en el que el intervalo se divide siempre a la mitad, y se evalúa si tiene un cambio de signo en cada sub-intervalo, solo uno de los dos deberá tener la respuesta.
Actualmente llamamos sucesión fundamental o también sucesión de Cauchy a una sucesión que tiene la siguiente propiedad: se puede lograr que dos términos difieran entre sí tan poco como se desee, a condición de considerar los subíndices de ambos a partir de alguno. Enunciado en lenguaje formal: $\forall \varepsilon > 0,\ \exists$ n0 $\in$ N, n0 = n0(ε) tal que si n > no, □ p>0 se tiene que |xn+p -xn|<ε ."

Teorema Si la función f(x) es continua respecto a la variable x entre los límites x=x0, x= X, y si se designa por b a una cantidad intermedia entre f(x0) y f(X), se podrá siempre satisfacer la ecuación f(x)=b para uno o varios valores reales de x comprendidos entre x0 y X.

Demostración: Para establecer la proposición precedente basta hacer ver que la curva que tiene por ecuación y = f(x) cruzará una o varias veces a la recta que tiene por ecuación y= b en el intervalo comprendido entre las ordenadas que corresponden a las abscisas x0 y X, es evidente que esto

tendrá lugar bajo la hipótesis admitida. En efecto, al ser continua la función f(x) entre los límites x= x0 , x= X, la curva que tiene por ecuación y= f(x) y que pasa primero por el punto correspondiente a las coordenadas x0 y f(x0) y segundo por el punto correspondiente a las coordenadas X y f(X) será continua entre estos dos puntos, y como la ordenada constante b de la recta que tiene por ecuación y= b se encuentra comprendida entre las ordenadas f(x0), f(X) de los dos puntos que estamos considerando, la recta pasará necesariamente entre esos dos puntos, lo que no se puede hacer sin reencontrar en el intervalo a la curva mencionada.

Teorema de Bolzano: Si f(x) es una función continua en el intervalo [a, b], y si, además, en los extremos del intervalo la función f(x) toma valores de signo opuesto (f(a) * f(b) < 0), entonces existe al menos un valor c Î (a, b) para el que se cumple: f(c) = 0.

En general:
Si f(a)*f(b)<0 Entonces existen 2*k+1 soluciones
Si f(a)*f(b)>0 Entonces existen 2*k soluciones
Donde k es un entero positivo.
Tratando de resolver el problema anterior implementamos una función para ese caso:

```
function y=fun(c)
g=9.81;m=70;t=50;v=100;
y=g*m*(1-exp(-c*t/m))./c-v;
```

Luego implementamos la función para calcular la raíz usando el método de la bisección.

```
function x=biseccion(f,a,b,errormax)
if (feval(f,a)*feval(f,b))>0
   disp('En este tramo el método no asegura nada');
    x=0;
else
    er=inf;
    while (er>=errormax)
    c=(a+b)/2;
       if (feval(f,a)*feval(f,c))<=0
           b=c;
           else
           a=c;
       end
    er=abs((b-a)/a);
    end
x=c;
end
```

Probamos para el tramo 1 a 100 y obtenemos:

```
>> sym(biseccion('fun',1,100,10^-10),'d')     % con 10 dígitos de precisión
 ans =6.8141623619558231439441442489624
>> sym(biseccion('fun',1,100,10^-11),'d')  % con 11 dígitos de precisión
 ans =6.81416236191080315620638430118 56
>>sym(biseccion('fun',1,5,10^-11),'d')
En este tramo el método no asegura nada
 ans =0
```

El método de la bisección es el mas sencillo de implementar, así como también es uno de los mas ineficientes, aunque la gran ventaja es que siempre converge, siempre que se encuentre un cambio de signo en la función. Así mismo el algoritmo indicado anteriormente puede ser optimizado si evaluamos la función la menor cantidad de veces, existe casos en el cual la función a evaluar contiene varias líneas de código, lo cual consumirá recursos de tiempo al computador, se propone el siguiente algoritmo para ello:

```
function x=biseccion_better(f,a,b,errormax)
fa=feval(f,a); fb=feval(f,b);
if (fa*fb)>0
  disp('En este tramo el método no asegura nada');
  x=0;
else
   er=inf;
   while (er>=errormax)
   c=(a+b)/2;    fc=feval(f,c);        p=fa*fc;
   if p==0
      er=0
   else
         er=abs((b-a)/a);
      if p<0
         b=c;  fb=fc;
      else
         a=c;
         fa=fc;
      end
   end
   end
x=c;
end
```

Con respecto al error cometido, este puede ser calculado a priori, debido a que el error es igual la mitad del intervalo es decir (a+b)/2. En cada iteración se reducirá a la mitad, por lo tanto:

$$\varepsilon_a^{\ k} = \frac{b-a}{2^k}$$

Donde k viene ha ser el número de iteración.

$$k = \log_2\left(\frac{(b-a)}{\varepsilon_{deseado}}\right)$$

Ahora tratemos de graficar c en función de t[0 .. 50], para ello modificamos los programas:

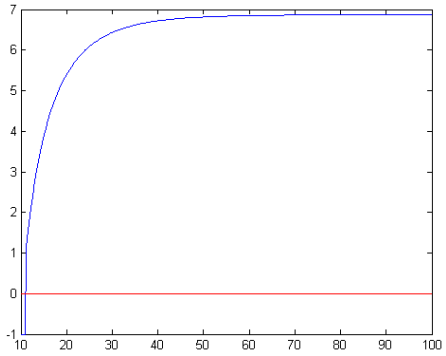
```
function y=fun(c,t)
g=9.81;m=70;v=100;
y=g*m*(1-exp(-c*t/m))-v*c;
```

Luego de desarrollar el programa ingresamos las siguientes instrucciones:

```
function x=biseccion_better(f,a,b,errormax,t)
fa=feval(f,a,t);fb=feval(f,b,t);
if (fa*fb)>0
  disp('En este tramo el método no asegura
nada');
  x=-1;
else
   er=inf;
   while (er>=errormax)
   c=(a+b)/2;    fc=feval(f,c,t);    p=fa*fc;
   if p==0
      er=0
   else
      er=abs((b-a)/a);
      if p<0
         b=c;       fb=fc;
      else
         a=c;      fa=fc;
      end
   end
   end
x=c;
end
```

```
c=[];
x=10:0.1:100;
for t=x
   c=[c biseccion_better('fun',1,10,10^-3,t)];
end
plot(x,c)
hold on
plot([x(1) x(end)],[0 0],'r')
```

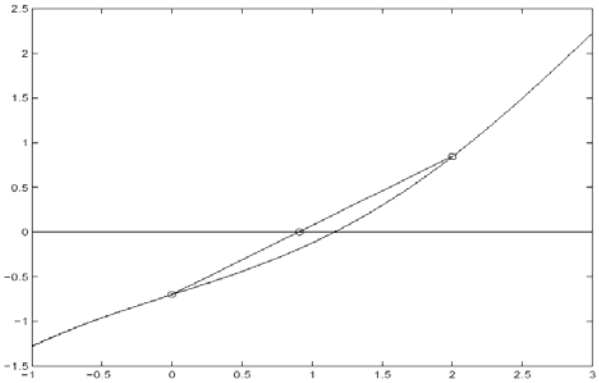
Estas últimas instrucciones nos permitirán graficar c en función del tiempo.



Gráfica c vs t

METODO DE LA FALSA POSICION

También llamado en latín “regula falsi”, método de interpolación o método de la secante. Este método también requiere indicar un intervalo en donde los extremos existe un cambio de signo, la diferencia con el método anterior es que este traza un segmento entre los 2 puntos de las graficas correspondiente a los extremos del intervalo, y considera el punto intermedio como el punto de intersección de este segmento con el ejex, para luego reducir el segmento según se evalué el cambio de signo.



Dicho punto puede ser calculado por la siguiente expresión:

$$\frac{f(a)}{c-a} = \frac{f(b)}{c-b}$$

$$c = b + \frac{f(b)*(b-a)}{f(a)-f(b)}$$

Implementamos los programas en matlab y lo acondicionamos para realizar las comparaciones entre el método de la bisección y la falsa posición:

```
function [ni,x]=bis (f,a,b,errormax,t)
fa=feval(f,a,t);fb=feval(f,b,t);
ni=0;
if (fa*fb)>0
   disp('No asegura nada');
   x=-1;
else
    er=inf;
    while (er>=errormax)
    ni=ni+1;    c=(a+b)/2;
    fc=feval(f,c,t);    p=fa*fc;
    if p==0
       er=0
    else
       er=abs((b-a)/a);
       if p<0
          b=c;     fb=fc;
       else
          a=c;      fa=fc;
       end
    end
    end
x=c;
end
```

```
function [ni,x]=falsi(f,a,b,errormax,t)
fa=feval(f,a,t);fb=feval(f,b,t);
ni=0;
if (fa*fb)>0
   disp('En este tramo el método no asegura
nada');
   x=-1;
else
    er=inf;
    while (er>=errormax)
    ni=ni+1;  c=b+fb*(b-a)/(fa-fb);
    fc=feval(f,c,t);     p=fa*fc;
    if p==0
       er=0
    else
       er=abs((b-a)/a);
       if p<0
          b=c;   fb=fc;
       else
          a=c;   fa=fc;
       end
    end
    end
x=c;
end
```

Y ejecutamos las funciones con los mismos parámetros.

```
>> [ni x]=bis('fun',1,100,10^-5,50)
ni =    22          x =    6.8142

>> [ni x]=falsi('fun',1,100,10^-5,50)
```

ni = 13 x = 6.8142

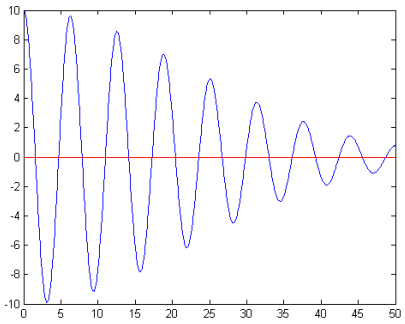
En este caso el método de la falsa posición es más rápido, es decir en menor cantidad de iteraciones llega a la respuesta.

MÉTODO DE BÚSQUEDA POR INCREMENTOS

Este método en realidad se basa también en el evento de cambio de signo y es muy útil para hallar varias soluciones, la búsqueda puede ser solamente con una búsqueda por pasos de valor h suficientemente pequeña

Ejemplo: Tomemos la siguiente gráfica generada por las siguientes sentencias.

```
xi=0;xf=50;np=1000;
h=(xf-xi)/np;x=xi:h:xf;
y=10*exp(-0.001*x.^2).*cos(x);
plot(x,y)
hold on
plot([x(1) x(end)],[0 0],'r')
```



Se puede observar en la gráfica que existen varias soluciones, con este método se pueden hallar todas estas soluciones, para ello usemos el siguiente programa de matlab:

```
function y=fun1(x)
g=9.81;m=70;v=100;
y=10*exp(-0.001*x.^2).*cos(x);
```

```
function [ni,x]=busqueda(f,a,b,h)
x=[];
ni=0;
for c=a:h:(b-h)
   if (feval(f,c)*feval(f,c+h)<=0)
      x=[x;c+h/2];
   end
ni=ni+1;
end
```

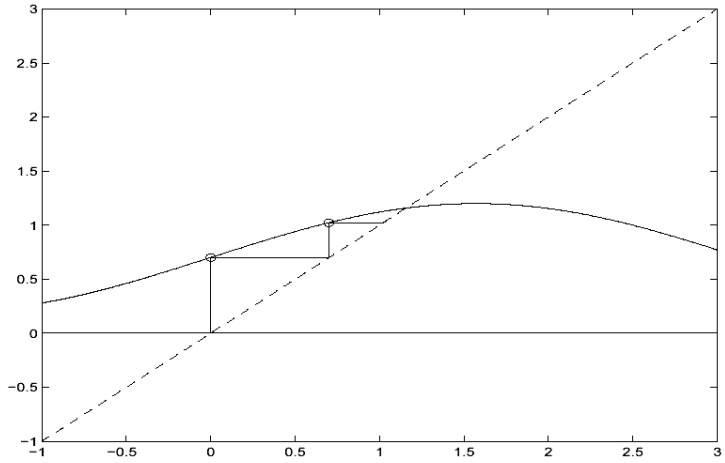
Obteniendo los siguientes resultados:

```
[ni x]=busqueda('fun1',0,50,10^-2)
ni =       5000
x =
   1.5750
   4.7150
   7.8550
  10.9950
  14.1350
  17.2750
  20.4250
  23.5650
  26.7050
  29.8450
  32.9850
  36.1250
  39.2650
```

42.4150
45.5550
48.6950
Estas soluciones le han tomado 5000 iteraciones.

Este método es muy bueno para obtener todas las raíces, pero tiene la desventaja que es muy lento, si se le da un h del orden de 10^-8 seguramente que el procesador del computador le tomará algunos minutos para resolverlo y si le damos un intervalo mayor podríamos perder algunas raíces.
Este método puede ser mejorado tomando un h mayor y a su vez para mejorar la precisión en un intervalo que se encuentre un cambio de signo, utilizar uno de los 2 métodos anteriores (bisección o falsa posición). Se deja propuesto implementar este programa.

METODO DEL PUNTO FIJO



Definición:
Diremos que una función
$f:[a,b]\to\Re$ tiene a $x\in[a,b]$ como punto fijo si f(x)=x

Teorema:
Sea $f:[a,b]\to\Re$ continua con *f*([a, b]) $\subset$ [a, b]. Entonces *f* tiene un punto fijo, si además, *f* es derivable y *f'(x)* $\neq 1 \quad \forall\ x\in[a,b]$ el punto fijo es único.

Teorema del Valor Medio:
Sea $f:[a,b]\to\Re$ continua y derivable en $(a,b)\Rightarrow\exists c\in(a,b)$ tal que *f(b)-f(a)= f'(c)(b-a)*

Teorema:
Sea $f:[a,b]\to\Re$ continua y derivable en (a, *b)* con:

$$f([a,b])\subset[a,b]$$

$$\left|f'(x)\right|\le k<1 \qquad \forall x\in(a,b)$$

Entonces si s_0 es cualquier número$\in[a,b]$, la sucesión definida por

$s_1=f(s_0),.....s_n=f(s_{n-1}) \qquad \forall n\in N$

Converge al único punto fijo $s\in[a,b]$

Además las cotas para el error son:

1.– $|S_n - S| \le K^n máx\{S_0 - a, b - S_0\} \le K^n (b-a)$

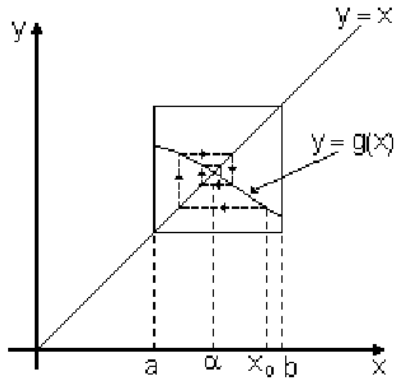
2.– $|S_n - S| \le \dfrac{K^n}{1-K}|S_0 - S_1|$

Teorema . *Sea g una función con derivada continua en un intervalo de la forma $(\alpha - r_0, \alpha + r_0)$ tal que $r_0 > 0$ y $g(\alpha) = \alpha$. Entonces si*

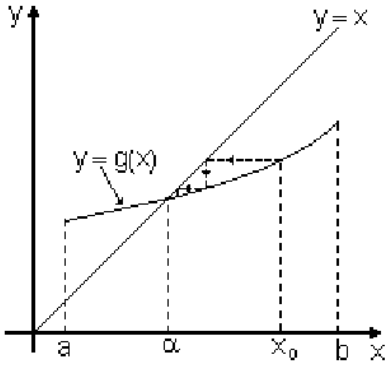
$$|g'(\alpha)| < 1,$$

existe $r < r_0$ tal que para $x_0 \in [\alpha - r, \alpha + r]$ la sucesión $x_{n+1} = g(x_n), n = 0, 1, ..$ está en $[\alpha - r, \alpha + r]$ y converge hacia α. Además, si los errores $e_n = x_n - \alpha \neq 0$ sus cocientes $\frac{e_{n+1}}{e_n} \to g'(\alpha)$ cuando $n \to \infty$.

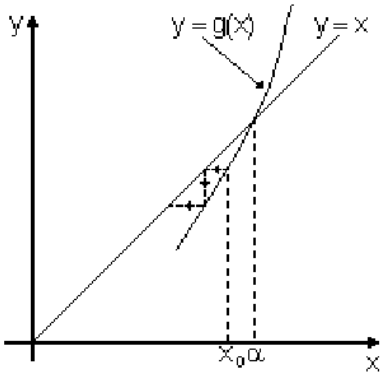
A continuación se presentan los siguientes casos:



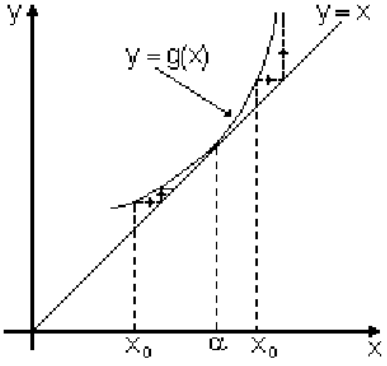
Convergencia. (La sucesión no es monótona)



Convergencia. (La sucesión es monótona)



Divergencia. No satisface las hipótesis del teorema de Punto Fijo.



Convergencia (dependiendo del punto inicial). No satisface las hipótesis del teorema de Punto Fijo.

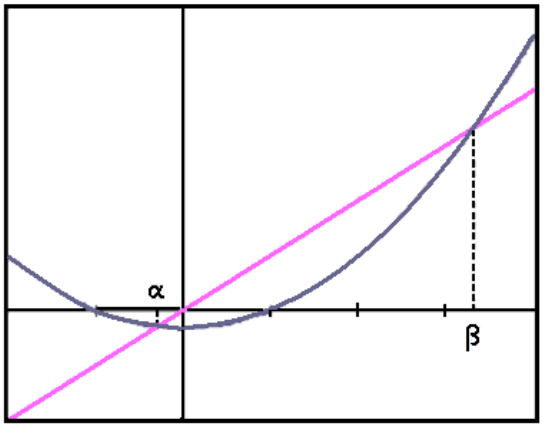
Ejercicio: Se considera la función $f(x) = \dfrac{x^2 - 1}{3}$.

a) Obtener, gráfica y analíticamente, los puntos fijos de f.

b) Demostrar que f verifica las dos hipótesis del teorema de convergencia global de punto fijo en [-1 , 1] y obtener tres términos de la sucesión generada por el algoritmo de punto fijo $x_{n+1} = f(x_n)$ con $x_0 = 0$. ¿ Cuál es el orden de convergencia ?.

Solución:

a) Los puntos fijos de f son los valores de x que verifican $f(x) = x$. Para ello se observa dónde se intersecan las gráficas de $y = f(x)$ e $y = x$.



Gráficamente vemos que hay dos puntos fijos de f : $\alpha \in (-1,0)$ y $\beta \in (3,4)$.
Vamos a comprobar lo anterior de manera analítica:

$$f(x) = x \Leftrightarrow \frac{x^2-1}{3} = x \Leftrightarrow x^2 - 1 = 3x \Leftrightarrow x^2 - 3x - 1 = 0$$

Resolviendo la ecuación de segundo grado se obtienen los valores de $\alpha = -0.3027$ y $\beta = 3.3027$.

b) f contractiva en $[-1,1] \Leftrightarrow \exists\ L \in [0,1)$ / $|f'(x)| \le L < 1 \quad \forall x \in (-1,1)$

$$|f'(x)| = \left|\frac{2}{3}x\right| = \frac{2}{3}|x| < \frac{2}{3} \quad \forall x \in (-1,1) \qquad \text{ya que} \quad \sup_{x\in(-1,1)} |f'(x)| = \frac{2}{3}$$

Nota aclaratoria: Si f' es monótona en (a, b), en este caso estrictamente creciente, entonces $|f'(x)|$ alcanza el supremo en uno de los extremos del intervalo.

¿ Se cumple $f : [-1,1] \to [-1,1]$?

$$f'(x) = 0 \Leftrightarrow \frac{2}{3}x = 0 \Leftrightarrow x = 0 \in (-1,1)$$

$$f(-1) = 0 = f(1) \text{ y } f(0) = -1/3 \Rightarrow \max_{x\in[-1,1]} f(x) = 0 \text{ y } \min_{x\in[-1,1]} f(x) = -1/3$$

Por tanto $f : [-1,1] \to [-1/3, 0] \subset [-1,1]$

$\forall x_0 \in [-1,1]$ la sucesión $\{x_n\}$ / $x_{n+1} = f(x_n)$, $n \ge 0$, converge al único punto fijo de f en $[-1,1]$, es decir, converge a α

$$x_0 = 0 \quad ; \qquad x_1 = f(0) = -\frac{1}{3} \; ; \qquad x_2 = f(-1/3) = \frac{\frac{1}{9} - 1}{3} = -\frac{8}{27}$$

$$x_3 = f(-8/27) = \frac{\frac{64}{729} - 1}{3} = -\frac{665}{2187} \approx -0.304069$$

$f'(x) = \frac{2}{3}x \Rightarrow f'(\alpha) \neq 0$. La convergencia es por tanto lineal (orden de convergencia 1).

Ejercicio:

4) Se considera la función $F(x) = e^x - 3x$

a) Demostrar gráfica y analíticamente que la ecuación $F(x) = 0$ tiene exactamente dos raíces reales y encontrar dos intervalos, cada uno de ellos con extremos enteros consecutivos, que las contengan.

b) Se quiere aproximar la raíz de la ecuación $F(x) = 0$ que pertenece a [0 ,1] usando el método de iteración de punto fijo con diferentes funciones:

$$f_1(x) = \frac{e^x}{3} \quad , \quad f_2(x) = \frac{e^x - x}{2} \quad , \quad f_3(x) = e^x - 3x$$

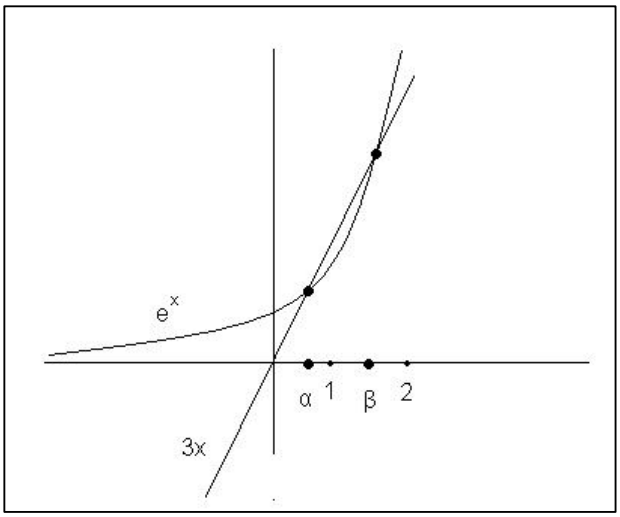
Estudiar cuáles de estas funciones son útiles para aproximar dicha raíz.

¿ Con cuál de ellas son necesarias menos iteraciones para que la cota de error sea menor que una cantidad prefijada ε, partiendo del mismo valor inicial?. Comprobarlo numéricamente si $x_0 = 0.6$ y $\varepsilon = 10^{-6}$.

c) Estudiar cuáles de las funciones anteriores son útiles para aproximar la otra raíz.

Solución:

a) $F(x) = 0 \Leftrightarrow e^x = 3x$



Gráficamente vemos que F tiene dos raíces reales $\alpha \in (0,1)$ y $\beta \in (1,2)$

Analíticamente:

$F'(x) = e^x - 3 = 0 \Leftrightarrow e^x = 3 \Leftrightarrow x = \log(3) \cong 1.098$, es decir, que F es estrictamente monótona en $(-\infty, \log(3))$ y por tanto la ecuación $F(x) = 0$ tiene a lo sumo una raíz en dicho intervalo. Análogamente, F es estrictamente monótona en $(\log(3), +\infty)$ y por tanto la ecuación $F(x) = 0$ tiene a lo sumo, una raíz en dicho intervalo. Hemos demostrado que la ecuación $F(x) = 0$ tiene, a lo sumo, dos raíces reales.

$\left.\begin{array}{l} F(0) = e^0 - 0 > 0 \\ F(1) = e - 3 < 0 \\ F \quad Continua \end{array}\right\} \Rightarrow$ la ecuación $F(x) = 0$ tiene, al menos, una raíz en el intervalo (0,1).

$$\left.\begin{array}{l} F(1) = e - 3 < 0 \\ F(2) = e^2 - 6 > 0 \\ F \quad \textit{Continua} \end{array}\right\} \Rightarrow \text{la ecuación } F(x) = 0 \text{ tiene, al menos, una raíz en el intervalo (1,2).}$$

En consecuencia, la ecuación $F(x) = 0$ tiene exactamente dos raíces reales:

$$\alpha \in [0,1] \qquad \text{y} \qquad \beta \in [1,2]$$

b) Lo primero que habría que comprobar es que f_1, f_2 y f_3 son funciones de iteración asociadas a F

$$f_1(x) = \frac{e^x}{3} = x \Leftrightarrow e^x = 3x \Leftrightarrow F(x) = e^x - 3x = 0$$

$$f_2(x) = \frac{e^x - x}{2} = x \Leftrightarrow e^x - x = 2x \Leftrightarrow F(x) = e^x - 3x = 0$$

$$f_3(x) = e^x - 3x = x \Leftrightarrow e^x - 4x = 0 \neq F(x)$$

f_3 no es función de iteración; f_1 y f_2 sí lo son.

Todo punto fijo de f_1 (análogamente de f_2) es raíz de la ecuación $F(x) = 0$. Además, en este caso, se verifica el recíproco, es decir, que toda raíz de la ecuación $F(x) = 0$, es punto fijo de f_1 y f_2.

f_3 no es útil; veamos si lo son f_1 y f_2 para aproximar la raíz $\alpha \in [0,1]$, es decir, veamos si son contractivas en algún intervalo que contenga a α.

$$f_1 \text{ es contractiva en } [0,1] \Leftrightarrow \exists\, L \in [0,1) \;/\; \left|f_1'(x)\right| \leq L < 1 \qquad \forall x \in (0,1)$$

$$\left|f'_1(x)\right| = \left|\frac{e^x}{3}\right| = \frac{e^x}{3} < \frac{e}{3} < 1 \qquad \forall x \in (0,1) \text{ ; nótese que } \frac{e^x}{3} \text{ es una función creciente y por tanto}$$

$$\sup_{x \in (0,1)} \frac{e^x}{3} = \frac{e}{3} \cong 0.906$$

f_1 es contractiva en $[0,1]$; por tanto f_1 es útil

$$f_2 \text{ es contractiva en } [0,1] \Leftrightarrow \exists\, L \in [0,1) \;/\; \left|f_2'(x)\right| \leq L < 1 \qquad \forall x \in (0,1)$$

$\forall\, x \in (0,1)$, $\quad |f'_2(x)| = \left|\frac{e^x - 1}{2}\right| = \frac{e^x - 1}{2}$ función creciente, ya que

$\left(\frac{e^x - 1}{2}\right)' = \frac{e^x}{2} > 0 \ \forall\ x$. Por tanto: $\sup_{x\in(0,1)} \frac{e^x - 1}{2} = \frac{e-1}{2} < 1$ es decir:

$$|f'_2(x)| = \frac{e^x - 1}{2} < \frac{e-1}{2} < 1$$

$$\Downarrow$$

$$L \cong 0.859 < 1$$

f_2 es contractiva en $[0,1]$: f_2 es útil.

¿ Con cuál de ellas son necesarias menos iteraciones, para que la cota del error sea menor que ε, partiendo del mismo valor inicial?

Con f_2 ya que la L correspondiente a esta función, está más próxima a cero, que la L correspondiente a f_1.

$$|x_n - \alpha| \le \frac{L^n}{1-L}|x_1 - x_0| < \varepsilon \rightarrow \quad n \cdot \log L - \log(1-L) + \log|x_1 - x_0| < \log\varepsilon$$

$$n > \frac{\log\varepsilon + \log(1-L) - \log|x_1 - x_0|}{\log L} \qquad \varepsilon = 10^{-6} \qquad x_0 = 0.6$$

$$f_1 \rightarrow x_1 = f_1(0.6) = \frac{e^{0.6}}{3} = 0.60737$$

$$n > \frac{\log 10^{-6} + \log(1 - \frac{e}{3}) - \log(\frac{e^{0.6}}{3} - 0.6)}{\log\frac{e}{3}} = 114.296 \qquad \textbf{n = 115}$$

$$f_2 \rightarrow x_1 = f_2(0.6) = \frac{e^{0.6} - 0.6}{2} = 0.6110$$

$$n > \frac{\log 10^{-6} + \log(1 - \frac{e-1}{2}) - \log(\frac{e^{0.6} - 0.6}{2} - 0.6)}{\log\frac{e-1}{2}} = 74.238 \qquad \textbf{n = 75}$$

c)

f_3 ya está descartada: veamos si f_1 y f_2 son contractivas en algún intervalo que contenga a β. Sabemos que $\beta \in [1,2]$

f_1 es contractiva en $[1,2] \Leftrightarrow \exists \ \ L \in [0,1) / |f_1'(x)| \leq L < 1 \qquad \forall x \in (1,2)$

$|f_1'(x)| = \left|\frac{e^x}{3}\right| = \frac{e^x}{3}$ Función creciente; $\sup_{x\in(1,2)} \frac{e^x}{3} = \frac{e^2}{3} = 2{,}46 > 1$

Por tanto f_1 no es contractiva en [1,2]. ¿Lo es en algún intervalo que contenga a β ?

$$\left.\begin{array}{l} F(1) = e - 3 < 0 \\ F(1.5) = e^{1.5} - 4.5 < 0 \\ F(2) = e^2 - 6 > 0 \end{array}\right\} \Rightarrow \beta \in [1.5,2]$$

$$\inf_{x\in(1.5,2)} \frac{e^x}{3} = \frac{e^{1.5}}{3} = 1.49 > 1$$

Por tanto $|f_1'(x)| > 1 \qquad \forall x \in (1.5,2)$. Así pues, no existe ningún intervalo que contenga a β donde f_1 sea contractiva. f_1 no es útil.

$\forall x \in (1,2)$, $|f_2'(x)| = \left|\frac{e^x - 1}{2}\right| = \frac{e^x - 1}{2}$ es una función creciente

$$\sup_{x\in(1,2)} \frac{e^x - 1}{2} = \frac{e^2 - 1}{2} = 3.19 > 1$$

Por tanto, f_2 no es contractiva en [1,2],¿ Lo es en algún intervalo que contenga a β ?

$\beta \in [1.5,\ 2]$, $\inf_{x\in(1.5,2)} \frac{e^x - 1}{2} = \frac{e^{1.5} - 1}{2} = 1.74 > 1$

Por tanto $|f'_2(x)| > 1 \qquad \forall x \in (1.5,2)$. Así pues, no existe ningún intervalo que contenga a β donde f_2 sea contractiva. f_2 no es útil.

Ejercicio: Se considera la función $F(x) = x^3 - x - 1$.

a) Determinar, gráfica y analíticamente, el número de raíces reales de la ecuación $F(x) = 0$ y separarlas en intervalos de longitud uno.
b) Determinar una función f de iteración y un intervalo [a, a +1], con a$\in$ Z, que satisfagan las hipótesis del teorema de convergencia global de punto fijo para aproximar una raíz de la ecuación $F(x) = 0$. ¿Cuántas iteraciones son suficientes realizar para garantizar que el error es menor que 10^{-3} si x_0 es cualquier punto del intervalo de convergencia?

Solución

a) $F(x) = 0 \Leftrightarrow x^3 = x + 1$

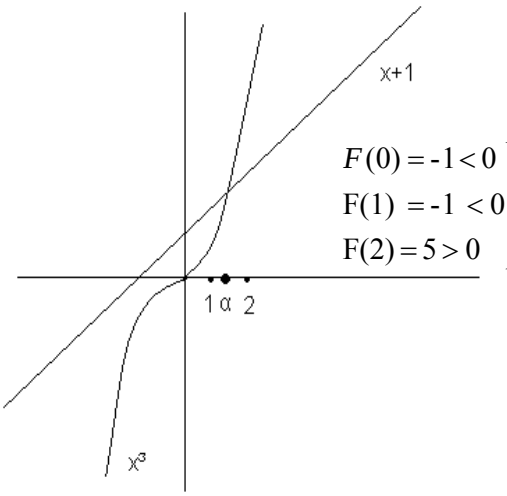
Aparentemente, por la gráfica la ecuación $F(x) = 0$ tiene una única raíz real $\alpha \in [1,2]$.

Analíticamente:

$$F'(x) = 3x^2 - 1 \Leftrightarrow x = \pm\frac{1}{\sqrt{3}}$$

Así pues, la ecuación $F(x) = 0$ tiene, a lo sumo, una raíz real en cada uno de los intervalos :

$$\left(-\infty, -\frac{1}{\sqrt{3}}\right), \left(-\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right) \text{ y } \left(\frac{1}{\sqrt{3}}, \infty\right)$$



$$\left.\begin{array}{l} F(0) = -1 < 0 \\ F(1) = -1 < 0 \\ F(2) = 5 > 0 \end{array}\right\} \Rightarrow \text{La ecuación F(x) = 0 tiene, al menos, una raíz real en (1,2)}$$

Como $(1,2) \subset \left(\frac{1}{\sqrt{3}}, +\infty\right)$ se concluye que la ecuación $F(x) = 0$ tiene exactamente una raíz real en $\left(\frac{1}{\sqrt{3}}, +\infty\right)$ que además sabemos que está en $(1,2)$.

$$\left.\begin{array}{l} F\left(\frac{-1}{\sqrt{3}}\right) = \frac{-1}{3\sqrt{3}} + \frac{1}{\sqrt{3}} - 1 < 0 \\ F\left(\frac{1}{\sqrt{3}}\right) = \frac{1}{3\sqrt{3}} - \frac{1}{\sqrt{3}} - 1 < 0 \\ \text{F es estrictamente monótona en } \left(-1/\sqrt{3}, 1/\sqrt{3}\right) \end{array}\right\} \Rightarrow \textit{La} \text{ ecuación F(x) = 0 no tiene raíces en el intervalo} \left(\frac{-1}{\sqrt{3}}, \frac{1}{\sqrt{3}}\right)$$

Se deduce, de lo anterior, que tampoco tiene raíz en el intervalo $\left(-\infty, -\frac{1}{\sqrt{3}}\right)$ ya que un polinomio de grado tres no puede tener dos raíces reales distintas sin tener una tercera.
Conclusión: Tiene una única raíz real α y está en $(1,2)$.

b) $F(x) = 0 \Leftrightarrow x = x^3 - 1$

Sea $f(x) = x^3 - 1$; $f'(x) = 3x^2$

$|f'(x)| = |3x^2| = 3x^2 > 1 \quad \forall x \in (1,2)$. Por tanto, f no es contractiva en ningún intervalo que contenga a α. No nos vale.

$F(x) = 0 \Leftrightarrow x^3 = x + 1 \Leftrightarrow x = \sqrt[3]{x+1}$

Sea $f(x) = (x+1)^{\frac{1}{3}}, \quad x \in [1,2]$

f es contractiva en $[1,2] \Leftrightarrow \exists L \in [0,1) / |f'(x)| \le L < 1 \quad \forall x \in (1,2)$

$$|f'(x)| = \left|\frac{1}{3}(x+1)^{\frac{-2}{3}}\right| = \left|\frac{1}{3\sqrt[3]{(x+1)^2}}\right| = \frac{1}{3\sqrt[3]{(x+1)^2}} = f'(x)$$

$$f''(x) = \frac{-2}{9}(x+1)^{\frac{-5}{3}} = \frac{-2}{9}\frac{1}{\sqrt[3]{(x+1)^5}} < 0 \qquad \forall x \in (1,2) \;\Rightarrow\; f'(x) \text{ es estrictamente}$$

decreciente en $(1,2)$;

por tanto $\sup_{x\in(1,2)} |f'(x)| = \sup_{x\in(1,2)} f'(x) = f'(1) = \frac{1}{3\sqrt[3]{4}} \approx 0.2099$, es decir:

$$|f'(x)| < \frac{1}{3\sqrt[3]{4}} < 1 \qquad \forall x \in (1,2)$$

f es contractiva en $[1,2]$

¿ Se cumple que $f : [1,2] \to [1,2]$?

$$\left.\begin{array}{l} f'(x) = \dfrac{1}{3\sqrt[3]{(x+1)^2}} > 0 \;\Rightarrow f \text{ es es estrictamente creciente} \\ \min\limits_{x\in[1,2]} f(x) = f(1) = \sqrt[3]{2} \approx 1.2599 \in [1,2] \\ \max\limits_{x\in[1,2]} f(x) = f(2) = \sqrt[3]{3} \approx 1.4422 \in [1,2] \end{array}\right\} \Rightarrow f : [1,2] \to [1,2]$$

Por tanto, $\forall x_0 \in [1,2]$ la sucesión $\{x_n\} / x_{n+1} = (x_n + 1)^{1/3}$ $n \geq 0$ converge al único punto fijo de $f(x) = (x+1)^{1/3}$ en $[1,2]$, es decir, converge a la raíz de la ecuación $F(x) = 0$.

$$|x_n - \alpha| \leq \frac{L^n}{1-L}|x_1 - x_0| < \frac{L^n}{1-L} < 10^{-3}$$

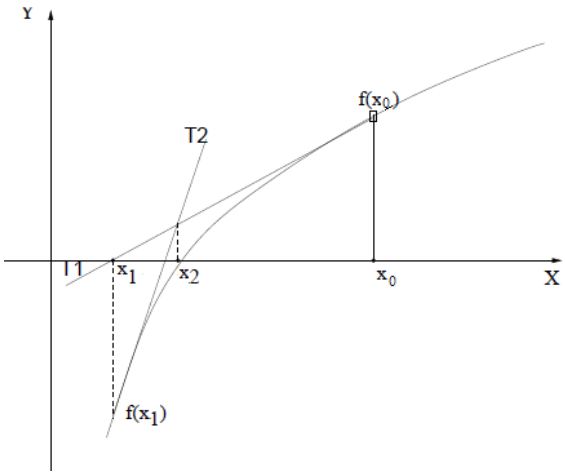
$$\left.\begin{array}{l} x_0 \in [2,3] \\ x_1 \in [2,3] \end{array}\right\} \Rightarrow |x_1 - x_0| \leq 3 - 2 = 1$$

$$n \log L - \log(1-L) < \log 10^{-3}; \qquad n > \frac{\log 10^{-3} + \log(1-L)}{\log L}$$

$$L = \frac{1}{3\sqrt[3]{4}} \approx 0.21 \quad ; \quad n > \frac{\log 10^{-3} + \log(1-0.21)}{\log 0.21} \approx 4.577 \quad ; \quad \mathbf{n = 5}$$

METODO DEL NEWTON-RAPHSON

Uno de los métodos más utilizados para resolver ecuaciones es el de Newton.*[i] Así como los métodos anteriores, éste también se basa en una aproximación lineal de la función, aunque aplicando una tangente a la curva. A partir de una estimación inicial simple, $x_{()}$, que no esté muy lejos de una raíz, se efectúa un desplazamiento a lo largo de la tangente hacia su intersección con el eje *x, y* se toma esta como la siguiente aproximación. Esto continúa hasta que valores x sucesivos están suficientemente próximos o el valor de la función está suficientemente cerca de cero. **



El esquema de cálculo se concluye de inmediato a partir del triángulo rectángulo que se muestra en la figura 1.3, que al igual que uno de sus ángulos agudos tiene el ángulo de inclinación de la recta tangente a la curva en $x = x_0$

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

Se continúa el esquema de cálculo al estimar

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

o , en términos más generales,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)} \qquad n = 0,1,2,...$$

---

*Newton no publicó un análisis extenso de este método, aunque resolvió un polinomio cúbico en Principio (1687). La versión que se proporciona aquí está considerablemente mejorada con respecto a su ejemplo original.

** La elección del criterio que debe aplicarse suele depender del problema físico particular al que se le aplica la ecuación. Por lo común se requiere coincidencias de valores sucesivos *x* a una tolerancia específica.

El algoritmo de Newton se emplea ampliamente porque, al menos en la vecindad próxima de una raíz, converge más pronto que cualquiera de los métodos analizados hasta ahora. En otra sección se demuestra que el método es cuadráticamente convergente, lo cual significa que el error de cada paso tiende a una constante *K* multiplicada por el cuadrado del error del paso anterior. El resultado neto de esto es que el número de cifras decimales de exactitud casi se duplica en cada iteración. No obstante, la compensación de esto es la necesidad de dos evaluaciones funcionales en cada paso $f(x_n)$ y $f'(x_n)$.*

Cuando el método de Newton se aplica a $f(x) = 3x + \text{sen } x - e^x = 0$, se tienen los cálculos siguientes:

$$f(x) = 3x + \text{sen } x - e^x,$$
$$f'(x) = 3 + \cos x - e^x.$$

Hay poca necesidad de usar MATLAB para obtener esta simple derivada pero, como práctica, a continuación se muestra cómo hacerlo:

1. Se define $f(x)$: fx = '3 * x + sen(x) − exp(x)';
2. Se obtiene la derivada: dfx = diff(fx)
   y se observa:

```
dfx =
3 + cos(x) - exp(x)
```

Obsérvese que para representar e^x se usa exp(x) y que la variable de diferenciación no es necesaria cuando es el valor por defecto *x*.
Si se empieza con $x_0 = 0.0$, se tiene

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)} = 0.0 - \frac{-1.0}{3.0} = 0.33333;$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} = 0.33333 - \frac{-0.068418}{2.54934} = 0.36017;$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)} = 0.36017 - \frac{-6.279 \times 10^{-4}}{2.50226} = 0.3604217.$$

Después de tres iteraciones, la raíz es correcta hasta siete dígitos significativos. Al comparar esto con los métodos anteriores se observa que el método de Newton converge considerablemente más pronto. No obstante, al comparar métodos numéricos se acostumbra contar el número de veces que es necesario evaluar las funciones. Debido a que el método de Newton requiere dos evaluaciones funcionales por paso, la comparación no está tan unilateralmente a favor del método de Newton como parece al principio; las tres iteraciones con este método requirieron seis evaluaciones funcionales. Cinco iteraciones con los métodos anteriores también requirieron seis evaluaciones. Si un problema difícil necesita muchas iteraciones para converger, el número de evaluaciones funcionales con el método de Newton puede ser mucho mayor que con los métodos de iteración lineal porque Newton siempre usa dos por iteración, mientras que los otros sólo utilizan una (después del primer paso aquél usa dos evaluaciones).

---

* Otro problema con el método de Newton es que puede ser difícil encontrar $f'(x)$. Los sistemas de álgebra por computadora pueden ser una verdadera ayuda.

Teorema: Sea $f \in C^2([a,b])$. Si $s \in [a,b]$ es tal que $f(s) = 0$ y $f'(s) \neq 0$. Entonces $\exists\ \delta > 0$ tal que $\forall\ s_0 \in [s-\delta, s+\delta]$ la sucesión de Newton -Raphson , $s_{n+1} = s_n - \frac{f(s_n)}{f'(s_n)}$, está bien definida y converge a s.

Nota: La sucesión de Newton-Raphson no siempre tiene que converger. Puede que incluso no esté definida (que sucederá si para algún n, $s_n \notin [a,b]$ ó $f'(s_n) = 0$).

Teorema: Sea $[a,b]$ y $f \in C^2[a,b]$, verificando:

i) $f(a)f(b) < 0$

ii) $\forall\ x \in [a,b]\ f'(x) \neq 0$

iii) $\forall\ x \in [a,b]\ f''(x) \geq 0$ ó $\forall\ x \in [a,b]\ f''(x) \leq 0$

Entonces f tiene una única raíz s en $[a,b]$ y si $s_0 \in [a,b]$ verifica que $f(s_0)f''(s_0) > 0$, la sucesión de Newton converge a s.

A continuación se presenta un planteamiento más formal del algoritmo del método de Newton, idóneo para implementarse en un programa para computadora.

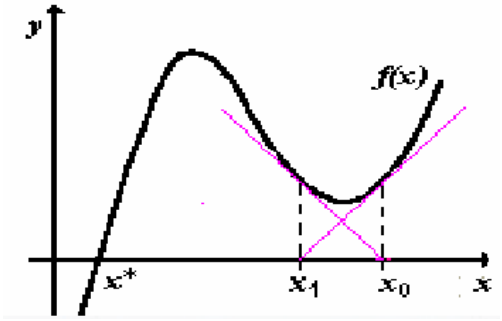
ALGORITMO DE NEWTON RAPHSON

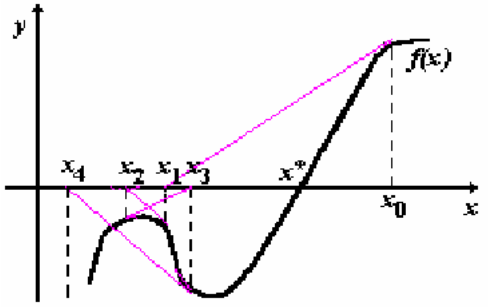
Dada *f(x)* y una aproximación inicial p_0
Entrada: aproximación inicial p_0; tolerancia TOL; máximo número de iteraciones N_0
Salida: Solución aproximada p o mensaje de falla

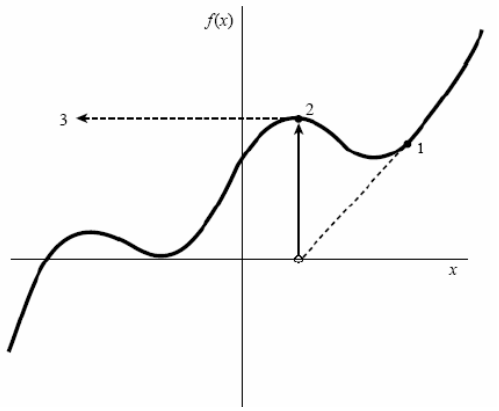
Paso 1: i = 1
Paso 2: Mientras i ≤ N0 hacer los pasos 3 – 6
Paso 3: p = p0 – f(p0) / f '(p0) //Calcular Pi
Paso 4: Si |p - p0| < TOL entonces
Salida = p
Terminar //Procedimiento terminado satisfactoriamente
Paso 5: i = i + 1
Paso 6: p0 = p //Actualizar p0
Paso 7: Salida = 'El método fallo luego de N0 iteraciones'
Terminar //Procedimiento terminado NO satisfactoriamente

Cuando el método de Newton se aplica a funciones polinomiales, hay técnicas especiales que facilitan dicha aplicación. En algunos casos el método de Newton no converge. Empezando con x_0 , nunca se alcanza la raíz r porque $x_6 — x_1$ y se cae en un ciclo sin fin. También observe que si alguna vez ha de alcanzarse el mínimo o el máximo de la curva, se disparará hasta el infinito.

Casos en el cual el método diverge:







Raíces complejas

El método de Newton funciona con raíces complejas si se proporciona un valor complejo para el valor inicial. A continuación se presenta un ejemplo.

Ejercicio: Use el método de Newton en f(x) = $x^3 + 2x^2 - x + 5$.
En la figura 1 se muestra la gráfica de f(x). Tiene una raíz real aproximadamente en $x = -3$, mientras las otras dos raíces son complejas porque ya no vuelve a cruzarse el eje *x*.
Si el método de Newton se inicia con $x_0 = 1 + i$ (se usa esto a falta de conocimiento sobre la raíz compleja) se obtienen las iteraciones sucesivas:

1. 0.486238 + 1.04587i
2. 0.448139 + 1.23665i
3. 0.462720 + 1.22242i
4. 0.462925 + 1.22253i
5. 0.462925 + 1.22253i

Debido a que las iteraciones cuarta y quinta coinciden hasta seis cifras significativas, se tiene la certeza de contar con una estimación aceptable.

Si se empieza con un valor inicial real, por ejemplo $x_0 = -3$, se obtiene convergencia
en la raíz en $x = -2.92585$.

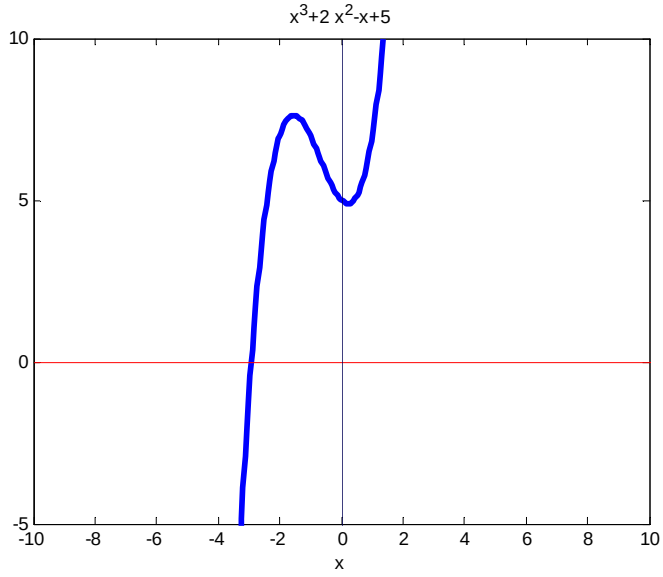


figura 1 Gráfico de f(x)

Ejercicio: Considera la función $F(x) = x + \ln(x)$. Aproximar la raíz α mediante el método de Newton eligiendo como punto inicial $x_0 = a$ y realizando dos iteraciones.

Solución

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

$$x_0 = a = \frac{1}{2}\ ;\ x_1 = \frac{\frac{1}{2}(1 - \ln\frac{1}{2})}{\frac{3}{2}} = \frac{1}{3}(1 + \ln 2) = 0.564382$$

$$x_2 = \frac{x_1(1 - \ln x_1)}{x_1 + 1} = 0.5671$$

Definición de orden de convergencia

Definición: Si $\lim_{n\to\infty} s_n = s$, se dice que (s_n) tiene orden de convergencia $\alpha > 0$ si:

$$\lim_{n\to\infty} \frac{|\ s_{n+1} - s\ |}{|\ s_n - s\ |^{\alpha}} = \lambda \neq 0$$

Si $\alpha = 1$, la convegencia es lineal.

Si $\alpha = 2$, la convegencia es cuadrática.

Teorema: la sucesión de Newton-Raphson tiene orden de convergencia cuadrática.

Teorema: Sea $f \in C^2[a,b]$, $s \in [a,b]$ una raíz de f tal que $f'(s) \neq 0$ y $f''(s) \neq 0$. Supongamos que existen m y M tales que $\forall x \in [a,b]\ |\ f'(x)\ | \geq m$ y $|\ f''(x)\ | \leq M$. Entonces si (s_n) es la sucesión de Newton-Raphson y converge a s se tiene

$$|\ s_{n+1} - s\ | \leq |\ s_{n+1} - s_n\ |^2\ \frac{M}{m}$$

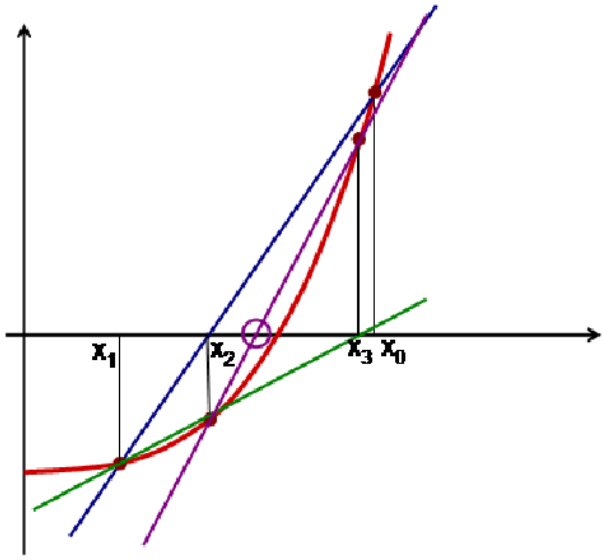
METODO DE LA SECANTE

Otra desventaja del método anterior es que se necesita saber la derivada de la función, esta operación puede ser superada con este método ya que considera una aproximación de la derivada a partir de la función sin necesidad de derivar.
Considerando una aproximación de la derivada como:

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

y reemplazando la ecuación en la fórmula recursiva de newton-raphson

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



Interpretación gráfica.
La desventaja de este método converge más lentamente que newton-raphson. Una modificación a este método es considerar un punto bastante cercano que difiera un h≈0 y calcular una aproximación de la derivada, obteniendo la siguiente relación:

$$f'(x) = \frac{f(x) - f(x-h)}{h}$$

Para efectos de la aproximación se puede usar un h=10^-3

Por lo que la fórmula de Newton-Raphson queda modificada a:

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

SISTEMAS DE ECUACIONES NO LINEALES

INTRODUCCION

Consideremos ahora el problema de resolver un sistema de ecuaciones no lineales de n ecuaciones con *n* variables. Sea $f_i : R^n \to R$, $1 \le i \le n$. funciones (no lineales) todas diferenciables.

Un sistema no lineal $n \times n$ se puede escribir de la forma:

$$\begin{cases} f_1(x_1, x_2, \ldots, x_n) = 0, \\ f_2(x_1, x_2, \ldots, x_n) = 0, \\ \vdots \\ f_n(x_1, x_2, \ldots, x_n) = 0, \end{cases} \qquad (1)$$

Si definimos F: $R^n \to R^n$ por $F=(f_1,f_2,\ldots,f_n)^t$ entonces podemos escribir (1) en forma vectorial como:

$$F(\mathbf{x})=0, \quad \mathbf{x}=(x_1,x_2,\ldots,x_n) \qquad (2)$$

Este sistema puede presentar múltiples soluciones matemáticamente posibles y su resolución numérica debe proporcionar la solución físicamente correcta. Existen distintos métodos para la solución.

EL METODO DEL PUNTO FIJO

Análogamente al caso unidimensional, el método iterativo del punto fijo se base en la posibilidad de escribir el sistema de ecuaciones $F(x)=0$ en otro equivalente de la forma

$$x_k = G(x_{k-1})$$

Donde $G : \mathbf{R}^n \to \mathbf{R}^n$,

Que es equivalente a:

$$\begin{cases} x_1 = g_1(x_1, x_2, \ldots, x_n) \\ x_2 = g_2(x_1, x_2, \ldots, x_n) \\ \vdots \\ x_n = g_n(x_1, x_2, \ldots, x_n) \end{cases}$$

Donde $g_1, g_2, \ldots, g_n$ son los componentes de G

Consiste entonces en generar una sucesión de puntos en $\mathbf{R}^n$ por medio de la relación de recurrencia

$$x^{(k)} = G(x^{(k-1)}), \qquad k = 1,2,\ldots,$$

a partir de un punto inicial $\boldsymbol{x}^{(0)}$. Se pretende que esta sucesión de puntos en $\mathbf{R}^n$ converja para un punto fijo *s* de la función ***G***, esto es, tal que *s* = ***G***(s) que será por tanto solución del sistema original, o sea, tal que ***F***(s)=**0.**

Para ello se desarrolla el siguiente algoritmo:

```
function x=senl_pfijo(G,x0,tol,max)
% Resuelve el sistema no lineal x=G(x) usando la iteración
del punto fijo
% Los vectores x y x0 son vectores fila
% la funcion G da un vector columna [g1(x)...gn(x)]'
% detener cuando la norma del cambio en el vector solución
sea menor a la tolerancia
% la siguiente aproximacion de solucion es x_new=x_old+y';
disp([0 x0]);
x_old=x0;
iter=1;
while (iter<=max)
y=feval(G,x_old)
x_new=y';
dif=norm(x_new-x_old);
disp([iter x_new dif]);
if dif<=tol
x=x_new;
disp('La iteración del punto fijo ha convergido')
return;
else
x_old=x_new;
end
iter=iter+1;
end
disp('La iteración del punto fijo no converge')
x=x_new;
```

Ejercicio 1:

$$f(x_1, x_2) = x_1^3 + 10x_1 - x_2 - 5 = 0$$
$$f(x_1, x_2) = x_1 + x_2^3 - 10x_2 + 1 = 0$$

Solución:

Lo reescribimos de la forma $x_1 = g_1(x_1, x_2)$ *y* $x_2 = g_2(x_1, x_2)$ de la siguiente manera:

$$x_1 = -0.1x_1^3 + 0.1x_2 + 0.5$$
$$x_2 = 0.1x_1 + 0.1x_2^3 + 0.1$$

para ello elaboramos una función en Matlab:

```
function G=fun2(x)
x1=x(1);
x2=x(2);
G=[(-0.1*x1^3+0.1*x2+0.5)
   0.1*x1+0.1*x2^3+0.1];
```

Y finalmente ejecutamos los comandos:

>> x= senl_pfijo('fun2',[1 1],0.001,10)

La iteración del punto fijo converge

x =

0.50235294667355 0.15060128678661

>> fun2(x)

ans =

0.50238282592176 0.15057686964446

Como puede verse se tiene una precisión de 3 dígitos exactos

Ejercicio 2:

Considere el sistema no lineal

$$3x_1 - \cos(x_2 x_3) - \frac{1}{2} = 0$$

$$x_1^2 - 81(x_2 + 0.1)^2 + \sin(x_3) + 1.06 = 0$$

$$e^{-x_1 x_2} + 20x_3 + \frac{10\pi - 3}{3} = 0$$

Aplique el método del punto fijo para aproximar la solución, realice 5 iteraciones y escoger $\boldsymbol{x}^{(0)}=(0.1, 0.1, -0.1)^t$

Solución

Si de la *i*-ésima ecuación se despeja x_i, el sistema puede cambiarse a un problema de punto fijo

$$x_1 = \frac{1}{3}\cos(x_2 x_3) + \frac{1}{6}$$

$$x_2 = \frac{1}{9}\sqrt{x_1^2 + \sin(x_3) + 1.06} - 0.1$$

$$x_3 = -\frac{1}{20}e^{-x_1 x_2} - \frac{10\pi - 3}{60}$$

se obtiene la siguiente expresión de recurrencia

$$x_1^{(k)} = \frac{1}{3}\cos(x_2^{(k-1)} x_3^{(k-1)}) + \frac{1}{6}$$

$$x_2^{(k)} = \frac{1}{9}\sqrt{(x_1^{(k-1)})^2 + \sin(x_3^{(k-1)}) + 1.06} - 0.1$$

$$x_3^{(k)} = -\frac{1}{20}e^{-x_1^{(k-1)} x_2^{(k-1)}} - \frac{10\pi - 3}{60}$$

Partiendo de la estimación inicial , $x_1^{(0)} = 0.1, x_2^{(0)} = 0.1, x_3^{(0)} = -0.1$ se obtiene los siguientes resultados

k	$x_1^{(k)}$	$x_2^{(k)}$	$x_3^{(k)}$
0	0.100	0.100000	-0.100
1	0.49998	0.009441	-0.52310
2	0.49999	0.000025	-0.52336
3	0.50000	0.000012	-0.5235981
4	0.50000	0.00000003	-0.5235984
5	0.50000	0.00000002	-0.5235987

EL METODO DE NEWTON RAPHSON

Este método tiene como ventaja una relativa simplicidad y una rápida convergencia pero tiene el inconveniente que su convergencia únicamente se puede asegurar si los valores iniciales de las incógnitas están lo suficientemente cerca de la solución (dominio de atracción de la solución).
Para sistemas mal condicionados el método de Newton puede presentar problemas de convergencia.
El método de Newton para la solución de sistemas de ecuaciones es también una generalización del método ya estudiado para el caso unidimensional. Consideremos nuevamente el sistema de ecuaciones $F(x)=0$, donde:

$$F(x) = \left(f_1(x), f_2(x), \ldots, f_n(x)\right)^t$$

con $f_i : R^n \to R, \quad i = 1,2,\ldots,n$

Definimos la matriz jacobiana de la función F como:

$$J_F(x) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \cdots & \cdots & \cdots & \cdots \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \cdots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

Sea $\boldsymbol{x_0}$ una aproximación inicial al sistema $F(\boldsymbol{x})=0$. Entonces usando el Teorema de Taylor para funciones de varias variables, podemos escribir que

$$F(x) \approx F(x_0) + J_F(x_0)(x - x_0)$$

Definimos ahora la siguiente aproximación $\boldsymbol{x_1}$ como la solución de

$$F(x_0) + J_F(x_0)(x - x_0) = 0$$

es decir

$$x_1 = x_0 - \left(J_F(x_0)\right)^{-1} F(x_0)$$

De esta forma continuamos así la versión para sistemas del *Método de Newton* dada por:

$$\begin{cases} x_{k+1} = x_k - \left(J_F(x_k)\right)^{-1} F(x_k), & k \geq 0 \\ x_0 \ \textit{dado} \end{cases}$$

La implementación del método de Newton para sistemas de ecuaciones no lineales

```
function [x,iter]= senl_nrapson(f,fp,x0,tol,itermax)
%NEWTON Metodo de Newton para sistemas no lineales
% Los datos de entrada son
% f: nombre de la función que representa el sistema.
% fp: nombre de la función que calcula el Jacobiano.
% x0: el punto inicial (vector columna).
% tol: tolerancia para el error relativo  en la solucion
calculada
% itermax: numero maximo de iteraciones que se repiten las
iteraciones
if nargin<4
   tol=1.0e-4;
end
if nargin<5
   itermax=20;
end
x=x0;
normx=0;
normz=inf;
iter=0;
while (normz>tol*normx)&(iter<=itermax)
   f0=feval(f,x);
   fp0=feval(fp,x);
   z=-fp0\f0;
   normz=norm(z,2);
   normx=norm(x,2);
   x=x+z;
   iter=iter+1;
end
```

Esta función se debe invocar con al menos tres argumentos. Si se omite alguno de los últimos dos argumentos, la función tiene unos valores que se asignan por omisión a estas variables.

Ejercicio1: Resuelva el siguiente sistema de ecuaciones:

$$f(x,y,z) = x^3 - 10x + y - z + 3 = 0$$

$$g(x,y,z) = y^3 + 10y - 2x - 2z - 5 = 0$$

$$h(x,y,z) = x + y - 10z + 2sen(z) + 5 = 0$$

Solución:

Para ello construimos el Jacobiano:

$$J(x) = \begin{bmatrix} 3x^2 - 10 & 1 & -1 \\ -2 & 3y^2 + 10 & -2 \\ 1 & 1 & -10 + 2\cos(z) \end{bmatrix}$$

Luego desarrollamos las funciones en Matlab:

```
function G=fun3(x)
G=[x(1)^3-10*x(1)+x(2)-x(3)+3
x(2)^3+10*x(2)-2*x(1)-2*x(3)-5
x(1)+x(2)-10*x(3)+2*sin(x(3))+5];
```

```
function G=jfun3(x)
G=[(3*x(1)^2)-10 1 -1
   -2 (3*x(2)^2)+10 -2
   1 1 -10+2*cos(x(3))];
```

```
iter =
    4

x =
  0.29703611177844            0.67480978718683            0.73065629971593

>> fun3(x)
ans =
  1.0e-013 *
  0.36859404417555            0.00888178419700                 0
```

Se puede observar que se alcanza a la solución con una buena precisión en solo 4 iteraciones.

EL METODO DE LA SECANTE

Como vimos para el caso uni-variado, el método de la secante es similar al de Newton, de hecho, algunos lo llaman el método de Newton-Raphson, solo que utiliza aproximaciones Lineales para el Jacobiano, en vez del Jacobiano.

Para ello consideremos el siguiente algoritmo en Matlab:

```
function x = senl_secante(fun,x0,tol,maxit);
% detener cuando la norma del cambio en el vector solución sea menor a la tolerancia
del = diag(max(abs(x0)*1e-4,1e-8));
n = length(x0);
for i=1:maxit
f = feval(fun,x0);
for j=1:n
J(:,j) = (f-feval(fun,x0-del(:,j)))/del(j,j);
end;
x = x0-inv(J)*f;
if norm(x-x0)<tol
break
end
x0 = x;
end
if i>=maxit
sprintf('Se alcanzaron %iteraciones',maxit)
end
i
```