

PRACTICA Núm. 2

Resolución numérica de ecuaciones no lineales

La resolución de ecuaciones no lineales es un problema que se presenta con mucha frecuencia en los distintos campos científicos y técnicos. No existen, en general, métodos de resolución formal de ecuaciones no lineales, solo para polinomios de grado menor que cinco. Por este motivo hay que desarrollar métodos numéricos para la mayoría de los problemas de búsqueda de soluciones.

Ejemplo: *Mathematica* puede hallar las raíces de cualquier polinomio de grado menor que cinco, por ejemplo

```
Solve[x^3 + 9 x + 2 == 0, x]
N[%]
{{x -> -\frac{3}{(-1 + 2\sqrt{7})^{1/3}} + (-1 + 2\sqrt{7})^{1/3}},
 {x -> \frac{3(1 + i\sqrt{3})}{2(-1 + 2\sqrt{7})^{1/3}} - \frac{1}{2}(1 - i\sqrt{3})(-1 + 2\sqrt{7})^{1/3}},
 {x -> \frac{3(1 - i\sqrt{3})}{2(-1 + 2\sqrt{7})^{1/3}} - \frac{1}{2}(1 + i\sqrt{3})(-1 + 2\sqrt{7})^{1/3}}}
{{x -> -0.221023}, {x -> 0.110511 + 3.0061 I}, {x -> 0.110511 - 3.0061 I}}
```

pero, en general, no resuelve polinomios de grado 5 o superior,

```
Solve[x^5 + 2 x + 2 == 0, x]
{{x -> Root[2 + 2*#1 + #1^5 &, 1]}, {x -> Root[2 + 2*#1 + #1^5 &, 2]},
 {x -> Root[2 + 2*#1 + #1^5 &, 3]}, {x -> Root[2 + 2*#1 + #1^5 &, 4]},
 {x -> Root[2 + 2*#1 + #1^5 &, 5]}}
```

También puede emplearse la orden **Roots[p==0,x]** para hallar las raíces del polinomio p.

Ejercicio: Calcular, si es posible, utilizando las órdenes anteriores, las raíces de los siguientes polinomios:

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

Puesto que los métodos que vamos a ver en esta práctica son iterativos, es necesario establecer condiciones bajo las cuales este proceso iterativo se interrumpa. Estos condicionales vienen dados en *Mathematica* por la sentencia **If[condición, proceso1, proceso2]**. Si **condición** es verdadera, se ejecutarán las órdenes dadas en **proceso1**, y en caso contrario se ejecutarán las de **proceso2**. Si se pretende salir de un bucle, entonces la sentencia que hay que escribir es **Break[]**. Apliquemos las órdenes anteriores para implementar el proceso de bisección.

Metodo de Bisección:

Ejemplo: Hallar la raíz de $f(x) = x^2 - 2$ en el intervalo $[0, 2]$.

Como $f(x)$ es continua en el intervalo dado y $f(0) f(2) < 0$, podemos asegurar que existe una raíz, y como $f(x)$ es diferenciable con derivada continua siendo $f'(x) > 0$ en $[0, 2]$, la raíz es única.

```
Clear[a, b, c, tol, f, x]
a=0; b=2; tol=10.^(-3); (* Datos de entrada *)
f[x_]= x^2-2; (* función f(x) *)
maxiter=Log[(b-a)/tol]/Log[2.]-1 (* máximo número de iteraciones a realizar *)
9.96578
```

```

For[i=1, i<=maxiter+1, i++,          (* comienzo del bucle *)
  c=(a+b)/2.;                        (* definición del punto medio *)
  Print[a," ",c," ",b," ",f[c]];    (* Escribir los datos por pantalla *)
  If[f[c]==0, Break[]];              (* caso improbable pero ... *)
  If[ b-a <tol, Break[] ];          (* si la longitud del intervalo [a,b] es
                                     menor que tol, acabar el proceso *)
  If[f[c]*f[a]<0, b=c, a=c];         (* el if más importante *)
  ];                                  (* fin del bucle *)
c=(a+b)/2;
error = N[c-Sqrt[2]]<tol

0 1. 2 -1.

1. 1.5 2 0.25

1. 1.25 1.5 -0.4375

1.25 1.375 1.5 -0.109375

1.375 1.4375 1.5 0.0664063

1.375 1.40625 1.4375 -0.0224609

1.40625 1.42188 1.4375 0.0217285

1.40625 1.41406 1.42188 -0.000427246

1.41406 1.41797 1.42188 0.0106354

1.41406 1.41602 1.41797 0.00510025

True

```

Puede apreciarse que el error en la salida es menor que la tolerancia pedida.

Ejercicio: Se pretende encontrar una aproximación de la raíz cúbica de 25.

- determinar un intervalo en el cual este la raíz.
- dibujar la gráfica de la función
- del apartado anterior, calcular el número de iteraciones para obtener una aproximación con 2 decimales con bisección.

Ejercicio: Demostrar que la función $f(x)=x^3+e^x-10$ tiene una única raíz en \mathbb{R} .

- determinar un intervalo en el cual este la raíz.
- dibujar la gráfica de la función
- calcular el número de iteraciones para 10 decimales con el algoritmo de bisección.

Método de Newton:

En el método de Newton, dada una aproximación x_n a la raíz del problema, se obtiene la siguiente aproximación x_{n+1} , hallando el punto de corte de la recta tangente a la curva $f(x)$ en el punto

$(x_n, f(x_n))$ con el eje de abscisas. Este proceso iterativo se repite hasta que $|x_{n+1} - x_n| < \text{tol}$ y/o $|f(x_n)| < \text{tol}$.

Se tiene el siguiente método iterativo: para x_0 dado, $x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$, $n = 0, 1, 2, \dots$

Han de hacerse algunas observaciones al método de Newton:

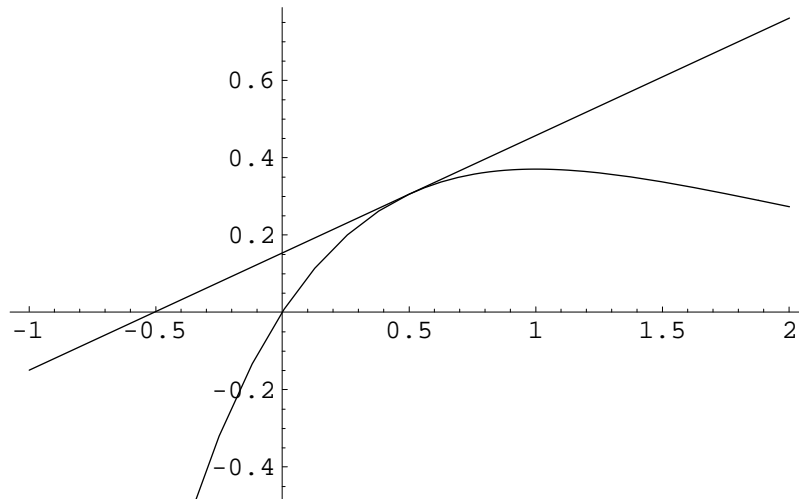
- La función $f(x)$ debe ser suficientemente derivable para poder trazar la recta tangente.
- Si la derivada se anula en algún punto x_n , no podremos continuar ya que aparecería una división por cero.
- el punto inicial debe ser una "buena" aproximación a la raíz exacta

Ejemplo: En este ejemplo, vamos a considerar la función $f(x) = x e^{-x}$ y vamos a calcular partiendo del punto inicial $x_0 = 0.5$ la recta tangente y el punto de corte de la misma con el eje de abscisas

```

Clear[f,x,recta,x0,x1]
f[x_]:=x*Exp[-x] (* función f(x) *)
x0=0.5; (* punto inicial *)
recta = Simplify[f[x0] + (x-x0)*f'[x0]]; (* recta tangente *)
Plot[{recta,f[x]},{x,-1,2}];
x1=Solve[recta==0,x]; (* punto de corte de la tangente *)
Print["el punto de corte de la recta tangente es ", x1]

```



el punto de corte de la recta tangente es $\{x \rightarrow -0.5\}$

Ejercicio: Considerar $f(x) = \cos\left(\frac{5+x^2}{x^2+1}\right)$ y $x_0 = -2$. Dibujar $f(x)$ en el intervalo $[-3,3]$ y hallar gráficamente la primera y segunda iteración del método de Newton.

Ejemplo: En este ejemplo vamos a considerar la función $f(x) = x e^{-x}$ y el punto inicial $x_0 = 2.0$. Se pretende implementar el método de Newton con la condición de parada $|f(x_n)| < \text{tol}$, con $\text{tol} = 10^{-4}$

```

Clear[f,x,recta,x0,x1,tol,maxiter]
f[x_]:=x*Exp[-x] (* función f(x) *)
x0=2; (* punto inicial *)
tol=10^(-4); (* tolerancia exigida *)
maxiter=20; (* máximo número de iteraciones *)
For[i=1,i<=maxiter,i++,
  x1= N[x0 - f[x0]/f'[x0],20];
  Print[i," ", x0, " ", N[f[x0]]];
  If[ Abs[f[x0]]<tol, Print["Convergencia en ",i," iteraciones"];
    Print["la solución es ", x1];Break[] ];
  x0=x1;
];
1 2. 0.270671
2 4. 0.0732626
3 5.33333 0.0257491
4 6.5641 0.00925597
5 7.74383 0.00335625
6 8.89211 0.00122239
7 10.0188 0.000446374
8 11.1297 0.000163274
9 12.2284 0.000059791

Convergencia en 9 iteraciones

la solución es 13.3175

```

puede apreciarse que en este caso el método de Newton lleva a una mala aproximación. Veamos ahora el caso del criterio $|x_{n+1} - x_n| < \text{tol}$.

```

Clear[f,x,recta,x0,x1,tol,maxiter]
f[x_]:=x*Exp[-x] (* función f(x) *)
x0=2; (* punto inicial *)
tol=10^(-4); (* tolerancia exigida *)
maxiter=15; (* máximo número de iteraciones *)
For[i=1,i<=maxiter,i++,
  x1= N[x0 - f[x0]/f'[x0],20];
  Print[i," ", x0 , " ", N[f[x0]]];
  If[ Abs[x1-x0]<tol, Print["Convergencia en ",i," iteraciones"];
    Print["la solución es ", x1];Break[] ];
  x0=x1;
];
1 2. 0.270671
2 4. 0.0732626
3 5.33333 0.0257491
4 6.5641 0.00925597
5 7.74383 0.00335625
6 8.89211 0.00122239
7 10.0188 0.000446374
8 11.1297 0.000163274
9 12.2284 0.000059791
10 13.3175 0.0000219137
11 14.3987 8.03642×10-6
12 15.4733 2.9486×10-6
13 16.5424 1.08226×10-6
14 17.6067 3.9735×10-7
15 18.6669 1.45922×10-7

```

Notar que se ha alcanzado el número máximo de iteraciones y en este caso el método de Newton no ha resultado convergente.

Ejercicio: Aplicar el método de Newton a $f(x) = \cos\left(\frac{5+x^2}{x^2+1}\right)$ y $x_0=-2$, observando la convergencia cuadrática.

Por otra parte *Mathematica* dispone de la orden **FindRoot[f(x),{x,x0}]** que constituye una implementación refinada del método de Newton, que hace que sea más potente que la implementación hecha arriba. Veamos algunos ejemplos

```

(* Calcular el cero de f (x)=x e-x partiendo de x0=2,
observando que el resultado no es correcto *)
FindRoot[x*Exp[-x], {x, 2.}]

```

```
{x → 17.6067}
```

```

(* Calcular el cero de f (x)=x e-x partiendo de x0=0.5,
donde el resultado en este caso sí es correcto*)
FindRoot[x*Exp[-x], {x, .5}]

```

```
{x → 0.}
```

(* Calcular el cero de $f(x) = \arctan(x)$ partiendo de $x_0=2$,
dando el resultado correcto *)
FindRoot[ArcTan[x], {x, 2.}]

{x → -2.71495×10^{-18} }

Ejercicio: Se pretende calcular una raíz de la función $f(x)=\arctan(x)$

- Demostrar que la raíz es única.
- Aplicar el método de Newton partiendo de $x_0=0.5$ verificando la convergencia cuadrática
- Explicar lo que ocurre si $x_0 = 2$.

Ejercicio: Se pretende calcular una raíz de la función $f(x)=x^3 - x - 3$

- Encontrar un intervalo alrededor del origen conteniendo una raíz
- Aplicar el método de Newton partiendo de $x_0= 0$.
- Aplicar el método de Newton partiendo de $x_0= 1.1$