Derivabilidad de una función real de variable real: Propiedades y aplicaciones.

Práctica 4 (Específica de la asignatura de Cálculo Matemático en E.U.A.T.)

Introducción

En esta práctica estudiaremos los comandos de *Mathematica* para el cálculo de derivadas. También analizaremos la interpretación geométrica de la derivada como límite de las rectas secantes e ilustraremos gráficamente esta situación. Después daremos ejemplos de aplicaciones prácticas del uso de las derivadas (cálculo de extremos, problemas de optimización) para concluir viendo cómo se puede manejar el polinomio de Taylor de una función en *Mathematica.*

Como apéndice se incluye el problema de la búsqueda de las raíces reales (ceros) de una función *f(x)* o búsqueda de las soluciones reales de la ecuación *f(x)*=0. En general no podemos resolver de manera exacta este problema. Veremos varias técnicas de aproximación de la solución buscada.

Antes de comenzar hacemos una limpieza de la memoria del programa.

```
Clear["Global`*"]
```

Derivada de una función en un punto: cálculo directo con *Mathematica*

La derivada de una función real de variable real *f(x)* en un punto *a* de su dominio se calcula (cuando existe) por medio de cualquiera de los dos expresiones siguientes

$$f'(x) = \lim_{x\to a} \frac{f(x)-f(a)}{x-a}$$

o

$$f'(x) = \lim_{h\to 0} \frac{f(a+h)-f(a)}{h}.$$

Además, una función es derivable en un conjunto (abierto) si lo es en cada uno de los puntos de dicho conjunto.

Como sabemos calcular límites, ya podemos calcular derivadas. Sin embargo, *Mathematica* posee comandos directos para calcular la derivada de una función en un punto. Así, si hemos definido previamente la función *f(x)* en la forma **f [x_]:= ...**, para calcular la derivada primera, en un punto *x* cualquiera, simplemente escribimos

f '[x]

y ejecutamos. Para la derivada segunda escibiremos dos ``primas´´ (**f ''[x]**) y así sucesivamente.

Otra forma de calcular la derivada n-ésima de una función en un punto es usando el comando **D**.

D[f[x],x] para la derivada primera de *f* con respecto a *x*;

D[f[x], {x,n}] para la derivada n-ésima de *f* con respecto a *x*.

Veamos algunos ejemplos.

■ **Ejemplo 1** (con límites)

```
f[x_] := x^2;    (* la derivada de x^2 en un punto a es 2 a *)
Limit[(f[x] - f[a]) / (x - a), x -> a]
```

■ **Ejemplo 2** (con **f '[x]**)

```
f[x_] := Log[x]; (* la derivada del logaritmo natural es 1/x cada punto x *)
f'[x]
```

■ **Ejemplo 3** (con el comando **D**)

```
f[x_] := Exp[x]; (* cualquier derivada de la exponencial es ella misma *)
Do[Print[n, "a. derivada de ⅇ^x es ", D[f[x], {x, n}]], {n, 1, 10}]
```

Interpretación gráfica de la derivada: pendiente de la recta tangente

Cuando existe, la derivada de una función *f* en un punto *a* coincide con la pendiente de la recta tangente a la gráfica de *f(x)* en el punto *(a,f(a))*. En efecto, si tenemos en cuenta que el cociente incremental *(f(a)-f(a+h))/h* es la pendiente de la recta secante que pasa por los puntos *(a,f(a))*, *(a+h,f(a+h))*, hacer tender *h* a cero equivale a encontrar la pendiente de la recta tangente como límite de las pendientes de las rectas secantes. Veamos esta interpretación en un ejemplo concreto.

■ **Ejemplo 4**

Consideramos la función definida en todo número real según la ley

```
f[x_] := ((x - 4)^2) / 8
```

La expresión explícita de la recta secante a la gráfica de *f(x)* que pasa por los puntos *(a, f(a))* y *(a+h, f(a+h))* es

```
secante[x_, a_, h_] := f[a] + ((f[a + h] - f[a]) / h) (x - a);
```

Y la expresión explícita de la recta tangente a la gráfica de *f(x)* que pasa por los puntos *(a, f(a))* es

```
tangente[x_, a_] := f[a] + f'[a] (x - a);
```

Estudiemos, aprovechando las posibilidades de animación de *Mathematica*, qué ocurre cuando h tiende a 0 en el punto $a=2$.

```
Do[Plot[{f[x], secante[x, 2, h], tangente[x, 2]}, {x, 0, 8}, PlotRange → {-1, 2},
   PlotStyle → {{Thickness[0.01], RGBColor[0, 0, 0]},
     {Thickness[0.01], RGBColor[0, 0, 1]}, {Thickness[0.01], RGBColor[1, 0, 0]}},
   AxesLabel → "Azul=secante // Rojo=tangente"], {h, 2.41, 0.01, -0.06}];
```

Derivabilidad, Teorema de Rolle y Teorema del Valor Medio

■ Continuidad y derivabilidad de una función definida a trozos

Veamos como podemos analizar la continuidad y derivabilidad de una función definida a trozos. Consideremos $h(x)$ definida por

$$h(x) = \begin{cases} c^2 x^2 + x + a\, e^x & \text{si } x <= 0 \\ x^3 + a\, x^2 + c\, e^{-x} & \text{si } x > 0 \end{cases}$$

Para que sea derivable es necesario que sea continua. Es claro que $h(x)$ es continua en todo punto salvo quizás en el punto $x=0$. Analicemos, entonces, lo que ocurre en este punto. Consideramos las funciones auxiliares

```
h1[x_] := c^2 x^2 + x + a Exp[x]
h2[x_] := x^3 + a x^2 + c Exp[-x]
```

Calculemos los límites laterales.

```
Limit[h1[x], x -> 0, Direction -> 1]
Limit[h2[x], x -> 0, Direction -> -1]
```

Así, para que exista el límite en cero debemos imponer la igualdad $a=c$. Por lo tanto, hay que exigir esta condición para que sea $h(x)$ sea continua.

De igual forma, para la derivabilidad en $x=0$ calculamos los límites

```
Limit[h1'[x], x -> 0, Direction -> 1]
Limit[h2'[x], x -> 0, Direction -> -1]
```

Concluimos que $h(x)$ será derivable si y sólo si $a=c$ y $1+a=-c$. La solución de este sistema es $a = c = -\frac{1}{2}$. Representemos $h(x)$ para estos valores usando el comando **Which**.

```
a = c = -1/2;
h[x_] := Which[x ≤ 0, c^2 x^2 + x + a Exp[x], x > 0, x^3 + a x^2 + c Exp[-x]]
Plot[h[x], {x, -6, 2}];
```

Recuerda que la orden **Which** no permite trabajar con **Limit**. Sin embargo sí funciona con la derivada (una vez probado que la función es derivable). En el caso anterior, es muy simple representar la derivada de la función $h(x)$.

```
Plot[h'[x], {x, -6, 2}];
```

También se puede calcular la derivada en un punto cualquiera del dominio.

```
h'[0]
```

■ Interpretación gráfica del teorema de Rolle

```
Clear[f];
```

Sea la función definida en todo número real

```
f[x_] := E^{(x + 1)^2 / 2}
```

Antes de intentar aplicar el Teorema de Rolle a dicha función, debemos asegurarnos de la existencia de un intervalo en el que dicha función verifique las hipótesis del teorema. Para esto nos ayudamos de la representación gráfica.

```
Plot[f[x], {x, -3, 3}, PlotRange → {0, 10}];
```

Una vez nos hemos convencido de que se puede asegurar la existencia de un punto donde se anula la derivada, vamos a encontrarlo empleando *Mathematica* (Ver apéndice).

```
valores = Solve[f'[x] == 0, x]
```

```
c = x /. valores[[1]]
```

Hemos usado aquí otro comando básico de *Mathematica*

``expresión que depende de x´´ /. {*x* −> valor}

Este comando da como salida el resultado de evaluar la **expresión** cuando ***x*** toma el **valor** dado **sin asignar este valor a la variable** *x*.

Por último comprobamos gráficamente que efectivamente hemos encontrado el valor dado por el Teorema de Rolle representado la función y la recta tangente en este punto.

```
Plot[{f'[c] (x - c) + f[c], f[x]}, {x, -3, 1}, PlotRange → {0, 10}];
```

■ Interpretación geométrica del Teorema del Valor Medio

Dada una función derivable *f* en un intervalo *[a,b]*, el Teorema del Valor Medio está relacionado con la siguiente cuestión: ¿en qué puntos es la tangente a la curva *y=f(x)* paralela a la secante que une los puntos *(a,f(a))* y *(b,f(b))*?

Veamos gráficamente el problema para la función *f(x)=x^2-3 x +5* en el intervalo de extremos *a=-1* y *b=0.25*.

```
f[x_] := x^2 - 3 x + 5;
a = -1;
b = 0.25;
secante[x_] := ((f[b] - f[a]) / (b - a)) (x - a) + f[a]
Plot[{f[x], secante[x]}, {x, -1., 0.25},
  PlotStyle → {{RGBColor[0, 1, 0]}, {RGBColor[0, 0, 1]}}];
```

Se trata, por tanto, de calcular los posibles valores , *c*, para los cuales las pendientes de la recta secante y la tangente en el punto *c* sean iguales. Concretamente, lo que asegura el Teorema del Valor Medio es que existe al menos un punto *c*.

Para calcular los posibles puntos con *Mathematica* podemos utilizar el comando **Solve**.

```
Clear[c];
valores = Solve[f'[c] == (f[b] - f[a]) / (b - a), c]
```

El siguiente comando asigna a *c* el valor correspondiente a la solución de la ecuación anterior:

```
c = c /. valores[[1]]
```

Aplicación: cálculo de extremos y optimización

■ Ejemplo 5

En este ejemplo vamos a calcular los extremos relativos y absolutos (si existen) de la siguiente función

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

Como la función es derivable en todo su dominio (todos los números reales), sabemos que los extremos relativos anulan a la derivada. Por lo tanto buscaremos los puntos que anulan a la derivada. Para resolver la ecuación $f'(x) = 0$ utilizaremos nuevamente el comando **Solve**.

```
f[x_] := 2 x^3 - 10 x^2 + x - 4 ;
sol = Solve[f'[x] == 0, x]

sol // N
```

Para ver si son máximos o mínimos calculamos la derivada segunda en dichos puntos. Para ello hemos asignado la salida del comando **Solve** a la variable **sol**.

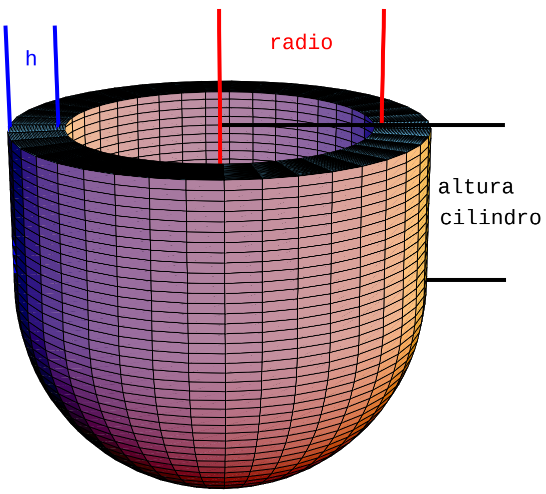
```
f''[x] /. sol[[1]] // N
f''[x] /. sol[[2]] // N
```

Tenemos que en $x = 0.0507734..$ f alcanza un máximo relativo mientras que en $x = 3.28256...$ alcanza un mínimo relativo. No existen máximos ni mínimos absolutos ya que f tiende a infinito en infinito y a menos infinito en menos infinito. Veamos la gráfica para confirmar los resultados.

```
Plot[f[x], {x, -10, 10}];
```

■ Ejemplo 6

Queremos construir un silo cuya base sea una semiesfera y cuyas paredes verticales tengan forma de cilindro (ver figura) de manera que su capacidad sea de 1.000 m^3. Teniendo en cuenta que todas las paredes han de tener un grosor uniforme de 50 cm, que el material empleado en la superficie esférica cuesta 200 euros$/m^3$ y el correspondiente al cilindro cuesta sólo 120 euros$/m^3$, hemos de calcular el radio y la altura de la pared lateral óptimos para minimizar el gasto de material en la construcción del silo.



■ Ejecutar para ver más esquemas del silo

```
f[x_] := Which[0 ≤ x < 1, Sqrt[1 - (x - 1)^2], 1 ≤ x, 1, x <= 0, 0];
fext[x_] := 1.3 * f[(x + 0.3) / 1.3];
Plot[{f[x], fext[x]}, {x, -0.3, 2}];

n = 10; (*n será el número de dibujos*)
Do[ParametricPlot3D[{{Cos[t] * f[z], Sin[t] * f[z], z},
   {Cos[t] * fext[z], Sin[t] * fext[z], z},
   {Cos[t] * (1 + 0.15 z), Sin[t] * (1 + 0.15 z), 2}},
  {t, 0, 2 π}, {z, -0.3, 2},
  Axes → False, AspectRatio → 1, PlotPoints → 40, Boxed → False, ViewPoint ->
   {0, 4 * Cos[0.4 + Sin[2 i Pi / n]], 4 * Sin[0.4 + Sin[2 i Pi / n]]}], {i, 0, n - 1}]
```

El hecho de que el volumen interior sea constante determinará una relación entre la altura y el radio. Llamamos *a* a la altura (en metros), *r* al radio (en metros) y *h* al grosor de la pared (en metros).

1.000 = Volumen = volumen cilindro + volumen semiesfera = (Base cilindro x altura) + (vol. esfera) / 2

$= \pi\, r^2 \mathrm{a} + \frac{4}{3} \pi\, r^3 / 2$

de donde $a = \frac{1}{\pi r^2}\,(1000 - 2/3\, \pi\, r^3)$. Definimos pues *a* como función de r.

```
Clear[a, h, V, r, x, f]
a[r_] := (1000 - (2 / 3) Pi r^3) / (Pi r^2)
```

El material a usar tanto en la semiesfera como en el cilindro será igual al volumen de las paredes correspondientes. En cada caso este volumen será el determinado por la cara externa y la cara interna. Distinguimos entre la semiesfera y el cilindro, ya que los costes son distintos.

Volumen (en m^3) de material en el cilindro = $\mathrm{Pi}\,(r+h)^2 a - \mathrm{Pi}\, r^2\, a$,

coste del cilindro (en euros) $= 120\,(\mathrm{Pi}\,(r+h)^2\, a - \mathrm{Pi}\, r^2\, a)$.

Volumen (en m^3) de material en la semiesfera = $\frac{2}{3}\mathrm{Pi}\,(r+h)^3 - \frac{2}{3}\mathrm{Pi}\, r^3$,

coste de la semiesfera (en euros) = $200\,(\frac{2}{3}\,\mathrm{Pi}\,(r+h)^3 - \frac{2}{3}\,\mathrm{Pi}\, r^3)$.

Definimos la función de coste dependiente del radio (sin olvidar que *a* es función de r):

```
h = 1 / 2;
f[r_] := 120 (Pi (r + h) ^2 a[r] - Pi r^2 a[r]) + 200 ((2 / 3) Pi (r + h) ^3 - (2 / 3) Pi r^3)
f[r] // Simplify
Plot[f[r], {r, 1, 30}];
```

Nota: Aunque no se haya dicho, es evidente que la función *f* sólo está definida para valores positivos (¡ha de existir radio!). Buscamos el mínimo de *f*, por lo que procederemos igual que en el ejemplo anterior. (Nota: obsérva la diferencia entre **Solve** y **NSolve**).

```
sol = NSolve[f '[r] == 0, r]
```

La segunda y tercera solución no tienen sentido ya que son complejas y la cuarta tampoco ya que es negativa. Por lo tanto la única solución posible sera la primera. (El orden de salida de las soluciones corresponde a la versión 5.2 de *Mathematica*. El orden puede ser diferente con otras versiones. Hay que tener esto en cuenta en las cuatro órdenes siguientes). Confirmamos que es un mínimo evaluando la derivada segunda.

```
f ''[r] /. sol[[1]]
```

Así pues el radio y la altura óptimos son

```
Print["radio óptimo = ", r /. sol[[1]], " m."]
Print["altura óptima = ", a[r] /. sol[[1]], " m."]
```

Y el coste

```
Print["Coste mínimo = ", f[r] /. sol[[1]], " euros"]
```

Polinomio de Taylor de grado n

Recordemos que el polinomio de Taylor de grado n de una función *f(x)* en un punto a viene dado por

$$T_{n,f}(x; a) = f(a) + f'(a)(x-a) + \frac{1}{2} f''(a)(x-a)^2 + \ldots + \frac{1}{n!} f^{n)}(a)(x-a)^n.$$

En *Mathematica* existe un comando directo que genera este polinomio y proporciona una pequeña información sobre el orden del resto de Lagrange

Series[f (x) , { x, a, n}] .

■ Ejemplo 7

```
Series[E^x, {x, 0, 10}]  (* Observa el efecto de añadir el comando Normal[] *)
Normal[Series[E^x, {x, 0, 10}]]
```

■ Ejemplo 8

Como sabemos, el polinomio de Taylor de grado n es el polinomio de grado n "más cercano" a la función *f(x)* en torno al punto *a*. Usemos esta herramienta para aproximar el valor del número Pi, teniendo en cuenta que Pi = 4 arcsen ($\sqrt{2}/2$).

```
Series[4*ArcSin[x], {x, 0, 11}]
```

$$4x + \frac{2x^3}{3} + \frac{3x^5}{10} + \frac{5x^7}{28} + \frac{35x^9}{288} + \frac{63x^{11}}{704} \text{ /. } \{x \to \text{Sqrt}[2]/2.\}$$

■ Ejemplo 9

Veamos ahora cómo el polinomio de Taylor se acerca cada vez más a la función si aumentamos el grado. Para ello consideramos la función seno y optamos por construir "a mano" el polinomio de Taylor.

```
f[x_] := Sin[x];
a = 0.;
taylor[x_, n_] := Sum[(D[f[t], {t, k}] /. {t → a}) / k! (x - a)^k, {k, 0, n}];
Do[Plot[{Sin[x], taylor[x, n]}, {x, -2 Pi, 2 Pi},
  PlotRange → {-2, 2}, PlotStyle → {{Thickness[0.01], RGBColor[0, 0.6, 1]},
   {Thickness[0.005], RGBColor[1, 0, 0]}}, AxesLabel → "orden" n], {n, 1, 9, 2}]
```

Comprobemos que el error cometido al aproximar el sen(π) (que es cero) por el de su polinomio de Taylor en x=0 de grado 9 es tan sólo de un 7%.

```
taylor[Pi, 9]
```

Ejercicios

1.- Calcula y simplifiqua las derivadas de las siguientes funciones.

i) $f(x) = \ln(x^2 + x + 1)^{1/2} - \frac{1}{\sqrt{6}} \operatorname{arctg}\left((2x+1) \big/ \sqrt{3}\right)$,

ii) $f(x) = (\cos(x)^x)^{\operatorname{tg}(x)}$.

2.- Estudia la continuidad y derivabilidad de las siguientes funciones en función de los parámetros a y b.

i) $f(x) = b\,x^2 \operatorname{sen}(x) - a\,e^x$ si $x \leq 0$,
$\log(x+1) + b\,x$ si $x > 0$.

ii) $g(x) = b\,e^{x+1}$ si $x \leq -1$,
$b \operatorname{tg}(x+1) + 2a$ si $x > -1$.

3.- Determina los intervalos en los que se puede aplicar el Teorema de Rolle en las siguientes funciones. Calcula el punto que da el teorema de Rolle en cada caso.

i) $f(x) = \log(x)$

ii) $g(x) = \frac{x-1}{x-2}$

iii) $h(x) = \frac{x-1}{(x-2)(x-8)(x-9)}$

4.- Calcula los extremos locales de las siguientes funciones, así como sus puntos de inflexión.

i) $f(x) = x^8 - 2x^5 + 3x^3 - x^2 - 2$

ii) $g(x) = x^{10} - 4x^7 + x^6 - x^3 - x - 2$

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

5.- Calcula (en metros) la altura óptima que ha de tener una caja de base cuadrada para que pueda contener 1000 litros y de manera que el coste del material para hacerla sea mínimo. (Para simplificar se supondrá que los lados de la caja no tienen grosor).

6.- Repite el problema del silo suponiendo que el precio del material de la semiesfera y del cilindro es el mismo y observa la altura resultante.

7.- Halla los puntos de la curva $y = x^2 - 3x + 5$ en los cuales la tangente es perpendicular a la recta y+x=0. (Sugerencia: piensa en la pendiente que tienen las rectas perpendiculares a la recta y+x=0).

8.- Calcula y representa las series de Taylor que se proponen.
i) Polinomio de Taylor de orden 10 de $f(x) =$ arctg(x) centrado en *x0=1*.
ii) Polinomio de Taylor de orden 7 de $g(x) = x^{10}-4x^7+x^6-x^3-x-2$ centrado en *x0=0*.
iii) Polinomios de Taylor de órdenes 15 y 16 de la función seno centrados en *x0=0*. ¿Algo que comentar?

9.- Haz una tabla con las gráficas de los 20 primeros polinomios de Taylor de la función $f(x) = e^x$ centrados en *x0=1*. Debes mostrar la función exponencial en rojo y los polinomios de Taylor en azul en el intervalo *[-5,5]*.

Apéndice: Resolución de ecuaciones

El problema que se plantea en este apéndice es el problema de encontrar las raíces reales (ceros) de una función *f(x)* o, lo que es lo mismo, buscar las soluciones reales de la ecuación *f(x)*=0. En multitud de ocasiones este problema no se puede resolver de forma exacta, por lo que debemos de recurrir a diversas técnicas de aproximación. Veremos algunos comandos útiles para nuestros propósitos.

Solve[ ]	resuelve analíticamente algunas ecuaciones;
NSolve[ ]	resuelve numéricamente la mayoría de las ecuaciones;
FindRoot[ ]	resuelve ecuaciones por el método de "Newton-Raphson";
Break[]	detiene un proceso iterativo.

■ Resolución analítica con el comando Solve

Para resolver una ecuación de forma analítica se puede utilizar la orden

Solve[ecuación, variable]

con la conveniencia de que ésta efectúa simultáneamente una discusión de casos, si interviene algún parámetro en la ecuación, según los valores que pueda tomar el mismo.

■ Ejemplo

```
Solve[x^6 == 1, x]
```

■ Ejemplo

```
Solve[Sin[x] Cos[x] == 0, x]
```

Observa que aparece un mensaje indicando que, debido al método de resolución empleado, puede que no se encuentren todas las posibles soluciones.

Solve tambien puede resolver ecuaciones dependientes de parámetros e incluso sistemas de ecuaciones.

■ Ejemplo

```
Solve[{x == 1 + 2 a y, y == 9 + 2 x}, {x, y}]
```

■ Ejemplo

```
Solve[E^(x+a) - x == 0,x]
```

Si el comando **Solve** no permite resolver la ecuación, tendremos que utilizar algún método de resolución numérica: **NSolve** o **FindRoot**.

■ Resolución directa con el comando NSolve

El comando **NSolve[ecuación, incógnitas]** resuelve numéricamente la ecuación planteada respecto de las variables "**incógnitas**". Veamos cómo usarlo para buscar los ceros de las funciones

$$f(x) = x^5 - 2x^3 + \operatorname{sen}(x) - e^{-4x} \quad \text{y} \quad g(x) = 3x^6 - x^3 + x^2 - 2.$$

Primero dibujamos las funciones para "ver" sus ceros.

```
f[x_] := x^5 - 2 x^3 + Sin[x] - E^(-4 x);
g[x_] := 3 x^6 - x^3 + x^2 - 2;
Plot[f[x], {x, 0, 2}];
Plot[g[x], {x, -1, 1}];
```

Ahora intentamos buscar sus raíces usando **NSolve.**

```
NSolve[g[x] == 0, x]
NSolve[f[x] == 0, x]
```

Observa que **¡el comando NSolve no siempre proporciona la información deseada!** Aunque ha encontrado con éxito todas las raices de *g* (seis en total, incluyendo complejas, puesto que *g(x)* es un polinomio de grado 6) no puede sin embargo hallar los ceros de la función *f(x)* que es "un poco más complicada".

Veamos otra forma de resolver este problema.

■ Resolución con el comando FindRoot

El comando **FindRoot** (Encontrar Raíces) usa un algoritmo numérico (el método de Newton-Raphson) para localizar **sólo una raíz real.** Este comando requiere de un valor inicial que esté "próximo" a la raíz buscada. Para ello podemos valernos de la representación gráfica de la función correspondiente para "intuir" ese valor inicial.

La sintaxis del comando es:

FindRoot[funcion==0, {variable, valor inicial}].

■ Ejemplos

```
FindRoot[f[x] == 0, {x, 0.3}]
FindRoot[f[x] == 0, {x, .8}]
FindRoot[f[x] == 0, {x, 1.2}]
FindRoot[g[x] == 0, {x, -.8}]
FindRoot[g[x] == 0, {x, 0.9}]
```

■ Método de bisección

A continuación vamos a programar el **Método de bisección**. Este método (o algoritmo) está basado en el **Teorema de Bolzano** y permite aproximar la raíz de una función *f(x)* **continua** sabiendo que cambia de signo en los extremos de un intervalo dado *[a,b]*.

El algoritmo consiste en ir sustituyendo en cada paso el intervalo por otro de longitud la mitad del anterior y de manera que en el nuevo intervalo la función siga cambiando de signo en los extremos. Se consigue así un intervalo tan pequeño como se desee en el que se sabe que hay una raíz de *f(x)* y se toma como aproximación de ésta el centro del intervalo.

Veamos una forma sencilla de programarlo con *Mathematica* haciendo que el proceso se detenga cuando la longitud del intervalo obtenido no supere una cierta **cota de error** (es decir, la aproximación no se aleje de la raíz más de esa cota) y limitando el **número total de pasos**.

```
a = -1.; b = 2.;  (* extremos del intervalo inicial *)
n = 25;  (* número máximo de pasos permitido (adaptable a cada caso) *)
error = 10^(-3);  (* cota de error permitida (~3 cifras decimales exactas) *)

paso = 0;
Do[c = (a + b) / 2;  (* calcula el punto medio *)
 Print["a=", a, "  b=", b, "  c=", c, "  error=", c - a];
 (* pinta: a, b, c y el error cometido *)
 If[c - a < error, Break[]];  (* se detiene si el error no supera la cota *)
 If[f[c] == 0, Break[]];  (* se detiene si ha dado exactamente con la raíz *)
 If[f[a] * f[c] < 0, b = c, a = c];  (* elegimos el nuevo intervalo *)
 paso = paso + 1,  (* aumentamos el contador de paso *)
 {i, n}]

Print["Número de pasos: ", paso]
Print["Solución aproximada: ", c]
```

■ Ejercicios

1.- Aproxima al menos dos soluciones reales distintas de la ecuación $x^3 + 3x^2 - 2x - 1 = 0$ mediante los comandos **NSolve**, **FindRoot** y el método de bisección con error máximo permitido 10^{-5}.

2.- Encuentra los puntos de corte de las funciones $f(x) = x^2 - 2x - 1$ y $g(x) = \text{sen}^2(x)$. Representa previamente sus gráficas.

3.- Realiza los ejercicios II.14 y II.15 de la relación de teoría del tema 2.