4. FUNCIONES DE UNA VARIABLE

1 Constantes matemáticas

Maxima puede manejar ciertos números de uso frecuente en matemáticas como pi o e o el número complejo i, del siguiente modo:
-Número pi: %pi,
-Número e: %e,
-Número complejo i: %i.

EJEMPLO 4.1. Calcule en forma decimal pi, e y 2e.

```
(%i1) float(%pi); float(%e); 2*%e, numer;
(%o1) 3.141592653589793
(%o2) 2.718281828459045
(%o3) 5.43656365691809
```

Obsérvese que la última equivale a

```
(%i4) float(2*%e);
(%o4) 5.43656365691809
```

2 Funciones matemáticas

Maxima tiene predefinidas la inmensa mayoría de funciones de uso corriente en Matemáticas. Consulte la Ayuda para tener una relación completa. Las que utilizaremos en este curso son las siguientes:
-abs: valor absoluto,
-min: mínimo de varios números,
-max: máximo de varios números,
-signum: signo de un número,
-sqrt: raiz cuadrada,
-exp: exponencial de base e,
-log: logaritmo natural de base e,
-sin: seno,
-cos: coseno,
-tan: tangente,
-sec: secante,
-csc: cosecante,
-cot: cotangente,
-asin, acos, atan, asec, acsc y acot son, respectivamente, las inversas de las 6 anteriores, que en Matemáticas usualmente se denotan arcsen, arccos, arctan, etc.
El factorial de un número n es n!
En las funciones trigonométricas el argumento ha de estar en radianes.

EJEMPLO 4.2. Halle sen(pi), arcsen(0.5), e^2 y log10(2), donde log10 denota el logaritmo en base 10.

(%i5) sin(%pi); asin(0.5); exp(2);

(%o5) 0

(%o6) 0.5235987755983

(%o7) $\%e^{2}$

La función logaritmo en base 10 no está predefinida en Maxima. Se puede definir mediante la fórmula del cambio de base de los logaritmos, del siguiente modo:

(%i8) log10(x):=log(x)/log(10);
log10(2);

(%o8) $\mathrm{log10}(x):=\frac{\log(x)}{\log(10)}$

(%o9) $\frac{\log(2)}{\log(10)}$

(%i10) float(%);

(%o10) 0.30102999566398

3 Definir una función

Hay dos formas de definir una función:
1) Mediante el operador de asignación (:=). Por ejemplo: f(x):=3*x/(x+1).
2) Con el comando define. Ejemplo: define(g(x), x^2-4*x).

EJEMPLO 4.3. Definir la funciones f(x)=4x-x^2 y g(x)=(2x-1)/x y calcúlense f(-2/3), g(5) y g(raiz de 2) en forma exacta y en forma decimal.
Halle también la función compuesta fog (en forma simplificada).

(%i11) f(x):=4*x-x^2; f(-2/3); float(f(-2/3));

(%o11) $f(x):=4\,x-x^{2}$

(%o12) $-\frac{28}{9}$

(%o13) -3.111111111111111

(%i14) g(x):=(2*x-1)/x; g(5); float(g(5)); g(sqrt(2)); %,numer;

(%o14) $g(x):=\frac{2\,x-1}{x}$

(%o15) $\frac{9}{5}$

(%o16) 1.8

(%o17) $\frac{2^{3/2}-1}{\sqrt{2}}$

(%o18) 1.292893218813453

(%i19) f(g(x));

(%o19) $\frac{4\,(2\,x-1)}{x}-\frac{(2\,x-1)^{2}}{x^{2}}$

```
(%i20) ratsimp(%);
```

(%o20) $\dfrac{4\,x^2-1}{x^2}$

La definición de una función mediante := a veces no es adecuada como lo pone de manifiesto el siguiente ejemplo y como veremos posteriormente a la hora de definir la función derivada.

EJEMPLO 4.4. Defina las funciones p(x)=(x-1)(x-2)(x-3) y p1(x)=p(x)/(x-1) (simplificada). Compruebe que p1 toma el valor 2 en 1.

```
(%i21) p(x):=(x-1)*(x-2)*(x-3);
       define(p1(x),p(x)/(x-1)); p1(1);
```

(%o21) $\mathrm{p}(x):=(x-1)\,(x-2)\,(x-3)$

(%o22) $\mathrm{p1}(x):=(x-3)\,(x-2)$

(%o23) 2

Hemos usado "define" para definir p1(x) porque con ":=" no funciona ya que no simplifica como vemos a continuación.

```
(%i24) h(x):=p(x)/(x-1);
```

(%o24) $\mathrm{h}(x):=\dfrac{\mathrm{p}(x)}{x-1}$

Al pedirle a Maxima h(1) se produce un error debido a que no ha simplificado:

```
(%i25) h(1);
expt: undefined: 0 to a negative exponent.
#0: h(x=1)
 -- an error. To debug this try: debugmode(true);
```

No se subsana si simplificamos en el momento de definir h:

```
(%i26) h(x):=ratsimp(p(x)/(x-1));
```

(%o26) $\mathrm{h}(x):=\mathrm{ratsimp}\left(\dfrac{\mathrm{p}(x)}{x-1}\right)$

```
(%i27) h(1);
expt: undefined: 0 to a negative exponent.
#0: h(x=1)
 -- an error. To debug this try: debugmode(true);
```

4 Tabla de valores de una función. Listas

Para generar una lista de números (o de expresiones más generales) de manera automática se utiliza la instrucción makelist. Su formato es:
makelist(expr, i, i_0, i_max, step)
donde expr es una expresión que depende de la variable i, i es la variable que va a tomar varios valores, i_0 es el valor inicial, i_max es el valor final y step es el incremento que se hace. Si step=1 no es necesario ponerlo. Esta opción se encuentra en el menú Álgebra > Construir lista.
Unos ejemplos nos aclararán cómo se usa.

EJEMPLO 4.5. Genere una lista con la expresión 3x-1, en la que x empieza en 1 y termina en 4 con incremento de 0.5 (paso o step = 0.5).

```
(%i28) makelist(3*x-1,x,1,4,0.5);
(%o28) [2,3.5,5.0,6.5,8.0,9.5,11.0]
```

¿Qué valores ha tomado x para generar la lista anterior?

```
(%i29) lista:makelist(x,x,1,4,0.5);
(%o29) [1,1.5,2.0,2.5,3.0,3.5,4.0]
```

Se puede hallar la suma y el producto de todos los elementos de una lista con las instrucciones apply("+",lista) y apply("*",lista), respectivamente. También puede hacerse desde el menú: Análisis > Calcular suma o Calcular producto.

EJEMPLO 4.6. a) Halle la suma de los números del 1 al 100.
b) Calcule el producto de los números del 1 al 10.

```
(%i30) lista:makelist(i,i,1,100);
       apply("+",lista);
(%o30) [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,
26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,
50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,
74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,
98,99,100]
(%o31) 5050
```

```
(%i32) lista2:makelist(i,i,1,10);
       apply("*",lista2);
(%o32) [1,2,3,4,5,6,7,8,9,10]
(%o33) 3628800
```

Esta última instrucción y la siguiente equivalen a hallar el factorial de 10, esto es,
10! = 10 · 9 · 8 · ... · 2 · 1.

```
(%i34) 10!;
(%o34) 3628800
```

EJEMPLO 4.7. Obtenga una tabla de valores de la función f(x)=x^2-6x-1 donde x va de -1 a 6 de 1 en 1 y la tabla de los valores que toma x.

```
(%i35) f(x):=x^2-6*x+1;
       makelist(x,x,-1,6);
       makelist(f(x),x,-1,6);
```

(%o35) $f(x):=x^2-6x+1$

(%o36) [-1,0,1,2,3,4,5,6]

(%o37) [8,1,-4,-7,-8,-7,-4,1]

Si la queremos en forma de pares:

```
(%i38) makelist([x,f(x)],x,-1,6);
```

(%o38) [[-1,8],[0,1],[1,-4],[2,-7],[3,-8],[4,-7],[5,-4],[6,1]]

Una función f puede evaluar cada elemento de una lista sin más que hallar f(lista).
Con los datos anteriores:

```
(%i39) lista:makelist(x,x,-1,6);
       f(lista);
```

(%o39) [-1,0,1,2,3,4,5,6]

(%o40) [8,1,-4,-7,-8,-7,-4,1]

5 Gráfica de una función

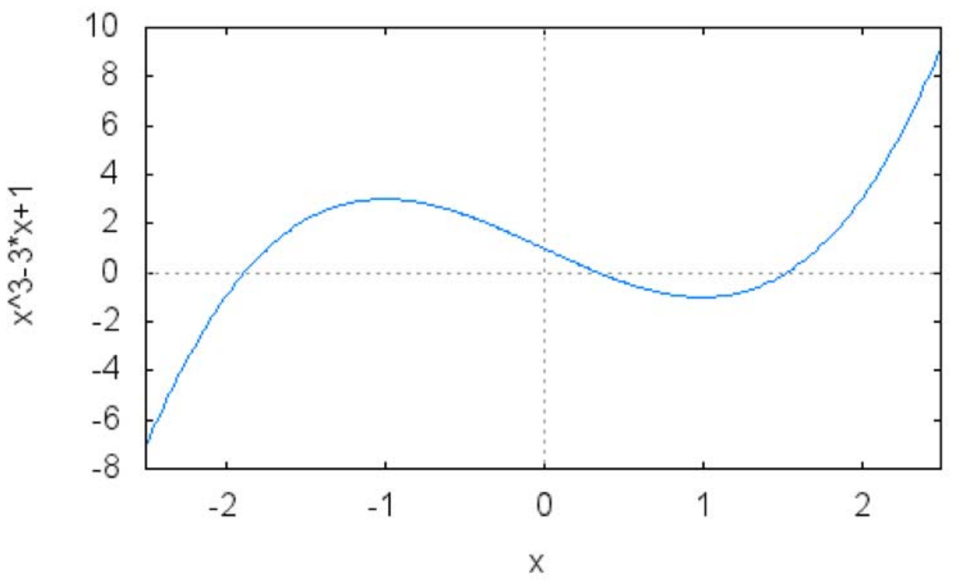
Para representar gráficamente una función se usan las instrucciones wxplot2d y wxdraw2d. Esta última requiere cargar previamente el paquete draw (con load(draw)).
Como siempre encontrará en la ayuda información sobre cómo usarlas.
Para ejecutar la instrucción wxplot2d es mejor utilizar el menú Gráficos > Gráficos 2D.

EJEMPLO 4.8. a) Dibuje la gráfica de y=x^3-3*x+1 con x en [-2.5, 2.5].
b) Dibuje en el mismo gráfico y=x^2 e y=x+2 con la misma unidad en ambos ejes.
c) Represente y=x^2-4x usando wxdraw2d.

a) La opción más simple es

```
(%i41) wxplot2d([x^3-3*x+1], [x,-2.5,2.5])$
```

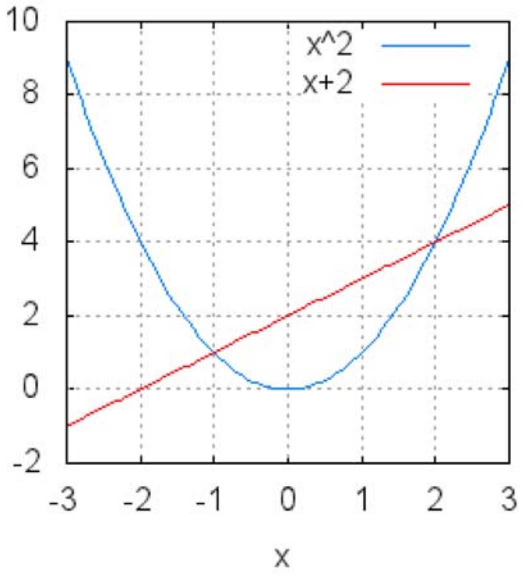
(%t41)



b) Si queremos que use la misma unidad en ambos ejes y una rejilla, sería:

```
(%i42) wxplot2d([x^2,x+2], [x,-3,3],
        [gnuplot_preamble, "set size ratio 1; set grid"])$
```

(%t42)

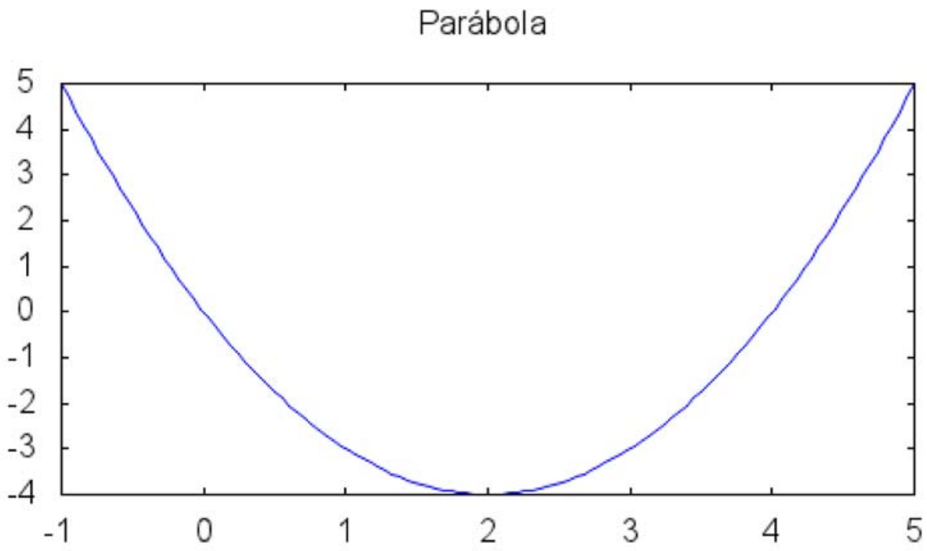


c) La vamos a respresentar en el intervalo [-1,5]

```
(%i43) load(draw)$
Loading C:/Users/user/maxima/binary/binary-gcl/share/draw/grcommon.o
Finished loading C:/Users/user/maxima/binary/binary-gcl/share/draw/grcommon.o
Loading C:/Users/user/maxima/binary/binary-gcl/share/draw/gnuplot.o
Finished loading C:/Users/user/maxima/binary/binary-gcl/share/draw/gnuplot.o
Loading C:/Users/user/maxima/binary/binary-gcl/share/draw/vtk.o
Finished loading C:/Users/user/maxima/binary/binary-gcl/share/draw/vtk.o
Loading C:/Users/user/maxima/binary/binary-gcl/share/draw/picture.o
Finished loading C:/Users/user/maxima/binary/binary-gcl/share/draw/picture.o
```

```
(%i44) wxdraw2d(title="Parábola",explicit(x^2-4*x,x,-1,5));
```

(%t44)



(%o44)

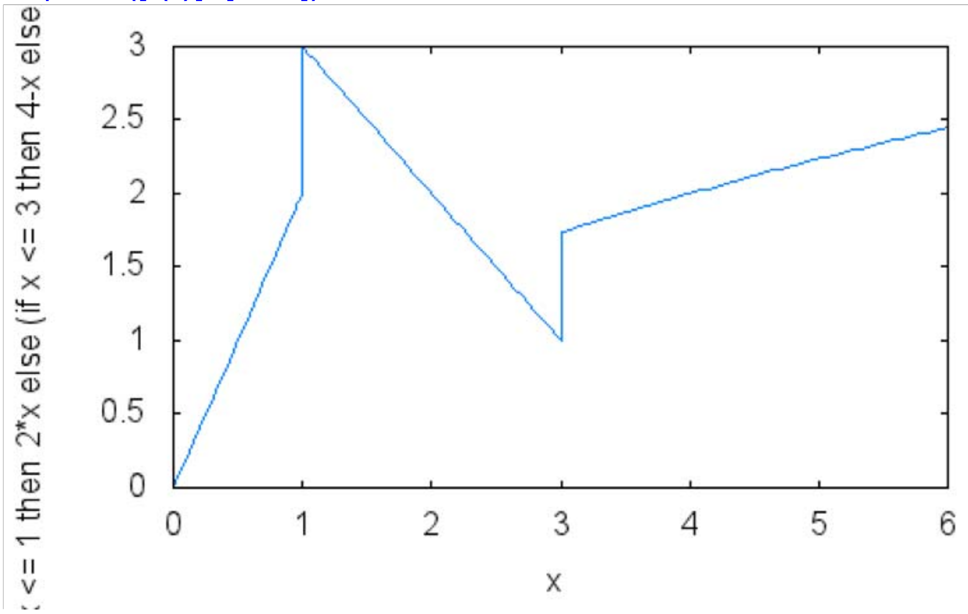
6 Funciones definidas a trozos

EJEMPLO 4.9. Defina la función siguiente, obtenga una tabla de valores y dibuje su gráfica:
f(x)= 2x si x<=1, 4-x si 1<x<=3, raiz de x si x>3.

```
(%i45) f(x):=if x<=1 then 2*x else (if x<=3 then 4-x else sqrt(x));
```
(%o45) $f(x):=\mathrm{if}\ x<=1\ \mathrm{then}\ 2\,x\ \mathrm{else}\ \mathrm{if}\ x<=3\ \mathrm{then}\ 4-x\ \mathrm{else}\ \sqrt{x}$

```
(%i46) makelist(f(x),x,0,5);
```
(%o46) $[0,2,2,1,2,\sqrt{5}]$

```
(%i47) wxplot2d([f(x)], [x,0,6])$
```
(%t47)



Nótese que Maxima une mediante segmentos verticales los puntos de discontinuidad de salto.
Con las funciones definidas a trozos no pueden calcularse límites ni derivadas ni integrales.

7 Definir una sucesión

EJEMPLO 4.10. Defina la sucesión a_n=(2n-1)/(n+1) y determine los 5 primeros términos

```
(%i48) a[n]:=(2*n-1)/(n+1);
```
(%o48) $a_n:=\frac{2\,n-1}{n+1}$

```
(%i49) makelist(a[n],n,1,5);
```
(%o49) $[\frac{1}{2},1,\frac{5}{4},\frac{7}{5},\frac{3}{2}]$

También se puede definir una sucesión mediante una ley recurrente.

EJEMPLO 4.11. Defina la sucesión siguiente: b_1=-9, b_{n+1}=2+b_n y halle los 6 primeros términos y b_23.

Aunque en la definición que nos dan relacionan b[n+1] con b[n], para que Maxima lo entienda hay que relacionar b[n] con el anterior b[n-1].

(%i50) b[1]:-9;b[n]:=2+b[n-1];

(%o50) -9

(%o51) $b_n := 2 + b_{n-1}$

(%i52) makelist(b[n],n,1,6); b[23];

(%o52) $[-9,-7,-5,-3,-1,1]$

(%o53) 35

8 Límites

El cálculo de límites tanto de funciones como de sucesiones se realiza con la instrucción limit cuyo formato es uno de los dos siguientes:

limit(expr, var, punto) o limit(expr, var, punto, plus/minus)

donde expr es la función o sucesión, var es la variable respecto de la cual queremos hallar el límite y punto es el número real. También para punto se admite inf (+infinito) y minf (-infinito).

El segundo formato se emplea para los límites laterales: plus por la derecha y minus por la izquierda.

También se pueden calcular límites en el desplegable Análisis > Calcular el límite.

EJEMPLO 4.12. Los siguientes límites son claros.

(Recuerde que la comilla (') impide que se evalúe la expresión)

(%i54) 'limit((x^2-9)/(x-3),x,3); limit((x^2-9)/(x-3),x,3);

(%o54) $\lim_{x \to 3} \frac{x^2-9}{x-3}$

(%o55) 6

(%i56) 'limit((2*n+1)/n,n,inf); limit((2*n+1)/n,n,inf);

(%o56) $\lim_{n \to \infty} \frac{2n+1}{n}$

(%o57) 2

(%i58) 'limit(1/x,x,0,minus); limit(1/x,x,0,minus);

(%o58) $\lim_{x \to 0^-} \frac{1}{x}$

(%o59) $-\infty$

(%i60) 'limit(sqrt(x^2+x)/(x+2),x,minf); limit(sqrt(x^2+x)/(x+2),x,minf);

(%o60) $\lim_{x \to -\infty} \frac{\sqrt{x^2+x}}{x+2}$

(%o61) -1

(%i62) limit(abs(x)/x,x,0,plus);

(%o62) 1

(%i63) limit((-1)^n,n,inf);
(%o63) *ind*

Este resultado indica que es indeterminado, que no existe, pero que está acotada la sucesión.

(%i64) limit(x*(cos(x))^2,x,inf);
(%o64) *und*

Esto indica que no existe y, o bien que la función no está acotada o bien que no es capaz de averiguar si está acotada.

Nótese que coseno de x al cuadrado, cos^2 x, con Maxima se escribe (cos(x))^2.

9 Derivadas

La derivada se calcula con la siguiente instrucción:
diff(expr, var)
donde expr es una expresión y var es la variable respecto de la cual se deriva.
Si se desea una derivada de orden mayor que 1, se añade el orden:
diff(expr, var, orden).

EJEMPLO 4.13. En las siguientes derivadas es sencillo comprender lo que se pide.

(%i65) diff(x^3,x);
(%o65) $3\,x^2$

(%i66) diff(x^2+b*x+c,x);
(%o66) $2\,x+b$

(%i67) f(x):=x^4-5*x^2+x; diff(f(x),x,2) /* derivada segunda de f(x) */;
(%o67) $\mathrm{f}(x):=x^4-5\,x^2+x$
(%o68) $12\,x^2-10$

(%i69) f(x):=(x^2-1)/(x^2+1); diff(f(x),x);
(%o69) $\mathrm{f}(x):=\dfrac{x^2-1}{x^2+1}$
(%o70) $\dfrac{2\,x}{x^2+1}-\dfrac{2\,x\,(x^2-1)}{(x^2+1)^2}$

Si se halla la derivada con "lápiz y papel" usando la regla del cociente no queda de esta forma sino de la siguiente:

(%i71) factor(%);
(%o71) $\dfrac{4\,x}{(x^2+1)^2}$

EJEMPLO 4.14. Determine la velocidad de un móvil cuya ecuación es x=A sen(omega t).

(%i72) x(t):=A*sin(omega*t); print("v(t)=", diff(x(t),t),"")$

(%o72) $x(t):=A\sin(\omega t)$

$v(t)=\omega\cos(\omega t)A$

REUTILIZACIÓN DE LA DERIVADA.
Si se necesita la derivada para realizar alguna acción con ella conviene definer la función derivada usando "define" ya que con ":=" no funciona.

EJEMPLO 4.15. Se considera la función f(x)=x^2-5x+1. Calcule la derivada en x=2 y x=4.

(%i74) f(x):=x^2-5*x+1; define(Df(x),diff(f(x),x));

(%o74) $f(x):=x^2-5x+1$

(%o75) $Df(x):=2x-5$

Hemos definido la función Df(x) que es la derivada de f(x). Nótese que si procedieramos de la forma que parece natural:

(%i76) f1(x):=diff(f(x),x);

(%o76) $f1(x):=diff(f(x),x)$

al hallar la derivada en 2 nos da un error:

(%i77) f1(2);

diff : second argument must be a variable; found 2
#0: f1(x=2)
-- an error. To debug this try: debugmode(true);

En cambio, con Df(x), no hay problema:

(%i78) Df(2); Df(4);

(%o78) -1

(%o79) 3

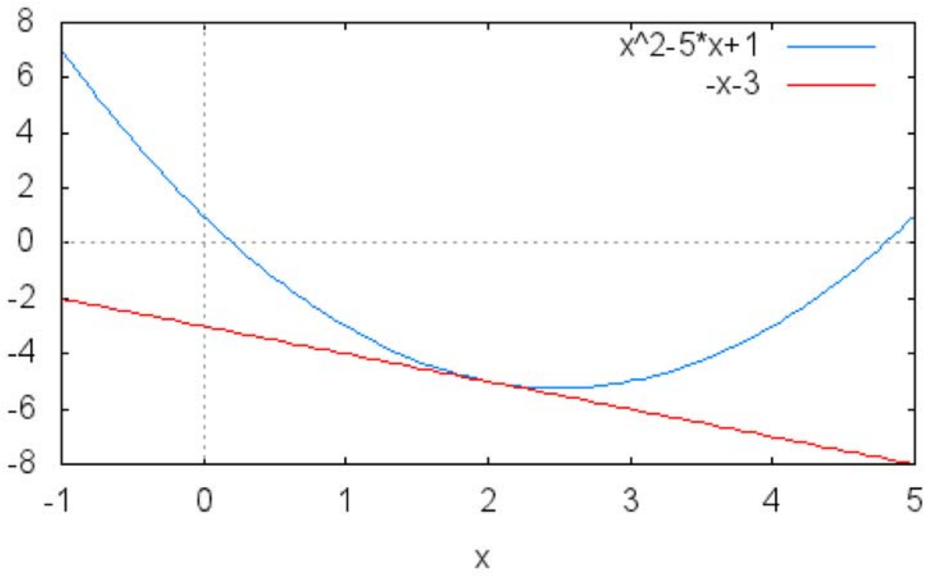
RECTA TANGENTE
Recordemos que la recta tangente a y=f(x) en el punto x=a viene dada por
y=f(a)+f'(a)(x-a).

EJEMPLO 4.16. Determine la ecuación de la recta tangente a la gráfica de la función anterior en el punto x=2 y represente la función y la tangente.

(%i80) define(tgte(x),f(2)+Df(2)*(x-2));

(%o80) $tgte(x):=-x-3$

(%i81) wxplot2d([f(x), tgte(x)], [x,-1,5])$

(%t81) 

EJEMPLO 4.17. Calcule las cuatro primeras derivadas de y=x^4-x^3.

(%i82) f(x):=x^4-x^3;
makelist(diff(f(x),x,i), i, 1, 4);

(%o82) $f(x):=x^4-x^3$

(%o83) $[4\,x^3-3\,x^2, 12\,x^2-6\,x, 24\,x-6, 24]$

10 Polinomio de Taylor de una función

El polinomio de Taylor de una función f(x) centrado en a y de orden n se calcula con la instrucción:
taylor(f(x),x,a,n)

EJEMPLO 4.18. Calcule el polinomio de Taylor de f(x)=cos(x) centrado en el origen y de orden 4. Represente f(x) y el polinomio hallado.

(%i84) f(x):=cos(x);
define(p(x),taylor(f(x),x,0,4));

(%o84) $f(x):=\cos(x)$

(%o85)/T/ $p(x):=1-\frac{x^2}{2}+\frac{x^4}{24}+...$

(%i86) wxplot2d([f(x),p(x)], [x,-4,4])$

(%t86)