Tema 6
Funciones reales de variable real. Representación gráfica. Límites y continuidad

Objetivos

1. Definir funciones con wxMaxima. Cálculos con funciones.
2. Realizar operaciones con funciones.
3. Representar funciones gráficamente.
4. Calcular límites de funciones.
5. Estudiar la continuidad de funciones.
6. Aplicación práctica de les propiedades de las funciones continuas.

Contenidos

06-1. Definición de funciones. Ejemplos. Operaciones con funciones.
06-2. Representación gráfica de funciones. Representación gráfica de curvas planas.
06-3. Límites de funciones. Cálculo y propiedades.
06-4. Funciones continuas. Estudio de la continuidad.
06-5. Propiedades de las funciones continuas. Aplicaciones.

Referencias

AEM11 ALANINOS PRATS, J; EXTREMERA LIZANA, J; MUÑOZ RIVAS, P. (2011)
Cálculo con wxMaxima.

APJ11 ALANINOS PRATS, J; APARICIO DEL PRADO, C; EXTREMERA LIZANA, J; MUÑOZ RIVAS, P.; VILLENA MUÑOZ, A.R. (2011)
Prácticas de ordenador con wxMaxima.

AP86 APOSTOL, T.M. (1986)
Análisis Matemático

BR09 BRUZÓN GALLEGO, M. DE LOS SANTOS; RAMÍREZ LABRADOR, JOSÉ (2009)
Modelos matemáticos con Maxima

BU07 DE BURGOS, JUAN (2007)
Cálculo Infinitesimal de una variable (segunda edición).

ES08 ESTELA CARBONELL, M. ROSA; SAÀ SEOANE, JOEL (2008)
Cálculo con soporte interactivo en moodle.

ES02 ESTEP, DONALD (2002)
Practical Analysis in one variable

GV07a GONZÁLEZ VEGA, LAUREANO (2007)
Laboratorio de Matemáticas. Vol. 1: números y ecuaciones

GV07b GONZÁLEZ VEGA, LAUREANO (2007)
Laboratorio de Matemáticas. Vol. 2: límites y derivadas

JB01 JARAUTA BRAGULAT, EUSEBI (2001)
Anàlisi Matemàtica d'una variable. Fonaments i aplicacions.

RR05 REDONDO NEBLE, M. VICTORIA; RODRÍGUEZ GALVÁN, J. RAFAEL (2005)
Introducción a Maxima

RR08b RODRÍGUEZ RIOTORTO, MARIO (2008)
Curso intensivo i-MATH de software libre orientado a Ciencias e Ingeniería

RU80 RUDIN, WALTER (1980)
Principios de Análisis Matemático.

SP95 SPIVAK, MICHAEL (1995)
Calculus (Càlcul Infinitesimal).

VR09 VALLEJO RODRÍGUEZ, JOSÉ ANTONIO (2009)
Cálculo diferencial con Maxima

06-1.- Definición de funciones. Ejemplos. Operaciones con funciones

Los contenidos de este apartado se desarrollan en el fichero **Tema_06-1.wxm**.

Recordemos que se llama función real de variable real a una terna (A, B, f) en la que A,B son subconjuntos (no vacíos) de números reales y f es una aplicación de A en B. El conjunto A se llama dominio o campo de existencia de la función y el conjunto B se llama recorrido. Habitualmente se escribe $f: A \subset \mathbb{R} \to B$ para designar una función f de A en B. Cuando en adelante digamos función, nos referiremos a una función real de variable real.

Una función se acostumbra a definir mediante una expresión analítica que indica como se transforma el elemento (antiimagen) del conjunto A para obtener el elemento (imagen) del conjunto B que le corresponde. Por ejemplo: $f(x) = 1 + 3x - 2x^2$, $x \geq 0$ sería la forma de dar una función definida mediante la expresión analítica indicada (polinomio) y de campo de existencia $A = [0, +\infty[$.

Para definir una función con wxMaxima, asignar un símbolo a la función, a continuación un paréntesis con el símbolo designado a la variable, a continuación los símbolos := y finalmente la expresión analítica de la función. Por ejemplo:

```
(%i1) f(x):=1+3*x-2*x^2;
```

(%o1) $f(x):=1+3\,x+(-2)\,x^2$

Esto indica que esta función se representa con el símbolo f y que su variable (es decir, un punto cualquiera de su campo de existencia) se representa por x.

Una manera alternativa de definir una función (que más adelante veremos que es muy útil) es con la instrucción "define":

```
(%i2) define(f(x) , 1+3*x-2*x^2 );
```

(%o2) $f(x):=-2\,x^2+3\,x+1$

Otros ejemplos de funciones:

```
(%i3) f1(x):=2*cos(3*x)+4*sin(2*x);
      f2(t):=1/(1+t^2);
      f3(u):=log(u+1);
```

(%o3) $f1(x):=2\cos(3\,x)+4\sin(2\,x)$

(%o4) $f2(t):=\dfrac{1}{1+t^2}$

(%o5) $f3(u):=\log(u+1)$

Se puede obtener la imagen de un punto del dominio con una sintaxis sencilla; por ejemplo:

```
(%i6) f(0);          f(2.34);
(%o6) 1
(%o7) -2.931199999999999
(%i8) f1(%pi);     f1(2.3456);     f2(3.457);
(%o8) -2
(%o9) -2.540661729040671
(%o10) 0.077215015015618
(%i11) f3(%e);     f3(2);        f3(2), numer;
(%o11) log(%e+1)
(%o12) log(3)
(%o13) 1.09861228866811
```

Se observa que en algunos casos parece que la imagen "se deje indicada" sin calcular el valor numérico; lo que pasa es que no se da este valor expresamente a menos que se indique que se quiere la representación decimal de la imagen o se dé la antiimagen en representación de coma flotante. También se puede obtener el valor de una función en un punto asignando un valor a la variable de la función (que más adelante habrá que eliminar si se quiere seguir trabajando con la función). Por ejemplo:

```
(%i14) x:5.23;      t:4.321;      u:6.543;
       f(x);        f2(t);        f3(u);
(%o14) 5.23
(%o15) 4.321
(%o16) 6.543
(%o17) -38.01580000000001
(%o18) 0.050836150460975
(%o19) 2.020619980871494
```

Hay que recordar la notación de las funciones elementales predefinidas en wxMaxima (véase la Tabla 02-1 del Tema 2).

Operaciones algebraicas con funciones. Para efectuar operaciones algebraicas (suma, diferencia, producto por un escalar, producto y cociente), la sintaxis es bastante sencilla. Por ejemplo:

- Suma y diferencia:

```
(%i1) f1(x):=2*cos(3*x)+4*sin(2*x)$      f2(x):=1/(1+x^2)$      f3(x):=log(x+1):
```

```
(%i4) f10(x):=f1(x)+f3(x);                'f10(x)=f10(x);
(%o4) f10(x):= f1(x)+f3(x)
```

(%o5) $f10(x)=\log(x+1)+2\cos(3x)+4\sin(2x)$

```
(%i6) f11(x):=f1(x)-f2(x);                'f11(x)=f11(x);
(%o6) f11(x):= f1(x)-f2(x)
```

(%o7) $f11(x)=2\cos(3x)+4\sin(2x)-\frac{1}{x^2+1}$

- Producto por un escalar y combinación lineal:

```
(%i8) f12(x):=3.456*f1(x);                          'f12(x)=f12(x);
      f13(x):=1.234*f1(x)-3.456*f3(x);              'f13(x)=f13(x);
(%o8) f12(x):= 3.456 f1(x)
```

(%o9) $f12(x)=3.456\,(2\cos(3x)+4\sin(2x))$

(%o10) $f13(x):=1.234\,f1(x)-3.456\,f3(x)$

(%o11) $f13(x)=1.234\,(2\cos(3x)+4\sin(2x))-3.456\log(x+1)$

- Producto y cociente:

```
(%i12) f14(x):=f1(x)*f2(x);               'f14(x)=f14(x);
(%o12) f14(x):= f1(x) f2(x)
```

(%o13) $f14(x)=\frac{2\cos(3x)+4\sin(2x)}{x^2+1}$

```
(%i14) f15(x):=f1(x)/f2(x);               'f15(x)=f15(x);
```

(%o14) $f15(x):=\frac{f1(x)}{f2(x)}$

(%o15) $f15(x)=(x^2+1)(2\cos(3x)+4\sin(2x))$

Composición de funciones. La sintaxis para la composición de funciones en wxMaxima, que se puede ver a continuación, no usa el signo habitual de composición, sino que se escribe la imagen como "función de la función". Obsérvese que se puede hacer la composición de más de dos funciones:

```
(%i1) f1(x):=2*x+5$        f2(x):=sqrt(3+x^2)$        f3(x):=sin(x)$
(%i4) F1(x):=f1(f2(x));                'F1(x)=F1(x);
(%o4) F1(x):= f1(f2(x))
```

(%o5) $F1(x)=2\sqrt{x^2+3}+5$

```
(%i6) F2(x):=f2(f1(x));                'F2(x)=F2(x);
(%o6) F2(x):= f2(f1(x))
```

$$(\%o7)\ F2(x)=\sqrt{(2\,x+5)^2+3}$$

```
(%i8) F3(x):=f1(f2(f3(x)));            'F3(x)=F3(x);
(%o8) F3(x):= f1(f2(f3(x)))
```

$$(\%o9)\ F3(x)=2\sqrt{\sin(x)^2+3}+5$$

El ejemplo sirve adicionalmente para recordar que la composición de funciones no cumple la propiedad conmutativa.

Funciones definidas a trozos o intervalos. Se pueden considerar y trabajar con funciones definidas a trozos o intervalos. La metodología consiste en designar con una referencia cada una de las diferentes expresiones y después organizar las alternativas en una instrucción. Así, por ejemplo, la función definida por:

$$F_1(x)=\begin{cases} 2x+1, & \text{si } x<0 \\ \sqrt{1+x^2}, & \text{si } x\geq 0 \end{cases}$$

se define en wxMaxima de la manera siguiente:

```
(%i1) f1(x):=2*x+1;              f2(x):=sqrt(1+x^2);
      F1(x):= if x<0 then f1(x) else f2(x);
(%o1) f1(x):=2 x+1
```

$$(\%o2)\ f2(x):=\sqrt{1+x^2}$$

```
(%o3) F1(x):=if  x<0  then  f1(x)  else  f2(x)
```

Si la función está definida en tres intervalos la sintaxis es igualmente sencilla; por ejemplo, la función:

$$G_1(x)=\begin{cases} e^{x/2}, & x\leq 0 \\ 1+x\log(x/2), & 0<x<2 \\ e^{2-x}, & x\geq 2 \end{cases}$$

se define en wxMaxima de la manera siguiente:

```
(%i4) g1(x):=exp(x/2)$    g2(x):=1+x*log(x/2)$     g3(x):=exp(2-x)$
(%i7) G1(x):= if x<=0 then g1(x) else (if x<=2 then g2(x) else g3(x));
(%o7) G1(x):=if  x<=0  then  g1(x)  else  if  x<=2  then  g2(x)  else  g3(x)
```

06-2.- Representación gráfica de funciones. Representación gráfica de curvas planas.

Los contenidos de este apartado se desarrollan en el fichero **Tema_06-2.wxm**.

La representación gráfica de funciones se realiza con la opción “Gráficos” del menú principal. Antes de hacer la representación gráfica hay que definir la función y después ir a “Plot” y decidir por una de las opciones que aparecen: Gráficos2D, Gráficos3D, Formato de gráficos. En el caso de funciones reales de variable real hay que ir a Gráficos2D; el Formato de gráficos se recomienda dejarlo con las opciones por defecto. El cuadro de diálogo de Gráficos2D es el representado en la Figura 06-1.

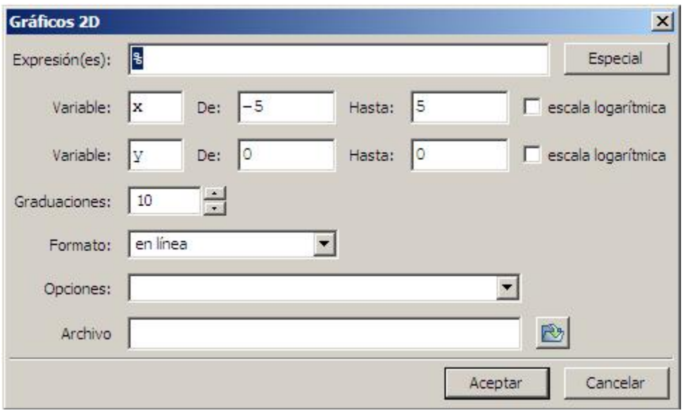


Figura 06-1. Cuadro de diálogo para gráficos en 2D.

Comentemos las indicaciones y consejos para cumplimentar el cuadro de diálogo:

- En la opción “Expresión(es)” hay que escribir el símbolo asignado a la función o funciones que se quieren representar, ya que se pueden incluir varias funciones en un mismo gráfico.
- En la opción “Variable” hay que especificar el rango de la variable; se pueden hacer pruebas para obtener la zona de gráfico más adecuada para la representación que se quiere obtener. No hace falta, si no se quiere, especificar el rango de las ordenadas.
- En la opción “Graduaciones” es para definir la escala de los ejes.
- En la opción “Formato” se puede optar por “predeterminado” o “en línea”. La primera, que se puede ejecutar con la instrucción “plot(...)”, muestra un gráfico en formato llamado “gnuplot” en el mismo documento de wxMaxima, que aparece cuando se ejecuta la instrucción y desaparece al ejecutar la instrucción siguiente; en este gráfico, yendo a la parte superior y usando el botón derecho del ratón, se acede a un menú de modificación de las opciones del gráfico que permite personalizarlo. En la segunda opción, más recomendable si se quiere mantener la gráfica visible, se incluye el gráfico enganchado a la salida y con el botón derecho del ratón se puede copiar y pegar después en otro documento. La opción de la gráfica “en línea” se puede ejecutar con la instrucción “wxplot(...)”. Hay una opción adicional que es

"openmath" que permite editar y manipular el gráfico, además de guardarlo en un fichero.

- En "Opciones" hay las opciones de gráfico: setzeroaxis, set size ratio 1, set grid. Estas opciones afectan el dibujo de los ejes de coordenadas, la medida relativa de las escalas en los ejes y al dibujo de una malla.
- La opción "Gráfico al archivo" permite exportar el gráfico a un fichero que en el botón que hay más a la derecha permite especificar el nombre del fichero y la carpeta de destino. El formato de los gráficos es postcript encapsulado, eps. Hay que tener pues un programa que permita abrir este tipo de archivo y, si es necesario, modificar después el formato gráfico.

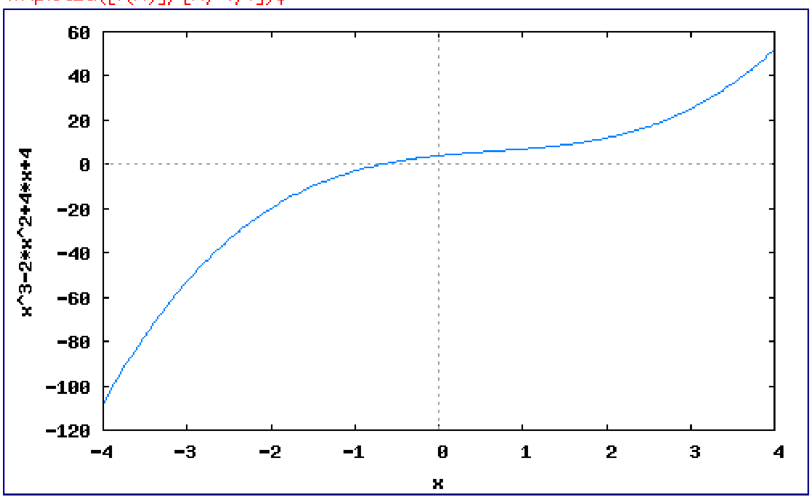
Veamos con un ejemplo como afectan las opciones en el resultado que da el programa. Se observará que en todos los casos se usa la instrucción "wxplot(...)" que permite tener la gráfica "pegada" al mismo archivo de wxMaxima. El aspecto de la gráfica se puede adaptar de alguna manera a las preferencias o gustos del usuario y a las necesidades objetivas de la representación gráfica.

(%i1) f(x):=x^3-2*x^2+4*x+4;   g(x):=1+x^2;

(%o1) $f(x):=x^3-2x^2+4x+4$

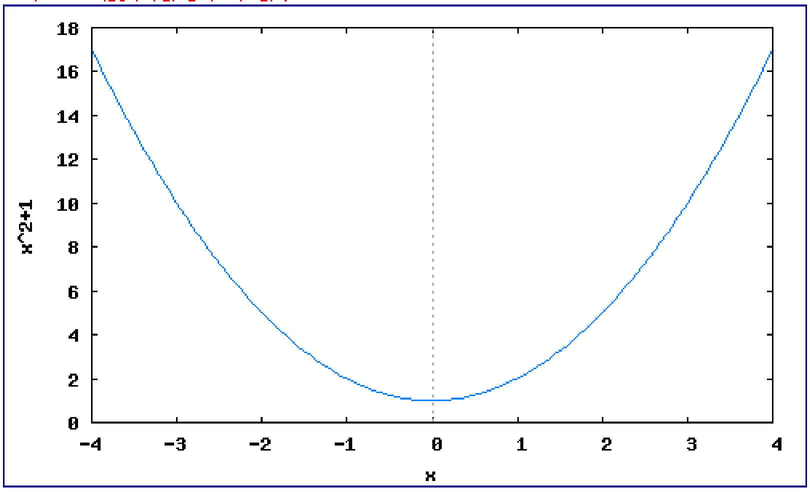
(%o2) $g(x):=1+x^2$

(%i3) wxplot2d([f(x)], [x,-4,4])$

(%t3) 

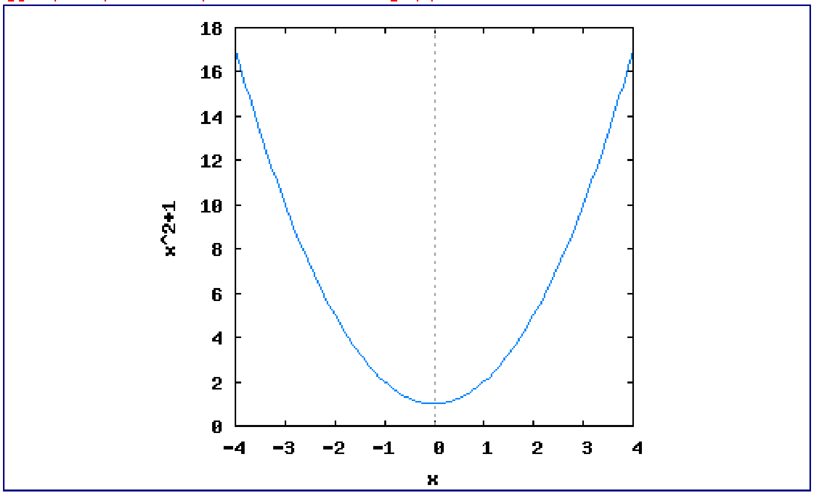
(%i4) wxplot2d([g(x)], [x,-4,4])$

(%t4)



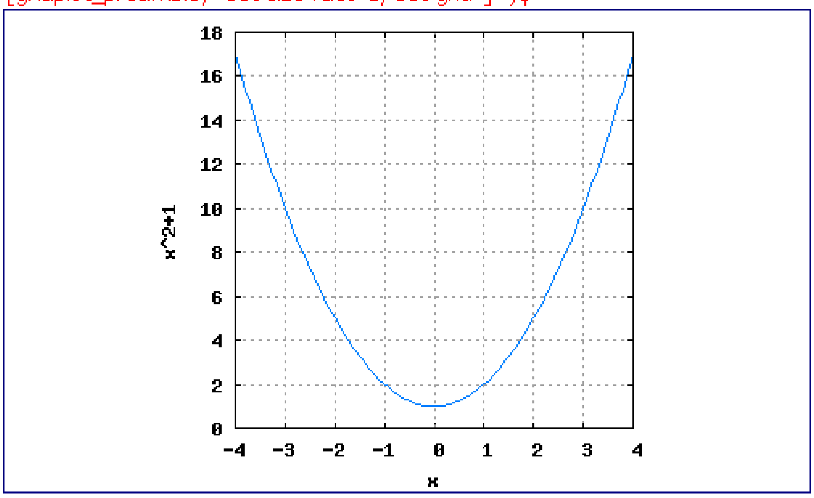
(%i5) wxplot2d([g(x)], [x,-4,4] ,
[gnuplot_preamble, "set size ratio 1"])$

(%t5)



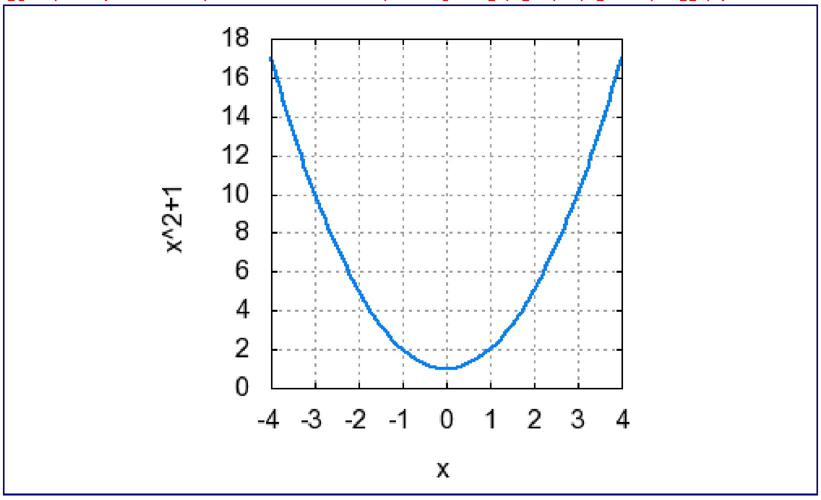
(%i6) wxplot2d([g(x)], [x,-4,4] ,
[gnuplot_preamble, "set size ratio 1; set grid"])$

(%t6)



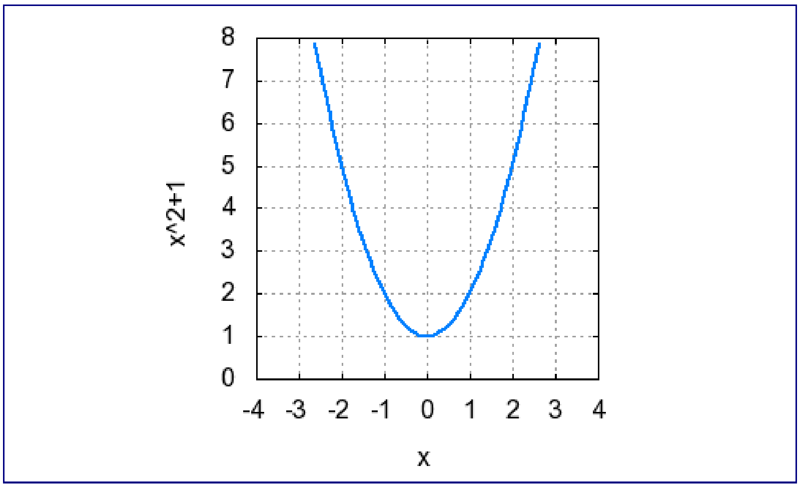
```
(%i7) wxplot2d([g(x)], [x,-4,4] ,
      [gnuplot_preamble, "set size ratio 1; set grid"] , [style, [lines, 2]] )$
```

(%t7)




```
(%i8) wxplot2d([g(x)], [x,-4,4] , [y, 0, 8] ,
                  [gnuplot_preamble, "set size ratio 1; set grid; "] , [style, [lines, 2]] )$
plot2d: some values were clipped.
```

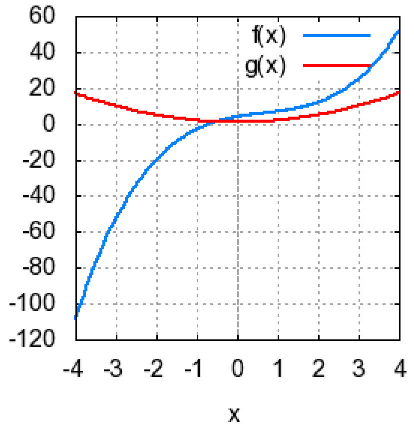
(%t8)



Se pueden incluir en un mismo gráfico la representación de dos funciones, como se puede ver a continuación. Para hacerlo, hay que simplemente escribir en “Expresión(es)” los símbolos de las dos funciones, separadas por una coma.

```
(%i9) wxplot2d([f(x),g(x)], [x,-4,4] ,
      [gnuplot_preamble, "set size ratio 1; set grid; "] , [legend, "f(x)" , "g(x)"] , [style, [lines, 2]] )$
```

(%t9)



Las funciones definidas a intervalos también se pueden representar gráficamente. Por ejemplo, si se quiere obtener la representación gráfica de la función definida por:

$$F_1(x) = \begin{cases} 2x+1, & \text{si } x < 0 \\ \sqrt{1+x^2}, & \text{si } x \geq 0 \end{cases}$$

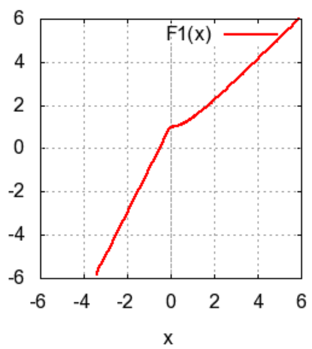
Primero hay que definir las expresiones de la función en cada uno de los intervalos, a continuación establecer cada una de las expresiones asociadas al intervalo correspondiente y finalmente ejecutar la instrucción de representación gráfica tal como se ha indicado. Así, en este caso, sería:

```
(%i1) f1(x):=2*x+1$
      f2(x):=sqrt(1+x^2)$
      F1(x):= if x<0 then f1(x) else f2(x);
(%o3) F1(x):=if  x<0  then  f1(x)  else  f2(x)
```

Y ahora ya se puede obtener la representación gráfica:

```
(%i4) wxplot2d([F1(x)], [x,-6,6] , [y, -6, 6],
      [gnuplot_preamble, "set size ratio 1; set grid; "] , [legend, "F1(x)"] , [style, [lines, 2, 2]] , [ylabel, ""] )$
plot2d: some values were clipped.
```

(%t4)



Si se considera la función definida por:

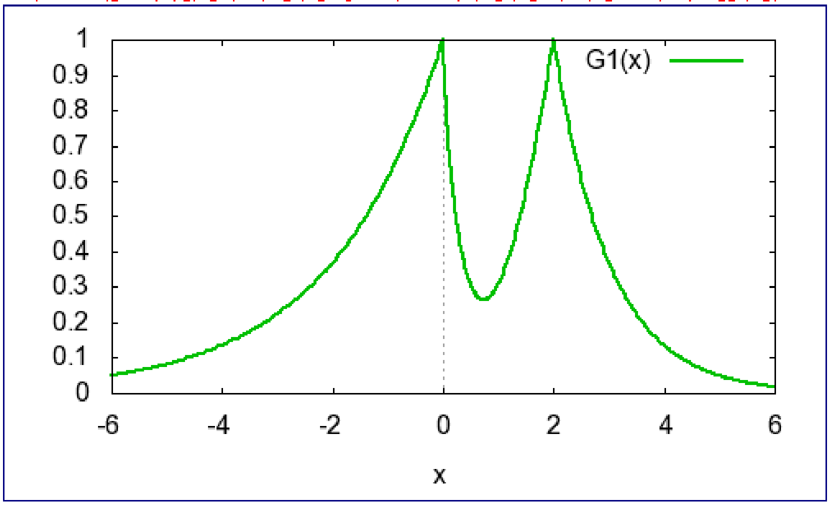
$$G_1(x) = \begin{cases} e^{x/2}, & x \le 0 \\ 1 + x\log(x/2), & 0 < x < 2 \\ e^{2-x}, & x \ge 2 \end{cases}$$

Entonces:

```
(%i5) g1(x):=exp(x/2)$
      g2(x):=1+x*log(x/2)$
      g3(x):=exp(2-x)$
      G1(x):= if x<=0 then g1(x) else (if x<=2 then g2(x) else g3(x));
(%o8) G1(x):=if  x≤0  then  g1(x)  else  if  x≤2  then  g2(x)  else  g3(x)
```

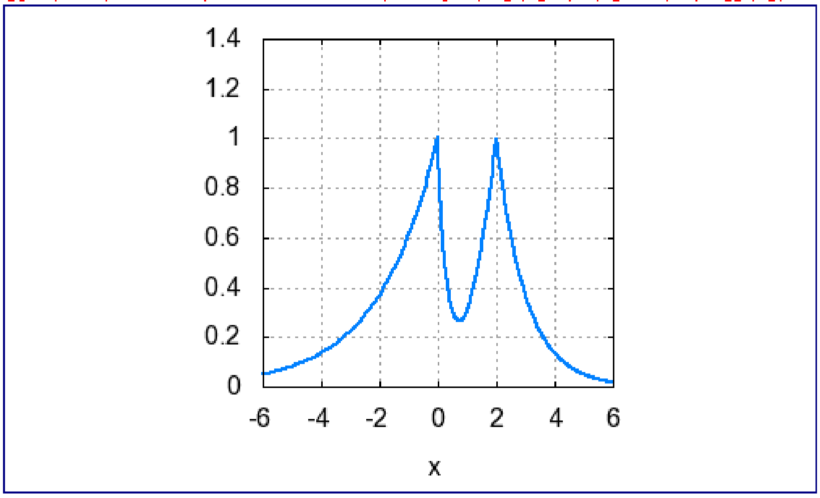
```
(%i9) wxplot2d([G1(x)], [x,-6,6] , [legend, "G1(x)"] , [style, [lines, 2, 3]] , [ylabel, ""])$
```

(%t9)




```
(%i10) wxplot2d([G1(x)], [x,-6,6] , [y, 0, 1.4],
       [gnuplot_preamble, "set size ratio 1; set grid; "] , [style, [lines, 2, 1]] , [ylabel, ""]  )$
```

(%t10)

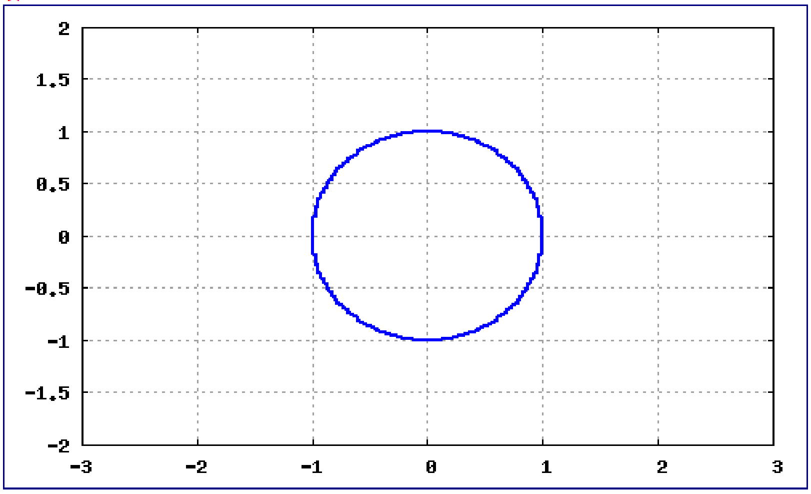


Para la representación gráfica de curvas planas dadas por una ecuación, es necesario utilizar la instrucción draw2d, tal y como se puede ver a continuación. Para más detalles sobre las opciones, consulte el manual del programa.

Representación gráfica de la circunferencia de centro el origen y radio unidad:

```
(%i49) load(draw)$
       wxdraw2d(
         grid = true,    line_type = solid,  line_width = 2, color = blue,
          implicit(x^2+y^2-1=0, x, -3.0,3.0, y, -2,2)
       );
```

(%t50)

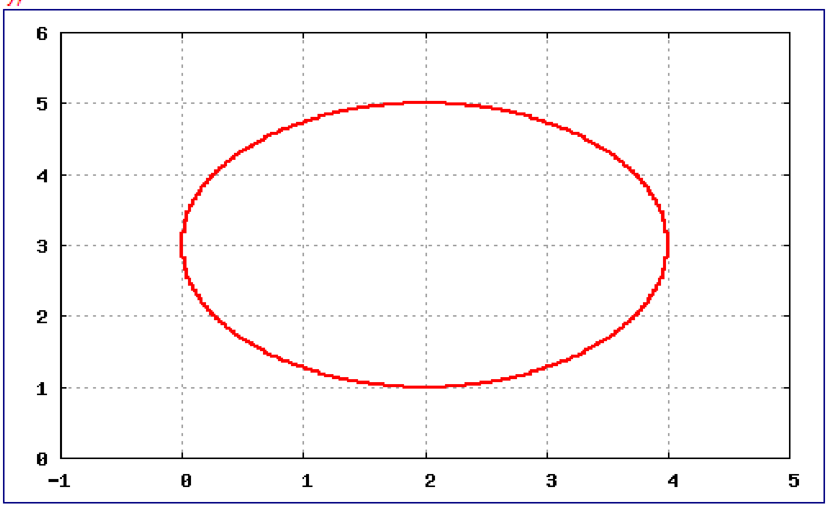


(%o50) [gr2d(implicit)]

Representación gráfica de la circunferencia de centro el punto (2,3) y radio 2:

```
(%i39) load(draw)$
       wxdraw2d(
         grid = true,    line_type = solid,  line_width = 2, color = red,
          implicit((x-2)^2+(y-3)^2-4=0, x, -1,5, y, 0,6)
       );
```

(%t40)



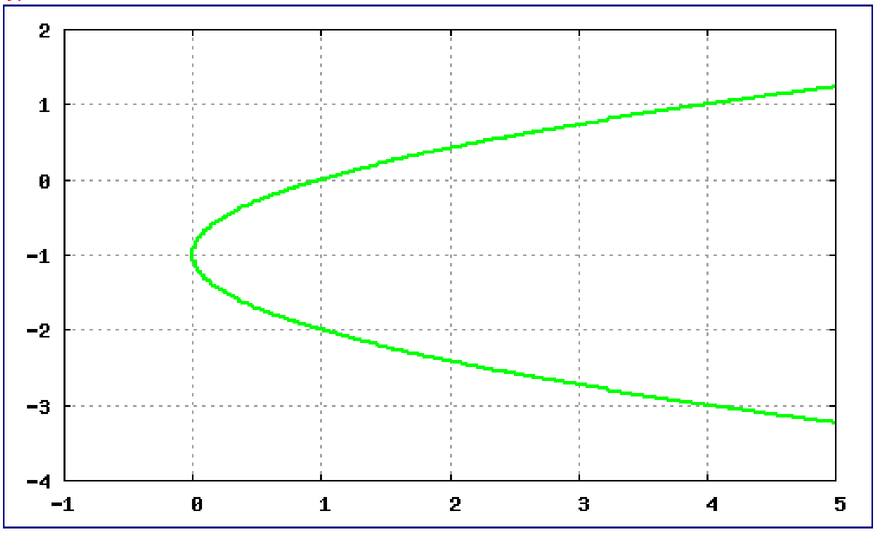
(%o40) [gr2d(implicit)]

Como se puede ver, la representación gráfica se hace por defecto en un formato que puede parecer extraño, ya que la escala de los ejes no es la misma y la curva puede aparecer deformada. Para corregir este efecto se puede copiar la gráfica haciendo clic con el ratón sobre la gráfica y pulsando la opción de copiar la imagen, una vez copiada y pegada en un documento, se puede conseguir una escala que permita una visualización más adecuada, como se puede ver con la circunferencia anterior.

Representación gráfica de la parábola de ecuación $x = y^2 + 2y + 1$:

```
(%i47) load(draw)$
       wxdraw2d(
         grid = true,    line_type = solid,  line_width = 2, color = green,
          implicit(y^2+2*y+1-x=0, x, -1,5, y, -4,2)
       );
```

(%t48)



(%o48) [gr2d(implicit)]

06-3.- Límites de funciones. Cálculo y propiedades

Los contenidos de este apartado se desarrollan en el fichero **Tema_06-3.wxm**.

Recordemos la definición de límite de una función en un punto. Si $f: A \subset \mathbb{R} \to \mathbb{R}$ y $a \in \mathbb{R}$ es un punto de acumulación del conjunto *A*, se dice que $L \in \mathbb{R}$ és el límite de la función f en el punto $a \in \mathbb{R}$, si cualquiera que sea $\varepsilon > 0$ existe $\delta > 0$ tal que siempre que $x \in A$ y $0 < |x - a| < \delta$, entonces se cumple $|f(x) - L| < \varepsilon$. Notación: $L = \lim\limits_{x \to a} f(x)$ o bien $L = \lim\limits_{a} f$. Si en la definición se substituye la condición $0 < |x - a| < \delta$ por $a < x < a + \delta$, entonces se tienen el límite lateral por la derecha de la función en el punto y si se substituye por $a - \delta < x < a$ entonces se tiene el límite lateral per la izquierda. El límite de una función en un punto existe si y sólo si, los límites laterales existen y son iguales.

El cálculo del límite de una función en un punto se lleva a cabo con el menú "Análisis/Calcular límite". Cuando se aplica esta opción, se abre el cuadro de diálogo de la Figura 06-5. En este cuadro de diálogo hay que especificar:

- la referencia de la función o su expresión analítica;
- la variable;
- el punto en el que se quiere calcular el límite de la función; hay una opción designada "Especial" en la que se puede indicar el número π, el número *e*, el infinito o el menos infinito;
- si el límite se quiere en el punto ("ambos lados") o lateral (izquierda o derecha).

Figura 06-5. Cuadro de diálogo para calcular el límite de una función en un punto.

Por ejemplo, si se consideran las funciones

$$f_1(x) = \sin(x)\,; \qquad f_2(x) = 1 + \log(x+1)$$

planteamos el cálculo del límite de la función $f2$ en el punto $a = 1$ así como los límites laterales:

```
(%i1) f1(x):=sin(x)$     f2(x):=1+log(x+1)$
      L1=limit(f2(x), x, 1);
      L2=limit(f2(x), x, 1, minus);
      L3=limit(f2(x), x, 1, plus);
```

(%o3) $L1 = \log(2) + 1$

(%o4) $L2 = \log(2) + 1$

(%o5) $L3 = \log(2) + 1$

Si se quiere una salida más completa, hay que indicar a wxMaxima que escriba el límite que se quiere calcular, el signo "=" y a continuación la instrucción de cálculo del límite. Así, en el caso anterior el procedimiento sería:

```
(%i6) 'limit(f2(x), x, 1)=limit(f2(x), x, 1);
      'limit(f2(x), x, 1, minus)=limit(f2(x), x, 1, minus);
      'limit(f2(x), x, 1, plus)=limit(f2(x), x, 1, plus);
```

(%o6) $\lim\limits_{x \to 1} \log(x+1) + 1 = \log(2) + 1$

(%o7) $\lim\limits_{x \to 1-} \log(x+1) + 1 = \log(2) + 1$

(%o8) $\lim\limits_{x \to 1+} \log(x+1) + 1 = \log(2) + 1$

Consideremos ahora la función definida a intervalos por:

$$F_1(x) = \begin{cases} \sin(x), & \text{si } x \le 0 \\ 1 + \log(x+1), & \text{si } x > 0 \end{cases}$$

y planteamos el cálculo del límite lateral por la izquierda en el punto $a = 0$; la sintaxis que en primera instancia utilizaríamos es:

```
(%i9) F1(x):= if x<=0 then f1(x) else f2(x);     'limit(F1(x), x, 0, minus)=limit(F1(x), x, 0, minus)
```

(%o9) $F1(x) := \text{if } x \le 0 \text{ then } f1(x) \text{ else } f2(x)$

(%o10) $\lim\limits_{x \to 0-} (\text{if } x \le 0 \text{ then } f1(x) \text{ else } f2(x)) = \lim\limits_{x \to 0-} \text{if } x \le 0 \text{ then } f1(x) \text{ else } f2(x)$

que, como se puede ver, no da el resultado deseado. Esto es porque en este tipo de funciones hay que escribir el símbolo de cada una de las expresiones analíticas que se utilizan en la definición de la función definida en intervalos.

Por lo tanto, la sintaxis correcta es:

```
(%i11) 'limit(f1(x), x, 0, minus)=limit(f1(x), x, 0, minus);
       'limit(f2(x), x, 0, plus)=limit(f2(x), x, 0, plus);
```

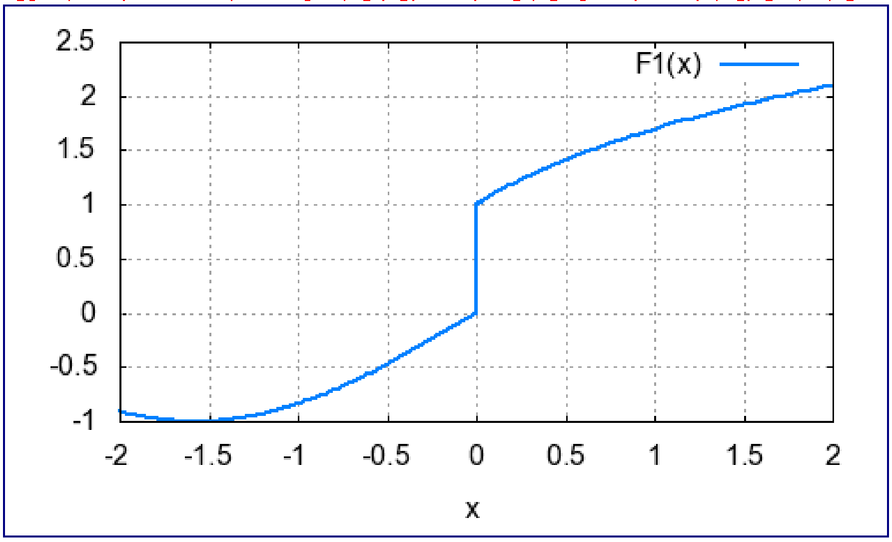
(%o11) $\lim_{x \to 0-} \sin(x) = 0$

(%o12) $\lim_{x \to 0+} \log(x+1)+1 = 1$

La representación gráfica que ilustra esta situación obtenida en wxMaxima es:

```
(%i13) wxplot2d([F1(x)], [x,-2,2],
       [gnuplot_preamble, "set grid;"] , [ylabel, ""] , [legend, "F1(x)"], [style, [lines, 2]]  )$
```

(%t13)



Consideremos ahora la función definida por:

$$f_3(x) = \frac{|x|}{x}, \; x \neq 0$$

El punto $a = 0$ no es del campo de existencia de esta función, pero es un punto de acumulación de este conjunto. Por lo tanto, tiene sentido calcular el límite de la función en este punto:

```
(%i14) f3(x):=(abs(x))/x;
```

(%o14) $f3(x) := \frac{|x|}{x}$

```
(%i15) 'limit(f3(x), x, 1)=limit(f3(x), x, 1);
```

(%o15) $\lim_{x \to 1} \frac{|x|}{x} = 1$

(%i16) 'limit(f3(x), x, 0)=limit(f3(x), x, 0);

(%o16) $\lim\limits_{x \to 0} \frac{|x|}{x} = und$

La respuesta "und", que es una abreviatura de "undefined" significa que el límite no existe. No obstante, podemos plantear el cálculo de los límites laterales en este punto y observar que son diferentes. La gráfica de la función se representa con la salida que da wxMaxima.

(%i17) 'limit(f3(x), x, 0, minus)=limit(f3(x), x, 0, minus);

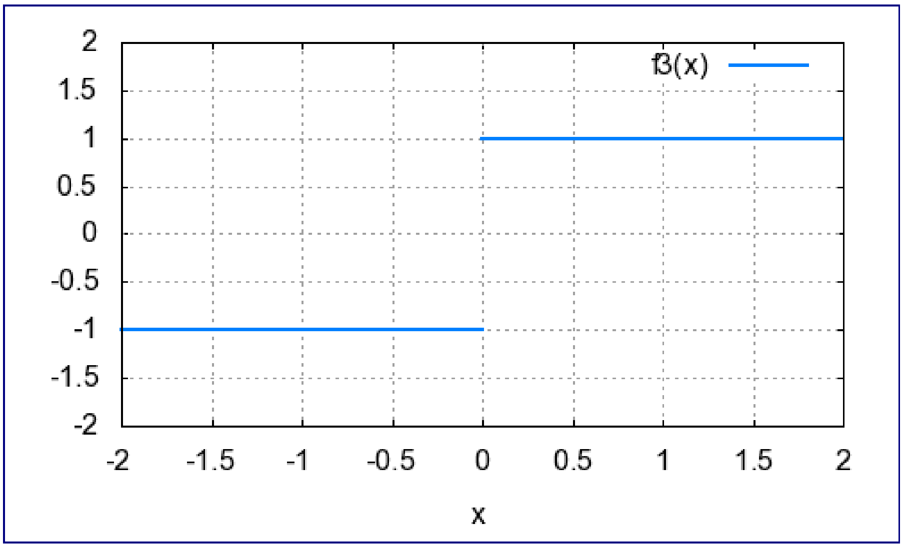
(%o17) $\lim\limits_{x \to 0-} \frac{|x|}{x} = -1$

(%i18) 'limit(f3(x), x, 0, plus)=limit(f3(x), x, 0, plus);

(%o18) $\lim\limits_{x \to 0+} \frac{|x|}{x} = 1$

(%i19) wxplot2d([f3(x)], [x,-2,2], [y, -2, 2], [ylabel, ""] , [legend, "f3(x)"], [style, [lines, 2]] , [gnuplot_preamble, "set grid;"])$

plot2d: expression evaluates to non-numeric value somewhere in plotting range.

(%t19) 

Un comentario adicional sobre cálculo de límites de funciones: se pueden calcular límites de funciones con parámetros en su expresión analítica y también en un punto genérico o arbitrario, tal como se muestra a continuación:

(%i20) f21(x):=(x+A)*(x^2+B);

(%o20) $f21(x):=(x+A)(x^2+B)$

(%i21) 'limit(f21(x), x, 1)=limit(f21(x), x, 1);

(%o21) $\lim\limits_{x \to 1} (A+x)(B+x^2) = (A+1)B+A+1$

(%i22) 'limit(f21(x), x, a)=limit(f21(x), x, a);

(%o22) $\lim_{x\to a} (A+x)(B+x^2)=(A+a)B+a^2 A+a^3$

Límites infinitos. Como sabemos, el límite de una función en un punto se generaliza a valores del límite de la recta real ampliada, es decir, límites infinitos. En el ejemplo siguiente se muestra la respuesta del programa en este caso:

(%i1) f1(x):=1/(x-1)^2;

(%o1) $f1(x):=\frac{1}{(x-1)^2}$

(%i2) 'limit(f1(x), x, 1)=limit(f1(x), x, 1);

(%o2) $\lim_{x\to 1} \frac{1}{(x-1)^2}=\infty$

Puede que en un caso la respuesta sea que el límite no existe, pero sí que exista uno de los límites laterales y sea infinito, como se puede ver en el ejemplo siguiente:

(%i3) 'limit(log(x), x, 0)=limit(log(x), x, 0);
'limit(log(x), x, 0, plus)=limit(log(x), x, 0, plus);

(%o3) $\lim_{x\to 0} \log(x)=\text{infinity}$

(%o4) $\lim_{x\to 0+} \log(x)=-\infty$

Recordemos que cuando una función tiene límite infinito o alguno de los límites laterales infinitos en un punto *a*, la recta de ecuación $x=a$ se dice que es una asíntota vertical de la gráfica de la función.

En ocasiones la respuesta del programa puede resultar sorprendente, pero en ningún caso debe ser ser engañosa en su interpretación. Tal es el caso de la función $f(x)=1/x,\ x\neq 0$ en el origen:

(%i5) f2(x):=1/x;

(%o5) $f2(x):=\frac{1}{x}$

(%i6) 'limit(f2(x), x, 0)=limit(f2(x), x, 0);

(%o6) $\lim_{x\to 0} \frac{1}{x}=\text{infinity}$

(%i7) 'limit(f2(x), x, 0, plus)=limit(f2(x), x, 0, plus);

(%o7) $\lim_{x \to 0+} \frac{1}{x} = \infty$

(%i8) 'limit(f2(x), x, 0, minus)=limit(f2(x), x, 0, minus);

(%o8) $\lim_{x \to 0-} \frac{1}{x} = -\infty$

Límites en el infinito. Los límites en $+\infty$ y en $-\infty$ se calculan con una sintaxis parecida a la de la instrucción de cálculo del límite de una función, especificando "inf" y "minf", respectivamente; por ejemplo:

(%i9) 'limit(exp(x), x, minf)=limit(exp(x), x, minf);
'limit(exp(x), x, inf)=limit(exp(x), x, inf);

(%o9) $\lim_{x \to -\infty} \%e^{x} = 0$

(%o10) $\lim_{x \to \infty} \%e^{x} = \infty$

(%i11) 'limit(2-exp(-x), x, minf)=limit(2-exp(-x), x, minf);
'limit(2-exp(-x), x, inf)=limit(2-exp(-x), x, inf);

(%o11) $\lim_{x \to -\infty} 2-\%e^{-x} = -\infty$

(%o12) $\lim_{x \to \infty} 2-\%e^{-x} = 2$

Como se puede ver, en algunos casos uno de estos límites puede ser también infinito. Cuando el límite en el infinito da un valor finito L, la recta de ecuación $y = L$ se dice que es una asíntota horizontal de la gráfica de la función, que lo será en el lado de la recta real correspondiente.

Hay funciones que tienen límites finitos en ambos casos del infinito de la recta real, como por ejemplo la función definida por:

$$g_1(x) = 1 + \frac{1}{1+x^2}$$

(%i13) g1(x):=1+1/(1+x^2);

(%o13) $g1(x) := 1 + \frac{1}{1+x^2}$

(%i14) 'limit(g1(x), x, minf)=limit(g1(x), x, minf);

(%o14) $\lim_{x \to -\infty} \frac{1}{x^2+1} + 1 = 1$

```
(%i15) 'limit(g1(x), x, inf)=limit(g1(x), x, inf);
```

(%o15) $\lim_{x \to \infty} \frac{1}{x^2+1} + 1 = 1$

Finalmente, el límite en el infinito se puede aplicar al cálculo de las asíntotas oblicuas de una función, que como se sabe son rectas que cumplen la condición

$$\lim_{\infty}\left(f(x) - \left(mx + b\right)\right) = 0$$

El cálculo de los coeficientes de la asíntota se lleva a cabo mediante las expresiones:

$$m = \lim_{\infty} \frac{f(x)}{x}; \quad b = \lim_{\infty}\left(f(x) - mx\right)$$

Calculamos, por ejemplo, la asíntota oblicua de la función $f(x) = x + \frac{1}{x},\ x \neq 0$:

```
(%i16) g2(x):=x+1/x;
       m1=m1:limit(g2(x)/x, x, minf);
       m2=m2:limit(g2(x)/x, x, inf);
```

(%o16) $g2(x) := x + \frac{1}{x}$

(%o17) $m1 = 1$

(%o18) $m2 = 1$

```
(%i19) b1=b1:limit((g2(x)-m1*x), x, minf);
       b2=b2:limit((g2(x)-m2*x), x, inf);
```

(%o19) $b1 = 0$

(%o20) $b2 = 0$

```
(%i21) y(x):=m1*x+b1;  'y=y(x);
```

(%o21) $y(x) := m1\, x + b1$

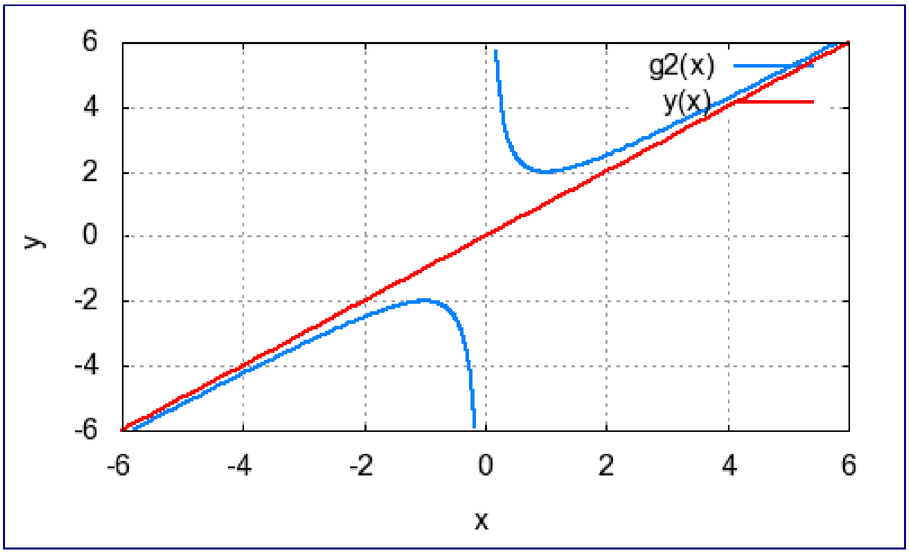
(%o22) $y = x$

La representación gráfica que da wxMaxima de la gráfica de esta función es la que se puede ver a continuación.

```
(%i23) wxplot2d([g2(x),y(x)], [x,-6,6], [y,-6,6],  [legend, "g2(x)", "y(x)"], [style, [lines, 2]] ,
          [gnuplot_preamble, "set grid;"]
         )$
```

plot2d: expression evaluates to non-numeric value somewhere in plotting range.
plot2d: some values were clipped.

(%t23)



06-4.- Funciones continuas. Estudio de la continuidad

Los contenidos de este apartado se desarrollan en el fichero **Tema_06-4.wxm**.

Recordemos la definición de función continua en un punto. Si $f : A \subset \mathbb{R} \to \mathbb{R}$ y $a \in A$ es un punto no aislado, se dice que la función es continua en este punto si se cumple la condición $\lim_{x \to a} f(x) = f(a)$. Esta definición implica que

- el límite de la función en el punto existe y es finito;
- el límite de la función en el punto es igual a la imagen del punto por la función.

Si una función no es continua en un punto se dice que es discontinua. Las discontinuidades se clasifican en:

- discontinuidad evitable: si el límite existe pero es diferente de la imagen;
- discontinuidad de salto: si los límites laterales existen, son finitos y diferentes;
- discontinuidad esencial: cuando uno de los límites laterales o los dos no existen o no son finitos.

El estudio de la continuidad se llevará a cabo aplicando los criterios de continuidad de funciones y, en todo caso, ayudándose o completando el estudio con unos cálculos de límites o representaciones gráficas. No hay ninguna instrucción de wxMaxima que dé una respuesta a una posible pregunta sobre la continuidad de una función en un punto.

- Ejemplo de discontinuidad evitable: la función definida por

$$f(x) = \begin{cases} \exp(x), & \text{si } x \neq 0 \\ 2, & \text{si } x = 0 \end{cases}$$

La gráfica se puede ver en la figura 06-6. Es inmediato verificar que se cumple:

$$\lim_{x \to 0-} f(x) = \lim_{x \to 0+} f(x) = \lim_{x \to 0} f(x) = 1 \neq f(0) = 2$$

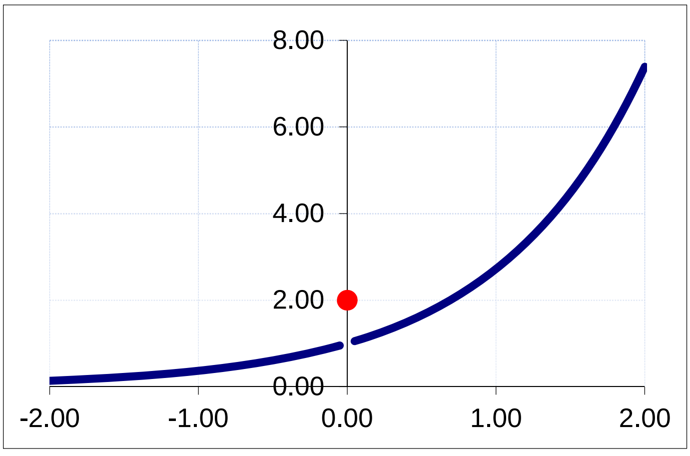


Figura 06-6. Ilustración de la discontinuidad evitable

- Ejemplo de discontinuidad de salto: la función definida por

$$f(x) = \begin{cases} x^2, & \text{si } x < 0 \\ x^2 + 2, & \text{si } x \geq 0 \end{cases}$$

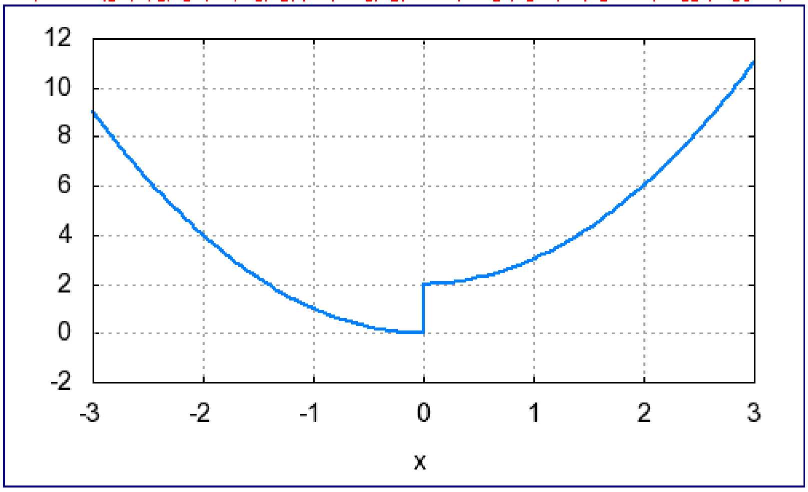
En wxMaxima la función se define mediante:

```
(%i1) f1(x):=x^2;    f2(x):=x^2+2;     f(x):= if x<0 then f1(x) else f2(x);
(%o1) f1(x):=x^2
(%o2) f2(x):=x^2+2
(%o3) f(x):=if x<0 then f1(x) else f2(x)
```

La representación gráfica de la función se puede obtener en wxMaxima:

```
(%i4) wxplot2d([f(x)], [x,-3,3], [y,-2,12], [ylabel, ""] , [style, [lines, 2]] ,  [gnuplot_preamble, "set grid;"])
```

(%t4)



Es inmediato verificar que se cumple:

$$\lim_{x \to 0-} f(x) = 0; \quad \lim_{x \to 0+} f(x) = 2$$

- Ejemplo de discontinuidad esencial: la función definida por

$$f(x) = \begin{cases} x^2, & \text{si } x \leq 0 \\ \sin(1/x), & \text{si } x > 0 \end{cases}$$

En wxMaxima la función se define mediante:

(%i5) f3(x):=sin(1/x);   g(x):= if x<=0 then f1(x) else f3(x);

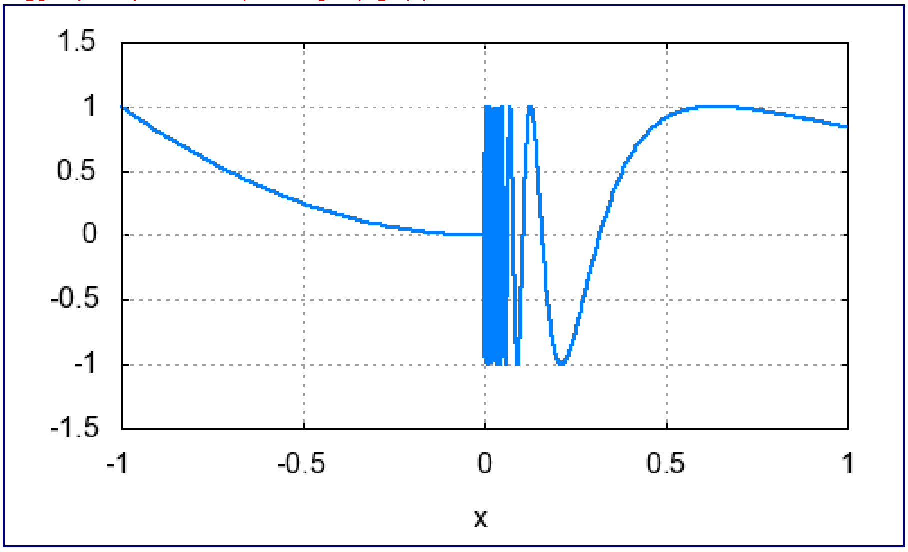
(%o5) $f3(x):=\sin\left(\frac{1}{x}\right)$

(%o6) $g(x):=\text{if } x\leq 0 \text{ then } f1(x) \text{ else } f3(x)$

La representación gráfica de la función se puede obtener en wxMaxima:

(%i7) wxplot2d([g(x)],   [x,-1,1],   [y,-1.5,1.5],   [ylabel, ""] ,   [style, [lines, 2]] ,
[gnuplot_preamble, "set grid;"])$

(%t7)



Es inmediato verificar que se cumple:

$$\lim_{x\to 0-} f(x) = 0; \quad \lim_{x\to 0+} f(x) = \text{no existeix}$$

06-5.- Propiedades de las funciones continuas. Aplicaciones

Los contenidos de este apartado se desarrollan en el fichero **Tema_06-5.wxm**.

Las propiedades más importantes de las funciones continuas en un intervalo cerrado son:

- el teorema de Bolzano;
- el teorema de los valores intermedios;
- el teorema del máximo y del mínimo;
- el teorema de Weierstrass.

El teorema de Bolzano permite afirmar la existencia de al menos una raíz o cero de una función continua en un intervalo cerrado, si la función tiene signo diferente en los extremos. Este teorema permite calcular raíces de ecuaciones que con la instrucción "solve" no se pueden determinar. Por ejemplo, se plantea calcular las raíces de la ecuación

$$\exp(-x) - x = 0$$

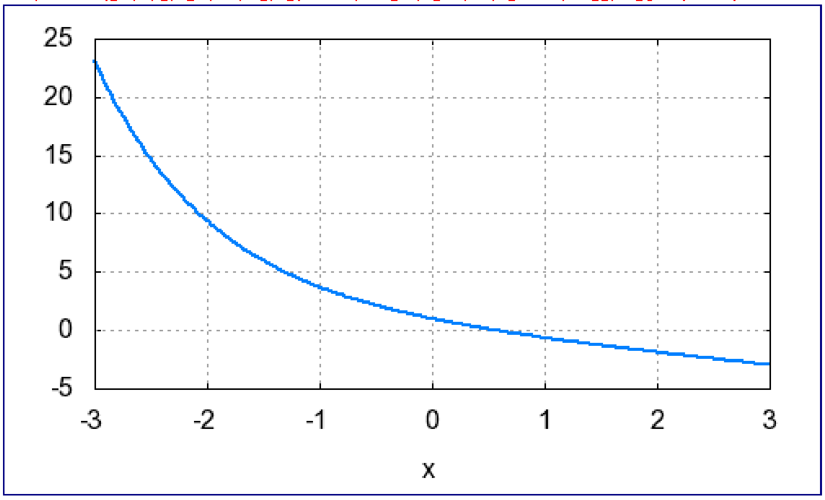
En primer lugar se define la función $f(x) = e^{-x} - x,\ x \in \mathbb{R}$. Observemos que esta función es continua en su campo de existencia. Ahora se plantea resolver la ecuación $f(x)=0$:

```
(%i1) f(x):=exp(-x)-x;        Eq1:f(x)=0;
      solve([Eq01], [x]);
(%o1) f(x):=exp(-x)-x
  (%o2) %e^-x-x=0
  (%o3) []
```

Evidentemente se observa un problema: wxMaxima no da ninguna respuesta y, por tanto, hay que buscar una alternativa. Esta pasa por observar que la función cambia de signo en el intervalo cerrado [-3,3], tal y como se puede ver en la gráfica que se obtiene en wxMaxima:

```
(%i4) wxplot2d([f(x)], [x,-3,3], [ylabel, ""] , [style, [lines, 2]], [gnuplot_preamble, "set grid;"] )$
```

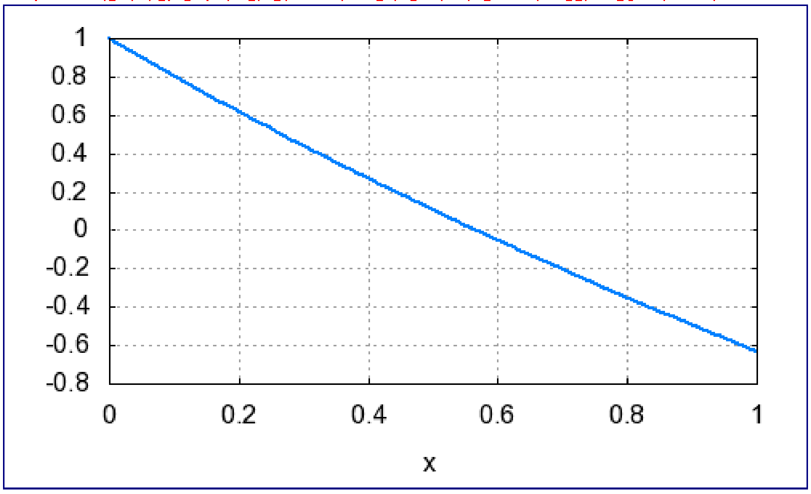
(%t4)



Se puede "afinar más" y afirmar que la función cambia de signo en el intervalo [0,1], centrando la representación gráfica en este intervalo:

```
(%i5) wxplot2d([f(x)], [x,0,1], [ylabel, ""] , [style, [lines, 2]], [gnuplot_preamble, "set grid;"] )$
```

(%t5)



Por tanto, en virtud del Teorema de Bolzano, se puede afirmar que hay una solución en este intervalo. En el menú "Ecuaciones" hay una opción "find root" que aplica un algoritmo numérico de aproximación de la solución de una ecuación o de una función en un intervalo cerrado. En este caso se obtiene:

```
(%i6) find_root(Eq1, x, 0, 2);
(%o6) 0.56714329040978
```

Hay que tener presente que esta opción funciona correctamente cuando el intervalo cerrado que se especifica para encontrar el cero, la función tiene sólo una raíz, de lo contrario da la mayor de las raíces. Por ejemplo, si se considera la función

$$g(x) = x^2 \sin\left(\frac{1}{x}\right), \ x > 0$$

y se observa su gráfica en el intervalo [0.05; 0.2], la respuesta de wxMaxima al cálculo de la raíz es un único resultado, que como se puede ver en el gráfico corresponde al mayor de los ceros de la función en ese intervalo :

```
(%i8) g(x):=x^2*sin(1/x);        Eq2:g(x)=0;
      find_root(Eq2, x, 0.05, 0.2);
```

(%o8) $g(x):=x^2\sin\left(\frac{1}{x}\right)$

(%o9) $\sin\left(\frac{1}{x}\right)x^2=0$

(%o10) 0.1591549430919

Asimismo, hay que asegurarse de que la función cambia de signo en los extremos del intervalo en el que se busca el cero, de lo contrario, wxMaxima lo advierte y no hace ningún cálculo. Por ejemplo, si se observa la gráfica anterior, se ve que la función tiene dos ceros en el intervalo [0.1; 0.2], sin embargo, la respuesta de wxMaxima es:

```
(%i11) find_root(Eq2, x, 0.1, 0.2);
find_root: function has same sign at endpoints: f(0.1)=-0.0054402111088937, f(0.2)=-0.03835697098652
 -- an error. To debug this try: debugmode(true);
```

Si se establece un intervalo en el que hay una única raíz, se obtiene la respuesta correcta:

```
(%i12) find_root(Eq2, x, 0.14, 0.2);
(%o12) 0.1591549430919
```

Representación gráfica de la función en [0.05, 0.20]:

```
(%i13) wxplot2d([g(x)], [x,0.05,0.2],  [ylabel, ""] , [style, [lines, 2]],   [gnuplot_preamble, "set grid;"] )$
```

(%t13)