Prácticas de Mecánica Computacional. Introducción a Maxima y TEXMACS

POR FELIPE GABALDÓN

Correo-e: felipe.gabaldon@upm.es

26 January 2010

1 Introducción

El objetivo de este documento es proporcionar una visión general de las capacidades del programa de Álgebra Simbólica MAXIMA, y del uso de TEXMACS como interfaz de dicho programa. Constituye el material de trabajo de la 1.ª sesión de las Prácticas de Mecánica Computacional que forman parte del curso de *Mecánica*, en la E.T.S. Ingenieros de Caminos de Madrid. El documento se imprime directamente desde la sesión de trabajo de TEXMACS. Asimismo, los cálculos realizados con MAXIMA que se reportan en este documento se pueden reproducir utilizando cualquier otra interfaz de usuario (por ejemplo `xmaxima`).

2 ¿Qué es MAXIMA?

MAXIMA (http://maxima.sourceforge.net) es un programa para hacer cálculo simbólico por ordenador[1] que se distribuye con licencia GPL (http://www.gnu.org/licenses/gpl.html). Proviene de uno de los sistemas CAS más antiguos, denominado MACSYMA. Éste programa comienza a desarrollarse en el MIT a partir de los años 60. En 1982 el MIT decide comercializar MACSYMA, denominándose MAXIMA a una de las versiones de MACSYMA, cuyo desarrollo se realiza en la Universidad de Texas bajo la dirección del profesor William Schelter. Posteriormente, a finales de los años 80, comienzan a comercializarse sistemas CAS propietarios como MAPLE y MATHEMATICA. En 1998 el profesor Schelter consigue la autorización del gobierno americano para distribuir el código fuente de MAXIMA con licencia GPL, y tras su muerte en 2001 se forma un grupo de voluntarios para desarrollar y distribuir MAXIMA como software libre.

Existen diferentes interfaces de usuario para trabajar con MAXIMA. La interfaz más básica es la de líneas de comandos, estando disponible cuando MAXIMA se compila con CLISP o con versiones de GCL iguales o superiores a la 2.5.0. Otras interfaces disponibles son las correspondientes al editor `Emacs`, y al entorno `Symaxx`. También es especialmente recomendable la interfaz `Xmaxima`, basada en Tcl/Tk. Funciona igual en diversas plataformas y normalmente aparece por omisión. Permite dibujar gráficos de manera autónoma sin programas adicionales, e incluye una amplia documentación en HTML.

A lo largo del Seminario de Mecánica Computacional se utilizará como interfaz de usuario TEXMACS. Este entorno de trabajo ofrece importantes ventajas, describiéndose sus características principales a continuación.

3 ¿Qué es TEXMACS?

El programa TeXmacs (http://www.texmacs.org/index.php3) es un editor de textos científicos inspirado en TEX y en Emacs, que se distribuye igualmente con licencia GPL. Permite escribir textos estructurados en forma WYSIWYG[2]. Incorpora las fuentes de TEX, obteniéndose de manera directa expresiones matemáticas con una alta calidad tipográfica. Asimismo puede actuar de manera interactiva como interfaz de diversos sistemas de cálculo numérico y algebraico como MAXIMA, Octave, Axiom, YACAS, GnuPlot, etc. Los documentos generados con TEXMACS se pueden convertir a pdf, HTML, XML, TEX/LATEX, etc.

1. En inglés este tipo de programas se engloban bajo la denominación CAS, que es el acrónimo de *Computer Algebra System*

2. WYSIWYG es el acrónimo de What You See Is What You Get

Con las características descritas, un sistema CAS de libre distribución ejecutándose sobre TEXMACS es un entorno de trabajo especialmente adecuado para plantear y resolver los modelos matemáticos que aparecen en los ejercicios y problemas de la Mecánica Racional. De esta forma pueden abordarse problemas y ejercicios que sería muy largo y tedioso resolver "a mano", y dejarlos documentados con comodidad, de forma clara y precisa, empleando una tipografía de gran calidad.

4 Introducción a las capacidades de TEXMACS y MAXIMA

En este apartado se va a desarrollar una típica sesión de trabajo en TEXMACS. Lo que has podido leer hasta ahora está escrito utilizando TEXMACS como procesador de textos. Para invocar una sesión de MAXIMA puedes pulsar el botón de la barra de herramientas que muestra un monitor y seleccionar `MAXIMA -> por defecto`, o bien a través del menú `Texto -> Sesión -> Maxima -> por defecto`. En cualquier caso, las instrucciones de MAXIMA que se explican en este apartado son validas utilizando cualquier otra interfaz de usuario.

4.1 Entrada/salida

Una vez abierta la sesión en MAXIMA se pueden hacer operaciones como en una calculadora. La entrada se realiza en las líneas cuyo "prompt" es una `C` y números naturales consecutivos `n`, y el resultado se muestra en la líneas con una `D` y el número natural correspondiente `n`. Obsérvese que las líneas de entrada finalizan con "punto y coma" (`;`).

```
GCL (GNU Common Lisp)  2.6.1 CLtL1   May  7 2004 21:58:44
Source License: LGPL(gcl,gmp), GPL(unexec,bfd)
Binary License:  LGPL
Modifications of this banner must retain notice of a compatible license
Dedicated to the memory of W. Schelter

Use (help) to get some basic information on how to use GCL.
Maxima restarted.
```

```
(%i1) 3*2+5;
```

(%o1) 11

```
(%i2) 7/5*log(2);
```

$$\text{(\%o2)}\quad \frac{7\log(2)}{5}$$

```
(%i3) %o2,numer;
```

(%o3) 0.97040605278392

```
(%i4) %o2 + %o3;
```

$$\text{(\%o4)}\quad \frac{7\log(2)}{5} + 0.97040605278392$$

```
(%i5)
```

Para salir temporalmente de la sesión en MAXIMA se señala debajo de la última línea de la sesión con el ratón y se pulsa el botón izquierdo. En el ejemplo anterior, escribiendo en la entrada `%o2`, `%o3` se hace referencia a los resultados obtenidos con las operaciones definidas en `%i2`, `%i3`, respectivamente.

4.2 Constantes, variables y asignaciones.

Existen constantes de valores predefinidos en MAXIMA, cuyo nombre siempre empieza con el carácter %. Las variables pueden tener cualquier nombre que no empiece por un número, y la asignación se hace con los "dos puntos" (:). MAXIMA distingue las mayúsculas de las minúsculas. Para volver a entrar en MAXIMA se vuelve a seleccionar dicha sesión en cualquiera de las formas indicadas anteriormente.

```
(%i5) %pi, numer;
(%o5) 3.141592653589793
(%i6) %e, numer;
(%o6) 2.718281828459045
(%i7) (3+2*%i)*(6-5*%i), expand;
(%o7) 28 − 3 i
(%i8) numero:2;
(%o8) 2
(%i9) NUMERO:3;
(%o9) 3
(%i10) numero+NUMERO;
(%o10) 5
(%i11) doble:2*numero$
(%i12) doble;
(%o12) 4
(%i13)
```

Obsérvese que con **expand** en (%i7) se obtiene el resultado del producto de los dos números complejos. Asimismo, si al final de la línea (%i11) se escribe el carácter dólar ($) en lugar de "punto y coma" (;) se ejecuta la instrucción correspondiente (en este caso una asignación) pero no se muestra el resultado de la misma.

4.3 Ecuaciones

El signo = se utiliza para definir ecuaciones. Con la instrucción **solve** se resuelven ecuaciones o sistemas de ecuaciones en forma cerrada:

```
(%i13) solve(x^2+p*x-2=0,x);
```

$$(\%o13)\quad \left[x=-\frac{\sqrt{p^2+8}+p}{2},x=\frac{\sqrt{p^2+8}-p}{2}\right]$$

```
(%i14) solve(3*x^3-2*x+1=0,x);
```

$$(\%o14)\quad \left[x=-\frac{\sqrt{3}\,\mathrm{i}-3}{6},x=\frac{\sqrt{3}\,\mathrm{i}+3}{6},x=-1\right]$$

Para obtener las soluciones numéricamente se utiliza **realroots** o **allroots**. En cualquier caso a la ecuación se le puede asignar un nombre y utilizarlo posteriormente para hacer referencia a la misma:

```
(%i15) mi_ecuacion:x^5-x^4+3*x^3-1=0;
```

(%o15) $x^5 - x^4 + 3\,x^3 - 1 = 0$

```
(%i16) realroots(mi_ecuacion,1.e-7),numer;
```

(%o16) $[x = 0.70999106764793]$

```
(%i17) allroots(mi_ecuacion);
```

(%o17) $[x = 0.70999105870993, x = 0.54230158378695\,\mathrm{i} - 0.38693451122808, x = -0.5423015837\backslash$
$8695\,\mathrm{i} - 0.38693451122808, x = 1.700185295890338\,\mathrm{i} + 0.53193898187312, x = 0.53193898187312 -$
$1.700185295890338\,\mathrm{i}]$

```
(%i18) eq1:x+y=-z;
```

(%o18) $y + x = -z$

```
(%i19) eq2:2*x-y+z=-1;
```

(%o19) $z - y + 2\,x = -1$

```
(%i20) eq3:3*x+2*z+1=0;
```

(%o20) $2\,z + 3\,x + 1 = 0$

```
(%i21) solve([eq1,eq2,eq3],[x,y,z]);
```

```
Dependent equations eliminated:  (3)
```

(%o21) $\left[\left[x = -\frac{2\,\%\mathrm{r1}+1}{3}, y = -\frac{\%\mathrm{r1}-1}{3}, z = \%\mathrm{r1}\right]\right]$

```
(%i22)
```

4.4 Funciones

Al definir una función se utilizan los símbolos `:=` para realizar la asignación:

```
(%i22) g(x):=3*x^3+5*x^2-x+6;
```

(%o22) $g(x) := 3\,x^3 + 5\,x^2 - x + 6$

```
(%i23) g(2);
```

(%o23) 48

Las funciones pueden ser manipuladas y también pueden dibujarse:

```
(%i24) g(x)**2;
```

(%o24) $\left(3\,x^3 + 5\,x^2 - x + 6\right)^2$

```
(%i26) plot2d(g(x),[x,-3,1]);
```

(%o26)

```
(%i27) f(x,y):=x**2-y**2;
```

(%o27) $f(x, y) := x^2 - y^2$

```
(%i28) plot3d(f(x,y),[x,-2,2],[y,-2,2],[grid,12,12]);
```

(%o28) **false**

También se pueden generar listas a partir de funciones:

```
(%i29) cubo[n]:=n*n*n;
```

(%o29) $\mathrm{cubo}_n := n\,n\,n$

```
(%i30) mi_lista:makelist(cubo[j],j,1,10);
```

(%o30) $[1, 8, 27, 64, 125, 216, 343, 512, 729, 1000]$

El uso de las listas se describe en el siguiente apartado:

4.5 Listas

Las listas constituyen uno de los elementos básicos para almacenar información en MAXIMA. El aspecto de una lista es el de un conjunto de elementos encerrados entre corchetes:

```
(%i31) lista1:[1,2,3,x+y];
```

(%o31) $[1, 2, 3, y + x]$

```
(%i32) lista2:[a,e,i,o,u];
```

(%o32) $[a, e, i, o, u]$

Las listas no se asignan como se asignan las variables, sino que hay que utilizar la instrucción `copylist` tal y como se detalla a continuación:

```
(%i33) lista_mal_asignada:lista2;
```

(%o33) $[a, e, i, o, u]$

```
(%i34) lista_bien_asignada:copylist(lista2);
```

(%o34) $[a, e, i, o, u]$

```
(%i35) lista2[1]:B;
```

(%o35) B

```
(%i36) lista2;
```

(%o36) $[B, e, i, o, u]$

```
(%i37) lista_mal_asignada;
```

(%o37) $[B, e, i, o, u]$

```
(%i38) lista_bien_asignada;
```

(%o38) $[a, e, i, o, u]$

Existen diversas funciones para trabajar con listas, describiendo aquí las que consideramos más relevantes para el desarrollo de las prácticas de mecánica computacional;

- La función `append` concatena dos listas:

```
(%i39) append(lista1,lista2);
```

(%o39) $[1, 2, 3, y + x, B, e, i, o, u]$

- La función `cons` añade un elemento al principio de una lista:

```
(%i40) cons(0,lista1);
```

(%o40) $[0, 1, 2, 3, y + x]$

- Análogamente la función `endcons` añade el elemento al final de la lista:

```
(%i41) endcons(0,lista1);
```

(%o41) $[1, 2, 3, y + x, 0]$

- La función `delete` borra un elemento, definido explícitamente, de una lista

```
(%i44) delete(y+x,lista1);
```

(%o44) $[1,2,3]$

```
(%i45)
```

- La función `last` devuelve el último elemento de una lista:

```
(%i45) lista3:[lista1,lista2];
```

(%o45) $[[1,2,3,y+x],[B,e,i,o,u]]$

```
(%i46) last(lista3);
```

(%o46) $[B,e,i,o,u]$

```
(%i47)
```

- La función `length` devuelve el número de elementos de una lista:

```
(%i47) length(lista1);
```

(%o47) 4

```
(%i48) length(lista3);
```

(%o48) 2

```
(%i49)
```

- La función `makelist` genera una lista. Esta instrucción se puede emplear con dos sintaxis: makelist(expresion,variable,valor1,valor2) o makelist(expresion,variable,lista). En el primer caso en la expresion se sustituyen los valores que toma la variable desde valor1 a valor2. En el segundo caso en la expresion la variable toma todos los valores definidos en la lista:

```
(%i49) makelist(concat(x,i),i,1,6);
```

(%o49) [x1, x2, x3, x4, x5, x6]

```
(%i50) makelist(concat(x,i),i,[2,4,6]);
```

(%o50) [x2, x4, x6]

```
(%i51) otroejemplo:solve(x^3=x);
```

(%o51) $[x=-1,x=1,x=0]$

```
(%i52) makelist(rhs(res),res,otroejemplo);
```

(%o52) $[-1,1,0]$

```
(%i53)
```

(%o58) 0

```
(%i59)
```

- La función `member` comprueba si un elemento pertenece a una lista:

```
(%i53) member(e,lista2);
```

(%o53) **true**

```
(%i54) member(b,lista2);
```

(%o54) **false**

```
(%i55)
```

- La función `rest` elimina los `n` primeros/últimos elementos de una lista:

```
(%i55) rest(lista2,3);
```

(%o55) $[o, u]$

```
(%i56) rest(lista2,-3);
```

(%o56) $[B, e]$

```
(%i57)
```

- La función `reverse` invierte el orden de los elementos de una lista:

```
(%i57) reverse(lista2);
```

(%o57) $[u, o, i, e, B]$

```
(%i58)
```

4.6 Manipulación de expresiones

Con MAXIMA es posible manipular expresiones algebraicas. Por ejemplo, se puede desarrollar las potencias de un polinomio:

```
(%i58) polinomio1:(y**3+x**2+x*y+1)**2;
```

(%o58) $\left(y^3+x\,y+x^2+1\right)^2$

```
(%i59) poli2:expand(polinomio1);
```

(%o59) $y^6+2\,x\,y^4+2\,x^2\,y^3+2\,y^3+x^2\,y^2+2\,x^3\,y+2\,x\,y+x^4+2\,x^2+1$

```
(%i60)
```

Sustituimos y por $\frac{1}{z-1}$:

```
(%i60) poli3:poli2,y=1/(z-1);
```

(%o60) $\frac{2\,x^3}{z-1}+\frac{2\,x}{z-1}+\frac{x^2}{\left(z-1\right)^2}+\frac{2\,x^2}{\left(z-1\right)^3}+\frac{2}{\left(z-1\right)^3}+\frac{2\,x}{\left(z-1\right)^4}+\frac{1}{\left(z-1\right)^6}+x^4+2\,x^2+1$

```
(%i61)
```

La expresión anterior puede reducirse a común denominador (no se muestra la salida por ser muy larga):

```
(%i61) poli4:ratsimp(poli3);
```

(%o61)
$$\frac{\left(x^4+2\,x^2+1\right)z^6+\left(-6\,x^4+2\,x^3-12\,x^2+2\,x-6\right)z^5+\left(15\,x^4-10\,x^3+31\,x^2-10\,x+15\right)z^4+\left(-20\,x^4+20\,x^3-42\,x\right.}{z^6-6\,z^5+15\,z^4-20\,z^3+1}$$

```
(%i62)
```

y para obtener una expresión más simple se utiliza la instrucción `factor`:

```
(%i62) factor(poli4);
```

(%o62) $\frac{\left(x^2\,z^3+z^3-3\,x^2\,z^2+x\,z^2-3\,z^2+3\,x^2\,z-2\,x\,z+3\,z-x^2+x\right)^2}{\left(z-1\right)^6}$

```
(%i63)
```

4.7 Expresiones trigonométricas

Con MAXIMA también se pueden manipular las expresiones trigonométricas circulares e hiperbólicas. La instrucción `trigexpand` desarrolla las funciones trigonométricas cuando su argumento es una suma de ángulos o el producto de un ángulo por una constante, aplicando las formulas correspondientes. A continuación demostraremos el posible interés de aplicar `trigexpand` recursivamente.

```
(%i63) my_trig:x+sin(3*x)/sin(x);
```

$$(\%o63)\quad \frac{\sin\left(3\,x\right)}{\sin\left(x\right)}+x$$

```
(%i64) trigexpand(my_trig);
```

$$(\%o64)\quad \frac{3\cos\left(x\right)^2\sin\left(x\right)-\sin\left(x\right)^3}{\sin\left(x\right)}+x$$

```
(%i65) trigexpand(sin(3*x+2*y));
```

$$(\%o65)\quad \cos\left(3\,x\right)\sin\left(2\,y\right)+\sin\left(3\,x\right)\cos\left(2\,y\right)$$

```
(%i66) auxtrig:trigexpand(trigexpand(sin(3*x+2*y)));
```

$$(\%o66)\quad \left(3\cos\left(x\right)^2\sin\left(x\right)-\sin\left(x\right)^3\right)\left(\cos\left(y\right)^2-\sin\left(y\right)^2\right)+2\left(\cos\left(x\right)^3-3\cos\left(x\right)\sin\left(x\right)^2\right)\cos\left(y\right)\sin\left(y\right)$$

```
(%i67)
```

Por el contrario, la instrucción `trigreduce` trata de convertir los productos y potencias de funciones trigonométricas de un ángulo en expresiones con el ángulo múltiple:

```
(%i67) trigreduce(auxtrig);
```

$$(\%o67)\quad \cos\left(3\,x\right)\sin\left(2\,y\right)+\sin\left(3\,x\right)\cos\left(2\,y\right)$$

```
(%i68)
```

La función `trigsimp` aplica las igualdades $\text{sen}^2x+\cos^2x=1$ o bien $\cosh^2x-\text{senh}^2x=1$ para simplificar una expresión:

```
(%i68) auxtrig2:tan(x)*sec(x)**2+(1-sin(x)**2)*cos(x)/cos(x)**2;
```

$$(\%o68)\quad \sec\left(x\right)^2\tan\left(x\right)+\frac{1-\sin\left(x\right)^2}{\cos\left(x\right)}$$

```
(%i69) trigsimp(auxtrig2);
```

$$(\%o69)\quad \frac{\sin\left(x\right)+\cos\left(x\right)^4}{\cos\left(x\right)^3}$$

```
(%i70)
```

Otra instrucción para simplificar expresiones trigonométricas es `trigrat`:

```
(%i70) trigrat(sin(3*a)/sin(a+%pi/3));
```

$$(\%o70)\quad \sqrt{3}\sin\left(2\,a\right)+\cos\left(2\,a\right)-1$$

```
(%i71)
```

4.8 Álgebra lineal

MAXIMA también permite trabajar con matrices y vectores. Los vectores se definen como listas, describiéndose el uso de estas en el siguiente apartado.

```
(%i71) A:matrix([1,2],[3,4]);
```

$$(\%o71)\quad \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
(%i72) B:matrix([1,1],[1,1]);
```

$$(\%o72)\quad \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$$

```
(%i73) A+B;
```

$$(\%o73)\quad \begin{pmatrix} 2 & 3 \\ 4 & 5 \end{pmatrix}$$

```
(%i74) A.B;
```

$$(\%o74)\quad \begin{pmatrix} 3 & 3 \\ 7 & 7 \end{pmatrix}$$

```
(%i75) A^^(-1);
```

$$(\%o75)\quad \begin{pmatrix} -2 & 1 \\ \frac{3}{2} & -\frac{1}{2} \end{pmatrix}$$

```
(%i76) eigenvectors(A);
```

$$(\%o76)\quad \left[\left[\left[-\frac{\sqrt{33}-5}{2},\frac{\sqrt{33}+5}{2}\right],[1,1]\right],\left[1,-\frac{\sqrt{33}-3}{4}\right],\left[1,\frac{\sqrt{33}+3}{4}\right]\right]$$

```
(%i77) v:[1,2];
```

$$(\%o77)\quad [1,2]$$

```
(%i78) w:[3,5];
```

$$(\%o78)\quad [3,5]$$

```
(%i79) v.A;
```

$$(\%o79)\quad \begin{pmatrix} 7 & 10 \end{pmatrix}$$

```
(%i80) (v.A).w;
```

$$(\%o80)\quad 71$$

```
(%i81)
```

En la asignación de matrices hay que utilizar la instrucción `copymatrix` pues si se asignan como variables con el signo : aparece el mismo problema descrito en la asignación de listas. Para la asignación de vectores se utiliza la instrucción `copylist`:

```
(%i81) A_mal_asignada:A;
```

$$(\%o81)\quad \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
(%i82) A_bien_asignada:copymatrix(A);
```

$$(\%o82)\quad \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

```
(%i83) A[2,2]:1000;
```

(%o83) 1000

```
(%i84) A;
```

(%o84) $\begin{pmatrix} 1 & 2 \\ 3 & 1000 \end{pmatrix}$

```
(%i85) A_mal_asignada;
```

(%o85) $\begin{pmatrix} 1 & 2 \\ 3 & 1000 \end{pmatrix}$

```
(%i86) A_bien_asignada;
```

(%o86) $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$

```
(%i87)
```

Existe un paquete con funciones avanzadas de cálculo con vectores, que se carga con la instrucción `load(vect)`. *En cualquier caso, no es recomendable trabajar con paquetes externos sin haber leído antes la documentación.*

4.9 Análisis matemático

MAXIMA también dispone de numerosas funciones relacionadas con el análisis matemático, pasando ahora revista a algunas de las más básicas.

Límites

Se pueden calcular límites:

```
(%i87) limit( sin(3*x)/x, x,0);
```

(%o87) 3

```
(%i88) limit( (2*x+1)/(3*x+2), x,inf );
```

(%o88) $\frac{2}{3}$

```
(%i89) limit( x**3/(5*x**2+2), x,inf );
```

(%o89) ∞

```
(%i90)
```

Derivadas

Se puede derivar funciones de una o varias variables:

```
(%i90) f(x):=exp(x)*cos(x)/(x**2+1);
```

(%o90) $f(x) := \frac{\exp(x)\cos(x)}{x^2+1}$

```
(%i91) diff(f(x),x);
```

(%o91) $-\frac{e^x \sin(x)}{x^2+1} + \frac{e^x \cos(x)}{x^2+1} - \frac{2\,x\,e^x \cos(x)}{(x^2+1)^2}$

```
(%i92) h(x,y,z):=sin(x)*(y+z)**2;
```

(%o92) $h(x,y,z) := \sin(x)\,(y+z)^2$

```
(%i93) diff(h(x,y,z),y);
```

(%o93) $2\sin(x)(z+y)$

```
(%i94)
```

y definir dependencias funcionales para deducir las correspondientes expresiones de la "regla de la cadena". Esto se consigue con la función `depends`:

```
(%i94) depends([V],[r,theta,phi],[r,theta,phi],[x,y,z]);
```

(%o94) $[V(r,\vartheta,\varphi), r(x,y,z), \vartheta(x,y,z), \varphi(x,y,z)]$

```
(%i95) diff(V,x);
```

(%o95) $\frac{d}{dx}\vartheta\left(\frac{d}{d\vartheta}V\right)+\frac{d}{dx}r\left(\frac{d}{dr}V\right)+\frac{d}{dx}\varphi\left(\frac{d}{d\varphi}V\right)$

Este resultado empieza a complicarse si queremos obtener a mano algo tan sencillo como, por ejemplo, una de las derivadas segundas de V:

$$\frac{d^2V}{dx\,dy}$$

Calcularlo con MAXIMA es inmediato:

```
(%i96) ratsimp(diff(V,x,1,y,1));
```

(%o96) $\frac{d}{dx}\vartheta\left(\frac{d}{dy}\vartheta\right)\left(\frac{d^2}{d\vartheta^2}V\right)+\frac{d^2}{dx\,dy}\vartheta\left(\frac{d}{d\vartheta}V\right)+\frac{d}{dx}r\left(\frac{d}{dy}r\right)\left(\frac{d^2}{dr^2}V\right)+$
$\left(\frac{d}{dx}r\left(\frac{d}{dy}\vartheta\right)+\frac{d}{dy}r\left(\frac{d}{dx}\vartheta\right)\right)\left(\frac{d^2}{dr\,d\vartheta}V\right)+\frac{d^2}{dx\,dy}r\left(\frac{d}{dr}V\right)+\frac{d}{dx}\varphi\left(\frac{d}{dy}\varphi\right)\left(\frac{d^2}{d\varphi^2}V\right)+$
$\left(\frac{d}{dx}\varphi\left(\frac{d}{dy}\vartheta\right)+\frac{d}{dy}\varphi\left(\frac{d}{dx}\vartheta\right)\right)\left(\frac{d^2}{d\varphi\,d\vartheta}V\right)+\left(\frac{d}{dx}\varphi\left(\frac{d}{dy}r\right)+$
$\frac{d}{dy}\varphi\left(\frac{d}{dx}r\right)\right)\left(\frac{d^2}{d\varphi\,dr}V\right)+\frac{d^2}{dx\,dy}\varphi\left(\frac{d}{d\varphi}V\right)$

```
(%i97)
```

Integrales

Con MAXIMA también se pueden hacer integrales indefinidas:

```
(%i97) assume(n>0);
```

(%o97) $[n>0]$

```
(%i98) integrate(a*x**n,x);
```

(%o98) $\frac{a\,x^{n+1}}{n+1}$

```
(%i99) forget(n>0);
```

(%o99) $[n>0]$

```
(%i100)
```

e integrales definidas:

```
(%i100) integrate( x/(1+x^2),x,0,1);
```

(%o100) $\dfrac{\log(2)}{2}$

(%i101)

Ecuaciones diferenciales ordinarias

Finalmente, también se pueden integrar ecuaciones diferenciales ordinarias de primer y segundo orden con la instrucción `ode2`. Para integrar la ecuación diferencial:

$$x^2\frac{dy(x)}{dx} + 3xy(x) = \frac{\operatorname{sen} x}{x} \quad (1)$$

$$y(1) = 1 \quad (2)$$

se hace:

```
(%i101) depends(y,x);
```

(%o101) $[y(x)]$

```
(%i102) soln1:ode2(x^2*diff(y,x) + 3*x*y = sin(x)/x, y, x);
```

(%o102) $y = \dfrac{\%c - \cos(x)}{x^3}$

```
(%i103) ic1(soln1, x=1, y=1);
```

(%o103) $y = -\dfrac{\cos(x) - \cos(1) - 1}{x^3}$

(%i104)

Con las ecuaciones de segundo orden se procede de idéntica manera pero las condiciones iniciales se definen con la instrucción `ic2`. A continuación se integrará la siguiente ecuación diferencial:

$$\frac{d^2y(x)}{dx^2} + y(x) = 4x \quad (3)$$

$$y(0) = 1 \quad (4)$$

$$y'(0) = 3 \quad (5)$$

NOTA: Obsérvese que ya se ha definido anteriormente la dependencia $y = y(x)$

```
(%i107) eqn2: diff(y,x,2) + y = 4*x;
```

(%o107) $\dfrac{d^2}{dx^2}y + y = 4x$

```
(%i108) soln2: ode2(eqn2, y, x);
```

(%o108) $y = \%k1\sin(x) + \%k2\cos(x) + 4x$

```
(%i109) ic2(soln2, x=0, y=1, diff(y,x)=3);
```

(%o109) $y = -\sin(x) + \cos(x) + 4x$

(%i110)

NOTA: En lugar de definir las condiciones iniciales con la función `ic2` se pueden imponer dos condiciones de contorno con la instrucción `bc2`