suma+ 61
Junio 2009, pp. 55-61

La potencia de las TIC para el cálculo simbólico

Desde esta sección Mate*más*TIC intentamos en cada número dar a conocer alguna herramienta informática relacionada con las matemáticas a la que poder sacarle partido en el aula. Dada la apuesta que desde distintas comunidades autónomas se ha hecho o se está haciendo por el software libre, las herramientas que damos a conocer son para este tipo de sistemas, existiendo en algunos casos la réplica de la misma aplicación para sistemas propietarios.

En nuestro propósito de, por un lado, dar a conocer estas aplicaciones, y por otro, intentar que las mismas incidan de forma directa en el proceso de enseñanza-aprendizaje que se desarrolla en el aula, debemos tener en cuenta que la utilización de estas herramientas debe ir acompañada de un cambio de metodología en la que los alumnos sean partícipes del propio proyecto.

Para que pueda servirnos de reflexión y como curiosidad, os aconsejamos que visualicéis el vídeo que se encuentra en la siguiente dirección del portal *Youtube:*

http://www.youtube.com/watch?v=IJY-NIhdw_4

En la imagen 1 observamos un fotograma del mencionado vídeo.

Imagen 1: Las TIC sin metodología

Mariano Real Pérez
CEP de Sevilla
matemastic@revistasuma.es

De nada nos servirá todo el equipamiento de que podamos disponer en nuestras aulas y el conocimiento de múltiples aplicaciones que podamos utilizar en nuestra materia si no incidimos en un cambio de metodología. Sirvan estas líneas para invitar a la reflexión sobre las aportaciones que un uso adecuado de las TIC pueden reportar a la labor que desarrollamos diariamente. Más aún pensando en la sociedad tecnológica en la que nosotros, los docentes, hemos entrado pero en la que nuestros alumnos han nacido.

Una vez hecha esta pequeña invitación a la reflexión y entrando de lleno en el trabajo que nos ocupa, hemos pensado para esta vez, analizar un software que, aunque no es tan atractivo, visualmente hablando, sí lo es por el potencial que el mismo nos ofrece. Un software que podríamos denominar *generalista* y que podemos localizar tanto para sistemas libres como propietarios.

Maxima es una evolución de Macsyma, el que fuera innovador sistema de álgebra computacional desarrollado a finales de 1960 en el instituto tecnológico de Massachusetts (MIT)

De entre las aplicaciones *generalistas* o de cálculo simbólico podemos localizar en el mercado algunas como *Derive* o *Mapple* a las que nada tienen que envidiar la que aquí presentamos en este número.

Dado lo extensa que es y la gran cantidad de herramientas que tiene, sería imposible hacer un recorrido completo por la misma, por lo que hemos decidido hacer este recorrido en tres pasos diferentes, siendo el de este número de SUMA el primero de ellos.

Cálculo simbólico I: Comenzando con Maxima

A medida que los alumnos van avanzando en el sistema educativo se le exige un mayor razonamiento y nivel de abstracción. En matemáticas esto se traduce en la resolución de ejercicios matemáticos en los que el nivel de simbolismo es creciente o en la generalización de problemas ya resueltos con el consiguiente aumento de las variables que intervienen en el original. En otras ocasiones, la resolución de determinados problemas conlleva la realización de múltiples y repetitivas operaciones con gran cantidad de variables que, lejos de profundizar y agilizar la resolución del mismo, propicia el alejamiento del objetivo final y del razonamiento inicial.

Con el fin de facilitar esta tarea, comenzamos en este número el recorrido por una aplicación de software libre con la que realizar cálculo simbólico. Una aplicación que se podría utilizar a partir de cuarto de ESO, incluyendo este curso como un nivel ideal para la introducción de la misma, siendo los cursos del bachillerato y universitarios en los que más partido se le puede sacar. El recorrido por la misma lo vamos a realizar en tres entregas siendo este número de *Suma* la primera de ellas.

El software que vamos a tratar se llama *Maxima*.

Maxima es una evolución de *Macsyma* (MAC's SYmbolic MAnipulation System, donde MAC, Machine Aided Cognition, era el nombre del Laboratory for Computer Science del MIT durante la fase inicial del proyecto *Macsyma*), el que fuera innovador sistema de álgebra computacional desarrollado a finales de 1960 en el instituto tecnológico de Massachusetts (MIT). *Mapple* y *Matemática* son software que también está basado en el que fuera revolucionario sistema *Macsyma*.

Maxima es un Sistema de Computación Algebraica (CAS), es decir, un programa informático que nos permite realizar manipulaciones algebraicas como, por ejemplo, operar con números en forma exacta, manejar expresiones con variables, factorizar enteros o polinomios, resolver ecuaciones de forma exacta, calcular derivadas y primitivas, operar con matrices y calcular determinantes que contienen parámetros, etc. También permite el empleo de enteros de longitud arbitraria y realizar cálculos aproximados con un número arbitrario de dígitos.

William Schelter fue quien mantuvo *Maxima* desde 1982 hasta su muerte en 2001. En 1998 obtuvo permiso para liberar el código fuente bajo la licencia pública general (GPL) de GNU, cosa que influyó definitivamente sobre este software para conseguir la espectacular evolución que presenta en la actualidad. Esto mismo ha influido para que este software se pueda compilar para cualquier sistema, no sólo sistemas libres como Linux, sino además para los sistemas propietarios. En un principio, los paquetes del sistema se encuentran en el repositorio de cada una de las distribuciones Linux. Pero estos mismos paquetes se pueden localizar en la web del proyecto *Maxima* (http://maxima.sourceforge.net) para todos los demás sistemas, incluido el propio Linux. Es fácil también encontrar en Internet otros sitios que ofrezcan *Maxima* para los sistemas propietarios. Entre esos sitios podemos destacar la web de CdLibre

www.cdlibre.org/consultar/catalogo/Matematicas_Calculo-simbolico.html

Como venimos haciendo en los distintos números de esta sección, el recorrido por la aplicación lo vamos a realizar sobre Linux.

Podemos decir que el sistema de álgebra computacional *Maxima* es un motor de cálculo simbólico escrito en lenguaje Lisp publicado bajo licencia GNU GPL como hemos mencionado anteriormente. *Maxima* cuenta con un amplio conjunto de funciones para hacer manipulación simbólica de polinomios, matrices, funciones racionales, integración, derivación, manejo de gráficos en 2D y 3D, manejo de números de coma flotante muy grandes, expansión en series de potencias y de Fourier, entre otras funcionalidades.

Maxima funciona en modo consola, sin embargo vamos a ver que en su evolución también ha ido mejorando su aspecto gráfico conservando su potencia.

Una vez que hemos instalado en nuestro equipo *Maxima* mediante *apt-get* o bien utilizando la aplicación gráfica *Synaptic* (Linux) debemos ejecutarlo mediante consola, tecleando únicamente *maxima*.

Si realizamos lo anterior, nos aparecerá la información sobre la versión de *Maxima* que estamos utilizando. Al final de esta información encontramos el símbolo:

(%i1)

Observamos el siguiente texto que se nos muestra en la consola una vez que ejecutamos *Maxima*:

```
Maxima 5.13.0 http://maxima.sourceforge.net
Using Lisp GNU Common Lisp (GCL) GCL 2.6.8 (aka GCL)
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
This is a development version of Maxima. The function
bug_report()
provides bug reporting information.
(%i1)
```

En el caso anterior nos dice que la versión que estamos utilizando es la 5.13.0, aunque en los enlaces que propusimos anteriormente encontraréis versiones posteriores.

De forma general en la aplicación, debemos conocer que (%*ik*) pide que introduzcamos (*i*=input) la expresión numérica *k* y de forma análoga (%*oj*) nos indica que la expresión resultante (*o*=output) numérica *j* es la que se muestra. Para familiarizarnos con este funcionamiento vamos a realizar operaciones sencillas con números y expresiones. En la Imagen 2 podemos observar varias operaciones que hemos realizado con *Maxima*, tanto con números como con expresiones algebraicas. La forma que tenemos de indicarle a la aplicación que hemos acabado de introducir una determinada operación o expresión es escribir punto y coma (;) y pulsar enter.

Es importante tener presente que debemos indicar todas las operaciones a la hora de escribir las distintas expresiones

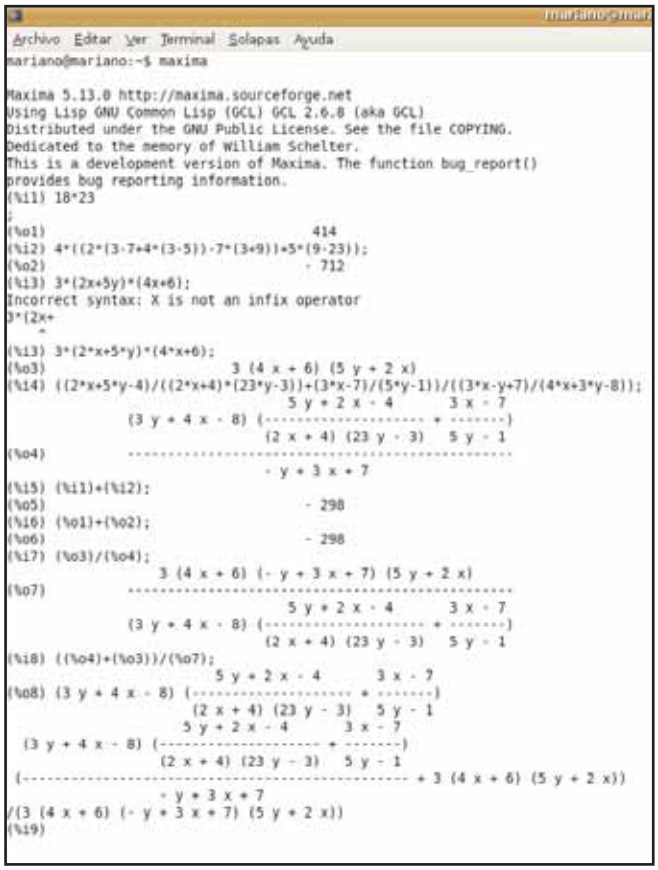

Imagen 2: Operaciones realizadas con *Maxima*

algebraicas, esto es, la falta de un símbolo de operación no es interpretada por la aplicación como un producto, indicándonos que hemos cometido un error. En la expresión tercera que hemos introducido en la Imagen 2 observamos que entre el número 2 y la variable *x*, y entre el número 4 y la variable *x* no hemos introducido ninguna operación. Como observamos en la Imagen 2, para introducir el exponente debemos hacerlo utilizando el símbolo ^ antes de colocar el número o expresión que servirá de exponente.

Si observamos la Imagen 2 podemos distinguir que hemos seguido los siguientes pasos:

1. Hemos introducido una expresión numérica en (%*i1*) detrás de la que hemos tecleado punto y coma (;) y nos ha aparecido resuelta en (%*o1*).

2. Hemos introducido otra expresión numérica que nos ha aparecido resuelta en (%*o2*).

3. Hemos introducido una expresión algebraica en (%*i3*) pero con errores ya que no hemos indicado la operación existente entre el número 2 y la variable *x*. Tras colocar el punto y coma correspondiente y pulsar enter, el sistema

nos informa del primero de los errores que detecta que hemos cometido. Al tener errores no nos proporciona la correspondiente salida (%*o3*) y vuelve a solicitarnos que introduzcamos nuevamente (%*i3*).

4. Hemos vuelto a introducir la expresión algebraica anterior, pero esta vez sin errores, lo que nos a proporcionado la correspondiente salida (%*o3*): 3 (4 x + 6) (5 y + 2 x)

5. Hemos introducido la expresión algebraica (%*i4*), proporcionándonos la correspondiente salida (%*o4*).

   Las correspondientes notaciones que van apareciendo a lo largo de la aplicación van acumulando los valores o expresiones que nos indican. Este hecho lo podemos observar en la entrada (%*i5*), en el que le indicamos que multiplique (%*i1*) y (%*i2*), proporcionándonos la salida (%*o5*) que es el resultado de realizar la anterior operación.

6.- En las siguiente entradas que aparecen en la imagen hemos realizado operaciones con las distintas entradas (%*ik*) o salidas (%*oj*) que habíamos utilizado anteriormente.

La aplicación funciona a base de órdenes y operaciones, marcando la finalización de cada una de ellas un punto y coma (;). Una vez que está funcionando la aplicación, si deseamos salir de ella debemos teclear la orden quit() seguida del punto y coma correspondiente.

La potencialidad de *Maxima* radica en la gran cantidad de comandos que tiene implementados y que pueden servirnos para resolver operaciones matemáticas que nos supondrían invertir gran cantidad de esfuerzo en realizarlas. Además, *Maxima* posee gran cantidad de información de cómo utilizar cada uno de los comandos que tiene implementados. Para obtener información sobre uno de los comandos basta con teclear *describe(comando)* seguido de punto y coma. Podemos probar, por ejemplo con:

describe(*factor*);

Si lo hacemos obtenemos la información que puedes observar en la Imagen 3.

En esta imagen observamos toda la información que la aplicación nos muestra sobre el comando factor. Esta información viene acompañada además de una gran cantidad de ejemplos.

Ahora vamos a realizar algunas de las operaciones con *Maxima* para observar el comportamiento que tiene la aplicación y el potencial que nos proporciona. Dado que sería imposible, por el espacio de que disponemos, realizar un recorrido por todos los comandos que tiene implementados la aplicación, utilizaremos algunos de estos comandos observando el comportamiento que tienen y la potencialidad que nos ofrecen. En los ejemplos anteriores hemos observado la forma de escribir distintas operaciones:

Imagen 3: Información sobre el comando *factor*

+ Para la suma. Ejemplo: 23 + 4
– Para la resta. Ejemplo: x – 5
* Para el producto. Ejemplo: 3*x + 7
/ Para la división. Ejemplo: (2*x + 9)/(x – 3)
^ Para la potencia. Ejemplo: 3*x^2 – 7*x + 8
  La potencia también se puede escribir como **.
  Ejemplo: 2*x**2 – 7*x + 8
sqrt Para la raíz cuadrada. Ejemplo: sqrt(x^3 + 7)

Además, *Maxima* ya tiene implementada algunas constantes que nos pueden resultar muy útiles como:

%PI es el número π.
%e es el número *e*.
%i indica el complejo puro.

En la Imagen 4 hemos utilizado las anteriores operaciones entre expresiones algebraicas y constantes, observando en la salida correspondiente la operación realizada o la expresión algebraica resultante.

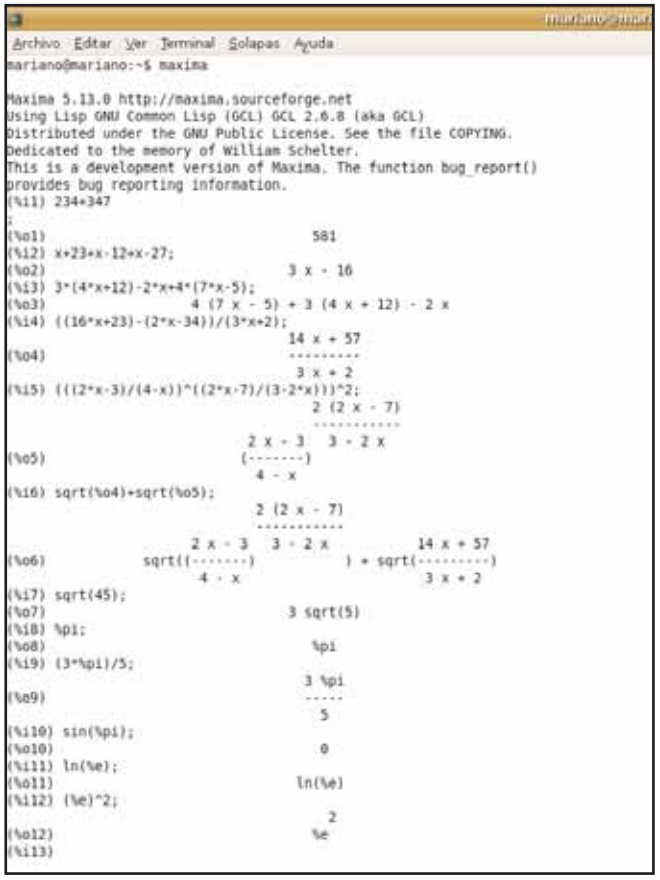

Imagen 4: Operaciones entre expresiones algebraicas y constantes

Como venimos indicando, *Maxima* es una herramienta muy potente y muy versátil para utilizarla en el aula de matemáticas, por lo que podemos intuir que, dado el funcionamiento que hemos contemplado a lo largo del texto anterior, la aplicación dispone de una gran cantidad de comandos que pueden dar respuesta a los distintos cálculos simbólicos que nos podemos plantear, disponiendo además de una serie de parámetros con los que poder configurar o que sirven de complemento a muchos de esos comandos. Así, *Maxima* sirve como herramienta para resolver ejercicios relacionados con operadores, expresiones, simplificación, gráficos, números y funciones de punto flotante, contextos, polinomios, constantes, logaritmos, trigonometría, funciones especiales, polinomios ortogonales, límites, diferenciación, integración, ecuaciones, ecuaciones diferenciales, cálculo numérico, estadística, tablas, matrices y álgebra lineal, espacios afines, tensores, ctensores, series, teoría de números, simetrías, grupos... *Maxima* posee además un entorno de programación con funciones lógicas que nos permite realizar múltiples acciones a la hora de afrontar la resolución de un problema e incluso realizar operaciones de cálculo simbólico compuestos a través de una única expresión.

Dado que sería imposible recoger ejemplos para todos los comandos que posee *Maxima*, hemos recogido algunos solamente que muestren la guinda de este pastel. Así, en la Imagen 5 podemos observar algunos de esos ejemplos.

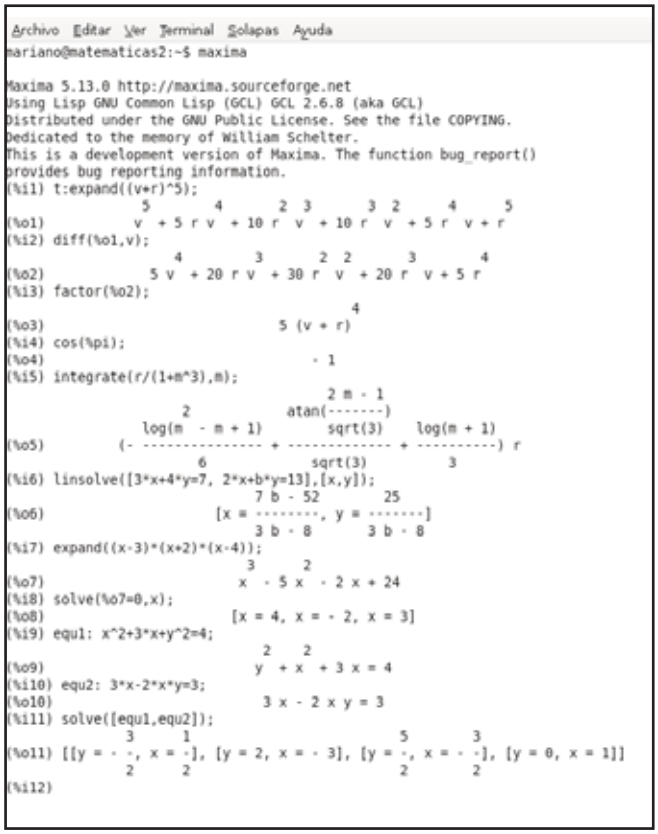

Imagen 5: Algunos comandos con *Maxima*

En la imagen 5 observamos algunos comandos de *Maxima* que hemos utilizado para efectuar cálculos simbólicos. Analicemos las distintas operaciones que se recogen en esta imagen:

a) Para comenzar hemos definido *t* como la potencia quinta de la expresión $(r + v)$, pero desarrollada. Para definirla, en lugar del símbolo igual hemos utilizado los dos puntos que es la sintaxis correspondiente para asignarle a una variable una expresión o un valor constante. Y para indicar que debe efectuar la operación hemos utilizado el comando *expand*, colocando al final el punto y coma (;) para finalizar el comando:

$$t\text{:}expand((r + v)\text{^}5);$$

Como resultado hemos obtenido la salida (%*o1*)

b) Nos planteamos ahora calcular la derivada de la expresión obtenida en el paso anterior, respecto a la variable *v*. Dado que la expresión anterior se llama *t* utilizaremos el siguiente comando:

diff(*t*,*v*);

Con esta acción obtenemos la salida (%*o2*). Debemos tener en cuenta que también podríamos haber escrito el comando de la siguiente forma:

diff(%*o1*,*v*);

ya que la expresión que pretendemos derivar es la obtenida como salida (%*o1*).

c) Posteriormente hemos realizado la factorización de la expresión obtenida en la salida (%*o2*). Para ello utilizamos el siguiente comando:

factor(%*o2*);

Obteniendo la salida (%*o3*).

d) En el siguiente paso calculamos el coseno del número π. Para ello utilizamos

cos(%*pi*);

La salida que hemos obtenido en este caso es (%*o4*) que nos indica que la operación tienen como resultado −1

e) Para calcular la integral indefinida de una función utilizamos el comando *integrate*. En este caso vamos a realizar la integral, respecto a la variable *m* de la función:

$$\frac{r}{1+m^3}$$

Para hacer esto utilizaremos el comando

integrate(*r*/(1 + *m*^3),*m*);

También podríamos haber realizado la operación en dos pasos. Escribiendo primero la función, asignándole un nombre, por ejemplo *h*. Posteriormente habríamos realizado la integral con el comando

integrate(*h*,*m*);

f) Con Maxima también podemos resolver sistemas de ecuaciones lineales. Para hacer esto utilizaremos la orden *linsolve*. Por ejemplo, para resolver el sistema de ecuaciones

$$\begin{cases} 2x+4y=7 \\ 2x+by=13 \end{cases}$$

utilizaremos el comando

linsolve([3*x + 4*y = 7, 2*x + b*y = 13], [x,y]);

Aquí le indicamos que resuelva el sistema anterior en el que las variables son *x* e *y*.

g) Con *Maxima* también podemos resolver ecuaciones no lineales. Para ello, vamos a calcular primero un polinomio con raíces enteras. En este caso volvemos a utilizar el comando *expand* que ya utilizamos en el apartado a). Vamos a calcular un polinomio cuyas raíces sean 3, −2 y 4. Para esto utilizaremos el siguiente comando

expand((*x* − 3)*(*x* + 2)*(*x* − 4));

Este comando ha dado como salida la (%*o7*)

h) Ahora nos planteamos calcular las raíces del polinomio anterior. Para ello vamos a resolver la ecuación, con variable *x*, que se obtiene al igualar el polinomio anterior a cero. El comando que utilizaremos será:

solve(%*o7*=0,*x*);

Observamos que nos proporciona las raíces del polinomio en la salida (%*o8*).

Aunque hemos calculado las raíces de un polinomio del que previamente conocíamos las raíces y sabíamos que eran números enteros, con este comando podemos calcular las raíces de cualquier polinomio. El comando nos proporciona todas las raíces, incluso las complejas.

i) Por último, en la imagen 5 hemos realizado los pasos necesarios para resolver un sistema de ecuaciones no lineales. El sistema de ecuaciones que nos hemos propuesto resolver es el siguiente:

$$\begin{cases} x^2+3x+y^2=4 \\ 2x+2xy=3 \end{cases}$$

Para resolverlo hemos definido cada una de las ecuaciones que compone el sistema no lineal. A la primera ecuación la hemos denominado *equ1* utilizando el siguiente comando:

equ1: *x*^2 + 3*x + *y*^2 = 4;

Esto nos ha proporcionado la salida (%*o9*). Posteriormente hemos denominado *equ2* a la segunda ecuación utilizando el comando:

equ2: 3*x − 2*x*y = 3;

Este comando nos ha proporcionado la salida (%*o10*). Ahora ya estamos en disposición de resolver el sistema de ecuaciones. Volvemos a utilizar el mismo comando que en expresiones anteriores:

solve([*equ1*,*equ2*]);

Podríamos haber utilizado un comando más directo que recogiera los tres pasos anteriores en un único paso. El comando que deberíamos haber escrito en este caso sería:

solve([*x*^2 + 3**x* + *y*^2 = 4,3**x* − 2**x***y* = 3]);

Otra forma de resolverlo hubiese sido con el comando

solve([*%o9*,*%o10*]);

Hagamos de la forma que lo hagamos, el comando nos proporciona como salida las cuatro soluciones posibles que tiene el sistema de ecuaciones.

Hasta aquí hemos recogido una pequeña muestra de los comandos con los que cuenta está aplicación y la forma de utilizarlos. Pero como venimos insistiendo a lo largo de todo el texto, *Maxima* es una completísima y potente aplicación para el cálculo simbólico, por lo que cuenta con numerosos comandos que podemos utilizar. En la siguiente dirección podemos obtener un manual de la misma:

http://maxima.sourceforg.net/docs/manual/es/maxima.pdf

El anterior manual ocupa 4.95Mb y tiene una extensión de 878 páginas, lo que nos puede dar una idea del potencial que se esconde detrás de *Maxima*, un software que nos ofrece además, la posibilidad de programar nuestros propios comandos.

El potencial y robustez de *Maxima* ha ido creciendo con la evolución de sus versiones, pero este potencial se veía mermado por lo complicado de su utilización ya que se necesitaban conocer numerosos comandos para poder sacarle partido a la misma. Tal como hemos podido comprobar, *Maxima* no tiene nada que ver en lo que a gráficos se refiere, con aplicaciones que se utilizan actualmente como las que hemos analizado en otras ocasiones en esta sección.

Para dar solución a este problema, paralelamente al desarrollo de *Maxima* se fue trabajando sobre un entorno gráfico que facilitara el uso de de este software. El primero de estos pasos se denomina *Xmaxima*, que es una implementación de *Maxima* basada en TCL/TK que puede ejecutarse en entornos Unix, Linux y sistemas propietarios y que está bajo licencia GNU-GPL.

Posteriormente, el trabajo sobre el entorno gráfico de *Maxima* evolucionó hasta el actual *Wxmaxima*.

Wxmaxima es el entorno gráfico basado en wxwidgets. *Wxmaxima* se debe instalar después de tener instalado en nuestro equipo *Maxima*, aunque en versiones más recientes, como la 5.17 de *Maxima*, las dos aplicaciones se instalan a la vez.

El funcionamiento de *Wxmaxima* y el potencial de este nuevo entorno será objeto de esta sección en el siguiente número de SUMA.

Con este recorrido hemos podido entrever la gran potencialidad que nos ofrece esta aplicación. Un software cuya introducción es recomendable comenzar a realizarla en 4º de ESO y que es muy útil en bachillerato y en niveles universitarios.

MatemásTIC ■

Sobre este mismo tema ver SUMA 60, pp. 7-20 [N. de la R.]

Ficha educativo - técnica	
Nombre	Maxima
Sistema	Aunque es una aplicación propia de Linux y para cada distribución cuenta con el archivo de instalación en su repositorio, también encontramos las versiones correspondientes para Windows y para Mac.
Descarga	Repositorio de la distribución de Linux correspondiente o http://maxima.sourceforge.net
Licencia	GPL
Contenido	Cálculo simbólico.
Nivel	Multinivelar: 4º ESO, Bachillerato y Universidad.
Metodología	Aplicación para utilizar a partir de 4º de ESO. Los alumnos utilizarán individualmente la aplicación como herramienta de ayuda para la resolución de problemas y tareas matemáticas