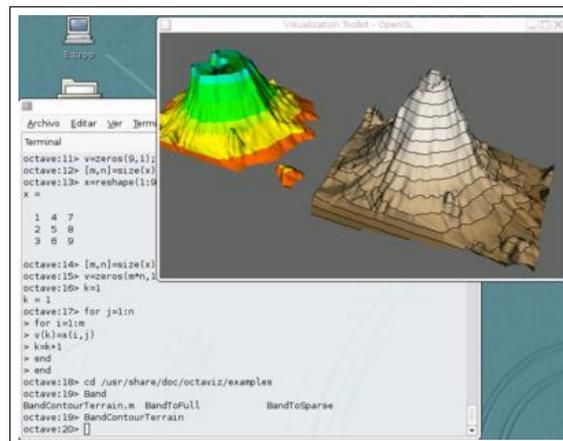


Matemáticas y Software libre para la docencia en la Universidad de Cádiz

J. Rafael Rodríguez Galván*

30 de Junio de 2009



Resumen

En el presente documento se recoge un censo y análisis previo de las distintas herramientas matemáticas con licencia libre existentes, con especial atención al cálculo simbólico y numérico.

*Departamento de Matemáticas de la Universidad de Cádiz (<http://www.uca.es/dept/matematicas>). OS-LUCA (Oficina de Software Libre y Conocimiento Abierto de la Universidad de Cádiz) (<http://www.uca.es/softwarelibre>).

Índice

I	Introducción	3
1.	Introducción	3
1.1.	El software libre	3
1.2.	¿Por qué el software libre en las aulas?	3
2.	Requisitos para las herramientas matemáticas	5
II	Censo y análisis previo de las herramientas disponibles	6
3.	Programas de cálculo simbólico	6
3.1.	Axiom	6
3.2.	YACAS	7
3.3.	GiNaC	8
3.4.	Pari	8
3.5.	Maxima	8
3.5.1.	wxMaxima	9
3.5.2.	xMaxima	10
4.	Programas de cálculo numérico y matricial	10
4.1.	EULER	10
4.2.	Octave	12
4.3.	R	12
4.4.	Scilab	12
4.5.	Tela	13
5.	Interfaces de usuario genéricas	13
5.1.	T _E Xmacs	13
5.2.	WIMS	13
5.3.	Emacs	15
	Referencias	16

©J. Rafael Rodríguez Galván, Universidad de Cádiz. Este documento se distribuye con licencia libre (Creative Commons ShareAlike 2.0 [3]) o superior.

Índice de figuras

1.	Axiom, captura de pantalla	6
2.	YACAS, captura de pantalla	7
3.	wxMaxima, captura de pantalla	9
4.	xMaxima, captura de pantalla	10
5.	EULER, captura de pantalla	11
6.	Octave, captura de pantalla	11
7.	Scilab, captura de pantalla	12
8.	TeXmacs (ejecutando una sesión Axiom), captura de pantalla	14
9.	WIMS, captura de pantalla	14
10.	Emacs (utilizando el modo Octave), captura de pantalla	15

Parte I

Introducción

1. Introducción

1.1. El software libre

Muchas personas están acostumbradas a no ser realmente propietarias del software que utilizan, en el sentido que su licencia, prohíbe utilizarlo libremente (por ejemplo, en más de un ordenador), estudiar su diseño, distribuir copias o modificarlo.

Existen, sin embargo, programas cuya licencia hace que el usuario sea realmente dueño del mismo, pues ésta permite e incluso fomenta libertades exactamente contrapuestas a las privaciones anteriores. Así, se llama *Software Libre* a aquél que garantice las siguientes libertades:

0. La libertad de usar el programa, en cualquier momento y con cualquier propósito.
1. La libertad de estudiarlo y modificar el *código fuente* del programa (la información completa sobre su diseño real).
2. La libertad de distribuir copias a quien pueda necesitarlas.
3. La libertad de mejorar el programa y hacer públicas las mejoras, de modo que toda la sociedad se beneficie de ellas.

A los programas que de alguna manera privan de estas libertades se les suele llamar *Software Privativo*.

1.2. ¿Por qué el software libre en las aulas?

En todos los niveles educativos, en particular en el universitario, el uso de las tecnologías de la información y de la comunicación aumenta día a día y en todos los ámbitos. En particular, el uso del ordenador en las aulas de asignaturas de matemáticas está abriendo las puertas en nuestros días a nuevas oportunidades y a terrenos inexplorados.

Durante el curso 2004/2005, cuatro asignaturas del departamento de Matemáticas en distintos centros de la Universidad de Cádiz iniciaron la experiencia de *utilizar software libre como herramienta para la docencia de las Matemáticas*. ¿Por qué esta decisión?.

A pesar de que en niveles educativos previos al universitario la comunidad autónoma andaluza ha apostado firmemente por el uso de software libre para la docencia, dentro del ámbito de la Universidad de Cádiz, como en el resto de las universidades andaluzas y españolas, sigue imponiéndose un modelo basado en el uso de software privativo. La elección de los programas

usados en el aula se ampara en la libertad de cátedra del profesor y se guía por consideraciones que deberían ser objetivas, algunas de las cuales como la calidad del producto y su adecuación a los descriptores de la asignatura son, por supuesto, requisitos previos. Pero en la mayoría de los casos, existe más de un programa informático que cumple razonablemente bien los requisitos anteriores, por lo que, para la definitiva elección, es preciso recurrir a consideraciones adicionales. Con frecuencia, estas consideraciones han estado relacionadas con la tradición y la inercia a usar las mismas herramientas de años anteriores.

Aunque raramente se tiene en cuenta un cuestión que, en la práctica, puede tener una gran importancia: la licencia del producto o los productos elegidos y las connotaciones que este factor implica.

Las primeras de estas connotaciones, quizás aquellas que pueden resultar más evidentes, están relacionadas con las restricciones que los programas con licencia privativa imponen para su uso. Por ejemplo, en la mayor parte de los casos, este tipo de licencias impone restricciones sobre el número de ordenadores en los que el software puede ser usado o en la localización de los mismos. Esto significa que, para que un programa pueda ser usado en las aulas, es necesario que la universidad apruebe la inversión necesaria para la adquisición de suficientes licencias y además que éste no podrá ser utilizado en puestos diferentes a aquellos en los que su uso esté autorizado: aulas de informática distintas de las previstas, puestos de libre acceso, hogares de los alumnos o de los profesores...

Además, las restricciones de tipo económico dificultan la posibilidad de que los alumnos puedan utilizar y comparar diferentes herramientas en el aula. Pero la velocidad con la que evoluciona la sociedad de la información puede hacer que tecnologías que hoy son hitos incuestionables sufran mañana severos cambios e incluso lleguen a ser superadas y olvidadas en cuestión de años. Por tanto *una formación basada en la excesiva dependencia de una única herramienta comercial, puede llegar, con el tiempo, a ser obsoleta*. Los estudiantes deberían estar formados en habilidades generales, en conocimiento neutral, y no en los productos concretos de una marca comercial. Sólo de esta manera se garantizará el carácter universal de los conocimientos adquiridos y se evitará que la no disponibilidad de un producto o sus carencias evidencien las lagunas del proceso formativo. El software con licencia libre, como contrapunto, permite y de hecho fomenta el *disponer de varias herramientas a la vez, complementarias o capaces de interactuar entre sí*, cada una de las cuales contará con sus puntos fuertes y sus debilidades. Aunque el profesor se decante por una de ellas, siempre podrá ofrecer a sus alumnos la enriquecedora posibilidad de experimentar con otras, de resolver un mismo problema desde distintas perspectivas y de saciar su curiosidad a aquellos que cuenten con mayores inquietudes.

Conviene destacar, además, que la dependencia de una herramienta privativa en el ámbito educativo conlleva problemas éticos añadidos, puesto que de forma irremisible provoca en el alumnado la *seducción por una marca cuyo precio hará que, en la mayoría de los casos, no pueda ser adquirida legalmente*, incitando su copia ilegal¹. En el otro extremo se sitúa el caso del software libre permitiendo que el profesor comparta con sus alumnos, con toda legalidad, las herramientas utilizadas (quizás, acompañadas de material docente propio), facilitándoles reproducir en sus hogares el entorno de trabajo del aula. Más aún, al usar en el aula una herramienta con licencia libre, el profesor cuenta con ventajas adicionales a la hora de la planificación y el desarrollo de la asignatura, derivadas de *tener la garantía de que un programa con software libre podrá ser instalado y usado por los alumnos en su propio domicilio*², y de que esta herramienta podrá ser instalada y usada en tantos puestos como sea necesario.

Por otra parte, en un nivel superior, el software libre ofrece la interesante posibilidad de que los alumnos puedan acceder al código fuente, a la forma en que está programada la herramienta que están utilizando en clase. Esto tiene, en matemáticas, el gran valor de poder estudiar la manera en la que un programa de primer nivel implementa en la práctica los algoritmos que han sido estudiados en las clases teórico-prácticas. El poder observar y modificar el código de un programa profesional, utilizado por miles de personas de todo el mundo, constituye una experiencia tre-

¹ Como corrobora la encuesta final realizada a los alumnos (sección ??)

² Casi la totalidad de los cuales cuenta con ordenador propio, hoy día, como asimismo se recoge en la encuesta final

mendamente gratificante, de gran valor docente, como refuerzo y motivación.

Y en último lugar, uno de los argumentos más importantes pero, con frecuencia, no suficientemente valorado, debido quizás al desconocimiento del software libre y a la asimilación social de los valores que conlleva el software privativo: impulsando el software en el aula y con él los valores éticos asociados, *estaremos basando la educación en pilares como la libertad, el conocimiento, la solidaridad y la colaboración*, metas expresamente reconocidas en los estatutos de la Universidad de Cádiz y de todas las universidades andaluzas y españolas.

2. Requisitos para las herramientas matemáticas

Se detallan a continuación los requisitos que debería proporcionar un programa para poder ser usado en los primeros niveles educativos de una titulación universitaria que cuente con ciertos requerimientos de cálculo científico.

- *Nivel 0*. Requisitos mínimos:
 - *Calculadora científica*. El requerimiento más básico: se trata de poder realizar todo tipo de operaciones numéricas, almacenar valores en variables, utilizar funciones exponenciales, logarítmicas, trigonométricas, tablas de valores leídos de un fichero...
 - *Resolución numérica de ecuaciones* y sistemas de ecuaciones, calculando (probablemente de forma aproximada) valores numéricos que solucionan las ecuaciones.
 - *Gráficas de funciones 2D y 3D* de distintos tipos: implícitas/explicitas, curvas de nivel, paramétricas,...
 - *Programación* de algoritmos que se ejecuten automáticamente, a partir de determinados parámetros, y produzcan un resultado.
- *Nivel 1*. En función de las necesidades de la asignatura, puede ser conveniente el utilizar programas que se centren en el cálculo con expresiones numéricas de tipo matricial o bien en la manipulación simbólica de expresiones algebraicas y simbólico. Aunque, con frecuencia, herramientas originalmente diseñadas para el cálculo simbólico son aptas para ser usadas para cálculo numérico y matricial (y viceversa), su diseño y su filosofía justifica que las herramientas matemáticas de cálculo se puedan agrupar en dos grupos diferentes:
 - *Tipo A*. Programas de cálculo simbólico. Son programas optimizados para las siguientes operaciones:
 - *Manipulación algebraica de expresiones*. Se trata de poder manipular (factorizar, desarrollar, simplificar, ...) expresiones algebraicas que incluyan variables y parámetros, incluyendo aritmética racional.
 - *Resolución simbólica de ecuaciones* y sistemas de ecuaciones.
 - *Cálculo diferencial e integral*: Cálculo de límites, derivar e integrar funciones (de forma exacta)...

Entre los ejemplos más conocidos en el mundo del software privativo: *Mathematica*, *Maple*, etc. En las siguientes secciones estudiaremos los más representativos con licencia libre.

- *Tipo B*. Programas de cálculo numérico y matricial, de los cuales el más conocido (con licencia privativa) es, sin duda, *Matlab*. Son programas centrados en optimizar y hacer fáciles a la vez las operaciones con expresiones numéricas.
 - *Cálculo matricial*: Operaciones numéricas con vectores, matrices, determinantes, trasposición, inversión, cálculo de autovalores,...
 - *Manipulación de rangos* de valores numéricos, extracción de rangos de valores de un vector y de submatrices.

Parte II

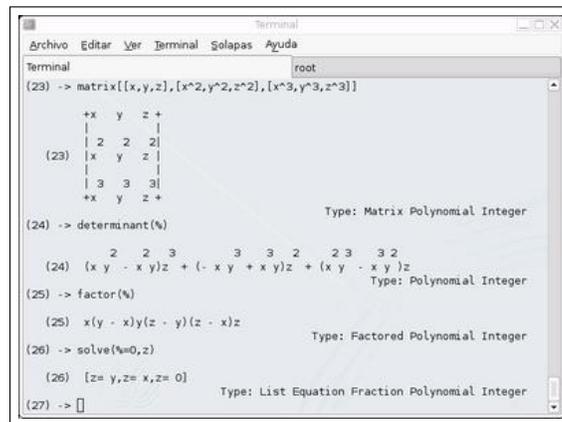
Censo y análisis previo de las herramientas disponibles

A continuación analizamos algunas de las opciones más extendidas en el mundo del software matemático con licencia libre, desde el punto de vista de su uso en la universidad (aunque, probablemente, lo que sigue sea de interés para otros niveles educativos).

Por supuesto, no están todas las herramientas relacionadas con las matemáticas disponibles con licencia libre (sería prácticamente imposible, dado su cantidad) y existen numerosas omisiones algunas de ellas intencionadas (excelentes programas relacionados con la geometría, fractales, calculadoras, juegos,...), pues no eran requeridas por las circunstancias que motivaron el presente estudio. Por otra parte, el análisis de los programas censados no ha sido realizado de forma rigurosa y lleva, por tanto, una cierta dosis de subjetividad, aunque se haya tratado de minimizarla.

3. Programas de cálculo simbólico

3.1. Axiom



```

Terminal
Archivo Editar Ver Terminal Solapas Ayuda
Terminal root
(23) -> matrix[[x,y,z],[x^2,y^2,z^2],[x^3,y^3,z^3]]
      +x y z +
      | 2 2 2 |
      | x y z |
      | 3 3 3 |
      +x y z +
      Type: Matrix Polynomial Integer
(24) -> determinant(%)
      2 2 3 3 3 2 2 3 3 2
      (x y - x y)z + (- x y + x y)z + (x y - x y )z
      Type: Polynomial Integer
(25) -> factor(%)
      (25) x(y - x)y(z - y)(z - x)z
      Type: Factored Polynomial Integer
(26) -> solve(%=0,z)
      (26) [z= y,z= x,z= 0]
      Type: List Equation Fraction Polynomial Integer
(27) -> []
  
```

Figura 1: Axiom, captura de pantalla

Axiom [2] es un potente sistema de computación científica, creado en 1971 por IBM y comercializado con licencia privativa. No fue hasta septiembre de 2002 cuando Axiom fue liberado por el NAG (Numerical Algorithms Group) [10], que lo había comprado a IBM en años 1990. A partir de su liberación, su comunidad de desarrolladores se está centrando en su extensión, incluyendo aspectos como su interfaz de usuario o soporte de gráficos.

Según el *Axiom Book* [1] (un libro de más de mil páginas, disponible en formato pdf a través de internet), la potencia de Axiom se basa en sus excelentes características estructurales y a su expansibilidad sin límites: es un sistema abierto, modular y diseñado para soportar un gran número de nuevas características con un mínimo incremento en su complejidad estructural”.

Características:

- Existe una excelente documentación.
- Es un programa sólido, potente y expansible, con una amplia comunidad que lo soporta.

- Soporte de datos estructurados, moderno lenguaje de programación que impulsa la programación estructurada y el chequeo fuerte de tipos, como en Pascal o en Ada.
- Su interfaz de usuario, por defecto, es en modo consola de texto, aunque admite otras posibilidades, desde su ejecución en un buffer de emacs hasta el uso de $\text{T}\text{E}\text{X}\text{m}\text{a}\text{c}\text{s}$ (sección 5.1), un editor de textos basado en $\text{L}\text{A}\text{T}\text{E}\text{X}$ que permite empotrar sesiones interactivas con otros programas, entre ellos Axiom.
- Disponible para distintos sistemas operativos: GNU/Linux, Mac OSX, MS Windows,...
- Gráficos: En el momento de escribir estas líneas, el soporte de gráficos 2D todavía no estaba disponible en Debian GNU/Linux³, aunque estaba a punto de ser incluido.

3.2. YACAS

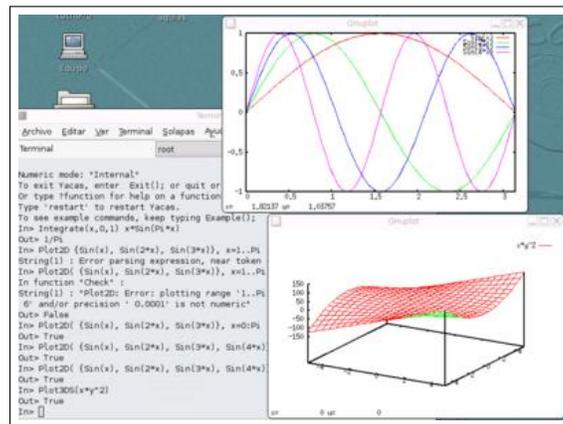


Figura 2: YACAS, captura de pantalla

YACAS [24] (*Yet Another Computer Algebra System*) es un entorno de cálculo simbólico de propósito general. Incluye todas las capacidades básicas que se suponen en este tipo de programas: cálculo en precisión arbitraria, aritmética racional, números complejos, cálculo de derivadas, resolución de ecuaciones (simbólica y numéricamente), etc.

Características:

- Es un programa moderno, escrito en C++, que está siendo desarrollado por una comunidad muy dinámica.
- Incluye una amplia documentación.
- Existen versiones para numerosos sistemas (GNU/Linux, Mac OSX, distintos sabores de UNIX, MS Windows, etc.)
- Interfaz con usuario: Se utiliza la clásica consola de texto para introducir los comandos de YACAS, aunque existe una interfaz de usuario propia, todavía experimental, llamada *proteus* y basada en la librería gráfica *fltk*. Además, existe la posibilidad de utilizar como interfaz el programa $\text{T}\text{E}\text{X}\text{m}\text{a}\text{c}\text{s}$ (sección 5.1).
- Gráficos

³Guadalinux_UCA, distribución basada en Debian GNU/Linux, fue la plataforma utilizada en el presente informe

- YACAS incluye soporte de gráficos 2D a través de gnuplot [7], conocida herramienta de gráficos 2D y 3D interactiva y orientada a línea de comandos.
- Además, existe soporte de gráficos 3D mejorado siempre que se cuente con el programa Superficie [18], que proporciona gráficos de gran calidad mediante OpenGL. Se trata de un programa en desarrollo que ya ofrece interesantes perspectivas. Eso sí, existen versiones para GNU/Linux pero no para MS Windows.

3.3. GiNaC

Acrónimo recursivo: GiNaC is not a Computer Algebra System. Ginac [6] es una biblioteca en C++ para realizar cálculos simbólicos. Incluye el programa *ginsh*, una interfaz interactiva que permite trabajar con la librería GiNaC a través de un intérprete de comandos.

Aunque *ginsh* puede tener aplicaciones directas a la docencia, más interesante puede ser el uso de la librería GiNaC como motor de cálculo en un programa C++. Esto puede ser especialmente interesante, por ejemplo, en computación científica, permitiendo manipular algebraicamente datos matemáticos (por ejemplo, funciones indicando valores iniciales o de contorno) leídos de un fichero o introducidos interactivamente por el usuario.

Otro tipo de aplicaciones existentes: EQC es un preprocesador de ficheros \LaTeX que, usando GiNaC, permite manipular algebraicamente expresiones matemáticas escritas en \LaTeX . Existe un modo para su uso integrado en Emacs.

3.4. Pari

Pari [12] es una biblioteca de álgebra computacional escrita, en este caso, en C e inicialmente diseñada para teoría de números por Henri Cohen (Université Bordeaux I, France) y otros colaboradores. Actualmente, permite numerosas operaciones adicionales (matrices, polinomios,...). Existe una interfaz, GP, que facilita el uso de Pari mediante un intérprete de comandos interactivo. El entorno 5.2 también utiliza Pari como motor matemático.

3.5. Maxima

En los años 60, el prestigioso MIT (*Massachusetts Institute of Technology*) inició el proyecto MAC (*Mathematics and Computation*) con el objetivo de desarrollar herramientas computacionales “inteligentes” para ser aplicadas al estudio de modelos físicos. Como fruto, nació *Macysma* (*MAC's SYmbolic MANipulator*, que constituyó el primer proyecto con éxito en la automatización de operaciones matemáticas de tipo simbólico. En este sentido, puede considerarse el padre de todos los programas de cálculo simbólico actuales. *Macysma* originó numerosas variantes, entre ellas *DOE-Macysma*, subvencionada por el Department of Energy (DOE) de los EE.UU. De esta variante nació Maxima [9], que fue mantenida entre 1982–2001, por W. Shelter, que intercedió para que, en 1998, el DOE autorizara su distribución con licencia libre (GPL).

Características:

- Cuenta con el apoyo una comunidad de desarrollo muy dinámica, que trabaja constantemente para su actualización, mejora y soporte.
- Disponible para distintas plataformas, tales como GNU/Linux, MS Windows, etc.
- Cubre sobradamente las necesidades básicas de cálculo simbólico para un alumno de primer curso de una titulación universitaria.
- Gráficos:

- Tiene comandos propios para la representación de gráficos 2D y 3D que, por defecto, utilizan el programa *gnuplot*. Por tanto, muchas de las ventajas e inconvenientes de los gráficos en Maxima son achacables a *gnuplot*.
 - Los comandos de representación de gráficos tienen previsto el utilizar, a bajo nivel, otros programas distintos de *gnuplot*, como *geomview* que permite la visualización de superficies 3D con calidad OpenGL e interactuar con ellos, cambiando luces, texturas, etc.
- Maxima cuenta con manuales y documentación que pueden descargarse libremente de internet. Además, existen numerosos recursos bibliográficos del original Macsyma, que son en un alto porcentaje, aplicables a Maxima.
 - Maxima incorpora un completo lenguaje de programación propio, derivado del Macsyma original. Eso sí, este lenguaje no es demasiado intuitivo y no incorpora las características de lenguajes más modernos (como estructuras de datos o programación orientada a objetos).
 - Interfaz con usuario: por defecto, utiliza una interfaz poco amigable (de tipo consola de texto). Pero existen numerosos programas libres que actúan como interfaz sobre Maxima, algunos de ellos de tipo genérico, como \TeX macs (estudiado en la sección 5.1), Emacs (sección 5.3) o WIMS (sección 5.2) y otros específicamente diseñados para facilitar el uso de Maxima, como *xMaxima* (sección 3.5.2) o *wxmaxima* (sección 3.5.1),

3.5.1. wxMaxima

wxMaxima [23] es una interfaz que pretende facilitar el uso de Maxima, a través de una serie de menús, barras de botones y ventanas asociadas.

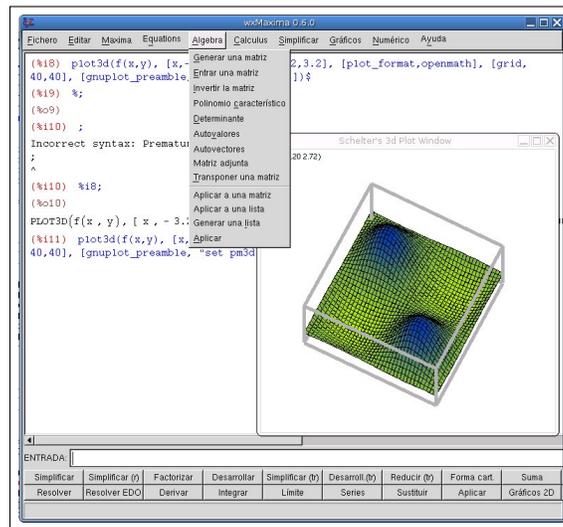


Figura 3: *wxMaxima*, captura de pantalla

Características

- Gracias a que utiliza la biblioteca gráfica *wxwidgets*, existen versiones tanto para sistemas operativos GNU/Linux como para Windows.
- Integra distintos documentos de ayuda de Maxima de forma que éstos son fácilmente accesibles a través del ratón.

- Introducción de matrices, creación de gráficas, cálculo de límites, de derivadas o integrales, etc. Son numerosos los conceptos que cuentan con ventanas específicas gracias a las cuales se simplifica el uso de Maxima.
- Para las personas que prefieren usar los comandos de Maxima, existe una entrada de diálogo que permite introducirlos, acceder y editar los comandos anteriores, etc.

3.5.2. xMaxima

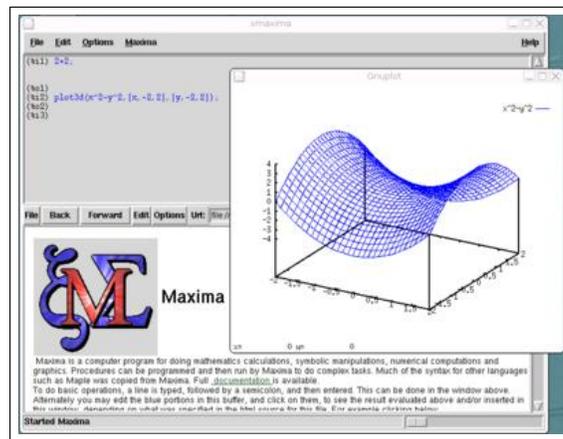


Figura 4: xMaxima, captura de pantalla

xMaxima es una interfaz gráfica que se incluye por defecto con el programa Maxima y que tiene una filosofía similar a wxMaxima. Históricamente, ha sido la interfaz gráfica “oficial” de Maxima y, de hecho, en algunos sistemas como MS Windows es la que se arranca por defecto. Sin embargo, presenta algunas desventajas con respecto a wxMaxima:

- Su interfaz gráfica, basada en las librerías *TCL/TK*, es antigua y tiene menos prestaciones.
- No cuenta con ventanas y menús que puedan servir como apoyo para la introducción de comandos en Maxima.

4. Programas de cálculo numérico y matricial

4.1. EULER

EULER [5] es un programa más de cálculo numérico matricial del estilo de Matlab y Octave (aunque no intenta ser compatible con ellos). Es un programa sencillo y flexible, que contiene, tanto en sus versiones GNU/Linux como MS Windows una interfaz de usuario propia bastante cómoda (utilizando las librerías *GTK* en GNU/Linux) y buenos gráficos 2D y 3D.

Otras características:

- moderno lenguaje de programación, con soporte valores por defecto para parámetros, número de parámetros variable, paso de funciones,...
- Incluye las funciones básicas requeridas para este tipo de programas: integración y diferenciación numérica, estadística, interpolación, transformada de fourier, ecuaciones diferenciales, etc.

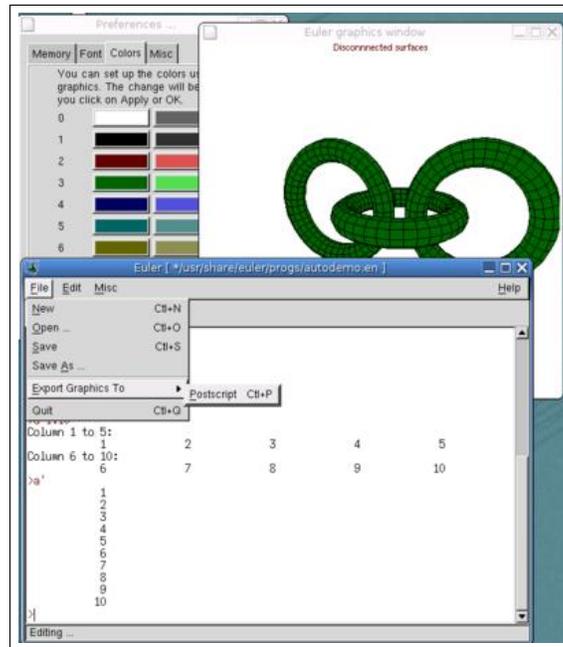


Figura 5: EULER, captura de pantalla

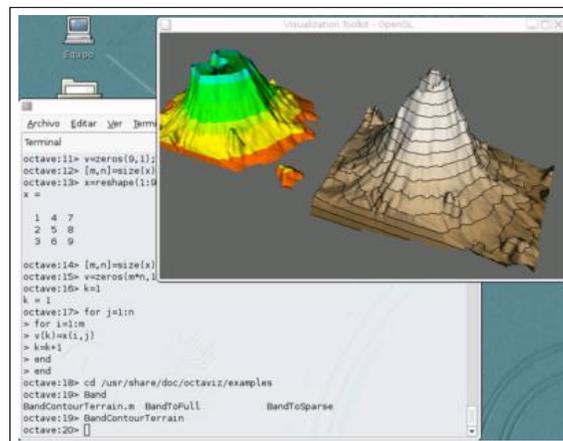


Figura 6: Octave, captura de pantalla

4.2. Octave

Octave [11] fue creado en 1988 por John W. Eaton como soporte a un libro de texto sobre ingeniería química en la Universidad de Texas con el objetivo de crear un lenguaje de cálculo numérico fácil de usar, interactivo y con una amplia comunidad de usuarios. En 1994 con la versión 1.0, obtuvo su madurez para la enseñanza, la investigación y la empresa. Sus autores lo definen como *A high-level interactive language, primarily intended for numerical computations that is mostly compatible with MATLAB.*

Existen distintas interfaces de usuario (entre ellas, algunas de propósito general, como `texmacs` (ver sección 5.1) y `emacs` (sección 5.3) y otras específicas, como `Octivate` (basada en `.NET+GTK` y, por tanto, multiplataforma, aunque se encuentra todavía en fase de desarrollo).

Octave cuenta con gráficos 2D y 3D a través de funciones que son, a priori, compatibles con `Matlab` y que utilizan `gnuplot` como motor gráfico, aunque también existe un soporte alternativo de gráficos 2D y 3D utilizando la librería `plplot` [14]. Además, existen extensiones para el uso de gráficos 3D de alta calidad mediante `VTK` [21] (como el que puede apreciarse en la captura de pantalla).

Además de estas extensiones para el uso de gráficos, Octave es fácilmente extensible mediante nuevas funciones creadas en el lenguaje de programación que incorpora (que también es, a priori, compatible con `Matlab`) o directamente en `C++`. Algunas de ellas, como las orientadas a la programación distribuida (con `MPI`), `wavelets`, lectura y escritura de vídeos de tipo `MPEG`, uso de las librerías `GTK`, etc, le proporcionan una gran potencia como plataforma para el cálculo numérico y científico. Puede encontrarse una interesante recopilación de información de este tipo en la página web del *Debian Octave Group* [4]

4.3. R

`R` [15] es un entorno (y un potente lenguaje) de *cálculo estadístico*, similar al programa `S` (con licencia privativa) y en gran parte compatible con él.

Entre sus características se encuentra además el contar con una abundante cantidad de operadores y funciones relacionadas con el cálculo matricial y que lo hacen apto para ser usado con tal propósito, motivo por el que se ha optado por analizarlo en esta sección.

`R` destaca, además de por la potencia de sus faceta estadística, por su efectividad, extensibilidad y por incluir una excelente serie de herramientas gráficas propias, relacionadas con la representación y análisis de datos. `R` es el programa que se ha realizado para analizar los datos de la encuesta final, incluida en la sección ??.

4.4. Scilab

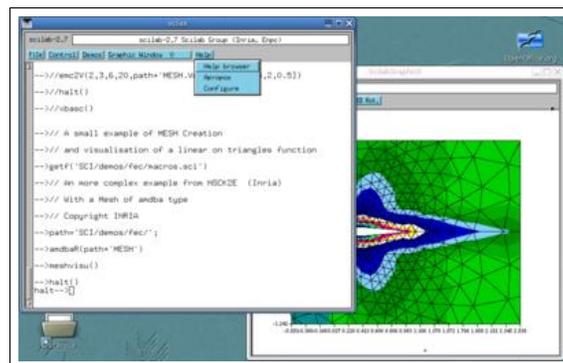


Figura 7: Scilab, captura de pantalla

Scilab [16] es un programa para la computación numérica y científica desarrollado a partir de 1990 por investigadores las insituciones francesas INRIA y ENPC.

Por lo demás, Scilab es un excelente programa, con cierto grado de compatibilidad con Matlab y con una interfaz de usuario integrada (línea de comandos incluyendo menús, gráficos, ayuda,...) que puede resultar agradable a los amantes de los entornos de ventanas, aunque la versión GNU/Linux (no así la versión MS Windows) resulta menos vistosa debido a las librerías gráficas utilizadas. Cuenta con módulos muy interesantes, entre ellos programación distribuida con PVM, programación gráfica (a golpe de ratón) y posibilidad de exportar en FORTRAN.

Desafortunadamente, aun siendo gratuito y estado disponible su código para su estudio y su uso, no puede considerarse 100% software libre (por ejemplo, no cumple las "Debian's Free Software Guidelines" y es considerado "non-free") debido que la siguiente cláusula en su licencia impone limitaciones a su libre distribución:

commercial use or circulation of the DERIVED SOFTWARE shall have been previously authorized by INRIA and ENPC.

4.5. Tela

Tela [19] es un lenguaje de cálculo numérico diseñado, especialmente, para tareas relacionadas con el pre-proceso y post-proceso numérico. Incluye un intérprete de comandos y soporte de gráficos 2D y 3D basados en plotmtv [13], un buen programa para la visualización de datos científicos.

Utiliza de forma nativa el formato HDF [8], uno de los estándares específicamente diseñados para la lectura y escritura e intercambio de datos científicos.

5. Interfaces de usuario genéricas

En esta sección se describen interfaces que pueden facilitar al usuario el interactuar con varios (en ocasiones casi todos) los programas listados anteriormente.

5.1. T_EXmacs

T_EXmacs [20] es un proyecto iniciado en 1998 por Joris van der Hoeven en el C.N.R.S (Instituto Nacional Francés para la Investigación Científica) y algunas personas más, con el propósito de escribir una aplicación destinada a redactar textos matemáticos de forma sencilla. Su nombre tiene raíces en el sistema TeX, en el que se basa y en el conocido editor de textos Emacs, del que toma parte de su filosofía.

El resultado es un editor de textos de tipo WYSWYG⁴ que permite la creación de textos matemáticos y científicos con la calidad que aporta L^AT_EX, utilizado como motor de maquetado.

Pero el verdadero potencial de T_EXmacs estriba en la posibilidad de incorporar y utilizar con comodidad sesiones interactivas de motores de álgebra simbólica (entre ellos entre ellos Axiom, YACAS, Maxima, etc) o de cálculo numérico (entre ellos Octave y R). En este sentido, T_EXmacs es "la interfaz universal" para programas matemáticos, aportando a los mismos ventajas adicionales: introducción de expresiones con la ayuda del ratón, control e interrupción de los procesos, mejora del aspecto de las salidas matemáticas, etc.

5.2. WIMS

Según el libro "Iniciación a WIMS" [25] de Georges Khaznadar, "WIMS es un programa servidor que distribuye unas páginas de ejercicios de ciencias a través de las redes. Respeta el estándar

⁴What You See is What You Get, lo que ves (en pantalla) es lo que obtienes (al imprimir)

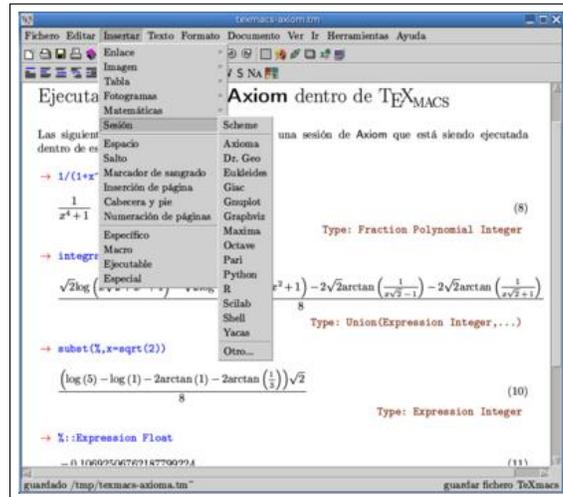


Figura 8: TExMacS (ejecutando una sesión Axiom), captura de pantalla

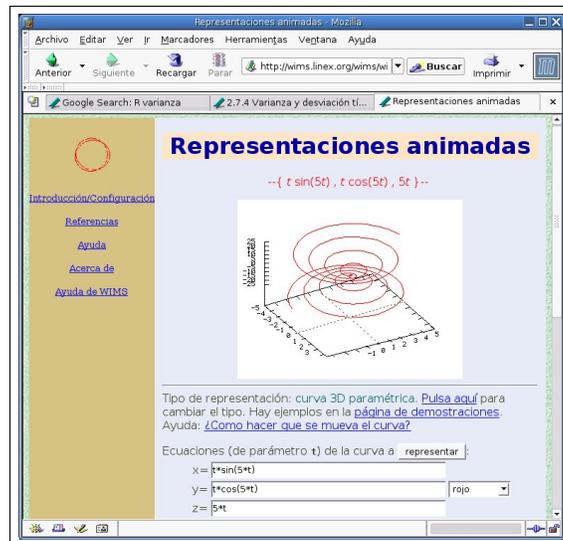


Figura 9: WIMS, captura de pantalla

del World Wide Web Consortium (W3C), por lo que WIMS es accesible desde cualquier navegador Web. WIMS contiene gran cantidad de ejercicios y de herramientas matemáticas. El motor de WIMS y una gran parte de los módulos actualmente distribuidos son debidos a Gang XIAO.”

“El acrónimo W.I.M.S. significa, entre otras cosas, Web Interactive Mathematical Server: Servidor Web Interactivo de Matemáticas. Se puede acceder por Web, como por cualquier servidor http. WIMS está constituido por un rico paquete de módulos científicos y matemáticos y es fácil de usar como recurso para la enseñanza. Además, una pequeña formación es suficiente para desarrollar uno mismo unos módulos extraordinarios.”

En la práctica, WIMS [22] es un recurso docente potente, fácil de usar y con interesantes posibilidades, para usar el cual basta con tener un navegador web y conectarse a cualquiera de los servidores distribuidos en Internet. Dentro de España, la Junta de Extremadura está promoviendo la traducción de los módulos que aún no tenían una versión en castellano, a la vez que mantiene su propio servidor [17].

WIMS utiliza, a bajo nivel, distintos motores matemáticos, como Maxima, o Pari, gráficos, como gnuplot, o de cualquier otro tipo, lo que lo convierte en una “navaja suiza” muy extensible, que trasciende a campos del saber (física, química,...) más allá de las matemáticas, para el que fue inicialmente diseñado.

5.3. Emacs

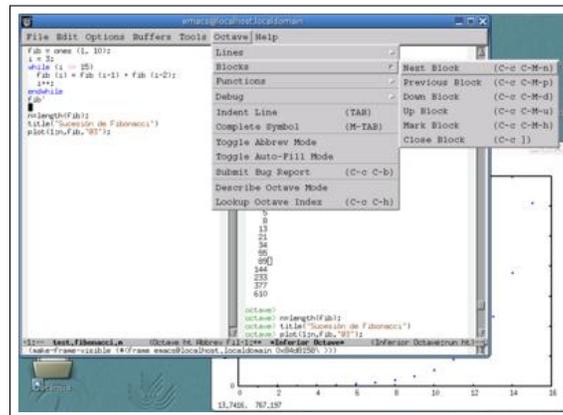


Figura 10: Emacs (utilizando el modo Octave), captura de pantalla

Emacs es mucho más que un potente editor de textos. Tremendamente configurable, constituye una plataforma de desarrollo con modos específicos para decenas de lenguajes de programación (desde los lenguajes más utilizados con propósito general, como C, hasta otros más específicos, como awk). Emacs incluye soporte para $\text{T}_{\text{E}}\text{X}$ / $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ (incluyendo interesantes entornos como whizzy o preview-latex) y por supuesto, modos para todos los programas matemáticos que se han descrito en las páginas anteriores.

Emacs incluye, por ejemplo, modos de trabajo para Octave y Maxima. Con respecto a Maxima, además de un modo que permite ejecutar una sesión maxima dentro de Emacs, existe una variante, “imaxima”, que usa $\text{T}_{\text{E}}\text{X}$ para embellecer las salidas (ver imagen anterior) haciéndolas más legibles.

Otro modo interesante es “emaxima”, ideado para la redacción de documentos $\text{T}_{\text{E}}\text{X}$ / $\text{L}_{\text{A}}\text{T}_{\text{E}}\text{X}$ que incluyan sesiones o comandos de Maxima. Este modo se ha utilizado para la redacción del manual de introducción y prácticas con Maxima [26], elaborado expresamente para esta experiencia docente, usado en algunas asignaturas de la Universidad de Cádiz durante el curso 2004/2005 y publicado con licencia libre.

Referencias

- [1] Axiom book. <http://page.axiom-developer.org/zope/Plone/refs/books/axiom-book2.pdf>.
- [2] Axiom, web page. <http://page.axiom-developer.org/zope/mathaction>.
- [3] Creative commons sharealike 2.0 license. <http://creativecommons.org/licenses/by-sa/2.0/>.
- [4] Debian octave group, web page. <http://www.octave.org/>.
- [5] Euler, web page. <http://euler.sourceforge.net/index.html>.
- [6] Ginac, web page. <http://www.gnu.org/software/emacs/emacs.html>.
- [7] gnuplot, web page. <http://www.gnuplot.info/>.
- [8] Hdf, web page. <http://hdf.ncsa.uiuc.edu/>.
- [9] Maxima, web page. <http://maxima.sourceforge.net>.
- [10] Nag (numerical algorithms group), web page. <http://www.nag.co.uk/>.
- [11] Octave, web page. <http://www.octave.org/>.
- [12] pari, web page. <http://www.math.u-psud.fr/~belabas/pari/>.
- [13] Plotmtv, web page. http://csepl.phy.ornl.gov/cornell_proceedings/tutorials/Plotmtv/overview.html.
- [14] plplot, web page. <http://plplot.sourceforge.net>.
- [15] R, web page. <http://www.r-project.org/>.
- [16] Scilab, web page. <http://scilabsoft.inria.fr/>.
- [17] Servidor wims en extremadura. <http://wims.linex.org/>.
- [18] Superficie, web page. <http://superficie.sourceforge.net/>.
- [19] tela, web page. <http://www.ava.fmi.fi/prog/tela.html>.
- [20] Texmacs, web page. <http://www.texmacs.org/>.
- [21] Vtk, web page. <http://plplot.sourceforge.net>.
- [22] Wims, web page. <http://wims.unice.fr/>.
- [23] wxmaxima, web page. <http://wxmaxima.sourceforge.net>.
- [24] Yacas, web page. <http://yacas.sourceforge.net>.
- [25] Georges Khaznadar. *Iniciación a wims*. <http://libro-wims.software-libre.org/wims.pdf>.
- [26] M. Victoria RedondoÑeble and J. Rafael Rodríguez Galván. *Introducción a Maxima*. Universidad de Cádiz, 2005.