

Mi experiencia usando software libre en titulaciones técnicas: octave

DAVID J. LÓPEZ¹

1. Departamento Matemática Aplicada y Estadística.
Universidad Politécnica de Cartagena

david.lopez@upct.es

Resumen

El objetivo del presente artículo no es la presentación de un análisis exhaustivo de software científico libre adecuado para titulaciones técnicas. Por una parte, porque siendo el autor del artículo un usuario bastante novel en estas lides se considera poco capacitado para hacerlo. Pero, por encima de todo, porque no quiere hacerlo. Así que, sin que sirva de precedente, abandonará el lenguaje culto, dejará de utilizar la tercera persona y se centrará en su propia experiencia. La primera sección versará sobre cómo llegó al software libre, y la segunda sobre el software que usa habitualmente en su vida laboral como investigador y docente en la escuela de Teleco, particularmente OCTAVE. El lector que sólo esté interesado en esta última parte puede sin pudor avanzar a la segunda sección.

1 Mi experiencia con el software libre

Si has decidido leer esta sección y saber qué me ha llevado a usar software libre permíteme que te tutee y me presente. Hola. Me llamo David, estudié Ciencias Matemáticas en la Universidad de Valladolid, y llevo 16 años dando clase de matemáticas en ingenierías de Teleco e Industriales. Mi especialidad de licenciatura, mi tesis y buena parte de mi investigación y mi docencia han estado enfocadas fundamentalmente hacia los métodos numéricos. En Valladolid usé lenguajes compilados como FORTRAN y C, e interpretados como MAPLE. Cuando comencé a trabajar en la UPCT hace 11 años cambié MAPLE por MATHEMATICA, del que fui un usuario activo hasta que hace un par de años decidí utilizar en mi docencia únicamente software libre, y lo reemplacé por OCTAVE y MAXIMA.

1.1 Qué es el software libre

Una respuesta técnica ocuparía varias páginas de disertaciones sobre matices y gráficos [1, 2, 3] que a duras penas soy capaz de seguir. Hay incluso una confusión semántica porque en inglés *free* significa tanto *libre* como *gratis* (de hecho hay angloparlantes que usan el término *libre software* para evitar la ambigüedad de *free software*). Ilustraré lo que para mí es el software libre con un ejemplo. Yo he liberado mis apuntes, prácticas, . . . , para que los use quien quiera. Los he puesto con licencia Creative Commons Reconocimiento-NoComercial-CompartirIgual [4], lo que viene a decir que te permito (más aún, te agradezco) que uses mis apuntes si te gustan. No te pido nada a cambio, si acaso un mínimo de civismo: que no tengas el morro de decir que los has hecho tú ni de venderlos. No estoy haciendo nada especial, desde el estudiante que deja los apuntes al compañero que ha faltado un día a clase hasta el profesor que organiza una visita de los institutos a su laboratorio, muchos hacemos cosas en nuestro trabajo cosas que podríamos calificar como “altruistas”.

Lo que yo busco en el software es la transcripción de ese altruismo a la informática. La Free Software Foundation ofrece una definición precisa [5], incluyendo el uso, estudio, mejora y distribución los programas mejorados. Como el estudio de un programa requiere de forma implícita del acceso al código fuente, el concepto de software libre es muy similar al de *código abierto* [6]. Hay pequeñas diferencias de matiz entre ambos conceptos, de hecho hay quien se refiere a ambas simultáneamente con las siglas FOSS (*free and open source software*). Personalmente a mí no me importan demasiado las etiquetas, busco “algo más que software gratis”, y a este concepto algo difuso es al que me refiero como software libre.

1.2 Por qué tienes que usar software libre

Pregunta mal formulada, no soy quien para decirle a nadie lo que tiene que hacer. Si estás leyendo esto porque buscas un empujón para usar software libre, ese empujón puede ser el artículo *Por qué las escuelas deberían usar exclusivamente software libre* de Richard Stallman [7]. Son solo 5 minutos de lectura y creo que no te arrepentirás.

1.3 Por qué he usado software no libre en mi docencia

Podría responder que porque (casi) todo el mundo a mi alrededor lo hacía. O porque mis estudiantes nunca se quejaron porque les estaba forzando a pagar por el software o a agenciárselo de forma ilegal si querían usarlo en sus casas. O porque mis universidades nunca protestaron y nunca provocaron que reflexionara si realmente yo necesitaba que pagaran esas licencias de software o si en cambio podía ahorrárselas a la universidad cambiando de software. (¿He dicho ahorrarSElas? Debería haber dicho ahorrarMElas, ahorrarNOSlas. Con tanta crisis y recortes flotando en el ambiente, me sorprende no haber escuchado nada sobre reducir gastos en software comercial innecesario.)

Podría responder todas esas cosas, pero no son más que excusas. La verdad, mi verdad, es más incómoda. En realidad he usado durante muchos años software privativo porque me han importado poco los demás. Siendo recién licenciado hace 15 años sí me preocupé de ofrecer a mis estudiantes el compilador DJCPP para que hicieran sus prácticas, pero con el paso del tiempo me he ido haciendo perezoso. Ni me molesté en levantar el teléfono para preguntar al servicio de informática en qué consistía la licencia campus del software que utilizaba en clase y si los estudiantes podían usarlo en sus casas. Y tampoco me preocupé de que, cuando terminaran sus estudios, esos programas no les servirían salvo que compraran la licencia del software con el que los algoritmos estaban programados.

Y ni siquiera creo que la prioridad deba ser la económica. En el fondo sabía que estaba haciendo las cosas mal, y que debía hacerlas bien, o como mínimo intentarlo. Y poco a poco fui seleccionando el software que utilizaba. Primero como usuario de escritorio, comencé cambiando mi navegador web por el Mozilla Firefox, y resultó que era igual que el que llevaba años usando. Después cambié el correo por el Mozilla Thunderbird, luego el reproductor multimedia por VLC y finalmente puse OpenOffice. Y todo me funcionaba igual de bien que siempre. Como usuario de escritorio ya usaba casi totalmente programas libres multiplataforma, así que el “salto al vacío” que supone sustituir windows por linux (ubuntu en mi caso) resultó bastante sencillo porque básicamente seguía usando los mismos programas. De hecho 3 años después sigo usando estos mismos, salvo que al bifurcarse OpenOffice me quedé con la rama LibreOffice.

Durante un tiempo pedí al servicio de informática versiones linux del software científico que usaba habitualmente para mi trabajo pero, cansado de licencias que caducaban, hace un par de años decidí completar mi travesía en el desierto y probar software científico libre. Este artículo es la prueba de que estoy encantado con el cambio.

1.4 Por qué ahora uso software libre

Personalmente no tengo problema en usar software privativo, gratuito o no, si no encuentro software libre que cumpla mis necesidades. Afortunadamente eso me ocurre pocas veces y para cosas bastante residuales, así que puedo decir con cierto orgullo que la inmensa mayoría del software que uso es libre. Además de las consideraciones filosóficas que he hecho hasta ahora, uso software libre porque he encontrado programas, tanto de carácter general como científicos, como mínimo igual de **buenos**, igual de **sencillos de usar**, igual de **actualizados**, igual de **documentados** e igual de **traducidos al castellano** que sus análogos privativos. Uso el signo *igual* para que no sonar a vendemotos, pero en muchos casos se me queda corto. Para mí linux es *mayor* que windows porque ni el mejor antivirus puede ser mejor a que no existan virus. Un par de ejemplos académicos de software libre que funciona muy bien pueden ser el aula

virtual (moodle) que usamos en la UPCT, o el lenguaje en el que está editada esta revista (latex).

Gracias al software libre ahora sé hacer cosas que antes ni imaginaba que se pudieran hacer, como escribir en un solo documento problemas diferentes para cada uno mis 40 estudiantes, y con solo un clic de ratón crear un pdf de las soluciones de cada uno de los 40 problemas distintos. Y más todavía, en menos de medio minuto puedo crear otros 40 (ó 400) distintos para el curso que viene. No digo que no se pueda, pero yo esto no lo sé hacer con software privativo y sí con latex [8] y MAXIMA [9].

1.5 Por qué usar software libre te resultará fácil

Tú también puedes (nótese que digo digo *puedes* y no *debes*) usar software libre porque sirve para hacer cosas tales como navegar por internet, mandar correos electrónicos, chatear, ver vídeos, escuchar música, editar fotos, escribir o leer documentos o copiar tus archivos en una memoria USB o un CD, y salvo que hayas nacido en Marte usarás el ordenador para hacer alguna de estas cosas. Pero además te resultará fácil porque lo usan mis hijos de 9 y 7 años y mi madre de 68, así que muy difícil no puede ser. El software libre es igual de fácil o difícil que el privativo de toda la vida: si quieres abrir un programa pincha en su icono, si quieres ver un vídeo pincha en el botón con triángulo, si quieres avanzar de página en un documento pincha en la flecha que apunta a la derecha. Las reglas son las mismas.

He de reconocer que me gusta la informática, así que la migración a software libre no me ha resultado nada traumática, más bien todo lo contrario. Al igual que a mí, hay mucha gente a la que le gusta trastear con el ordenador (y por mi experiencia diría que muchos están en la escuela de Teleco). Para nosotros no es difícil abandonar *el lado oscuro* del software. Pero hay muchas más personas para las que la informática es puramente una herramienta y no una diversión. Tal vez estés de acuerdo en la filosofía del software libre, pero si perteneces a esta mayoría supongo que agradecerías que te pusieran las cosas fáciles.

Mi consejo es que comiences cambiando programas privativos por otros libres y multiplataforma como hice yo. En internet hay infinitas bases de datos y buscadores como [10, 11] con software específico, pero si empiezas de cero es probable que no sepas qué programas son buenos y cuáles no. Pregunta. Hay una lista de correo en la UPCT [12] en la que exponer tus dudas, inquietudes, etc. No solo los de la UPCT, en general los usuarios de software libre solemos ser bastante solícitos, así que si pides ayuda en el foro del programa con el que tengas problemas es probable que en poco tiempo recibas una respuesta.

La UPCT está apostando por el software libre, el verano pasado hubo varios cursos de software libre (que yo recuerde, linux, MAXIMA, latex y R) para PDI, hay una propuesta para ampliar este año al PAS alguno de éstos, e imagino que

las delegaciones de alumnos tendrán también capacidad para proponer cursos, talleres o *install parties* si hay demanda por parte de los estudiantes. Hay al menos dos aulas de informática (el ALA de la ETSIT y la del Departamento de Matemática Aplicada y Estadística) que tienen instaladas versiones muy recientes de ubuntu/linux, además de un montón de programas libres tanto para linux como para windows.

1.6 ¿Puedes usar exclusivamente software libre?

He dejado esta pregunta para el final porque no puedo responder con rotundidad. Como usuario general de escritorio no creo que tengas grandes problemas para seguir trabajando como lo haces habitualmente. Dependiendo de hasta qué punto seas “purista” con la libertad del software es posible que acabes echando en falta alguna pequeña cosa. Quizá prefieras hacer como yo y saltarte un poco las “normas” en pequeñas cosas. Yo no tengo problemas, por ejemplo, en usar adobe reader para firmar actas o en que algunos estudiantes me envíen sus prácticas comprimidas con rar (dos ejemplos de que no es lo mismo *gratis* que *libre*).

En cuanto a software científico-técnico, depende de lo que uses. Siempre he oído que no hay ningún CAD libre de una calidad similar a los comerciales. Es fácil encontrar en internet listados con programas libres alternativos pero sinceramente no sé si alguno está a la altura porque yo no soy usuario de esa clase de programas. Por mi docencia yo necesito programas de Cálculo Numérico, de Cálculo Simbólico y de edición de textos científicos. Por suerte he encontrado excelentes programas libres para lo que yo necesito, como los citados MAXIMA para computación y latex para escritura. La siguiente sección trata sobre el programa que uso para Cálculo Numérico: OCTAVE [13].

2 Cálculo Numérico con octave

2.1 Un clon de matlab

Uno de los programas más utilizados en Ingeniería es sin duda MATLAB. Su principal alternativa libre es OCTAVE, un lenguaje libre interpretado diseñado para Cálculo Numérico, programado fundamentalmente John W. Eaton de la Universidad de Wisconsin-Madison. Aunque la idea original data de 1988, la primera versión fue publicada en 1994. OCTAVE es multiplataforma, se pueden descargar las versiones de windows y mac de [14], mientras que para linux se recomienda instalar desde los repositorios oficiales de tu distribución.

OCTAVE incluye un extenso manual de 725 páginas, que puede consultarse tanto en versiones pdf o html, como desde la ayuda interna del propio programa. Aunque tal vez nunca necesites un manual porque es probable que sin saberlo ya seas un experto en OCTAVE, gracias a que **su nomenclatura es extraordinariamente compatible con** MATLAB. Por ejemplo, en las

entradas española e inglesa de la wikipedia dedicadas a MATLAB [15, 16] aparecen multitud de códigos compatibles con OCTAVE, desde un simple

```
disp('Hola mundo');% Muestra el mensaje.
```

a definiciones de variables,

```
x = 17
x = 'hat'
y = x + 0
```

construcción de vectores,

```
x = [3*4, pi/2]
array = 1:2:9
array = 1:3:9
ari = 1:5
```

matrices particulares,

```
A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
eye(3)
zeros(2,3)
ones(2,3)
```

o funciones externas,

```
function [fa,dfa]=funcion_y_derivada(x)
fa=0;
n=1;
h=1;
dfa=0;
while (n <= (10*(x-(1/2))^2) + 10)
    fa=fa+(((x^2)^n)/factorial(2*n));
    h=h/2;
    dfa=dfa+((((-(x+h)^2)^n)/factorial(2*n))
        -(((x^2)^n)/factorial(2*n)))/h);
    n=n+1;
end
```

pasando por gráficas en 2 dimensiones,

```
x = 0:pi/100:2*pi;
y = sin(x);
plot(x,y)
```

o en 3 dimensiones,

```
[X,Y] = meshgrid(-10:0.25:10,-10:0.25:10);
f = sinc(sqrt((X/pi).^2+(Y/pi).^2));
mesh(X,Y,f);
axis([-10 10 -10 10 -0.3 1])
xlabel('\bfx')
ylabel('\bfy')
zlabel('\bfsinc (\bfR)')
hidden off
```

desde ejemplos puramente matemáticos como la exponencial de una matriz,

```
close all % Cierra todas las ventanas.
clear all % Borra todas las variables del espacio de trabajo.
clc % Limpia la pantalla.
m=[5 6;6 7]; % Declaramos la matriz m
e=zeros(2,2); % Declaramos una matriz de ceros de 2x2
for k=(1:30) % Iniciamos un for de k=1 supongamos hasta 30
    e=e+(m^k)/factorial(k) % Introducimos la ecuacion
    como serie de potencias
end % Terminamos el for
```

a ejemplos aplicados a las Telecomunicaciones, como el procesado de una señal usando polinomios trigonométricos de Fourier,

```
n=input('número de sumandos= '); % creamos una serie de pulsos
    útil para el procesados de señales y sistemas
    dentro del mundo de las telecomunicaciones
t=-2:.01:2;
pulso=zeros(1,length(t));
for k=1:n
    pulso=pulso+sin(2*(2*k-1)*pi*t)/(2*k-1);
end
plot(t,pulso)
grid
```

Todos estos ejemplos propios de MATLAB **funcionan en OCTAVE sin tocar una sola coma.**

MATLAB es muy configurable gracias a que se pueden añadir pequeños paquetes de funciones para problemas específicos (*toolboxes*), lo que permite a cada usuario ajustar el software a sus necesidades. Por ejemplo, el filtro digital extraído también de [15],

```

close all % Cierra todas las ventanas.
clear all % Borra todas las variables del espacio de trabajo.
clc % Limpia la pantalla.
Fc=200; % Frecuencia de corte.
Fm=1000; % Frecuencia de muestreo.
BT=100; % Banda de transición.
Rs=40; % Ganancias.
rs=10^(-Rs/20);
Rp=2;
rp=(10^(Rp/20)-1)/(10^(Rp/20)+1);
[n1b,wn1]=buttord(2*Fc/Fm,2*(Fc+BT)/Fm,Rp,Rs); % Orden
del filtro (función buttord).
[B1,A1]=butter(n1b,wn1); % Coeficientes
del filtro (función butter).
h1=freqz(B1,A1); % Respuesta en frecuencia (función freqz).
plot(abs(h1)) % Representación de la respuesta.

```

usa funciones específicas de la *toolbox* de procesamiento de señales de MATLAB. Como no iba a ser menos, **octave también posee *toolboxes* libres**. Usando la *toolbox* octave-signal se puede correr este ejemplo en OCTAVE nuevamente sin tocar una sola coma del código. En [14] pueden verse las *toolboxes* propias de OCTAVE, que incluyen Álgebra Lineal, Ecuaciones Diferenciales, series temporales, manipulación de imágenes, ...

Gracias a esta enorme compatibilidad entre OCTAVE y MATLAB en la asignatura de Álgebra Lineal y Métodos Numéricos de primer curso de los Grados en Ingeniería en Sistemas de Telecomunicación y en Ingeniería Telemática siempre hemos impartido las prácticas en lo que denominamos OCTAVE/MATLAB: aunque sean programas distintos, a todos los efectos para nosotros es como si fueran dos versiones del mismo, ya que todos los programas que usamos funcionan simultáneamente en ambos programas, y son profesor y estudiante quienes deciden en cuál de los dos correrlos.

2.2 Incompatibilidades entre octave y matlab

Aunque los tratemos como clones, hay pequeñas diferencias entre ambos. En general OCTAVE es algo más flexible que MATLAB, de forma que lo que funciona en MATLAB también lo hace en OCTAVE, como muestran los ejemplos de [15, 16] que he usado hasta ahora. En sentido inverso hay algunas pequeñas cosas OCTAVE admite y MATLAB no, como definiciones de funciones dentro del programa, operadores de incremento tipo ++, dobles comillas, etc. Una lista más detallada puede verse en [17].

La única pega importante que yo me he encontrado entre ambos programas es **incompatibilidad en el Cálculo Simbólico**. No siendo el punto fuerte de ninguno de los dos, justo es reconocer que OCTAVE es bastante más pobre en este aspecto y no admite muchas de las funciones de MATLAB. Además hay

diferencias de notación, como que OCTAVE renombra a las funciones simbólicas con la primera letra mayúscula o que deriva con `differentiate`, mientras que MATLAB no distingue el nombre de funciones simbólicas y numéricas y deriva con `diff`. Así por ejemplo, en [15] se incluye un código de MATLAB para aproximar segundas derivadas de funciones simbólicas:

```
function v=aproxima_derivada2(funci,a)
syms x;
i=1;
h=1;
v(i)=(subs(funci,a+h)+subs(funci,a-h)-(2*subs(funci,a)))/(h^2);
error=1;
while(error>1e-10)
    h=h/2;
    i=i+1;
    v(i)=(subs(funci,a+h)+subs(funci,a-h)-(2*subs(funci,a)))/(h^2);
    error= abs(v(i)-v(i-1));
end
```

Para ejecutarlo bastaría con hacer una llamada de tipo

```
matlab:1> aproxima_derivada2(sin(x),1)
ans =
Columns 1 through 16:
-0.7736 -0.8241 -0.8371 ... -0.8415 -0.8415
```

Este código presenta varias incompatibilidades con la *toolbox* de Cálculo Simbólico de OCTAVE, para la que la función `subs` necesita conocer la variable simbólica (así que `x` pasará como parámetro en la función) y que no admite vectores simbólicos como `v` (hay que pasarlo a coma flotante mediante `to_double`). Un código compatible con OCTAVE sería

```
function v=aproxima_derivada2(funci,x,a) (con variable simbólica)
i=1;
h=1;
v(i)=to_double((subs(funci,x,a+h)+subs(funci,x,a-h)
    -(2*subs(funci,x,a)))/(h^2)); (no se admiten vectores simbólicos)
error=1;
while(error>1e-10)
    h=h/2;
    i=i+1;
    v(i)=to_double((subs(funci,x,a+h)+subs(funci,x,a-h)
    -(2*subs(funci,x,a)))/(h^2)); (no se admiten vectores simbólicos)
    error= abs(v(i)-v(i-1));
end
```

y se ejecutaría en OCTAVE como

```

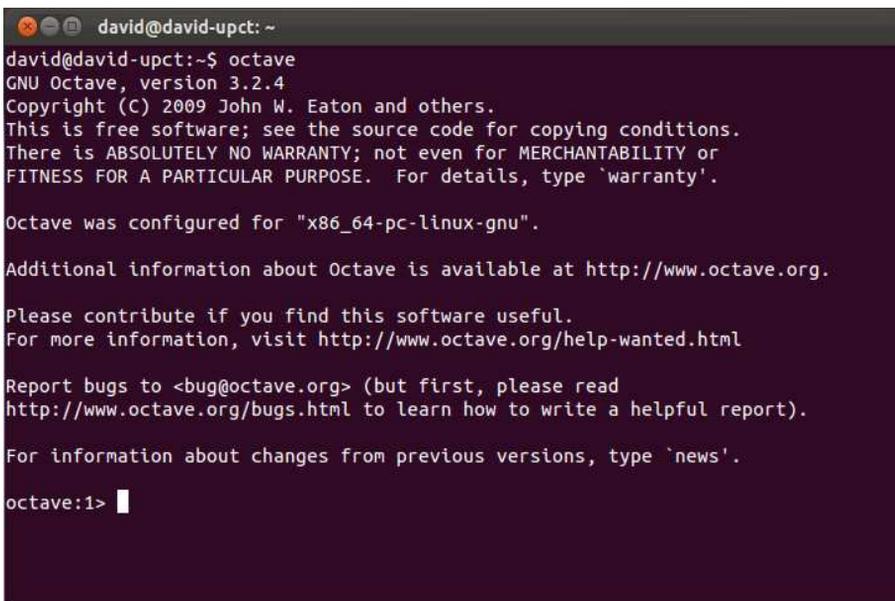
octave:1> symbols (para cargar la toolbox)
octave:2> x=sym('x') (la variable simbólica irá como parámetro)
octave:3> aproxima_derivada2(Sin(x),x,1) (nótese la mayúscula)
ans =
Columns 1 through 20:
   -0.77364  -0.82409  -0.83710  ...  -0.84147  -0.84147

```

Por desgracia este nuevo código de OCTAVE tampoco es compatible con MATLAB, porque la función `to_double` de OCTAVE se llama simplemente `double` en MATLAB. Estas pequeñas incompatibilidades, unidas a la pobreza del Cálculo Simbólico de OCTAVE (y al de MATLAB, tampoco nos vamos a engañar) me ha hecho decantarme por MAXIMA en lugar de OCTAVE siempre que necesito realizar una manipulación simbólica. Para lo demás OCTAVE me ha funcionado de maravilla.

2.3 El entorno gráfico `qtoctave`

OCTAVE no posee un entorno gráfico por defecto. Esto quiere decir que la pantalla será una simple consola (o ventana de MS-DOS en la versión de windows) sin ningún añadido.



```

david@david-upct: ~
david@david-upct:~$ octave
GNU Octave, version 3.2.4
Copyright (C) 2009 John W. Eaton and others.
This is free software; see the source code for copying conditions.
There is ABSOLUTELY NO WARRANTY; not even for MERCHANTABILITY or
FITNESS FOR A PARTICULAR PURPOSE. For details, type `warranty'.

Octave was configured for "x86_64-pc-linux-gnu".

Additional information about Octave is available at http://www.octave.org.

Please contribute if you find this software useful.
For more information, visit http://www.octave.org/help-wanted.html

Report bugs to <bug@octave.org> (but first, please read
http://www.octave.org/bugs.html to learn how to write a helpful report).

For information about changes from previous versions, type `news'.

octave:1> █

```

Figura 1: OCTAVE en una consola.

Es perfectamente posible trabajar desde una consola; por ejemplo, usando el comando `cd` para cambiar el directorio de trabajo, o llamando a la ayuda de

una función mediante `help` o `doc`. Sin embargo algunos usuarios pueden sentirse *desnudos* sin un entorno gráfico. Para ellos es ideal QTOCTAVE.

QTOCTAVE es un entorno gráfico (*front-end*) desarrollado por Pedro Luis Lucas Rosado que hay que instalar por separado (desde repositorio en linux, descargándolo de [18] el resto). QTOCTAVE no intenta clonar el aspecto de MATLAB, pero hace una apuesta similar: el programa funciona como una especie de “ecosistema” en sí mismo en el que se pueden ir encajando las ventanas que uno desee: terminal de OCTAVE, navegador y editor de archivos, listas de comandos y variables usadas, ayuda fija o dinámica, portapapeles, ... En el menú se incluyen las opciones más habituales para operar matrices, calcular integrales, dibujar gráficas, etc. Además lleva integrada la opción de buscar e instalar *toolboxes*.

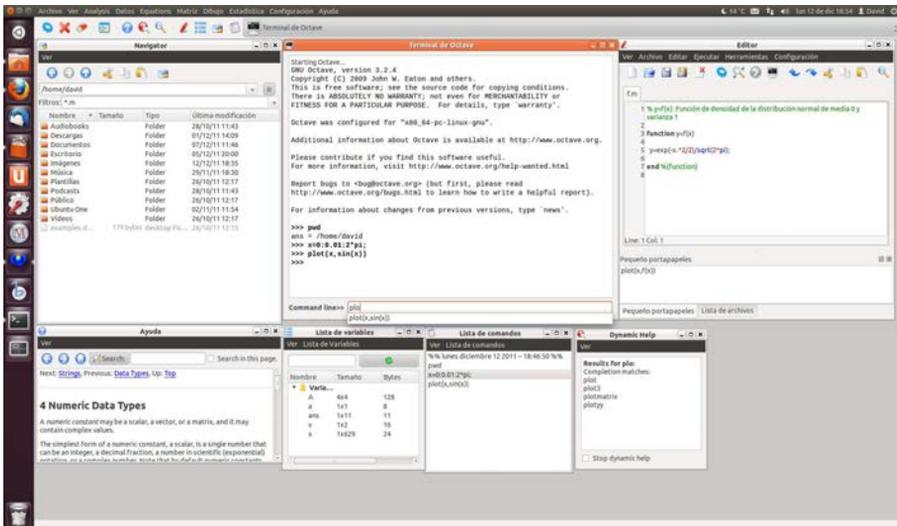


Figura 2: Una de múltiples posibilidades de configuración de QTOCTAVE

2.4 octave en tu bolsillo

Durante más de una década usé software privativo en mi docencia, y nunca recibí ningún tipo de respuesta por parte de mi alumnado. En cambio tras un par de años usando software libre están siendo mis estudiantes los que me están empezando a enseñar cosas a mí, como por ejemplo a manejar el zoom de las gráficas de OCTAVE. Fue precisamente un alumno quien me informó de que había encontrado una versión portátil de OCTAVE que era una pequeña joya. No es más que un ejemplo de cómo el software libre es, por encima de todo, un concepto cooperativo. Gracias a Nazareth por el aviso.

El proceso es bastante simple, se descarga esta versión especial de QTOCTAVE desde [19] y se instala en un dispositivo externo, por ejemplo un *pendrive*. Ahora podremos ejecutar QTOCTAVE conectando ese *pendrive* a cualquier equipo que tenga windows, sin que sea necesario instalar nada en el ordenador. Esta es una opción muy útil para trabajar en un PC que no sea nuestro, o si el ordenador está congelado como suele ocurrir con los de las aulas de informática (aunque OCTAVE ya se encuentra instalado en las aulas de informática de la ETSIT).

2.5 Alternativas

En general OCTAVE es la alternativa libre a MATLAB más usada, aunque en Francia es muy popular SCILAB [20], desarrollada por el Instituto Nacional de Investigación en Informática y Automática francés. SCILAB también es multiplataforma e incluye un pequeño entorno gráfico, editor, historial de comandos, lista de variables, un sistema de *toolboxes*,... Aunque no es un clon como OCTAVE, la nomenclatura de SCILAB también es similar a la de MATLAB, y además incluye un conversor de archivos para facilitar la migración de nuestros programas de MATLAB a SCILAB.

Existen otras opciones libres como FREEMAT [21], pero como no las he probado no puedo opinar sobre ellas.

Referencias

- [1] http://es.wikipedia.org/wiki/Archivo:Mapa_conceptual_del_software_libre.svg
- [2] <http://www.gnu.org/philosophy/category.es.jpg>
- [3] <http://upload.wikimedia.org/wikipedia/commons/7/79/MapaConceptualFLOSS.png>
- [4] <http://es.creativecommons.org/licencia>
- [5] <http://www.gnu.org/philosophy/free-sw.es.html>
- [6] http://es.wikipedia.org/wiki/Código_abierto
- [7] <http://www.gnu.org/education/edu-schools.es.html>
- [8] <http://www.latex-project.org>
- [9] <http://maxima.sourceforge.net>
- [10] http://en.wikipedia.org/wiki/List_of_free_and_open_source_software_packages
- [11] <http://www.freealts.com>
- [12] freesoftware-list@upct.es
- [13] <http://www.gnu.org/software/octave>

- [14] <http://octave.sourceforge.net>
- [15] <http://es.wikipedia.org/wiki/Matlab>
- [16] <http://en.wikipedia.org/wiki/Matlab>
- [17] [http://en.wikibooks.org/wiki/MATLAB_Programming/
Differences_between_Octave_and_MATLAB](http://en.wikibooks.org/wiki/MATLAB_Programming/Differences_between_Octave_and_MATLAB)
- [18] <http://qtoctave.wordpress.com>
- [19] <http://portableapps.com/node/23235>
- [20] <http://www.scilab.org>
- [21] <http://freemat.sourceforge.net>