

Unidad de Disseminación de Software Libre (<http://nux.ula.ve>)
Corporación Parque Tecnológico de Mérida
Universidad de Los Andes

Introducción a GNU Octave

Versión 1.0 – Agosto 2007

David A. Hernández Aponte

Copyright (c) 2007 David A. Hernández Aponte.

Se concede permiso de copiar, distribuir y/o modificar este documento bajo los términos de la Licencia de Documentación Libre GNU, GDFL, Versión 1.2 o cualquier otra versión posterior publicada por la Free Software Foundation; sin partes no modificables y sin añadidos en la portada o contraportada. Una copia de esta licencia se incluye en la sección titulada "GNU Free Documentation License".

Copyright (c) 2007 David A. Hernández Aponte.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.2 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts. A copy of the license is included in the section entitled "GNU Free Documentation License".

Contenido

¿Qué es Octave?.....	4
Iniciar y salir de Octave.....	4
La ayuda en Octave.....	4
Cálculos básicos.....	5
Historial.....	7
Variables.....	7
Vectores y matrices.....	9
Secuencias.....	10
Funciones matemáticas.....	10
Operadores lógicos.....	13
Operadores booleanos.....	14
Sistema de ecuaciones.....	14
Funciones definidas por el usuario.....	15
Gráficos 2D en Octave.....	17
Gráficos 3D en Octave.....	19
Modificar gráficas.....	22
Recursos Web para Octave.....	23
GNU Free Documentation License.....	24

¿Qué es Octave?

GNU Octave es un lenguaje de alto nivel destinado para el cálculo numérico. Provee una interfaz sencilla, orientada a la línea de comandos (consola), que permite la resolución de problemas numéricos, lineales y no lineales, además permite la ejecución de *scripts* y puede ser usado como lenguaje orientado al procesamiento por lotes.

Octave nació alrededor del año 1988, y fue concebido originalmente para ser usado en un curso de diseño de reactores químicos para los alumnos de Ingeniería Química de la Universidad de Texas y la Universidad de Wisconsin-Madison.

Octave posee una gran cantidad de herramientas que permiten resolver problemas de algebra lineal, cálculo de raíces de ecuaciones no lineales, integración de funciones ordinarias, manipulación de polinomios, integración de ecuaciones diferenciales ordinarias y ecuaciones diferenciales algebraicas. Sus funciones también se pueden extender mediante funciones definidas por el usuario escritas en el lenguaje propio de Octave o usando módulos dinámicamente cargados escritos en lenguajes como C, C++ y Fortran entre otros.

Iniciar y salir de Octave

Para ejecutar el programa escribimos en la consola el comando **octave** y presionamos la tecla [Enter]. A continuación veremos en nuestras pantallas lo siguiente:

```
$ octave
GNU Octave, version 2.9.9 (i486-pc-linux-gnu).
Copyright (C) 2006 John W. Eaton.
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'.

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).

octave:1>
```

Para salir de Octave podemos usar el comando **quit** ó **exit** indistintamente:

```
octave:4> exit
```

La ayuda en Octave

El comando **help** nos muestra una lista de todos los operadores y funciones disponibles en Octave. También podemos invocar al comando **help** para que nos muestre una breve

descripción de estos operadores y funciones. Para ello basta con escribir help seguido del nombre de la función u operador: `help nombre_funcion`

Por ejemplo, queremos que Octave nos muestre la ayuda relacionada con el comando **doc**:

```
octave:23> help doc
-- Command: doc FUNCTION_NAME
   Displays documentation for the function FUNCTION_NAME.  For
   example, if you want to see the documentation for the Octave
   random number generator `rand', type
   `doc rand'
```

See also: help.

doc is the built-in command from the file
/usr/share/octave/2.9.9/m/miscellaneous/doc.m

Additional help for built-in functions and operators is available in the on-line version of the manual. Use the command ``doc <topic>'` to search the manual index.

Help and information about Octave is also available on the WWW at <http://www.octave.org> and via the help@octave.org mailing list.

Tal como dice la ayuda, `doc nombre_funcion` es otro comando de ayuda que busca el nombre de la función en el índice de la versión en línea del manual de Octave.

Cálculos básicos

Vamos a empezar a trabajar con la tarea más básica que podemos darle a esta herramienta, vamos a utilizarla como una calculadora.

A continuación, un ejemplo sencillo de suma, multiplicación y división:

```
octave:1> 45 + 76 * 12.5 / 1.5
ans = 678.33
```

De igual manera también podemos calcular promedios:

```
octave:2> (12+15+20)/3
ans = 15.667
```

Para calcular potencias utilizamos el símbolo “^” o el doble asterisco “**”:

```
octave:3> 2^8
ans = 256
```

Octave, como toda calculadora cumple con cierta regla de precedencia para el uso de operadores, esta es: las expresiones se evaluarán de izquierda a derecha, la potencia tendrá el mayor orden de precedencia, seguido de la multiplicación y división, y con la suma y resta como los operadores con menor precedencia.

No es lo mismo “ $4 + 5 / 4 - 3$ ” que “ $(4 + 5) / (4 - 3)$ ”. En la primera expresión se evaluará primero la división entre 5 y 4 y luego se le sumará (restará) el 4 y el 3, en la segunda expresión primero se evaluarán las operaciones dentro de los paréntesis y luego se dividirá el resultado de estos valores.

La siguiente tabla muestra los operadores aritméticos usados en Octave.

Tabla 1: Operadores aritméticos

$x + y$	Suma. Si ambos operandos son matrices el número de columnas debe coincidir con el número de filas. Si uno de los elementos es escalar, su valor es sumado a todos los elementos del otro operando.
$x .+ y$	Suma elemento por elemento. Este operador es equivalente a usar +.
$x - y$	Resta. Si ambos operandos son matrices el número de columnas debe coincidir con el número de filas.
$x .- y$	Resta elemento por elemento. Este operador es equivalente a usar -.
$x * y$	Multiplicación de matrices. El número de columnas debe coincidir con el número de filas.
$x .* y$	Multiplicación de matrices elemento por elemento. Si ambos operandos son matrices el número de columnas debe coincidir con el número de filas.
x / y	División por la derecha.
$x ./ y$	División por la derecha elemento por elemento.
$x \ y$	División por la izquierda.
$x .\ y$	División por la izquierda elemento por elemento.
x^y ó $x**y$	Potenciación. Si tanto x como y son escalares, este operador calculará a x elevado a la y potencia. Si ambos operandos son matrices, el resultado devolverá un error, si uno de los dos es una matriz, este deberá ser una matriz cuadrada.
$x.^y$ ó $x.**y$	Potenciación elemento por elemento. Si ambos operandos son matrices el número de columnas debe coincidir con el número de filas.
$-x$	Negación.
$+x$	Suma unaria. Este operador no afecta al operando.
x'	Conjugada compleja transpuesta. Para argumentos reales este operador es equivalente al operador de la transpuesta.
$x.'$	Transpuesta.
$++x$	Pre-incremento. Equivalente a ejecutar la expresión $x = x + 1$ antes de acceder al valor de x
$x++$	Post-incremento. Equivalente a ejecutar la expresión $x = x + 1$ después de acceder al valor de x
$--x$	Pre-decremento. Equivalente a ejecutar la expresión $x = x - 1$ antes de acceder al valor de x

<code>x--</code>	Post-decremento. Equivalente a ejecutar la expresión <code>x = x - 1</code> después de acceder al valor de <code>x</code>
------------------	---------------------------------------------------------------------------------------------------------------------------

Historial

Al igual que en los entornos **nix*, Octave almacena los comandos ejecutados previamente por el usuario, incluso órdenes ejecutadas en sesiones anteriores, el archivo donde se guarda el historial se encuentra en el directorio 'home' del usuario y lleva el nombre de `.octave_hist`. Gracias a esta característica podemos buscar un comando ejecutando previamente haciendo uso de las teclas de dirección arriba y abajo de nuestro teclado. Una vez encontrado el comando que queremos usar podemos editarlo desplazándonos con las teclas de dirección izquierda y derecha de nuestro teclado ó con las teclas inicio y fin y borrar los caracteres con la tecla suprimir o retroceso. Para que se nos muestre una lista de todos los comandos ejecutados hasta el momento debemos usar el comando **history**.

Al salir de Octave se perderán todas las variables que se han creado, para guardar una sesión y recuperarla mas tarde solo debe utilizar los comandos **save** y **load** respectivamente seguidos de un nombre asignado por el usuario.

```
octave:2> save nombre_sesion
```

Esto almacenará la sesión en un archivo de nombre `nombre_sesion` dentro del directorio de trabajo. Para recuperar la sesión escribiremos lo siguiente en cualquier momento dentro del entorno Octave.

```
octave:1> load nombre_sesion
```

También podremos recuperar variables específicas usando el siguiente formato

```
load nombre_sesion var1 var2 ...
```

Variables

Las variables son identificadores que permiten almacenar datos, los cuales pueden cambiar durante la ejecución de un programa. Las variables nos permiten asignarle nombres a los valores para luego poder hacer referencia a estos. No hay límite para la longitud en el nombre de una variable, pero estos deben estar constituidos por una secuencia de letras, *underscores* (símbolo de subrayado o guión bajo) o números y sólo pueden empezar con letras o *underscores*.

Algunos ejemplos de nombres válidos son:

```
a
z_
asdfqwertytr
a23_____werewtyrfvbc
_a1
```

```
a_1  
a1_  
A1_
```

Los nombres de las variables son sensibles a mayúsculas, `a1_` y `A1_` representan a dos variables completamente diferentes.

Para crear las variables sólo debemos escribir el nombre que le daremos y asignarle un valor, el operador de asignación es el símbolo '=' el cual se coloca a la derecha de la variable y antes del valor que se desea asignar. Luego de creada, la variable podrá ser usada nuevamente escribiendo su nombre.

Por ejemplo, calculemos el área (**A**) de un trapecio con base mayor (**B**) de 10 cm , base menor (**b**) de 6 cm y la altura (**h**) de 4 cm.

```
octave:1> B=10  
B = 10  
octave:2> b=6  
b = 6  
octave:3> h=4  
h = 4  
octave:4> A=(h*(B+b))/2  
A = 32
```

También podemos introducir todos los comandos en una misma línea, para ello separamos cada comando con una coma ',', el ejemplo anterior queda así:

```
octave:5> B=10, b=6, h=4, A=(h*(B+b))/2  
B = 10  
b = 6  
h = 4  
A = 32
```

Si deseamos evitar que el resultado de alguna operación se muestre en pantalla debemos agregar un punto y coma ';' al final de la orden que deseamos ocultar. En el ejemplo anterior, si sólo queremos ver el resultado de **A**, sustituimos la coma ',' al final de cada variable por un punto y coma ';' :

```
octave:5> B=10; b=6; h=4; A=(h*(B+b))/2  
A = 32
```

Si queremos borrar una variable usaremos el comando `clear` seguido del nombre de la variable:

```
octave:5> clear A
```

Ahora veremos que si hacemos la llamada a la variable **A**, que tenía el valor 32, nos mostrará un mensaje de error indicándonos que la variable **A** no está definida.

```
octave:6> A  
error: `A' undefined near line 6 column 1
```


Existen un conjunto de variables predefinidas utilizadas por el programa que no conviene cambiar, entre ellas podemos mencionar:

```
ans muestra el último resultado
pi 3.1416
e 2.7183
i y j 0 + 1i
Inf Infinito
```

Vectores y matrices

En el área de la computación un vector generalmente es definido como un arreglo, es decir, un conjunto de datos a los cuales se accede por medio de índices. Un vector es la forma mas simple de una matriz, podemos decir que es una matriz de una dimensión.

En Octave disponemos de una gran variedad de formas para definir vectores y matrices, usualmente lo hacemos encerrando los elementos dentro de corchetes, los elementos separados por espacios o comas (,) definen un vector fila, los elementos separados por el retorno de carro o por punto y coma (;) definen una nuevo vector fila en la matriz.

```
octave:1> a=[1,3,5]
a =
    1    3    5

octave:2> b=[5 7 9]
b =
    5    7    9

octave:3> c=[1;2;3]
c =
    1
    2
    3

octave:4> d=[2
> 1;0]
d =
    2
    1
    0

octave:5> e=[a ; b ; 2, 4 6]
e =
    1    3    5
    5    7    9
    2    4    6
```

Secuencias

Una forma sencilla de producir una secuencia de números es utilizando la notación **n:m** donde **n** es el número inicial y **m** el final

```
octave:2> 1:10
ans =
    1    2    3    4    5    6    7    8    9   10
```

También podemos usar la notación **p:q:r** para crear una secuencia que inicia en **p**, finaliza en **r** con intervalos de **q**. En el siguiente caso almacenaremos en una variable **b** una secuencia partiendo de cero y finalizando en diez, con un intervalo de dos entre cada número.

```
octave:3> b=0:2:10
b =
    0    2    4    6    8   10
```

Tenemos otras dos funciones que nos permiten crear vectores de elementos secuenciales, estos son **linspace** y **logspace**, el primero separa los números uniformemente y el segundo lo hace logarítmicamente.

```
octave:39> linspace(1,4,6)
ans =
    1.0000    1.6000    2.2000    2.8000    3.4000    4.0000
```

Funciones matemáticas

Octave incluye una serie de funciones matemáticas y trigonométricas que nos ayudan a simplificar algunos cálculos, la siguiente tabla muestra algunas de ellas.

Tabla 2: Funciones matemáticas

Función	Descripción
sqrt(x)	Raíz cuadrada de x
abs(x)	Valor absoluto de x
log(x)	Logaritmo neperiano de x
log2(x)	Logaritmo en base a 2 de x
log10(x)	Logaritmo en base a 10 de x
exp(x)	e^x
pow2(x)	Para cada elemento de x , calcula 2^x
rem(x,y)	Resto entre la división de x e y (módulo)
round(x)	Redondeo de x al entero mas cercano

ceil(x)	Redondeo al entero superior de x
floor(x)	Redondeo al entero inferior de x
fix(x)	Redondeo hacia el entero mas cercano a cero
sin(x) cos(x) tan(x) sec(x) csc(x) ctg(x)	Funciones trigonométricas ordinarias para x
asin(x) acos(x) atan(x) asec(x) acsc(x) actg(x)	Funciones trigonométricas inversas para x
sinh(x) cosh(x) tanh(x) sech(x) csch(x) ctgh(x)	Funciones trigonométricas hiperbólicas para x
asinh(x) acosh(x) atanh(x) asech(x) acsch(x) actgh(x)	Funciones trigonométricas hiperbólicas inversas para x
arg(x) angle(x)	Regresa los argumentos de los números complejos de x
conj(x)	Regresa el complejo conjugado de x
real(x)	Regresa la parte real de x
imag(x)	Regresa la parte imaginaria de x
gcd(x,y)	Calcula el máximo común divisor entre x e y
lcm(x,y)	Calcula el mínimo común múltiplo entre x e y
sign(x)	Devuelve 1 para los elementos positivos y 0 para los negativos
x(n)	regresa el n-ésimo elemento del vector x. (recorrido por columnas)
x(n,m)	regresa el elemento que se encuentra en la fila n y columna m.
x(n:m)	regresa el conjunto de elementos partiendo desde el n-ésimo elemento hasta el m-ésimo elemento. (recorrido por columnas)
x(n,:)	devuelve todos los elementos de la fila n

x(:,m)	devuelve todos los elementos de la columna m
rows(x)	devuelve el número de filas del vector x
columns(x)	devuelve el número de columnas del vector x
length(x)	devuelve la longitud de la matriz (columnas)
size(x)	devuelve el numero de filas y el número de columnas
rand(n,m)	Crea una matriz aleatoria uniformemente distribuida de $n \times m$, si se invoca con un solo escalar crea una matriz aleatoria cuadrada de n elementos
rand*(n,m)	Crea una matriz aleatoria $n \times m$ cuya distribución dependerá del valor *, * puede ser n, e, g ó p, para una distribución normal, exponencial gamma o poisson respectivamente. Si se invoca con un solo escalar crea una matriz aleatoria cuadrada de n elementos
eye(n,m)	Crea una matriz $n \times m$ con diagonal 1, si se invoca con un solo escalar crea una matriz identidad cuadrada de n elementos
diag(x,k)	Si x es un vector, la función crea una matriz cuya diagonal serán los elementos del vector x la cual empezará la k-ésima columna de la matriz creada. Si x es una matriz creará un vector columna con la diagonal que empieza en la k-ésima columna de la matriz x. Si k es positivo devuelve la super-diagonal, si es negativo devuelve la sub-diagonal. Si se invoca con un solo escalar el valor k por defecto es 0 y devuelve la super-diagonal principal.
zeros(n,m)	Crea una matriz $n \times m$ con sus elementos iguales a 0, si se invoca con un solo escalar crea una matriz cuadrada de n elementos
ones(n,m)	Crea una matriz $n \times m$ con sus elementos iguales a 1, si se invoca con un solo escalar crea una matriz cuadrada de n elementos
linspace(p,q,n)	Crea una matriz de n elementos espaciados uniformemente desde p hasta q
logspace(p,q,n)	Crea una matriz de n elementos espaciados logarítmicamente desde p hasta q
rank(x)	Devuelve el rango de la matriz x
inv(x)	Calcula la inversa de la matriz cuadrada x
det(x)	Calcula el determinante de la matriz cuadrada x
trace(x)	Suma los elementos de la diagonal principal de la matriz x (traza)
sum(x)	Devuelve la suma de los elementos del vector x o la suma de los elementos de cada columna de la matriz x
prod(x)	Devuelve el producto de los elementos del vector x ó el producto de los elementos de cada columna de la matriz x
sumsq(x)	Devuelve la suma del cuadrado de los elementos del vector x ó la suma del cuadrado de los elementos de cada columna de la matriz x
cumsum(x)	Devuelve la suma acumulada de los elementos del vector x ó la suma acumulada de los elementos de cada columna de la matriz x
cumprod(x)	Devuelve el producto acumulado de los elementos del vector x ó el producto acumulado de los elementos de cada columna de la matriz x
max(x)	Devuelve el máximo para cada columna de la matriz x o el máximo de los elementos del vector x
min(x)	Devuelve el mínimo elemento del vector x o el mínimo elemento para cada columna de la matriz x
sort(x)	Ordena de menor a mayor los elementos del vector x ó los elementos de cada

	columna de la matriz x
median(x)	Calcula la mediana del vector x ó la mediana para cada columna de la matriz x
mean(x)	Calcula la media del vector x ó la media para cada columna de la matriz x
std(x)	Calcula la desviación estándar del vector x ó la desviación estándar para cada columna de la matriz x
cov(x,y)	Calcula la covarianza entre las variables x e y
corrcoef(x,y)	Calcula el coeficiente de correlación entre las variables x e y
expm(x)	Calcula la exponencial de una matriz cuadrada, se calcula por desarrollo de la serie de Taylor.
logm(x)	Calcula el logaritmo de una matriz cuadrada.
sqrtn(x)	Calcula la raíz cuadrada de una matriz cuadrada.
reshape(x,n,m)	Reordena el vector o matriz x para ajustarse a las dimensiones n×m
fliplr(x)	Devuelve x con las columnas en orden inverso (derecha-izquierda)
flipud(x)	Devuelve x con las filas en orden inverso (arriba-abajo)
rot90(x,n)	Devuelve la matriz x rotada n veces en sentido antihorario. Si n se omite se asume n=1
vec(x)	Devuelve un vector columna formado por las columnas de la matriz x.

Las funciones pueden contener otras funciones como argumentos:

```
octave:7> sqrt( round( 25/3 ) + 1 )
ans = 3
```

En Octave, se asume que los argumentos de funciones trigonométricas están en radianes por lo que si queremos calcular los grados de alguna función trigonométrica debemos aplicar la siguiente formula:

```
octave:8> sin(90*pi/180)
ans = 1
```

```
octave:9> sin(90)
ans = 0.89400
```

El primer ejemplo se refiere al cálculo para el seno de 90° mientras que el segundo se refiere al cálculo para el seno de 90 en radianes.

Operadores lógicos

A continuación presentamos una tabla con los operadores lógicos elementales:

Tabla 3: Operadores lógicos

$x < y$	verdadero si x es menor que y
$x > y$	verdadero si x es mayor que y
$x == y$	verdadero si x es igual que y
$x <= y$	verdadero si x es menor o igual que y
$x >= y$	verdadero si x es mayor o igual que y
$x != y$ o $x \neq y$	verdadero si x es diferente que y

Al efectuar operaciones lógicas el resultado será **1** para verdadero y **0** para falso.

Operadores booleanos

En las operaciones booleanas se comparan cada uno de los elementos de un operando con su correspondiente elemento en el otro operando y se considera falso cuando el cálculo entre estos elementos da cero, y verdadero para cualquier otro caso.

La siguiente tabla muestra los operadores booleanos elementales:

Tabla 4: Operadores booleanos

$x \& y$	y lógico (and). Cada elemento del resultado es verdadero si ambos elementos correspondientes en los operandos son verdaderos.
$x y$	o lógico (or). Cada elemento del resultado es verdadero si alguno de los dos elementos correspondientes en los operandos es verdadero.
$!x$ ó $\sim x$	negación (not). Cada elemento del resultado es verdadero si el elemento correspondiente en el operando es falso.

Sistema de ecuaciones

Para la resolución de sistemas de ecuaciones del tipo $Ax = b$ utilizamos la notación $a \backslash b$, por ejemplo, para calcular el siguiente sistema de ecuaciones de primer grado con dos incógnitas,

$$\begin{aligned} 6 \cdot x - 7 \cdot y &= 5 \\ 8 \cdot x - 9 \cdot y &= 7 \end{aligned}$$

haremos lo siguiente: guardaremos los elementos x e y en una matriz a y la igualdad en un vector b , para finalmente ejecutar el comando $a \backslash b$.

```
octave:4> a=[6, -7;8,-9]
a =
```

```
6  -7
8  -9
```

```
octave:5> b=[5;7]
b =

  5
  7
```

```
octave:6> a\b
ans =

  2
  1
```

Funciones definidas por el usuario

En Octave podemos crear nuestras propias funciones, podemos escribirlas directamente desde la línea de comandos o ejecutarlas desde un archivo externo.

Las funciones que creamos desde la línea de comandos deben cumplir con el siguiente formato:

```
function variable_salida = nombre_funcion (argumentos_entrada)
    cuerpo _funcion
endfunction
```

En caso de devolver varias variables, éstas deben estar encerradas entre corchetes “[]”:

```
function [salida1,salida2] = nombre_funcion (argumentos_entrada)
    cuerpo _funcion
endfunction
```

Por ejemplo, crearemos una función que calcula el seno(x) en grados:

```
octave:1> function s = sind(x)
> %SIND(X) Calcula seno(x) en grados
> s = sin(x*pi/180);
> endfunction
```

Y ejecutamos,

```
octave:2> sind(45)
ans = 0.70711

octave:3> sind(90)
ans = 1
```

Para que Octave ejecute un archivo éste debe tener extensión `.m` y debe encontrarse en el directorio desde donde estemos ejecutando Octave. También podemos agregar rutas adicionales con el comando `addpath('ruta')`, donde `ruta` es el camino al directorio que contiene los scripts. Para borrar la ruta usamos `rmpath('ruta')`.

Ejemplo: Suma de los elementos que integran una matriz, **sumaelementos.m**

```
M=[round(rand(3,3)*100)] % creamos una matriz aleatoria de 3x3
[f,c]=size(M); % f #filas; c #columnas
s=0; % inicializamos s=0 (suma inicial)
for i = 1:f % recorre filas
    for j=1:c % recorre columnas (en cada fila)
        s=s+M(i,j); % suma cada elemento a s
    end
end
s % muestra la suma final
```

Para ejecutarlo escribiremos el nombre del archivo en la línea de comandos, deberá aparecer lo siguiente,

```
octave:1> sumaelementos
M =

    96    77    77
    36    29    58
    19    63    71

s = 526
```

Para crear programas donde el usuario debe introducir datos utilizamos el comando **input** dentro del archivo.

Ejemplo: Función que convierte kilos en libras, **kilo2libra.m**

```
%*****
% kilo2libra.m
% Programa que permite convertir un valor en kilos a libras.
%*****
disp('\nPrograma de conversion de kilos a libras\n');
kilo = input('Introduzca el peso en kilogramos: ');
libra=kilo*2.20462262185;
fprintf('\n%g kilogramos son %f libras.\n', kilo, libra);
%*****
```

Haremos la conversión de 40 kilogramos en libras,

```
octave:2> kilo2libra

Programa de conversion de kilos a libras

Introduzca el peso en kilogramos: 40

40 kilos son 88.184905 libras.
```


Gráficos 2D en Octave

La forma mas simple para producir gráficos bidimensionales en Octave se hace a través del comando `plot(x[,y][,fmt])`, donde `x` y opcionalmente `y` representan los vectores coordenadas para cada punto, adicionalmente se le puede especificar una serie de formatos que cambian la apariencia de salida de la gráfica, esto incluye el estilo de línea, el color y otras características que explicaremos en los siguientes ejemplos.

En este primer ejemplo crearemos un vector aleatorio de 50 elementos que luego graficaremos.

```
octave:1> x = (rand(50,1));
octave:2> plot(x)
```

Acto seguido se abre una ventana mostrándonos la gráfica resultante:

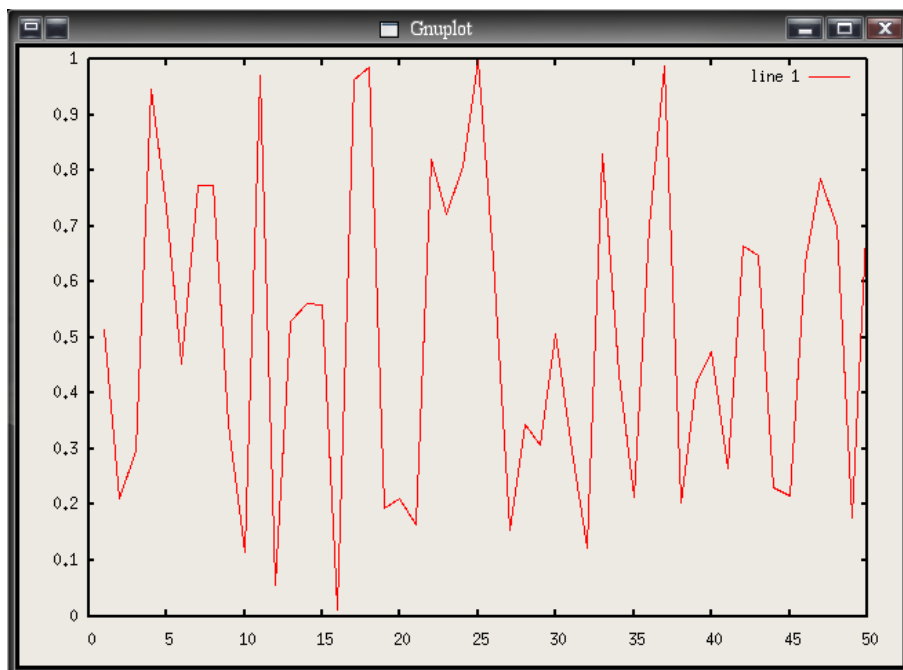


Imagen 1: Gráfica de los elementos de un vector con valores por defecto

Por defecto Octave une los puntos mediante segmentos de línea, usando los mismos datos le diremos a Octave que no nos una los puntos, y que estos los presente con un mayor tamaño, para ello hacemos lo siguiente:

```
octave:3> plot(x,'@')
```

Para obtener lo mostrado en la siguiente imagen:

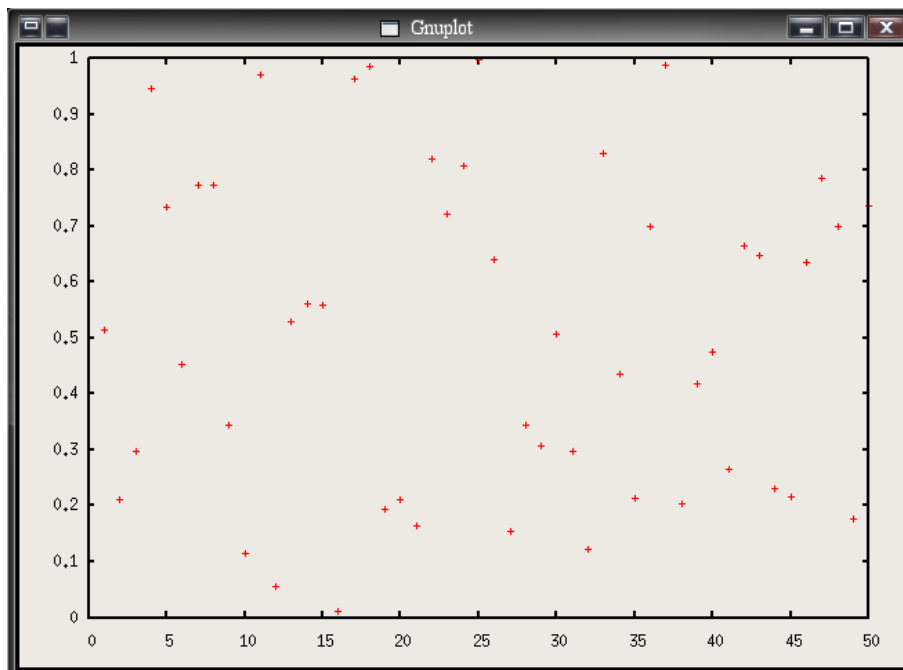


Imagen 2: Gráfica de los elementos de un vector con valores modificados

A continuación presentamos los diferentes argumentos que podemos usar para modificar el formato de la gráfica, se puede usar tanto comillas simples (') como dobles (").

- '-' une los datos mediante segmentos de línea (opción por defecto).
- '.' asigna un punto a cada dato
- '@' asigna un punto grande para cada dato
- '-@' une los puntos (grandes) de cada dato con segmentos de línea.
- '^' gráfica estilo impulso (línea que va desde cero hasta el punto)
- 'L' gráfica tipo escalera
- '#' diagrama de cajas
- 'n' asigna un color a los datos, donde **n** es un dígito del 0 al 9 que representa el color.
- 'nm' asigna un color y un estilo de punto , donde **n** y **m** son dígitos del 0 al 9, **n** representa un color y **m** el estilo del punto.

El argumento *fmt* puede también ser usado para asignarle una etiqueta a los datos, para ello colocaremos la etiqueta inmediatamente después del argumento, hay que asegurarse de colocar el texto entre punto y coma (;) antes de cerrar con la comilla.

Por ejemplo, vamos a graficar un vector *t*, su seno y su coseno usando diferentes formatos para su presentación.

```
octave:13> t=0:0.1:4.5;
octave:14> plot (t,';t;', cos(t), '38;cos(t);', sin(t), '06;sin(t);')
```

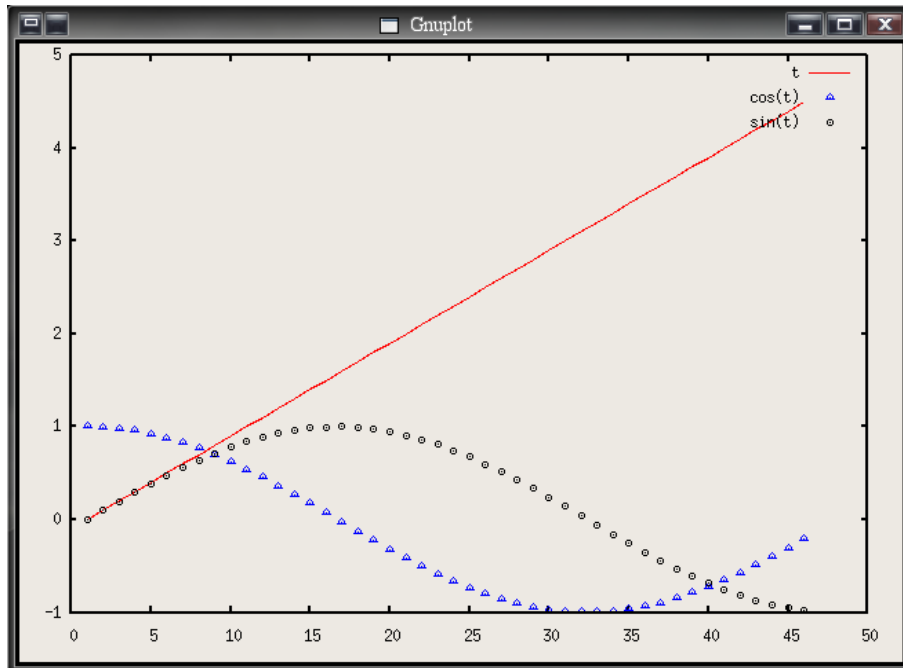


Imagen 3: Gráfica con diferentes parámetros modificados

Otras funciones para graficar en dos dimensiones son:

semilogx(x[,y][,fmt]) realiza la gráfica utilizando una escala logarítmica en el eje x.

semilogy(x[,y][,fmt]) realiza la gráfica utilizando una escala logarítmica en el eje y.

loglog(x[,y][,fmt]) utiliza una escala logarítmica en ambos ejes.

polar(t,r[,fmt]) produce una gráfica con coordenadas polares theta y ro.

bar(y) produce un diagrama de barras donde **y** es un vector. Representa los valores de **y** frente a sus índices.

bar(x,y) donde **x** e **y** son vectores de igual dimensión, esta función produce un diagrama de barras de **y** frente a **x**.

stairs(y) produce un diagrama tipo escalera donde **y** es un vector. Representa los valores de **y** frente a sus índices.

stairs(x,y) donde **x** e **y** son vectores de igual dimensión, esta función produce un diagrama tipo escalera.

hist(x,y) crea un histograma con los valores proporcionados.

Gráficos 3D en Octave

Para producir gráficos en 3D disponemos de varias opciones, la mas simple es usar el comando `plot3(x,y,z)` donde cada argumento es tomado para convertirse en los vértices del gráfico tridimensional.

Si todos los argumentos son vectores de la misma longitud se dibujará una única línea continua. En caso de todos los argumento sean matrices cada una de las columnas de las matrices serán tratada como líneas separadas.

En caso de que sólo se le pasen dos argumentos en lugar de tres, `plot3(x,c)`, el segundo argumento 'c' debe ser un número complejo, así, las partes reales e imaginarias de éste son usadas como las coordenadas y e z respectivamente.

Si sólo se le pasa un argumento, `plot3(c)`, las partes reales e imaginarias de los argumentos son usados como los valores y y z, y se trazan frente su índice.

El comando `plot3` también acepta los argumentos que permiten modificar el formato de presentación de la gráfica descritos en la sección de gráficas bidimensionales.

Ejemplo:

```
octave:51> z = [0:0.05:5];
octave:52> plot3(z, exp(2i*pi*z), "3;sinusoidal compleja;")
```

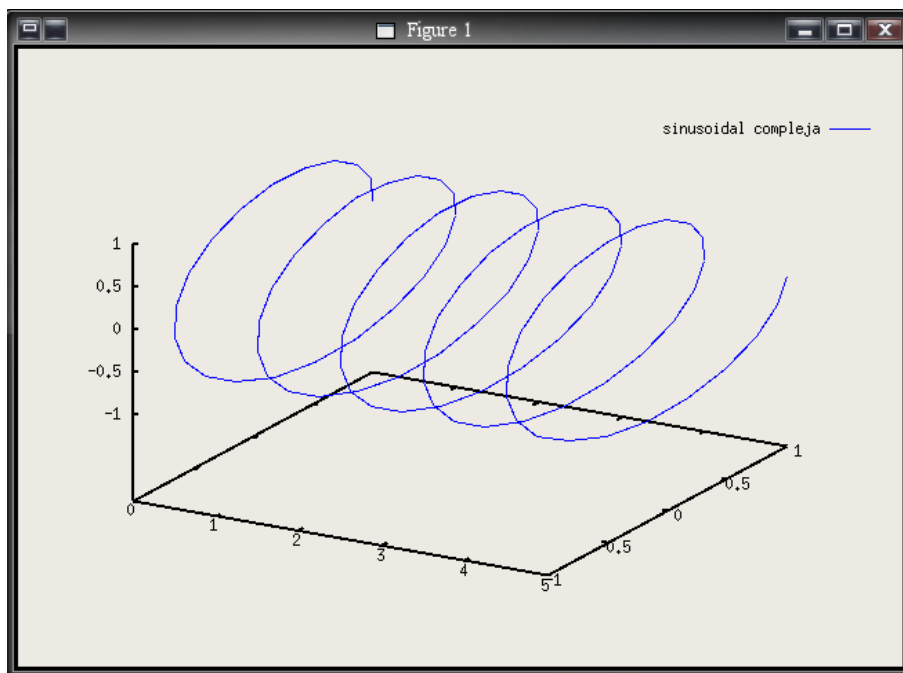


Imagen 4: Gráfica de una sinusoidal compleja con 'plot3()'

El comando `mesh(x,y,z)` hace una representación tridimensional dado dos vectores x e y, y una matriz bidimensional z. Generalmente se usa el comando `meshgrid` para generar los datos que usará `mesh` para para representar los ejes x e y.

Ejemplo:

```
octave:1> x=[-2:0.1:2]; % genera el vector
octave:3> [xx,yy] = meshgrid(x,x); % genera las matrices de ejes
octave:4> z=sin(xx.^2 - yy.^2); % funcion z=sen(x^2 - y^2)
octave:5> grid % genera una rejilla
octave:6> mesh(x,x,z) % crea el grafico
```

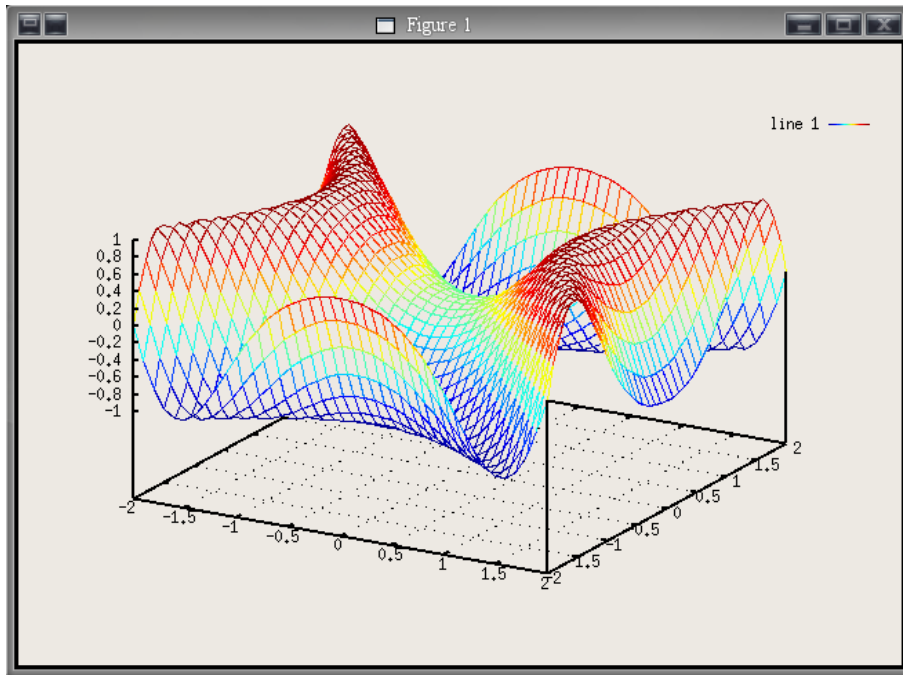


Imagen 5: Generación de gráficos mediante 'mesh()'

La función `contour(x,y,z)` recibe los mismos argumentos que `mesh()` y dibuja las curvas de nivel de la superficie.

Vamos a generar las curvas de nivel del gráfico generado en el ejemplo anterior:

```
octave:10> contour(x,x,z);
```

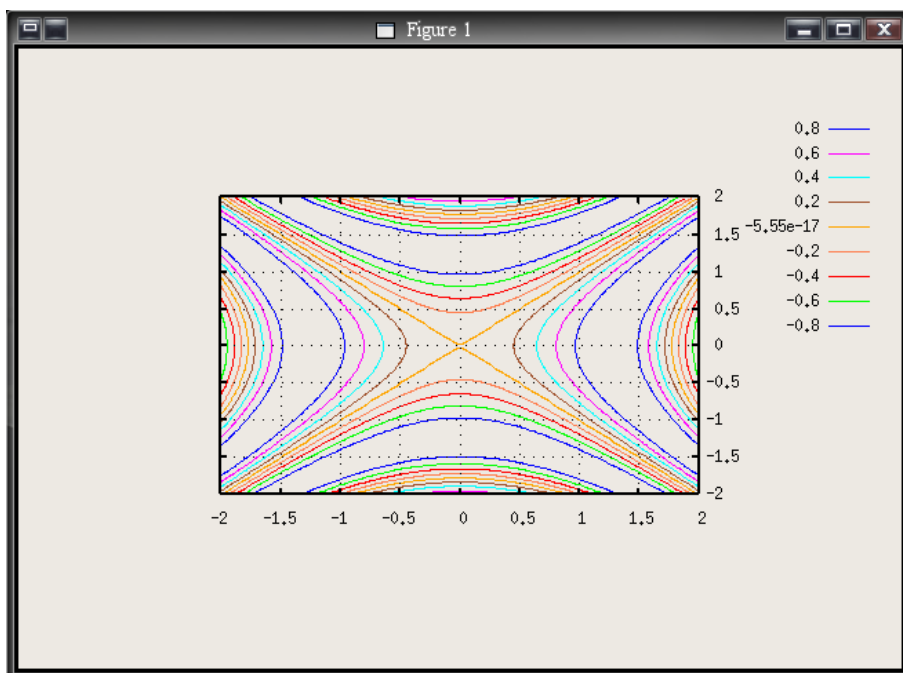


Imagen 6: Curvas de nivel generadas por medio de 'contour()'

Modificar gráficas

Los siguientes comandos nos permiten controlar y modificar las gráficas mostradas en la ventana de gráficos.

clearplot borra el contenido de la ventana de gráficos actual.

clg igual que clearplot

hold on mantiene el gráfico actual en la pantalla, permitiendo superponer gráficos subsiguientes.

hold off desactiva la superposición de gráficos.

hold cambia el estado actual en que se encuentra hold

ishold () devuelve 1 si está activada la superposición de gráficos (hold on)

axis(v) donde **v** es un vector de 2, 4 o 6 elementos, los cuales representan los límites de los ejes x, y o z.

axis activa el autoescalado.

grid("on") dibuja una rejilla en la ventana (opción por defecto).

grid("off") desactiva la rejilla de la ventana.

title("titulo") agrega un título a la gráfica.

xlabel("etiqueta") agrega una etiqueta al eje x.

ylabel("etiqueta") agrega una etiqueta al eje y.

zlabel("etiqueta") agrega una etiqueta al eje z.

closeplot cierra la ventana de gráficos.

Gráficos múltiples

En Octave podemos mostrar diferentes gráficos en una misma ventana, para ello utilizamos la función **subplot(filas, columnas, indice)**. El comando subplot divide la ventana de gráficos en un arreglo de ventanas mas pequeñas. El argumento *indice* especifica el la ventana actual dentro del arreglo.

El *índice* se desplaza primero por cada ventana dentro de una fila para luego pasar a la otra, de izquierda a derecha y desde arriba hacia abajo.

Por ejemplo, para un plot de 3x2 el índice se movería de la siguiente manera:

1	2	3
4	5	6

En el siguiente ejemplo crearemos dos gráficos, uno encima del otro.

```
octave:29> x = linspace(-10, 10);
octave:30> subplot(2,1,1)           % plot de 2 filas, 1 columna, situado en 1
octave:31> plot(x, sin(x))          % genera el primer gráfico
octave:32> subplot(2,1,2)          % ahora nos situamos en la segunda ventana
octave:33> plot(x, sin(x)./x)      % genera el segundo gráfico
```

El gráfico generado es el siguiente:

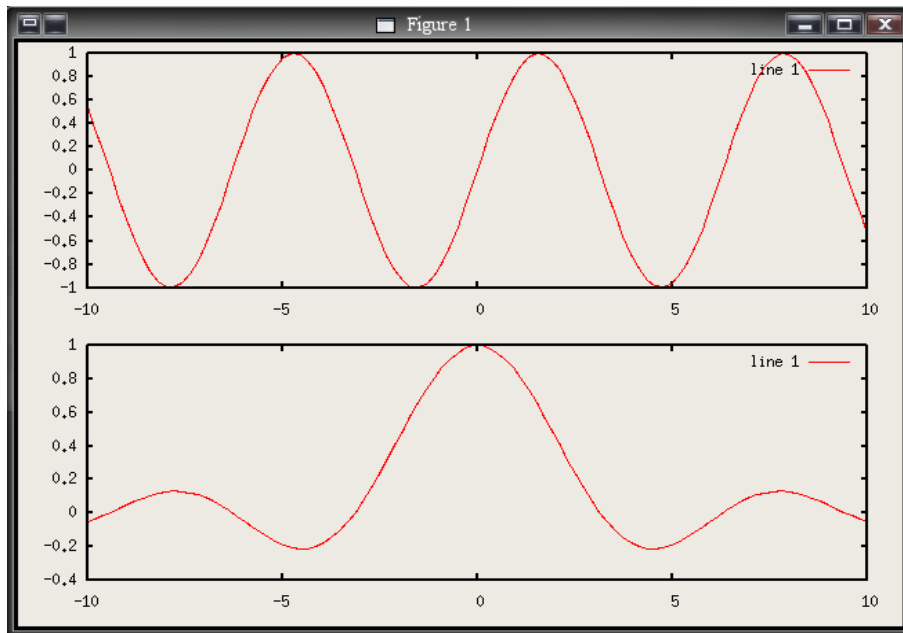


Imagen 7: Gráficos multiples en una ventana con 'subplot'

Recursos Web para Octave

Página web del proyecto Octave (Inglés)

<http://www.gnu.org/software/octave/>

Manual en línea del proyecto Octave (Inglés)

<http://www.gnu.org/software/octave/doc/interpreter/>

Apuntes y ejercicios. Página de la asignatura *Informática básica* de la Universitat Jaume I de España.

<https://www.unoweb-s.uji.es/0304/N13/ficheros0/>

Manual en línea. *Introducción al Octave*, por Alberto F. Hamilton Castro.

<http://www.isaatc.ull.es/asignaturas/octave/ApuntesOctave/ApuntesOctave.html>

Manual en línea. *Herramientas en GNU/Linux para estudiantes universitarios.*

Gnu/Octave: cálculo numérico por ordenador, por Juan José García Rojo

<http://www.isaatc.ull.es/asignaturas/octave/ApuntesOctave/ApuntesOctave.html>

Manual en línea. *Introduction to Octave*, por Hubert Selhofer, revisado Marcel Oliver (Inglés)

<http://math.jacobs-university.de/oliver/teaching/iub/resources/octave/octave-intro/octave-intro.html>

Manual. *Introduction to Octave*, por Dr. P.J.G. Long. Departamento de Ingeniería de la Universidad de Cambridge . (Inglés)

<http://www-mdp.eng.cam.ac.uk/CD/engapps/octave/octavetut.pdf>

GNU Free Documentation License

Version 1.2, November 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc.
51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA
Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

The purpose of this License is to make a manual, textbook, or other functional and useful document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondly, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In

addition, you must do these things in the Modified Version:

- A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.
- C. State on the Title page the name of the publisher of the Modified Version, as the publisher.
- D. Preserve all the copyright notices of the Document.
- E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- H. Include an unaltered copy of this License.
- I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.
- N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.
- O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See <http://www.gnu.org/copyleft/>.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.