Curso de GNU Octave

Primeros pasos con la herramienta

Por

David Pacios Izquierdo

UNIVERSIDAD COMPLUTENSE
MADRID

ASCII - FDI-UCM
Oficina de Software Libre y Tecnologías Abiertas

Guía para comenzar con la herramienta GNU Octave orientada al curso académico 2018-2019. Realizada para la Oficina de Software Libre de la UCM y para la Asociación Socio-Cultural de Ingenierías en Informática.

Julio 2018

Sobre OTEA

Misión
La Oficina de Software Libre podría ser considerada como un proyecto que recoge la esencia de lo que la Universidad Complutense de Madrid significa: una Institución educativa que pretende ser un núcleo de formación, investigación y construcción con una meta clara: el desarrollo de sujetos LIBRES (LIBERTAD) que, con su talento, contribuyen a un mundo cada vez más complejo y global (UNIVERSO, UNIVERSIDAD).

OFICINA DE **SOFTWARE LIBRE**
VICERRECTORADO DE TECNOLOGÍAS DE LA INFORMACIÓN
UNIVERSIDAD **COMPLUTENSE** MADRID

Visión
Ofrecer a la comunidad universitaria complutense (personal docente investigador, personal administrativo y servicios, estudiantes) alternativas libres al software que tiene a su disposición, tanto en el puesto de trabajo, como fuera del mismo.
La exploración, valoración y elección de estas tecnologías no supone que sean concebidas como la única alternativa posible o como el resultado de una especie de imposición, sino más bien de un proceso continuo y dinámico que comprende conocimiento, experimentación y comparación con lo existente con una meta formativa tanto en lo conceptual como en lo práctico.

Más información: `https://www.ucm.es/oficina-de-software-libre/`

Sobre ASCII

ASCII es la asociación a la que pertenezco (y de la que en el momento de la creación de este manual, presido) es una asociación creada en 1993 con la idea de reunir y tener un equilibrio entre el conocimiento y el ocio. Uno no excluye el otro, la idea de poder trabajar de forma distinta, como una familia, es la que se lleva intentando mostrar desde el comienzo.

Figura 1: *Logo de la asociación ASCII*

En los últimos años hemos tenido el honor de poder participar en iniciativas interesantes como:

- **FDIst** (Grupo de seguridad de la Facultad de Informática)
- **OTEA** (Oficina de Software Libre de la UCM)
- **GamersParty** (Colaboración con LAG: Asociación de videojuegos de la UCM) con motivo solidario.
- **Cryptoparty:** Evento de ciberseguridad.
- **Cursos, talleres y Clases de apoyo.**

Mantenemos una gran relación con el decanato, gerencia y personal docente de la Facultad de Informática. También mantenemos una gran relación con otras facultades gracias a la iniciativa de **"El Concilio de asociaciones"**. Podemos contar con orgullo como muchos de nuestros socios han acabado investigando o siendo profesores de la misma facultad. Actualmente seguimos teniendo esa mentalidad de ayudar a todos los alumnos con nuestros talleres, conferencias, cursos y de seguir mostrando ese ambiente de ocio friky que tanto nos caracteriza. Entre estas actividades destacamos:

- **Sesiones de ROL** con masters expertos.
- **Préstamo de juegos de mesa** para uso y disfrute de todos.
- **Comidas y celebraciones** (Navidad, Halloween) con eventos temáticos.
- Concurso de comer Gofres y Chocolate.

En el año 2018 también inauguramos, gracias a gerencia de la facultad de informática, al Ayuntamiento de Madrid, a cuidado de jardines de la UCM y a un error del presidente, **un estupendo bosque con 3 árboles** propiedad de la asociación.

Más información: `https://ascii.fdi.ucm.es/`

Sobre TEFLON

Teflon(cc by-nc 4.0) es una plantilla de LaTeX creada por David Pacios Izquierdo con fecha de Enero de 2018. Con atribuciones de uso CC by-nc

Esta plantilla fue desarrollada para facilitar la creación de documentación profesional para Trabajos de Fin de Grado o Trabajos de Fin de Máster. La versión usada es la 1.2.

Contacto
Autor: David Pacios Izquiero
Correo: dpacios@ucm.es
ASCII: asciifdi@gmail.com
Despacho 110 - Facultad de Informática

Sobre el curso de GNU Octave

El curso de Octave, al igual que la siguiente documentación, esta indicada para niveles básicos de conocimiento de GNU Octave. Esta herramienta bien usada puede ser útiles en campos como: Medicina, Ingeniería, Ciencias, Métodos Estadísticos... También es una herramienta para generar demostraciones y gráficas de forma profesional. Es un certificado de calidad para documentos oficiales y para informes de cualquier tipo.

Con este curso se intenta hacer llegar una gran herramienta libre a las personas interesadas. El nivel citado anteriormente es el básico. Para un uso medio o avanzado se recomienda investigar y leer fuentes oficiales y documentación oficial.

Índice general

Capítulo 1

GNU Octave

GNU Octave es un software con un alto nivel de programación, utilizado para la numeración computacional. Mediante su interfaz podemos resolver numéricamente problemas tanto lineales como no lineales.[1]
Por otro lado, Octave tiene muchas herramientas para resolver problemas tipo algebraico, integrales, funciones ordinarias e integrales. Además, si se poseen conocimientos de C++, C, FORTRAN u otros lenguajes se puede modificar la interfaz de Octave.[2]

1.1. Ventajas de GNU Octave

No tiene interfaz gráfica, por lo que acepta estructuras más complejas.
Cada día recibe nuevas contribuciones de distintos programadores para introducirle nuevas mejoras.
Parte del proyecto GNU y todas las publicaciones de licencia GPL se basa en bibliotecas de código abierto.[3]

1.2. Diferencias con Herramienta Matlab

Matlab es un gran programa con una buena interfaz gráfica. Tiene una gran variedad de formas de representación gráfica. Su interfaz gráfica está escrita en Java, por lo que no es tan estable como se desearía.
Por otro lado, Octave no tiene una interfaz gráfica[4] por lo que acepta estructuras más complicadas. Y además, permite un acceso interno a su maquina interna y a su librería si el programador tiene conocimientos en C++.

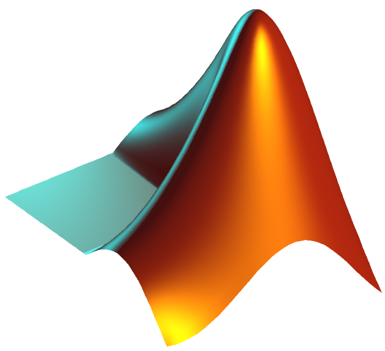


Figura 1.1: *Herramienta Matlab*

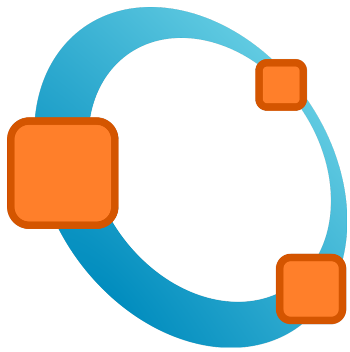


Figura 1.2: *Herramienta GNU Octave*

1.3. Usos de Octave

Octave se utiliza generalmente para la resolución de ecuaciones tanto lineales como no lineales, integrales de distinto orden, polinomios de distinto grado, distintas ecuaciones algebraicas y la representación gráfica de distintas funciones.
En este curso, vamos a centrarnos en la representación gráfica, la construcción de matrices, todo tipo de operaciones con matrices y resolución de ecuaciones sencillas.

Capítulo 2

Instalación

La instalación de la herramienta puede parecer algo intuitivo, y lo es, pero nunca viene mal recordar los conceptos. También vamos a enseñar los distintos elementos del panel de herramientas de GNU Octave.

2.1. Instalación en Manjaro

¿Qué es Manjaro?

Para el escritor de este documento, el mejor sistema operativo que existe. Está basado en Arch Linux y es un Sistema Operativo libre y gratuito.

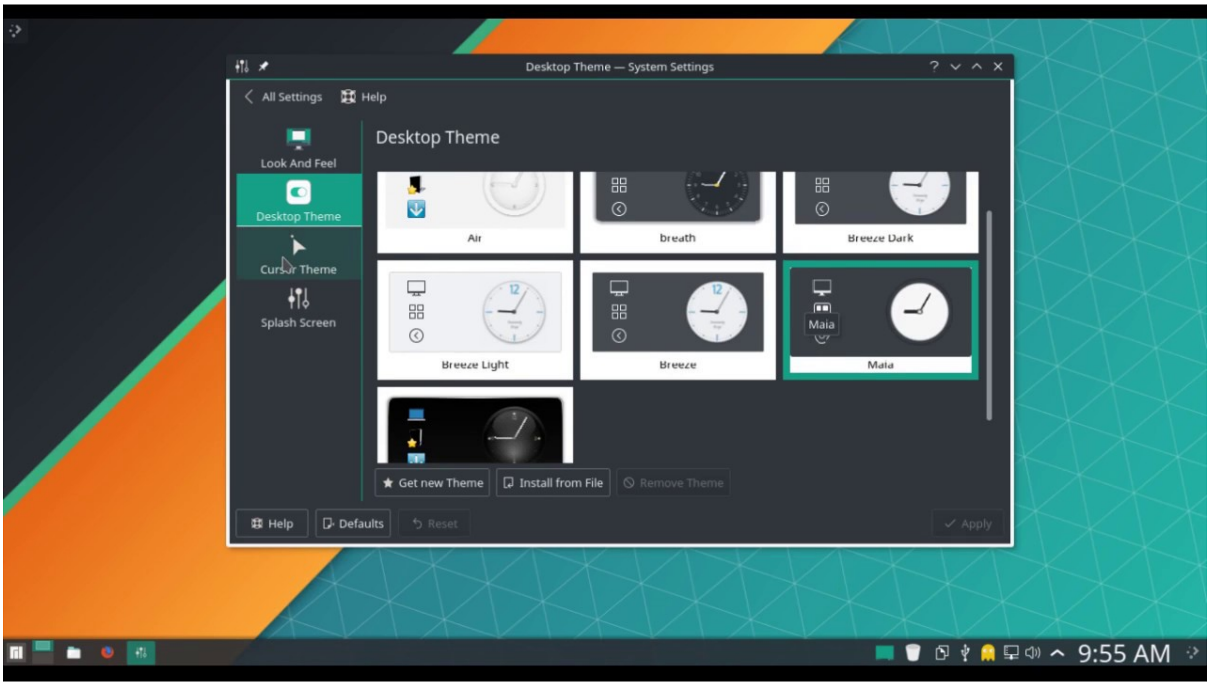


Figura 2.1: *Escritorio de Manjaro*

Vamos a proceder a instalar la herramienta en Manjaro, para este manual, se usó la imagen para VirtualBox con las siguientes características.

VirtualBox - Manjaro 17.1.6
SHA256:
23411c3f4df8a140a27a589e3378cdfab80f2a46ada24815ae1175036513a481
Username: osboxes
Password: osboxes.org
Root Account Password: osboxes.org
VB Guest Additions VMware Tools: Not Installed
VMware Compatibility: Version 10+
Link: `https://www.osboxes.org/manjaro/#manjaro-17-1-6-info`

Figura 2.2: *Pantalla inicial en la VM*

Ahora vamos a abrir una consola de comandos en Manjaro, para ello nos situaremos en la esquina inferior izquierda y abrimos una consola de comandos.

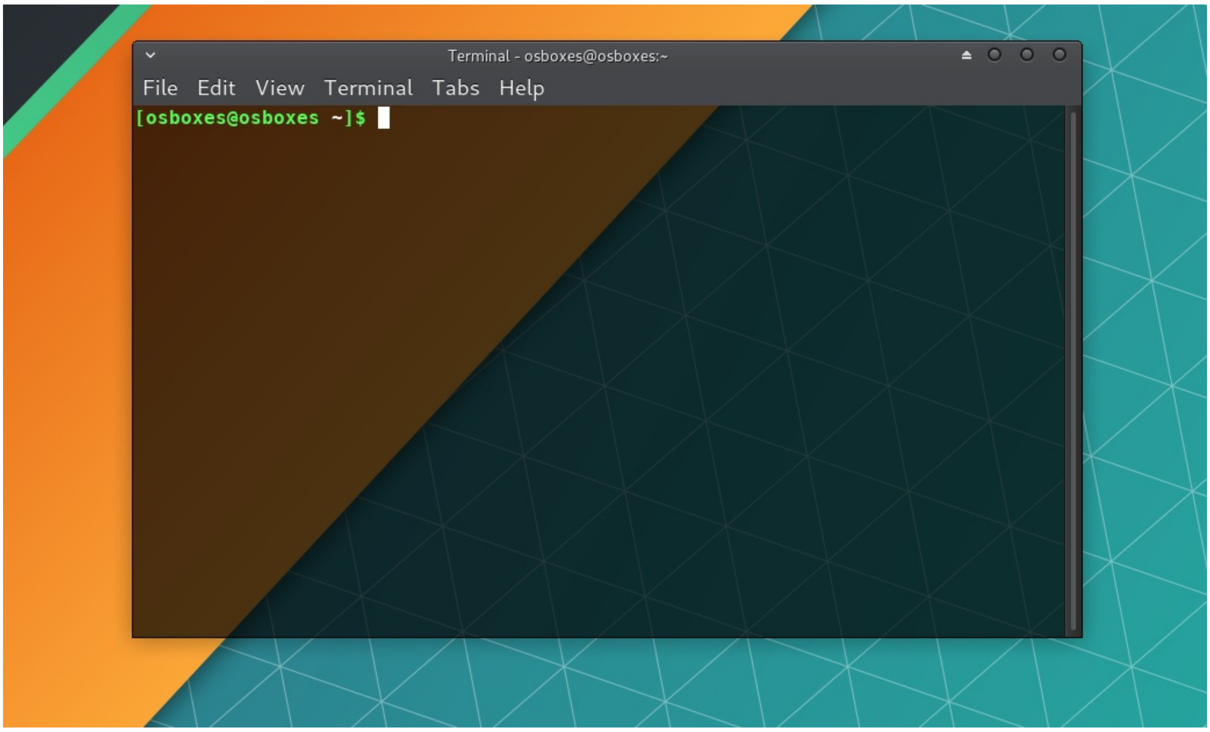


Figura 2.3: *Vista de la Consola de comandos de Manjaro*

Ahora vamos a introducir el comando siguiente:

SUDO PACMAN -SYU OCTAVE

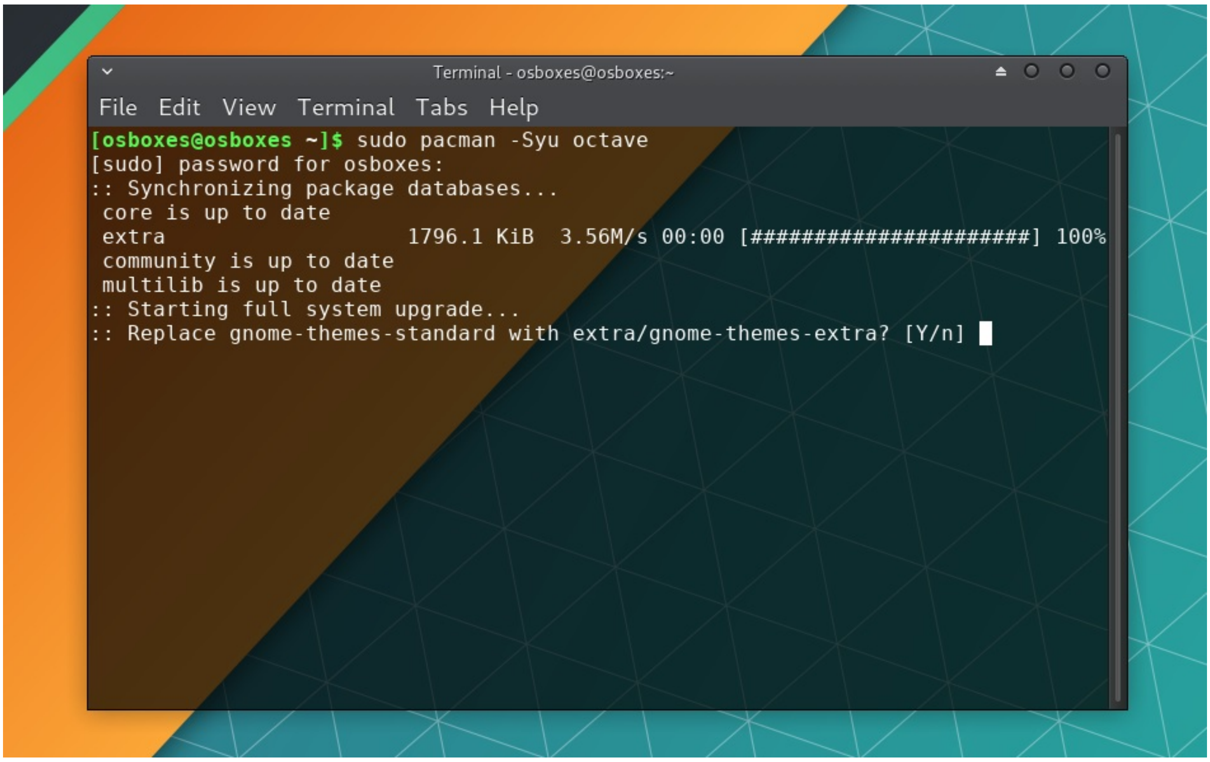


Figura 2.4: *Vista de la instalación*

Después de pulsar la tecla 'Y' varias veces, comienza la instalación.

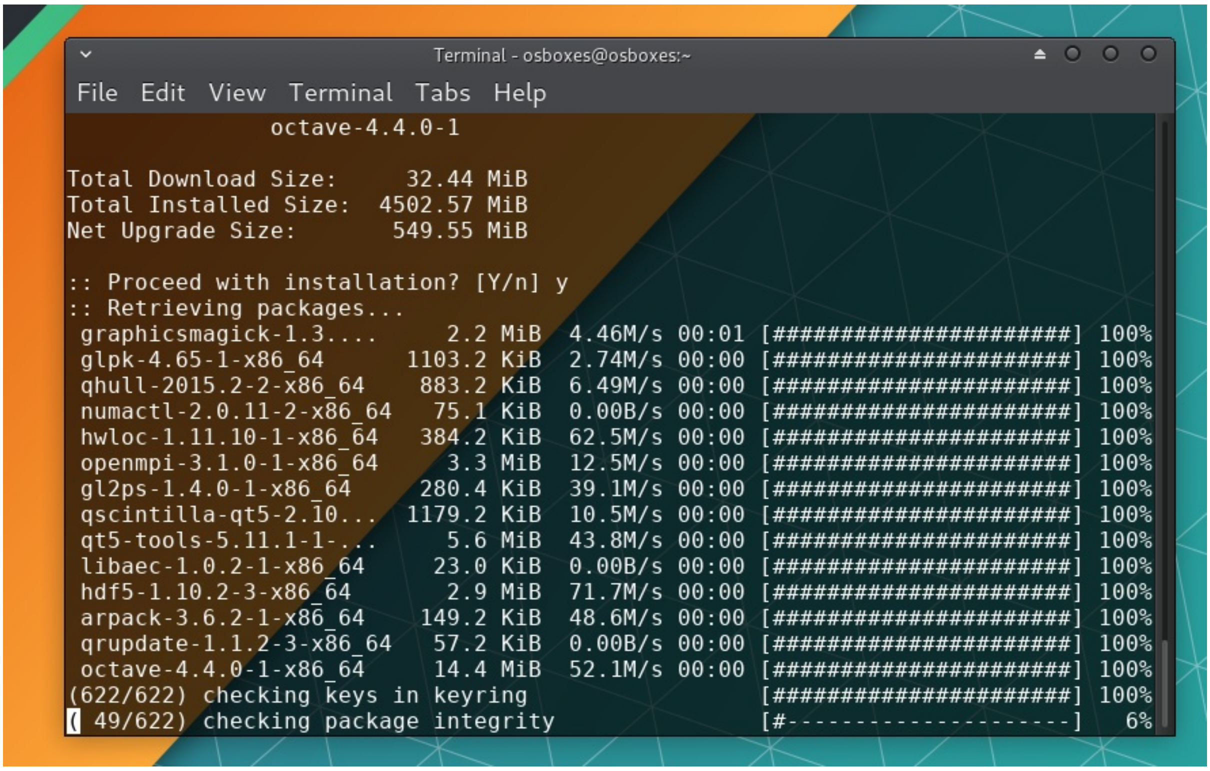


Figura 2.5: *Vista de la instalación 2*

Ya tendríamos instalado Octave en nuestro Manjaro

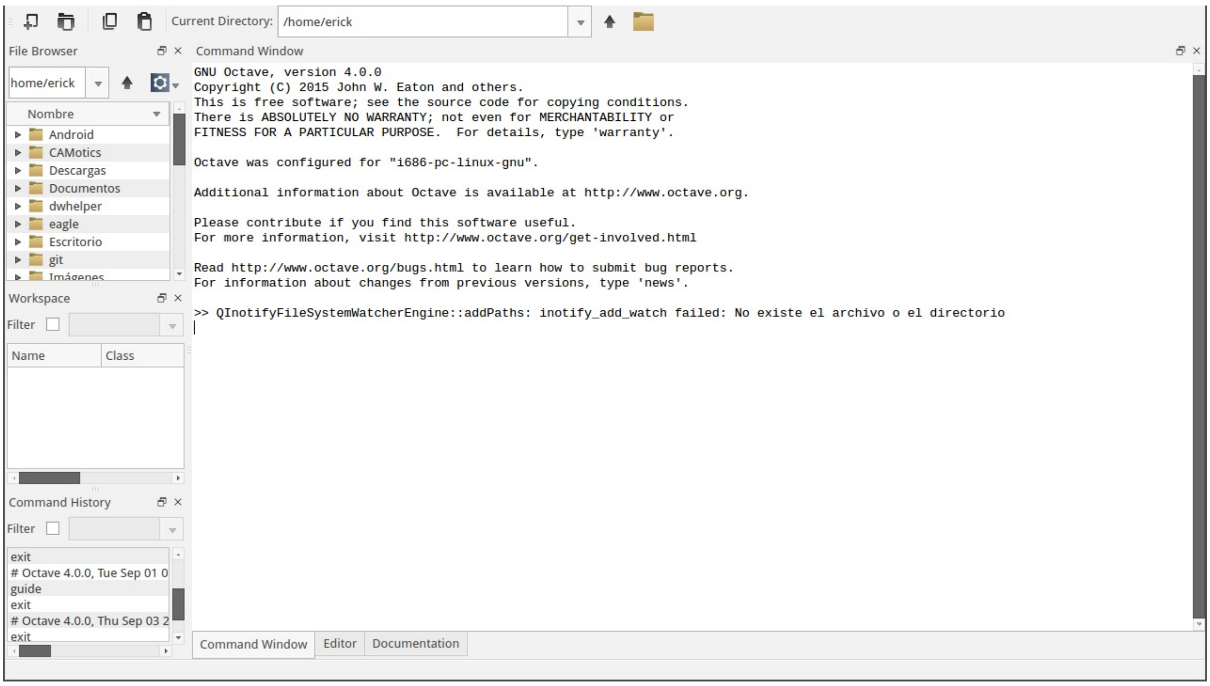


Figura 2.6: *Vista de Octave*

2.2. GNU Octave para GNU/Linux

GNU Octave viene preinstalado en la mayoría de las distribuciones conocidas de GNU/Linux y no habría que hacer nada para tenerlo. Si se es un usuario avanzado, se recomienda instalar los paquetes de C++ para podre crear plugins avanzados.

Más Información: HTTPS://WWW.GNU.ORG/SOFTWARE/OCTAVE/DOWNLOAD.HTML

2.3. Instalación para Windows

Para la instalación usaremos Windows 10 (cualquier versión), entraremos en la página citada arriba: `https://www.gnu.org/software/octave/download.html`

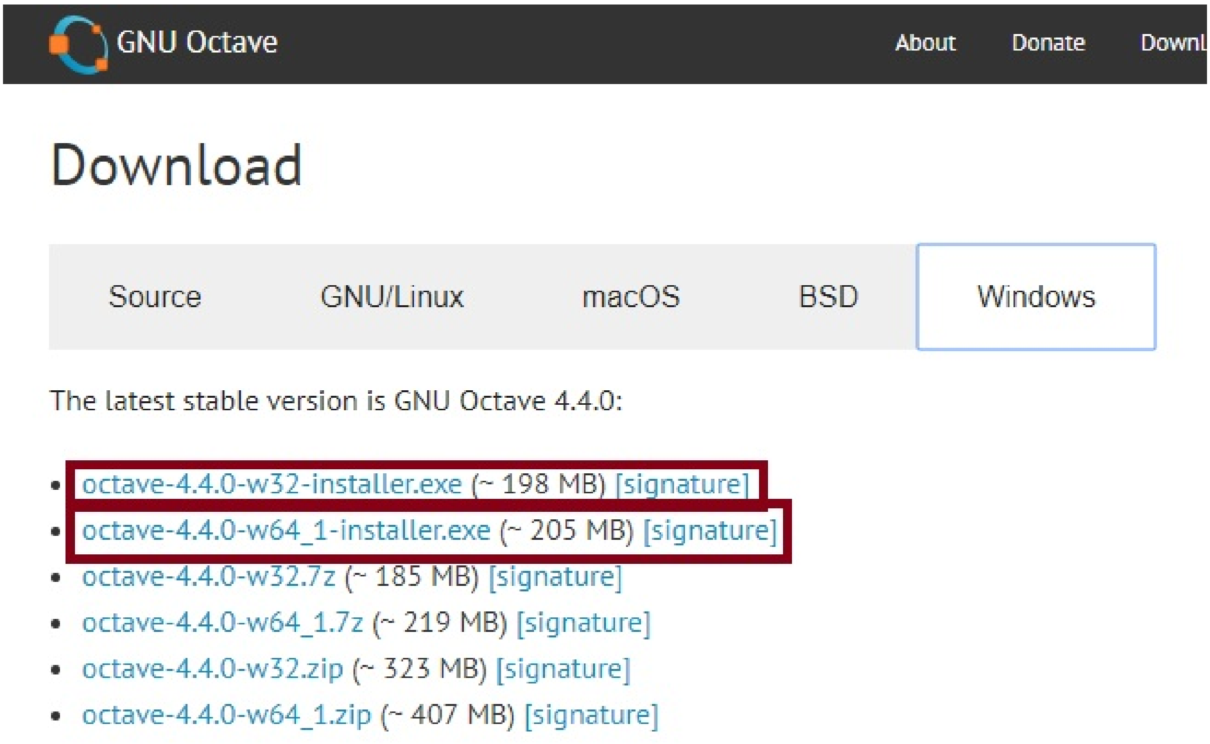


Figura 2.7: *Descarga para Windows*

Aquí nos tendremos que dar cuenta si nuestro sistema operativo es de 32 bits por lo que descargaríamos el de arriba. Si nuestro sistema operativo es de 64, descargamos el señalado abajo.

La instalación es sencilla, vamos a descargar el archivo y guardarlo donde proceda. Ahora lo ejecutamos. Pondrá un aviso diciendo que no se ha probado totalmente en Windows 10. Lo ignoramos y pulsamos (SI).

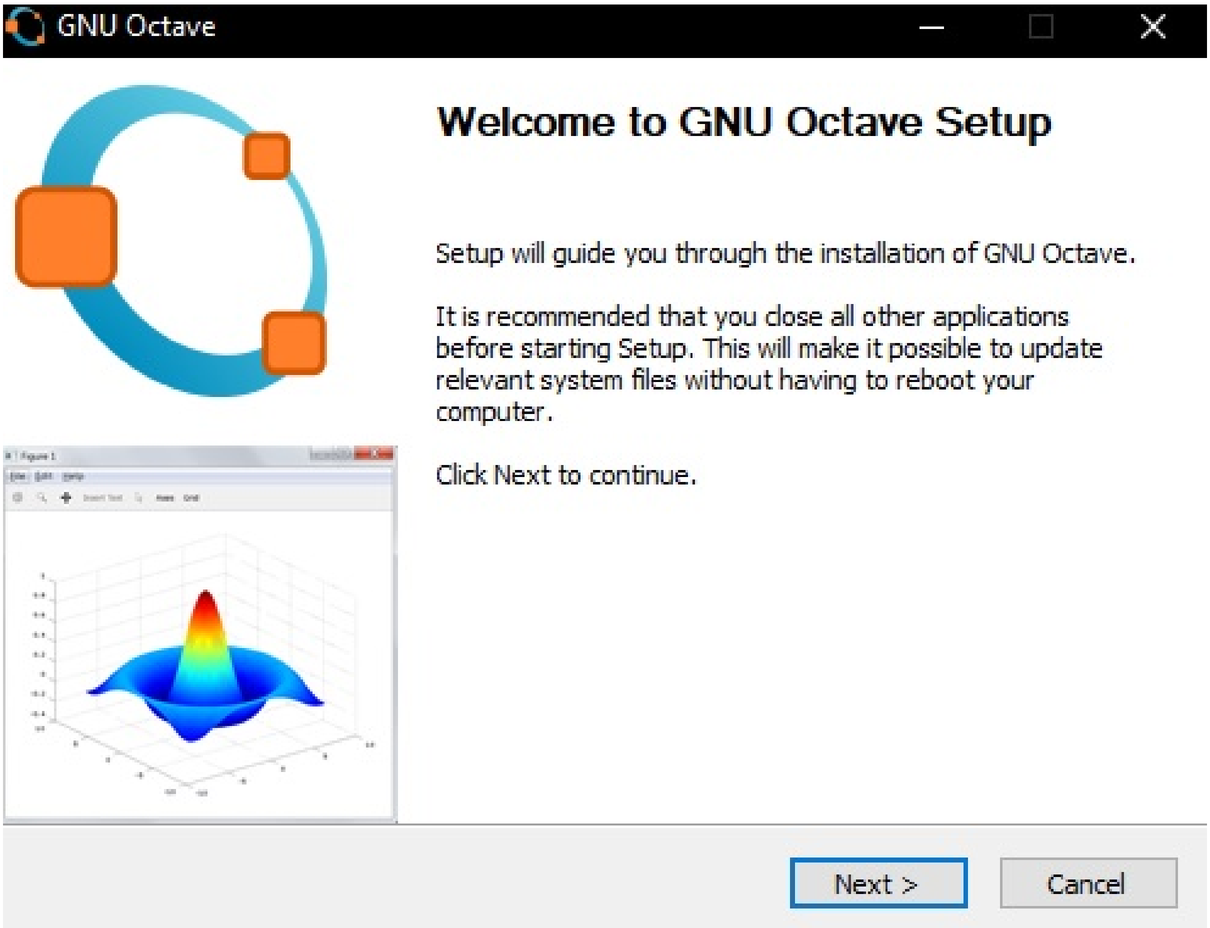


Figura 2.8: *Vista del instalador*

Despues de pulsar varias veces "next" hora tenemos la herramienta instalada.
Lo primero que nos fijamos es que podemos ejecutar la GUI (Herramienta con interfaz) o la consola de comandos de Octave.

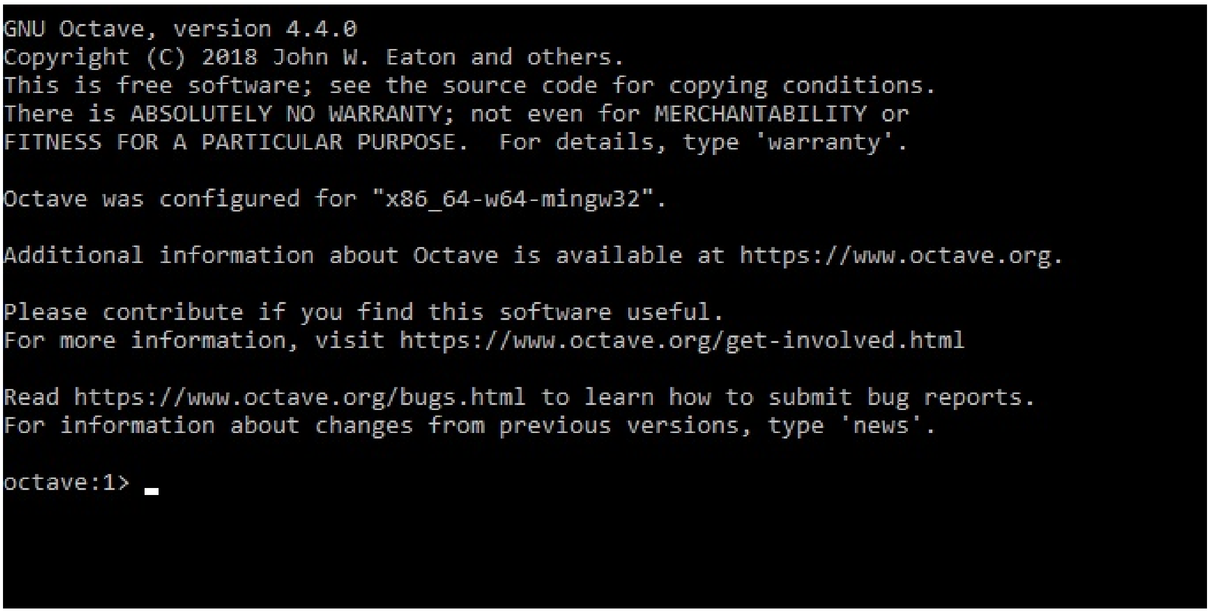


Figura 2.9: *Consola de comandos de Octave*

La consola de comandos se puede coger directamente pulsando el ejecutable de octave-cli. Puede usarse para funciones básicas y programación, pero no para dibujar gráficas ni resultados.

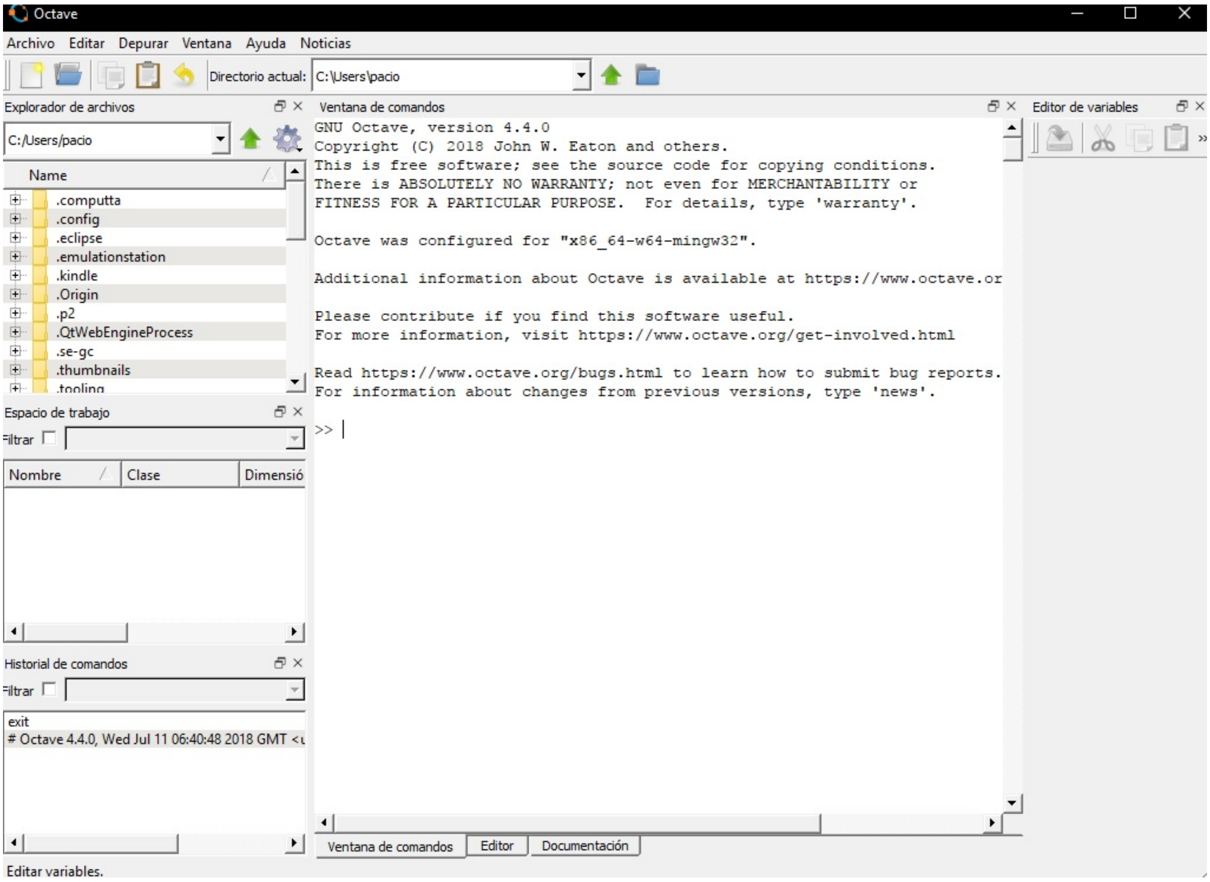


Figura 2.10: *GUI de Octave*

2.4. Interfaz de Octave

La Interfaz de Octave es compleja, nos vamos a centrar en lo principal para realizar el curso y poder seguir aprendiendo con lo que se de en el manual. Siempre es recomendable tener los documentos oficiales a mano. La mejor documentación y manuales posibles son los que se sitúan en GNU Octave.

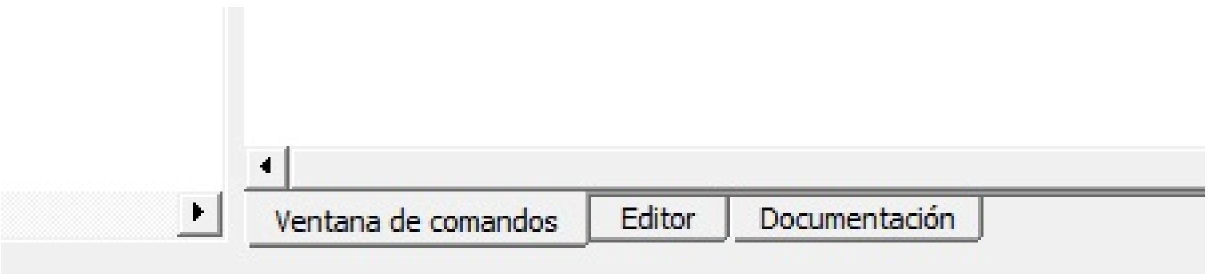


Figura 2.11: *Panel inferior de Octave*

En este panel inferior se sitúan los 3 botones que usaremos, las 3 pestañas. La ventana de comandos es la principal, en ella se pueden realizar operaciones básicas al igual que en la consola anterior (octave-cli).

```
>> 40*10+20
ans =  420
>>
```

Figura 2.12: *Uso de la consola de comandos de Octave*

El panel de editor nos servirá, como veremos más adelante, para escribir operaciones complejas como: ecuaciones, gráficas y también nos servirá para crear constraints (restricciones) de parámetros. Por ejemplo para optimización de problemas, uno de los usos más avanzados.

```
%Integral del polinomio x^2+1 del punto 0 a 3
c=[1,0,1];
integral=polyint(c);
area=polyval(integral,3)-polyval(integral,0)
```

Figura 2.13: *Función del editor*

En el panel de herramientas lo que más vamos a usar es la ruleta de guardar y ejecutar, marcada abajo.

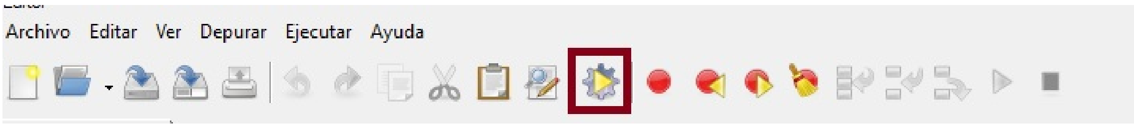


Figura 2.14: *Panel de herramientas*

Y de las cosas más útiles, la documentación. Mucho de este curso se ha sacado de ahí, perfectamente referenciado.

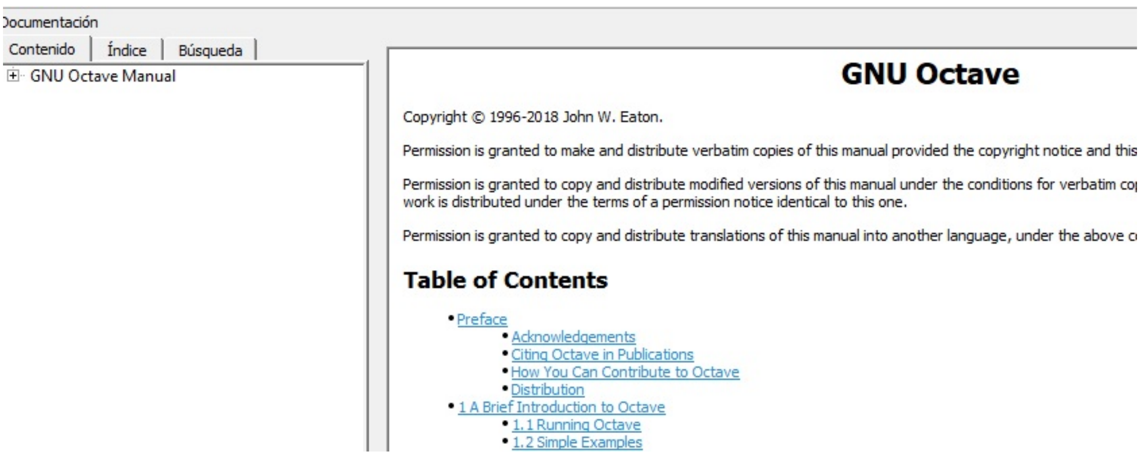


Figura 2.15: *Documentación de GNU Octave*

Capítulo 3

Sintaxis básica

3.1. Operaciones básicas

Primero, vamos a empezar con el cálculo básico de operaciones este campo es conocido como las operaciones aritméticas.
Por un lado, podemos sumar dos números sencillos en la misma ventana de comandos y nos dará el resultado en la misma ventana con el comando **ans**.

```
>> 45+76
ans =  121
```

Figura 3.1: Ejemplo suma en ventana de comandos

También podemos verificar la identidad de Euler, ya que, Octave reconoce variables como e o pi.[5]

```
>> 1*pi
ans =  3.1416
>> exp(i*pi)
ans = -1.0000e+000 + 1.2246e-016i
```

Figura 3.2: Euler en ventana de comandos

3.2. Matriz y vectores

Con este tipo de operaciones tenemos el nivel más básico de Octave. Ahora, vamos a ver como se construyen matrices y vectores.

```
>> A=[1,1;1,1]
A =

   1   1
   1   1
```

Figura 3.3: Matriz en ventana de comandos

Como vemos en la imagen, la matriz se representa entre corchetes, separando las filas con comas y las columnas con puntos y comas. También podemos crear matrices aleatorias con el comando **rand** indicando entre paréntesis las dimensiones de nuestra matriz como podemos ver en el siguiente ejemplo:

```
>> B=rand(3,2)
B =

   0.59850   0.48918
   0.41901   0.36475
   0.75436   0.50149
```

Figura 3.4: Comando rand en ventana de comandos

Como he dicho anteriormente, podemos representar vectores, del mismo modo que una matriz pero no vamos a utilizar los puntos y coma.

```
>> C=[1,2,3,4,5]
C =

   1   2   3   4   5
```

Figura 3.5: Representación vector

3.3. Operaciones básicas con matrices y vectores

Octave utiliza la notación estándar matemática que no requiere un alto nivel de programación para hacer este tipo de operaciones.
Con esta herramienta, podemos desde multiplicar matrices hasta dividir matrices y vectores de dimensión similar.

```
>> A=[1,2,3;4,5,6]
A =

   1   2   3
   4   5   6

>> 2*A
ans =

    2    4    6
    8   10   12
```

Figura 3.6: Multiplicación entre un número y matriz

Como se puede ver, no hay ningún problema en multiplicar un número por una matriz cualquiera.
Ahora vamos a ver como se pueden multiplicar dos matrices entre ellas y la división de las mismas.

```
>> B.*A
ans =

    4   10   18
   28   40   54
```

Figura 3.7: Multiplicación entre matrices

Si queremos que la multiplicación o la división se haga elemento a elemento colocaremos un punto antes de la operación para que el programa lo realice de esta manera.

```
>> B.\A
ans =

   0.25000   0.40000   0.50000
   0.57143   0.62500   0.66667
```

Figura 3.8: División entre matrices

```
>> A=[1,2,3;4,5,6];
>> A'
ans =

   1   4
   2   5
   3   6
```

Figura 3.9: Matriz inversa

Con este apóstrofe, le indicamos que tiene que hacer la matriz inversa de esta matriz. Con estos pequeños ejemplos tenemos una idea clara de las distintas operaciones que podemos realizar con Octave. Ahora voy a colocar una tabla con los distintos símbolos y lo qué significan.

Funciones	**Significado**
abs	Valor absoluto
rats	Argumento de una fracción
sqrt	Raíz cuadrada
exp	Exponencial
log	Logaritmo neperiano
sin,cos,tan,sec,csc,cot	Razones trigonométricas medidas en radianes
asin,acos,atan,asec,acsc,acto	Funciones trigonométricas inversas

3.4. Resolviendo ecuaciones lineales

Después de ver las operaciones sencillas con matrices, vamos a ver cómo Octave resuelve ecuaciones lineales sencillas. Para ello, primero damos un valor a una variable x y a partir de ella, veremos como resolver una variable y.

```
>> x=B.*A
x =

    4   10   18
   28   40   54

>> y=x+10./A
y =

   14.000   15.000   21.333
   30.500   42.000   55.667
```

Figura 3.10: Ejemplo de ecuación lineal

3.5. Resolviendo ecuaciones diferenciales

Seguidamente, vamos a ver como Octave resuelve las ecuaciones diferenciales. Estas son más complicadas de poner en el editor o en la ventana de comando del Octave.

```
%Condición inicial
x(t=t0)=x0
x0=[1;2];
function xdot=f(x,t)
  r=0.25;
  k=1.4;
  a=1.5;
  b=0.16;
  c=0.9;
  d=0.8;
  xdot(1)=r*x(1)*(1-x(1)/k)-a*x(1)*x(2)\(1+b*x(1));
  xdot(2)=c*a*x(1)*x(2)/(1+b*x(1))-d*x(2);
endfunction
t=linspace(0,50,200)';
x=Isode('f',x0,t);
```

Figura 3.11: Ejemplo de ecuación diferencial

Para escribir la ecuación diferencial primero debemos definir una ecuación $f(x,t)$. Y mediante la función **function xdot** vamos a definir el intervalo de la ecuación diferencial. También incluye la función **Isode** que es esencial para la resolución de la ecuación diferencial.[6]

Capítulo 4

Resolución de ecuaciones

Como hemos visto en el capítulo anterior, Octave es capaz de resolver desde una ecuación con una incógnita hasta una ecuación con varias incógnitas. Además puede resolver desde una simple suma hasta una integral en dos variables.

4.1. Resolución de ecuaciones con una incógnita

Primero, escribimos las variables de qué dependen el valor de x y después escribimos la ecuación de x. Con sólo estos dos pasos, resolveremos una ecuación con una incógnita sin problemas.

```
>> A=2;
>> B=12;
>> x=A^2+B^2+2*A*B
x =  196
```

Figura 4.1: Ejemplo de ecuación con una incógnita

Este es un ejemplo muy simple, pero además de resolver ecuaciones tan simples, podemos resolver ecuaciones de segundo grado.

```
disp('Resolución de una ecuación de segundo grado')
disp('=')
disp('')
disp('Introducir el valor de los parámetros ax^2+bx+c
a=input('a=>');
b=input('b=>');
c=input('c=>');
disp('')
disp('Soluciones')
disp('=')
discri=sqrt(b^2-4*a*c);
x1=(-b+discri)/(2*a)
x2=(-b-discri)/(2*a)
```

Figura 4.2: *Código ecuación segundo grado[7]*

```
Resolucin de una ecuacin de segundo grado
=

Introducir el valor de los parmetros ax^2+bx+c=0
a=>22
b=>44
c=>89

Soluciones
=
x1 = -1.0000 + 1.7451i
x2 = -1.0000 - 1.7451i
```

Figura 4.3: *Resolución ecuación*

4.2. Resolución de ecuaciones con varias incógnitas

Con una incógnita hemos visto anteriormente que es muy sencillo, pero cuando se da un caso de un sistema de m ecuaciones lineales y de n incógnitas se pueden dar estos tres casos:

- **Solución única**.
- **No existe solución**.
- **Infinitas soluciones**.

Para saber si el sistema tiene una solución única, no existe o tiene varias. Obtenemos de él la matriz y la matriz ampliada y las dividimos entre sí y el sistema nos dará el valor de las n incógnitas.
Ejemplo:

$$\begin{cases} 3x + 2y + z = 1 \\ 5x + 3y + 4z = 2 \\ x + y - z = 1 \end{cases}$$

```
a=[3,2,1;5,3,4;1,1,-1];
b=[1;2;1];
a\b
```

Figura 4.4: *Sistema de ecuaciones*

```
ans =

  -4.00000
   6.00000
   1.00000
```

Figura 4.5: *Solución sistema de ecuaciones*

4.3. Resolución de derivadas

Ahora vamos a ir complicando la cosa, vamos a ver cómo Octave es capaz de hacer derivadas y más adelante veremos cómo se representan gráficamente.

```
d=[2,1,1,1,1];
derivada= polyder(d)
```

Figura 4.6: *Función polyder*

```
>> derivadas

derivada =

   8   3   2   1
```

Figura 4.7: *Resultados de la derivada*

Con la función **polyder** debemos indicar previamente el polinomio f mediante un vector en el que se ordenarán los grados del polinomio de x y después el programa nos mostrará los resultados de la derivada como un vector.

4.4. Resolución de integrales

Posteriormente, vamos a ver cómo se integra con Octave, para ello utilizaremos la función **polyint**[8], sólo con esta función nos devuelve el valor de la integral como un vector, lo mismo que el ejemplo anterior.

```
%Integral del polinomio x^2+1 del punto 0 a 3
c=[1,0,1];
integral=polyint(c);
area=polyval(integral,3)-polyval(integral,0)
```

Figura 4.8: *Función polyint*

```
>> integrales

area =  12
```

Figura 4.9: *Resultados de la integral*

4.5. Ejercicios propuestos

Ejercicio 1. Resuelve el valor de la incógnita x si la función $x = A^2 + 2B$:

$$A = \begin{bmatrix} 1 & 3 & 6 \\ 4 & 2 & 3 \end{bmatrix}; B = \begin{bmatrix} 8 & 2 & 3 \\ 5 & 2 & 1 \end{bmatrix}$$

Solución propuesta:

```
A=[1,3,6;4,2,3];
B=[8,2,3;5,2,1];

x=A.^2+2.*B
```

```
x =

   17   13   42
   26    8   11
```

Figura 4.10: Solución ejercicio 1

Ejercicio 2. Resuelve el valor de la incógnita x si la función $x = A^2 + B^2 + 2AB$ si:

$$A = \begin{bmatrix} -1 & 2 \\ 2 & 2 \end{bmatrix}; B = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

Solución propuesta:

```
A=[-1,2;2,2];
B=[1,1;1,1];

x=A.^2+B.^2+2.*(B.*A)
```

```
x =

   0   9
   9   9
```

Figura 4.11: Solución ejercicio 2

Ejercicio 3. Resuelve el valor de las incógnitas x e y si las funciones $x = \sqrt{A+B} + \dfrac{B}{A}$ e $y = \dfrac{B}{A} + C \cdot D - E^3$.

$$A = 12; B = -21; C = 100, D = 56, E = 150$$

Solución propuesta:

```
A=12;
B=-21;
C=100;
D=56;
E=150;

x=sqrt(A+B)+B.\A;
y=B.\A+C*D-E^3;
```

```
>> x
x = -0.57143 + 3.00000i
>> y
y = -3369400.57143
```

Figura 4.12: Solución ejercicio 3

Ejercicio 4. Resuelve mediante la definición de la ecuación de segundo grado, los dos valores de x si conocemos estos valores:

$$a = 4; b = 8; c = 16$$

Solución propuesta:

```
a=4;
b=8;
c=16;

x1=-b+sqrt(b^2-4*a*c)\2*a;
x2=-b-sqrt(b^2-4*a*c)\2*a;
```

```
>> x1
x1 = -8.00000 - 0.57735i
>> x2
x2 = -8.00000 + 0.57735i
```

Figura 4.13: Solución ejercicio 4

Ejercicio 5. Realiza una suma sencilla de dos matrices A y B:

$$A = \begin{bmatrix} 4 & 4 \\ 6 & 8 \end{bmatrix}; B = \begin{bmatrix} 8 & 10 \\ -6 & -12 \end{bmatrix}$$

SOLUCIÓN PROPUESTA:

```
A=[4,4;6,8];
B=[8,10;-6,-12];

A+B
```

```
ans =

   12   14
    0   -4
```

Figura 4.14: Solución ejercicio 5

Ejercicio 6. Verifica si se puede realizar la multiplicación entre dos matrices y justifica por qué no se puede realizar en el caso que no fuera posible:

$$A = \begin{bmatrix} 1 & 2 & 8 & 9 \\ 10 & 5 & 5 & 20 \\ 7 & 11 & 20 & 40 \end{bmatrix}; B = \begin{bmatrix} 11 & 22 \\ 40 & 50 \end{bmatrix}$$

SOLUCIÓN PROPUESTA:

```
A=[1,2,8,9;10,5,5,20;7,11,20,40];
B=[11,22;40,50];

x=A.*B;
```

```
error: ejercicio6: product: nonconformant arguments (op1 is 3x4, op2 is 2x2)
```

Figura 4.15: Solución ejercicio 6

Ejercicio 7. Una fábrica produce dos modelos de lavadora, A y B, en tres terminaciones: N, L y S. Produce del modelo A: 400 unidades en la terminación N, 200 unidades en la terminación L y 50 unidades en la terminación S. Produce del modelo B: 300 unidades en la terminación N, 100 unidades en la terminación L y 30 unidades en la terminación S. La terminación N lleva 25 horas de taller y 1 hora de administración. La terminación L lleva 30 horas de taller y 1.2 horas de administración. La terminación S lleva 33 horas de taller y 1.3 horas de administración.[9]

A. Representar la información en dos matrices.

B. Hallar una matriz que exprese las horas de taller y de administración empleadas para cada uno de los modelos.

SOLUCIÓN PROPUESTA:

```
 Solucion apartado A
M=[400,200,50;300,100,30];
N=[25,1;30,1.2;33,1.3];
 Solucion apartado B
O=M*N
```

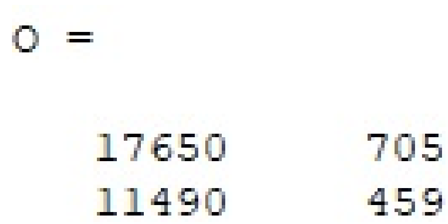


Figura 4.16: Solución ejercicio 7B

Ejercicio 8. Resolver el siguiente sistema de ecuaciones:[10]

$$\begin{cases} x+y+z=6 \\ x+2y+5z=12 \\ x+4y+25z=36 \end{cases}$$

Solución propuesta:

```
a=[1,1,1;1,2,5;1,4,25];
b=[6;12,;36];
a\b
```

Figura 4.17: Solución ejercicio 8

Ejercicio 9. Una empresa de muebles fabrica tres modelos de estanterías: A, B y C. En cada uno de los tamaños, grande y pequeño. Produce 1000 estanterías grandes y 8000 pequeñas de tipo A, 8000 grandes y 6000 pequeñas de tipo B, y 4000 grandes y 6000 pequeñas de tipo C. Cada estantería pequeña lleva 12 tornillos y 4 soportes, en cualquiera de los tres modelos.[11]

A. Representar esta información en dos matrices.

B. Hallar una matriz que represente la cantidad de tornillos y de soportes necesarios para la producción diaria de cada uno de los seis modelos-tamaño de estantería.

Solución propuesta:

```
%Apartado A
M=[1000,8000;8000,6000;4000,6000];
N=[16,6;12,4];
%Apartado B
O=M*N
```

```
O =

   112000    38000
   200000    72000
   136000    48000
```

Figura 4.18: Solución ejercicio 9

Ejercicio 10. Resuelve el siguiente sistema:[12]

$$\begin{cases} 3x+2y+z=1 \\ 5x+3y+4z=2 \\ x+y-z=1 \end{cases}$$

Solución propuesta:

```
a=[3,2,1;5,3,4;1,1,-1];
b=[1;2;1];
a\b
```

```
ans =

  -4.00000
   6.00000
   1.00000
```

Figura 4.19: Solución ejercicio 10

Ejercicio 11. Resuelve el siguiente sistema:

$$\begin{cases} 5x+4y+6z=12 \\ 2x+y+z=1 \\ 4x+5y+z=6 \end{cases}$$

Solución propuesta:

```
a=[5,4,6;2,1,1;4,5,1];
b=[12;1;6];
a\b
```

```
ans =

  -1.4167
   1.9583
   1.8750
```

Figura 4.20: Solución ejercicio 11

Ejercicio 12. Resuelve el siguiente sistema:

$$\begin{cases} x + 2y + 4z + 5w = 45 \\ 5x + 4y + 11z + w = 145 \\ 20x + 150y + z + 23w = 500 \\ 10x + y + z + 5w = 69 \\ x + y + z + w = 3 \end{cases}$$

Solución propuesta:

```
a=[1,2,4,5;5,4,11,1;20,150,1,23;10,1,1,5;1,1,1,1];
b=[45;145;500;69;3];
a\b
```

```
ans =

   6.1992
   2.6249
   9.4617
  -1.1840
```

Figura 4.21: Solución ejercicio 12

Capítulo 5

Plot, gráficas de funciones matemáticas

La utilización de gráficos es muy útil para mostrar las ideas de una forma clara y sencilla. Pueden servir para comparar la diferencia entre distintos aparatos hasta la variedad de edad de una patología.

5.1. Gráficas simples

Para este caso vamos a utilizar el algoritmo **plot**, al que añadiremos el título con el algoritmo **title**, cambiaremos el nombre de los ejes x e y con **xlabel** e **ylabel** y pondremos un encuadrado con el algoritmo **grind on**.

```
x=linspace(0.2.*pi,100);
y=sin(x);
plot(x,y);
title('Gráfica simple de seno')
xlabel('Valores de pi')
ylabel('Valores de seno')
```

Figura 5.1: *Código gráfica*

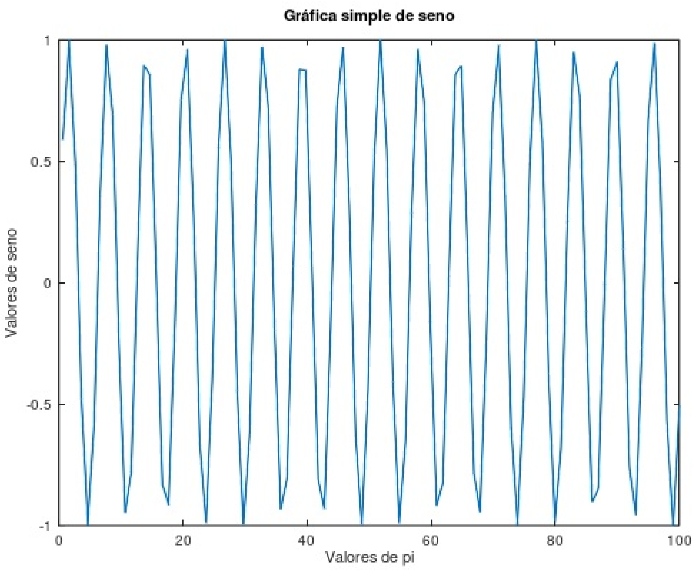


Figura 5.2: *Gráfica seno*

Aquí vemos una gráfica un poco complicada del seno, pero podemos representar funciones sencillas, siempre que le demos antes un valor a x e y. Es muy importante tener en cuenta el algoritmo plot y haber definido bien la función.

```
x=−10:1:10; % Valor de x
plot(x,x.^2,'r') % Representaci n de la grafica
```

```
title('Representaci n de la funci n x^2') %Titulo
xlabel('Valor x') %Nombre del eje de x
ylabel('Valor y') %Nombre del eje de y
```

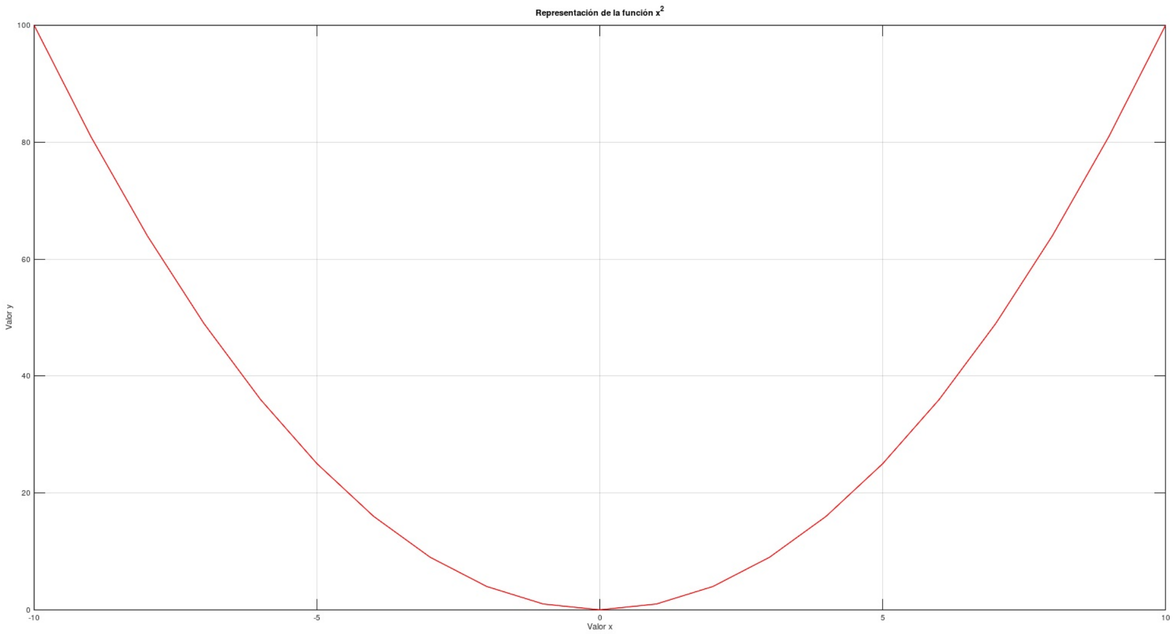


Figura 5.3: Gráfica x^2

Como se puede observar, en la gráfica hecha previamente, primero le damos los valores que queremos a x para poder definir la función que vamos a representar, después con plot le estamos indicando que la represente y después, si queremos, podemos cambiar el nombre al título con el comando title o cambiar de color la línea de la gráfica, de lo que hablaremos más adelante.
Estas dos gráficas son relativamente sencillas de representar, ahora vamos a representar tres tipos de gráficas:

- Polinomio.
- Logarítmica.
- Exponencial.

Función polinómica:

```
x=10:0.1:100;
plot(x.^3+4.*x.^2+x+6,'k');
title('Polinomio')
xlabel('Valor␣de␣x');
ylabel('Valor␣de␣y')
```

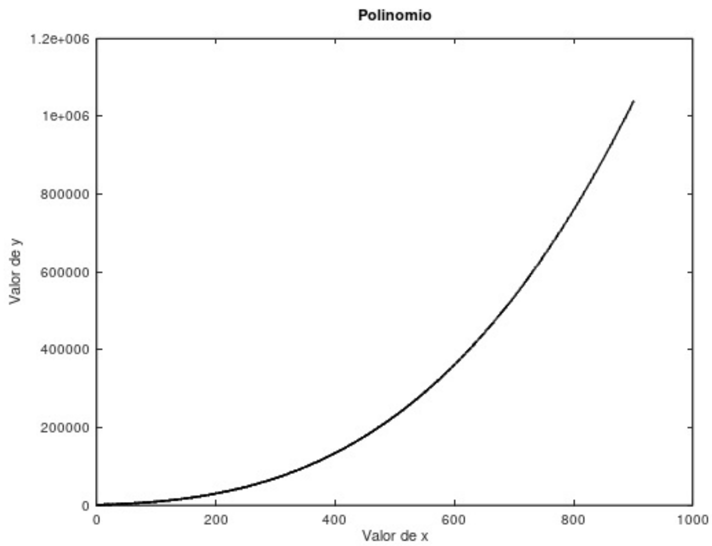


Figura 5.4: Gráfica polinómica

Función logarítmica:

```
x=[-1:1:100];
plot(x,log(x),'g');
title('Logaritmica');
xlabel('Valor x');
xlabel('Valor y')
```

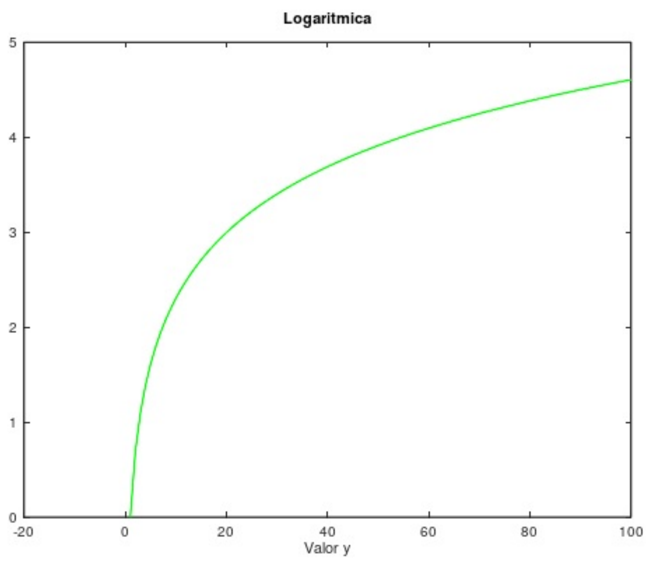


Figura 5.5: Gráfica logarítmica

Función exponencial:

```
x=1:1:10;
plot(x,exp(x));
title('Exponencial')
xlabel('Valor de x');
ylabel('Valor de y')
```

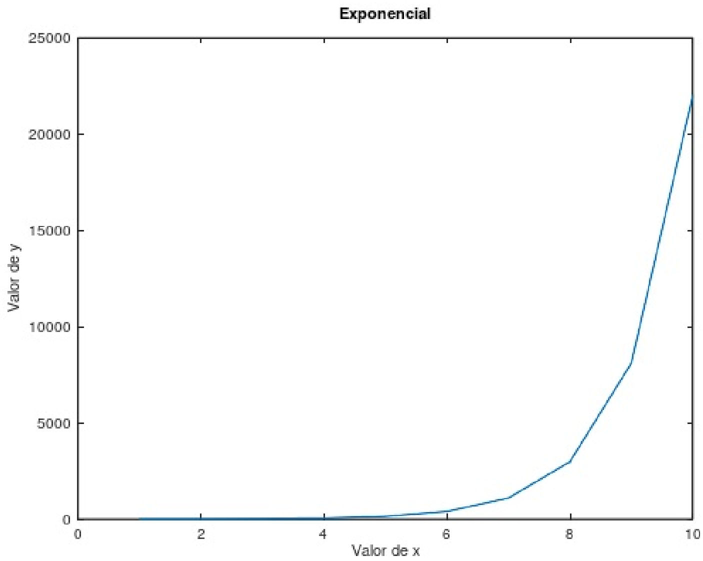


Figura 5.6: Gráfica exponencial

5.2. Gráficas múltiples

Ya hemos visto como poner una gráfica individual de una función, ahora vamos a ver cómo poner varias funciones en una misma gráfica y varias gráficas en una sola figura. Primero voy a poner dos funciones, una de seno y otra de coseno, juntas.[13]

```
x=linspace(pi,10);
y=sin(x); z=cos(x);
plot(x,y,x,z);
title('Grafica de coseno y seno de x');
xlabel('x'), replot
```

Figura 5.7: *Código gráfica*

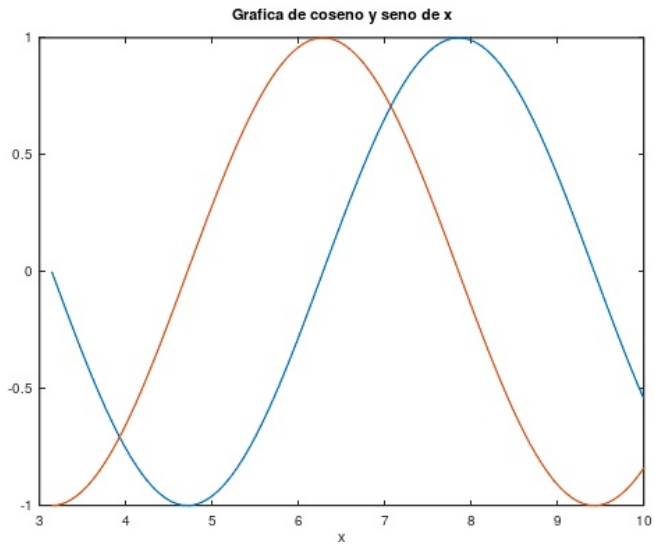


Figura 5.8: *Gráfica seno y coseno*

No es muy complicado de hacer, es muy importante definir las funciones, después con **plot** las enunciamos juntas y con **replot** nos aseguramos que vayan juntas.
También podemos modificar el color o las líneas como será explicado más adelante cada modificador, pero de momento, vamos a ponerle líneas discontinuas a la función anterior.

```
x=linspace(pi,10);
y=sin(x); z=cos(x);
plot(x,y,'-o',x,z,'+');
title('Grafica de coseno y seno de x');
xlabel('x'), replot
```

Figura 5.9: *Código gráfica*

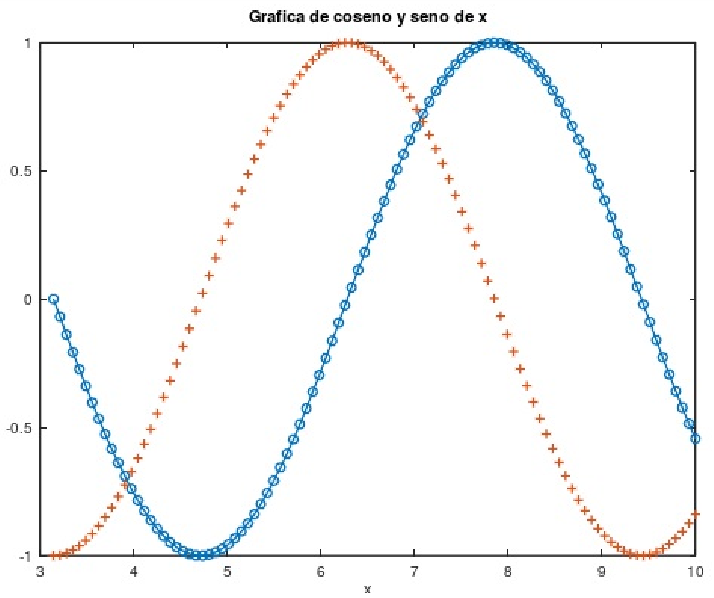


Figura 5.10: *Gráfica seno y coseno*

Después de ver cómo poner varias funciones en una gráfica, vamos a ver cómo poner varias gráficas en una sola figura. Para ello, necesitaremos el algoritmo **subplot** y colocaremos entre paréntesis la posición donde queremos que vaya, por ejemplo, si es una matriz de 2 × 2, tendremos disponibles cuatro huecos de la figura, si la queremos colocar en la posición 1, la indicaremos de la siguiente manera:(2,2,1). Y así seguidamente, hasta que lleguemos a la posición 4.[14]

```
x = 0:.1:4*pi;
y1 = sin(x);
y2 = cos(x);
y1 = 2*sin(x);
y2 = 2*cos(x);

subplot(2,2,1); p1 = plot(x,y1,'Color', 'blue'); title('
subplot(2,2,2); p2 = plot(x,y2,'Color', 'red'); title('c
subplot(2,2,3); p1 = plot(x,y1,'Color', 'blue'); title('
subplot(2,2,4); p2 = plot(x,y2,'Color', 'red'); title('2
```

Figura 5.11: *Código gráfica*

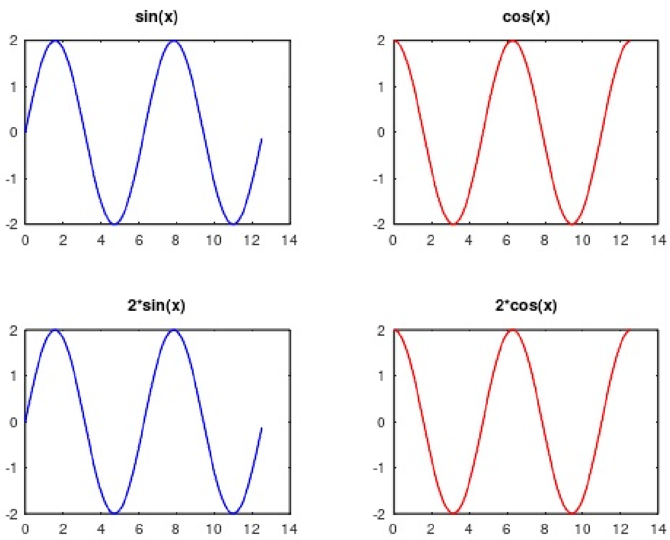


Figura 5.12: *Varias gráficas en una imagen*

5.3. Cambiando el aspecto de las gráficas

Con varios comandos dentro del plot y del subplot podemos cambiar varios aspectos de las gráficas como por ejemplo: el color de las gráficas y el aspecto de las líneas. Voy a colocar una tabla con los comandos de color y cambios de aspecto de las líneas de las gráficas.

Símbolos de marcadores	**Tipo de punto y línea**
-	Segmento
.	Puntos pequeños
x	Aspas
*	Asterisco
+	Cruces
o	Círculo
^	Triángulo
@	Siguiente tipo de punto disponible

Símbolos de marcadores	**Color de la curva**
r	Rojo
g	Verde
b	Azul
m	Magenta
c	Cian
w	Blanco
k	Negro

Una vez que sabemos cómo modificar nuestras gráficas vamos a mostrar un par de ejemplos, una modificando el color de la gráfica y otro cambiando el estilo de las líneas de la gráfica.

```
x=-1:0.1:10;
plot(x,log(x)+x.^2+2.*x,'*');%Asteriscos
```

Figura 5.13: *Código gráfica*

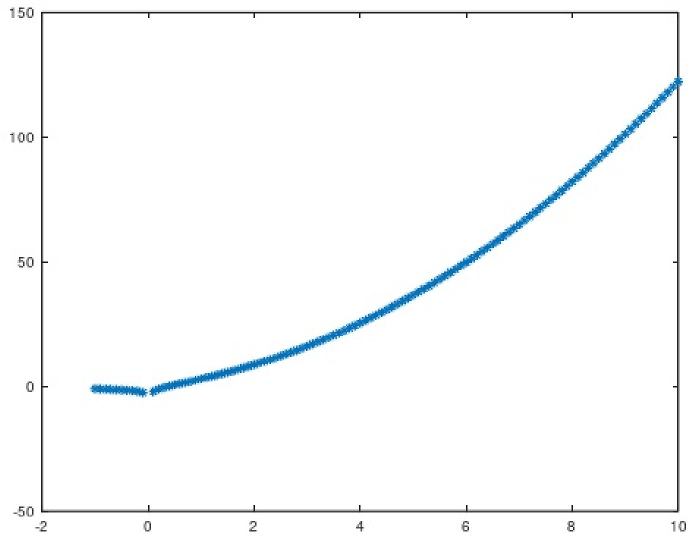


Figura 5.14: *Gráfica con asteriscos*

```
x=2:1:25;
y=cos(x);
z=x.^2+y+5;
plot(x,y,'g',x,z,'c')
```

```
title('Gr fica con color distinto')
replot
```

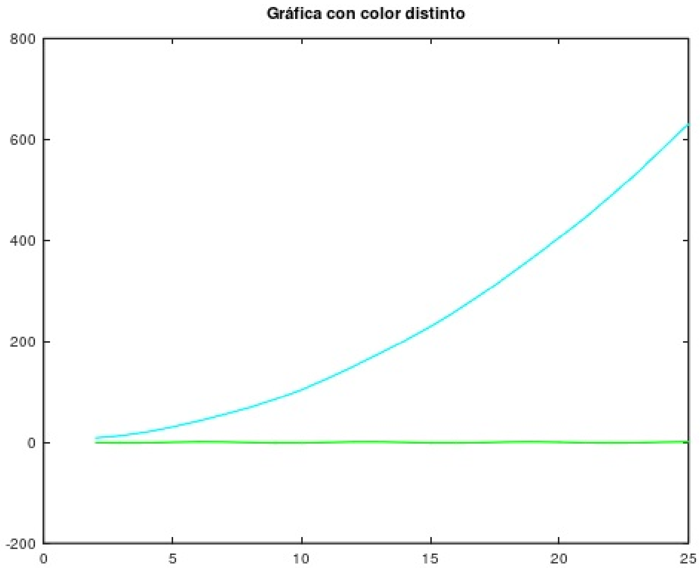


Figura 5.15: *Gráfica de color verde y cian*

5.4. Ejercicios propuestos

Ejercicio 1. Representa gráficamente la función $y = 1 + x$:

Solución propuesta:

```
x=1:0.1:100;
y=1+x;
plot(x,y)
title('Ejercicio 1')
xlabel('Valor de x')
ylabel('Valor de y')
```

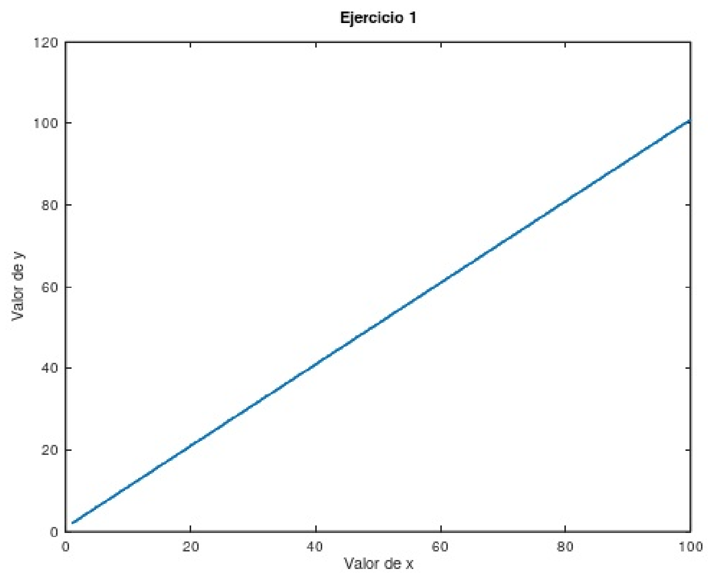


Figura 5.16: Solución ejercicio 1

Ejercicio 2. Representa la función $y = 2x^3 + x^2 + 1$ y resuelve el valor de y si $x = 10$:

SOLUCIÓN PROPUESTA:

```
x=1:0.1:10;
y=2.*x.^3+x.^2+1;
plot(x,y)
title('Grafica')
xlabel('Valor de x')
ylabel('Valor de y')
x1=10;
y1=y*x1;
```

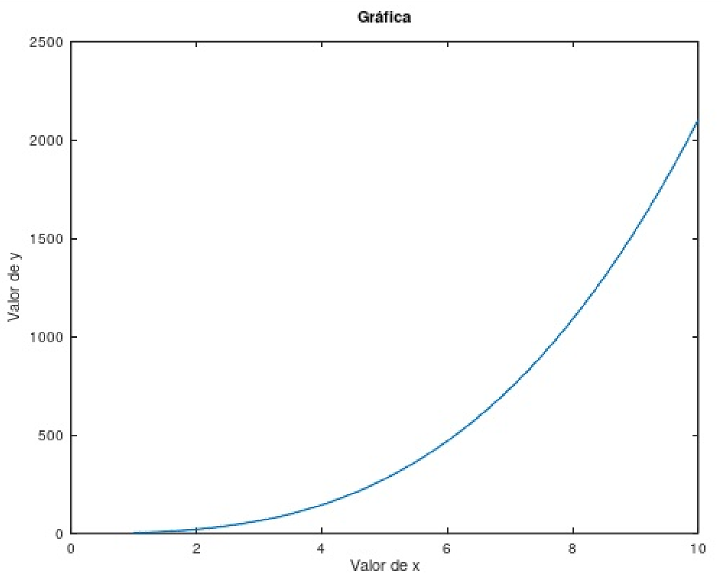


Figura 5.17: Solución ejercicio 2

Ejercicio 3. Representa la función $y = \sin x$ de color verde y con asteriscos dispuestos en la línea:[15]

Solución propuesta:

```
x=0:0.1:3*pi;
y=sin(x);
plot(x,y,'g*')
title('Grafica de seno en verde con asteriscos')
xlabel('Fase(rad)')
ylabel('Amplitud')
```

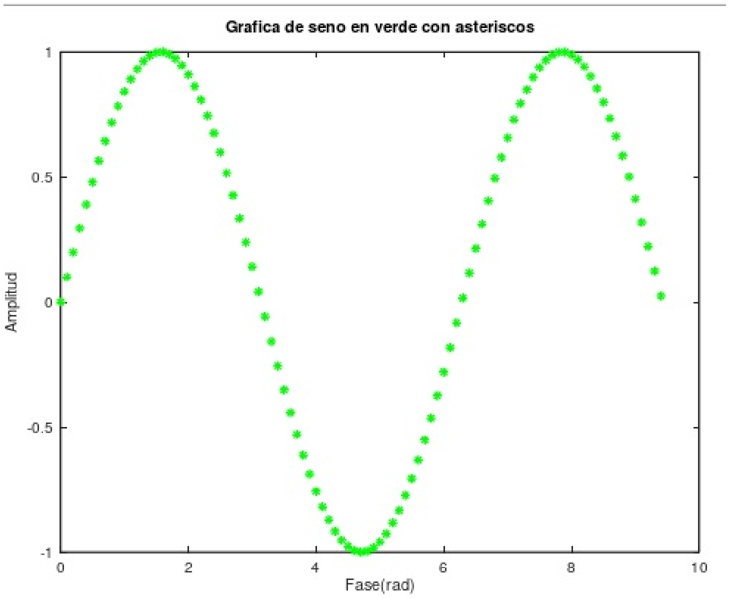


Figura 5.18: Solución ejercicio 3

Ejercicio 4. Representa la función $y = \cos x$ con $z = \operatorname{sen} x$, una de color verde y otra de color rojo.

Solución propuesta:

```
x=0:0.1:3*pi;
y=sin(x);
z=cos(x);
plot(x,y,'g',x,z,'r')
title('Seno y coseno')
xlabel('Fase')
ylabel('Amplitud')
```

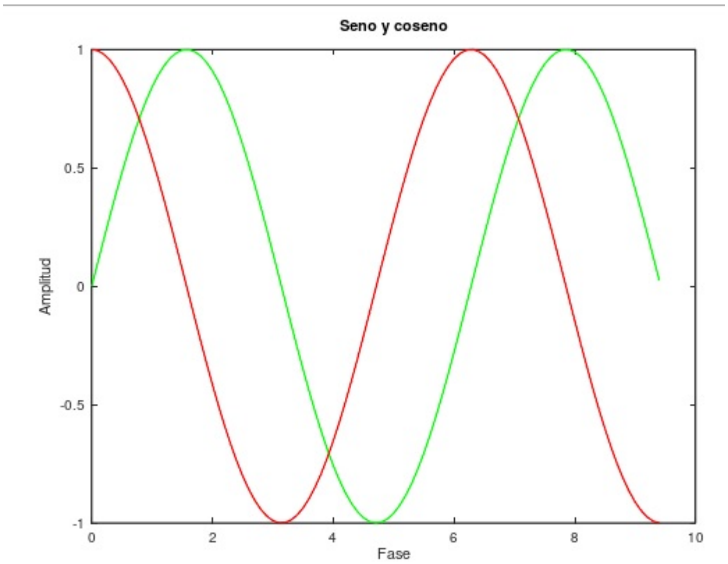


Figura 5.19: Solución ejercicio 4

Ejercicio 5. Compara en cuatro gráficas, el seno, el coseno, la tangente y la cotangente de x:

Solución propuesta:

```
x=0:0.1:3*pi;
y1=sin(x);
y2=cos(x);
y3=tan(x);
y4=cot(x);
subplot(2,2,1); plot(x,y1);
subplot(2,2,2); plot(x,y2);
subplot(2,2,3); plot(x,y3);
subplot(2,2,4); plot(x,y4);
```

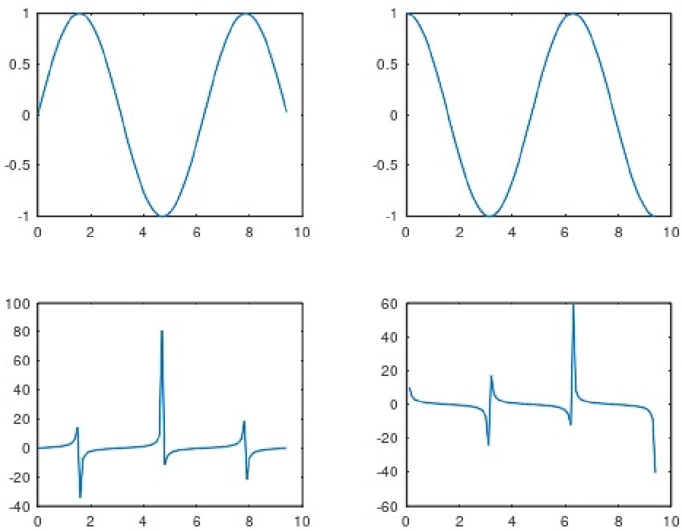


Figura 5.20: Solución ejercicio 5

Ejercicio 6. Representa en una misma gráfica las funciones $y = x^2$ y $z = x^3 + 2$:

Solución propuesta:

```
x=1:0.1:10;
y=x.^2;
z=x.^3+2;
plot(x,y,'b',x,z,'r')
title('Comparaci n de dos funciones');
xlabel('Valor de x')
ylabel('Valor de y')
```

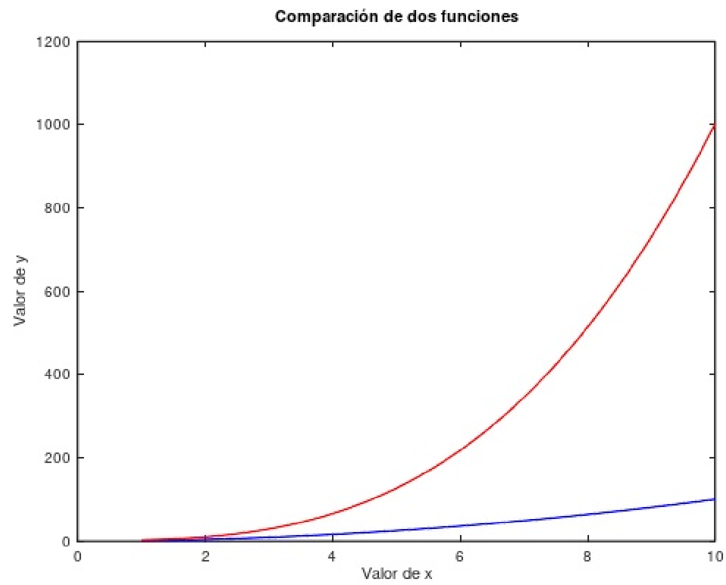


Figura 5.21: Solución ejercicio 6

Ejercicio 7. Calcula el coeficiente de dispersión de Rayleigh a partir de la siguiente fórmula:

$$rs = a \cdot fray \cdot \left(\frac{1}{\frac{landa}{500}} \right)^{-4}$$

$$a = 48; b = 0{,}7; fray = 0{,}41; landa = 400 - 1200nm$$

Solución propuesta:

```
landa=[400:1:1200]; %Longitud de onda
a=48;
b=0.7;
fray=0.41;
%Ecuacion Rayleigh
rs_ray=a.*fray.*(1./(landa/500)).^4;
plot(landa,rs_ray);
title('Rayleigh scattering')
xlabel('Longitud de onda');
ylabel('Rayleigh scattering')
```

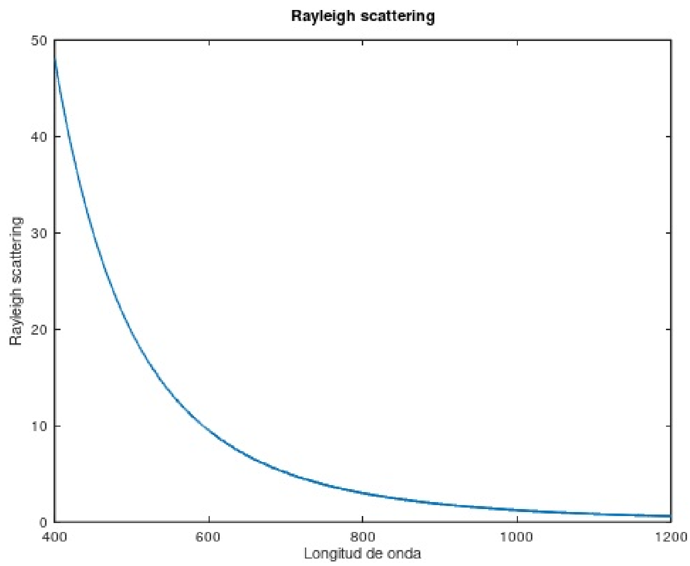


Figura 5.22: Solución ejercicio 7

Ejercicio 8. Calcula el índice de refracción a partir de la siguiente fórmula y comparalo con la longitud de onda propuesta:

$$n = \frac{A+B}{\lambda^2} + \frac{C}{\lambda^4}$$

$$A = 13696; B = 3916{,}8; C = 2558{,}8$$

Solución propuesta:

```
A=13696;
B=3916.8;
C=2558.8;
landa=300:1:1600;
n=(A+B)./(landa.^2)+(C./landa.^4);
plot(landa,n);
title('Indice de refracci n frente a longitud de onda')
xlabel('Longitud de onda')
ylabel('Indice de refraccion')
```

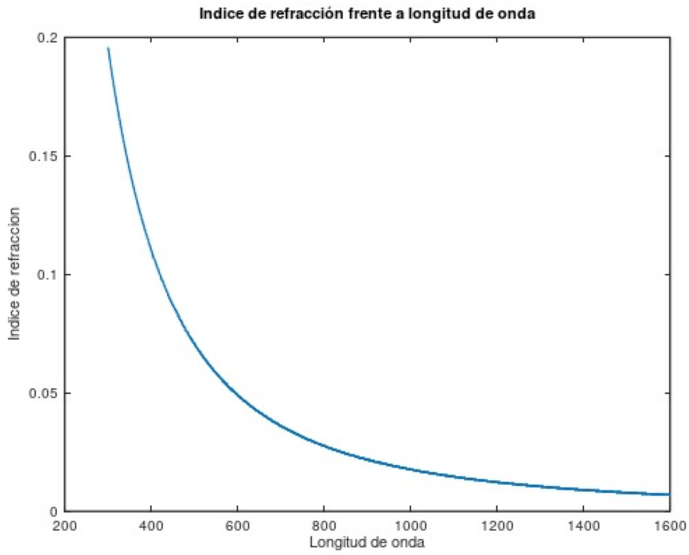


Figura 5.23: Solución ejercicio 8

Ejercicio 9. Representa en la misma gráfica $x = [1, 100]$, $y = x^2 + x^3 + 2x + 6$, $z = \sin x^2$ y $w = \dfrac{\sin x}{\cos x}$:

Solución propuesta:

```
x=1:1:100;
y=x.^2+x.^3+2.*x+6;
z=sin(x.^2);
w=sin(x)./cos(x);
plot(x,y,'r',x,z,'g',x,w,'b')
title('Todas las graficas juntas')
xlabel('Valor de x')
ylabel('Valor de y')
```

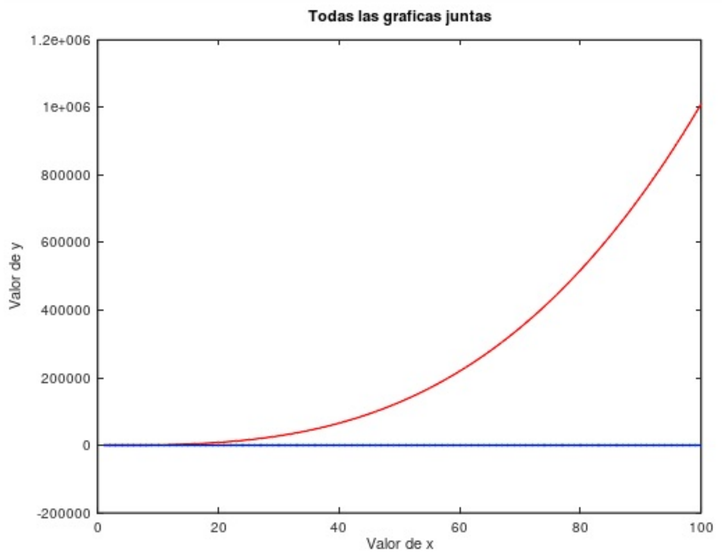


Figura 5.24: Solución ejercicio 9

Ejercicio 10. Representa la función $y = \dfrac{x^2 + x + 1}{x + 1}$:

Solución propuesta:

```
x=1:1:100;
y=(x.^2+x+1)./(x+1);
plot(x,y)
title('Funci n sencilla')
xlabel('Valor de x')
ylabel('Valor de y')
```

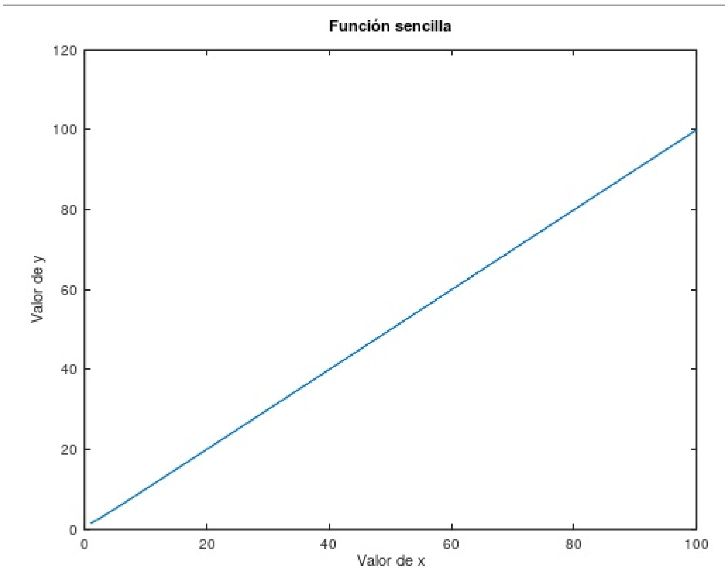


Figura 5.25: Solución ejercicio 10

5.5. Gráficas en 3D

Las gráficas en 3D se utilizan para representar superficies y vólumenes. De este tipo de gráficas sólo se van a mostrar ejemplos, ya que, son un campo muy extenso y complicado. Se pueden dividir en tres tipos:

- Líneas.
- Superficies.
- Contornos.

Líneas:
Las líneas en tres dimensiones se representan por el logaritmo **plot3** y tiene que tener tres variables x, y y z. En este ejemplo vamos a representar:[13]

$$x = e^{-0{,}05t}\sin(t)$$
$$y = e^{-0{,}05t}\cos(t)$$
$$z = t$$

```
t=[0:pi/50:10*pi];
x=exp(-0.05*t).*sin(t);
y=exp(-0.05*t).*cos(t);
z=t
```

```
plot3(x,y,z)
title('Linea')
xlabel('x')
ylabel('y')
zlabel('z')
```

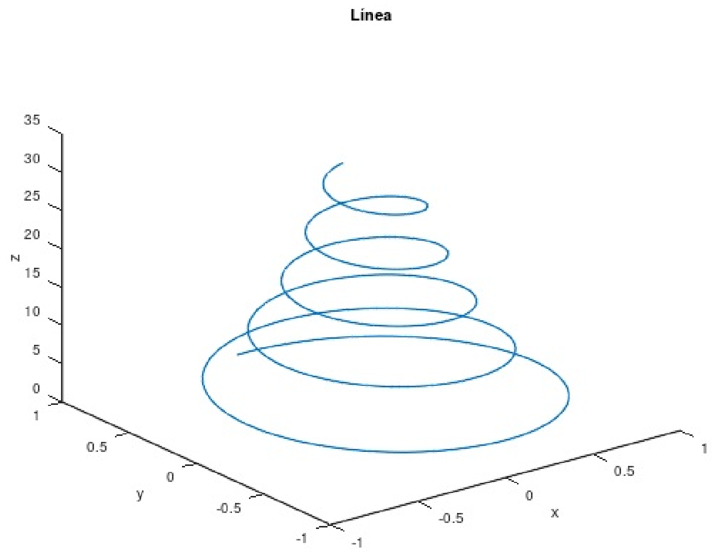


Figura 5.26: Gráfico linea

Superficies:
La función xyz se representa mediante el algoritmo **meshgrid**, donde x e y son los vectores de la variable x. Una forma de ilustrarlo es mediante este sencillo ejemplo:

$$x = xe^{-|(x-y^2)^2+y^2|} \quad \text{para} \quad -2 \leq x \leq 2 \quad \text{y} \quad -2 \leq y \leq 2$$

```
x=[-2:0.1:2],y=x;
[X,Y]=meshgrid(x,y);
Z=X.*exp(-((X-Y.^2).^2+Y.^2));
mesh(X,Y,Z),xlabel('x'),ylabel('y'),zlabel('z')
title('Superficie')
```

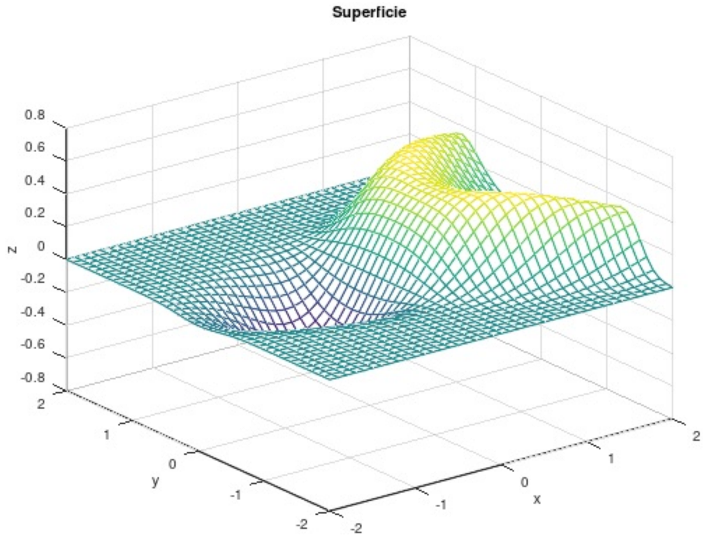


Figura 5.27: Gráfico superficie

Contornos:
Las representaciones topográficas se representan mediante líneas de contorno y son definidas mediante el algoritmo **contour**, que es similar al mesh anteriormente descrito.El ejemplo anterior se puede realizar mediante contorno de la siguiente manera:

```
x=[-2:0.1:2],y=x;
[X,Y]=meshgrid(x,y);
Z=X.*exp(-((X-Y.^2).^2+Y.^2));
contour(X,Y,Z),title('Contorno'),xlabel('x'),ylabel('y'),zlabel('z')
```

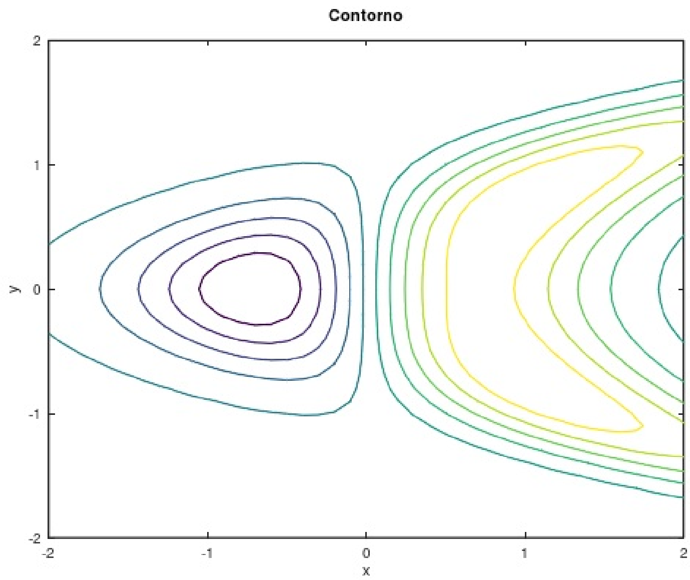


Figura 5.28: Gráfico contorno

Capítulo 6

Gráficos y diagramas

Para la representación de los gráficos y diagramas es muy importante tener en cuenta la media aritmética y la desviación estándar.[16] Primero, vamos a hacer una media de un conjunto de datos:

```
data=[25;50;100;55;77;100;85;200];
mean=(data(:,2:end));
```

Y después de esto, vamos a realizar la desviación estándar:

```
ddata=[50;100;200;800;1000;500];
sigma=std(data(:,2:end));
```

Ya con estas herramientas podemos realizar diagramas de barras, histogramas y rectas de dispersión de datos.[17]
Histogramas:

```
data=3+2*rand(1000,1) %Generamos una serie de numeros aleatorios
figure(1); hist(data) %Histograma con datos aleatorios
figure(2); hist(data,30) %Histograma de 30 clases
[heights,centers]=hist(data,[-2.25:0.5:9.25]);
figure(3); bar(centers,heights)
```

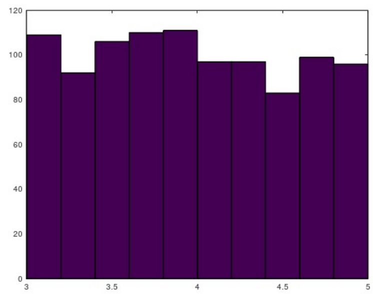


Figura 6.1: *Histograma 1*

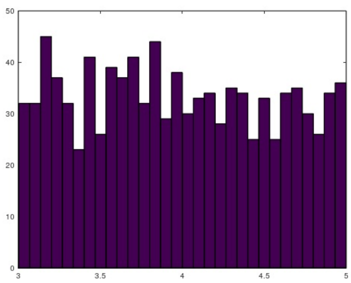


Figura 6.2: *Histograma 2*

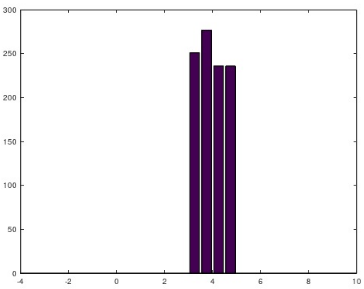


Figura 6.3: *Histograma 3*

Diagrama de barras:
Al igual que podemos realizar histogramas, podemos realizar un diagrama de barras para representar una serie de datos. Para realizarlo, es necesario el algoritmo **bar**.

```
ages=20:27; students=[2;1;4;3;2;2;0;1];
figure(1); bar(ages,students);
title('Diagrama de barras')
xlabel('edad de estudiantes'); ylabel('numero de estudiantes')
```

```
figure(2); barh(ages,students);
title('Diagrama de barras')
ylabel('numero de estudiantes'); xlabel('edad de estudiantes')
```

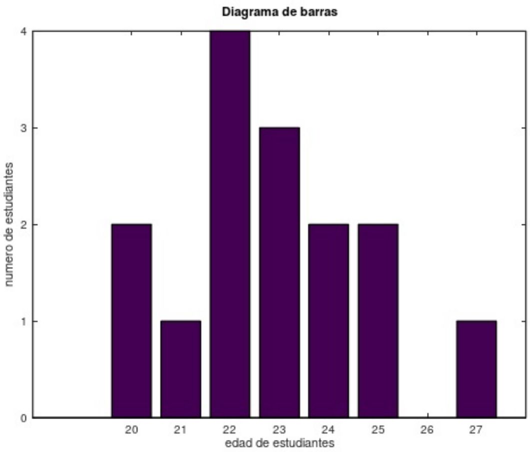


Figura 6.4: *Diagrama de barras 1*

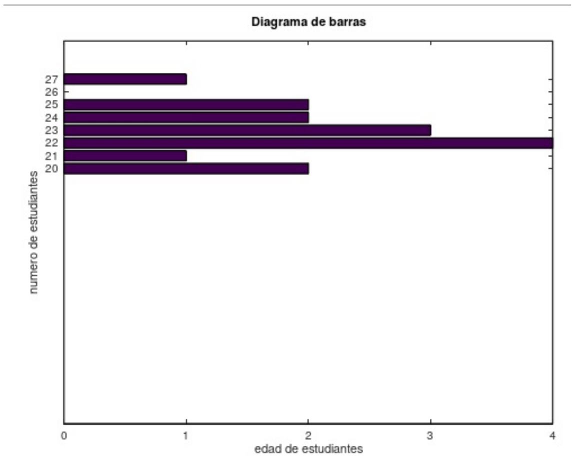


Figura 6.5: *Diagrama de barras 2*

Gráfico circular:
Como hemos visto anteriormente, podemos mostrar nuestros datos estadísticos de manera que queramos, en este caso de manera de un gráfico circular. Para este tipo de gráfico se utilizarán los algoritmos **pie** y **pie3**, en el caso de éste último se presentará en un gráfico 3D.

```
strength=[55;52;36;28;13;16];
Labels={'SVP','SP','FDP','CVP','GR','Div'}
figure(1); pie(strength)
figure(2); pie(strength,[0;1;0;0;0;0],Labels)
figure(3); pie3(strength,Labels)
```

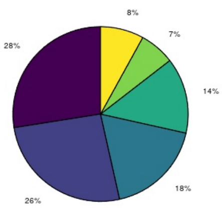


Figura 6.6: *Gráfico circular 1*

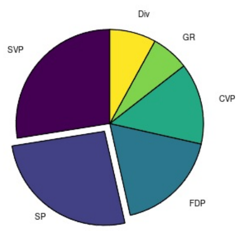


Figura 6.7: *Gráfico circular 2*

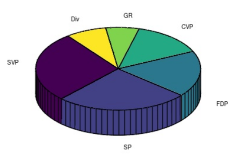


Figura 6.8: *Gráfico circular 3*

6.1. Regresión lineal

La regresión lineal es esencial para ver una serie de datos, es muy utilizada en los distintos usuarios de patologías y diferentes distribuciones de medidas. Es muy importante tener en cuenta la ecuación de la recta de regresión lineal:

$$y = mx + b$$

```
X=[1,2,3]; %valores del eje X
Y=[2.1,4.1,5.9]; %valores del eje Y
N=3; %numero de puntos del ajuste


%Calculo de la penmdiente y la ordenada en el origen con su error
xm=mean(X);ym=mean(Y);
sumxx=sum((X-xm).^2);sumxy=sum((X-xm).*(Y-ym));
sumyy=sum((Y-ym).^2);
m=sumxy/sumxx
b=ym-m*xm
d=Y-m*X-b;
Dm= sqrt(sum(d.^2)/(sumxx*(N-2)))
Db= sqrt((1/N+xm^2/sumxx)*sum(d.^2)/(N-2))
r=sumxy/(sqrt(sumxx*sumyy))

%funcion ajuste
yfit=m*X+b;
plot(X,Y,'ro','Markersize',5);hold on
plot(X,yfit,'k-','linewidth',1);
xlabel('Concentraci n (M)');
ylabel('Absorbancia');

%rango representaci n ejes
minX=min(X);maxX=max(X);deltaX=(maxX-minX)/100;
minY=min(Y);maxY=max(Y);deltaY=(maxY-minY)/100;
xlim([minX-2*deltaX maxX+2*deltaX]);ylim([minY-2*deltaY maxY+
2*deltaY]);
```

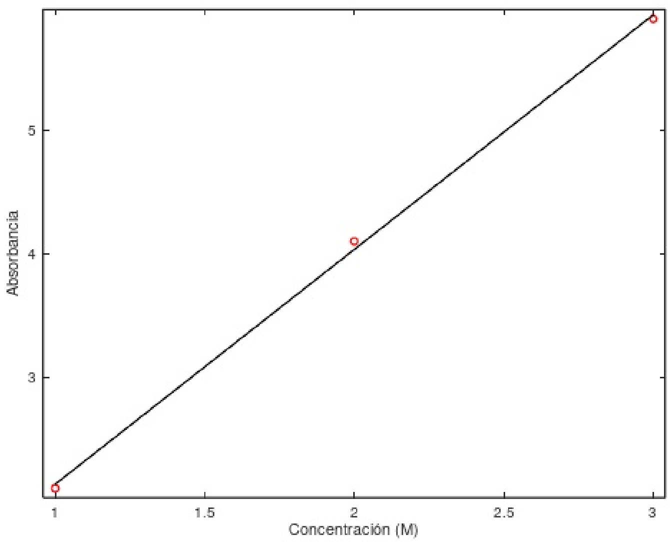


Figura 6.9: *Recta de regresión*

6.2. Abrir y guardar archivos .mat

Es muy importante, antes de mostrar el algoritmo de salvar y guardar, conocer el directorio donde está guardado el archivo el cual queremos abrir y guardar.

```
%Cargar datos
load datos(1).mat
%Guardar datos
save datos(1).mat
```

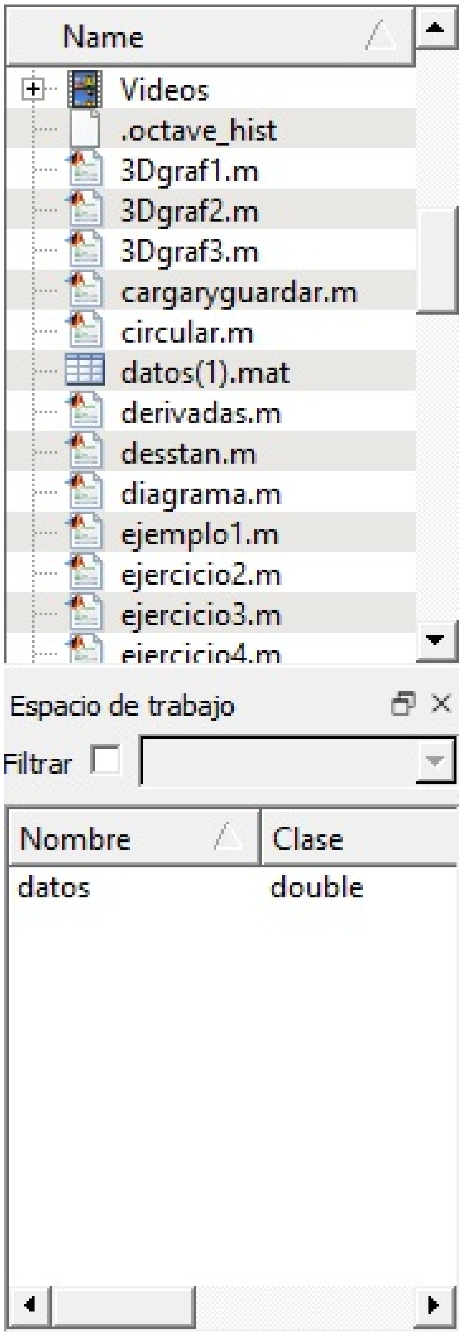
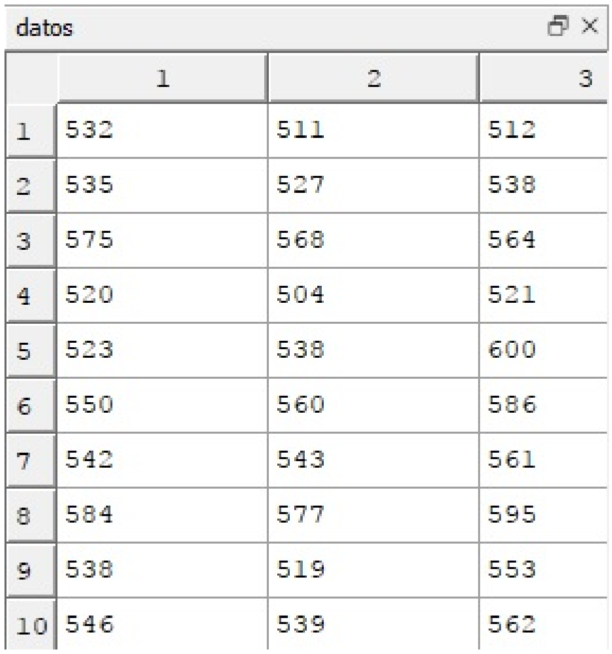


Figura 6.10: *Datos ya guardados*


datos

	1	2	3
1	532	511	512
2	535	527	538
3	575	568	564
4	520	504	521
5	523	538	600
6	550	560	586
7	542	543	561
8	584	577	595
9	538	519	553
10	546	539	562

Figura 6.11: *Contenido datos*

Los archivos .mat son característicos de Matlab, con este sencillo algoritmo podremos guardar y cargar estos archivos de Matlab a Octave. Si no estuviera en la misma carpeta que los datos de Octave, a la hora de cargarlos habría que poner el directorio donde están para que los cargue desde allí.

Bibliografía

[1] About. https://www.gnu.org/software/octave/about.html.

[2] Semejanzas y diferencias con matlab. http://nereida.deioc.ull.es/~pcgull/ihiu01/cdrom/matlab/contenido/node77.html.

[3] Octave. https://torroja.dmt.upm.es/media/files/paper_logrono.pdf.

[4] Matlab vs octave. https://torroja.dmt.upm.es/media/files/paper_logrono.pdf.

[5] Cálculo elemental. https://octave.org/doc/interpreter/Simple-Examples.html#Elementary-Calculations.

[6] Diferencial. https://cimec.org.ar/foswiki/pub/Main/Cimec/MetodosNumericosYSimulacion/metnums.pdf.

[7] Segundo. https://gradocienciasdelmar.files.wordpress.com/2012/03/practica1-resuelta.pdf.

[8] Integrales. https://octave.org/doc/v4.0.1/Derivatives-_002f-Integrals-_002f-Transforms.html.

[9] Siete. https://www.vitutor.com/algebra/matrices/ma_7.htmll.

[10] Ocho. https://www.vitutor.com/algebra/matrices/ma_10.html.

[11] Nueve. https://www.vitutor.com/algebra/matrices/me_5.html.

[12] Diez. https://www.vitutor.com/ecuaciones/2/g_e1.html.

[13] Multiples. http://repositori.uji.es/xmlui/bitstream/handle/10234/5956/3_Graficos.pdf;jsessionid=727D79A2CFA321172C1A04A3BFDD7E2D?sequence=1.

[14] Varias. https://mecatronicauaslp.wordpress.com/2014/04/22/graficas-2d-en-matlaboctave/.

[15] Severde. http://www.dtic.upf.edu/~msordo/FMIV/material/Octave.pdf.

[16] Media. http://www.mttmllr.com/computaller/beamers/polinomios%20y%20estadistica.pdf.

[17] Histogramas. https://web.ti.bfh.ch/~sha1/StatisticsWithMatlabOctave.pdf.