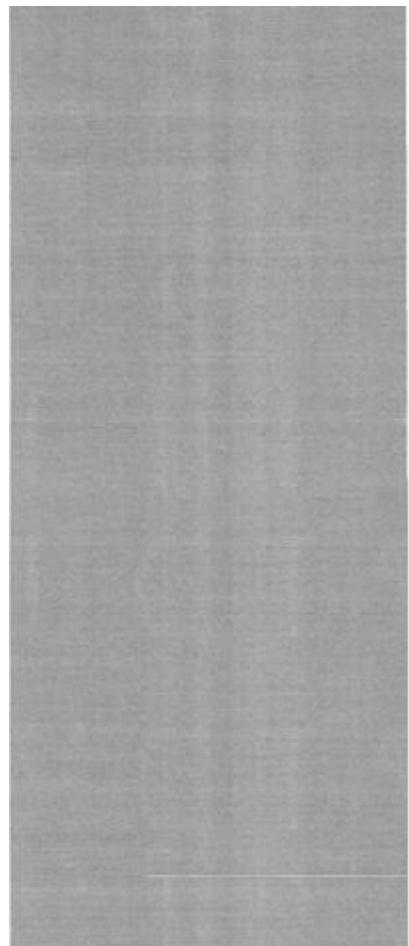


# MATLAB

## a través de ejemplos

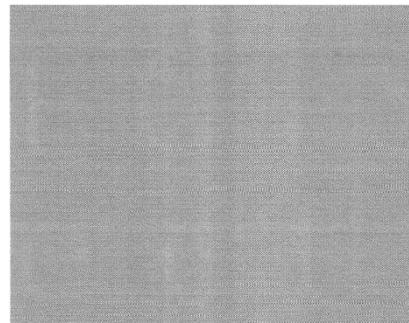
César Pérez

**César Pérez López**  
*Instituto de Estudios Fiscales (IEF)*  
*Universidad Complutense de Madrid*



# MATLAB a través de ejemplos

*Garceta*  
grupo editorial



## MATLAB a través de ejemplos

### César Pérez López

ISBN: 978-84-9281-243-1  
IBERGARCETA PUBLICACIONES, S.L., Madrid 2011

Edición: 1.<sup>a</sup>  
Impresión: 1.<sup>a</sup>  
N.<sup>o</sup> de páginas: 444  
Formato: 17 x 24 cm  
Materia CDU: 51 Matemáticas.

Reservados los derechos para todos los países de lengua española. De conformidad con lo dispuesto en el artículo 270 y siguientes del código penal vigente, podrán ser castigados con penas de multa y privación de libertad quienes reprodujeren o plagiaren, en todo o en parte, una obra literaria, artística o científica fijada en cualquier tipo de soporte sin la preceptiva autorización. Ninguna parte de esta publicación, incluido el diseño de la cubierta, puede ser reproducida, almacenada o trasmisida de ninguna forma, ni por ningún medio, sea éste electrónico, químico, mecánico, electro-óptico, grabación, fotocopia o cualquier otro, sin la previa autorización escrita por parte de la editorial.

Diríjase a CEDRO (Centro Español de Derechos Reprográficos), [www.cedro.org](http://www.cedro.org), si necesita fotocopiar o escanear algún fragmento de esta obra.

COPYRIGHT © 2011 IBERGARCETA PUBLICACIONES, S.L.  
[info@garceta.es](mailto:info@garceta.es)

MATLAB a través de ejemplos

### César Pérez López

1<sup>a</sup> edición, 1<sup>a</sup> impresión  
OI: 70-2011  
ISBN: 978-84-9281-243-1  
Deposito Legal: M-26260-2011

Impresión: Reprográficas Malpe, S.A.

IMPRESO EN ESPAÑA - PRINTED IN SPAIN

Nota sobre enlaces a páginas web ajena: Este libro puede incluir referencias a sitios web gestionados por terceros y ajenos a IBERGARCETA PUBLICACIONES, S.L., que se incluyen sólo con finalidad informativa. IBERGARCETA PUBLICACIONES, S.L., no asume ningún tipo de responsabilidad por los daños y perjuicios derivados del uso de los datos personales que pueda hacer un tercero encargado del mantenimiento de las páginas web ajena a IBERGARCETA PUBLICACIONES, S.L., y del funcionamiento, accesibilidad y mantenimiento de los sitios web no gestionados por IBERGARCETA PUBLICACIONES, S.L., directamente. Las referencias se proporcionan en el estado en que se encuentran en el momento de publicación sin garantías, expresas o implícitas sobre la información que se proporcione en ellas.

# INDICE

---

<b><i>Capítulo 1. Introducción y entorno de trabajo de MA TLAB .....</i></b>	<b>1</b>
Introducción a MATLAB .....	1
Desarrollo de algoritmos y aplicaciones .....	3
Análisis y acceso a datos .....	5
Visualización de datos .....	7
Cálculo numérico .....	10
Publicación de resultados y distribución de aplicaciones .....	10
El entorno de trabajo de MATLAB .....	12
Ayuda en MATLAB .....	16
 <b><i>Capítulo 2. Números, variables, operadores y funciones .....</i></b>	 <b>21</b>
Números, operaciones aritméticas y formatos .....	21
Números enteros y funciones de variable entera .....	24
Números reales y funciones de variable real .....	26
Funciones trigonométricas.....	29
Funciones hiperbólicas .....	29
Funciones exponenciales y logarítmicas .....	30
Funciones específicas de variable numérica.....	31
Variables unidimensionales, vectoriales y matriciales .....	32
Elementos de variables vectoriales .....	35
Elementos de variables matriciales .....	36
Funciones matriciales específicas .....	39
Números aleatorios .....	42
Operadores .....	44
Operadores aritméticos .....	44

Operadores lógicos .....	46
Operadores relacionales .....	47
Variables simbólicas .....	48
Funciones simbólicas y operaciones funcionales. Funciones compuesta e inversa.....	50
Comandos que manejan variables en el espacio de trabajo y las almacenan en ficheros .....	53
<b>Capítulo 3. Números complejos y funciones de variable compleja .....</b>	<b>79</b>
Números complejos .....	79
Funciones generales de variable compleja.....	80
Funciones trigonométricas de variable compleja .....	80
Funciones hiperbólicas de variable compleja .....	81
Funciones exponenciales y logarítmicas de variable compleja .....	82
Funciones específicas de variable compleja .....	83
Funciones básicas con argumento un vector complejo .....	84
Funciones básicas con argumento una matriz compleja.....	90
Funciones generales con argumento una matriz compleja .....	95
Funciones trigonométricas de variable matriz compleja .....	95
Funciones hiperbólicas de variable matriz compleja.....	99
Funciones exponenciales y logarítmicas de variable matriz compleja.....	102
Funciones específicas de variable matricial compleja.....	104
Operaciones con variables matriciales reales y complejas .....	107
<b>Capítulo 4. Gráficos .....</b>	<b>125</b>
Gráficos exploratorios de datos.....	125
Curvas en coordenadas explícitas, implícitas, paramétricas y polares .....	132
Curvas alabeadas .....	141
Superficies explícitas y paramétricas. Curvas de nivel .....	143
Formas geométricas tridimensionales.....	150
Gráficos especializados.....	153
Opciones de gráficos 2D y 3D .....	159
<b>Capítulo 5. Expresiones algebraicas, polinomios, ecuaciones y sistemas .....</b>	<b>177</b>
Expansión, simplificación y factorización de expresiones algebraicas .....	177
Polinomios.....	180
Interpolación polinómica .....	185
Resolución de ecuaciones y sistemas .....	191
Métodos generales .....	191
Método de los gradientes biconjugados .....	193

---

Método de los gradientes conjugados .....	196
Método residual.....	197
Método simétrico y de mínimos cuadrados no negativos .....	200
Resolución de ecuaciones y sistemas lineales .....	201
<b><i>Capítulo 6. Matrices, espacios vectoriales y aplicaciones lineales</i></b> .....	<b>231</b>
Operaciones con matrices .....	232
Descomposición de matrices. Valores propios y vectores propios .....	240
Espacios vectoriales, aplicaciones lineales y formas cuadráticas .....	251
<b><i>Capítulo 7. Límites de sucesiones y funciones. Continuidad.</i></b>	
<b><i>Una y varias variables</i></b> .....	<b>281</b>
Límites .....	281
Sucesiones de funciones .....	284
Continuidad.....	286
Límites en varias variables. Límites iterados y direccionales .....	289
Continuidad en varias variables.....	293
<b><i>Capítulo 8. Series numéricas, series de potencias y desarrollos en serie</i></b> .....	<b>305</b>
Series numéricas de términos no negativos .....	305
Criterios de convergencia: criterio del cociente .....	306
Criterio de Raabe .....	307
Criterio de la raíz.....	309
Otros criterios de convergencia .....	311
Series numéricas alternadas. Criterios de Dirichlet y Abel.....	313
Series de potencias .....	314
Desarrollos en series de potencias .....	316
<b><i>Capítulo 9. Derivación en una y varias variables. Aplicaciones</i></b> .....	<b>327</b>
Derivadas .....	327
Derivadas parciales .....	330
Aplicaciones de las derivadas. Tangentes, asíntotas, extremos y puntos de inflexión .....	331
Derivabilidad en varias variables.....	336
Extremos en varias variables .....	341
Extremos condicionados. El método de “los multiplicadores de Lagrange”.....	349
Cálculo diferencial vectorial.....	352

Teorema de la función compuesta .....	354
Teorema de la función implícita.....	355
Teorema de la función inversa .....	356
Teorema del cambio de variable.....	358
Desarrollos en serie para varias variables .....	359
Rotacional, divergencia y laplaciano.....	360
Coordenadas cilíndricas, esféricas y rectangulares.....	363
 <i>Capítulo 10. Integración en una y varias variables. Aplicaciones</i> .....	 385
Integrales.....	385
Integrales inmediatas, por cambio de variable y por partes .....	387
Integrales por reducción y cíclicas.....	389
Integrales racionales, irracionales y binomias .....	390
Integral definida y aplicaciones. Una variable .....	391
Longitud de un arco de curva .....	391
Superficie entre dos curvas .....	395
Superficies de revolución .....	400
Volúmenes de revolución.....	401
Integrales curvilíneas .....	403
Integrales impropias .....	404
Integrales dependientes de un parámetro .....	405
Integración numérica aproximada.....	407
Integrales especiales .....	409
Integral definida y aplicaciones. Varias variables .....	412
Área de figuras planas y doble integración .....	413
Área de superficies por doble integración .....	415
Cálculo de volúmenes por integrales dobles .....	416
Cálculo de volúmenes e integrales triples.....	417
El teorema de Green .....	419
Teorema de la divergencia.....	420
El teorema de Stokes .....	421

## CAPITULO 1

# INTRODUCCION Y ENTORNO DE TRABAJO DE MATLAB

## INTRODUCCIÓN A MATLAB

MATLAB es una plataforma de cálculo científico y programación de alto nivel a través de un entorno interactivo que permite realizar tareas de cálculo complejas de forma más rápida que con los lenguajes de programación tradicionales, como C, C++ y Fortran. Se trata de la plataforma de cálculo más utilizada actualmente en las ciencias y la ingeniería.

MATLAB es un lenguaje de computación técnica de alto nivel y un entorno interactivo para desarrollo de algoritmos, visualización de datos, análisis de datos y cálculo numérico. MATLAB es adecuado para resolver problemas de cálculo técnico mediante algoritmos optimizados que se incorporan en comandos de fácil uso.

Es posible usar MATLAB en una amplia gama de aplicaciones que incluyen cálculo matemático, álgebra, estadística, econometría, control de calidad, series temporales, procesamiento de señales e imágenes, comunicaciones, diseño de sistemas de control, sistemas de prueba y medición, modelado y análisis financiero, biología computacional, etc. Los conjuntos de herramientas complementarios llamados *toolbox* (colecciones de funciones de MATLAB para propósitos especiales, que están disponibles por separado) amplían el entorno de MATLAB permitiendo resolver problemas especiales en diferentes áreas de aplicación.

Además, MATLAB contiene una serie de funciones para documentar y compartir el trabajo. Es posible integrar código de MATLAB con otros lenguajes y aplicaciones, y distribuir los algoritmos y aplicaciones que se desarrollen usando MATLAB.

Son características importantes de MATLAB las siguientes:

- Lenguaje de alto nivel para cálculo técnico
- Entorno de desarrollo para la gestión de código, archivos y datos
- Herramientas interactivas para exploración, diseño y resolución de problemas iterativos
- Funciones matemáticas para álgebra lineal, estadística, análisis de Fourier, filtraje, optimización e integración numérica
- Funciones gráficas bidimensionales y tridimensionales para visualización de datos
- Herramientas para crear interfaces gráficas de usuario personalizadas
- Funciones para integrar los algoritmos basados en MATLAB con aplicaciones y lenguajes externos, tales como C/C++, FORTRAN, Java, COM y Microsoft Excel

El entorno de desarrollo MATLAB permite desarrollar algoritmos y analizar datos, visualizar archivos de datos y gestionar proyectos de modo interactivo (Figura 1-1).

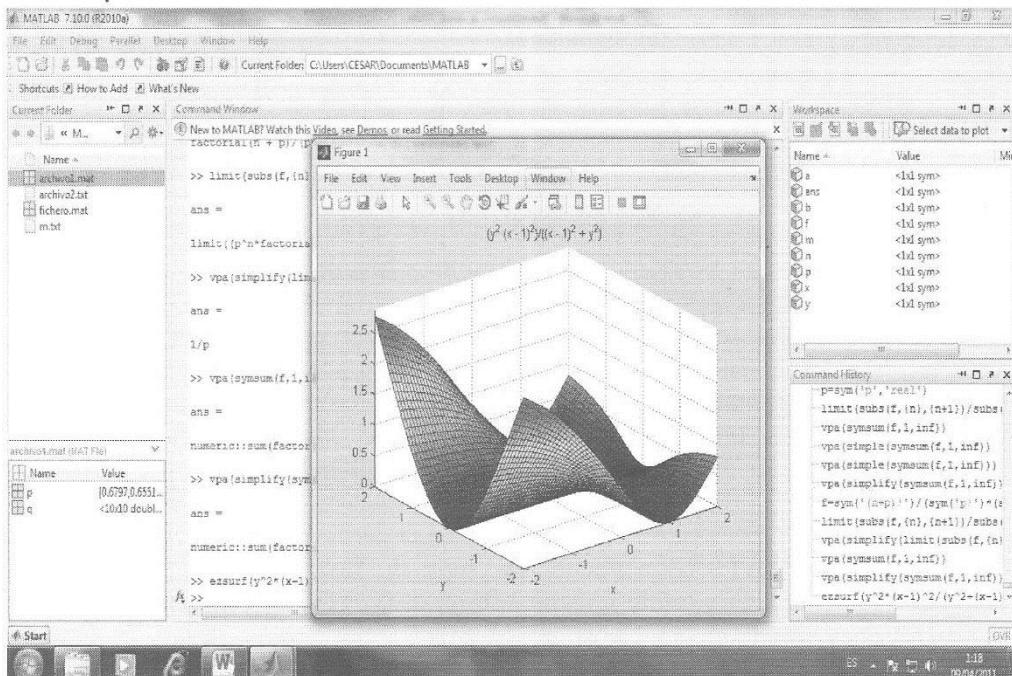


Figura 1-1

## Desarrollo de algoritmos y aplicaciones

MATLAB proporciona un lenguaje de alto nivel y herramientas de desarrollo con los que es posible desarrollar y analizar algoritmos y aplicaciones rápidamente.

El lenguaje MATLAB incluye operaciones vectoriales y matriciales que son fundamentales para resolver los problemas científicos y de ingeniería. Agiliza, tanto el desarrollo, como la ejecución.

Con el lenguaje de MATLAB, es posible programar y desarrollar algoritmos más rápidamente que con los lenguajes tradicionales porque ya no es necesario realizar tareas administrativas de bajo nivel, tales como declarar variables, especificar tipos de datos y asignar memoria. En muchos casos, MATLAB elimina la necesidad de bucles "for". En consecuencia, una línea de código de MATLAB generalmente reemplaza a varias líneas de código C o C++.

Al mismo tiempo, MATLAB ofrece todas las características de los lenguajes de programación tradicionales, que incluyen operadores aritméticos, control de flujo, estructuras de datos, tipos de datos, programación orientada a objetos (OOP) y depuración.

En la Figura 1-2 observamos un algoritmo de modulación de comunicaciones que genera 1.024 bits aleatorios, realiza la modulación, añade ruido gaussiano complejo y representa gráficamente el resultado. Todo en sólo 9 líneas de código de MATLAB.

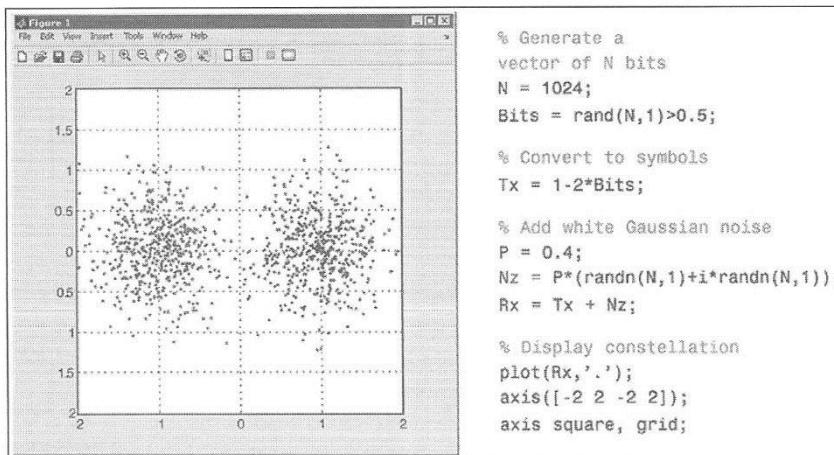


Figura 1-2

MATLAB posibilita ejecutar comandos o grupos de comandos uno a uno, sin compilar ni enlazar, y repetir su ejecución hasta lograr la solución óptima.

Para ejecutar rápidamente cálculos matriciales y vectoriales complejos, MATLAB utiliza bibliotecas optimizadas para el procesador. Para cálculos escalares de aplicación general, MATLAB genera instrucciones en código máquina utilizando su tecnología JIT (*Just-In-Time*). Gracias a esta tecnología, que está disponible para la mayoría de las plataformas, las velocidades de ejecución son mucho más rápidas que las de los lenguajes de programación tradicionales.

MATLAB incluye *herramientas de desarrollo* que ayudan a implementar algoritmos eficientemente. Las siguientes son algunas de ellas:

- **MATLAB Editor** - Funciones de edición y depuración estándar, como establecimiento de puntos de interrupción y simulaciones paso a paso
- **M-Lint Code Checker** - Analiza el código y recomienda modificaciones para mejorar el rendimiento y mantenimiento (Figura 1-3)
- **MATLAB Profiler** - Registra el tiempo que tarda en ejecutarse cada línea de código
- **Directory Reports** - Explora todos los archivos de un directorio y crea informes sobre la eficiencia del código, las diferencias entre los archivos, las dependencias de los archivos y la cobertura del código

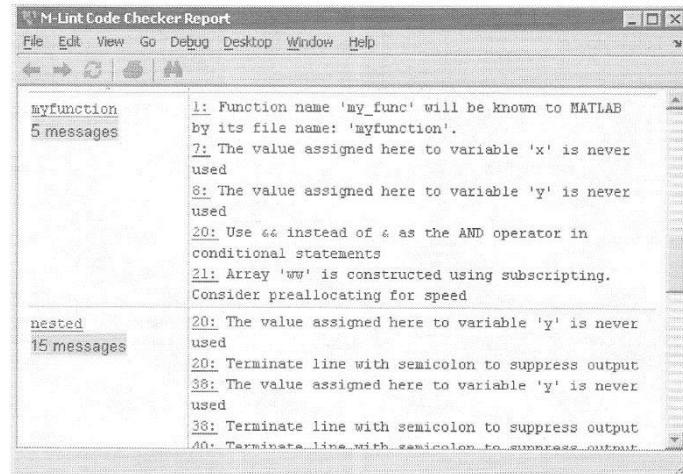


Figura 1-3

También se puede usar la herramienta interactiva GUIDE (*Graphical User Interface Development Environment*) para diseñar y editar interfaces de usuario. Esta herramienta permite incluir listas de selección, menús desplegables, botones de pulsación, botones de opción y deslizadores, así como diagramas de MATLAB y controles ActiveX. También se pueden crear interfaces gráficas de usuario por medio de programación usando las funciones de MATLAB.

En la Figura 1-4 se observa la interfaz de usuario de análisis de ondículas en la herramienta GUIDE (arriba) e interfaz terminada (abajo).

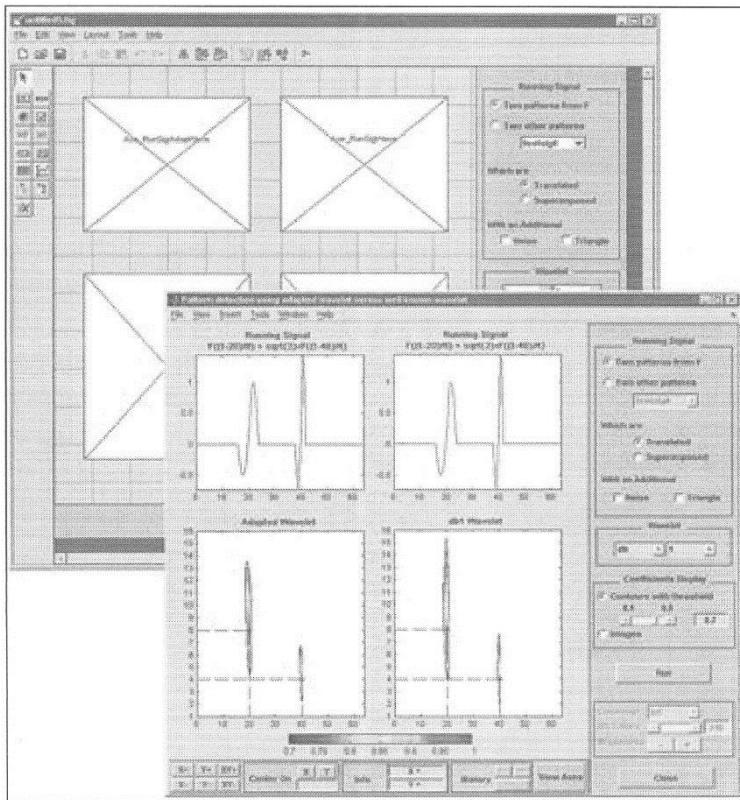


Figura 1-4

### **Análisis y acceso a datos**

MATLAB soporta todo el proceso de análisis de datos, desde la adquisición de datos de dispositivos externos y bases de datos, pasando por el preprocessado, visualización y análisis numérico, hasta la producción de resultados con calidad de presentación.

MATLAB proporciona herramientas interactivas y funciones de línea de comandos para operaciones de análisis de datos, entre las que se incluyen: interpolación, extracción de secciones de datos, escalado y promediación, umbralización y suavizado, correlación, análisis de Fourier y filtraje, búsqueda de picos unidimensionales y ceros, estadística básica y ajuste de curvas, análisis matricial, etc.

En la Figura 1-5 se observa un diagrama en el que se muestra una curva ajustada a las diferencias de presión atmosférica promediadas entre Easter Island y Darwin en Australia.

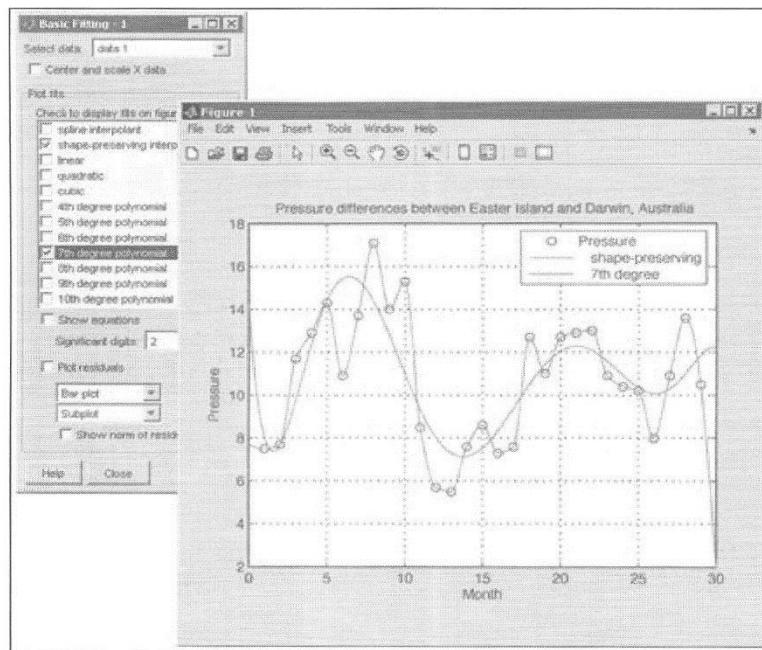


Figura 1-5

En cuanto a acceso a datos, MATLAB es una plataforma eficiente para acceder a datos de archivos, otras aplicaciones, bases de datos y dispositivos externos. Puede leer datos guardados en los formatos más conocidos, tales como Microsoft Excel, archivos de texto ASCII o binarios, archivos de imagen, sonido y vídeo, y archivos científicos como HDF y HDF5. Las funciones de E/S de archivos binarios de bajo nivel le permiten trabajar con archivos de datos en cualquier formato. Las funciones adicionales le permiten leer datos de páginas Web y XML.

Es posible llamar a otras aplicaciones y lenguajes, tales como C, C++, objetos COM, DLLs, Java, FORTRAN y Microsoft Excel y acceder a sitios FTP y servicios Web. Utilizando el Database Toolbox, incluso puede acceder a bases de datos ODBC/JDBC.

## Visualización de datos

Todas las funciones gráficas necesarias para visualizar datos de ingeniería y científicos están disponibles en MATLAB. Incluye funciones de representación de diagramas bidimensionales y tridimensionales, visualización de volumen tridimensional, herramientas para crear diagramas en forma interactiva y la posibilidad de exportar los resultados a los formatos gráficos más conocidos. Es posible personalizar los diagramas añadiendo varios ejes, cambiando los colores de las líneas y marcadores, añadiendo anotaciones, ecuaciones LaTeX, leyendas y trazando formas.

Se pueden visualizar vectores de datos con funciones de representación de diagramas bidimensionales para crear:

- Diagramas de líneas, de área, de barras y de sectores
- Diagramas de dirección y de velocidad
- Histogramas
- Polígonos y superficies
- Diagramas de dispersión/burbujas
- Animaciones

La Figura 1-6 muestra los diagramas lineales de los resultados de varias pruebas de emisión de un motor, con una curva ajustada a los datos.

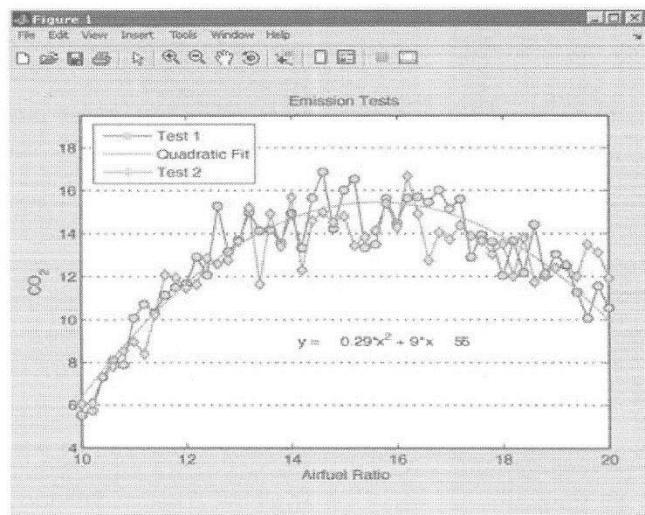


Figura 1-6

MATLAB también proporciona funciones para la visualización de matrices bidimensionales, datos escalares tridimensionales y datos vectoriales tridimensionales. Es posible usar estas funciones para visualizar y comprender grandes cantidades de datos multidimensionales, generalmente complejos. También es posible definir las características de los diagramas, tales como el ángulo de orientación de la cámara, la perspectiva, el efecto de iluminación, la ubicación de la fuente de luz y la transparencia. Las funciones de creación de diagramas tridimensionales incluyen:

- Superficie, contorno y malla
- Diagramas de imagen
- Cono, pastel, flujo e isosuperficie

En la Figura 1-7 se observa un diagrama tridimensional de isosuperficie que revela la estructura geodésica abovedada de una molécula de fullerene carbono-60.

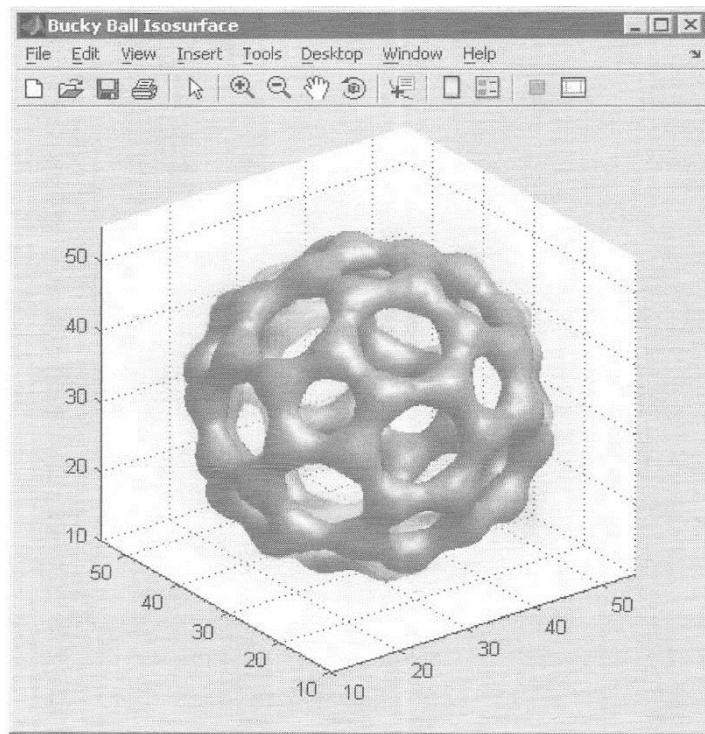


Figura 1-7

MATLAB incluye herramientas interactivas para el diseño y modificación de gráficos. Desde un diagrama de MATLAB, puede realizar cualquiera de las tareas siguientes:

- Arrastrar y soltar nuevos conjuntos de datos en la figura
- Cambiar las propiedades de cualquier objeto de la figura
- Cambiar el zoom, la rotación, hacer una panorámica y cambiar el ángulo de la cámara y la iluminación
- Añadir anotaciones y etiquetas de datos
- Trazar formas

Generar una función de código M para reutilizarla con datos diferentes

En la Figura 1-8 se observa una colección de gráficos, creada interactivamente arrastrando conjuntos de datos a la ventana de diagramas, creando nuevos subdiagramas, cambiando las propiedades tales como colores y tipos de letra y añadiendo anotaciones.

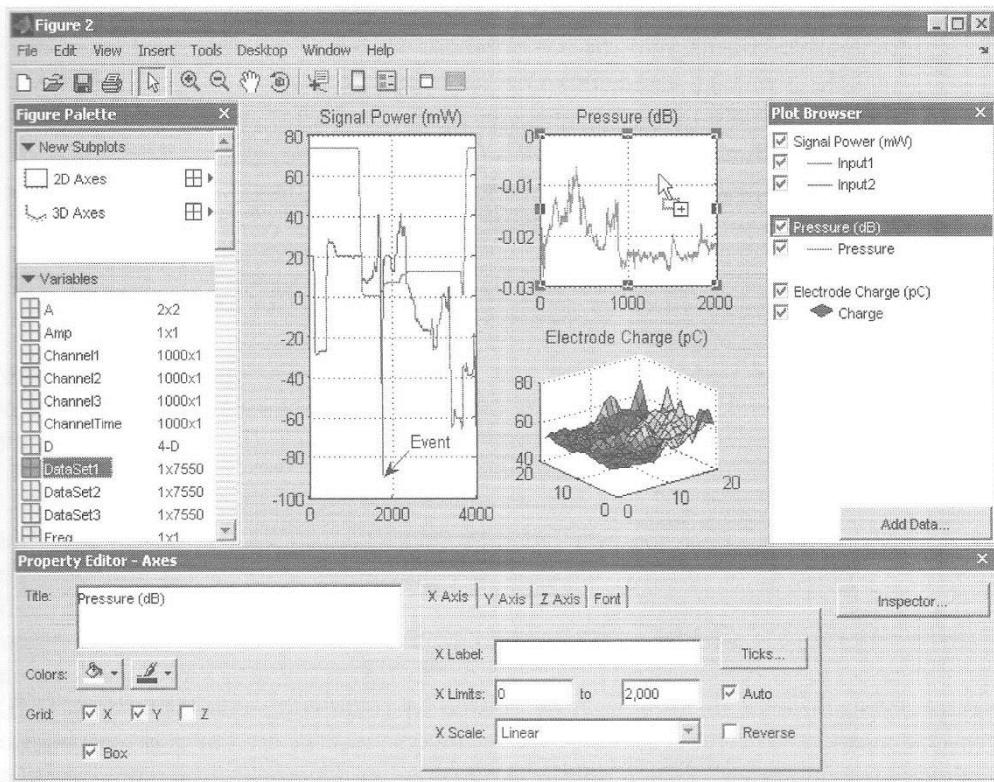


Figura 1-8

MATLAB es compatible con los formatos de archivos de datos y de gráficos más conocidos, tales como GIF, JPEG, BMP, EPS, TIFF, PNG, HDF, AVI, y PCX. Gracias a ello, es posible exportar diagramas de MATLAB a otras aplicaciones, como Microsoft Word y Microsoft Powerpoint, o software de autoedición. Antes de exportar, puede crear y aplicar plantillas de estilo que contengan el diseño, los tipos de letra, la definición del grosor de las líneas, etc., necesarios para cumplir con las especificaciones para su publicación.

## **Cálculo numérico**

MATLAB contiene funciones matemáticas, estadísticas y de ingeniería que soportan la mayoría de las operaciones a realizar en estos campos. Estas funciones, desarrolladas por matemáticos expertos, son el fundamento del lenguaje de MATLAB. Por citar algunos ejemplos, MATLAB implementa siguientes funciones matemáticas y de análisis de datos en los siguientes campos:

- Manipulación de matrices y álgebra lineal
- Polinomios e interpolación
- Análisis de Fourier y filtros
- Análisis de datos y estadística
- Optimización e integración numérica
- Ecuaciones diferenciales ordinarias (ODEs)
- Ecuaciones diferenciales en derivadas parciales (PDEs)
- Operaciones con matrices dispersas

## **Publicación de resultados y distribución de aplicaciones**

Además, MATLAB contiene una serie de funciones para documentar y compartir su trabajo. Puede integrar su código MATLAB con otros lenguajes y aplicaciones, y distribuir sus algoritmos y aplicaciones de MATLAB como programas autónomos o módulos de software.

MATLAB permite exportar los resultados en forma de diagrama o de informes completos. Se pueden exportar los diagramas a todos los formatos de gráficos más conocidos para luego importarlos en otros paquetes como Microsoft Word o Microsoft PowerPoint. Utilizando MATLAB Editor, se puede publicar automáticamente su código MATLAB en formato HTML, Word, LaTeX, etc. En la Figura 1-9 se observa un programa de código M (izquierda) publicado en HTML (derecha) usando MATLAB Editor. Los resultados, que se envían a la ventana de comandos o a diagramas, se capturan e incluyen en el documento y los comentarios se convierten en títulos y texto en HTML

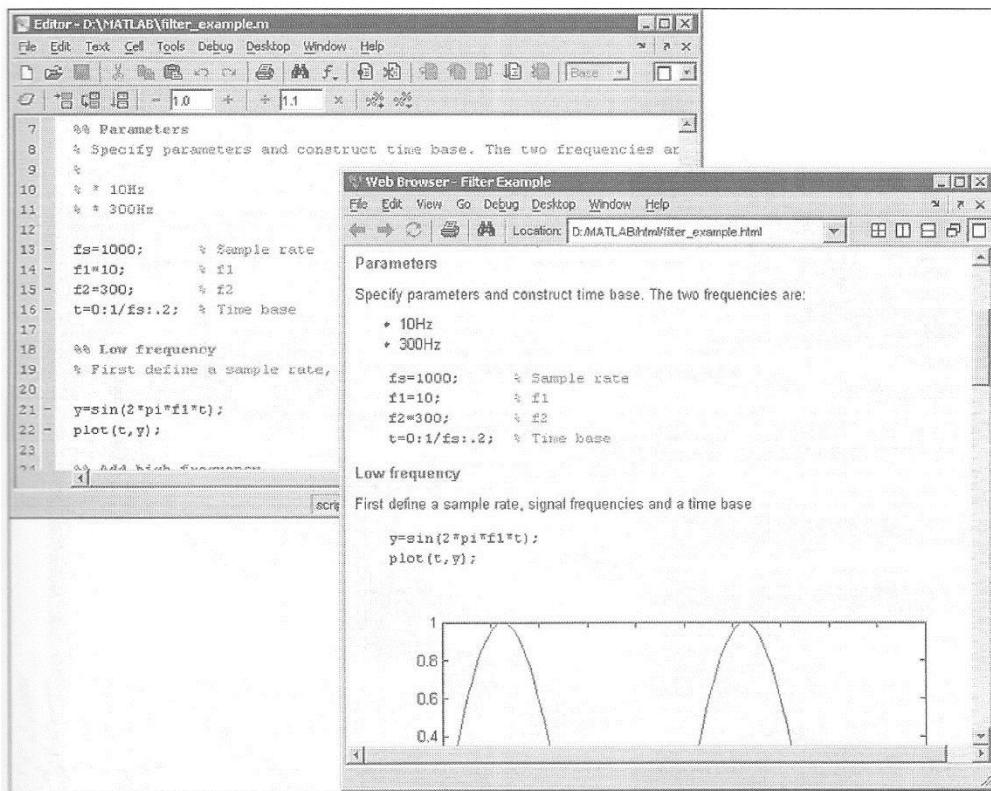


Figura 1-9

Es posible crear informes más complejos, tales como ejecuciones simuladas y varias pruebas de parámetros, utilizando MATLAB Report Generator (disponible por separado).

En cuanto a la integración del código MATLAB con otros lenguajes y aplicaciones, MATLAB proporciona funciones para integrar código C y C++, código FORTRAN, objetos COM y código Java en sus aplicaciones. Puede llamar a DLLs, clases de Java y Controles ActiveX. Utilizando la biblioteca engine de MATLAB, también puede llamar a MATLAB desde código C, C++ o FORTRAN.

En cuanto a distribución de aplicaciones, pueden crearse algoritmos en MATLAB y distribuirlos a otros usuarios de MATLAB como código M. Usando el MATLAB Compiler (disponible por separado), se pueden distribuir algoritmos, ya sea como aplicaciones autónomas o como módulos de software incluido un proyecto, a los usuarios que no tengan MATLAB. Productos adicionales permiten convertir algoritmos en un módulo de software que puede llamarse desde COM o Microsoft Excel.

## EL ENTORNO DE TRABAJO DE MATLAB

En la Figura 1-10 se muestra la pantalla de entrada al programa, que constituye el entorno de trabajo primario de MATLAB.

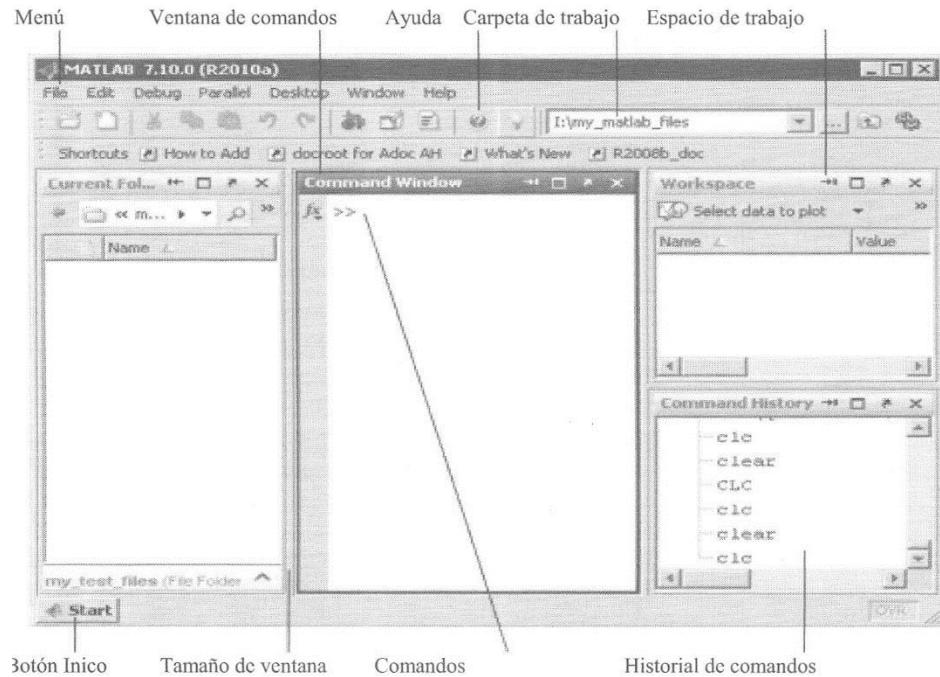


Figura 1-10

La tabla siguiente resume las componentes del entorno de trabajo de MATLAB.

Herramienta	Descripción
Historial de comandos	Permite ver los comandos introducidos durante la sesión en la ventana de comandos, así como copiarlas y ejecutarlos (parte inferior derecha de la Figura 1-11)
Ventana de comandos	Ventana de ejecución de los comandos de MATLAB (parte central de la Figura 1-11)
Espacio de trabajo	Permite ver el contenido del espacio de trabajo (variables...) o <i>Workspace</i> (parte superior derecha de la Figura 1-11)
Ayuda	Ofrece ayuda y demos sobre MATLAB
Botón de inicio (Start)	Permite ejecutar herramientas y acceder a documentación de los componentes de MATLAB (Figura 1-12)

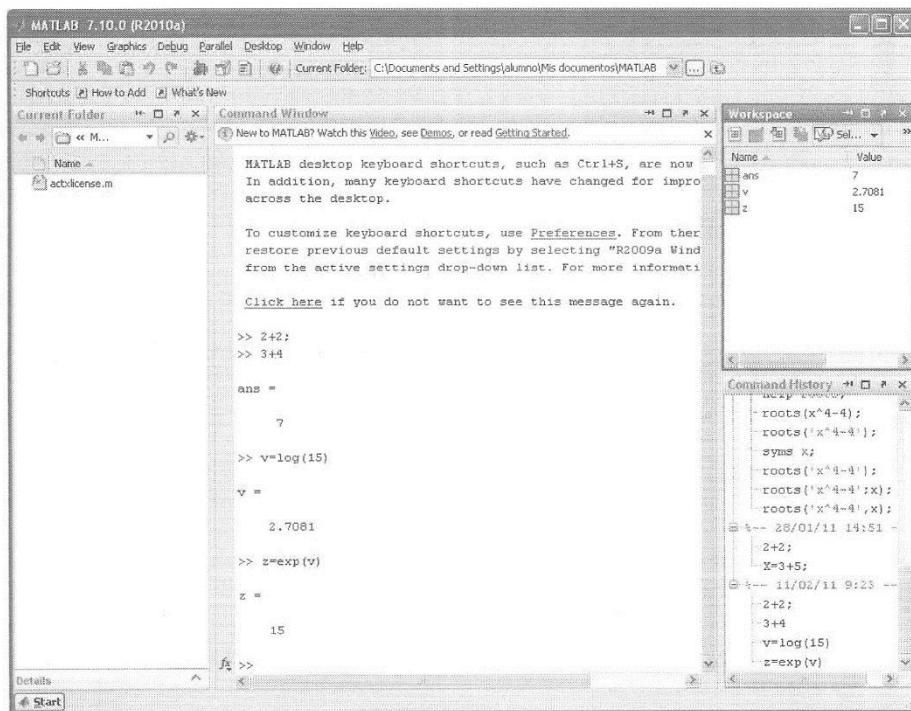


Figura 1-11

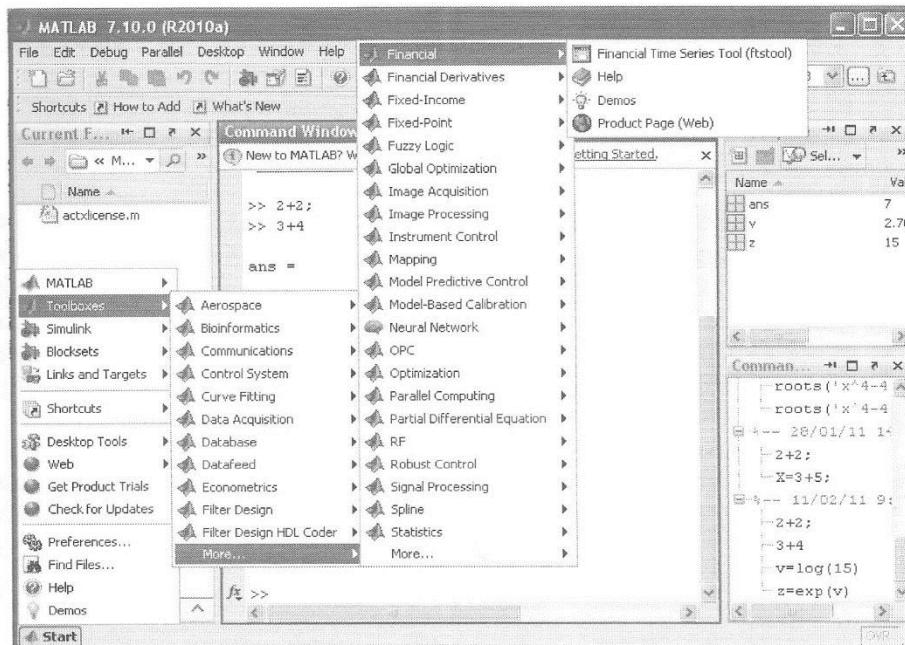
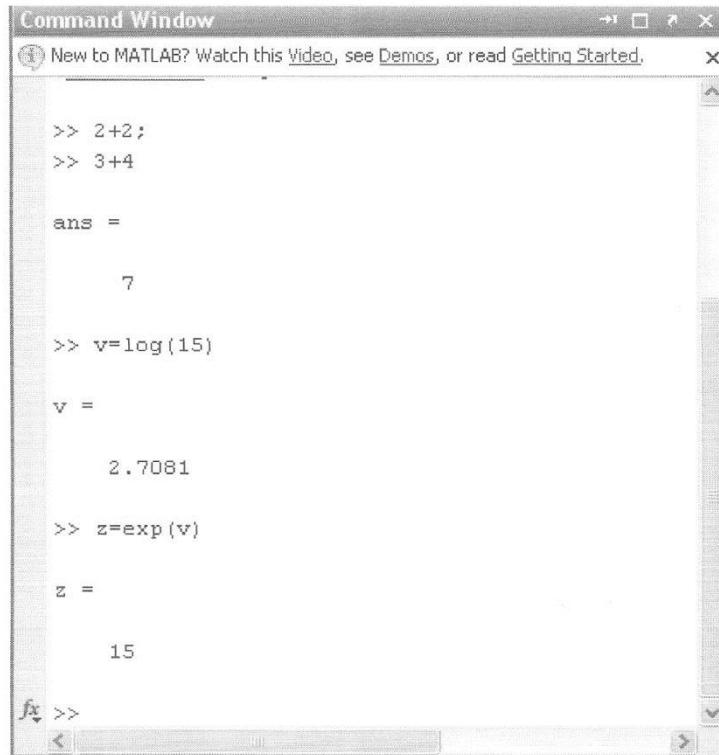


Figura 1-12

Cualquier entrada a ejecutar en MATLAB se escribe a la derecha del *input* del usuario encabezado con el prompt “>>” obteniéndose la respuesta en las líneas inmediatamente inferiores. Después de salida vuelve a aparecer el input de usuario para introducir más entradas (Figura 1-13).



The screenshot shows the MATLAB Command Window titled "Command Window". It displays the following session:

```
>> 2+2;
>> 3+4

ans =

    7

>> v=log(15)

v =

    2.7081

>> z=exp(v)

z =

    15

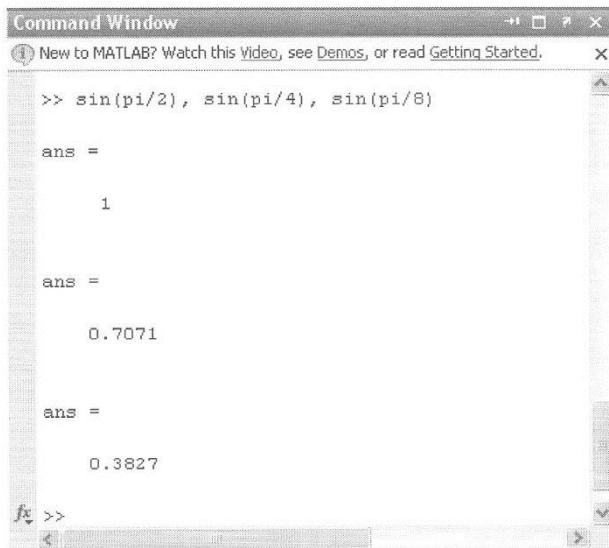
f2 >>
```

Figura 1-13

Cuando en la entrada (*input* de usuario) que se le proponga a MATLAB en la ventana de comandos no se cita una variable para recoger el resultado, devuelve la respuesta utilizando la expresión **ans**=. Tal y como se observa al principio de la Figura 1-13. Si en la entrada de MATLAB definimos una variable que recoja los resultados, posteriormente podremos utilizar esa variable como argumento de entradas posteriores. Es el caso de la variable *v* de la Figura 1-13, que posteriormente se utiliza como entrada de una exponencial.

Para que MATLAB ejecute una entrada, basta con pulsar *Enter* una vez escrita la misma. Si al final de la entrada colocamos un punto y coma, el programa ejecuta los cálculos y los guarda en la memoria (*Workspace*), pero no muestra el resultado en la pantalla (primera entrada de la Figura 1-13). Aparece el prompt de *input* “>>” para indicar que se puede introducir una nueva entrada.

Al igual que el lenguaje C de programación, MATLAB es sensible a la diferencia entre mayúsculas y minúsculas; por ejemplo,  $\text{Sin}(x)$  es distinto que  $\sin(x)$ . Los nombres de todas las funciones incorporadas empiezan con minúscula. No debe haber espacios en los nombres de las órdenes, en símbolos de más de una letra ni en los nombres de las funciones. En los demás casos, los espacios no se tienen en cuenta. Se pueden poner algunos para hacer más legibles su *input*. También se pueden realizar varias entradas en la misma línea de la ventana de comandos separándolas por comas y pulsando *Enter* al final de la última entrada (Figura 1-14). En caso de utilizar un punto y coma al final de una de las entradas de la línea, se omite su correspondiente salida.



The screenshot shows the MATLAB Command Window with the title "Command Window". It displays the following command and its execution:

```
>> sin(pi/2), sin(pi/4), sin(pi/8)

ans =
    1

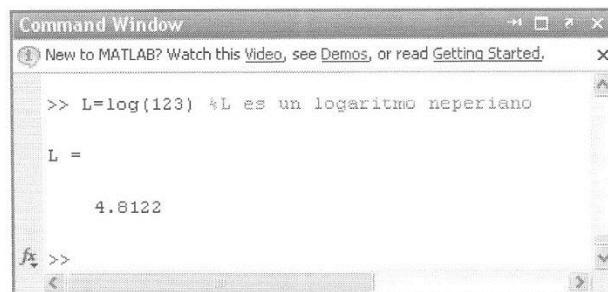
ans =
    0.7071

ans =
    0.3827
```

The window has a standard Windows-style title bar and scroll bars on the right side. The command prompt is at the bottom left.

Figura 1-14

Es posible introducir comentarios descriptivos en una línea de entrada de la ventana de comandos iniciándolos con el signo “%”. Al ejecutar la entrada, Matlab ignora la zona de comentario y procesa el resto (Figura 1-15).



The screenshot shows the MATLAB Command Window with the title "Command Window". It displays the following command and its execution:

```
>> L=log(123) %L es un logaritmo neperiano

L =
    4.8122
```

The window has a standard Windows-style title bar and scroll bars on the right side. The command prompt is at the bottom left.

Figura 1-15

Para simplificar el proceso de introducción de comandos para ser evaluados por el intérprete de Matlab (vía ventana de comandos), se pueden utilizar las teclas de flechas del ordenador. Por ejemplo, si pulsamos la flecha hacia arriba una vez, recuperamos la última entrada submitida en Matlab. Si pulsamos la tecla hacia arriba dos veces, recuperamos la penúltima entrada submitida, y así sucesivamente.

Si tecleamos una secuencia de caracteres determinada en la zona de *input* y a continuación pulsamos la flecha hacia arriba, recuperamos la última entrada que comienza con la secuencia especificada.

Los comandos introducidos durante una sesión de Matlab son almacenados temporalmente en el buffer (*Workspace*) hasta que finalice la sesión con el programa, en cuyo momento se pueden guardar permanentemente en fichero o perderlos definitivamente.

A continuación se presenta un resumen de las teclas que se pueden utilizar en la zona de *input* de Matlab (líneas de comandos), así como sus funciones:

<b>Flecha hacia arriba (Ctrl-P)</b>	Recupera la entrada anterior a la actual.
<b>Flecha hacia abajo (Ctrl-N)</b>	Recupera la entrada siguiente a la actual.
<b>Flecha hacia la izquierda (Ctrl-B)</b>	Lleva el cursor un carácter a la izquierda.
<b>Flecha hacia la derecha (Ctrl-F)</b>	Lleva el cursor un carácter a la derecha.
<b>Ctrl-Flecha a la izquierda</b>	Lleva el cursor una palabra a la izquierda.
<b>Ctrl-Flecha a la derecha</b>	Lleva el cursor una palabra a la derecha.
<b>Inicio (Ctrl-A)</b>	Lleva el cursor al comienzo de la línea.
<b>Fin (Ctrl-E)</b>	Lleva el cursor al final de la línea actual.
<b>Escape</b>	Borra la línea de comandos.
<b>Supr (Ctrl-D)</b>	Borra el carácter indicado por el cursor.
<b>Backspace</b>	Borra el carácter a la izquierda del cursor.
<b>Ctrl-K</b>	Borra toda la línea actual.

El comando *clc* borra la ventana de comandos, pero no borra su contenido del área de trabajo (el contenido permanece en memoria).

## AYUDA EN MATLAB

Se puede obtener ayuda en Matlab a través del botón Ayuda de la barra de iconos  o a través de la opción *Help* de la barra de menús. Además, la ayuda también puede obtenerse a través de comandos implementados como objetos de Matlab. El comando *help* ofrece ayuda general sobre todas los comandos de Matlab (Figura 1-16). Haciendo clic sobre cualquiera de ellos, se obtiene su ayuda particular. Por ejemplo, si hacemos clic sobre el comando *graph2d*, se obtiene la ayuda para gráficos en dos dimensiones (Figura 1-17).

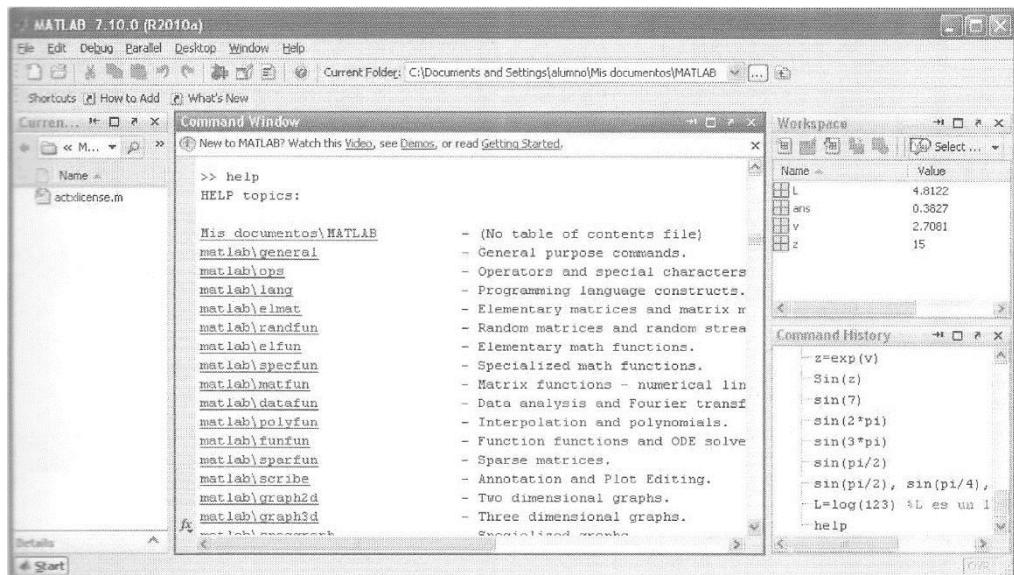


Figura 1-16

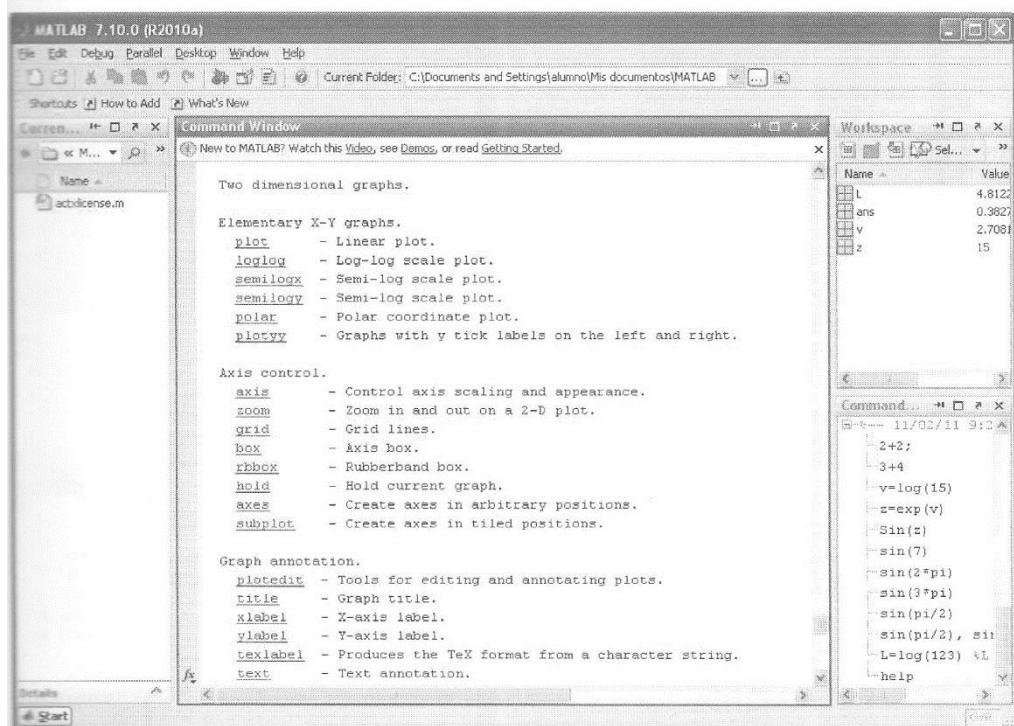


Figura 1-17

Se puede pedir ayuda sobre un comando específico (Figura 1-18) o sobre cualquier tema del contenido (Figura 1-19) utilizando el comando *help comando* o *help tema*.

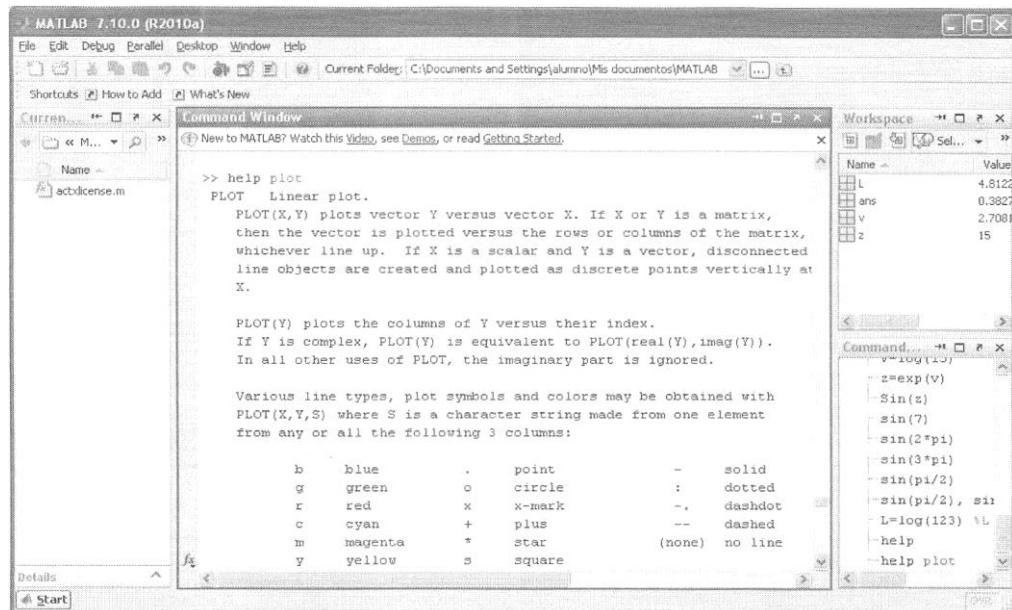


Figura 1-18

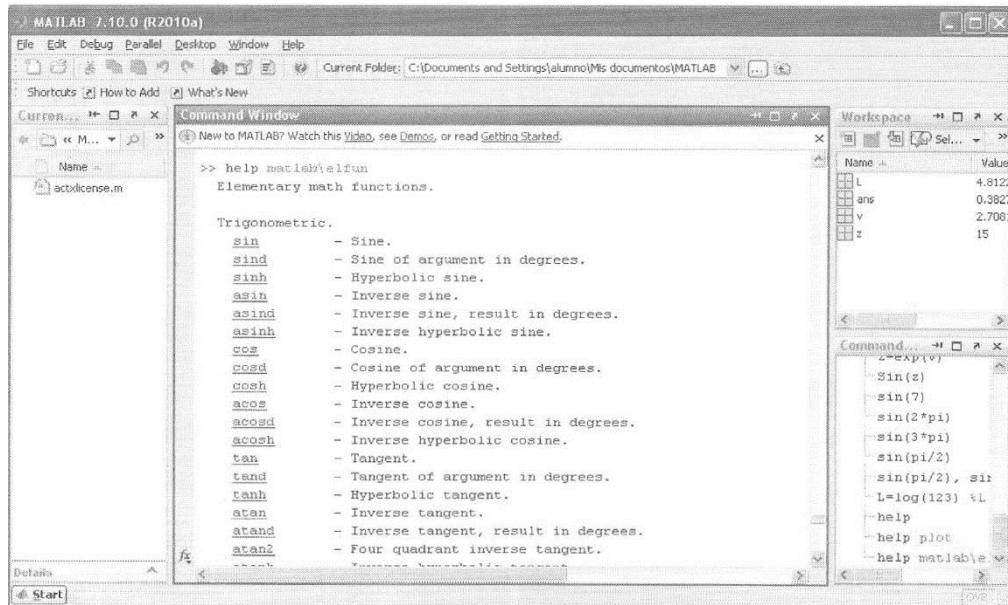


Figura 1-19

El comando *lookfor secuencia* permite encontrar todas aquellas funciones o comandos de Matlab que se refieren a dicha secuencia o la contienen. Este comando es muy útil, bien cuando no se dispone de ayuda directa para la secuencia especificada, o bien para ver la ayuda de todos los comandos que contienen la secuencia. Por ejemplo, si buscamos ayuda para todos los comandos que contienen la secuencia *inv*, podemos usar el comando *lookfor inv* (Figura 1-20).

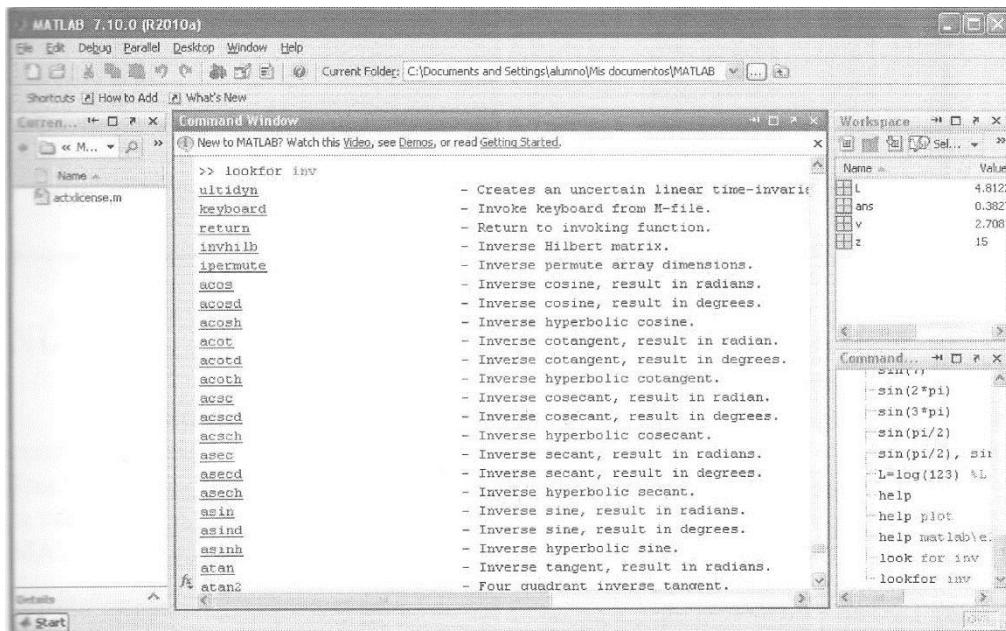


Figura 1-20

## CAPITULO 2

# NUMEROS, VARIABLES, OPERADORES Y FUNCIONES

## NUMEROS, OPERACIONES ARITMETICAS Y FORMATOS

El trabajo numérico en MATLAB es muy amplio. Se pueden utilizar números enteros, racionales, reales y complejos, que a su vez pueden ser argumentos de funciones dando lugar a las funciones enteras, racionales, reales y complejas. Por tanto, la variable compleja es un campo tratable en MATLAB. Las operaciones aritméticas en MATLAB se definen de acuerdo con las convenciones matemáticas estándar. El cuadro siguiente presenta la sintaxis de las operaciones aritméticas básicas:

<b>x + y</b>	<i>Suma</i>  <code>&gt;&gt; 4 + 8</code>  <code>ans =</code>  <code>12</code>
<b>x-y</b>	<i>Diferencia</i>  <code>&gt;&gt; 4-8</code>  <code>ans =</code>  <code>-4</code>
<b>x*y ó x y</b>	<i>Producto</i>  <code>&gt;&gt; 4*8</code>  <code>ans =</code>  <code>32</code>

<b>x/y</b>	<i>División</i>
	>> 4/8 ans = 0.5000
<b>x ^y</b>	<i>Potencia</i>
	>> 4^8 ans = 65536

MATLAB realiza las operaciones aritméticas como si se tratara de una calculadora convencional, pero con precisión total en el cálculo. Los resultados de las operaciones son presentados bien en forma exacta o bien especificando el usuario el grado de precisión que desea. Pero esto sólo afecta a la presentación, ya que la precisión en el cálculo es ilimitada tratándose de la característica que diferencia a MATLAB de otros programas de cálculo numérico en los que la longitud de palabra con que trabaja el ordenador determina la precisión (precisión dependiente del hardware). MATLAB puede representar los resultados con la exactitud que se le requiera, aunque internamente siempre trabaja con cálculos exactos para no arrastrar errores de redondeo. De esta forma tenemos los diferentes formatos de representación aproximada, que en ocasiones facilitan la interpretación de los resultados.

A continuación se citan los comandos relativos a formatos de presentación de los resultados:

<b>format short</b>	<i>Ofrece los resultados con 4 cifras decimales. Se trata del formato por defecto de MATLAB</i>  >> sqrt(23)  ans =  4.7958
<b>format long</b>	<i>Ofrece los resultados con 16 cifras decimales</i>  >> format long; sqrt(23)  ans =  4.795831523312719

<b>x/y</b>	<i>División</i> <pre>&gt;&gt; 4/8 ans = 0.5000</pre>
<b>x^y</b>	<i>Potencia</i> <pre>&gt;&gt; 4^8 ans = 55536</pre>

MATLAB realiza las operaciones aritméticas como si se tratara de una calculadora convencional, pero con precisión total en el cálculo. Los resultados de las operaciones son presentados bien en forma exacta o bien especificando el usuario el grado de precisión que desea. Pero esto sólo afecta a la presentación, ya que la precisión en el cálculo es ilimitada tratándose de la característica que diferencia a MATLAB de otros programas de cálculo numérico en los que la longitud de palabra con que trabaja el ordenador determina la precisión (precisión dependiente del hardware). MATLAB puede representar los resultados con la exactitud que se le requiera, aunque internamente siempre trabaja con cálculos exactos para no arrastrar errores de redondeo. De esta forma tenemos los diferentes formatos de representación aproximada, que en ocasiones facilitan la interpretación de los resultados.

A continuación se citan los comandos relativos a formatos de presentación de los resultados:

<b>format short</b>	<i>Ofrece los resultados con 4 cifras decimales. Se trata del formato por defecto de MATLAB</i> <pre>&gt;&gt; sqrt(23) ans = 4.7958</pre>
<b>format long</b>	<i>Ofrece los resultados con 16 cifras decimales</i> <pre>&gt;&gt; format long; sqrt(23) ans = 4.795831523312719</pre>

<b>format long e</b>	<i>Ofrece los resultados con 16 decimales más la potencia de 10 necesaria</i> <pre>&gt;&gt; format long e; sqrt(23)  ans = 4.795831523312719e+000</pre>
<b>format short e</b>	<i>Ofrece los resultados con 4 decimales más la potencia de 10 necesaria</i> <pre>&gt;&gt; format short e; sqrt(23)  ans = 4.7958e+000</pre>
<b>format long g</b>	<i>Ofrece los resultados en formato largo óptimo</i> <pre>&gt;&gt; format long g; sqrt(23)  ans = 4.79583152331272</pre>
<b>format short g</b>	<i>Ofrece los resultados en formato corto óptimo</i> <pre>&gt;&gt; format short g; sqrt(23)  ans = 4.7958</pre>
<b>format bank</b>	<i>Ofrece los resultados con 2 cifras decimales</i> <pre>&gt;&gt; format bank; sqrt(23)  ans = 4.80</pre>
<b>format rat</b>	<i>Ofrece los resultados en forma de número racional aproximado</i> <pre>&gt;&gt; format rat; sqrt(23)  ans = 1151/240</pre>
<b>format +</b>	<i>Ofrece el signo de los resultados (+, -) e ignora la parte imaginaria de los números complejos</i> <pre>&gt;&gt; format +; sqrt(23)  ans = +</pre>

<b>format hex</b>	<i>Ofrece los resultados en el sistema hexadecimal</i>  <pre>&gt;&gt; format hex; sqrt(23)  ans = 40132eee75770416</pre>
<b>vpa 'operaciones' n</b>	<i>Ofrece el resultado de las operaciones con n dígitos decimales exactos</i>  <pre>&gt;&gt; vpa 'sqrt(23)' 20  ans = 4.7958315233127195416</pre>

## NUMEROS ENTEROS Y FUNCIONES DE VARIABLE ENTERA

MATLAB trabaja con números enteros y con funciones de variable entera de forma exacta. Independientemente del formato de presentación de los resultados, los cálculos son exactos. De todas formas, el comando *vpa* permite obtener salidas exactas con la precisión que se requiera.

En cuanto a las funciones con variable entera, las más importantes que contempla MATLAB son las siguientes (las expresiones entre comillas tienen formato cadena);

<b>rem(n,m)</b>	<i>Resto de la división de n entre m (función válida para n y m reales)</i>  <pre>&gt;&gt; rem(15,2)  ans = 1</pre>
<b>sign(n)</b>	<i>Signo de n (1 si n&gt;0, -1 si n&lt;0, n real)</i>  <pre>&gt;&gt; sign(-8)  ans = -1</pre>
<b>max(n1,n2)</b>	<i>Máximo de los números n1 y n2</i>  <pre>&gt;&gt; max(17,12)  ans = 17</pre>

<b>min(n1,n2)</b>	<i>Mínimo de los números n1 y n2</i> <pre>&gt;&gt; min(17,12) ans =     12</pre>
<b>gcd(n1,n2)</b>	<i>Máximo común divisor de n1 y n2</i> <pre>&gt;&gt; gcd(17,12) ans =     1</pre>
<b>lcm(n1,n2)</b>	<i>Mínimo común múltiplo de n1 y n2</i> <pre>&gt;&gt; lcm(17,12) ans =     204</pre>
<b>factorial(n)</b>	<i>Factorial de n (n(n-1)(n-2)...1)</i> <pre>&gt;&gt; factorial(9) ans =     362880</pre>
<b>factor(n)</b>	<i>Descompone n en factores primos</i> <pre>&gt;&gt; factor (51) ans =     3    17</pre>
<b>dec2base(decimal,n_base)</b>	<i>Convierte el número decimal (base 10) especificado a la nueva base n base dada</i> <pre>&gt;&gt; dec2base(2345, 7) ans =     6560</pre>
<b>base2dec(numero,B)</b>	<i>Convierte el número dado en base B a decimal</i> <pre>&gt;&gt; base2dec('abl2579',12) ans =     32621997</pre>

<b>dec2bin(decimal)</b>	<i>Convierte el número decimal especificado a base 2 (binario)</i>  >> dec2bin(213)  ans =  11010101
<b>dec2hex(decimal)</b>	<i>Convierte el número decimal especificado a base 16 (hexadecimal)</i>  >> dec2hex(213)  ans =  D5
<b>bin2dec(binario)</b>	<i>Convierte el número binario especificado a base decimal</i>  >> bin2dec('1110001')  ans =  113
<b>hex2dec(hexadecimal)</b>	<i>Convierte el número base 16 especificado a base decimal</i>  >> hex2dec('FFAA23')  ans =  16755235

## NUMEROS REALES Y FUNCIONES DE VARIABLE REAL

Los números reales son la unión disjunta de los números racionales y los números irracionales. Un número racional es de la forma  $p/q$ , donde  $p$  es un entero y  $q$  otro entero. La forma en que MATLAB trata los racionales es distinta a la de la mayoría de calculadoras. Los números racionales son cocientes de enteros, y MATLAB también puede trabajar con ellos en modo exacto, de manera que el resultado de expresiones en las que intervienen números racionales es siempre otro número racional o entero. Para ello es necesario activar el formato racional con el comando *format rat*. MATLAB también devuelve aproximaciones decimales de los resultados si el usuario así lo desea, activando cualquier otro tipo de formato (por ejemplo, *format short* o *format long*). Consideramos el siguiente ejemplo:

```
> format rat;
```

```
>> 1/6+1/5-2/10
```

```
ans =
```

```
1/6
```

MATLAB está tratando con racionales como cocientes de enteros y los mantiene en esta forma durante los cálculos. De esta manera, no se arrastran errores de redondeo en los cálculos con fracciones, que pueden llegar a ser muy graves, como demuestra la Teoría de errores. Una vez habilitado el formato racional, las operaciones con racionales serán exactas hasta que no se habilite otro formato distinto. Cuando está habilitado el formato racional, un número con coma flotante, o sea, un número con punto decimal, se interpreta como exacto y MATLAB trata toda la expresión como racional exacta representando el resultado en números racionales. A su vez, si existe un número irracional en una expresión racional, MATLAB lo hace corresponder a la fracción más aproximada para trabajar en formato racional. Consideraremos los siguientes ejemplos:

```
>> format rat;
```

```
>> 2.64/25+4/100
```

```
ans =
```

```
91/625
```

```
>> 2.64/sqrt(25)+4/100
```

```
ans =
```

```
71/125
```

```
>> sqrt(2.64) /25 + 4/100
```

```
204/1943
```

MATLAB también trabaja con los números irracionales representando los resultados con la mayor precisión que puede o con la precisión requerida por el usuario, teniendo presente la acotación de que los irracionales no se pueden representar exactamente como la razón entre dos enteros. A continuación se presenta un ejemplo.

```
>> sqrt(235)
```

```
ans =
```

```
15.3297
```

Existen constantes reales muy típicas que MATLAB representa como sigue:

<b>pi</b>	<i>Número <math>\pi = 3,1415926</math></i>  >> 2*pi  ans =  6.2832
<b>exp(1)</b>	<i>Número <math>e = 2,7182818</math></i>  >> exp (1)  ans =  2.7183
<b>inf</b>	<i>Infinito (por ejemplo <math>1/0</math>)</i>  >> 1/0  ans =  Inf
<b>NaN</b>	<i>Indeterminación (por ejemplo <math>0/0</math>)</i>  >> 0/0  ans =  NaN
<b>realmin</b>	<i>Menor número real positivo utilizable</i>  >> realmin  ans =  2.2251e-308
<b>realmax</b>	<i>Mayor número real positivo utilizable</i>  >> realmax  ans =  1.7977e+308

MATLAB dispone de una gama de funciones de variable real predefinidas, que por supuesto serán válidas para variables racionales, irracionales y enteras. En los párrafos siguientes se presentan las más importantes.

## Funciones trigonométricas

A continuación se presenta una tabla con las funciones trigonométricas y sus inversas que incorpora MATLAB ilustradas con ejemplos.

<i>Función</i>	<i>Inversa</i>
<b>sin(x)</b> Seno <pre>&gt;&gt; sin(pi/2)</pre> ans = 1	<b>asin(x)</b> Arco seno <pre>&gt;&gt; asin(1)</pre> ans = 1.5708
<b>cos(x)</b> Coseno <pre>&gt;&gt; cos(pi)</pre> ans = -1	<b>acos(x)</b> Arco coseno <pre>&gt;&gt; acos(-1)</pre> ans = 3.1416
<b>tan(x)</b> Tangente <pre>&gt;&gt; tan(pi/4)</pre> ans = 1.0000	<b>atan(x) y atan2(x)</b> <i>Arco tangente</i> <pre>&gt;&gt; atan(1)</pre> ans = 0.7854
<b>csc(x)</b> Coseno <pre>&gt;&gt; CSC(pi/2)</pre> ans = 1	<b>acsc(x)</b> Arco coseno <pre>&gt;&gt; acsc(1)</pre> ans = 1.5708
<b>sec(x)</b> Secante <pre>&gt;&gt; sec(pi)</pre> ans = -1	<b>asec(x)</b> Arco secante <pre>&gt;&gt; asec(-1)</pre> ans = 3.1416
<b>cot(x)</b> Cotangente <pre>&gt;&gt; cot(pi/4)</pre> ans = 1.0000	<b>acot(x)</b> Arco cotangente <pre>&gt;&gt; acot(1)</pre> ans = 0.7854

## Funciones hiperbólicas

A continuación se presenta una tabla con las funciones hiperbólicas y sus inversas que incorpora MATLAB ilustradas con ejemplos.

<i>Función</i>	<i>Inversa</i>
<b><i>sinh(x) Seno hiperbólico</i></b> <pre>&gt;&gt; sinh (2) ans = 3.6269</pre>	<b><i>asinh(x) Arco seno hiperbólico</i></b> <pre>&gt;&gt; asinh(3.6269) ans = 2.0000</pre>
<b><i>cosh(x) Coseno hiperbólico</i></b> <pre>&gt;&gt; cosh(3) ans = 10.0677</pre>	<b><i>acosh(x) Arco coseno hiperbólico</i></b> <pre>&gt;&gt; acosh(10.0677) ans = 3.0000</pre>
<b><i>tanh(x) Tangente hiperbólica</i></b> <pre>&gt;&gt; tanh(1) ans = 0.7616</pre>	<b><i>atanh(x) Arco tangente hiperbólica</i></b> <pre>&gt;&gt; atanh(0.7616) ans = 1.0000</pre>
<b><i>csch(x) Cosecante hiperbólica</i></b> <pre>&gt;&gt; csch(3.14159) ans = 0.0866</pre>	<b><i>acsch(x) Arco cosecante hiperbólica</i></b> <pre>&gt;&gt; acsch(0.0866) ans = 3.1415</pre>
<b><i>sech(x) Secante hiperbólica</i></b> <pre>&gt;&gt; sech(2.7182818) ans = 0.1314</pre>	<b><i>asech(x) Arco secante hiperbólica</i></b> <pre>&gt;&gt; asech(0.1314) ans = 2.7183</pre>
<b><i>coth(x) Cotangente hiperbólica</i></b> <pre>&gt;&gt; coth(9) ans = 1.0000</pre>	<b><i>acoth(x) Arco cotangente hiperbólica</i></b> <pre>&gt;&gt; acoth(0.9999) ans = 4.9517 + 1.5708i</pre>

### *Funciones exponenciales y logarítmicas*

A continuación se presenta una tabla con las funciones exponenciales y logarítmicas que incorpora MATLAB ilustradas con ejemplos.

<i>Función</i>	<i>Significado</i>
<b><i>exp(x)</i></b>	Función exponencial en base e ( $e^x$ ) <pre>&gt;&gt; exp(log(7)) ans = 7</pre>

<b>log(x)</b>	<i>Función logaritmo en base e de x</i> <pre>&gt;&gt; log(exp(7)) ans =     7</pre>
<b>log10(x)</b>	<i>Función logaritmo en base 10 de x</i> <pre>&gt;&gt; log10(1000) ans =     3</pre>
<b>log2(x)</b>	<i>Función logaritmo en base 2 de x</i> <pre>&gt;&gt; log2(2^8) ans =     8</pre>
<b>pow2(x)</b>	<i>Función potencia de base 2 de x</i> <pre>&gt;&gt; pow2(log2(8)) ans =     8</pre>
<b>sqrt(x)</b>	<i>Función raíz cuadrada de x</i> <pre>&gt;&gt; sqrt(2^8) ans =     16</pre>

### Funciones específicas de variable numérica

MATLAB incorpora un grupo de funciones de variable numérica cuya finalidad es obtener resultados aproximados de operaciones y redondeos, trabajar con signos, etc. Entre estas funciones tenemos las siguientes:

<i>Función</i>	<i>Significado</i>
<b>abs(x)</b>	<i>Valor absoluto del real x</i> <pre>&gt;&gt; abs (-8) ans =     8</pre>
<b>floor(x)</b>	<i>El mayor entero menor o igual que el real x</i> <pre>&gt;&gt; floor(-23.557) ans =     -24</pre>

<b>ceil(x)</b>	<i>El menor entero mayor o igual que el real x</i> <pre>&gt;&gt; ceil (-23.557) ans = -23</pre>
<b>round(x)</b>	<i>El entero más próximo al real x</i> <pre>&gt;&gt; round(-23.557) ans = -24</pre>
<b>fix(x)</b>	<i>Elimina la parte decimal del real x</i> <pre>&gt;&gt; fix(-23.557) ans = -23</pre>
<b>rem(a,b)</b>	<i>Da el resto de la división entre los reales a y b</i> <pre>&gt;&gt; rem(7,3) ans = 1</pre>
<b>sign(x)</b>	<i>Signo del real x (1 si <math>x &gt; 0</math>, -1 si <math>x &lt; 0</math>)</i> <pre>&gt;&gt; sign(-23.557) ans = -1</pre>

## VARIABLES UNIDIMENSIONALES, VECTORIALES Y MATRICIALES

MATLAB es un software enfocado al lenguaje matricial, por lo tanto enfoca especialmente las tareas al trabajo con matrices.

La forma inicial de definir una variable es muy sencilla. Basta con utilizar la sintaxis siguiente:

### Variable = objeto

Donde el objeto puede ser un escalar, un vector o una matriz.

Si la variable es un vector, su sintaxis en función de sus componentes se puede escribir de una de las dos formas siguientes:

**Variable = [v<sub>1</sub> v<sub>2</sub> v<sub>3</sub> ... v<sub>n</sub>]**  
**Variable = [v<sub>1</sub>, v<sub>2</sub>, v<sub>3</sub>, ..., v<sub>n</sub>]**

Si la variable es una matriz, su sintaxis en función de sus componentes se puede escribir de una de las dos formas siguientes:

**Variable = [v<sub>11</sub> v<sub>12</sub> v<sub>13</sub> ... v<sub>1n</sub>; v<sub>21</sub> v<sub>22</sub> v<sub>23</sub> ... v<sub>2n</sub>; ...]**  
**Variable = [v<sub>11</sub>, v<sub>12</sub>, v<sub>13</sub>,..., v<sub>1n</sub>; v<sub>21</sub>, v<sub>22</sub>, v<sub>23</sub>,..., v<sub>2n</sub>; ...]**

La escritura de variables se realiza en la ventana de comandos de MATLAB (*Command Window*) de modo natural, tal y como se observa en la Figura 2-1.

Vemos que en la ventana de espacio de trabajo (*Workspace*) se sitúan todas las variables que vamos definiendo en la sesión. El valor de cualquiera de estas variables es recuperable posteriormente tecleando su nombre sobre la ventana de comandos (Figura 2-2).

Asimismo, en la ventana de historial de comandos (*Command History*) se va situando toda la sintaxis que ejecutamos desde la ventana de comandos.

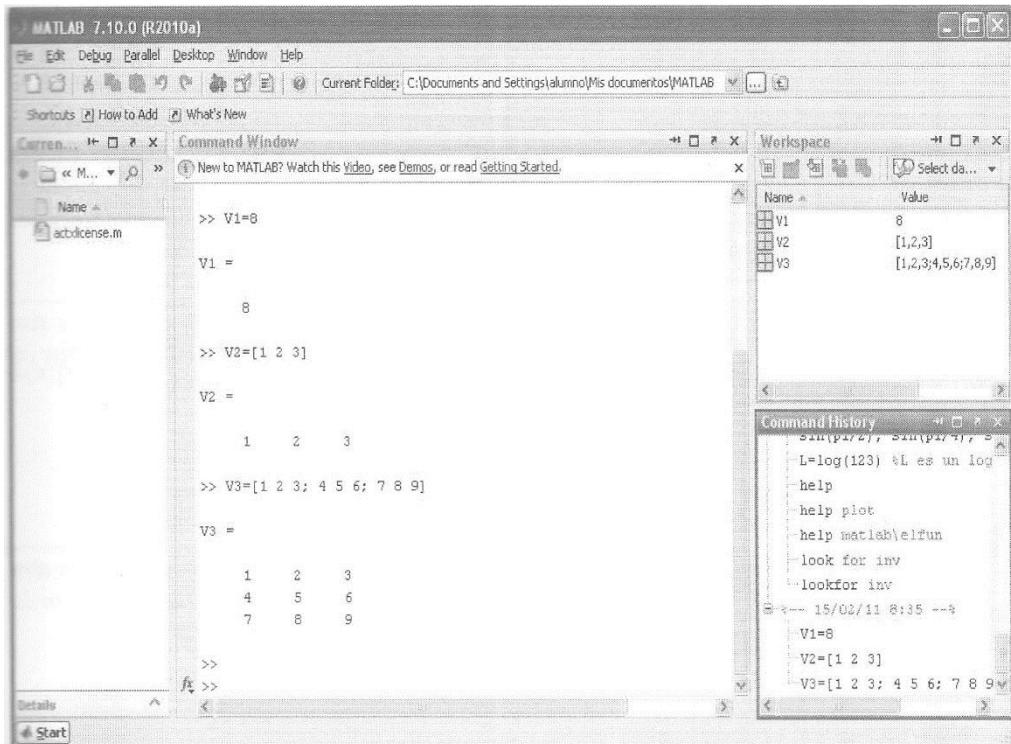


Figura 2-1

Una vez definida una variable, podemos operar sobre ella utilizándola como una variable habitual en matemáticas teniendo presente que los nombres de las variables son sensibles a mayúsculas y minúsculas. En la Figura 2-3 se observan algunas operaciones con variables unidimensionales, vectoriales y matriciales. Es importante notar que al calcular el logaritmo de V2 se cometió el error de usar la variable en minúsculas.

En MATLAB los nombres de las variables comienzan por una letra seguida de cualquier número de letras, dígitos o subrayados hasta 31 caracteres como máximo.

```

Command Window
>> V3
V3 =
    1     2     3
    4     5     6
    7     8     9

>> V2
V2 =
    1     2     3

>> V1
V1 =
    8

```

```

Command Window
>> V3^2
ans =
    30     36     42
    66     81     96
   102    126    150

>> log(v2)
?? Undefined function or variable 'v2'.

>> log(V2)
ans =
    0    0.6931    1.0986

>> atan(V1)
ans =
    1.4464

```

Figura 2-2

Figura 2-3

También existen formas específicas de definición de variables vectoriales, entre las que se encuentran las siguientes;

<b>variable=[a:b]</b>	<p>Define el vector cuyos primero y último elementos son a y b, respectivamente, y los elementos intermedios se diferencian en una unidad</p> <pre>&gt;&gt; vector1=[2:6] vector1 =     2     3     4     5     6</pre>
<b>variable=[a:s:b]</b>	<p>Define el vector cuyos primero y último elementos son a y b, y los elementos intermedios se diferencian en la cantidad s especificada por el incremento</p> <pre>&gt;&gt; vector2=[2:2:8] vector2 =     2     4     6     8</pre>

<b>variable=linspace [a,b,n]</b>	<i>Define el vector cuyos primero y último elementos son a y b, y que tiene en total n elementos uniformemente espaciados entre sí</i> <pre>&gt; vector3=linspace(10, 30, 6) vector3 =     10    14    18    22    26    30</pre>
----------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### **Elementos de variables vectoriales**

MATLAB permite la selección de elementos de variables vectoriales mediante los siguientes comandos:

<b>x(n)</b>	<i>Devuelve el enésimo elemento del vector x</i> <pre>&gt;&gt; X=(2:8) X =     2    3    4    5    6    7    8 &gt;&gt; X(3) ans =     4</pre>
<b>x(a:b)</b>	<i>Devuelve los elementos del vector x situados entre el a-ésimo y el b-ésimo, ambos inclusive</i> <pre>&gt;&gt; X(3:5) ans =     4    5    6</pre>
<b>x(a:p:b)</b>	<i>Devuelve los elementos del vector x situados entre el a-ésimo y el b-ésimo, ambos inclusive, pero separados de p en p unidades (a&gt;b)</i> <pre>&gt;&gt; X(1:2:6) ans =     2    4    6</pre>
<b>x(b:-p:a)</b>	<i>Devuelve los elementos del vector x situados entre el b-ésimo y el a-ésimo, ambos inclusive, pero separados de p en p unidades y empezando por el b-ésimo (b&gt;a)</i> <pre>&gt;&gt; X(6:-2:1) ans =     7    5    3</pre>

## Elementos de variables matriciales

Lo mismo que para el caso de los vectores, MATLAB permite la selección de elementos de variables matriciales mediante los siguientes comandos:

<b>A(m,n)</b>	<i>Define el elemento (m,n) de la matriz A (fila m y columna n)</i>  <pre>&gt;&gt; A = [ 3 5 7 4 ; 1 6 8 9 ; 2 6 8 1 ] A =     3     5     7     4     6     8     9     2     6     8     1 &gt;&gt; A ( 2 , 4 ) ans =     9</pre>
<b>A(a:b,c:d)</b>	<i>Define la submatriz de A formada por las filas que hay entre la a-ésima y la b-ésima y por las columnas que hay entre la c-ésima y la d-ésima</i>  <pre>&gt;&gt; A(1:2,2:3) ans =     5     7     5     8</pre>
<b>A(a:p:b,c:q:d)</b>	<i>Define la submatriz de A formada por las filas que hay entre la a-ésima y la b-ésima tomándolas de p en p, y por las columnas que hay entre la c-ésima y la d-ésima tomándolas de q en q</i>  <pre>&gt;&gt; A(1:2:3,1:2:4) ans =     3     7     2     8</pre>
<b>A([ab],[c d])</b>	<i>Define la submatriz de A formada por la intersección de las filas a-ésima y b-ésima y las columnas c-ésima y d-ésima</i>  <pre>&gt;&gt; A([2 3] , [2 4]) ans =     7     9     6     1</pre>

<b>A([a b c ...], [e f g ...])</b>	Define la submatriz de A formada por la intersección de las filas a, b, c, ... y las columnas e, f, g, ... <pre>&gt;&gt; A ([ 2 3 ] , [ 1 2 4 ] )</pre> <p>ans =</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>1</td><td>6</td><td>9</td></tr> <tr><td>2</td><td>6</td><td>1</td></tr> </table>	1	6	9	2	6	1			
1	6	9								
2	6	1								
<b>A(:,c:d)</b>	Define la submatriz de A formada por todas las filas de A y por las columnas que hay entre la c-ésima y la d-ésima <pre>&gt;&gt; A (:, 2:4)</pre> <p>ans =</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>5</td><td>7</td><td>4</td></tr> <tr><td>6</td><td>8</td><td>9</td></tr> <tr><td>6</td><td>8</td><td>1</td></tr> </table>	5	7	4	6	8	9	6	8	1
5	7	4								
6	8	9								
6	8	1								
<b>A(:,[c d e ...])</b>	Define la submatriz de A formada por todas las filas de A y por las columnas c, d, e, ... <pre>&gt;&gt; A (:, [ 2 , 3 ] )</pre> <p>ans =</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>5</td><td>7</td></tr> <tr><td>6</td><td>8</td></tr> <tr><td>6</td><td>8</td></tr> </table>	5	7	6	8	6	8			
5	7									
6	8									
6	8									
<b>A(a:b,:)</b>	Define la submatriz de A formada por todas las columnas de A y por las filas que hay entre la a-ésima y la b-ésima <pre>&gt;&gt; A (2:3, ;)</pre> <p>ans =</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>1</td><td>6</td><td>8</td><td>9</td></tr> <tr><td>2</td><td>6</td><td>8</td><td>1</td></tr> </table>	1	6	8	9	2	6	8	1	
1	6	8	9							
2	6	8	1							
<b>A([a b c ...],:)</b>	Define la submatriz de A formada por todas las columnas de A y por las filas a, b, c, ... <pre>&gt;&gt; A ([ 1 , 3 ] , :)</pre> <p>ans =</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>3</td><td>5</td><td>7</td><td>4</td></tr> <tr><td>2</td><td>6</td><td>8</td><td>1</td></tr> </table>	3	5	7	4	2	6	8	1	
3	5	7	4							
2	6	8	1							
<b>A(a,:)</b>	Define la fila a-ésima de la matriz A <pre>&gt;&gt; A (3, :)</pre> <p>ans =</p> <table style="margin-left: auto; margin-right: auto;"> <tr><td>2</td><td>6</td><td>8</td><td>1</td></tr> </table>	2	6	8	1					
2	6	8	1							

<b>A(:,b)</b>	<i>Define la columna b-ésima de la matriz A</i> <pre>&gt;&gt; A ( : , 3 )</pre> <pre>ans =     7     8     8</pre>
<b>A(:)</b>	<i>Define un vector columna cuyos elementos son las columnas de A situadas por orden una debajo de otra</i> <pre>&gt;&gt; A ( : )</pre> <pre>ans =</pre> <pre>     3     1     2     5     6     6     7     8     8     4     9     1</pre>
<b>A(:,:,b)</b>	<i>Equivale a toda la matriz A</i> <pre>&gt;&gt; A ( : , : )</pre> <pre>ans =     3     5     7     4     1     6     8     9     2     6     8     1</pre>
<b>[A,B,C,...]</b>	<i>Define la matriz formada por las submatrices A, B, C, ...</i> <pre>&gt;&gt; A1=[ 2  6 ;  4  1 ] , A2=[ 3  8 ;  6  9 ]</pre> <pre>A1 =     2     6     4     1</pre> <pre>A2 =     3     8     5     9</pre> <pre>&gt;&gt; [A1,A2]</pre> <pre>ans =     2     6     3     8     4     1     6     9</pre>

## Funciones matriciales específicas

En MATLAB se utilizan un grupo de funciones matriciales predefinidas que facilitan el trabajo en el campo matricial. Las más importantes son las siguientes:

<b>diag(v)</b>	<i>Crea una matriz diagonal con el vector v en la diagonal</i> <pre>&gt;&gt; diag ([ 2  0  9  8  7 ] )</pre> <p>ans =</p> <table border="0"> <tr><td>2</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>9</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>8</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0</td><td>7</td></tr> </table>	2	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	8	0	0	0	0	0	7
2	0	0	0	0																						
0	0	0	0	0																						
0	0	9	0	0																						
0	0	0	8	0																						
0	0	0	0	7																						
<b>diag(A)</b>	<i>Extrae la diagonal de la matriz A como vector columna</i> <pre>&gt;&gt; A = [ 1  3  5 ; 2  0  B ; -1  -3  2 ]</pre> <p>A =</p> <table border="0"> <tr><td>1</td><td>3</td><td>5</td></tr> <tr><td>2</td><td>0</td><td>8</td></tr> <tr><td>-1</td><td>-3</td><td>2</td></tr> </table> <pre>&gt;&gt; diag (A)</pre> <p>ans =</p> <table border="0"> <tr><td>1</td></tr> <tr><td>0</td></tr> <tr><td>2</td></tr> </table>	1	3	5	2	0	8	-1	-3	2	1	0	2													
1	3	5																								
2	0	8																								
-1	-3	2																								
1																										
0																										
2																										
<b>eye(n)</b>	<i>Crea la matriz identidad de orden n</i> <pre>&gt;&gt; eye ( 4 )</pre> <p>ans =</p> <table border="0"> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> </table>	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1									
1	0	0	0																							
0	1	0	0																							
0	0	1	0																							
0	0	0	1																							
<b>eye(m,n)</b>	<i>Crea la matriz de orden mxn con unos en la diagonal principal y ceros en el resto</i> <pre>&gt;&gt; eye ( 3 , 5 )</pre> <p>ans =</p> <table border="0"> <tr><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td></tr> </table>	1	0	0	0	0	0	1	0	0	0	0	0	1	0	0										
1	0	0	0	0																						
0	1	0	0	0																						
0	0	1	0	0																						
<b>zeros(m,n)</b>	<i>Crea la matriz nula de orden mxn</i> <pre>&gt;&gt; zeros (2 , 3 )</pre> <p>ans =</p> <table border="0"> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td></tr> </table>	0	0	0	0	0	0																			
0	0	0																								
0	0	0																								

<b>ones(m,n)</b>	<i>Crea la matriz de orden mxn con todos sus elementos 1</i> <pre>&gt;&gt; ones (2,3)  ans =  1 1 1 1 1 1</pre>
<b>rand(m,n)</b>	<i>Crea una matriz aleatoria uniforme de orden mxn</i> <pre>&gt;&gt; rand(4,5)  ans =  0.8147    0.6324    0.9575    0.9572    0.4218 0.9058    0.0975    0.9649    0.4854    0.9157 0.1270    0.2785    0.1576    0.8003    0.7922 0.9134    0.5469    0.9706    0.1419    0.9595</pre>
<b>randn(m,n)</b>	<i>Crea una matriz aleatoria normal de orden mxn</i> <pre>&gt;&gt; randn(4,5)  ans =  0.6715    0.4889    0.2939   -1.0689    0.3252 -1.2075    1.0347   -0.7873   -0.8095   -0.7549 0.7172    0.7269    0.8884   -2.9443    1.3703 1.6302   -0.3034   -1.1471    1.4384   -1.7115</pre>
<b>flipud(A)</b>	<i>Devuelve la matriz cuyas filas están colocadas en orden inverso (de arriba abajo) a las filas de A</i> <pre>&gt;&gt; flipud(A)  ans =  -1   -3    2  2     0    8  1     3    5</pre>
<b>fliplr(A)</b>	<i>Devuelve la matriz cuyas columnas están colocadas en orden inverso (de izquierda a derecha) a las de A</i> <pre>&gt;&gt; fliplr(A)  ans =   5     3     1  8     0     2  2    -3    -1</pre>
<b>rot90(A)</b>	<i>Rota 90 grados la matriz A</i> <pre>&gt;&gt; rot90(A)  ans =   5     8     2  2     0    -3  1     2    -1</pre>
<b>reshape(A,m,n)</b>	<i>Devuelve la matriz de orden mxn extraída de la matriz A tomando elementos consecutivos de A por columnas</i>

	<pre>&gt;&gt; reshape (A, 3, 3)  ans =     1    3    5     2    0    8    -1   -3    2</pre>
<b>size(A)</b>	<p><i>Devuelve el orden (tamaño) de la matriz A</i></p> <pre>&gt;&gt; size(A)  ans =     3    3</pre>
<b>length(v)</b>	<p><i>Devuelve la longitud del vector v</i></p> <pre>&gt;&gt; length([1 3 4 5 -1])  ans =     5</pre>
<b>tril(A)</b>	<p><i>Devuelve la parte triangular inferior de la matriz A</i></p> <pre>&gt;&gt; tril(A)  ans =     1    0    0     2    0    0    -1   -3    2</pre>
<b>triu(A)</b>	<p><i>Devuelve la parte triangular superior de la matriz A</i></p> <pre>&gt;&gt; triu(A)  ans =     1    3    5     0    0    8     0    0    2</pre>
<b>A'</b>	<p><i>Devuelve la matriz transpuesta de A</i></p> <pre>&gt;&gt; A'  ans =     2    2   -1     3    0   -3     5    8    2</pre>
<b>inv(A)</b>	<p><i>Devuelve la matriz inversa de A</i></p> <pre>&gt;&gt; inv(A)  ans =     -0.5714  0.5000 -0.5714     0.2857   -0.1667 -0.0476     0.1429           0  0.1429</pre>

## NUMEROS ALEATORIOS

MATLAB incorpora funciones que posibilitan el trabajo con números aleatorios de modo sencillo. Las funciones *rand* y *randn* son las funciones básicas que generan números aleatorios distribuidos uniformemente y normalmente respectivamente. A continuación se presentan las funciones más habituales para el trabajo con números aleatorios.

<b>rand</b>	<i>Devuelve un número decimal aleatorio distribuido uniformemente en el intervalo [0,1]</i> <pre>&gt;&gt; rand ans = 0.8147</pre>
<b>rand(n)</b>	<i>Devuelve una matriz de dimensión nxn cuyos elementos son números decimales aleatorios distribuidos uniformemente en el intervalo [0,1]</i> <pre>&gt;&gt; rand(3) ans = 0.9058    0.6324    0.5469 0.1270    0.0975    0.9575 0.9134    0.2785    0.9649</pre>
<b>rand(m,n)</b>	<i>Devuelve una matriz de dimensión mxn cuyos elementos son números decimales aleatorios distribuidos uniformemente en el intervalo [0,1]</i> <pre>&gt;&gt; rand(2,3) ans = 0.1576    0.9572    0.8003 0.9706    0.4854    0.1419</pre>
<b>rand(size(A))</b>	<i>Devuelve una matriz del mismo tamaño que la matriz A y cuyos elementos son números decimales aleatorios distribuidos uniformemente en el intervalo [0,1]</i> <pre>&gt;&gt; rand(size(eye(3))) ans = 0.4218    0.9595    0.8491 0.9157    0.6557    0.9340 0.7922    0.0357    0.6787</pre>
<b>rand('seed')</b>	<i>Devuelve el valor actual de la semilla generadora de los números aleatorios uniformes</i> <pre>&gt;&gt; rand('seed') ans = 931316785</pre>

<b>rand('seed',n)</b>	<i>Coloca en la cantidad n el valor actual de la semilla generadora de los números aleatorios uniformes</i> <pre>&gt;&gt; rand('seed')  ans =  931316785  &gt;&gt; rand('seed', 1000)  &gt;&gt; rand('seed')  ans =  1000</pre>
<b>randn</b>	<i>Devuelve un número decimal aleatorio distribuido según una normal de media 0 y varianza 1</i> <pre>&gt;&gt; randn  ans =  -0.4326</pre>
<b>randn(n)</b>	<i>Devuelve una matriz de dimensión nxn cuyos elementos son números decimales aleatorios distribuidos según una normal de media 0 y varianza 1</i> <pre>&gt;&gt; randn (3)  ans =  -1.6656    -1.1465    -0.0376  0.1253     1.1909     0.3273  0.2877     1.1892     0.1746</pre>
<b>randn(m,n)</b>	<i>Devuelve una matriz de dimensión mxn cuyos elementos son números decimales aleatorios distribuidos según una normal de media 0 y varianza 1</i> <pre>&gt;&gt; randn(2,3)  ans =  -0.1867    -0.5883    -0.1364  0.7258     2.1832     0.1139</pre>
<b>randn(size(A))</b>	<i>Devuelve una matriz del mismo tamaño que la matriz A y cuyos elementos son números decimales aleatorios distribuidos según una normal de media 0 y varianza 1</i> <pre>&gt;&gt; randn(size(eye(3)))  ans =   1.0668    -0.8323    0.7143  0.0593     0.2944    1.6236  -0.0956   -1.3362   -0.6918</pre>

<b>randn('seed')</b>	<i>Devuelve el valor actual de la semilla generadora de los números aleatorios normales</i>  <pre>&gt;&gt; randn('seed') ans = 931316785</pre>
<b>randn('seed',n)</b>	<i>Coloca en la cantidad n el valor actual de la semilla generadora de los números aleatorios uniformes</i> <pre>&gt;&gt; randn('seed',1000) &gt;&gt; randn('seed') ans = 1000</pre>

## OPERADORES

MATLAB es un lenguaje que incorpora operadores aritméticos, lógicos y relacionales de igual forma que cualquier otro lenguaje. A continuación se enumeran los tipos de operadores citados en el ámbito del lenguaje de MATLAB.

### *Operadores aritméticos*

MATLAB incorpora los operadores aritméticos habituales (suma, resta, multiplicación y división) para trabajar con números. Pero a la vez amplía el sentido de estos operadores para el trabajo con escalares, vectores y matrices, tal y como se muestra en la tabla siguiente:

<i>Operador</i>	<i>Función que desempeña</i>
<b>+</b>	<i>Suma de escalares, vectores o matrices</i>  <pre>&gt;&gt; A=[1 3 -2 6];B=[4 -5 8 2]; c=3; &gt;&gt; V1=A+c V1 = 4      6      1      9 &gt;&gt; V2=A+B V2 = 5      -2      6      8</pre>
<b>-</b>	<i>Resta de escalares, vectores o matrices</i>  <pre>&gt;&gt; V3=A-c V3 = -2      0      -5      3</pre>

	<pre>&gt;&gt; V4=A-B V4 =</pre>
*	<p><i>Producto de escalares o de matrices o de escalares por vectores o matrices</i></p> <pre>&gt;&gt; V3=A*c V3 = 3   9   -6   18</pre>
.*	<p><i>Producto de escalares o de vectores elemento a elemento</i></p> <pre>&gt;&gt; V4=A.*B V4 = 4   -15   -16   12</pre>
.^	<p><i>Potencia de vectores (<math>A.^B = [A(i,j)^{B(i,j)}]</math>, A y B vectores)</i></p> <pre>&gt;&gt; V6=A.^B V6 = 1.0000  0.0041  256.0000  36.0000 &gt;&gt; A.^c ans = 1   27   -8   216 &gt;&gt; c.^A ans = 3.0      27.0000  0.1111  729.0000</pre>
/	<p><i>A./B = [A(i,j)/B(i,j)], siendo A y B vectores [<math>\dim(A)=\dim(B)</math>]</i></p> <pre>&gt;&gt; V7=A./B V7 = 0.2500  -0.6000  -0.2500  3.0000</pre>
.\	<p><i>A.\B = [B(i,j)/A(ij)J, siendo A y B vectores [<math>\dim(A)=\dim(B)</math>]</i></p> <pre>&gt;&gt; V8=A.\B V8 = 4.0000  -1.6667  -4.0000  0.3333</pre>
\	<p><i>A\B = inv(A)*B, siendo A y B matrices</i></p> <pre>&gt;&gt; A=rand(4) A =</pre>

	<pre> 0.6868      0.5269      0.7012      0.0475 0.5890      0.0920      0.9103      0.7361 0.9304      0.6539      0.7622      0.3282 0.8462      0.4160      0.2625      0.6326  &gt;&gt; B=randn(4)  B = </pre> <table border="1"> <tbody> <tr><td>-0.1356</td><td>-0.0449</td><td>-0.0562</td><td>0.4005</td></tr> <tr><td>-1.3493</td><td>-0.7989</td><td>0.5135</td><td>-1.3414</td></tr> <tr><td>-1.2704</td><td>-0.7652</td><td>0.3967</td><td>0.3750</td></tr> <tr><td>0.9846</td><td>0.8617</td><td>0.7562</td><td>1.1252</td></tr> </tbody> </table> <pre> &gt;&gt; A\b  ans = </pre> <table border="1"> <tbody> <tr><td>25.0843</td><td>17.5201</td><td>-1.8497</td><td>7.1332</td></tr> <tr><td>-25.8285</td><td>-17.9297</td><td>2.4290</td><td>-4.6114</td></tr> <tr><td>-4.4616</td><td>-3.1424</td><td>-0.2409</td><td>-2.7058</td></tr> <tr><td>-13.1598</td><td>-8.9778</td><td>2.1720</td><td>-3.6075</td></tr> </tbody> </table>	-0.1356	-0.0449	-0.0562	0.4005	-1.3493	-0.7989	0.5135	-1.3414	-1.2704	-0.7652	0.3967	0.3750	0.9846	0.8617	0.7562	1.1252	25.0843	17.5201	-1.8497	7.1332	-25.8285	-17.9297	2.4290	-4.6114	-4.4616	-3.1424	-0.2409	-2.7058	-13.1598	-8.9778	2.1720	-3.6075
-0.1356	-0.0449	-0.0562	0.4005																														
-1.3493	-0.7989	0.5135	-1.3414																														
-1.2704	-0.7652	0.3967	0.3750																														
0.9846	0.8617	0.7562	1.1252																														
25.0843	17.5201	-1.8497	7.1332																														
-25.8285	-17.9297	2.4290	-4.6114																														
-4.4616	-3.1424	-0.2409	-2.7058																														
-13.1598	-8.9778	2.1720	-3.6075																														
/	<p><i>Cociente escalar o <math>B/A = B*inv(A)</math>, siendo A y B matrices</i></p> <pre> &gt;&gt; B/A  ans = </pre> <table border="1"> <tbody> <tr><td>-4.8909</td><td>0.0743</td><td>4.8972</td><td>-1.6273</td></tr> <tr><td>4.6226</td><td>0.9230</td><td>-4.2151</td><td>-1.3541</td></tr> <tr><td>-10.2745</td><td>1.2107</td><td>10.4001</td><td>-5.4409</td></tr> <tr><td>-9.6925</td><td>-0.1247</td><td>11.1342</td><td>-3.1260</td></tr> </tbody> </table>	-4.8909	0.0743	4.8972	-1.6273	4.6226	0.9230	-4.2151	-1.3541	-10.2745	1.2107	10.4001	-5.4409	-9.6925	-0.1247	11.1342	-3.1260																
-4.8909	0.0743	4.8972	-1.6273																														
4.6226	0.9230	-4.2151	-1.3541																														
-10.2745	1.2107	10.4001	-5.4409																														
-9.6925	-0.1247	11.1342	-3.1260																														
^	<p><i>Potencia de escalares o potencia escalar de matriz (<math>M^p</math>)</i></p> <pre> &gt;&gt; A^3  ans = </pre> <table border="1"> <tbody> <tr><td>3.5152</td><td>2.1281</td><td>3.1710</td><td>1.6497</td></tr> <tr><td>4.0584</td><td>2.3911</td><td>3.7435</td><td>1.9881</td></tr> <tr><td>4.6929</td><td>2.8299</td><td>4.2067</td><td>2.2186</td></tr> <tr><td>3.5936</td><td>2.1520</td><td>3.2008</td><td>1.7187</td></tr> </tbody> </table>	3.5152	2.1281	3.1710	1.6497	4.0584	2.3911	3.7435	1.9881	4.6929	2.8299	4.2067	2.2186	3.5936	2.1520	3.2008	1.7187																
3.5152	2.1281	3.1710	1.6497																														
4.0584	2.3911	3.7435	1.9881																														
4.6929	2.8299	4.2067	2.2186																														
3.5936	2.1520	3.2008	1.7187																														

## Operadores lógicos

MATLAB también incorpora los operadores lógicos habituales utilizando la notación más habitual para los mismos. Habitualmente los resultados de los operadores lógicos suelen ser 1 si cierto y 0 si falso. En la tabla siguiente se muestran estos operadores.

~A	<p><i>Negación lógica (NOT) o complementario de A</i></p> <pre> &gt;&gt; not(2&gt;3)  ans = </pre> <p style="text-align: center;">1</p>
----	-----------------------------------------------------------------------------------------------------------------------------------------

<b>A &amp; B</b>	<i>Conjunción lógica (AND) o intersección de A y B</i>  <pre>&gt;&gt; (2&gt;3) &amp; (5&gt;1)</pre> <pre>ans = 0</pre>
<b>A   B</b>	<i>Disyunción lógica (OR) o unión de A y B</i>  <pre>&gt;&gt; (2&gt;3)    (5&gt;1)</pre> <pre>ans = 1</pre>
<b>xor(A,B)</b>	<i>OR exclusivo (XOR) o diferencia simétrica de A y B (vale 1 si A o B, pero no ambos, son 1)</i>  <pre>&gt;&gt; xor ( (2&gt;3) , (5&gt;1))</pre> <pre>ans = 1</pre>

### Operadores relacionales

Asimismo MATLAB se ocupa de las operaciones relacionales que ejecutan comparaciones elemento a elemento entre dos matrices y devuelven una matriz del mismo tamaño cuyos elementos son ceros si la correspondiente relación es cierta, o unos si la correspondiente relación es falsa. Los operadores relacionales también permiten comparar escalares con vectores o matrices, en cuyo caso se compara el escalar con todos los elementos de la matriz. La tabla siguiente muestra los operadores relacionales de MATLAB.

<	<i>Menor (para complejos sólo afecta a partes reales)</i>  <pre>&gt;&gt; 3&lt;5</pre> <pre>ans = 1</pre>
<=	<i>Menor o igual (sólo afecta a partes reales)</i>  <pre>&gt;&gt; 4&gt;=6</pre> <pre>ans = 0</pre>
>	<i>Mayor (sólo afecta a partes reales)</i>  <pre>&gt;&gt; X=3*ones (3, 3)</pre> <pre>X =</pre>

	<pre> 3   3   3      3   3   3      3   3   3 &gt;&gt; X&gt;[1 2 3; 4 5 6; 1 2 3] ans =     1     1     0     0     0     0     1     1     0 </pre>
>=	<p><b>Mayor o igual (sólo afecta a partes reales)</b></p> <pre> &gt;&gt; X&gt;=[1 2 3; 4 5 6; 1 2 3] ans =     1     1     1     0     0     0     1     1     1 </pre>
x == y	<p><b>Igualdad (afecta a los números complejos)</b></p> <pre> &gt;&gt; x==ones(3, 3) ans =     0     0     0     0     0     0     0     0     0 </pre>
x ~= y	<p><b>Desigualdad (afecta a los números complejos)</b></p> <pre> &gt;&gt; x~=ones(3, 3) ans =     1     1     1     1     1     1     1     1     1 </pre>

## VARIABLES SIMBOLICAS

Hasta ahora hemos manejado siempre variables numéricas. No obstante, MATLAB permite manejar perfectamente el cálculo matemático simbólico, manipular con facilidad y rapidez las fórmulas y expresiones algebraicas y realizar la mayoría de operaciones con las mismas.

Sin embargo, para realizar estas tareas es necesario disponer del módulo de MATLAB *Symbolic Math Toolbox*. Para que una variable o expresión algebraica sea simbólica, previamente hay que declararla como tal con el comando *syms*, con el comando *sym* o introduciéndola entre comillas. En la tabla siguiente se presenta la sintaxis de los comandos de conversión a variables y expresiones simbólicas.

<b>syms x y z...t</b>	<i>Convierte las variables x, y, z, t en simbólicas</i> <pre>&gt;&gt; syms x y &gt;&gt; x+x+y-6*y ans = 2*x - 5*y</pre>
<b>syms x y z...t real</b>	<i>Convierte las variables x, y, z, t en simbólicas con valores reales</i> <pre>&gt;&gt; syms a b c real; &gt;&gt; A = [abc; cab; bca] A = [ a, b, c ] [ c, a, b ] [ b, c, a ]</pre>
<b>syms x y z...t unreal</b>	<i>Convierte las variables x, y, z, t en simbólicas con valores no reales</i>
<b>syms</b>	<i>Lista las variables simbólicas en el espacio de trabajo</i> <pre>&gt;&gt; syms 'A' 'a' 'ans' 'b' 'e' 'x' 'y'</pre>
<b>y=sym('x')</b>	<i>Convierte la variable o número x en simbólica (equivale a syms x)</i> <pre>&gt;&gt; rho = sym('(1 + sqrt(5))/2') rho = 5^(1/2)/2 + 1/2</pre>
<b>y=sym('x',real)</b>	<i>Convierte ax en una variable simbólica real</i>
<b>y=sym('x',unreal)</b>	<i>Convierte ax en una variable simbólica no real</i>
<b>S=sym(A)</b>	<i>Crea un objeto simbólico a partir de A, donde A puede ser una cadena, una escalar, una matriz, una expresión numérica, etc.</i> <pre>&gt;&gt; S=sym([0.5 0.75 1; 0 0.5 0.1; 0.2 0.3 0.4]) S = [ 1/2, 3/4, 1] [ 0, 1/2, 1/10] [ 1/5, 3/10, 2/5]</pre>
<b>S = sym(A,'opcion')</b>	<i>Convierte la matriz, escalar o expresión numérica A a simbólicos según la opción especificada. La opción puede ser f para punto flotante, 'r' para racional, 'e' para formato de error y 'd' para decimal</i>
<b>numeric(x) ó double(x)</b>	<i>Convierte la variable o expresión x a numérica de doble precisión</i>
<b>pretty(expr)</b>	<i>Convierte la expresión simbólica a escritura matemática</i>

	<pre>&gt;&gt; pretty(rho)</pre> $\frac{1}{2} + \frac{1}{2}$
<b>digits</b>	<p><i>Da la precisión actual para variables simbólicas</i></p> <pre>&gt;&gt; digits</pre> <pre>Digits = 32</pre>
<b>digits(d)</b>	<p><i>Sitúa la precisión de las variables simbólicas en d dígitos decimales exactos</i></p> <pre>&gt;&gt; digits(25)</pre> <pre>&gt;&gt; digits</pre> <pre>Digits = 25</pre>
<b>vpa('expr')</b>	<p><i>Resultado numérico de la expresión con los dígitos decimales de precisión situados en digits</i></p> <pre>&gt;&gt; phi = vpa('(1+sqrt(5))/2')</pre> <pre>phi =</pre> <pre>1.618033988749894848204587</pre>
<b>vpa('expr', n)</b>	<p><i>Resultado numérico de la expresión con n dígitos decimales</i></p> <pre>&gt;&gt; phi = vpa('(1+sqrt(5))/2', 10)</pre> <pre>phi =</pre> <pre>1.618033989</pre>
<b>vpa(expr, n)</b>	<p><i>Resultado numérico de la expresión con n dígitos decimales</i></p> <pre>&gt;&gt; phi = vpa((1+sqrt(5))/2, 20)</pre> <pre>phi =</pre> <pre>1.6180339887498949025</pre>

## FUNCIONES SIMBOLICAS Y OPERACIONES FUNCIONALES. FUNCIONES COMPUESTA E INVERSA

En MATLAB es posible definir funciones a medida de una y varias variables mediante la sintaxis  $f = \text{función}$ .

También se puede utilizar la sintaxis  $f = \text{función}$  si todas las variables han sido definidas previamente como simbólicas con `syms`. Posteriormente es posible realizar sustituciones en sus argumentos de acuerdo a la notación que se presenta en la tabla siguiente.

Los resultados suelen simplificarse con los comandos `simplify` y `simple`.

<b>f ='funcion'</b>	<i>Define la función f como simbólica</i>  <pre>&gt;&gt; f1='x^3' f1 = x^3 &gt;&gt; f2='z^2+2*t' f2 = z^2+2*t &gt;&gt; syms x z t &gt;&gt; g1=x^2 g1 = x^2 &gt;&gt; g2=sqrt(x+2*z+exp(t)) g2 = (x + 2*z + exp(t))^(1/2)</pre>
<b>subs(f,a)</b>	<i>Aplica la función f en el punto a</i>  <pre>&gt;&gt; subs(f1,2) ans = 8</pre>
<b>subs(f,variable, valor)</b>	<i>Sustituye en la ecuación de f la variable por el valor</i>  <pre>&gt;&gt; subs(f1,x,3) ans = 27</pre>
<b>subs(f,{x,y,...}, {a,b,...})</b>	<i>Sustituye en la ecuación de f las variables {x,y,...} por los valores {a,b,..}</i>  <pre>&gt;&gt; subs(f2, {z,t}, {1,2}) ans = 5</pre>

Adicionalmente MATLAB implementa varias operaciones funcionales que se resumen en la tabla siguiente:

<b>f+g</b>	<i>Suma las funciones f y g (f+g)</i>  <pre>&gt;&gt; syms x &gt;&gt; f=x^2+x+1; g=2*x^2-x^3+cos(x); h=-x+log(x);</pre>
------------	------------------------------------------------------------------------------------------------------------------------------

	<pre>&gt;&gt; f+g ans = x + cos (x) + 3*x^2 - x^3 + 1</pre>
<b>f+g+h+...</b>	<p><i>Realiza la suma f+g+h+....</i></p> <pre>&gt;&gt; f+g+h ans = cos (x) + log(x) + 3*x^2 - x^3 + 1</pre>
<b>f-g</b>	<p><i>Realiza la diferencia de f y g (f-g)</i></p> <pre>&gt;&gt; f-g ans = x - cos (x) - x^2 + x^3 + 1</pre>
<b>f-g-h-...</b>	<p><i>Realiza la diferencia f-g-h-...</i></p> <pre>&gt;&gt; f-g-h ans = 2*x - cos (x) - log(x) - x^2 + x^3 + 1</pre>
<b>f*g</b>	<p><i>Realiza el producto de f y g (f*g)</i></p> <pre>&gt;&gt; f*g ans = (cos (x) + 2*x^2 - x^3)*(x^2 + x + 1)</pre>
<b>f*g*h*...</b>	<p><i>Realiza el producto f*g*h*...</i></p> <pre>&gt;&gt; f*g*h ans = -(x - log(x))*(cos (x) + 2*x^2 - x^3)*(x^2 + x + 1)</pre>
<b>f/g</b>	<p><i>Realiza el cociente entre f y g (f/g)</i></p> <pre>&gt;&gt; f/g ans = (x^2 + x + 1) / (cos (x) + 2*x^2 - x^3)</pre>
<b>f/g/h/...</b>	<p><i>Realiza el cociente f/g/h/...</i></p> <pre>&gt;&gt; f/g/h ans = -(x^2 + x + 1) / ((x - log(x))*(cos (x) + 2*x^2 - x^3))</pre>
<b>f^k</b>	<p><i>Eleva f a la potencia k (k es un escalar)</i></p> <pre>&gt;&gt; f^2 ans (x^2 + x + 1)^2</pre>

<b>F^g</b>	<i>Eleva una función a otra función (<math>f^g</math>)</i>
	<pre>&gt;&gt; f^g ans = (x^2 + x + 1)^-(cos(x) + 2*x^2 - x^3)</pre>
<b>compose(f,g)</b>	<i>Función compuesta de f y g (<math>f \circ g(x) = f(g(x))</math>)</i>
	<pre>&gt;&gt; compose (f,g) ans = cos(x) + (cos(x) + 2*x^2 - x^3)^2 + 2*x^2 - x^3 + 1</pre>
<b>compose(f,g,u)</b>	<i>Función compuesta de f y g, tomando la expresión u como dominio de f y g</i>
	<pre>&gt;&gt; compose (f,g,h) ans = cos(log(x) - x) + 2*(x - log(x))^2 + (x - log(x))^3 + (cos(log(x)</pre>
<b>Z = finverse(f)</b>	<i>Función inversa de f</i>
	<pre>&gt;&gt; finverse(f) Warning: finverse (x^2 + x + 1) is not unique. ans = (4*x - 3)^(1/2)/2 - 1/2</pre>

### **Comandos que manejan variables en el espacio de trabajo y las almacenan en ficheros**

MATLAB dispone de un grupo de comandos que permiten definir y manejar variables, así como almacenarlas en ficheros con formato propio de MATLAB (extensión *mat*) o en formato ASCII, de modo muy simple. Cuando se realizan cálculos extensos, es conveniente dar nombres a resultados intermedios. Estos resultados intermedios se asignan a variables para hacer más fácil su uso. No olvidemos que el valor asignado a una variable es permanente, hasta que no se cambie expresamente o hasta que no se salga de la presente sesión de trabajo. MATLAB incorpora un grupo de comandos para el manejo de variables entre los que se encuentran los siguientes:

<b>clear</b>	<i>Borra todas las variables del espacio de trabajo</i>
<b>clear(v1,v2,vn)</b> <b>clear('v1' 'v2'..., 'vn')</b>	<i>Borra las variables numéricas especificadas</i>
<b>disp(X)</b>	<i>Muestra un array sin incluir su nombre</i>
<b>length(X)</b>	<i>Muestra la longitud del vector X y, si X es una matriz o array, muestra su mayor dimensión</i>

<b>load</b>	<i>Lee todas las variables del fichero MATLAB.mat Lee todas las variables del fichero .mat especificado Lee las variables X, Y, Z del fichero .mat dado Lee el fichero como ASCII sea cual sea su extensión Lee el fichero como .mat sea cual sea su extensión Devuelve el contenido de un fichero .mat en la variable S</i>
<b>saveas(h,'f.ext')</b> <b>saveas(h,'f','formato')</b>	<i>Salva la figura o modelo h como el fichero f.ext Salva la figura o modelo h como el fichero f con el formato especificado</i>
<b>d = size(X)</b> <b>[m,n] = size(X)</b>  <b>[d1,d2,d3,...,dn] = size(X)</b>	<i>Devuelve en un vector los tamaños de cada dimensión Devuelve las dimensiones de la matriz X como dos variables de nombres m y n Devuelve las dimensiones del array X como variables de nombres d1, d2, ..., dn</i>
<b>who</b> <b>whos</b>  <b>who('-file','fichero')</b> <b>whos('file','fichero')</b> <b>who('var1','var2',...)</b> <b>who('-file','filename','var1','var2',...)</b>  <b>s = who(...)</b>  <b>s = whos(...)</b> <b>who -file filename var1 var2 ...</b>  <b>whos -file filename var1 var2...</b>	<i>Lista las variables del workspace Lista las variables del workspace con sus tamaños y tipos Lista las variables en el fichero .mat dado Lista las variables en el fichero .mat dado y sus tamaños y tipos Lista las variables cadena del workspace dadas Lista las variables cadena especificadas en el fichero .mat dado  Almacena en s las variables listadas Almacena en s las variables con sus tamaños y tipos Lista las variables numéricas especificadas en el fichero .mat dado Lista las variables numéricas especificadas en el fichero .mat dado con sus tamaños y tipos</i>
<b>workspace</b>	<i>Muestra un navegador para gestionar el workspace</i>

El comando *save* es el instrumento esencial para guardar datos en ficheros *.mat* tipo MATLAB (sólo legibles por el programa MATLAB) y en ficheros tipo ASCII (legibles por cualquier aplicación). Por defecto, el almacenamiento de las variables suele realizarse en ficheros con formato MATLAB *.mat*. Para almacenar variables en ficheros con formato ASCII es necesario utilizar el comando *save* con las opciones que se presentan a continuación:

Opción	Modo de almacenamiento de los datos
<b>-append</b>	<i>Las variables se añaden al final del fichero</i>
<b>-ascii</b>	<i>Las variables se almacenan en un fichero en formato ASCII de 8 dígitos</i>
<b>-ascii -double</b>	<i>Las variables se almacenan en un fichero en formato ASCII de 16 dígitos</i>

<b>-ascii -tabs</b>	<i>Las variables se almacenan en un fichero en formato ASCII de 8 dígitos delimitado por tabuladores</i>
<b>-iscii -double -tabs</b>	<i>Las variables se almacenan en un fichero en formato ASCII de 16 dígitos delimitado por tabuladores</i>
<b>-mat</b>	<i>Las variables se almacenan en un fichero .mat binario con formato MATLAB (MATfile)</i>

*Ejercicio 1. Calcular el valor de 7 elevado a la potencia 400 con 500 cifras decimales exactas.*

```
>> vpa '7^400' 500
ans =
1.094500604336113085424254456486662175299754873359706186335419407515439
06316349209002147856846968715280739995373528253861552495710170702637728
89172085286838471044006674397286276116996066357907929105887893308827487
56981780249770882233963982655555969164735367924371346327397193899696906
3052331711311727683195819839003492006097994729312240001*10^338
```

En este caso no son necesarias 500 cifras para expresar el valor exacto del resultado. Se observa que es suficiente con 338 cifras. Si se quiere el resultado con menos cifras exactas que las que tiene el número realmente, MATLAB completa el resultado con potencias de 10. Veamos:

```
>> vpa '7^400' 45
ans =
1.09450060433611308542425445648666217529975487*10^338
```

*Ejercicio 2. Calcular el máximo común divisor y el mínimo común múltiplo de los números 1000, 500 y 625*

```
>> gcd(gcd(1000,500),625)
ans =
125
```

Como la función gcd sólo admite dos argumentos en MATLAB, hemos aplicado la propiedad:  $gcd(a,b,c) = gcd(gcd(a,b),c) = gcd(a,gcd(b,c))$ . Como la propiedad es análoga para el mínimo común múltiplo:  $lcm(a,b,c) = lcm(lcm(a,b),c) = lcm(a,lcm(b,c))$ . Realizamos su cálculo de la forma siguiente:

```
>> lcm(lcm(1000,500),625)
ans =
5000
```

<b>-ascii -tabs</b>	<i>Las variables se almacenan en un fichero en formato ASCII de 8 dígitos delimitado por tabuladores</i>
<b>-ascii -double -tabs</b>	<i>Las variables se almacenan en un fichero en formato ASCII de 16 dígitos delimitado por tabuladores</i>
<b>-raat</b>	<i>Las variables se almacenan en un fichero .mat binario con formato MATLAB (MAT-file)</i>

*Ejercicio 1. Calcular el valor de 7 elevado a la potencia 400 con 500 cifras decimales exactas.*

```
>> vpa '7^400' 500
```

```
1.094500604336113085424254456486662175299754873359706186335419407515439
06316349209002147856846968715280739995373528253861552495710170702637728
8917208528683847104400667439728627616996066357907929105887893308827487
56981780249770882233963982655555969164735367924371346327397193899696906
30523317113111727683195819839003492006097994729312240001*10^338
```

En este caso no son necesarias 500 cifras para expresar el valor exacto del resultado. Se observa que es suficiente con 338 cifras. Si se quiere el resultado con menos cifras exactas que las que tiene el número realmente, MATLAB completa el resultado con potencias de 10. Veamos:

```
>> vpa '7^400' 45
ans =
1.09450060433611308542425445648666217529975487*10^338
```

*Ejercicio 2. Calcular el máximo común divisor y el mínimo común múltiplo de los números 1000, 500 y 625*

```
>> gcd(gcd(1000,500),625)
ans =
125
```

Como la función gcd sólo admite dos argumentos en MATLAB, hemos aplicado la propiedad:  $gcd(a,b,c) = gcd(gcd(a,b),c) = gcd(a,gcd(b,c))$ . Como la propiedad es análoga para el mínimo común múltiplo:  $lcm(a,h,c) = lcm(lcm(a,b),c) = lcm(a,lcm(b,c))$ . rsalizanemos su cálculo de la forma siguiente:

```
>> lcm(lcm(1000,500),625)
ans =
5000
```

*Ejercicio 3. ¿Es primo el número 99.991? Hallar los factores primos del número 135.678.742*

Descomponemos en factores primos ambos números:

```
>> factor(99991)  
ans =  
99991  
  
>> factor(135678742)  
ans =  
2           1699           39929
```

Se observa que 99991 es número primo porque al descomponerlos en factores primos resulta ser el único factor primo de sí mismo. También se observa que el número 135678742 tiene tres factores primos.

*Ejercicio 4. Hallar el resto de la división de  $2^{134}$  entre 3. ¿Es divisible por 17 el número  $2^{32} - 1$ ? Hallar también el número N que al dividirlo por 16, 24, 30 y 32 da de resto 5.*

La primera parte del problema se resuelve como sigue:

```
>> rem (2^134, 3)  
ans =  
0
```

Se observa que  $2^{134}$  es un múltiplo de 3.

Para ver si  $2^{32} - 1$  es divisible por 17, lo descomponemos en factores primos:

```
>> factor (2^32-1)  
ans =  
3           5           17           257           65537
```

Se observa que 17 es uno de sus factores, luego es divisible por 17.

Para resolver la última parte del problema, calculamos el mínimo común múltiplo de todos los números y le sumamos 5.

```
>> N=5+lcm(16,lcm(24,lcm(30,32)))
```

```
N = 485
```

*Ejercicio 5. Calcular el valor que ofrece MATLAB por defecto para el factorial de 100. Calcularlo también con 70 y con 200 cifras significativas.*

```
>> factorial(100)

ans =
9.3326e+157 >>

vpa(factorial(100))

ans =
9.332621544394 410218 8 325 60610857

5*10^157 >> vpa(factorial(100) , 70)

ans =
9.332 6215443944102188325606108575267240944254854960571509166910400408*10^ 157

>> vpa(factorial(100),200)

ans =
9.332621544394410218832560610857526724094425485496057150916691040040799
50642429371486326940304505128980429892969444748982587372043112366414775
61877016501813248*10^157
```

*Ejercicio 6. Obtener en base 5 el resultado de la operación siguiente:*

$$a25aa6 + 6789aba + 35671 + 1100221 - 1250$$

16	12	8	3
----	----	---	---

Como se trata de una operación entre números en sistemas de numeración con bases distintas, habrá que convertirlos todos base 10 y realizar la operación para calcular el resultado en base 10. Posteriormente se pasa el resultado a base 5.

```
>>
R10=base2dec('a25aa6',16)+base2dec('6789aba',12)+base2dec('35671',8) +
base2dec('1100221',3) - 1250

R10 =
190096544

>> R5=dec2base(R10,5)
```

R5 =

*Ejercicio 7. Obtener, en base 13, el resultado de la operación siguiente:*

$$(666551)(aal99800a)+(fffaal25)/(33331+6)$$

Utilizamos la estrategia del ejercicio anterior. En primer lugar, realizamos la operación con todos los números a base 10 y por último pasamos el resultado a base 13.

```

>>
R10=vpa(base2dec('666551',7)*base2dec('aal199800a',11)+79*base2dec('fffa
aa125',16)/(base2dec('33331',4)+6))

R10 =
275373340490851.53125

>> R13=dec2base(275373340490852, 13)

R13 =
BA867963C149

```

*Ejercicio 8. Realizar las siguientes operaciones con números racionales:*

- a)  $3/5 + 2/5 + 7/5$   
 b)  $1/2 + 1/3 + 1/4 + 1/5 + 1/6$   
 c)  $1/2 - 1/3 + 1/4 - 1/5 + 1/6$   
 d)  $(2/3 - 1/6) - (4/5 + 2/3 + 1/3) + (4/5 - 7/7)$

a)

```
>> format rat  
>> 375+2/5+7/5  
ans =  
    12/5
```

b)

```
>> 1/2+1/3+1/4+1/5+1/6  
ans =  
    29/20
```

c)

```
>> 1/2-1/3+1/4-1/5+1/6  
ans =  
    23/60
```

d)

```
>> (2/3-1/6)-(4/5+2+1/3)+(4-5/7)
ans =
137/210
```

Alternativamente, también se pueden efectuar las operaciones como sigue:

```
>> simplify(sym(3/5+2/5+7/5))
ans =
12/5
>> simplify(sym(1/2+1/3+1/4+1/5+1/6))
ans =
29/20
>> simplify(sym(1/2-1/3+1/4-1/5+1/6))

23/60
>> simplify(sym((2/3-1/6)-(4/5+2+1/3)+(4-5/7)))
ans =
137/210
```

*Ejercicio 9. Realizar las siguientes operaciones racionales:*

- a)  $3/a + 2/a + 7/a$
- b)  $1/(2a) + 1/(3a) + 1/(4a) + 1/(5a) + 1/(6a)$
- c)  $1/(2a) + 1/(3b) + 1/(4a) + 1/(5b) + 1/(6c)$

Al tratarse de operaciones con expresiones que contienen la variable simbólica  $a$ , es necesario anteponer el comando **syms a** para declarar la variable  $a$  como simbólica, y utilizar después **simplify o simple**.

a)

```
>> syms a
>> simplify(3/a+2/a+7/a)
ans =
12/a
>> 3/a+2/a+7/a
ans =
12/a
>> Ra=simplify(3/a+2/a+7/a)
```

```
Ra =  
12/a
```

b)

```
>> simplify(1/(2*a)+1/(3*a)+1/(4*a)+1/(5*a)+1/(6*a))
```

```
ans =  
29/(20*a)
```

```
>> Rb=simple(1/(2*a)+1/(3*a)+1/(4*a)+1/(5*a)+1/(6*a))
```

```
Rb =  
29/(20*a)
```

c)

```
>> syms a b c  
>> pretty(simplify(1/(2*a)+1/(3*b)+1/(4*a)+1/(5*b)+1/(6*c)))
```

$$\frac{3}{4a} + \frac{8}{15b} + \frac{1}{6c}$$

```
>> pretty(simple(1/(2*a)+1/(3*b)+1/(4*a)+1/(5*b)+1/(6*c)))
```

$$\frac{3}{4a} + \frac{8}{15b} + \frac{1}{6c}$$

*Ejercicio 10. Simplificar al máximo las siguientes operaciones racionales:*

- a)  $(1-a^9)(1-a^3)$
- b)  $(3a+2a+7a)/(a^3+a)$
- c)  $1/(1+a)+1/(1+a)^2+1/(1+a)^3$
- d)  $1+a/(a+b)+a^2/(a+b)^2$

a)

```
>> syms a b  
>> pretty(simple((1-a^9)/(1-a^3)))  

$$\frac{a^6 + a^3 + 1}{a^6 - a^3 + 1}$$

```

b)

```
>> pretty(simple((3*a+2*a+7*a)/(a^3+a)))  

$$\frac{12}{a^2 + 1}$$

```

c)

```
>> pretty (simple((1/(1+a)+1/(1+a)^2+1/(1+a)^3)))
```

$$\frac{a^2 + 3a + 3}{(a + 1)^3}$$

d)

```
>> pretty(simple((1+a/(a+b)+ a^2/(a+b)^2)))
```

$$\frac{2a^2 + ba + 1}{(a + b)^2}$$

*Ejercicio 11. Realizar las siguientes operaciones con números irracionales:*

a)  $3\sqrt{a} a + 2\sqrt{a} - 5\sqrt{a} + 7\sqrt{a}$

b)  $\sqrt{2} + 3\sqrt{2} - \sqrt{2}/2$

c)  $4a^{1/3} - 3b^{1/3} - 5a^{1/3} - 2b^{1/3} + ma^{1/3}$

d)  $\sqrt{3}a \sqrt{27a}$

e)  $\sqrt{a} \sqrt[3]{a}$

f)  $\sqrt[5]{(a^5\sqrt{a})}$

a)

Utilizamos el comando *simplify* o el comando *simple*.

```
>> syms a b m
>> pretty(simplify(3*sqrt(a) + 2*sqrt(a) - 5*sqrt(a) + 7*sqrt(a)))

$$7\sqrt{a}$$

```

b)

```
>> pretty(simplify(sym(sqrt(2))+3*sqrt(2)-sqrt(2)/2))
```

$$\frac{7\sqrt{2}}{2}$$

c)

```
>> syms a b m  
>> pretty (simplify(4*a^(1/3)-3*b^(1/3)-5*a^(1/3)-2*b^(1/3)+m*a^(1/3)))
```

$$\frac{1}{a^{\frac{3}{m}}} - \frac{1}{a^{\frac{3}{2}}} - 5a^{\frac{1}{3}} - 2b^{\frac{1}{3}}$$

d)

```
>> pretty(simplify(sqrt(3*a)*sqrt(27*a))) 9 a
```

e)

```
>> pretty (simplify (a^(1/2)*a^(1/3)))
```

$$\frac{a^{\frac{5}{6}}}{a}$$

f)

```
>> pretty (simplify (sqrt (a*(a^(1/5)))))
```

$$\frac{a^{\frac{6}{5}}}{a}$$

*Ejercicio 12. Realizar en las siguientes expresiones irracionales la racionalización de denominadores:*

$$a) \frac{2}{\sqrt{2}} \quad b) \frac{3}{\sqrt{3}} \quad c) \frac{a}{\sqrt{a}}$$

En estos casos de racionalización, el simple uso del comando *simplify* resuelve los problemas. También se puede usar el comando *radsimp*.

a)

```
>> simplify(sym( 2/sqrt(2)))  
ans =  
2^(1/2)
```

b)

```
>> simplify(sym(3/sqrt(3)))  
ans =  
3^(1/2)
```

c)

```
>> syms a
>> simplify (sym(a/sqrt(a)))
ans =
a^(1/2)
```

*Ejercicio 13. Dadas las variables vectoriales  $a=[\pi, 2\pi, 3\pi, 4\pi, 5\pi]$  y  $b=[e, 2e, 3e, 4e, 5e]$  calcular  $c=\operatorname{Sen}(a)+b$ ,  $d=\operatorname{Cosh}(a)$ ,  $e=\ln(b)$ ,  $f=c*d$ ,  $g=c/d$ ,  $h=d^2$ .*

```
a=pi,2*pi,3*pi,4*pi,5*pi],b=[exp(1),2*exp(1),3*exp(1),4*exp(1),5*exp(1)]
a=
3.1416      6.2832      9.4248      12.5664      15.7080

b=
2.7183      5.4366      8.1548      10.8731      13.5914

>> c=sin(a)+b,d=cosh(a),e=log(b),f=c.*d,g=c./d,h=d.^2

c=
2.7183      5.4366      8.1548      10.8731      13.5914

d=
1.0e+006 *
0.0000      0.0003      0.0062      0.1434      3.3178

e=
1.0000      1.6931      2.0986      2.3863      2.6094

f=
1.0e+007 *
0.0000      0.0001      0.0051      0.1559      4.5094

g=
0.2345      0.0203      0.0013      0.0001      0.0000

h=
1.0e+013 *
0.0000      0.0000      0.0000      0.0021      1.1008
```

*Ejercicio 14. Dado el vector de los 10 primeros números naturales, hallar:*

*Su sexto elemento*

*Sus elementos situados entre el cuarto y el séptimo ambos inclusive*

*Sus elementos situados entre el segundo y el noveno ambos inclusive de tres en tres*

*Los elementos del apartado anterior, pero de mayor a menor*

```
>>x=(1:10)
```

```
x =
```

```
1 2 3 4 5 6 7 8 9 10
```

```
>>x(6)
```

```
ans =
```

```
6
```

Hemos obtenido el sexto elemento del vector x.

```
>> x(4:7)
```

```
ans =
```

```
4 5 6 7
```

Hemos obtenido los elementos del vector x situados entre el cuarto y el séptimo, ambos inclusive.

```
>> x(2:3:9)
```

```
ans =
```

```
2 5 8
```

Hemos obtenido los elementos del vector x situados entre el segundo y el noveno, ambos inclusive, pero separados de tres en tres unidades.

```
>> x(9:-3:2)
```

```
ans =
```

```
9 6 3
```

Hemos obtenido los elementos del vector x situados entre el noveno y el segundo, ambos inclusive, pero separados de tres en tres unidades y empezando por el noveno.

*Ejercicio 15. Dada la matriz  $2 \times 3$  cuyas filas son los seis primeros números impares consecutivos, se pide:*

*Anular su elemento  $(2,3)$ , transponerla y adosarle la matriz identidad de orden  $3$  a su derecha.*

*Sillamamos  $C$  a la matriz obtenida anteriormente, construir una matriz  $D$  extrayendo las columnas impares de la matriz  $C$ , una matriz  $E$  formada por la intersección de las dos primeras filas de  $C$  y sus columnas tercera y quinta, y una matriz  $F$  formada por la intersección de las dos primeras filas y las tres últimas columnas de la matriz  $C$ .*

*Construir la matriz diagonal  $G$  tal que los elementos de su diagonal principal son los mismos que los de la diagonal principal de  $D$ .*

*Construimos la matriz  $H$ , formada por la intersección de la primera y tercera filas de  $C$  y sus segundas, tercera y quintas columnas.*

Consideramos en primer lugar la matriz  $2 \times 3$  cuyas filas son los 6 primeros impares consecutivos:

```
>> A=[1 3 5; 7 9 11]
```

```
A =
```

$$\begin{matrix} 1 & 3 & 5 \\ 7 & 9 & 1 \end{matrix}$$

Ahora vamos a anular el elemento  $(2,3)$ , o sea, su último elemento:

```
>> A(2,3)=0
```

```
A =
```

$$\begin{matrix} 1 & 3 & 5 \\ 7 & 9 & 0 \end{matrix}$$

A continuación, consideramos la matriz  $B$  transpuesta de  $A$ :

```
>> B=A'
```

```
B =
```

$$\begin{matrix} 1 & 7 \\ 3 & 9 \\ 5 & 0 \end{matrix}$$

Ahora construimos una matriz  $C$ , formada por la matriz  $B$  y la matriz identidad de orden  $3$  adosada a su derecha:

```
>> C=[B eye(3)]
```

```
C =
```

1	7	1	0	0
3	9	0	1	0
5	0	0	0	1

Vamos a construir una matriz  $D$  extrayendo las columnas impares de la matriz  $C$ , una matriz  $E$  formada por la intersección de las dos primeras filas de  $C$  y sus columnas tercera y quinta, y una matriz  $F$  formada por la intersección de las dos primeras filas y las tres últimas columnas de la matriz  $C$ :

```
>> D=C(:,1;2:5)
```

$D =$

1	1	0
3	0	0
5	0	1

```
>> E=C([1 2],[3 5])
```

$E =$

1	0
0	0

```
>> F=C([1 2],3:5)
```

$F =$

1	0	0
0	1	0

Ahora construimos la matriz diagonal  $G$  tal que los elementos de su diagonal principal son los mismos que los de la diagonal principal de  $D$ :

```
>> G=diag(diag(D))
```

$G =$

1	0	0
0	0	0
0	0	1

A continuación, construimos la matriz  $H$ , formada por la intersección de la primera y tercera filas de  $C$  y sus segunda, tercera y quinta columnas:

```
>> H=C([1 3], [2 3 5])
```

$H =$

7	1	0
0	0	1

*Ejercicio 16. Construir una matriz I formada por la matriz identidad de orden 5x4 y las matrices nula y unidad del mismo orden adosadas a su derecha. A continuación, extraer la primera fila de I y, por último, formar la matriz J con las filas impares y las columnas pares de Y y calculamos su orden (tamaño).*

*Adicionalmente, construir una matriz aleatoria K de orden 3x4 e invertir primero el orden de sus filas, después el orden de sus columnas y después el orden de sus filas y columnas a la vez. Por último, hallar la matriz L de orden 4x3 cuyas columnas resultan de tomar los elementos de las columnas de K consecutivamente.*

```
>> D=C(:,1:2:5)

D =
    1     1     0
    3     0     0
    5     0     1

>> E=C([1 2], [3 5])

E=
    1     0
    0     0

>> F=C([1 2], 3:5)

F=
    1     0     0
    0     1     0
```

Ahora construimos la matriz diagonal G tal que los elementos de su diagonal principal son los mismos que los de la diagonal principal de D:

```
>> G=diag(diag(D))
```

```
G=
```

```
    1     0     0
    0     0     0
    0     0     1
```

A continuación, construimos la matriz H, formada por la intersección de la primera y tercera filas de C y sus segunda, tercera y quinta columnas:

```
>> H=C([1 3], [2 3 5])
```

```
H=
```

```
    7     1     0
    0     0     1
```

Ahora construimos una matriz  $I$  formada por la matriz identidad de orden  $5 \times 4$  y las matrices nula y unidad del mismo orden adosadas a su derecha. A continuación, extraemos la primera fila de  $I$  y, por último, formamos la matriz  $J$  con las filas impares y las columnas pares de  $Y$  y calculamos su orden (tamaño).

```
>> I = [eye(5,4) zeros(5,4) ones(5,4)]  
  
ans =  
  
1 0 0 0 0 0 0 0 1 1 1 1  
0 1 0 0 0 0 0 0 1 1 1 1  
0 0 1 0 0 0 0 0 1 1 1 1  
0 0 0 1 0 0 0 0 1 1 1 1  
0 0 0 0 0 0 0 0 1 1 1 1  
  
>> I(1,:)  
  
ans =  
  
1 0 0 0 0 0 0 0 1 1 1  
  
>>  
  
J=I(1:2:5,2:2:12)  
  
0 0 0 0 1 1  
0 0 0 0 1 1  
0 0 0 0 1 1  
  
>> size(J)  
  
ans =  
  
3 6
```

A continuación, construimos una matriz aleatoria  $K$  de orden  $3 \times 4$  e invertimos primero el orden de sus filas, después el orden de sus columnas y después el orden de sus filas y columnas a la vez. Por último, hallamos la matriz  $L$  de orden  $4 \times 3$  cuyas columnas resultan de tomar los elementos de las columnas de  $K$  consecutivamente.

```
>>  
  
K =  
  
0.5269 0.4160 0.7622 0.7361  
0.0920 0.7012 0.2625 0.3282  
0.6539 0.9103 0.0475 0.6326  
  
>> K(3:-1:1,:)  
  
ans =  
  
0.6539 0.9103 0.0475 0.6326  
0.0920 0.7012 0.2625 0.3282  
0.5269 0.4160 0.7622 0.7361  
  
>> K(:,4:-1:1)
```

```

0.7361      0.7622      0.4160      0.5269
0.3282      0.2625      0.7012      0.0920
0.6326      0.0475      0.9103      0.6539

>> K(3:-1;1,4:-1:1)

ans =

0.6326      0.0475      0.9103      0.6539
0.3282      0.2625      0.7012      0.0920
0.7361      0.7622      0.4160      0.5269

>> L=reshape(K,4,3)

L =

0.5269      0.7012      0.0475
0.0920      0.9103      0.7361
0.6539      0.7622      0.3282
0.4160      0.2625      0.6326

```

*Ejercicio 17. Dada la matriz cuadrada de orden 3, cuyas filas son los 9 primeros números naturales, obtener su inversa, su transpuesta y su diagonal. Trasformarla en una matriz triangular inferior y en otra superior y rotarla 90 grados. Obtener la suma de los elementos de la primera fila y la suma de los elementos de la diagonal. Extraer la submatriz cuya diagonal son los elementos  $a_{11}$  y  $a_{22}$  y extraer también la submatriz cuyos elementos de la diagonal son  $a_{11}$  y  $a_{33}$ .*

```

>> M=[1,2,3;4,5,6;7,8,9]

M =

1      2      3
4      5      6
7      8      9

>> A=inv(M)

Warning: Matrix is close to singular or badly scaled.
          Results may be inaccurate. RCOND = 2.937385e-018

A =

1.0e+016 *

0.3152   -0.6304    0.3152
-0.6304    1.2609   -0.6304
  0.3152   -0.6304    0.3152

```

```
1      4      7  
2      5      8  
3      6      9
```

```
>> V=diag(M)
```

```
V =
```

```
1  
5  
9
```

```
>> TI=tril(M)
```

```
TI =
```

```
1      0      0  
4      5      0  
7      8      9
```

```
>> TS=triu(M)
```

```
TS =
```

```
1      2      3  
0      5      6  
0      0      9
```

```
>> TR=rot90(M)
```

```
TR =
```

```
3      6      9  
2      5      8  
1      4      7
```

```
>> s=M(1,1)+M(1,2)+M(1,3)
```

```
s =
```

```
6
```

```
>>
```

```
sd=M(1,1)+M(2,2)+M(3,3)
```

```
sd =  
15
```

```
>> SM=M(1:2,1:2)
```

```
SM =
```

```
1      2  
4      5
```

```
>> SM1=M([1 3],[1 3])
```

```
SM1 =
```

```
1      3  
7      9
```

*Ejercicio 18. Dada la matriz cuadrada de orden 3, cuyas filas son los 9 primeros números naturales (sin el cero), identificar sus valores menores que 5.*

*Si consideramos ahora el vector cuyos elementos son los 9 primeros números naturales (sin el cero) y el vector con los mismos elementos de mayor a menor, identificar los valores resultantes de restar a los elementos del segundo vector la cantidad 1 si el correspondiente elemento del primer vector es mayor que 2, o la cantidad 0 si es menor o igual que 2.*

```
>>X=5*ones (3,3); X>= [1 2 3; 4 5 6; 7 8 9]
```

**ans =**

1	1	1
1	1	0
0	0	0

Los elementos de la matriz  $X$  que son mayores o iguales que los de la matriz  $[1 \ 2 \ 3; \ 4 \ 5 \ 6; \ 7 \ 8 \ 9]$  se corresponden con un 1 en la matriz respuesta. El resto de los elementos se corresponden con un 0 (el resultado de la operación sería el mismo si comparáramos el escalar 5 con la matriz  $[1 \ 2 \ 3; \ 4 \ 5 \ 6; \ 7 \ 8 \ 9]$ , mediante la expresión  $>> X=5; X>=[1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$ ).

```
>> X=5; X>= [1 2 3; 4 5 6; 7 8 9]
```

**ans =**

1	1	1
1	1	0
0	0	0

A continuación, resolvemos la segunda parte del ejercicio:

```
>> A=1:9, B=10-A, Y=A>4, Z=B-(A>2)
```

**A =**

1	2	3	4	5	6	7	8	9
---	---	---	---	---	---	---	---	---

**B =**

9	8	7	6	5	4	3	2	1
---	---	---	---	---	---	---	---	---

**Y =**

0	0	0	0	1	1	1	1	1
---	---	---	---	---	---	---	---	---

**Z =**

9	8	6	5	4	3	2	1	0
---	---	---	---	---	---	---	---	---

Los valores de  $Y$  iguales a 1 se corresponden con los elementos de  $A$  mayores que 4. Los valores de  $Z$  resultan de restar a los correspondientes elementos de  $B$  la cantidad 1 si el correspondiente elemento de  $A$  es mayor que 2, o la cantidad 0 si el correspondiente elemento de  $A$  es menor o igual que 2.

*Ejercicio 19. Si consideramos ahora el vector  $A$  cuyos elementos son los 9 primeros números naturales (sin el cero), identificar sus valores mayores que 2 y menores que 6.*

*Si llamamos  $P$  al vector de la misma dimensión que  $A$  que asocia un 1 a los resultados identificados en el apartado anterior y un cero al resto, identificar cuándo  $A$  o  $P$ , pero no ambos, valen 1.*

```
>> A=1:9; P=(A>2) & (A<6)
```

```
P =
```

```
0     0     1     1     1     0     0     0     0
```

Devuelve 1 cuando  $A$  es mayor que 2 y menor que 6, y devuelve 0 en caso contrario.

Ahora resolvemos el segundo apartado del problema.

```
>> P=(A>=1) & (A<6), xor(A, P)
```

```
P =
```

```
1     1     1     1     1     0     0     0     0
```

```
ans =
```

```
0     0     0     0     0     1     1     1     1
```

Los unos del último resultado son la solución pedida.

*Ejercicio 20. Hallar la matriz diferencia entre una matriz aleatoria cuadrada de orden 4 y una matriz aleatoria normal de orden 4. Calcular la transpuesta y la inversa de la citada diferencia. Comprobar que la inversa está correctamente calculada.*

```
>> A=rand(4)-randn(4)
```

```
A =
```

```
0.9389   -0.0391    0.4686    0.6633  
-0.5839    1.3050   -0.0698   1.2727  
-1.2820   -0.4387   -0.5693  -0.0881  
-0.5038   -1.0834    1.2740   1.2890
```

Calculamos la inversa y la multiplicamos por la matriz inicial para comprobar que resulta la matriz identidad.

```
>> B=inv(A)

B =
0.9630    -0.1824     0.1288    -0.3067
-0.8999     0.4345    -0.8475    -0.0239
-1.6722     0.0359    -1.5242     0.7209
1.2729     0.2585     0.8445    -0.0767

>> A*B

ans =
1.0000    0.0000      0    -0.0000
0       1.0000    0.0000    -0.0000
-0.0000   -0.0000    1.0000    0.0000
0.0000   -0.0000    0.0000    1.0000

>> B*A

ans =
1.0000    0.0000    0.0000    0.0000
-0.0000    1.0000   -0.0000    0.0000
0.0000   -0.0000    1.0000    0.0000
-0.0000   -0.0000      0    1.0000

>> A'

ans =
0.9389   -0.5839   -1.2820   -0.5038
-0.0391    1.3050   -0.4387   -1.0834
0.4686   -0.0698   -0.5693    1.2740
0.6633    1.2727   -0.0881    1.2890
```

*Ejercicio 21. Dada la función  $f(x) = x^3$  calcular  $f(2)$  y  $f(b+2)$ . Consideramos ahora la función de dos variables  $f(a,b) = a+b$  y queremos calcular  $f(4,b), f(a,5)$  y  $f(3,5)$ .*

```
>> f='x^3'
f =
x^3
>> A=subs(f,2)
A =
8
>> syms b
```

```
>> B=subs(f,b+2)

B =

(b+2)^3

>> syms a b
>> subs(a+b,a,4)

ans =

4+b

>> subs(a+b,{a,b},{3,5})

ans =

8
```

*Ejercicio 22. En el ejemplo siguiente, dadas las funciones  $f(x)=x^2+x$ ,  $g(x)=x^3+1$  y  $h(x)=\operatorname{Sen}(x)+\operatorname{Cos}(x)$ , calcularemos  $(f+g)(x)$ ,  $(f-g+h)(x)$ ,  $(f/g)(x)$ ,  $f(g(x-1))$ ,  $f(h(\pi/3))$  y  $f(g(h(\operatorname{Sen}(x))))$ .*

```
>> syms x
>> f=x^2;g=x^3+1; h=sin(x)+cos(x);
>> suma=f+g

suma =

x^2+x^3+1

>> combinada= f-g+h

combinada =

x^2-x^3-1+sin (x) +cos (x)

>> cociente=f/g cociente = x^2/(x^3 + 1)

>> compuesta=subs(compose(g,f),x-1)

compuesta =

(x-1)^6+1

>> compuestal=subs(compose(f,h),pi/3)

compuestal =

1.8660
```

```
>> compuestal=subs(compose(f,h), 'pi/3')
compuestal =
(sin((pi/3))+cos((pi/3)))^2
>> compuesta2=subs(compose(f,compose(g,h)), 'sin(x)')
compuesta2 =
((sin(sin(x))+cos(sin(x)))^3+1)^2
```

*Ejercicio 23. Hallar la función inversa de  $f(x) = \operatorname{Sen}(x^2)$  y comprobar que el resultado es correcto.*

```
>> syms x
>> f=sin(x^2)
f =
sin (x ^ 2)

>> g=finverse(f)
g =
asin(x)^(1/2)

>> compose(f,g)
ans
=
x
```

*Ejercicio 24. Dadas las funciones  $f(x)=\sin(\cos(x^{1/2}))$  y de  $g(x)=\sqrt{\tan(x^2)}$  calcular la compuesta de  $f$  y  $g$  y la compuesta de  $g$  y  $f$ . Calcular también las funciones inversas de  $f$  y  $g$ .*

```
>> syms x, f= sin(cos(x^(1/2)));
>> g=sqrt (tan (x^2));
>> simple(compose(f,g))
ans =
sin (cos(tan(x^2)^(1/4)))
>> simple(compose(g,f))
ans =
tan(sin(cos(x^(1/2)))^2)^(1/2)
>> F=finverse(f)
```

```
F =
acos(asin(x))^2
>> G=finverse(g)
G =
atan(x^2)^(1/2)
```

*Ejercicio 25. Dada la función  $h$  definida por:  $h(x,y) = (\cos(x^2-y^2), \sin(x^2-y^2))$ ; calcular  $h(1,2)$ ,  $h(-\pi,\pi)$  y  $h(\cos(a^2), \cos(1-a^2))$ .*

```
>> syms x y a
>> h=[cos(x^2-y^2), sin(x^2-y^2)]
h =
[ cos(x^2 - y^2), sin(x^2 - y^2) ]
>> subs(h,{x,y},{1,2})
ans =
-0.9900 -0.1411 >>
subs(h,{x,y},{-pi,pi}) ans
=
1 0
>> pretty (simple (subs (h,{x,y},{cos(a^2),cos(1-a^2)})))
+-          22          2 2          2 2          2 2  - +
|  cos(- cos(1-a) + cos(a ) ), sin(-cos(1 - a) + cos(a ) ) |  - +
+-
```

*Ejercicio 26. Generar un vector que sea una matriz fila de 10 números aleatorios uniformes y generar otro vector que sea la matriz de unos de orden 10. Guardar ambos vectores como variables en un archivo de formato MATLAB y en otro archivo de formato ASCII. Por último, leer en MATLAB los dos archivos guardados previamente.*

```
>> p = rand(1, 10)
P =
Columns 1 through 9
0.6797    0.6551    0.1626    0.1190    0.4984    0.9597    0.3404
0.5853    0.2238
Column 10
0.7513
```

```
>> q = ones (10)

q =

1   1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1   1
1   1   1   1   1   1   1   1   1
```

Ahora guardamos los resultados anteriores como variables de un archivo en formato MATLAB (extensión *.mat*) de nombre *archivo1*. También guardamos los resultados anteriores como variables de un archivo en formato ASCII (extensión *.txt*) de nombre *archivo2.txt*.

```
>>save('archivo1', 'p', 'q')
>>save archivo1 p q
>>save('archivo2.txt', 'p', 'q', '-ASCII')
>>save archivo2.txt p q -ASCII
```

Posteriormente, estos archivos podrían leerse e incorporar sus variables al espacio de trabajo de la siguiente forma:

```
>>load archivo2.txt -ASCII
>>load archivo1
```

## CAPITULO 3

# NUMEROS COMPLEJOS Y FUNCIONES DE VARIABLE COMPLEJA

---

### NÚMEROS COMPLEJOS

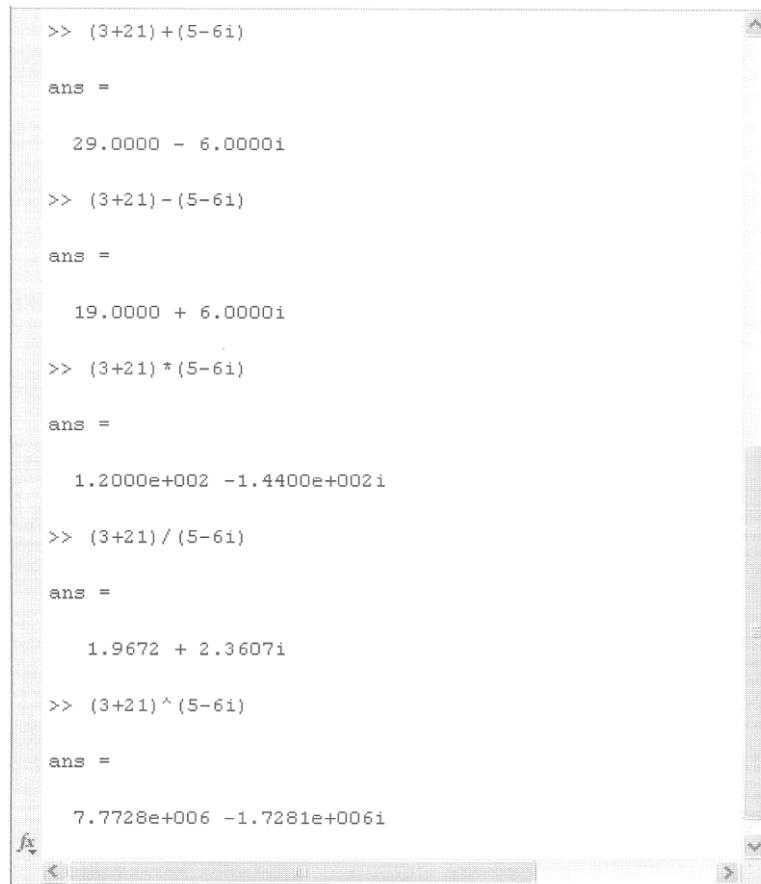
MATLAB implementa de modo sencillo el trabajo con números complejos en forma binómica representando la unidad imaginaria mediante los símbolos  $i$  o  $j$ .

Ya sabemos que un número complejo en forma binómica se representa en la forma  $a+bi$  o  $a+bj$ . No es necesario incluir el símbolo del producto (el asterisco) antes de la unidad imaginaria, aunque si se incluye, todo funciona correctamente. Lo que sí hay que tener presente es que no se pueden introducir espacios entre el número de la parte imaginaria y la unidad imaginaria  $i$ .

Los números complejos pueden tener parte real simbólica o parte imaginaria simbólica o ambas, realizándose las operaciones tanto en modo exacto como en modo decimal con la precisión que se fije con el comando *format*. Por lo tanto, también será posible trabajar con los números complejos en formato racional exacto a través del comando *format rat*.

En cuanto a las operaciones habituales (suma, diferencia, producto, división y potenciación) con números complejos, se realizan de la forma habitual. La Figura 2-1 muestra ejemplos.

Evidentemente, como los números reales son un subconjunto de los números complejos, las funciones de variable compleja también serán válidas para variable real.



```

>> (3+2i)+(5-6i)

ans =
    29.0000 - 6.0000i

>> (3+2i)-(5-6i)

ans =
    19.0000 + 6.0000i

>> (3+2i)*(5-6i)

ans =
    1.2000e+002 -1.4400e+002i

>> (3+2i)/(5-6i)

ans =
    1.9672 + 2.3607i

>> (3+2i)^(5-6i)

ans =
    7.7728e+006 -1.7281e+006i

```

Figura 2-4

## FUNCIONES GENERALES DE VARIABLE COMPLEJA

MATLAB dispone de una gama de funciones generales de variable compleja predefinidas, que por supuesto serán válidas para variables racionales, irracionales y enteras. En los párrafos siguientes se presentan las más importantes.

### *Funciones trigonométricas de variable compleja*

A continuación se presenta una tabla con las funciones trigonométricas de variable compleja y sus inversas que incorpora MATLAB ilustradas con ejemplos.

<i>Función</i>	<i>Inversa</i>
$\sin(z)$ Seno $>> \sin(5-6i)$	$\text{asin}(z)$ Arco seno $>> \text{asin}(1-i)$

ans = -1.9343e+002 -5.7218e+001i	ans = 0.6662 - 1.0613i
<b>cos(z) Coseno</b>  <pre>&gt;&gt; cos (3+4i) ans = -27.0349 - 3.8512i</pre>	<b>acos(z) Arco coseno</b>  <pre>&gt;&gt; acos (-i) ans = 1.5708 + 0.8814i</pre>
<b>tan(z) Tangente</b>  <pre>&gt;&gt; tan(pi/4i) ans = 0 - 0.6558i</pre>	<b>atan(z) y atan2(z) Arco tangente</b>  <pre>&gt;&gt; atan(-pi*1) ans = 1.5708 - 0.3298i</pre>
<b>csc(z) Cosecante</b>  <pre>&gt;&gt; csc(1-i) ans = 0.6215 + 0.3039i</pre>	<b>acsc(z) Arco cosecante</b>  <pre>&gt;&gt; acsc(2i) ans = 0 - 0.4812i</pre>
<b>sec(z) Secante</b>  <pre>&gt;&gt; sec (-i) ans = 0.6481</pre>	<b>asec(z) Arco secante</b>  <pre>&gt;&gt; asec(0.6481+0i) ans = 0 + 0.9999i</pre>
<b>cot(z) Cotangente</b>  <pre>&gt;&gt; cot (-j) ans = 0 + 1.3130i</pre>	<b>acot(z) Arco cotangente</b>  <pre>&gt;&gt; acot(1-6j) ans = 0.0277 + 0.1635i</pre>

### Funciones hiperbólicas de variable compleja

A continuación se presenta una tabla con las funciones hiperbólicas de variable compleja y sus inversas que incorpora MATLAB ilustradas con ejemplos.

Función	Inversa
<b>sinh(z) Seno hiperbólico</b>  <pre>&gt;&gt; sinh(1+i) ans = 0.6350 + 1.2985i</pre>	<b>asinh(z) Arco seno hiperbólico</b>  <pre>&gt;&gt; asinh(0.6350 + 1.2985i) ans = 1.0000 + 1.0000i</pre>
<b>cosh(z) Coseno hiperbólico</b>  <pre>&gt;&gt; cosh(1-i)</pre>	<b>acosh(z) Arco coseno hiperbólico</b>  <pre>&gt;&gt; acosh(0.8337 - 0.9889i)</pre>

ans = 0.8337 - 0.9889i	ans = 1.0000 - 1.0000i
<b>tanh(z) Tangente hiperbólica</b>  <pre>&gt;&gt; tanh(3-5i) ans = 1.0042 + 0.0027i</pre>	<b>atanh(z) Arco tangente hiperbólica</b>  <pre>&gt;&gt; atanh(3-4i) ans = -0.0263 - 1.5708i</pre>
<b>csch(z) Cosecante hiperbólica</b>  <pre>&gt;&gt; csch(i) ans = 0 - 1.1884i</pre>	<b>acsch(z) Arco cosecante hiperbólica</b>  <pre>&gt;&gt; acsch(- 1.1884i) ans = 0 + 1.0000i</pre>
<b>sech(z) Secante hiperbólica</b>  <pre>&gt;&gt; sech(i^i) ans = 0.9788</pre>	<b>asech(z) Arco secante hiperbólica</b>  <pre>&gt;&gt; asech(5-0i) ans = 0 + 1.3694i</pre>
<b>coth(z) Cotangente hiperbólica</b>  <pre>&gt;&gt; coth(9+i) ans = 1.0000 - 0.0000i</pre>	<b>acoth(z) Arco cotangente hiperbólica</b>  <pre>&gt;&gt; acoth(1-i) ans = 0.4024 + 0.5536i</pre>

### Funciones exponenciales y logarítmicas de variable compleja

A continuación se presenta una tabla con las funciones exponenciales y logarítmicas que incorpora MATLAB ilustradas con ejemplos.

Función	Significado
<b>exp(z)</b>	<b>Función exponencial en base e (<math>e^x</math>)</b>  <pre>&gt;&gt; exp(1-i) ans = 1.4687 - 2.2874i</pre>
<b>log(x)</b>	<b>Función logaritmo en base e de x</b>  <pre>&gt;&gt; log(1.4687 - 2.2874i) ans = 1.0000 - 1.0000i</pre>
<b>log10(x)</b>	<b>Función logaritmo en base 10 de x</b>  <pre>&gt;&gt; log10(100+100i)</pre>

	<pre>ans = 2.1505 + 0.3411i</pre>
<b>log2(x)</b>	<p><b>Función logaritmo en base 2 de x</b></p> <pre>&gt;&gt; log2 (4-6i) ans = 2.8502 - 1.4179i</pre>
<b>pow2(x)</b>	<p><b>Función potencia de base 2 de x</b></p> <pre>&gt;&gt; pow2(2.8502 - 1.4179i) ans = 3.9998 - 6.0000i</pre>
<b>sqrt(x)</b>	<p><b>Función raíz cuadrada de x</b></p> <pre>&gt;&gt; sqrt(1+i) ans = 1.0987 + 0.4551i</pre>

### **Funciones específicas de variable compleja**

MATLAB incorpora un grupo de funciones de variable compleja específicas para trabajar con módulos, argumentos y partes real e imaginaria. Entre estas funciones tenemos las siguientes:

<b>Función</b>	<b>Significado</b>
<b>abs(Z)</b>	<p><i>Módulo del complejo Z</i></p> <pre>&gt;&gt; abs(12.425-8.263i) ans = 14.9217</pre>
<b>angle(Z)</b>	<p><i>Argumento del complejo Z</i></p> <pre>&gt;&gt; angle(12.425-8.263i) ans = -0.5869</pre>
<b>conj(Z)</b>	<p><i>Conjugado del complejo Z</i></p> <pre>&gt;&gt; conj(12.425-8.263i) ans = 12.4250 + 8.2630i</pre>
<b>real(Z)</b>	<p><i>Parte real del complejo Z</i></p>

	<pre>&gt;&gt; real(12.425-8.263i) ans =</pre>
imag(Z)	<i>Parte imaginaria del complejo Z</i>  <pre>&gt;&gt; imag(12.425-8.263i) ans = -8 .2630</pre>
floor(Z)	<i>Aplica la función floor a real(Z) y a imag(Z)</i>  <pre>&gt;&gt; floor(12.425-8.263i) ans = 12.0000 - 9.0000i</pre>
ceil(Z)	<i>Aplica la función ceil a real(Z) y a imag(Z)</i>  <pre>&gt;&gt; ceil(12.425-8.263i) ans = 13.0000 - 8.0000i</pre>
round(Z)	<i>Aplica la función round a real(Z) y a imag(Z)</i>  <pre>&gt;&gt; round(12.425-8.263i) ans = 12.0000 - 8.0000i</pre>
fix(Z)	<i>Aplica la función fix a real(Z) y a imag(Z)</i>  <pre>&gt;&gt; fix(12.425-8.263i) ans = 12.0000 - 8.0000i</pre>

## FUNCIONES BÁSICAS CON ARGUMENTO UN VECTOR COMPLEJO

MATLAB permite trabajar con funciones de variable compleja vectorial e incluso matricial. No olvidemos que estas funciones también son válidas para variable real, ya que los números reales son un caso particular de los números complejos con parte imaginaria nula. A continuación se presenta una tabla con las funciones de variable compleja vectorial específicas que incorpora MATLAB. Más adelante, cuando se presenten las funciones de variable compleja matricial, observaremos que todas ellas son válidas también para variable vectorial, por ser un vector un caso particular de matriz.

<b>max(V)</b>	<i>Mayor componente (para complejos se calcula max(abs(V)))</i>  <pre>&gt;&gt; max([1-i 1+i 3-5i 6i]) ans =     0 + 6.0000i &gt;&gt; max([1, 0, -23, 12, 16]) ans =     16</pre>
<b>min(V)</b>	<i>Menor componente (para complejos se calcula min(abs(V)))</i>  <pre>&gt;&gt; min([1-i 1+i 3-5i 6i]) ans =     1.0 - 1.0000i &gt;&gt; min ([1, 0, -23, 12, 16]) ans =     -23</pre>
<b>mean(V)</b>	<i>Media de las componentes de V</i>  <pre>&gt;&gt; mean([1-i 1+i 3-5i 6i] ) ans =     1.2500 + 0.2500i &gt;&gt; mean([1, 0, -23, 12, 16] ) ans =     1.2000</pre>
<b>median(V)</b>	<i>Mediana de las componentes de V</i>  <pre>&gt;&gt; median([1-i 1+i 3-5i 6i]) ans =     2.0000 - 2.0000i &gt;&gt; median([1, 0, -23, 12, 16]) ans =     1</pre>
<b>std(V)</b>	<i>Desviación típica de las componentes de V</i>  <pre>&gt;&gt; std([1-i 1+i 3-5i 6i]) ans =     4.7434</pre>

	<pre>&gt;&gt; std{[1, 0, -23, 12, 16]) ans = 15.1888</pre>
<b>sort(V)</b>	<p><i>Ordena de forma ascendente las componentes de V. Para complejos hace la ordenación según los valores absolutos</i></p> <pre>&gt;&gt; sort([1-i 1+i 3-5i 6i]) ans = Columns 1 through 2 1.0000 - 1.0000i 1.0000 + 1.0000i Columns 3 through 4 3.0000 - 5.0000i 0 + 6.0000i &gt;&gt; sort ([1, 0, -23, 12, 16]) ans = -23 0 1 12 16</pre>
<b>sum(V)</b>	<p><i>Suma las componentes de V</i></p> <pre>&gt;&gt; sum([1-i 1+i 3-5i 6i]) ans = 5.0000 + 1.0000i &gt;&gt; sum([1, 0, -23, 12, 16]) ans = 6</pre>
<b>prod(V)</b>	<p><i>Multiplica los elementos de V, con lo que <math>n! = \text{prod}(1:n)</math></i></p> <pre>&gt;&gt; prod([1-i 1+i 3-5i 6i]) ans = 60.0000 +36.0000i &gt;&gt; prod([1, 0, -23, 12, 16]) ans = 0</pre>
<b>cumsum(V)</b>	<p><i>Da el vector de sumas acumuladas de V</i></p> <pre>&gt;&gt; cumsum([1-i 1+i 3-5i 6i]) ans = Columns 1 through 2 1.0000 - 1.0000i 2.0000</pre>

	<pre>Columns 3 through 4 5.0000 - 5.0000i 5.0000 + 1.0000i &gt;&gt; cumsum([1, 0, -23, 12, 16]) ans = 1     1    -22   -10      6</pre>
<b>cumprod(V)</b>	<p><i>Da el vector de productos acumulados de V</i></p> <pre>&gt;&gt; cumprod([1-i 1+i 3-5i 6i]) ans = Columns 1 through 2 1.0    -1.0000i    2.0000 Columns 3 through 4 6.0    -10.0000i   60.0000 +36.0000i &gt;&gt; cumprod([1, 0, -23, 12, 16]) ans = 1     0     0     0     0</pre>
<b>diff(V)</b>	<p><i>Da el vector de primeras diferencias de las componentes de V (V, -V<sub>t-1</sub>)</i></p> <pre>&gt;&gt; diff([1-i 1+i 3-51 6i]) ans = 0 + 2.0000i    2.0000 - 6.0000i   -3.0000 +11.0000i &gt;&gt; diff([1, 0, -23, 12, 16]) ans = -1    -23    35     4</pre>
<b>gradient(V)</b>	<p><i>Aproxima el gradiente de V</i></p> <pre>&gt;&gt; gradient ([ 1-i 1+i 3-5i 6i]) ans = Columns 1 through 3 0 + 2.0000i    1.0000 - 2.0000i   -0.5000 + 2.5000i Column 4 -3.0000 +11.0000i &gt;&gt; gradient ([1, 0, -23, 12, 16]) ans = -1.0000   -12.0000    6.0000   19.5000    4.0000</pre>

<b>del2(V)</b>	<i>Laplaciano de V (discreto de 5 puntos)</i>
	<pre>&gt;&gt; del2([1-i 1+i 3-5i 6i]) ans = Columns 1 through 3 2.2500 - 8.2500i 0.5000 - 2.0000i -1.2500 + 4.2500i Column 4  -3.0000 +10.5000i &gt;&gt; del2([1, 0, -23, 12, 16]) ans = -25.5000   -5.5000   14.5000   -7.7500   -30.0000</pre>
<b>fft(V)</b>	<i>Transformada discreta de Fourier de V</i>
	<pre>&gt;&gt; fft([1-i 1+i 3-5i 6i]) ans = Columns 1 through 3 5.0000 + 1.0000i   -7.0000 + 3.0000i   3.0000 -13.0000i Column 4 3.0000 + 5.0000i &gt;&gt; fft { [1, 0, -23, 12, 16] } ans = Columns 1 through 3 6.0000   14.8435 +35.7894i   -15.3435 -23.8824i Columns 4 through 5 -15.3435 +23.8824i   14.8435 -35.7894i</pre>
<b>fft2(V)</b>	<i>Transformada discreta bidimensional de Fourier de V</i>
	<pre>&gt;&gt; fft2 ( [1-i 1+i 3-5i 6i]) ans = Columns 1 through 3 5.0   + 1.0000i -7.0000 + 3.0000i 3.0000 -13.0000i Column 4 3.0   + 5.00001 &gt;&gt; fft2 ( [1, 0, -23, 12, 16]) ans =</pre>

	<pre>Columns 1 through 3 6.0000 14.8435 +35.7894i -15.3435 -23.8824i Columns 4 through 5 -15.3435 +23.8824i 14.8435 -35.7894i</pre>
<b>ifft(V)</b>	<p><i>Inversa de la transformada discreta de Fourier de V</i></p> <pre>&gt;&gt; ifft([1-i 1+i 3-5i 6i]) ans =  Columns 1 through 3 1.2500 + 0.2500i    0.7500 + 1.2500i    0.7500 - 3.2500i Column 4  -1.7500 + 0.7500i &gt;&gt; ifft([1, 0, -23, 12, 16]) ans =  Columns 1 through 3 1.2000    2.9687 - 7.1579i    -3.0687 + 4.7765i Columns 4 through 5  -3.0687 - 4.7765i    2.9687 + 7.1579i</pre>
<b>ifft2(V)</b>	<p><i>Inversa de la transformada 2-D discreta de Fourier de V</i></p> <pre>&gt;&gt; ifft2([1-i 1+i 3-5i 6i]) ans =  Columns 1 through 3 1.2500 + 0.2500i    0.7500 + 1.2500i    0.7500 - 3.2500i Column 4  -1.7500 + 0.7500i &gt;&gt; ifft2([1, 0, -23, 12, 16]) ans =  Columns 1 through 3 1.2000    2.9687 - 7.1579i    -3.0687 + 4.7765i Columns 4 through 5  -3.0687 - 4.7765i    2.9687 + 7.1579i</pre>

## FUNCIONES BASICAS CON ARGUMENTO UNA MATRIZ COMPLEJA

Las funciones del cuadro anterior también admiten como argumento una matriz compleja Z, en cuyo caso el resultado es un vector fila cuyas componentes son los resultados de aplicar la función a cada columna de la matriz. No olvidemos que estas funciones también son válidas para variable real, ya que los números reales son un caso particular de los números complejos con parte imaginaria nula.

<b>max(Z)</b>	<p><i>Vector con las mayores componentes (para complejos se calcula max(abs(Z))) de cada columna de la matriz Z</i></p> <pre>&gt;&gt; Z=[1-i 3i 5; -1+i 0 2i; 6-5i 8i -7] Z =     1.0000 - 1.0000i  0 + 3.0000i  5.0000    -1.0000 + 1.0000i  0             0 + 2.0000i     6.0000 - 5.0000i  0 + 8.0000i  -7.0000  &gt;&gt; Z=[1-i 3i 5-12i;-1+i 0 2i;6-5i 8i -7+6i] Z =     1.0000 - 1.0000i  0 + 3.0000i  5.0000 -12.0000i    -1.0000 + 1.0000i  0             0 + 2.0000i     6.0000 - 5.0000i  0 + 8.0000i  -7.0000 + 6.0000i  &gt;&gt; max (Z) ans =     6.0000 - 5.0000i  0 + 8.0000i  5.0000 -12.0000i  &gt;&gt; Z1=[1 3 5;-1 0 2;6 8 -7] Z1 =     1     3     5    -1     0     2     6     8    -7  &gt;&gt; max(Z1) ans =     6     8     5</pre>
<b>min(Z)</b>	<p><i>Vector con las menores componentes (para complejos se calcula min(abs(Z))) de cada columna de la matriz Z</i></p> <pre>&gt;&gt; min(Z) ans =     1.0000 - 1.0000i  0             0 + 2.0000i</pre>

	<pre>&gt;&gt; min(Z1)  ans = -1 0 -7</pre>
<b>mean(Z)</b>	<p><i>Vector de medias de las componentes de las columnas de Z</i></p> <pre>&gt;&gt; mean(Z) ans = 2.0000 - 1.6667i 0 + 3.6667i -0.6567 - 1.3333i &gt;&gt; mean(Z1) ans = 2.0000 3.6667 0</pre>
<b>median(Z)</b>	<p><i>Vector de medianas de las componentes de las columnas de Z</i></p> <pre>&gt;&gt; median(Z) ans = -1.0000 + 1.0000i 0 + 3.0000i -7.0000 + 6.0000i &gt;&gt; median(Z1) ans = 1 3 2</pre>
<b>std(Z)</b>	<p><i>Vector de desviaciones típicas de las componentes de las columnas de Z</i></p> <pre>&gt;&gt; std(Z) ans = 4.7258 4.0415 11.2101 &gt;&gt; std(Z1) ans = 3.6056 4.0415 6.2450</pre>
<b>sort(Z)</b>	<p><i>Ordena de forma ascendente las componentes de las columnas de Z.</i>  <i>Para complejos hace la ordenación según los valores absolutos</i></p> <pre>&gt;&gt; sort(Z) ans = 1.0000 - 1.0000i 0 0 + 2.0000i -1.0000 + 1.0000i 0 + 3.0000i -7.0000 + 6.0000i 6.0000 - 5.0000i 0 + 8.0000i 5.0000 -12.0000i &gt;&gt; sort(Z1) ans = -1 0 -7 1 3 2 6 8 5</pre>

<b>sum(Z)</b>	<i>Vector con la suma de las componentes de las columnas de la matriz Z</i>
	<pre>&gt;&gt; sum(Z) ans =     6.0000 - 5.0000i 0 +11.0000i -2.0000 - 4.0000i &gt;&gt; sum(Z1) ans =     6      11      0</pre>
<b>prod(Z)</b>	<i>Vector con los productos de los elementos de las columnas de Z</i>
	<pre>&gt; prod{z) ans = 1.0e+002 *     0.1000 + 0.1200i      0      -2.2800 + 0.7400i &gt;&gt; prod(Z1) ans =     -6      0     -70</pre>
<b>cumsum(Z)</b>	<i>Da la matriz de sumas acumuladas de las columnas de Z</i>
	<pre>&gt;&gt; cumsum(Z) ans =     1.0000 - 1.0000i      0 + 3.0000i      5.0000 -12.0000i             0                  0 + 3.0000i      5.0000 -10.0000i     6.0000 - 5.0000i      0 +11.0000i     -2.0000 -4.0000i  &gt;&gt; cumsum(Z1) ans =     1      3      5     0      3      7</pre>
<b>cumprod(V)</b>	<i>Da la matriz de productos acumulados de las columnas de Z</i>
	<pre>&gt;&gt; cumprod(Z) ans = 1.0e+002 *     0.0100 - 0.0100i      0 + 0.0300i      0.0500 - 0.1200i             0 + 0.0200i      0                  0.2400 + 0.1000i     0.1000 + 0.1200i      0                  -2.2800 + 0.7400i  &gt;&gt; cumprod(Z1) ans =     1      3      5</pre>

<b>diff(Z)</b>	<i>Da la matriz de primeras diferencias de las componentes de las columnas de Z</i> <pre>&gt;&gt; diff(Z) ans = -2.0000 + 2.0000i 0 - 3.0000i -5.0000 +14.0000i 7.0000 - 6.0000i 0 + 8.0000i -7.0000 + 4.0000i &gt;&gt; diff(Z1) ans = -2 -3 -3</pre>
<b>gradient(Z)</b>	<i>Aproxima la matriz de gradientes de las columnas de Z</i> <pre>&gt;&gt; gradient(Z) ans = -1.0000 + 4.0000i 2.0000 - 5.5000i 5.0000 -15.0000i 1.0000 - 1.0000i 0.5000 + 0.5000i 0 +2.0000i -6.0000 +13.0000i -6.5000 + 5.5000i -7.0000 -2.0000i &gt;&gt; gradient(Z1) ans = 2.0000 2.0000 2.0000 1.0000 1.5000 2.0000 2.0000 -6.5000 -15.0000</pre>
<b>del2(V)</b>	<i>Aproxima la matriz de laplacianos de las columnas de Z</i> <pre>&gt;&gt; del2(Z) ans = 3.7500 - 6.7500i 1.5000 - 2.0000i 1.0000 - 7.2500i 2.0000 - 1.2500i -0.2500 + 3.5000i -0.7500 - 1.7500i 2.0000 - 5.7500i -0.2500 - 1.0000i -0.7500 - 6.2500i &gt;&gt; del2(Z1) ans = 2.2500 2.7500 -1.5000 2.5000 3.0000 -1.2500 -2.0000 -1.5000 -5.7500</pre>
<b>fft(Z)</b>	<i>Matriz con las transformadas discreta de Fourier de las columnas de Z</i> <pre>&gt;&gt; fft(Z) ans = 6.0000 - 5.0000i 0 + 11.0000i -2.0000 - 4.0000i 3.6962 + 7.0622i -6.9282 - 1.0000i 5.0359 -22.0622i -6.6962 - 5.0622i 6.9282 - 1.0000i 11.9641 - 9.9378i &gt;&gt; fft(Z1)</pre>

	<pre>ans = 6.0000 11.0000 0 -1.5000 + 6.0622i -1.0000 + 6.9282i 7.5000 - 7.7942i -1.5000 - 6.0622i -1.0000 - 6.9282i 7.5000 + 7.7942i</pre>
<b>fft2(Z)</b>	<p><i>Matriz con las transformadas discretas bidimensionales de Fourier de las columnas de la matriz Z</i></p> <pre>&gt;&gt; fft2(Z) ans = 4.0000 + 2.0000i 19.9904 -10.2321i -5.9904 - 6.7679i 1.8038 - 16.0000i 22.8827 +28.9545i -13.5981 + 8.2321i 12.1962 -16.0000i -8.4019 + 4.7679i -23.8827 - 3.9545i &gt;&gt; fft2 (Z1) ans = 17.0000 0.5000 - 9.5263i 0.5000 + 9.5263i 5.0000 + 5.1962i 8.0000 + 13.8564i -17.5000 - 0.8660i 5.0000 - 5.1962i -17.5000 + 0.8660i 8.0000 - 13.8564i</pre>
<b>ifft(Z)</b>	<p><i>Matriz con las inversas de las transformadas discretas de Fourier de las columnas de Z</i></p> <pre>&gt;&gt; ifft(Z) ans = 2.0000 - 1.6667i 0 + 3.6667i -0.6667 - 1.3333i -2.2321 - 1.6874i 2.3094 - 0.3333i 3.9880 - 3.3126i 1.2321 + 2.3541i -2.3094 - 0.3333i 1.6786 - 7.3541i &gt;&gt; ifft(Z1) ans = 2.0000 3.6667 0 -0.5000 - 2.0207i -0.3333 - 2.3094i 2.5000 + 2.5981i -0.5000 + 2.0207i -0.3333 + 2.3094i 2.5000 - 2.5981i</pre>
<b>ifft2(Z)</b>	<p><i>Matriz con las inversas de la transformadas 2~D discretas de Fourier de las columnas de Z</i></p> <pre>&gt;&gt; ifft2(Z) ans = 0.4444 + 0.2222i -0.6656 - 0.7520i 2.2212 - 1.1369i 1.3551 - 1.7778i -2.6536 - 0.4394i -0.9335 + 0.5298i 0.2004 - 1.7778i -1.5109 + 0.9147i 2.5425 + 3.2172i &gt;&gt; ifft2(Z1) ans = 1.8889 0.0556 + 1.0585i 0.0556 - 1.0585i 0.5556 - 0.5774i 0.8889 - 1.5396i -1.9444 + 0.0962i 0.5556 + 0.5774i -1.9444 - 0.0962i 0.8889 + 1.5396i</pre>

## FUNCIONES GENERALES CON ARGUMENTO UNA MATRIZ COMPLEJA

MATLAB incorpora un grupo amplio de funciones trigonométricas, hiperbólicas, exponenciales y logarítmicas que admiten como argumento una matriz compleja. Evidentemente, todas estas funciones también admitirán un vector complejo como argumento, ya que el vector es un caso particular de matriz. Todas las funciones se aplican elemento a elemento en la matriz.

### *Funciones trigonométricas de variable matriz compleja*

A continuación se presenta una tabla con las funciones trigonométricas de variable compleja y sus inversas que incorpora MATLAB ilustradas con ejemplos. Para todas las funciones se usan como argumentos las matrices Z y Z1 definidas en la función seno.

#### *Función trigonométricas directas*

##### **sin(z)** Función seno

```
>> z=[1-i, 1+i, 2i;3-6i, 2+4i, -i;i,2i,3i]
Z =
1.0000 - 1.0000i 1.0000 + 1.0000i 0 + 2.0000i
3.0000 - 6.0000i 2.0000 + 4.0000i 0 - 1.0000i
0 + 1.0000i 0 + 2.0000i 0 + 3.0000i

>> z1=[1,1,2;3,2,-1;1,2,3]
Z1 =
1 1 2
3 2 -1
1 2 3

>> sin(Z)
ans =
1.0e+002 *
0.0130 - 0.0063i 0.0130 + 0.0063i 0 + 0.0363i
0.2847 + 1.9969i 0.2483 - 0.1136i 0 - 0.0118i
0 + 0.0118i 0 + 0.0363i 0 + 0.1002i

>> sin(Z1)
ans =
0.8415 0.8415 0.9093
0.1411 0.9093 -0.8415
0.8415 0.9093 0.1411
```

***cos(z) Función coseno***

```
>> cos(Z)
ans =
1.0e+002 *
0.0083 + 0.0099i 0.0083 - 0.0099i 0.0376
-1.9970 + 0.2847i -0.1136 - 0.2481i 0.0154
0.0154 0.0376 0.1007

>> cos(Z1)
ans =
0.5403 0.5403 -0.4161
-0.9900 -0.4161 0.5403
0.5403 -0.4161 -0.9900
```

***tan(z) Función tangente***

```
>> tan(Z)
ans =
0.2718 - 1.0839i 0.2718 + 1.0839i 0 + 0.9640i
-0.0005 - 1.0000i -0.0005 + 1.0004i 0 - 0.7616i
0 + 0.7616i 0 + 0.9640i 0 + 0.9951i

>> tan(Z1)
ans =
1.5574 1.5574 -2.1850
-0.1425 -2.1850 -1.5574
1.5574 -2.1850 -0.1425
```

***csc(z) Función cosecante***

```
>> csc(Z)
ans =
0.6215 + 0.3039i 0.6215 - 0.3039i 0 - 0.2757i
0.0007 - 0.0049i 0.0333 + 0.0152i 0 + 0.8509i
0 - 0.8509i 0 - 0.2757i 0 - 0.0998i

>> csc(Z1)
ans =
1.1884 1.1884 1.0998
7.0862 1.0998 -1.1884
1.1885 1.0998 7.0862
```

***sec(z) Función secante***

```
>> sec(Z)
ans =
0.4983 - 0.5911i 0.4983 + 0.5911i 0.2658
-0.0049 - 0.0007i -0.0153 + 0.0333i 0.6481
0.6481 0.2658 0.0993
```

```
>> sec(z1)

ans =
1.8508    1.8508   -2.4030
-1.0101   -2.4030    1.8508
1.8508   -2.4030   -1.0101
```

**cot(z)** Función cotangente

```
>> cot(z)

ans =
0.2176 + 0.8680i   0.2176 - 0.8680i   0 - 1.0373i
-0.0000 + 1.0000i  -0.0005 - 0.9996i   0 + 1.3130i
0 - 1.3130i         0 - 1.0373i     0 - 1.0050i

>> cot(z1)

ans =
0.6421    0.6421   -0.4577
-7.0153   -0.4577   -0.6421
0.6421   -0.4577   -7.0153
```

**Funciones trigonométricas inversas****asin(z)** Función arco seno

```
>> asin(z)

ans =
0.6662 - 1.0613i   0.6662 + 1.0613i   0 + 1.4436i
0.4592 - 2.5998i   0.4539 + 2.1986i   0 - 0.8814i
0 + 0.8814i        0 + 1.4436i     0 + 1.8184i

>> asin(z1)

ans =
1.5708                1.5708                1.5708 - 1.3170i
1.5708 - 1.7627i    1.5708 - 1.3170i   -1.5708
1.5708                1.5708 - 1.3170i   1.5708 - 1.7627i
```

**acos(z)** Función arco coseno

```
>> acos(z)

ans =
0.9046 + 1.0613i   0.9046 - 1.0613i   1.5708 - 1.4436i
1.1115 + 2.5998i   1.1169 - 2.1986i   1.5708 + 0.8814i
1.5708 - 0.8814i   1.5708 - 1.4436i   1.5708 - 1.8184i

>> acos(z1)

ans =
0                      0                      0 + 1.3170i
0 + 1.7627i           0 + 1.3170i       3.1416
0                      0 + 1.3170i       0 + 1.7627i
```

**atan(z) y atan2(z)** Función arco tangente

```
>> atan(Z)
Warning: Singularity in ATAN. This warning will be removed in
a future release.
    Consider using DBSTOP IF NANINF when debugging.
Warning: Singularity in ATAN. This warning will be removed in
a future release.
    Consider using DBSTOP IF NANINF when debugging.

ans =

 1.0172 - 0.4024i  1.0172 + 0.4024i  -1.5708 + 0.5493i
 1.5030 - 0.1335i  1.4670 + 0.2006i  0 - Infi
 0      + Infi     -1.5708 + 0.5493i  -1.5708 + 0.3466i

>> atan(Z1)

ans =

 0.7854      0.7854      1.1071
 1.2490      1.1071      -0.7854
 0.7854      1.1071      1.2490
```

**acsc(z)** Función arco cosecante

```
>> acsc(Z)

ans =

 0.4523 + 0.5306i  0.4523 - 0.5306i  0 - 0.4812i
 0.0661 + 0.1332i  0.0982 - 0.1996i  0 + 0.8814i
 0 - 0.8814i       0 - 0.4812i       0 - 0.3275i

>> acsc(Z1)

ans =

 1.5708      1.5708      0.5236
 0.3398      0.5236      -1.5708
 1.5708      0.5236      0.3398
```

**asec(z)** Función arco secante

```
>> asec(Z)

ans =

 1.1185 - 0.5306i  1.1185 + 0.5306i  1.5708 + 0.4812i
 1.5047 - 0.1332i  1.4726 + 0.1996i  1.5708 - 0.8814i
 1.5708 + 0.8814i  1.5708 + 0.4812i  1.5708 + 0.3275i

>> asec(Z1)

ans =

 0      0      1.0472
 1.2310 1.0472  3.1416
 0      1.0472  1.2310
```

**acot(z)** Función arco cotangente

```
>> acot(Z)
Warning: Singularity in ATAN. This warning will be removed in
```

```

a future release.
Consider using DBSTOP IF NANINF when debugging.
> In acot at 13

ans =
0.5536 + 0.4024i      0.5536 - 4024i      0 - 0.5493i
0.0678 + 0.1335i      0.1037 - 2006i      0 + Infii
0 - Infii              0 - 5493i       0 - 0.3466i

>> acot(Z1)

ans =
0.7854   0.7854   0.4636
0.3218   0.4636   -0.7854
0.7854   0.4636   0.3218

```

## **Funciones hiperbólicas de variable matriz compleja**

A continuación se presenta una tabla con las funciones hiperbólicas de variable compleja y sus inversas que incorpora MATLAB ilustradas con ejemplos.

### **Funciones hiperbólicas directas**

#### **sinh(z) Función seno hiperbólico**

```

>> sinh(Z)

ans =
0.6350 - 1.2985i      0.6350 + 1.2985i      0 + 0.9093i
9.6189 + 2.8131i      -2.3707 - 2.8472i      0 - 0.8415i
0 + 0.8415i           0 + 0.9093i       0 + 0.1411i

>> sinh(Z1)

ans =
1.1752      1.1752      3.6269
10.0179     3.6269     -1.1752
1.1752      3.6269     10.0179

```

#### **cosh(z) Función coseno hiperbólico**

```

>> cosh(Z)

ans =
0.8337 - 0.9889i      0.8337 + 0.9889i      -0.4161
9.6667 + 2.7991i      -2.459i - 2.7448i      0.5403
0.5403                  -0.416i          -0.9900

>> cosh(Z1)

ans =
1.5431      1.5431      3.7622
10.0677     3.7622     1.5431
1.5431      3.7622     10.0677

```

**tanh(z)** Función tangente hiperbólica

```
>> tanh(z)

ans =
    1.0839 - 0.2718i  1.0839 + 0.2718i  0 - 2.1850i
    0.9958 + 0.0026i  1.0047 + 0.0364i  0 - 1.5574i
    0 + 1.5574i      0 - 2.1850i     0 - 0.1425i

>> tanh(z1)

ans =
    0.7616   0.7616   0.9640
    0.9951   0.9640   -0.7616
    0.7616   0.9640   0.9951
```

**csch(z)** Función cosecante hiperbólica

```
>> csch(z)

ans =
    0.3039 + 0.6215i  0.3039 - 0.6215i  0 - 1.0998i
    0.0958 - 0.0280i  -0.1727 + 0.2074i  0 + 1.1884i
    0 - 1.1884i       0 - 1.0998i     0 - 7.0862i

>> csch(z1)

ans =
    0.8509   0.8509   0.2757
    0.0998   0.2757   -0.8509
    0.8509   0.2757   0.0998
```

**sech(z)** Función secante hiperbólica

```
>> sech(z)

ans =
    0.4983 + 0.5911i  0.4983 - 0.5911i  -2.4030
    0.0954 - 0.0276i  -0.1811 + 0.2021i  1.8508
    1.8508             -2.4030            -1.0101

>> sech(z1)

ans =
    0.6481   0.6481   0.2658
    0.0993   0.2658   0.6481
    0.6481   0.2658   0.0993
```

**coth(z)** Función cotangente hiperbólica

```
>> coth(z)

ans =
    0.8680 + 0.2176i  0.8680 - 0.2176i  0 + 0.4577i
    1.0042 - 0.0027i  0.9940 - 0.0360i  0 + 0.6421i
    0 - 0.6421i       0 + 0.4577i     0 + 7.0153i
```

```
>> coth(Z1)
ans =
1.3130    1.3130    1.0373
1.0050    1.0373   -1.3130
1.3130    1.0373    1.0050
```

***Funciones hiperbólicas inversas******asinh(z) Función arco seno hiperbólico***

```
>> aslnh(z)
ans =
1.0613 - 0.6662i  1.0613 + 0.6662i  1.3170 + 1.5708i
2.5932 - 1.1027i  1.1836 + 1.0969i      0 - 1.5708i
          0 + 1.5708i  1.3170 + 1.5708i  1.7627 + 1.5708i

>> aslnh(Z1)
ans =
0.8814    0.8814    1.4436
1.8184    1.4436   -0.8814
0.8814    1.4436    1.8184
```

***acosh(z) Función arco coseno hiperbólico***

```
>> acosh(z)
ans =
1.0613 - 0.9046i  1.0613 + 0.9046i  1.4436 + 1.5708i
2.5998 - 1.1115i  2.1986 + 1.1169i  0.8814 - 1.5708i
          0.8814 + 1.5708i  1.4436 + 1.5708i  1.8184 + 1.5708i

>> acosh(Z1)
ans =
0          0          1.3170
1.7627    1.3170    0          + 3.1416i
0          1.3170    1.7627
```

***atanh(z) Función arco tangente hiperbólica***

```
>> atanh(z)
ans =
0.4024 - 1.0172i  0.4024 + 1.0172i  0 + 1.1071i
0.0656 - 1.4377i  0.0964 + 1.3715i  0 - 0.7854i
          0 + 0.7854i  0 + 1.1071i  0 + 1.2490i

>> atanh(Z1)
ans =
Inf          Inf          0.5493 + 1.5708i
0.3466 + 1.5708i  0.5493 + 1.5708i      -Inf
          Inf          0.5493 + 1.5708i  0.3466 + 1.5708i
```

**acsch(z)** Función arco cosecante hiperbólica

```
>> acsch(Z)
ans =
0.5306 + 0.4523i 0.5306 - 0.4523i 0 - 0.5236i
0.0672 + 0.1334i 0.1019 - 0.2003i 0 + 1.5708i
0 - 1.5708i 0 - 0.5236i 0 - 0.3398i

>> acsch(Z1)
ans =
0.8814 0.8814 0.4812
0.3275 0.4812 -0.8814
0.8814 0.4812 0.3275
```

**asech(z)** Función arco secante hiperbólica

```
>> asech(Z)
ans =
0.5306 + 1.1185i 0.5306 - 1.1185i 0.4812 - 1.5708i
0.1332 + 1.5047i 0.1996 - 1.4726i 0.8814 + 1.5708i
0.8814 - 1.5708i 0.4812 - 1.5708i 0.3275 - 1.5708i

>> asech(Z1)
ans =
0 0 0 + 1.0472i
0 + 1.2310i 0 + 1.0472i 0 + 3.1416i
0 0 + 1.0472i 0 + 1.2310i
```

**acoth(z)** Función arco cotangente hiperbólica

```
>> acoth(Z)
ans =
0.4024 + 0.5536i 0.4024 - 0.5536i 0 - 0.4636i
0.0656 + 0.1331i 0.0964 - 0.1993i 0 + 0.7854i
0 - 0.7854i 0 - 0.4636i 0 - 0.3218i

>> acoth(Z1)
ans =
Inf Inf 0.5493
0.3466 0.5493 -Inf
Inf 0.5493 0.3466
```

**Funciones exponenciales y logarítmicas de variable matriz compleja**

A continuación se presenta una tabla con las funciones exponenciales y logarítmicas que incorpora MATLAB ilustradas con ejemplos. Las matrices Z y Z1 son las mismas de los ejemplos anteriores.

<b>Función</b>	<b>Significado</b>
<b>exp(z)</b>	<p><i>Función exponencial en base e (e^x)</i></p> <pre>&gt;&gt; exp(Z) ans = 1.4687 - 2.2874i 1.4687 + 2.2874i -0.4161 + 0.9093i 19.2855 + 5.6122i -4.8298 - 5.5921i 0.5403 - 0.8415i 0.5403 + 0.8415i -0.4161 + 0.9093i -0.9900 + 0.1411i</pre> <pre>&gt;&gt; exp(Z1) ans = 2.7183 2.7183 7.3891 20.0855 7.3891 0.3679 2.7183 7.3891 20.0855</pre>
<b>log(x)</b>	<p><i>Función logaritmo en base e de x</i></p> <pre>&gt;&gt; log(Z) ans = 0.3466 - 0.7854i 0.3466 + 0.7854i 0.6931 + 1.5708i 1.9033 - 1.1071i 1.4979 + 1.1071i 0 - 1.5708i 0 + 1.5708i 0.6931 + 1.5708i 1.0986 + 1.5708i</pre> <pre>&gt;&gt; log(Z1) ans = 0 0 0.6931 1.0986 0.6931 0 + 3.1416i 0 0.6931 1.0986</pre>
<b>log10(x)</b>	<p><i>Función logaritmo en base 10 de x</i></p> <pre>&gt;&gt; log10(Z) ans = 0.1505 - 0.3411i 0.1505 + 0.3411i 0.3010 + 0.6822i 0.8266 - 0.4808i 0.6505 + 0.4808i 0 - 0.6822i 0 + 0.6822i 0.3010 + 0.6822i 0.4771 + 0.6822i</pre> <pre>&gt;&gt; log10(Z1) ans = 0 0 0.3010 0.4771 0.3010 0 + 1.3644i 0 0.3010 0.4771</pre>
<b>log2(x)</b>	<p><i>Función logaritmo en base 2 de x</i></p> <pre>&gt;&gt; log2(Z) ans = 5.000 - 1.1331i 0.5000 + 1.1331i 1.0000 + 2.2662i 2.7459 - 1.5973i 2.1610 + 1.5973i 0 - 2.2662i 0 + 2.2662i 1.0000 + 2.2662i 1.5850 + 2.2662i</pre>

	<pre>&gt;&gt; log2 (Z1) ans =     0         0      1.0000     1.5850    1.0000      0 + 4.53241i     0         1.0000    1.5850</pre>
<b>pow2(x)</b>	<p><i>Función potencia de base 2 de x</i></p> <pre>&gt;&gt; pow2 (Z) ans =     1.5385 - 1.2779i  1.5385 + 1.2779i  0.1835 + 0.9830i     -4.2054 + 6.8055i -3.7307 + 1.4427i  0.7692 - 0.6390i     0.7692 + 0.6390i  0.1835 + 0.9830i  -0.4870 + 0.8734i  &gt;&gt; pow2(Z1) ans =     2.0000    2.0000    4.0000     8.0000    4.0000    0.5000     2.0000    4.0000    8.0000</pre>
<b>sqrt(x)</b>	<p><i>Función raíz cuadrada de x</i></p> <pre>&gt;&gt; sqrt (Z) ans =     1.0987 - 0.4551i  1.0987 + 0.4551i  1.0000 + 1.0000i     2.2032 - 1.3617i  1.7989 + 1.1118i  0.7071 - 0.7071i     0.7071 + 0.7071i  1.0000 + 1.0000i  1.2247 + 1.2247i  &gt;&gt; sqrt(Z1) ans =     1.0000    1.0000    1.4142     1.7321    1.4142      0 + 1.0000i     1.0000    1.4142    1.7321</pre>

### Funciones específicas de variable matricial compleja

MATLAB incorpora un grupo de funciones de variable compleja específicas para trabajar con módulos, argumentos y partes real e imaginaria. Entre estas funciones tenemos las siguientes:

<i>Función</i>	<i>Significado</i>
<b>abs(Z)</b>	<p><i>Módulo del complejo Z</i></p> <pre>&gt;&gt; abs (Z) ans =     1.4142    1.4142    2.0000     6.7082    4.4721    1.0000     1.0000    2.0000    3.0000</pre>

	<pre>&gt;&gt; abs(Z1)  ans = 1      1      2 3      2      1 1      2      3</pre>
<b>angle(Z)</b>	<p><b>Argumento del complejo Z</b></p> <pre>&gt;&gt; angle(Z)  ans = -0.7854    0.7854    1.5708 -1.1071    1.1071   -1.5708 1.5708    1.5708    1.5708  &gt;&gt; angle(Z1)  ans = 0 0 0 0 0 3.1416 0 0 0</pre>
<b>conj(Z)</b>	<p><b>Conjugado del complejo Z</b></p> <pre>&gt;&gt; conj(Z)  ans = 1.0000 + 1.0000i  1.0000 - 1.0000i  0 - 2.0000i 3.0000 + 6.0000i  2.0000 - 4.0000i  0 + 1.0000i 0 - 1.0000i       0 - 2.0000i     0 - 3.0000i  &gt;&gt; conj(Z1)  ans = 1      1      2 3      2     -1 1      2      3</pre>
<b>real(Z)</b>	<p><b>Parte real del complejo Z</b></p> <pre>&gt;&gt; real(Z)  ans = 1      1      0 3      2      0 0      0      0  &gt;&gt; real(Z1)  ans = 1      1      2 3      2     -1 1      2      3</pre>
<b>imag(Z)</b>	<p><b>Parte imaginaria del complejo Z</b></p> <pre>&gt;&gt; imag(Z)</pre>

	<pre>&gt;&gt; log2 (z1)  ans =       0         0        1.0000</pre>
<b>pow2(x)</b>	<p><i>Función potencia de base 2 de x</i></p> <pre>&gt;&gt; pow2 (z)  ans =     1.5385 - 1.2779i   1.5385 + 1.2779i   0.1835 + 0.9830i    -4.2054 + 6.8055i  -3.7307 + 1.4427i   0.7692 - 0.6390i     0.7692 + 0.6390i   0.1835 + 0.9830i  -0.4870 + 0.8734i  &gt;&gt; pow2(z1)  ans =       2.0000    2.0000    4.0000       8.0000    4.0000    0.5000       2.0000    4.0000    8.0000</pre>
<b>sqrt(x)</b>	<p><i>Función raíz cuadrada de x</i></p> <pre>&gt;&gt; sqrt (z)  ans =     1.0987 - 0.4551i  1.0987 + 0.4551i  1.0000 + 1.0000i     2.2032 - 1.3617i  1.7989 + 1.1118i  0.7071 - 0.7071i     0.7071 + 0.7071i  1.0000 + 1.0000i  1.2247 + 1.2247i  &gt;&gt; sqrt(z1)  ans =       1.0000          1.0000          1.4142       1.7321          1.4142          0 + 1.0000i       1.0000          1.4142          1.7321</pre>

## Funciones específicas de variable matricial compleja

MATLAB incorpora un grupo de funciones de variable compleja específicas para trabajar con módulos, argumentos y partes real e imaginaria. Entre estas funciones tenemos las siguientes:

<b>Función</b>	<b>Significado</b>
<b>abs(Z)</b>	<p><i>Módulo del complejo Z</i></p> <pre>&gt;&gt; abs (z)  ans =       1.4142    1.4142    2.0000       6.7082    4.4721    1.0000       1.0000    2.0000    3.0000</pre>

	<pre>&gt;&gt; abs (z1)  ans = 1      1      2 3      2      1 1      2      3</pre>
<b>angle(Z)</b>	<p><i>Argumento del complejo Z</i></p> <pre>&gt;&gt; angle(z)  ans = -0.7854    0.7854   1.5708 -1.1071    1.1071  -1.5708 1.5708    1.5708   1.5708  &gt;&gt; angle(z1)  ans = 0      0      0 0      0      3.1416 0      0      0</pre>
<b>conj(Z)</b>	<p><i>Conjugado del complejo Z</i></p> <pre>&gt;&gt; conj(z)  ans = 1.0000 + 1.0000i  1.0000 - 1.0000i 0 - 2.0000i 3.0000 + 6.0000i  2.0000 - 4.0000i 0 + 1.0000i 0 - 1.0000i       0 - 2.0000i 0 - 3.0000i  &gt;&gt; conj(z1)  ans = 1      1      2 3      2     -1 1      2      3</pre>
<b>real(Z)</b>	<p><i>Parte real del complejo Z</i></p> <pre>&gt;&gt; real(z)  ans = 1      1      0 3      2      0 0      0      0  &gt;&gt; real(z1)  ans = 1      1      2 3      2     -1 1      2      3</pre>
<b>imag(Z)</b>	<p><i>Parte imaginaria del complejo Z</i></p> <pre>&gt;&gt; Imag(z)</pre>

	<pre> ans =       -1      1      2       -6      4     -1        1      2      3  &gt;&gt; Imag(Z1)  ans =       0      0      0       0      0      0       0      0      0 </pre>
<b>floor(Z)</b>	<p><i>Aplica la función floor a real(Z) y a imag(Z)</i></p> <pre> &gt;&gt; floor(12.357*z)  ans =       12.0000 -13.0000i   12.0000 + 12.0000i   0 + 24.0000i       37.0000 -75.0000i   24.0000 + 49.0000i   0 - 13.0000i                   0 +12.0000i           0 + 24.0000i   0 + 37.0000i  &gt;&gt; floor(12.357*z1)  ans =       12      12      24       37      24     -13       12      24      37 </pre>
<b>ceil(Z)</b>	<p><i>Aplica la función ceil a real(Z) y a imag(Z)</i></p> <pre> &gt;&gt; ceil(12.357*z)  ans =       13.0000 - 12.0000i   13.0000 + 13.0000i   0 + 25.0000i       38.0000 - 74.0000i   25.0000 + 50.0000i   0 - 12.0000i                   0 + 13.0000i           0 + 25.0000i   0 + 38.0000i  &gt;&gt; ceil(12.357*z1)  ans =       13      13      25       38      25     -12       13      25      38 </pre>
<b>round(Z)</b>	<p><i>Aplica la función round a real(Z) y a imag(Z)</i></p> <pre> &gt;&gt; round(12.357*z)  ans =       12.0000  - 12.0000i   12.0000 + 12.0000i   0 + 25.0000i       37.0000  - 74.0000i   25.0000 + 49.0000i   0 - 12.0000i                   0 + 12.0000i           0 + 25.0000i   0 + 37.0000i  &gt;&gt; round(12.357*z1)  ans = </pre>

	$\begin{array}{ccc} 12 & 12 & 25 \\ 37 & 25 & -12 \\ 12 & 25 & 37 \end{array}$
<b>fix(Z)</b>	<p>Aplica la función fix a real(Z) y a imag(Z)</p> <pre>&gt;&gt; fix(12.357*z) ans = 12.0000 - 12.0000i 12.0000 + 12.0000i 0 + 24.0000i 37.0000 - 74.0000i 24.0000 + 49.0000i 0 - 12.0000i 0 + 12.0000i 0 + 24.0000i 0 + 37.0000i &gt;&gt; fix(12.357*z1) ans = 12 12 24 37 24 -12 12 24 37</pre>

## OPERACIONES CON VARIABLES MATRICIALES REALES V COMPLEJAS

MATLAB contempla las operaciones habituales de suma, diferencia, producto, potenciación, exponenciación e inversión con variables matriciales complejas. Evidentemente todas estas operaciones serán válidas también para variables matriciales reales por ser un caso particular de las complejas. La tabla siguiente resume estas operaciones que son válidas tanto para variables matriciales numéricas reales y complejas como para variables matriciales algebraicas.

<b>A+B</b>	<p><i>Suma de variables matriciales</i></p> <pre>&gt;&gt; A=[1+i, 1-i, 2i; -i,-3i,6-5i;2+3i, 2-3i, i] A = 1.0000 + 1.0000i 1.0000 - 1.0000i 0 + 2.0000i 0 - 1.0000i 0 - 3.0000i 6.0000 - 5.0000i 2.0000 + 3.0000i 2.0000 - 3.0000i 0 + 1.0000i  &gt;&gt; B=[i, -1, 2i; 1-i,7-3i,2-5i;8-6i, 5-i, 1+i] B = 0 + 1.0000i 0 - 1.0000i 0 + 2.0000i 1.0000 - 1.0000i 7.0000 - 3.0000i 2.0000 - 5.0000i 8.0000 - 6.0000i 5.0000 - 1.0000i 1.0000 + 1.0000i  &gt;&gt; A1=[1 6 2;3 5 0; 2 4 -1] A1 = 1 6 2 3 5 0 2 4 -1</pre>
------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<pre>&gt;&gt; B1=[-3 -6 1;-3 -5 2; 12 14 -10]  B1 =     -3    -6     1     -3    -5     2     12    14   -10  &gt;&gt; A+B  ans =     1.0000 + 2.0000i  1.0000 - 2.0000i      0 + 4.0000i     1.0000 - 2.0000i  7.0000 - 6.0000i      8.0000 -10.0000i     10.0000 - 3.0000i 7.0000 - 4.0000i    1.0000 + 2.0000i  &gt;&gt; A1+B1  ans =     -2     0     3      0     0     2     14    18   -11</pre>
<b>A-B</b>	<p><i>Diferencia de variables matriciales</i></p> <pre>&gt;&gt; A-B  ans =     1.0000          1.0000          0    -1.0000         -7.0000         4.0000    -6.0000 + 9.0000i -3.0000 - 2.0000i -1.0000  &gt;&gt; A1-B1  ans =     4     12     1     6     10    -2    -10    -10     9</pre>
<b>A*B</b>	<p><i>Producto de variables matriciales</i></p> <pre>&gt;&gt; A*B  ans =     11.0000 + 15.0000i  7.0000 - 1.0000i  -7.0000 - 3.0000i     16.0000 - 79.0000i  15.0000 -52.0000i  -2.0000 - 5.0000i     2.0000 + 5.0000i   9.0000 -24.0000i  -18.0000 -11.0000i  &gt;&gt; A1*B1  ans =     3     -8     -7    -24    -43     13    -30    -46     20</pre>
<b>A^n</b>	<p><i>Potencia enésima de la variable matricial A</i></p> <pre>&gt;&gt; A^3</pre>

	<pre> ans = 1.0e+002 * 0.1000 - 0.3400i -0.3200 - 0.1200i 0.3400 - 0.3600i 0.0900 - 0.0300i -1.0700 + 0.2100i -2.2500 - 0.6700i 0.3700 - 0.7900i -1.0300 - 0.0300i -0.0700 - 0.3700i  &gt;&gt; A1^3  ans = 155      358      46 159      347      30 106      232      19  ans = Columns 1 through 2 1.0000    - 1.0000i    2.0000 Columns 3 through 4 6.0000 -10.0000i   60.0000 +36.0000i &gt;&gt; cumprod([1, 0, -23, 12, 16])  ans = 1     0     0     0     0 </pre>
<b>P<sup>A</sup></b>	<p><i>Escalar p elevado a la variable matricial A</i></p> <pre> &gt;&gt; 3^A  ans = 0.0159 - 1.2801i   -0.5297 + 2.8779i   -1.9855 + 3.0796i -10.3372 + 0.4829i  17.0229 +12.9445i  14.7327 +20.1633i -5.0438 + 0.2388i   7.0696 + 6.9611i   5.7189 + 9.5696i  &gt;&gt; 3*A1  ans = 1.0e+003 * 2.2230      4.9342      0.4889 2.1519      4.7769      0.4728 1.4346      3.1844      0.3156 </pre>
<b>A'</b>	<p><i>Variable matricial transpuesta de la variable A</i></p> <pre> &gt;&gt; A'  ans = 1.0000 - 1.0000i          0 + 1.0000i    2.0000 - 3.0000i 1.0000 + 1.0000i          0 + 3.0000i    2.0000 + 3.0000i           0 - 2.0000i        6.0000 + 5.0000i    0 - 1.0000i  &gt;&gt; A1'  ans = </pre>

	<table border="1"> <tr><td>1</td><td>3</td><td>2</td></tr> <tr><td>6</td><td>5</td><td>4</td></tr> <tr><td>2</td><td>0</td><td>-1</td></tr> </table>	1	3	2	6	5	4	2	0	-1
1	3	2								
6	5	4								
2	0	-1								
	<p><i>Variable matricial inversa de la variable matricial A</i></p> <pre>&gt;&gt; A^-1 ans = -2.5000 + 2.0000i   -0.0500 + 0.6500i   0.8500 - 1.0500i  0.5000 + 3.0000i   0.5500 + 0.3500i   -0.3500 - 0.9500i -1.0000 - 1.0000i   -0.2000 + 0.1000i   0.4000 + 0.3000i</pre>									
	<pre>&gt;&gt; A1^-1 ans = -0.2941   0.8235   -0.5882  0.1765   -0.2941   0.3529  0.1176   0.4706   -0.7647</pre> <pre>&gt;&gt; A*A^-1 ans = 1.0000      0.0000 - 0.0000i   -0.0000 + 0.0000i -0.0000 - 0.0000i   1.0000 + 0.0000i   0.0000  0.0000 + 0.0000i   0.0000      1.0000 + 0.0000i</pre> <pre>&gt;&gt; A1*A1^-1 ans = 1.0000   -0.0000   0 -0.0000   1.0000   0</pre>									
A\B	<p><i>Si A es cuadrada A\B=(A<sup>-1</sup>)*B y si A no es cuadrada A\B es la solución en el sentido de mínimos cuadrados del sistema AX=B</i></p> <pre>&gt;&gt; A\B ans = -0.9000 -15.3000i   6.8000 + 1.1000i   1.0500 - 3.6500i -10.6000 - 5.2000i   5.2000 - 4.1000i   -2.5500 - 2.3500i  5.9000 - 0.7000i   0.2000 + 3.4000i   2.2000 - 0.1000i</pre> <pre>&gt;&gt; A1\B1 ans = -8.6471   -10.5882   7.2353  4.5882    5.3529   -3.9412 -10.9412   -13.7647   8.7059</pre>									
B/A	<p><i>Coincide con (A<sup>-1</sup>)B</i></p> <pre>&gt;&gt; B/A ans = 3.0000   - 5.0000i   -0.5000 - 1.0000i   -0.5000 + 2.0000i  5.0000 + 27.0000i   5.6000 + 2.7000i   -3.2000 - 8.9000i -2.5000 + 43.5000i   6.3000 + 6.6000i   -2.1000 - 17.2000i</pre>									

```

>> A'\B'
ans =
 3.0000 + 5.0000i   5.0000 -27.0000i   -2.5000 -43.5000i
-0.5000 + 1.0000i   5.6000 - 2.7000i   6.3000 - 6.6000i
-0.5000 - 2.0000i   -3.2000 + 8.9000i   -2.1000 +17.2000i

>> B1/A1
ans =

-0.0588   -0.2353   -1.1176
 0.2353   -0.0588   -1.5294
-2.2353   1.0588   5.5294

>> A1'\B1'
ans =

-0.0588   0.2353   -2.2353
-0.2353   -0.0588   1.0588
-1.1176   -1.5294   5.5294

```

*Ejercicio 1. Dados los números complejos  $z_1=1-i$  y  $z_2=5i$ , calcular:  $z_1^3$ ,  $z_1^2/z_2^4$ ,  $z_1^{1/2}, z_3^{3/2}$ ,  $\ln(z_1+z_2)$ ,  $\operatorname{sen}(z_1-z_2)$  y  $\tanh(z_1/z_2)$ .*

```

>> z1=1-i
z1 =
 1.0000 - 1.0000i

>> z2=5i
z2 =
 0 + 5.0000i

>> z1^3
ans =
 -2.0000 - 2.0000i

>> z1^2/z2^4
ans =
 0 - 0.0032i

>> sqrt(z1)
ans =
 1.0987 - 0.4551i

>> sqrt(z2^3)

```

```
ans =
7.9057 - 7.9057i
>> log(z1+z2)
ans =
1.4166 + 1.3258i
>> sin(z1-z2)
ans =
1.6974e+002 -1.0899e+002i
>> tanh(z1/z2)
ans =
-0.2052 - 0.1945i
```

*Ejercicio 2. Realizar las siguientes operaciones con números complejos:*

$$\frac{i^8 - i^{-8}}{3-4i} + 1, i^{\operatorname{sen}(1+i)}, (2 + \operatorname{Ln}(i))^{1/i}, (1+i)^i, i\operatorname{Ln}^{(1+i)}, (1+\sqrt{3}i)^{1-i}$$

```
>> (i[^8-i^(-8))/(3-4*i) + 1
ans =
1
>> i^(sin(1+i))
ans =
-0.16665202215166 + 0.32904139450307i
>> (2+log(i))-(1/i)
ans =
1.15809185259777 - 1.56388053989023i
>> (1+1)^i
ans =
0.42882900629437 + 0.15487175246425i
>> i^(log(1+i))
ans =
0.24911518828716 + 0.15081974484717i
>> (1+sqrt(3)*i)^(1-i)
```

```
ans =
5.34581479196611 + 1.97594883452873i
```

*Ejercicio 3. Hallar parte real, parte imaginaria, módulo y argumento de los siguientes complejos:*

$$i^{3+i}, (1+\sqrt{3}i)^{1-i}, i^{ii}, i^i$$

```
>> z1=i^3*i; z2= (1+sqrt(3)*i)^(1-i) ; z3=(i^i)^i; z4=i^i;
>> format short
>> real ([z1 z2 z3 z4])
ans =
1.0      5.3458      0.0000      0.2079
>> imag ([z1 z2 z3 z4] )
ans =
0      1.9759     -1.0000      0
>> abs([z1 z2 z3 z4])
ans =
1.0      5.6993      1.0000      0.2079
>> angle([z1 z2 z3 z4])
ans=
0      0.3541     -1.5708      0
```

*Ejercicio 4. Consideramos la matriz  $M$  producto de la unidad imaginaria por la matriz cuadrada de orden 3 cuyos elementos por filas son los nueve primeros números enteros positivos.*

*Obtener su cuadrado, su raíz cuadrada y sus exponentiales de bases 2 y -2.*

*Obtener su logaritmo neperiano elemento a elemento y su exponencial de base e elemento a elemento.*

*Obtener  $e^M$  y  $\ln(M)$ .*

```
>> M=i*[1 2 3;4 5 6;7 8 9]
M =
0 + 1.0000i      0 + 2.0000i      0 + 3.0000i
0 + 4.0000i      0 + 5.0000i      0 + 6.0000i
0 + 7.0000i      0 + 8.0000i      0 + 9.0000i
```

```
>> C=M^2
C =
   -30    -36    -42
   -66    -81    -96
  -102   -126   -150

>> D=M^(1/2)
D =
  0.8570 - 0.2210i  0.5370 + 0.2445i  0.2169 + 0.7101i
  0.7797 + 0.6607i  0.9011 + 0.8688i  1.0224 + 1.0769i
  0.7024 + 1.5424i  1.2651 + 1.4930i  1.8279 + 1.4437i

>> 2^M
ans =
  0.7020 - 0.6146i  -0.1693 - 0.2723i  -0.0407 + 0.0699i
  -0.2320 - 0.3055i  0.7366 - 0.3220i  -0.2947 - 0.3386i
  -0.1661 + 0.0036i  -0.3574 - 0.3717i  0.4513 - 0.7471i

>> (-2)^M
ans =
  17.3946 - 16.8443i  4.3404 - 4.5696i  -7.7139 + 7.7050i
  1.5685 - 1.8595i   1.1826 - 0.5045i   -1.2033 + 0.8506i
  -13.2575 + 13.1252i -3.9751 + 3.5607i  6.3073 - 6.0038i

>> log(M)
ans =
  0 + 1.5708i  0.6931 + 1.5708i  1.0986 + 1.5708i
  1.3863 + 1.5708i  1.6094 + 1.5708i  1.7918 + 1.5708i
  1.9459 + 1.5708i  2.0794 + 1.5708i  2.1972 + 1.5708i

>> exp(M)
ans =
  0.5403 + 0.8415i  -0.4161 + 0.9093i  -0.9900 + 0.1411i
  -0.6536 - 0.7568i  0.2837 - 0.9589i   0.9602 - 0.2794i
  0.7539 + 0.6570i   -0.1455 + 0.9894i  -0.9111 + 0.4121i

>> logm(M)
ans =
  -5.4033 - 0.8472i  11.9931 - 0.3109i  -5.3770 + 0.8846i
  12.3029 + 0.0537i  -22.3087 + 0.8953i  12.6127 + 0.4183i
  -4.7574 + 1.6138i  12.9225 + 0.7828i  -4.1641 + 0.6112i
```

```
>> expm(M)
ans =
0.3802 - 0.6928i  -0.3738 - 0.2306i  -0.1278 + 0.2316i
-0.5312 - 0.1724i  0.3901 - 0.1434i  -0.6886 - 0.1143i
-0.4426 + 0.3479i -0.8460 - 0.0561i  -0.2493 - 0.4602i
```

*Ejercicio 5. Consideramos el vector V suma del vector complejo Z= (i,-i, i) y el vector real R=(0,1,1). Hallar la media, mediana, desviación típica, varianza, suma, producto, máximo y mínimo de los elementos de V, así como su gradiente, la transformada discreta de Fourier y su inversa.*

```
>> Z=[i,-i,i]
0 + 1.0000i          0 - 1.0000i          0 + 1.0000i
>> R=[0,1,1]
R =
0      1      1
>> V=Z+R
V =
0 + 1.0000i          1.0000 - 1.0000i          1.0000 + 1.0000i
>> [mean(V),median(V),std(V),var(V),sum(V),prod(V),max(V) , min(V) ] '
ans =
0.6667 - 0.3333i
1.0000 + 1.0000i
1.2910
1.6667
2.0000 - 1.0000i
0      - 2.0000i
1.0000 + 1.0000i
0      - 1.0000i
>> gradient(V)
ans =
1.0000 - 2.0000i          0.5000          0 + 2.0000i
>> fft(V)
ans =
2.0000 + 1.0000i          -2.7321 + 1.0000i          0.7321 + 1.0000i
>> ifft(V)
```

```
ans =
```

$$0.6667 + 0.3333i \quad 0.2440 + 0.3333i \quad -0.9107 + 0.3333i$$

*Ejercicio 6. Dadas las matrices:*

$$A1 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad A2 = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad B1 = \begin{bmatrix} 0 & 1 & 2 \\ 0 & -1 & 3 \\ 0 & 0 & 0 \end{bmatrix} \quad B2 = \begin{bmatrix} -i & i & -i \\ 0 & 0 & i \\ 0 & 0 & i \end{bmatrix}$$

$$C1 = \begin{bmatrix} 1 & -1 & 0 \\ -1 & \text{sqrt}(2)i & \text{sqrt}(2)i \\ 0 & 0 & -1 \end{bmatrix} \quad C2 = \begin{bmatrix} 0 & 2 & 1 \\ 1 & 0 & 0 \\ 1 & -1 & 0 \end{bmatrix}$$

*Calcular inicialmente  $A1+A2$ ,  $B1-B2$  y  $C1+C2$ .*

*Posteriormente calcular  $AB - BA$ ,  $A^2 + B^2 + C^2$ ,  $ABC$ ,  $\text{sqrt}(A) + \text{sqrt}(B) - \text{sqrt}(C)$ ,  $e^A$  ( $e^B + e^C$ ), sus inversas y sus transpuestas.*

*Finalmente comprobar que al multiplicar cualquier matriz por su inversa se obtiene la matriz identidad.*

```
>> A1=eye(3)

A1 =
    1     0     0
    0     1     0
    0     0     1

>> A2 =[0 1 0; 0 0 1;0 0 0]

A2 =
    0     1     0
    0     0     1
    0     0     0

>> A= A1+A2

A =
    1     1     0
    0     1     1
    0     0     1

>> B1 = [0 1 2;0 -1 3;0 0 0]
```

```

B1 =
 0      1      2
 0      1      3
 0      0      0

>> B2=[-i i -i;0 0 i;0 0 i]

B2 =
 0 - 1.0000i   0 + 1.0000i   0 - 1.0000i
 0             0             0 + 1.0000i
 0             0             0 + 1.0000i

>> B=B1-B2

 0 + 1.0000i   1.0000- 1.0000i   2.0000 + 1.0000i
 0             -1.0000           3.0000 - 1.0000i
 0             0               0 - 1.0000i

>> C1=[1, -1, 0;-1,sqrt(2)*1,-sqrt(2)*i;0,0,-1]

 1.0000      -1.0000      0
 -1.0000      0 + 1.4142i   0 - 1.4142i
 0             0             -1.0000

>> C2=[0 2 1;1 0 0;1 -1 0]

Z2 =
 0      2      1
 1      0      0
 0      1      0

>> C=C1+C2

C =
 1.0000      1.0000      1.0000
 0             0 + 1.4142i   0 - 1.4142i
 1.0000      -1.0000     -1.0000

>> M1=A*B-B*A

M1 =
 0      -1.0000 - 1.0000i   2.0000
 0              0             1.0000 - 1.0000i
 0              0             0

>> M2=A^2+B^2+C^2

M2 =

```

```
2.0000      2.0000 + 3.4142i      3.0000 - 5.4142i
0 - 1.4142i -0.0000 + 1.4142i      0.0000 - 0.5858i
0          2.0000 - 1.4142i      2.0000 + 1.4142i

>> M3=A*B*C

M3 =
5.0000 + 1.0000i   -3.5858 + 1.0000i   -6.4142 + 1.0000i
3.0000 - 2.0000i   -3.0000 + 0.5858i   -3.0000 + 3.4142i
0 - 1.0000i         0 + 1.0000i       0 + 1.0000i

>> M4=sqrtn(A)+sqrtn(B)-sqrtn(C)

M4 =
0.6356 + 0.8361i   -0.3250 - 0.8204i   3.0734 + 1.2896i
0.1582 - 0.1521i   0.0896 + 0.5702i   3.3029 - 1.8025i
-0.3740 - 0.2654i  0.7472 + 0.3370i   1.2255 + 0.1048i

>> M5=expm(A)*(expm(B)+expm(C))

M5 =
14.1906 - 0.0822i   5.4400 + 4.2724i   17.9169 - 9.5842i
4.5854 - 1.4972i   0.6830 + 2.1575i   8.5597 - 7.6573i
3.5528 + 0.3560i   0.1008 - 0.7488i   3.2433 - 1.8406i

>> inv(A)

ans =
1     -1     1
0      1    -1
0      0     1

>> inv(B)

ans =
0 - 1.0000i   -1.0000 - 1.0000i   -4.0000 + 3.0000i
0           -1.0000                  1.0000 + 3.0000i
0                 0                   0 + 1.0000i

>> inv(C)

ans =
0.5000      0      0.5000
0.2500      0 - 0.3536i -0.2500
0.2500      0 + 0.3536i -0.2500

>> [A*inv(A) B*inv(B) C*inv(C)]

ans =
```

```

1      0      0      1      0      0      1      0      0
0      1      0      0      1      0      0      1      0
0      0      1      0      0      1      0      0      1

>> A'

ans =
0      0      0
0      1      0
0      1      1

>> B'

ans =
0 - 1.0000i      0      0
1.0000 + 1.0000i  -1.0000      0
2.0000 - 1.0000i  3.0000 + 1.0000i  0 + 1.0000i

>> C'

ans =
1.0000      0      1.0000
1.0000  0 - 1.4142i  -1.0000
1.0000  0 + 1.4142i  -1.0000

```

Ejercicio 7. Dadas las matrices:

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad B = \begin{bmatrix} 1+i & 2+i \\ 3+i & 4+i \end{bmatrix}$$

Aplicar las funciones seno, exponencial de base, logaritmo, raíz cuadrada, módulo, argumento y funciones de redondeo a sus elementos.

Calcular  $e^B$  y  $\ln(A)$ .

```
>> A=[1 2 3; 4 5 6; 7 8 9]
```

```

A =
0      2      3
4      5      6
7      8      9

>> sin(A)

ans =

```

```
0.8415      0.9093      0.1411
-0.7568     -0.9589     -0.2794
 0.6570      0.9894      0.4121

>> B=[1+i 2+i;3+i,4+i]

B =
1.0      + 1.0000i  2.0000 + 1.0000i
3.0      + 1.0000i  4.0000 + 1.0000i

>> sin(B)

ans =
1.2985 + 0.6350i  1.4031 - 0.4891i
0.2178 - 1.1634i -1.1678 - 0.7682i

>> exp (A)

ans =
1.0e+003 *
0.0027      0.0074      0.0201
0.0546      0.1484      0.4034
1.0966      2.9810      8.1031

>> exp (B)

ans =
1.4687 + 2.2874i  3.9923 + 6.2177i
10.8523 + 16.9014i 29.4995 + 45.9428i

>> log(B)

ans =
0.3466 + 0.7854i  0.8047 + 0.4636i
1.1513 + 0.3218i  1.4166 + 0.2450i

>> sqrt(B)

ans =
1.0987 + 0.4551i  1.4553 + 0.3436i
1.7553 + 0.2848i  2.0153 + 0.2481i

>> abs(B)

ans =
1.4142      2.2361
3.1623      4.1231

>> imag(B)
```

```

ans=
1      1
1      1

>> fix(sin(B))
ans =
1.0000      1.0000
0 - 1.0000i   - 1.0000

>>

ceil(log(A))

ans =
0      1      2
2      2      2
2      3      3

>> sign(B)
ans =
0.7071 + 0.7071i   0.8944 + 0.4472i
0.9487 + 0.3162i   0.9701 + 0.2425i

```

Las funciones exponencial, raíz cuadrada y logaritmo usadas anteriormente se aplican elemento a elemento a la matriz, y no tienen nada que ver con las funciones matriciales exponenciales y logarítmicas que se usan a continuación.

```

>> expm(B)
ans =
1.0e+002 *
-0.3071 + 0.4625i -0.3583 + 0.6939i
-0.3629 + 1.0431i -0.3207 + 1.5102i

>> logm(A)
ans =
-5.6588 + 2.7896i   12.5041 - 0.4325i   -5.6325 - 0.5129i
12.8139 - 0.7970i   -23.3307 + 2.1623i   13.1237 - 1.1616i
-5.0129 - 1.2421i   13.4334 - 1.5262i   -4.4196 + 1.3313i

```

*Ejercicio 8. Resolver en el campo complejo la ecuación:*

$$\operatorname{Sen}(z) = 2$$

```

>> vpa(solve('sin(z)=2'))
ans =
1.315957896924815708625046347308*i + 1.5707963267948966192313216915398
1.5707963267948966192313216916398 - 1.316957896924815708625046347308*i

```

*Ejercicio 9. Resolver las siguientes ecuaciones:*

- a)  $1+x+x^2+x^3+x^4+x^5=0$
- b)  $x^2+(6-i)x+8-4i=0$
- c)  $\tan(Z)=3i/5$

```
>> solve ('1+x+x^2+x^3+x^4+x^5 = 0')
```

```
ans =
```

$$\begin{aligned} & -\frac{1}{2} - \frac{(3^{(1/2)}i)/2}{2} \\ & \frac{1}{2} - \frac{(3^{(1/2)}i)/2}{2} \\ & \frac{1}{2} + \frac{(3^{(1/2)}i)/2}{2} \\ & \frac{1}{2} + \frac{(3^{(1/2)}i)/2}{2} \end{aligned}$$

```
>> solve ('x^2+(6-i)*x+8-4*i = 0')
```

```
ans =
```

$$\begin{aligned} & -4 \\ & i - 2 \end{aligned}$$

```
>> vpa(solve('tan(Z) = 3*i/5'))
```

```
ans =
```

0.69314718055994530941723212145818\*i

*Ejercicio 10. Realizar las siguientes operaciones:*

- a) raíces cuartas de -1 y de 1
- b) raíces quintas de  $2+2i$  y de  $-1+i\sqrt{3}$
- c) parte real de  $\tan(i\ln((a+ib)/(a-ib)))$
- d) parte imaginaria de  $Z=(2+i)^{\cos(4-i)}$

```
>> solve('x^4+1=0')
```

```
ans =
```

$$\begin{aligned} & 2^{(1/2)} * (-i/2 - 1/2) \\ & 2^{(1/2)} * (i/2 - 1/2) \\ & 2^{(1/2)} * (1/2 - i/2) \\ & 2^{(1/2)} * (i/2 + 1/2) \end{aligned}$$

```
>> pretty(solve('x^4+1=0'))
```

$$\begin{aligned} & \pm \frac{1/2}{2} \sqrt{i} \frac{1}{2} \pm \frac{1}{2} \\ & \pm \frac{1/2}{2} \sqrt{-i} \frac{1}{2} \pm \frac{1}{2} \\ & \pm \frac{1/2}{2} \sqrt{2} \frac{1}{2} \pm \frac{1}{2} \end{aligned}$$

```

|      1/2 /1   i\|
|      2 | - - - | |
|          \2   2/  |
|          |
|      1/2 / i   1 \
|      2 | - + - | |
|          \ 2   2 /  |
+-           +-+
>>                               solve('x^4-1=0')

ans =
-1
1
-i
i

>> vpa(solve('x^5-2-2*i=0'))

0.19259341768888084906125263406469*i + 1.2159869826496146992458377919696
-0.87055056329612413913627001747975*i - 0.87055056329612413913627001747975
0.55892786746600970394985946846702*i - 1.0969577045083811131206798770216
0.55892786746600970394985946846702 - 1.0969577045083811131206798770216*i
1.2159869826496146992458377919696*i + 0.19259341768888084906125263406469

>> vpa(solve('x^5+1-sqrt(3)*i=0'))

ans =
0.46721771281818786757419290603946*i + 1.0493881644090691705137652947201
1.1424056652180689506550734259384*i - 0.1200716738059215411240904754285
0.76862922680258900220179378744147 - 0.85364923855044142809268986292246*i
- 0.99480195671282768870147766609475*i - 0.57434917749851750339931347338896
0.23882781722701229856490119703938*i - 1.1235965399072191281921551333441

>> simp1fy(vpa(real(tan(l*log((a+1*b) / (a-1*b))))))

ans =
- 0.5*tanh(conj(log((a^2 + 2.0*a*b*i - 1.0*b^2) / (a^2 + b^2)))*i +
(0.5*((a^2 + 2.0*a*b*i - 1.0*b^2)^2/(a^2 + b^2)^2 - 1)*i)/((a^2 +
2.0*a*b*i - 1.0*b^2)^2/(a^2 + b^2)^2 + 1)

>> simplify(vpa(imag ((2+i)^cos(4-i)))))

ans =
-0.62107490808037524310236676683417

```

## CAPITULO 4

# GRAFICOS

---

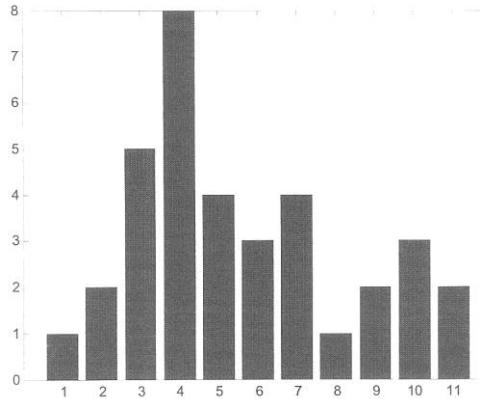
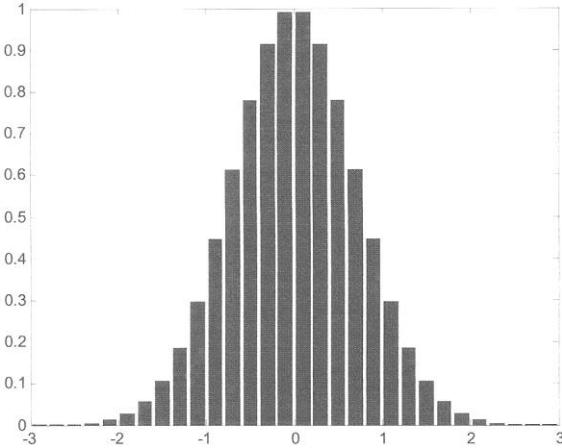
MATLAB es un software científico que implementa altas prestaciones gráficas. Permite realizar gráficos sencillos de análisis exploratorio de datos en dos y tres dimensiones, gráficos de curvas en coordenadas explícitas, implícitas y polares, gráficos de superficies en coordenadas explícitas, implícitas y paramétricas, gráficos de mallas y curvas de nivel, gráficos de volúmenes y gráficos especializados.

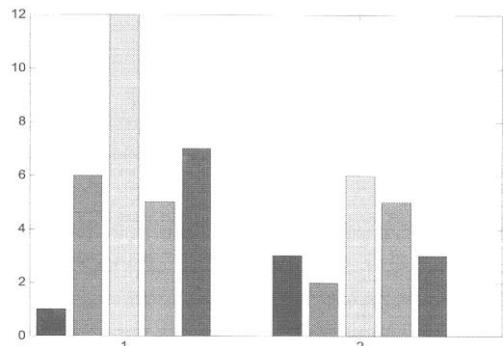
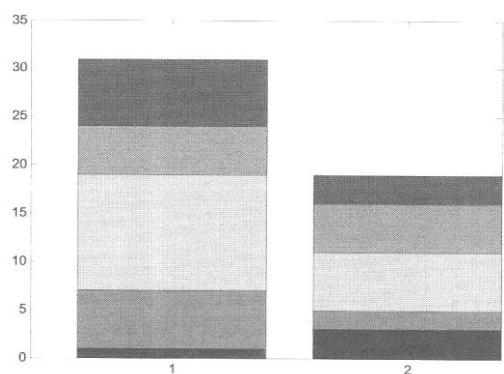
Asimismo, es posible elegir ventanas y posiciones para los gráficos, elegir características de líneas y marcadores, situar límites de ejes, marcas y mallas, situar anotaciones, etiquetas y leyendas, exportar gráficos a distintos formatos y otras posibilidades que se irán mostrando a lo largo del capítulo.

## GRÁFICOS EXPLORATORIOS DE DATOS

MATLAB incorpora comandos que permite realizar gráficos exploratorios básicos, como son los histogramas, gráficos de barras, gráficos de sectores, diagramas de flechas, etc. La tabla siguiente resume estos comandos. Para todos ellos es necesario definir previamente el campo de variación de la variable.

<b>bar(Y)</b>	<i>Gráfico de barras relativo al vector de frecuencias Y. Si Y es matriz se obtiene el gráfico de barras múltiple para cada fila de Y</i>  » x = [ 1 2 5 8 4 3 4 1 2 3 2 ] ; > > bar(x)
---------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

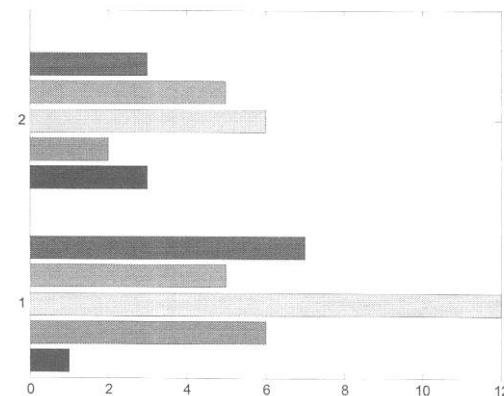
	
<b>bar(x,Y)</b>	<p>Gráfico de barras relativo al vector de frecuencias Y siendo x un vector que define los espacios en el eje X para situar las barras</p> <pre>» X = -2.9:0.2:2.9; » bar(x,exp(-x.*x))</pre> 
<b>bar(...,anchura)</b>	<p>Gráfico con anchura de las barras dada. Por defecto, la anchura es 0,8 y la anchura 1 provoca que las barras se toquen</p>
<b>bar(...,'estilo')</b>	<p>Gráfico con estilo para las barras dado. Los estilos son 'group' (estilo por defecto con barras verticales agrupadas) y 'stack' (barras apiladas). Si la matriz Y es (m,n), el gráfico agrupado tiene m grupos de n barras verticales</p> <pre>» A=[1 6 12 5 7; 3 2 6 5 3]; » bar(A, 'stack') » bar(A, 'qgroup')</pre>

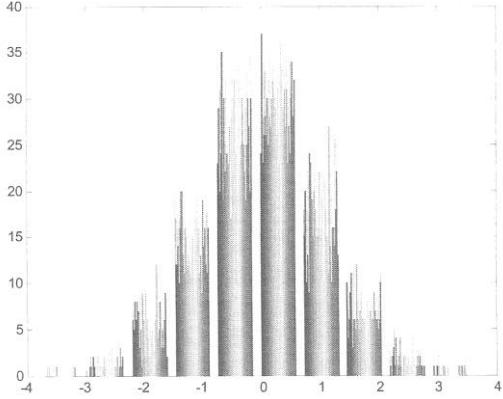
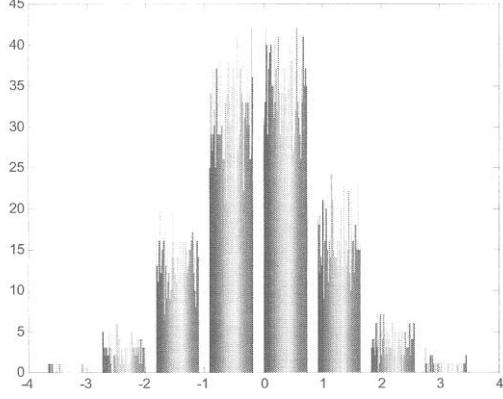


**bar(,,color)** Las barras son todas del color especificado (*r*=rojo, *g*=verde, *b*=azul, *c*=cyan, *m*=magenta y *y*=yellow, *k*=black y *w*=white)

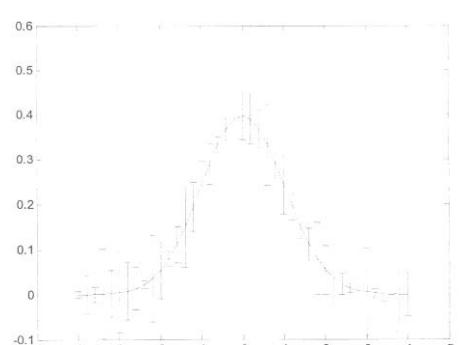
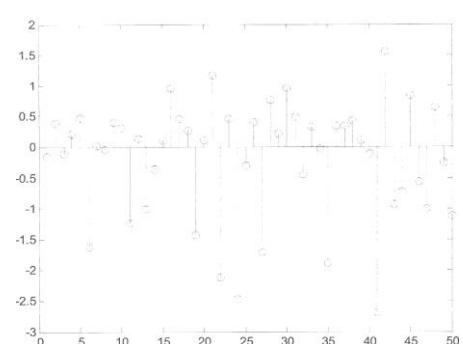
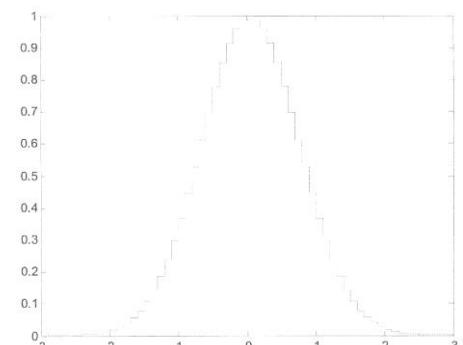
**barh(,,)** Gráficos de barras horizontales

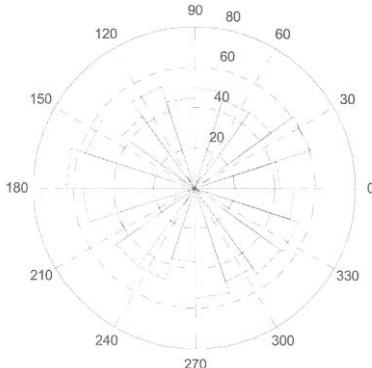
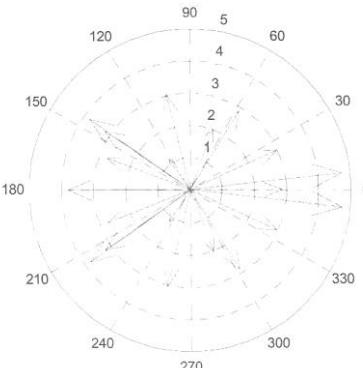
```
» barh(A, 'group')
```



<b>hist(Y)</b>	<i>Dibuja el histograma relativo al vector de frecuencias Y usando 10 rectángulos iguales de igual base. Si Y es una matriz, se construye un histograma para cada una de sus columnas</i>
	<pre>&gt;&gt; Y=randn(100); &gt;&gt; hist(Y)</pre> 
<b>hist(Y,x)</b>	<i>Dibuja el histograma relativo al vector de frecuencias Y usando tantas cajas como elementos tiene el vector x y centrándolas en los sucesivos valores de x</i>
<b>hist(Y,k)</b>	<i>Dibuja el histograma relativo al vector de frecuencias Y usando tantas cajas como indica el escalar k.</i>
	<pre>&gt;&gt; hist(Y, 8)</pre> 
<b>[n,x] = hist(...)</b>	<i>Devuelve los vectores n y x con las frecuencias asignadas a cada caja del histograma y los valores en los que se centran las cajas</i>

<b>pie(X)</b>	<p>Realiza el gráfico de sectores relativo al vector de frecuencias X</p> <pre>» x=[3 5 12 4 7 10]; » pie (X)</pre> <table border="1"> <thead> <tr> <th>Sector (%)</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>29%</td> </tr> <tr> <td>2</td> <td>17%</td> </tr> <tr> <td>3</td> <td>10%</td> </tr> <tr> <td>4</td> <td>12%</td> </tr> <tr> <td>5</td> <td>7%</td> </tr> <tr> <td>6</td> <td>24%</td> </tr> </tbody> </table>	Sector (%)	Percentage	1	29%	2	17%	3	10%	4	12%	5	7%	6	24%
Sector (%)	Percentage														
1	29%														
2	17%														
3	10%														
4	12%														
5	7%														
6	24%														
<b>pie(X,Y)</b>	<p>Realiza el gráfico de sectores relativo al vector de frecuencias X desplazando hacia fuera los sectores en los que <math>Y_i \neq 0</math></p> <pre>» pie(x, [0 0 1 0 1 1])</pre> <table border="1"> <thead> <tr> <th>Sector (%)</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>29%</td> </tr> <tr> <td>2</td> <td>17%</td> </tr> <tr> <td>3</td> <td>10%</td> </tr> <tr> <td>4</td> <td>12%</td> </tr> <tr> <td>5</td> <td>7%</td> </tr> <tr> <td>6</td> <td>24%</td> </tr> </tbody> </table>	Sector (%)	Percentage	1	29%	2	17%	3	10%	4	12%	5	7%	6	24%
Sector (%)	Percentage														
1	29%														
2	17%														
3	10%														
4	12%														
5	7%														
6	24%														
<b>errorbar(x,y,e)</b>	<p>Realiza el gráfico del vector x contra el vector y con los errores especificados en el vector e. Pasando por cada punto <math>(x_i, y_i)</math> se dibuja una línea vertical de longitud <math>2ei</math> cuyo centro es el punto <math>(x_i, y_i)</math></p> <pre>&gt;&gt; X = -4:.2:4; y=(1/sqrt(2*pi))*exp(-(x.^2)/2); e=rand(size(x))/10; errorbar(x,y,e)</pre>														

	
<b>stem(Y)</b>	<i>Dibuja el gráfico de racimo relativo al vector Y. Cada punto de Y es unido al eje x por una línea vertical</i> <pre>&gt;&gt; y=randn(50,1); stem(y)</pre> 
<b>stem(X,Y)</b>	<i>Dibuja el gráfico de racimo relativo al vector Y cuyos elementos son dados a través del vector X</i>
<b>stairs(Y)</b>	<i>Dibuja el gráfico escalonado relativo al vector Y</i>
<b>stairs(X,Y)</b>	<i>Gráfico escalonado del vector Y con elementos a través del vector X</i> <pre>&gt;&gt; x=-3:0.1:3; stairs(x,exp(-x.^2))</pre> 

<b>rose(Y)</b>	Dibuja el histograma angular relativo al vector Y, de ángulos en radianes utilizando 20 radios iguales  » <code>y=randn(1000,1)*pi; rose(y)</code>
	
<b>rose(Y,n)</b>	Dibuja el histograma angular del vector Y usando n radios iguales
<b>rose(X,Y)</b>	Dibuja el histograma angular relativo al vector Y utilizando radios que miden lo especificado en los elementos del vector X
<b>compas(Z)</b>	Realiza un diagrama de flechas que salen del origen y cuya magnitud y dirección vienen determinadas por el módulo y el argumento de los números complejos componentes del vector Z. La flecha relativa al complejo Zi une el origen con el afijo de Zi  » <code>z=eig(randn(20,20)); compass(z)</code>
	
<b>compas(X,Y)</b>	Equivale a <code>compas(X+i*Y)</code>
<b>compas(Z,S) o compas(X,Y,S)</b>	Especifica en S el tipo de linea a usar en las flechas

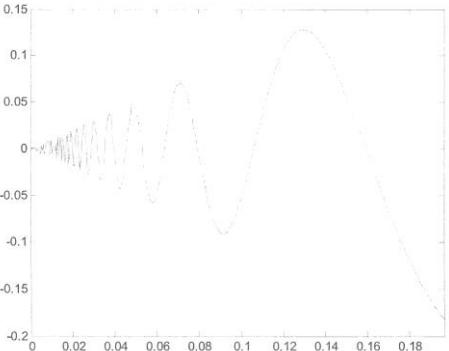
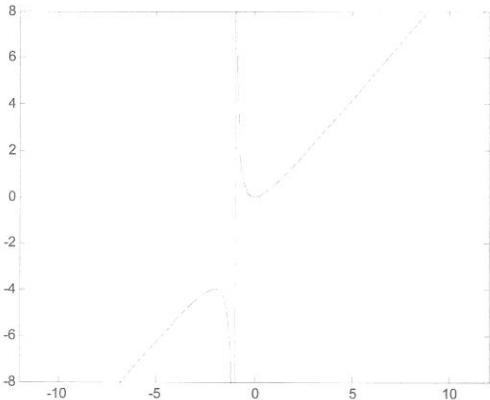
<b>feather(Z) o</b> <b>feather(X,Y) o</b> <b>feather(Z,S) o</b> <b>feather(X,Y,S)</b>	<p>Es igual que compás, con la única diferencia de que el origen de las flechas no está en el origen de coordenadas, sino que salen de puntos igualmente espaciados de una línea horizontal</p> <pre>&gt;&gt; z=eig(randn(20,20)); feather(z)</pre>
------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## CURVAS EN COORDENADAS EXPLÍCITAS, IMPLÍCITAS, PARAMÉTRICAS Y POLARES

Los comandos más importantes de MATLAB para representar curvas en dos dimensiones en coordenadas explícitas, implícitas y polares se presentan en la tabla siguiente.

<b>plot(X,Y)</b>	<p>Dibuja el conjunto de puntos <math>(X, Y)</math>, donde <math>X</math> e <math>Y</math> son vectores fila. Para graficar una función <math>y=f(x)</math> es necesario conocer un conjunto de puntos <math>(X, f(X))</math>, para lo que hay que definir inicialmente un intervalo de variación vectorial <math>X</math> para la variable <math>x</math>. <math>X</math> e <math>Y</math> pueden ser matrices de la misma dimensión, en cuyo caso se hace una gráfica por cada par de filas y sobre los mismos ejes. Para valores complejos de <math>X</math> e <math>Y</math> se ignoran las partes imaginarias. Para <math>x=x(t)</math> e <math>y = y(f)</math> con la variación de <math>t</math> dada, gráfica la curva paramétrica plana especificada</p> <pre>&gt;&gt; x=0:0.1:6*pi; y=x.*sin(x); plot(x,y)</pre>
------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>plot(Y)</b>	<p>Gráfica los elementos del vector <math>Y</math> contra sus índices, es decir, da la gráfica del conjunto de puntos <math>(t,y)</math>, <math>t=1,2,\dots,n</math> (<math>n=\text{length}(Y)</math>). Es útil para graficar series temporales. Si <math>Y</math> es una matriz, <math>\text{plot}(Y)</math> realiza un gráfico para cada columna de <math>Y</math>, presentándolos todos sobre los mismos ejes. Si los componentes del vector <math>Y</math> son complejos, <math>\text{plot}(Y)</math> es equivalente a <math>\text{plot}(\text{real}(Y),\text{imag}(Y))</math></p> <pre>» Y=[1, 3, 9, 27, 81, 243, 729]; plot (Y)</pre>
<b>plot(X,Y,S)</b>	<p>Gráfica de <math>\text{plot}(X,Y)</math> con las opciones definidas en <math>S</math>. Usualmente, <math>S</math> se compone de dos dígitos entre comillas simples, el primero de los cuales fija el color de la línea del gráfico y el segundo el carácter a usar en el graficado. Los valores posibles de colores y caracteres son, respectivamente, los siguientes: y (amarillo), m (magenta), c (cyan), r (rojo), g (verde), b (azul), w (blanco), k (negro), . (puntos), o (círculos), x (x-marcas), + (signo más), - (sólido), * (estrellas), : (dos puntos), -. (guiones y punto) y - (semisólido)</p> <pre>» plot([1,2,3,4,5,6,7,8,9], [1, 1/2, 1/3,1/4,1/5,1/6, 1/7,1/8,1/9], 'r *')</pre>

<b>plot(X1,Y1,S1,X2,Y2,S2,...)</b>	Combina, sobre los mismos ejes, los gráficos definidos para las tripletas $(X_i, Y_i, S_i)$ . Se trata de una forma de representar varias funciones sobre el mismo gráfico
<b>fplot('f',[xmin, xmax])</b>	Grafica la función explícita $y=f(x)$ en el intervalo de variación de $x$ dado <pre>» fplot ('x*sin(1/x) ', [0,pi/16])</pre> 
<b>fplot('f',[xmin, xmax, ymin, ymax], S)</b>	Grafica la función explícita $y=f(x)$ en los intervalos de variación de $x$ e $y$ dados, con las opciones de color y caracteres dadas por $S$ <pre>» fplot ('x^2/(x+1)', [-12,12,-8, 8])</pre> 
<b>fplot('f,[xmin, xmax],..,t)</b>	Grafica $f$ con la tolerancia $t$
<b>fplot('f,[xmin, xmax],..,n)</b>	Grafica $f$ con la tolerancia $t$ como $n+I$ puntos como mínimo
<b>fplot(['f1,f2,...,fn ],[xmin, xmax, ymin, ymax], S)</b>	Grafica las funciones $f_1, f_2, \dots, f_0$ sobre los mismos ejes en los intervalos de variación de $x$ e $y$ especificados y con las opciones de color y caracteres definidas en $S$

	<pre>» fplot ('[sin(x), sin(2*x), sin(3*x)]', [0,2*pi])</pre>
	<pre>» fplot ('[sin(x), sin(2*x), sin(3*x)]', [0,2*pi], 'k*')</pre>
<b>ezplot('r, [xmin xmax])</b>	<p>Gráfica la función explícita <math>y=f(x)</math> o implícita <math>f(x,y)=k</math> en el intervalo de variación de <math>x</math> dado. El intervalo de variación de la variable puede omitirse</p> <pre>» ezplot ('y*x''2+x*y^2=10', [-10,10])</pre>

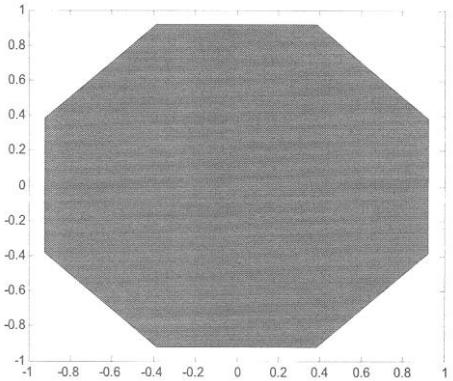
	<pre>&gt;&gt; ezplot ('x^2/(x^2-1)')</pre>
<code>ezplot('f',[xmin, xmax, ymin, ymax])</code>	<p>Gráfica la función explícita <math>y=f(x)</math> o implícita <math>f(x,y)=k</math> en los intervalos de variación de <math>x</math> e <math>y</math> dados (o no dados)</p> <pre>&gt;&gt; ezplot ('x^2+y^3=1/2', [-10,10,-8,8]) &gt;&gt; ezplot ('x^2-y^4=1')</pre> 

<b>ezplot(x,y)</b>	<p>Gráfica la curva paramétrica plana <math>x = x(t)</math> e <math>y = y(t)</math> sobre el dominio <math>0 &lt; t &lt; 2\pi</math></p> <pre>» ezplot('4*cos(t)-cos(4*t)', '4*sin(t)-sin(4*t)')</pre>
<b>ezplot('f', [xmin xmax])</b>	<p>Gráfica la curva paramétrica plana <math>x = x(t)</math> e <math>y = y(t)</math> sobre el dominio <math>xmin &lt; t &lt; xmax</math></p> <pre>» ezplot('t*sin(t)', 't*cos(t)', [-4*pi, 4*pi]) x = t sin(t), y = t cos(t)</pre>
<b>ezplot('f')</b>	<p>Gráfica la curva <math>f</math> en coordenadas implícitas en <math>[-2\pi, 2\pi]</math></p> <pre>» ezplot ('y^4-x^4-24*y^2+25*x^2=0')</pre>

<b>loglog(X,Y)</b>	<p>Realiza gráficos similares a <code>plot(X,Y)</code>, pero con escala logarítmica en los dos ejes</p> <pre>&gt;&gt; x=0:0.1:pi; y=x.*sin(x); loglog(x,y)</pre>
<b>semilogx(X,Y)</b>	<p>Realiza gráficos similares a <code>plot(X,Y)</code>, pero con escala logarítmica en el eje X y escala normal en el eje Y</p> <pre>&gt;&gt; x=0:0.1:pi; y=x.*sin(x); semilogx(x,y)</pre>

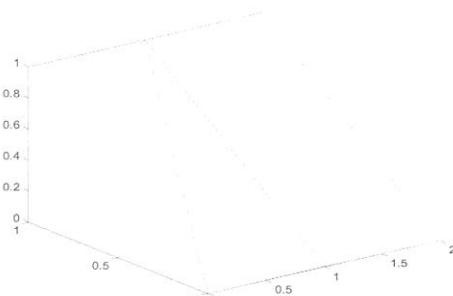
<b>semilogy(X,Y)</b>	<p>Realiza gráficos similares a <code>plot(X,Y)</code>, pero con escala logarítmica en el eje Y y escala normal en el eje X</p> <pre>» x=0:0.1:pi; y=x.*sin(x); semilogy(x,y)</pre>
<b>polar(a,r)</b>	<p>Dibuja la curva en coordenadas polares <math>r=r(\alpha)</math></p> <pre>» t=0:0.1:2*pi;r=sin(t).*cos(t); polar(t,r)</pre>
<b>polar(a,r,S)</b>	<p>Dibuja la curva en coordenadas polares <math>r=r(\alpha)</math> con el estilo de líneas dado por S</p> <pre>» t=0:0.05:2*pi;r=sin(t).*cos(t); polar(t,r,'*r')</pre>

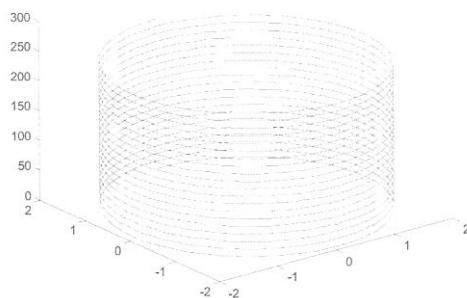
<b>ezpolar(r)</b> <b>ezpolar(r,[a,b])</b>	<p>Dibuja la curva en coordenadas polares <math>r = r(\alpha)</math> con el campo de variación de <math>\alpha</math> en <math>[0 2\pi]</math> si no se especifica nada o entre <math>a</math> y <math>b</math> en su caso</p> <pre>&gt;&gt; ezpolar('1+cos(t)')</pre> <p style="text-align: center;"><math>r = 1 + \cos(t)</math></p>
<b>fill(X,Y,C)</b>	<p>Dibuja el polígono compacto cuyos vértices son los pares de componentes <math>(X_i, Y_i)</math> de los vectores columnas <math>X</math> e <math>Y</math>. <math>C</math> es un vector de la misma dimensión de <math>X</math> e <math>Y</math>, que contiene los colores <math>C_i</math> de cada punto <math>(X_i, Y_i)</math>. Los valores de <math>C_i</math> pueden ser: '<math>r</math>', '<math>g</math>', '<math>b</math>', '<math>c</math>', '<math>m</math>', '<math>y</math>', '<math>w</math>', '<math>k</math>', cuyos significados ya conocemos. Si <math>C</math> es un solo carácter, se pintarán todos los puntos del polígono del color correspondiente al carácter. Si <math>X</math> e <math>Y</math> son matrices de la misma dimensión, se representarán simultáneamente varios polígonos correspondientes a cada par de vectores columna <math>(X_{ij}, Y_{ij})</math>. En este caso, <math>C</math> puede ser un vector fila cuyos elementos <math>C_j</math> determinan el color único de cada polígono correspondiente al par de vectores columna <math>(X_{ij}, Y_{ij})</math>. <math>C</math> puede ser también una matriz de la misma dimensión que <math>X</math> e <math>Y</math>, en cuyo caso sus elementos determinan los colores de cada punto <math>(X_{ij}, Y_{ij})</math> del conjunto de polígonos</p> <pre>&gt;&gt; t = (1/16:1/8:1)*2*pi; x = sin(t); y = cos(t); fill(x,y, 'r')</pre>

	
<b>fill(X1,Y1,C1,...)</b>	Dibuja el polígono compacto cuyos vértices vienen dados por los puntos $(X_i, Y_i, C_i)$

## CURVAS ALABEADAS

MATLAB incorpora comandos que permiten representar curvas alabeadas en tres dimensiones. En la tabla siguiente se presentan los más importantes.

<b>plot3(X,Y,Z)</b>	<p>Dibuja el conjunto de puntos <math>(X, Y, Z)</math>, donde <math>X</math> <math>Y</math> y <math>Z</math> son vectores fila. <math>X</math>, <math>Y</math> y <math>Z</math> pueden ser coordenadas paramétricas o matrices de la misma dimensión, en cuyo caso se hace una gráfica por cada tripleta de filas y sobre los mismos ejes. Para valores complejos de <math>X</math>, <math>Y</math> y <math>Z</math> se ignoran las partes imaginarias</p> <pre>» X = [0 1 1 2; 1 1 2 2; 0 0 1 1]; Y = [1 1 1 1; 1 0 1 0; 0 0 0 0]; Z = [1 1 1 1; 1 0 1 0; 0 0 0 0]; » plot3(X,Y,Z)</pre>  <pre>&gt;&gt; t=0:pi/100:2*pi; plot3(2*sin(2*t), 2*cos(2*t), 4*t)</pre>
---------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------



<b>plot3(X,Y,Z,S)</b>	Gráfica de $\text{plot}(X,Y,Z)$ con las opciones definidas en $S$ . Usualmente $S$ se compone de dos dígitos entre comillas simples, el primero de los cuales fija el color de la línea del gráfico y el segundo el carácter a usar en el graficado. Los valores posibles de colores y caracteres se han descrito ya al explicar el comando $\text{plot}$
<b>plot3(X1,Y1,Z1,S1, X2,Y2,Z2,S2, X3,Y3,Z3,S3,...)</b>	Combina, sobre los mismos ejes, los gráficos definidos para las tripletas $(X_i, Y_i, Z_i, S_i)$ . Se trata de una forma de representar varias funciones sobre el mismo gráfico
<b>fill3(X,Y,Z,C)</b>	Dibuja el polígono compacto cuyos vértices son las tripletas de componentes $(X_i, Y_i, Z_i)$ de los vectores columna $X$ , $Y$ y $Z$ . $C$ es un vector de la misma dimensión de $X$ , $Y$ y $Z$ , que contiene los colores $C_i$ de cada punto $(X_i, Y_i, Z_i)$ . Los valores de $C_i$ pueden ser ' $r$ ', ' $g$ ', ' $b$ ', ' $c$ ', ' $m$ ', ' $y$ ', ' $w$ ', ' $k$ ', cuyos significados ya conocemos. Si $C$ es un sólo carácter, se pintarán todos los puntos del polígono del color correspondiente al carácter. Si $X$ , $Y$ y $Z$ son de la misma dimensión, se representarán simultáneamente varios polígonos correspondientes a cada tripleta de vectores columna $(X_j, Y_j, Z_j)$ . En este caso, $C$ puede ser un vector fila cuyos elementos $C_j$ determinan el color único de cada polígono correspondiente a la tripleta de vectores columna $(X_j, Y_j, Z_j)$ . $C$ puede ser también una matriz de la misma dimensión que $X$ , $Y$ y $Z$ , en cuyo caso sus elementos determinan los colores de cada punto $(X_{ijk}, Y_{ijk}, Z_{ijk})$ del conjunto de polígonos

```
» X = [ 0 1 1 2 ; 1 1 2 2 ; 0 0 1 1 ] ;
Y= [ 1 1 1 1;1 0 1 0;0 0 0 0];
Z = [ 1 1 1 1 ; 1 0 1 0 ; 0 0 0 0 ] ;
C = [0.5000 1.0000 1.0000 0.5000;
      1.0000 0.5000 0.5000 0.1667;
      0.3330 0.3330 0.5000 0.5000]; fill3(X,Y,Z,C)
```

<b>fill3(X1,Y1,Z1,C1, X2,Y2,Z2, C2,...)</b>	Dibuja el polígono compacto cuyos vértices vienen dados por los puntos $(X_i, Y_i, Z_i, C_i)$
<b>ezplot3(x(t),y(t),z(t))</b>  <b>ezplot3(x(t),y(t),z(t), [tmin,tmax])</b>	<p>Dibuja la curva alabeada definida por sus tres componentes paramétricas</p> <p>Dibuja la curva alabeada definida por sus tres componentes paramétricas para una campo de variación del parámetro dado</p> <pre>» ezplot3('sin(t)', 'cos(t)', 't', [0,6*pi]) x = sin(t), y = cos(t), z = t</pre>

## SUPERFICIES EXPLICITAS Y PARAMÉTRICAS. CURVAS DE NIVEL

MATLAB incorpora comandos que permiten representar superficies de ecuación  $z=f(x,y)$ . El primer paso consiste en utilizar el comando *meshgrid*, que define la matriz de puntos  $(X, Y)$  sobre los cuales se evalúa la función para representarla.

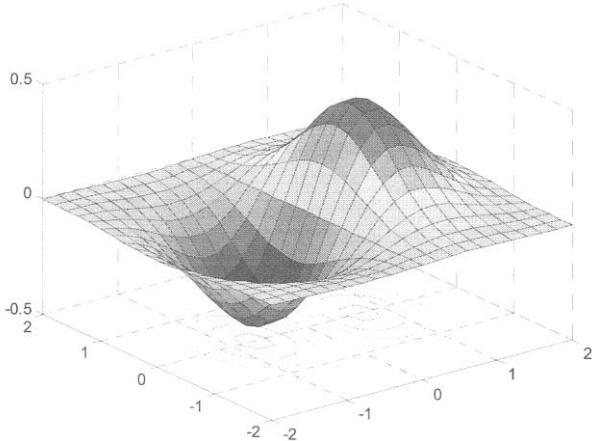
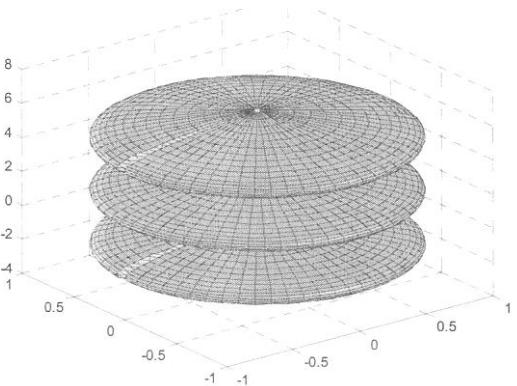
A continuación se utiliza el comando *surf* para realizar la representación de la superficie.

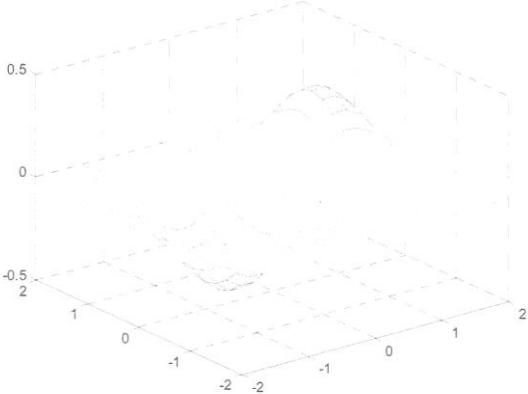
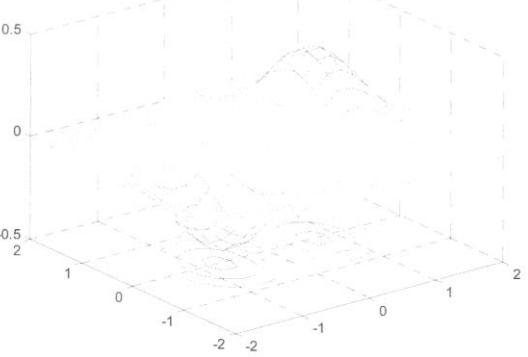
También se utiliza el comando *mesh* para representar un gráfico de malla, que viene definido por una función  $z=f(x,y)$ , de tal forma que los puntos de la superficie se representan sobre una rejilla, resultado de levantar los valores de  $z$  dados por  $f(x,y)$  sobre los correspondientes puntos del plano  $(x,y)$ . El aspecto de un gráfico de malla es como una red de pesca, con los puntos de la superficie sobre los nudos de la red. Realmente, es un gráfico de superficie cuyo grafo tiene forma de red.

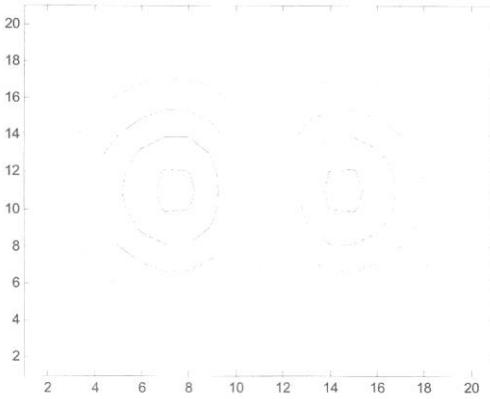
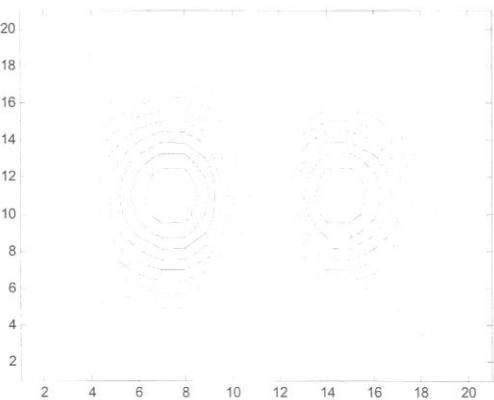
Asimismo, es posible representar las curvas de nivel de una superficie mediante el comando *contour*. Estas curvas se caracterizan porque son puntos  $(x,y)$  sobre las cuales el valor de  $f(x,y)$  es constante.

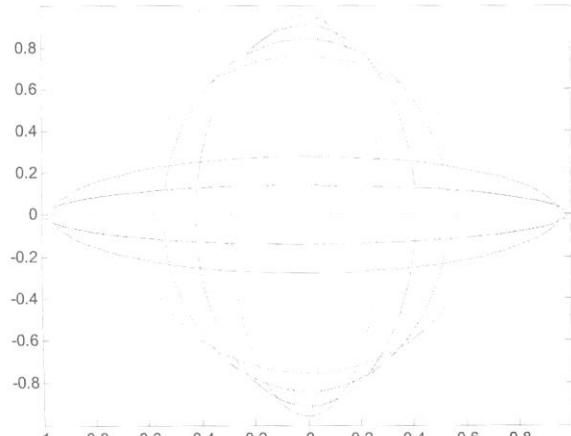
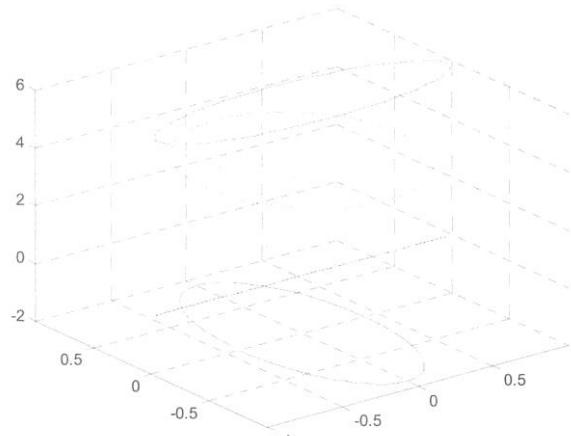
La tabla siguiente muestra los comandos de MATLAB para representar mallas, curvas de nivel y superficies, tanto en coordenadas explícitas como paramétricas.

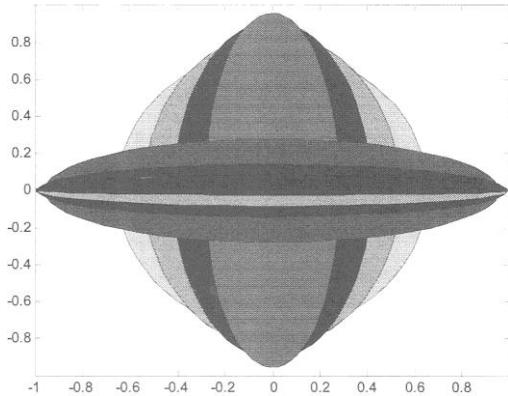
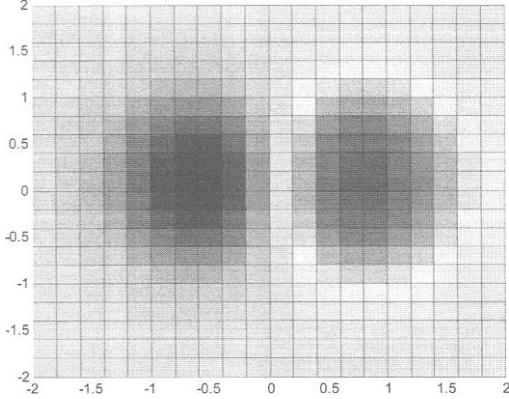
<b>[X,Y] = meshgrid(x,y)</b>	Transforma el campo de definición dado de las variables $x$ e $y$ de la función a representar $z=f(x,y)$ en argumentos matriciales utilizables por los comandos <i>surf</i> y <i>mesh</i> para obtener gráficos de superficie y malla, respectivamente
<b>surf(X,Y,Z,C)</b>	<p>Representa la superficie explícita <math>z=f(x,y)</math> o la paramétrica <math>x=x(t,u)</math>, <math>y=y(t,u)</math>, <math>z=z(t,u)</math>, realizando el dibujo con los colores especificados en <math>C</math>. El argumento <math>C</math> se puede ignorar</p> <pre>» [X, Y] = meshgrid(-2:.2:2, -2:.2:2); Z = X .* exp(-X.^2 - Y.^2) ; surf(X, Y, Z)</pre>

<b>surf(X,Y,Z,C)</b>	<p>Representa la superficie explícita <math>z=f(x,y)</math> o la paramétrica <math>x=x(t,u)</math>, <math>y=y(t,u)</math>, <math>z=z(t,u)</math>, junto con el gráfico de contorno correspondiente (curvas de nivel proyectadas sobre el plano XY)</p> <pre>&gt;&gt; [X, Y] = meshgrid(-2:.2:2, -2:.2:2); Z = X .* exp(-X.^2 - Y.^2); surf(X, Y, Z)</pre> 
<b>surfl(X,Y,Z)</b>	<p>Representa la superficie explícita <math>z=f(x,y)</math> o la paramétrica <math>x=x(t,u)</math>, <math>y=y(t,u)</math>, <math>z=z(t,u)</math>, con el dibujo con sombreado</p> <pre>&gt;&gt; r=(0:0.1:2*pi)'; t=(-pi:0.1:2*pi); X=cos(r)*sin(t); Y=sin(r)*sin(t); Z=ones(1,size(r))'*t; surfl(X, Y, Z)</pre> 

<b>mesh(X,Y,Z,C)</b>	<p>Representa la superficie explícita <math>z=f(x,y)</math> o la paramétrica <math>x=x(t,u), y=y(t,u), z=z(t,u)</math>, dibujando las líneas de la rejilla que componen la malla con los colores especificados en <math>C</math> (opcional)</p> <pre><code>&gt;&gt; [X, Y] = meshgrid(-2:.2:2, -2:.2:2); Z = X .* exp(-X.^2 - Y.^2); mesh(X, Y, Z)</code></pre> 
<b>meshz(X,Y,Z,C)</b>	<p>Representa la superficie explícita <math>z=f(x,y)</math> o la paramétrica <math>x=x(t,u), y=y(t,u), z=z(t,u)</math> con una especie de cortina o telón en la parte inferior</p>
<b>meshc(X,Y,Z,C)</b>	<p>Representa la superficie explícita <math>z=f(x,y)</math> o la paramétrica <math>x=x(t,u), y=y(t,u), z=z(t,u)</math> junto con el gráfico de contorno correspondiente (curvas de nivel proyectadas sobre el plano XY)</p> <pre><code>&gt;&gt; [X, Y] = meshgrid(-2:.2:2, -2:.2:2); Z = X .* exp(-X.^2 - Y.^2); meshc(X, Y, Z)</code></pre> 

<b>contour(Z)</b>	<p>Dibuja el gráfico de contorno (curvas de nivel) para la matriz Z. El número de líneas de contorno a utilizar se elige automáticamente</p> <pre data-bbox="454 352 899 419">&gt;&gt; [X,Y] = meshgrid(-2:.2:2, -2:.2:2); Z = X .* exp(-X.^2 - Y.^2) ; &gt;&gt; contour(Z)</pre> 
<b>contour(Z,n)</b>	<p>Dibuja el gráfico de contorno (curvas de nivel) para la matriz Z usando n líneas de contorno</p> <pre data-bbox="454 1023 899 1090">&gt;&gt; [X,Y] = meshgrid(-2:.2:2, -2:.2:2); Z = X .* exp(-X.^2 - Y.^2); &gt;&gt; contour(Z)</pre> 
<b>contour(x,y,Z,n)</b>	<p>Dibuja el gráfico de contorno (curvas de nivel) para la matriz Z usando en los ejes X e Y el escalado definido por los vectores x e y (n líneas de contorno)</p>

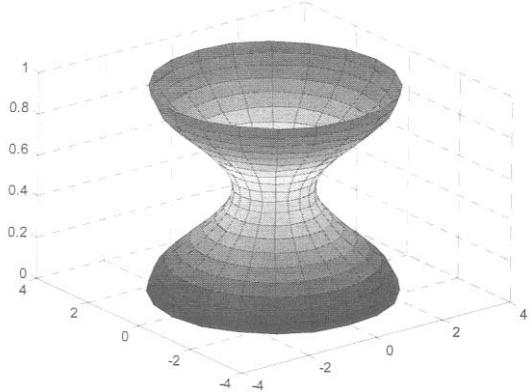
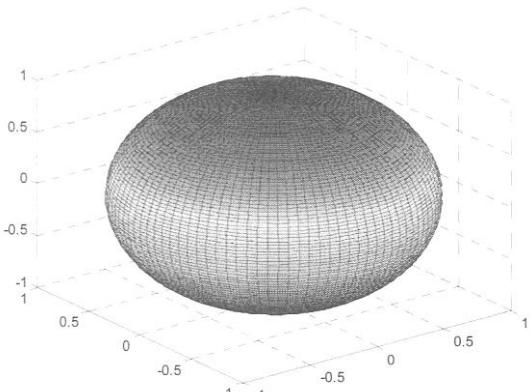
	<pre>» r= (0:0.1:2*pi)';t=(-pi:0.1:2*pi); X=cos(r)*cos(t);Y=sin(r)*sin(t);Z=ones(1,size(r))'*t; &gt;&gt; contour (X,Y,Z)</pre> 
<b>contour3(Z), contour3(Z,n) y contour3(x,y,Z,n)</b>	<p><i>Dibujan los gráficos de contorno en 3 dimensiones</i></p> <pre>» r=(0:0.1:2*pi)';t=(-pi:0.1:2*pi); X=cos(r)*cos(t);Y=sin(r)*sin(t); Z=ones(1,size(r))'*t; &gt;&gt; contour3(X,Y,Z)</pre> 
<b>contourf(...)</b>	<p><i>Dibuja un gráfico de contorno y rellena las áreas entre las isolíneas</i></p> <pre>» r=(0:0.1:2*pi)';t=(-pi:0.1:2*pi); X=cos(r)*cos(t);Y=sin(r)*sin(t);Z=ones(1,size(r))'*t; &gt;&gt; contourf (X,Y,Z)</pre>

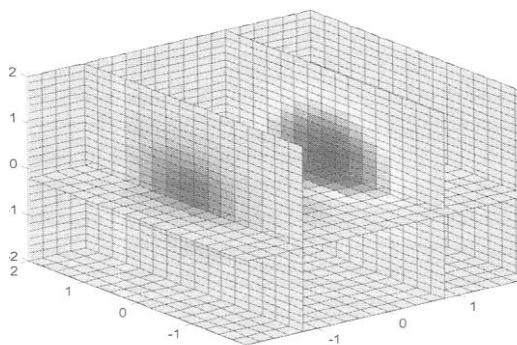
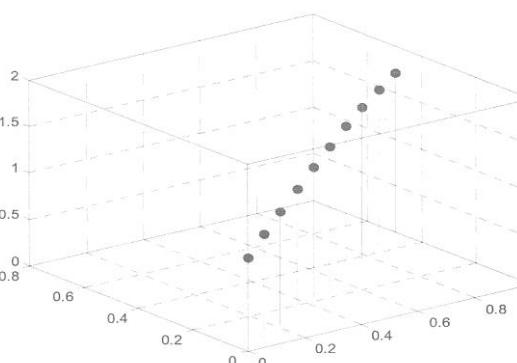
	
<b>pcolor(X,Y,Z)</b>	<p>Dibuja un gráfico de contorno (curvas de nivel) para la matriz <math>(X, Y, Z)</math> utilizando una representación basada en densidades de colores. Suele denominarse gráfico de densidad</p> <pre>&gt;&gt; [X,Y] = meshgrid(-2:.2:2, -2:.2:2); Z = X .* exp(-X.^2 - Y.^2); meshc(X,Y,Z) &gt;&gt; pcolor(X,Y,Z)</pre> 
<b>trimesh(Tri,X,Y,Z,C)</b>	Muestra triángulos definidos en la matriz $Tri$ como una malla. Cada fila de la matriz $Tri$ define una cara triangular simple y $C$ define los colores como en $surf$ . El argumento $C$ es opcional
<b>trisurf(Tri,X,Y,Z,C)</b>	Muestra triángulos definidos en la matriz $Tri$ como una superficie. Cada fila de la matriz $Tri$ define una cara triangular simple y $C$ define los colores como en $surf$ . El argumento $C$ es opcional

## FORMAS GEOMÉTRICAS TRIDIMENSIONALES

Con MATLAB es posible la representación de cilindros, esferas, gráficos de barras, secciones, tallos, cascada y otras formas geométricas tridimensionales. La tabla siguiente resume los comandos al efecto.

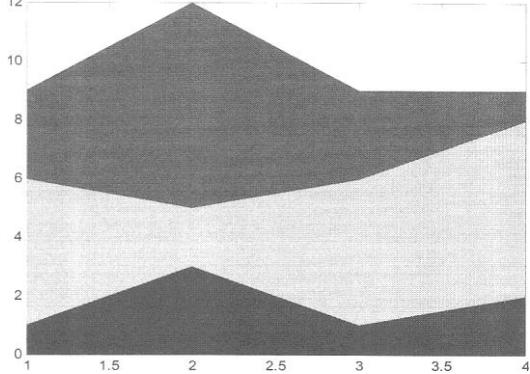
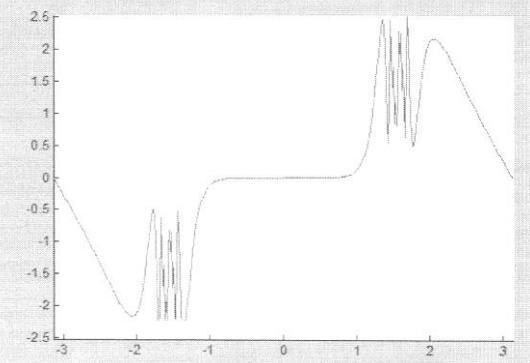
<b>bar3(Y)</b>	<p>Gráfico de barras relativo al vector de frecuencias Y. Si Y es matriz se obtiene el gráfico de barras múltiple para cada fila de Y</p> <pre>&gt;&gt; bar3(rand(4,4))</pre>
<b>bar3(x,Y)</b>	Gráfico de barras relativo al vector de frecuencias Y siendo x un vector que define los espacios en el eje X para situar las barras
<b>bar3(...,anchura)</b>	Gráfico con anchura de las barras dada. Por defecto, la anchura es 0,8 y la anchura 1 provoca que las barras se toquen
<b>bar3(...,'estilo')</b>	Gráfico con estilo para las barras dado. Los estilos son 'detached' (estilo por defecto) 'grouped' (estilo con barras verticales agrupadas) y 'stacked' (barras apiladas).
<b>bar3(,,,color)</b>	Las barras son todas del color especificado (r=rojo, g=verde, b=azul, c=cyan, m=magenta y=yellow, k=black y w=white)
<b>comet3(z)</b> <b>comet3(x,y,z)</b>	<p>Gráfico de cometa relativo al vector z Gráfico de cometa paramétrico (<math>x(t)</math>, <math>y(t)</math>, <math>z(t)</math>)</p> <pre>&gt;&gt; t = -pi:pi/500:pi; comet3(sin(5*t),cos(3*t),t)</pre>

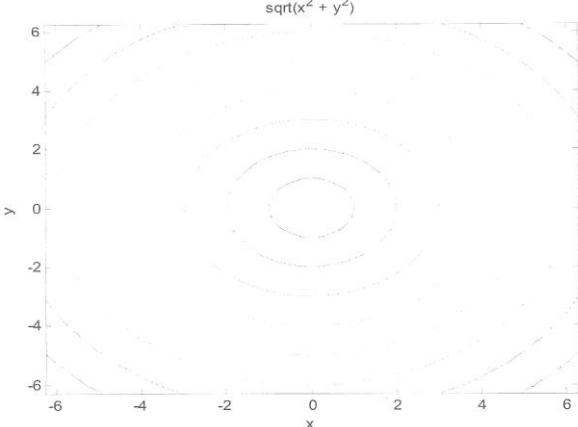
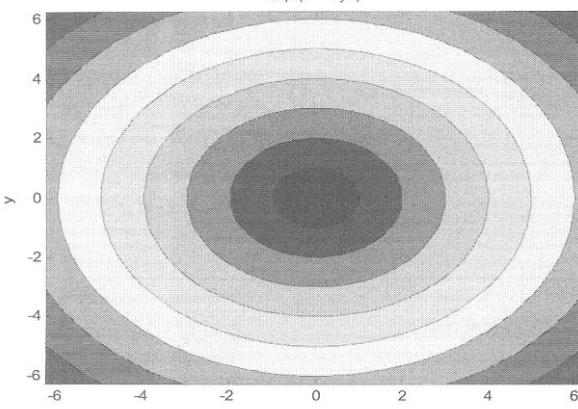
<p><b>[X,Y,Z]= cylinder</b>  <b>[X,Y,Z] = cylinder(r(t))</b>  <b>[X,Y,Z] = cylinder(r(t),n)</b></p> <p><b>cylinder(...)</b></p>	<p>Da las coordenadas del cilindro unidad Da las coordenadas del cilindro generado por la curva <math>r</math> Da las coordenadas del cilindro generado por la curva <math>r</math> con <math>n</math> puntos en la circunferencia sección horizontal alineado con el eje <math>Z</math> (<math>n= 20</math> por defecto)  <i>Gráfica los cilindros anteriores</i></p> <pre>» t = 0:pi/10:2*pi; [X, Y, Z] = cylinder(2+cos(t)); surf (X, Y, Z)</pre> 
<p><b>sphere</b></p>	<p><i>Gráfica la esfera unidad usando 20x20 caras</i></p>
<p><b>sphere(n)</b></p>	<p><i>Genera una esfera usando nxn caras</i></p> <pre>» sphere(100)</pre> 
<p><b>[X,Y,Z1] = sphere(n)</b></p>	<p><i>Da las coordenadas de la esfera en tres matrices (n+1)x(n+1) Dibuja cortes a lo largo de las direcciones x,y ,z en el volumen V (array mxnxp) definidos por los vectores (sx,sy,sz)</i></p>

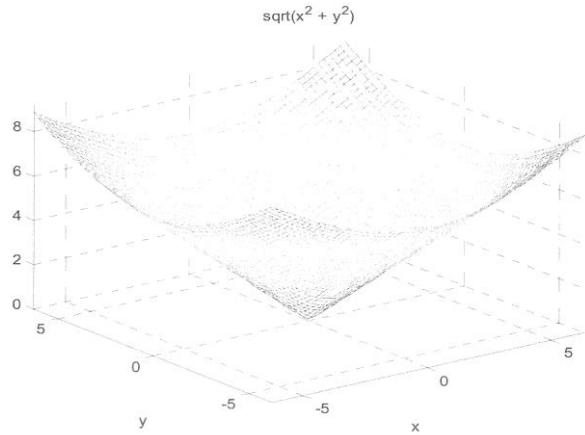
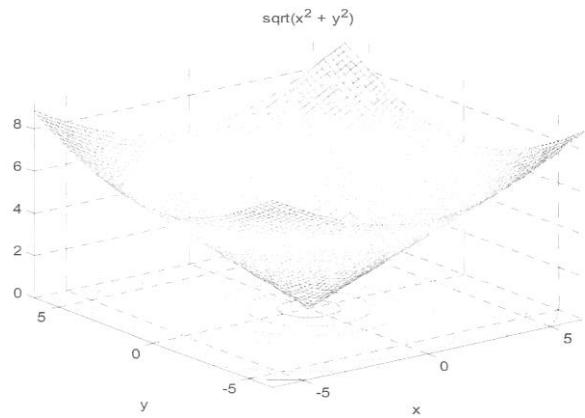
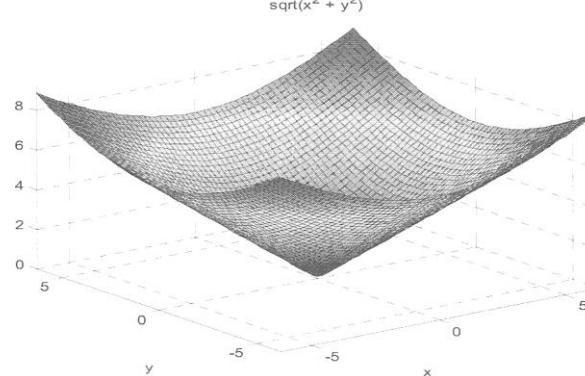
<b>slice(X,Y,Z,V,sx,sy,sz)</b> <b>slice(V,XI,YI,ZI)</b> <b>slice(X,Y7,V,XI,YI,ZI)</b> <b>slice(...,'método')</b>	<p>Dibuja cortes definidos por los vectores <math>(sx, sy, sz)</math> en el volumen <math>V</math> definido por los arrays tridimensionales <math>(X, Y, Z)</math></p> <p>Dibuja cortes en el volumen <math>V</math> definidos por las matrices <math>(XI, YI, ZI)</math> que generan una superficie</p> <p>Dibuja cortes definidos por las matrices <math>(XI, YI, ZI)</math> que generan una superficie en el volumen <math>V</math> definido por los arrays tridimensionales <math>(X, Y, Z)</math></p> <p>Dibuja cortes según el método de interpolación especificado (linear para lineal, cubic para cúbica y nearest para vecino)</p> <pre>&gt;&gt; [x,y,z] = meshgrid (-2:.25:2, -2:.25:2, -2:.16:2); V = X .* exp(-x.^2 - y.^2 - z.^2); slice(x,y,z,v, [-1.2 .8 2], 2, [-2 -.2])</pre> 
<b>stem3(Z)</b> <b>stem3(X,Y,Z)</b> <b>stem3(...,'fill')</b> <b>stem3(...,S)</b>	<p>Dibuja la secuencia <math>Z</math> como un gráfico de tallos en el plano <math>(x,y)</math></p> <p>Dibuja la secuencia <math>Z</math> en los valores especificados por <math>X</math> e <math>Y</math> Rellena de color los círculos de las puntas de los tallos Realiza el gráfico de tallos con las especificaciones de <math>S</math> (color, ...)</p> <pre>&gt;&gt; X = linspace(0,1,10); Y = X./2; Z = sin(X) + cos(Y); stem3(X,Y,Z,'fill')</pre> 

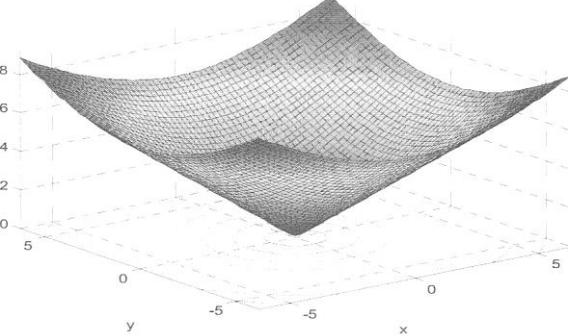
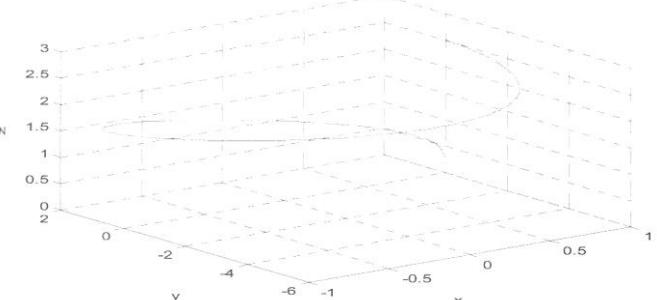
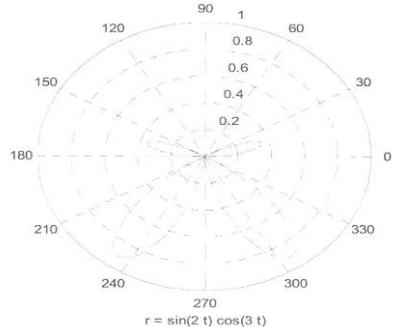
## GRÁFICOS ESPECIALIZADOS

MATLAB dispone de comandos para representar gráficos de áreas, cajas, sectores tridimensionales, gráficos de Pareto y otros gráficos especializados. En la tabla siguiente se presenta la sintaxis de estos comandos.

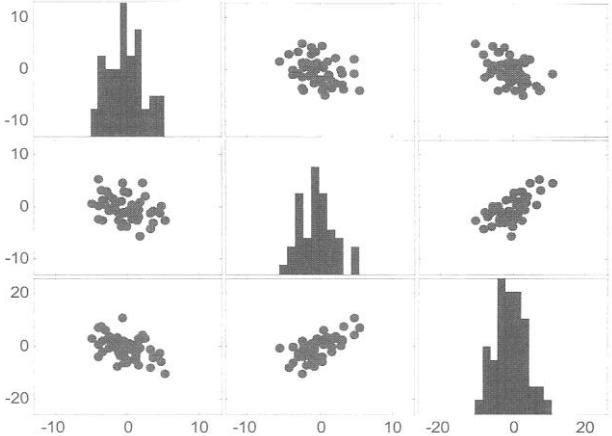
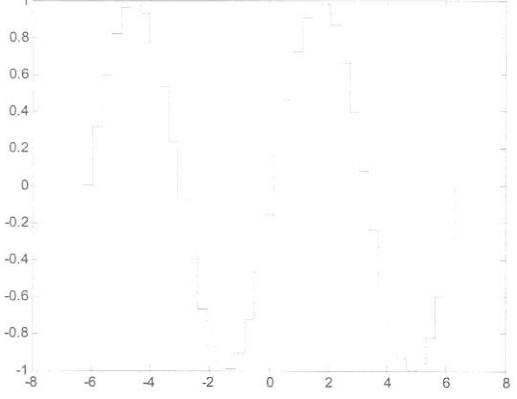
<b>area(Y)</b>	Realiza el gráfico de áreas relativo al vector de frecuencias Y
<b>area(X,Y)</b>	Realiza el gráfico de áreas relativo al vector de frecuencias Y cuyos elementos son dados a través del vector X
<b>area(...,ymin)</b>	Especifica el límite inferior en la dirección de y del área de relleno <pre>&gt;&gt; Y = [1, 5, 3; 3, 2, 7; 1, 5, 3; 2, 6, 1]; area(Y)</pre> 
<b>box on, box off</b>	Habilita y deshabilita cajas en los ejes para gráficos 2-D y 3-D
<b>comet(y)</b>	Realiza el gráfico de cometa relativo al vector de frecuencias Y
<b>comet(x,y)</b>	Realiza el gráfico de cometa relativo al vector de frecuencias Y cuyos elementos son dados a través del vector X <pre>&gt;&gt; t = -pi:pi/200:pi; comet(t,tan(sin(t))-sin(tan(t)))</pre> 

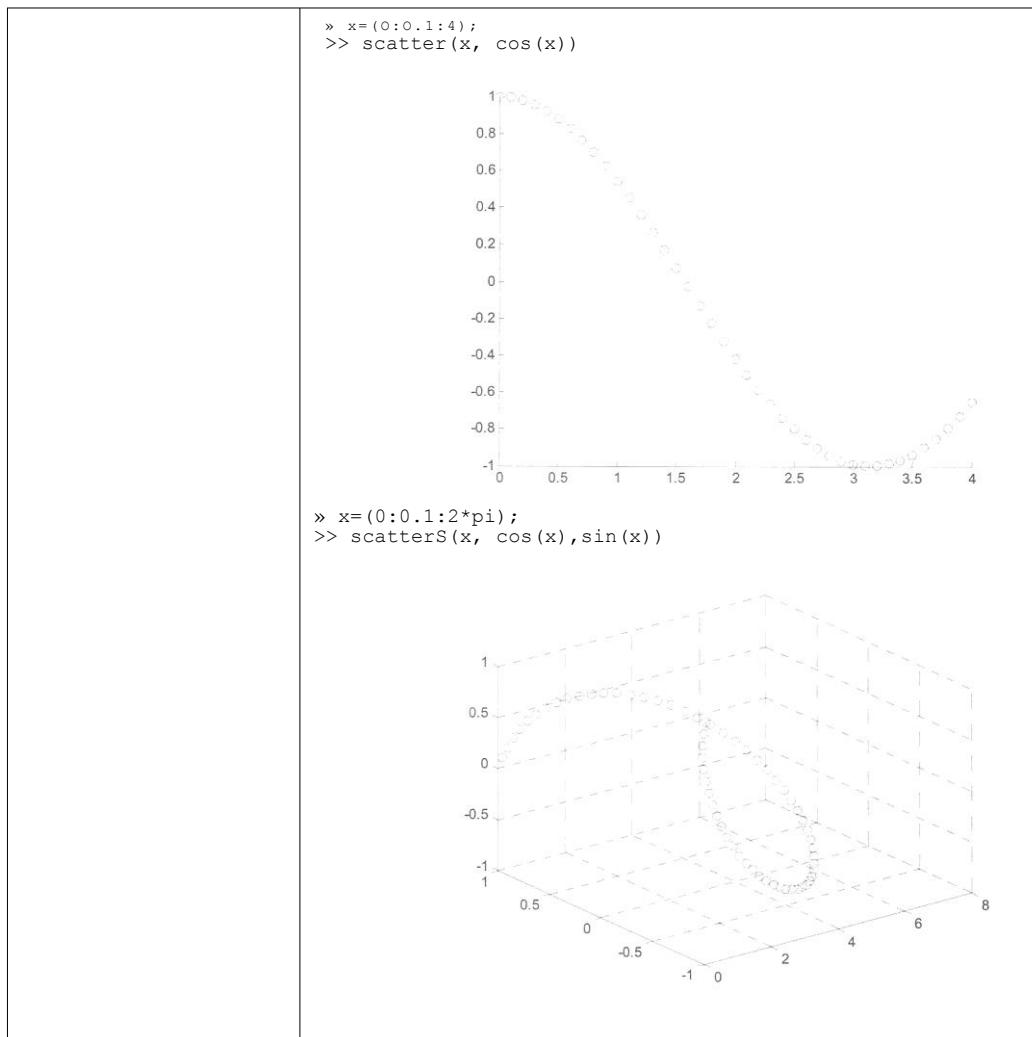
<b>ezcontour(f)</b> <b>ezcontour(f,dominio)</b> <b>ezcontour(...,n)</b>	<p>Gráfico de contorno de <math>f(x,y)</math> en <math>[-2\pi, 2\pi] \times [-2\pi, 2\pi]</math></p> <p>Gráfico de contorno de <math>f(x,y)</math> en el dominio dado</p> <p>Gráfico de contorno de <math>f(x,y)</math> en la malla <math>n \times n</math></p> <pre>&gt;&gt; ezcontour ('sqrt(x^2 + y^2)')</pre> 
<b>ezcontourf(f)</b> <b>ezcontourf(f,dominio)</b> <b>ezcontourf(...,n)</b>	<p>Gráfico de contorno de <math>f(x,y)</math> relleno en <math>[-2\pi, 2\pi] \times [-2\pi, 2\pi]</math></p> <p>Gráfico de contorno de <math>f(x,y)</math> relleno en el dominio dado</p> <p>Gráfico de contorno de <math>f(x,y)</math> relleno en la malla <math>n \times n</math></p> <pre>&gt;&gt; ezcontourf ('sqrt(x^2 + y^2)')</pre> 
<b>ezmesh(f)</b> <b>ezmesh(f,dominio)</b> <b>ezmesh(...,n) ezmesh(x,y,z)</b> <b>ezmesh(x,y,z, dominio)</b> <b>ezmesh(,,,'circ')</b>	<p>Gráfico de malla de <math>f(x,y)</math> relleno en <math>[-2\pi, 2\pi] \times [-2\pi, 2\pi]</math></p> <p>Gráfico de malla de <math>f(x,y)</math> relleno en el dominio dado</p> <p>Gráfico de malla de <math>f(x,y)</math> relleno en la malla <math>n \times n</math></p> <p>Gráfico de malla para <math>x=x(t,u)</math>, <math>y=y(t,u)</math>, <math>z=z(t,u)</math> <math>t,u \in [-2\pi, 2\pi]</math></p> <p>Gráfico de malla para <math>x=x(t,u)</math>, <math>y=y(t,u)</math>, <math>z=z(t,u)</math> <math>t,u \in</math> dominio</p> <p>Gráfico de malla sobre un disco centrado en el dominio</p> <pre>&gt;&gt; ezmesh ('sqrt(x^2 + y^2)')</pre>

	 <p><math>\sqrt{x^2 + y^2}</math></p>
<b>ezmeshc(f)</b> <b>ezmeshc(f,dominio)</b> <b>ezmeshc(...,n)</b> <b>ezmeshc(x,y,z)</b> <b>ezmeshc(x,y,z, dominio)</b> <b>ezmeshc(..., 'circ')</b>	<p>Realiza una combinación de gráfico de malla y contorno</p> <pre>&gt;&gt; ezmeshc ('sqrt (x^2 + y^2) ')</pre>  <p><math>\sqrt{x^2 + y^2}</math></p>
<b>ezsurf(f)</b> <b>ezsurf(f,dominio)</b> <b>ezsurf(...,n)</b> <b>ezsurf(x,y,z)</b> <b>ezsurf(x,y,z, dominio)</b> <b>ezsurf(..., 'circ')</b>	<p>Realiza un gráfico de superficie coloreado</p> <pre>&gt;&gt; ezsurf ('sqrt(x^2 + y^2) ')</pre>  <p><math>\sqrt{x^2 + y^2}</math></p>

<b>ezsurf(f)</b> <b>ezsurf(f,dominio)</b> <b>ezsurf(...,n)</b> <b>ezsurf(x,y,z)</b> <b>ezsurf(x,y,z, dominio)</b> <b>ezsurf(..., 'circ')</b>	<p>Realiza una combinación de gráfico de superficie y contorno</p> <pre>&gt;&gt; ezsurf('sqrt(x^2 + y^2)') sqrt(x^2 + y^2)</pre> 
<b>ezplot3(x,y,z)</b> <b>ezplot3(x,y,z,dominio)</b> <b>ezplot3(..., 'animate')</b>	<p>Curva paramétrica 3D <math>x=x(t)</math>, <math>y=y(t)</math>, <math>z=z(t)</math> <math>t \in [-\pi, \pi]</math>      Curva paramétrica 3D <math>x=x(t)</math>, <math>y=y(t)</math>, <math>z=z(t)</math> <math>t \in \text{dominio}</math>      Curva paramétrica 3D con animación</p> <pre>&gt;&gt; ezplot3('cos(t)''t.*sin(t)''sqrt(t)') x = cos(t), y = t sin(t), z = sqrt(t)</pre> 
<b>ezpolar(f)</b> <b>ezpolar(f, [a,b])</b>	<p>Grafica la curva polar <math>r=f(c)</math> con <math>c \in [0, 2n]</math>      Grafica la curva polar <math>r=f(c)</math> con <math>c \in [a, b]</math>  <code>ezpolar('sin(2*t).*cos(3*t)', [0 pi])</code></p> 

<b>pareto(Y)</b> <b>pareto(X,Y)</b>	<p>Realiza el gráfico de Pareto relativo al vector de frecuencias Y</p> <p>Realiza el gráfico de Pareto relativo al vector de frecuencias Y cuyos elementos son dados a través del vector X</p> <pre>&gt;&gt; lineas_de_codigo = [200 120 555 608 1024 101 57 687]; codificadores = ... {'Fred','Ginger','Norman','Max','Julia','Wally','Heidi','Pat'}; pareto(lineas_de_codigo, codificadores) title('Lineas de código por programador')</pre>
<b>pie3(X)</b> <b>pie3(X,explode)</b>	<p>Gráfico de sectores tridimensional para las frecuencias X</p> <p>Gráfico de sectores tridimensional desgajado</p> <pre>&gt;&gt; pie3([2 4 3 5],[0 1 1 0],{'Norte','Sur','Este','Oeste'})</pre>
<b>plotmatrix(X,Y)</b>	<p>Gráfico de dispersión de las columnas de X contra las de Y</p> <pre>&gt;&gt; x= randn (50,3)    y = x*[-1 2 1;2 0 1;1 -2 3];' ;plotmatrix(y)</pre>

	
<b>stairs(Y)</b> <b>stairs(X,Y)</b> <b>stairs(...,LineSpec)</b>	<p>Gráfico en escalera con los elementos de Y</p> <p>Gráfico en escalera de los elementos de Y correspondientes con los de X</p> <p>Especificaciones de línea para el gráfico en escalera</p> <pre>&gt;&gt; x = linspace(-2*pi,2*pi,40);stairs(x,sin(x))</pre> 
<b>scatter(X,Y,S,C)</b> <b>scatter(X,Y)</b> <b>scatter(X,Y,S)</b> <b>scatter(...,marcador)</b> <b>scatter(...,'filled')</b>	Gráfico de dispersión para los vectores (X, Y) según los colores C el área de cada marcador S. También se puede obtener el gráfico relleno (opción fill) y utilizar distintos tipos de marcadores
<b>scatter3(X,Y,Z,S,C)</b> <b>scatter3(X,Y,Z)</b> <b>scatter3(X,Y,Z,S)</b> <b>scatter3(...,marcador)</b> <b>scatter3(...,'filled')</b>	Gráfico de dispersión tridimensional para los vectores (X, Y, Z) según los colores C y el área de cada marcador S. También se puede obtener el gráfico relleno (opción fill) y utilizar distintos tipos de marcadores



## OPCIONES DE GRÁFICOS 2D Y 3D

Matlab presenta muchas posibilidades de manejo de opciones gráficas como títulos, etiquetas, colocación, punto de vista, etc. La tabla siguiente presenta las más importantes.

<b>title('texto')</b>	Añade el texto como título del gráfico en la parte superior del mismo en gráficos 2-D y 3-D
<b>xlabel('texto')</b>	Sitúa el texto al lado del eje x en gráficos 2-D y 3-D
<b>ylabel('texto')</b>	Sitúa el texto al lado del eje y en gráficos 2-D y 3-D
<b>zlabel('texto')</b>	Sitúa el texto al lado de eje z en un gráfico 3-D
<b>clabel(C,h)</b>	Rota etiquetas y las sitúa en el interior de las líneas de contorno

<b>clabel(C,h,v)</b>	<i>Crea etiquetas sólo para los niveles de contorno dados por el vector v y las rota y sitúa en el interior de las líneas de contorno</i>
<b>datetick(eje)</b>	<i>Etiqueta las marcas del eje especificado ('x', 'y' o 'z') basándose en el máximo y el mínimo del eje especificado</i>
<b>datetick(eje, fecha)</b>	<i>Etiqueta las marcas del eje especificado ('x', 'y' o 'z') con el formato de fecha dado (un entero entre 1 y 28)</i>
<b>legend('cadena1', 'cadena2',...)</b>	<i>Sitúa las leyendas especificadas por las cadenas en n gráficos consecutivos</i>
<b>legend(h, 'cadena1', 'cadena2',...)</b>	<i>Sitúa las leyendas especificadas por las cadenas en los gráficos manejados según el vector h</i>
<b>legend('off')</b>	<i>Elimina las leyendas de los ejes actuales</i>
<b>text(x,y,'texto')</b>	<i>Sitúa el texto en el punto (x,y) dentro del gráfico 2-D</i>
<b>text(x,y,z,'texto')</b>	<i>Sitúa el texto en el punto (x,y,z) en el gráfico 3-D</i>
<b>gtext('texto')</b>	<i>Permite situar el texto en un punto seleccionado con el ratón dentro de un gráfico 2-D</i>
<b>grid</b>	<i>Sitúa rejillas en los ejes de un gráfico 2-D o 3-D. La opción grid on coloca las rejillas y greed off las elimina. La opción grid permuta entre on y off</i>
<b>hold</b>	<i>Permite mantener el gráfico existente con todas sus propiedades, de modo que el siguiente gráfico que se realice se sitúe sobre los mismos ejes y se superponga al existente. La opción hold on activa la opción j hold off la elimina. La opción hold permuta entre on y off. Válido para 2-D y 3-D</i>
<b>axis([xmin xmax ymin ymax zmin zmax])</b>	<i>Sitúa los valores máximo y mínimo para los ejes X, Y y Z en el gráfico corriente</i>
<b>axis('auto')</b>	<i>Sitúa los ejes en la escala automática por defecto (la dada por xmin = min(x), xmax = max(x) e y libre)</i>
<b>axis(axis)</b>	<i>Congela el escalado de ejes en los límites corrientes, de tal forma que al situar otro gráfico sobre los mismos ejes (con hold en on) la escala no cambie</i>
<b>V=axis</b>	<i>Da el vector V de 4 elementos, conteniendo la escala del gráfico corriente</i>
<b>axis('xy')</b>	<i>Sitúa coordenadas cartesianas con el origen en la parte inferior izquierda del gráfico</i>
<b>Axis('tight')</b>	<i>Sitúa los límites de los ejes en el rango de los datos</i>
<b>axis('ij')</b>	<i>Sitúa coordenadas con el origen en la parte superior izquierda del gráfico</i>
<b>axis('square')</b>	<i>Convierte el rectángulo de graficado en un cuadrado, con lo que las figuras se abomban</i>
<b>axis('equal')</b>	<i>Sitúa el mismo factor de escala para ambos ejes</i>
<b>axis('normal')</b>	<i>Elimina las opciones square y equal</i>

<b>axis('off')</b>	<i>Elimina las etiquetas y marcas de los ejes y las rejillas, manteniendo el título del gráfico y los textos situados en él con text y gtext</i>
<b>axis('on')</b>	<i>Coloca de nuevo las etiquetas, marcas y rejillas de los ejes</i>
<b>subplot(m,n,p)</b>	<i>Divide la ventana gráfica en <math>m \times n</math> subventanas y coloca el gráfico corriente en la ventana <math>p</math>-ésima, empezando a contar por la parte superior izquierda y de izquierda a derecha hasta acabar la línea, para pasar a la siguiente</i>
<b>plotyy(X1,Y1,X2,Y2)</b>	<i>Etiqueta la gráfica de <math>X_1</math> contra <math>Y_1</math> en la izquierda, y la gráfica de <math>X_2</math> contra <math>Y_2</math> en la derecha.</i>
<b>plotyy(X1,Y1,X2,Y2, 'función')</b>	<i>Igual que el comando anterior, pero la función puede ser plot, loglog, semilogx, semilogy, ítem o cualquier <math>h=function(x,y)</math>. Igual que el comando anterior, pero usando cada función <math>f_i</math> para el par <math>(X_i Y_i)</math></i>
<b>plotyy(X1,Y1,X1,Y2, 'función1','función2')</b>	
<b>axis([xmin xmax ymin ymax zmin zmax])</b>	<i>Sitúa los intervalos de variación de los ejes en los valores indicados. También acepta las opciones 'ij', 'square', 'equal', etc, idénticas a las ya vistas para dos dimensiones</i>
<b>view([x,y,z])</b>	<i>Sitúa el punto de vista de la figura en el punto de coordenadas cartesianas <math>(x,y,z)</math></i>
<b>view([az, el])</b>	<i>Sitúa el ángulo de vista de la figura en el punto de azimut (rotación horizontal) 'az' y elevación (elevación vertical) 'el'</i>
<b>hidden</b>	<i>Controla la presencia de las líneas ocultas en el gráfico. Dichas líneas aparecen con hidden on y desaparecen con hidden off</i>
<b>shading</b>	<i>Controla el tipo de sombreado de una superficie creada con los comandos surf mesh, pcolor, fill y fill3. La opción shading flat sitúa un sombreado suave, la opción shading interp sitúa un sombreado denso y la opción shading faceted (opción por defecto) sitúa un sombreado normal</i>
<b>colormap(M)</b>	<i>Sitúa la matriz <math>M</math> como el mapa corriente de colores. <math>M</math> debe tener tres columnas y contener valores sólo entre 0 y 1. También puede ser una matriz cuyas filas sean vectores RGB del tipo [r g b]. Existen en MATLAB matrices <math>M</math> ya definidas, que son las siguientes: bone(p), contrast(p), cool(p), copper(p), flag(p), gray(p), hsv(p), hot(p), jet(p), pink(p), prism(p) y white(p). Todas las matrices tienen 3 columnas y <math>p</math> filas. Por ejemplo, la sintaxis colormap(hot(8)) sitúa la matriz hot(8) como el mapa corriente de colores (sistema completo de colores de la figura actual)</i>
<b>brighten(p)</b>	<i>Ajusta la iluminación de la figura. Si <math>0 &lt; p &lt; 1</math>, la figura será brillante, y si <math>-1 &lt; p &lt; 0</math>, la figura será oscura. La variación de <math>p</math> es el intervalo (-1,1), y a medida que los valores de <math>p</math> se acercan a -1, la figura se oscurece, mientras que a medida que los valores de <math>p</math> se acercan a 1, la figura se ilumina</i>

<b>image(A)</b>	Produce una imagen bidimensional con brillos proporcionales a los elementos de la matriz A, y se usa para mostrar fotografías y dibujos adaptados a los brillos dados en la matriz A. Cada elemento (m,n) de la matriz A afecta a una celda del dibujo
<b>pcolor(A)</b>	Produce una figura bidimensional con colores proporcionales a los elementos de la matriz A, y se utiliza para mostrar objetos matemáticos abstractos con colores variados. Cada elemento (m,n) de la matriz A afecta a una rejilla de la figura
<b>caxis([cmin cmax])</b>	Sitúa los valores mínimo y máximo de la escala de colores (definida por colormap e intrínsecamente relacionada con las divisiones que se hacen en los ejes, vía las rejillas) para un gráfico. Por lo tanto, permite utilizar sólo un subconjunto de colores del definido por colormap para la figura
<b>figure(h) o h=figure</b>	Crea la figura como un objeto de nombre h, y la sitúa como figura corriente. Se utiliza el comando gcf(h) para referir cualquier propiedad a la figura h. El comando close(h) cierra la figura h. El comando whitebg(h) cambia el color del fondo de la figura h. El comando clf cierra la figura corriente. El comando graymon sitúa la escala de grises. El comando newplot determina los ejes para hacer una nueva figura. El comando refresh redibuja la figura
<b>axes(e) o e=axes</b>	Crea los ejes como un objeto de nombre e en la figura corriente. Se utiliza el comando gca(e) para referir cualquier propiedad a los ejes e. Se utiliza el comando cla para borrar todos los objetos referentes a los ejes corrientes
<b>l=line(x,y) o l=line(x,y,z)</b>	Crea, como un objeto de nombre l, la línea que une los puntos (X,Y) en el plano o (X, Y, Z) en el espacio
<b>p=patch(X,Y,C) o patch(X,Y,Z,C)</b>	Crea un área poligonal opaca que está definida por el conjunto de puntos (X,Y) en el plano o (X,Y,Z) en el espacio, y cuyo color está dado por C, como un objeto de nombre p
<b>s=surface(X,Y,Z,C)</b>	Crea la superficie paramétrica definida por X, Y y Z y cuyo color está dado por C como un objeto de nombre s
<b>i=image(C)</b>	Crea la imagen definida por los colores dados en la matriz C como un objeto de nombre i
<b>t=text(x,y,'cadena') o t=text(x,y,z'cadena')</b>	Crea el texto definido por la cadena, localizado en el punto (x,y) del plano, o en el punto (x,y, z) del espacio
<b>set(h, 'propiedad1', 'propiedad2',...)</b>	Sitúa las propiedades especificadas en el objeto h (gca para límites de ejes, gcf para colores, gco, gcbo, gcb, etc.)
<b>get(h, 'propiedad')</b>	Devuelve el valor corriente de la propiedad dada para el objeto h
<b>object=gco</b>	Devuelve el objeto actual de la figura actual
<b>rotate(h, [a, e], a, [P,q,r])</b>	Rota el objeto h un ángulo a, según los ejes de azimut a y elevación e, siendo el origen el punto (p,q,r)
<b>reset(h)</b>	Actualiza todas las propiedades asignadas al objeto h y coloca sus propiedades por defecto
<b>delete(h)</b>	Borra el objeto h

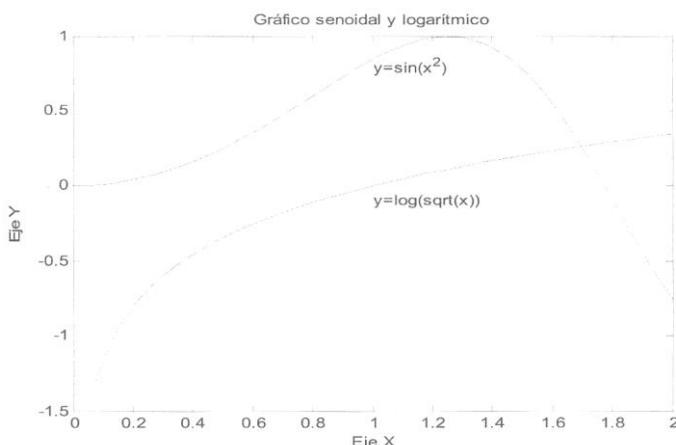
Asimismo, las propiedades más típicas a situar en los objetos gráficos de MATLAB son las siguientes:

<b>Objeto</b>	<b>Propiedades</b>	<b>Possibles valores</b>
<b>Figure</b>	<i>Color</i> (color de fondo)	'y', 'm', 'c', 'r', 'g', 'b', 'w', 'k'
	<i>ColorMap</i> (color de mapa)	<i>hot(p), gray(p), pink(p),...</i>
	<i>Position</i> (posición en pantalla)	[left, bottom, width, height]
	<i>Name</i> (nombre)	cadena con el nombre
	<i>MinColorMap</i> (n.ºmín. de color)	n.º mínimo de colores para el mapa
	<i>NextPlot</i> (modo del gráf. siguiente.)	new, add, replace
	<i>NumberTitle</i> (n.º en la figura)	on, off
	<i>Units</i> (unidades de medida)	pixels, inches, centimeters, points
	<i>Resize</i> (tamaño figura con ratón)	on (se puede cambiar), off (no)
<b>Axes</b>	<i>Box</i> (caja para el gráfico)	on, off
	<i>Color</i> (color de los ejes)	'y', 'm', 'c', 'r', 'g', 'b', 'w', 'k'
	<i>GridLineStyle</i> (línea para malla)	‘-’, ‘-.’, ‘:’, ‘.’
	<i>Position</i> (posición en pantalla)	[left, bottom, width, height]
	<i>TickLength</i> (long. entre marcas)	un valor numérico
	<i>TickDir</i> (dirección de las marcas)	in, out
	<i>Units</i> (unidades de medida)	pixels, inches, centimeters, points
	<i>View</i> (punto de vista)	[azimuth, elevación]
	<i>FontAngle</i> (ángulo de fuente)	normal, italic, oblique
	<i>FontName</i> (nombre de fuente)	texto con el nombre de la fuente
	<i>FontSize</i> (tamaño de fuente)	valor numérico
	<i>FontWeight</i> (peso de la fuente)	light, normal, demi, bold
	<i>DrawMode</i> (modo de dibujo)	normal fast (rápido)
	<i>Xcolor, Ycolor, Zcolor</i> (color ejes)	[min, max]
	<i>XDir, YDir, ZDir</i> (dirección ejes)	normal (crecen de izq. a der.), reverse
	<i>XGrid, YGrid, Zgrid</i> (rejillas)	on, off
	<i>XLabel, YLabel, Zlabel</i> (etiquetas)	cadena con el texto de las etiquetas
	<i>XLim, YLim, Zlim</i> (valores límites)	[min, max] (intervalo de variación)
	<i>XScale, YScale, ZScale</i> (escalas)	linear (lineal), log(logarítmica)
	<i>XTick, YTick, ZTick</i> (marcas)	[m1, m2, ...] (situación marcas en eje)
<b>Line</b>	<i>Color</i> (color de la línea)	'y', 'm', 'c', 'r', 'g', 'b', 'w', 'k'
	<i>LineStyle</i> (estilo de línea)	‘-’, ‘-.’, ‘:’, ‘.’, ‘+’, ‘*’, ‘.’, ‘x’
	<i>LineWidth</i> (anchura de línea)	valor numérico
	<i>Visible</i> (línea visible o no en pant.)	on, off
	<i>Xdata, Ydata, Zdata</i> (coordenad.)	conjunto de coordenadas de la línea
<b>Text</b>	<i>Color</i> (color de texto)	'y', 'm', 'c', 'r', 'g', 'b', 'w', 'k'
	<i>FontAngle</i> (ángulo de fuente)	normal, italic, oblique
	<i>FontName</i> (nombre de fuente)	texto con el nombre de la fuente
	<i>FontSize</i> (tamaño de fuente)	valor numérico
	<i>FontWeight</i> (peso de la fuente)	light, normal, demi, bold
	<i>HorizontalAlignment</i> (ajuste hor.)	left, center, right
	<i>VerticalAlignment</i> (ajuste vert.)	top, cap, middle, baseline, bottom

	<i>Position</i> (posición en pantalla)	$[x, y, z]$ (punto de situación)
	<i>Rotation</i> (orientación del texto)	$0, \pm 90, \pm 180, \pm 270$
	<i>Units</i> (unidades de medida)	<i>pixels, inches, centimeters, points</i>
	<i>String</i> (cadena de texto a situar)	cadena con el texto
<b>Surface</b>	<i>Cdata</i> (color de cada punto)	matriz de colores
	<i>Edgecolor</i> (color de re/lillas)	'y', 'm', ..., 'none', flat, interp
	<i>Facecolor</i> (color de las caras)	'y', 'm', ..., 'none', flat, interp
	<i>LineStyle</i> (estilo de línea)	'-', '--', ':', '-.', '+', '*', '.', 'x'
	<i>LineWidth</i> (anchura de línea)	valor numérico
	<i>MeshStyle</i> (líneas en filas y col.)	row, colum, both
	<i>Visible</i> (línea visible o no en pant.)	on, off
	<i>Xdata, Ydata, Zdata</i> (coordenad.)	conjunto de coordenadas de la superficie
<b>Patch</b>	<i>Cdata</i> (color de cada punto)	matriz de colores
	<i>Edgecolor</i> (color de los ejes)	'y', 'm', ..., 'none', flat, interp
	<i>Facecolor</i> (color de las caras)	'y', 'm', ..., 'none', flat, interp
	<i>LineWidth</i> (anchura de línea)	valor numérico
	<i>Visible</i> (línea visible o no en pant.)	on, off
	<i>Xdata, Ydata, Zdata</i> (coordenad.)	conjunto de coordenadas de la superficie
<b>Image</b>	<i>Cdata</i> (color de cada punto)	matriz de colores
	<i>Xdata, Ydata</i> (coordenadas)	conjunto de coordenadas de la imagen

Veamos algunos ejemplos ilustrativos.

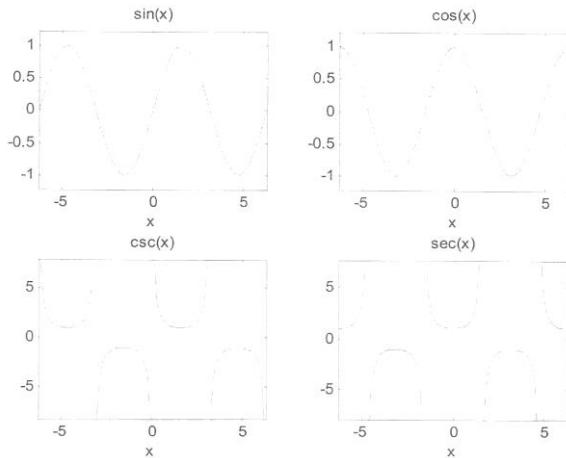
```
>> x=linspace(0,2,30);
y=sin(x.^2);
plot(x,y)
text(1,0.8, 'y=sin(x^2)')
hold on
z=log(sqrt(x));
plot(x,z)
text(1,-0.1, 'y=log(sqrt(x))')
xlabel('Eje X');
ylabel('Eje Y');
title('Gráfico senoidal y logarítmico');
```



```

» subplot (2,2,1);
ezplot('sin(x)',[-2*pi 2*pi])
subplot (2,2,2);
ezplot('cos(x)[-2*pi 2*pi]')
subplot (2,2,3);
ezplot ('csc(x)', [-2*pi 2*pi])
subplot (2,2,4);
ezplot('sec(x)', [-2*pi 2*pi])

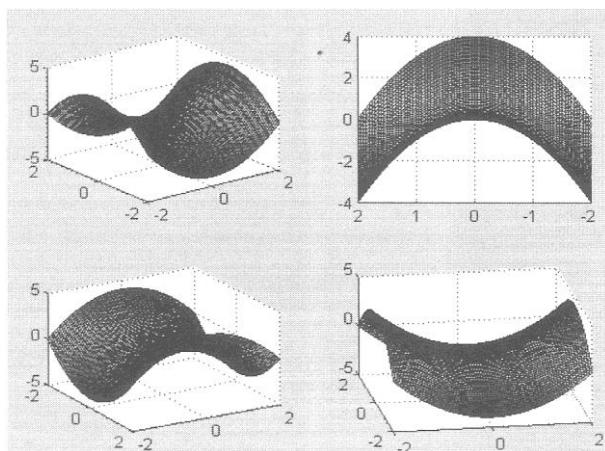
```

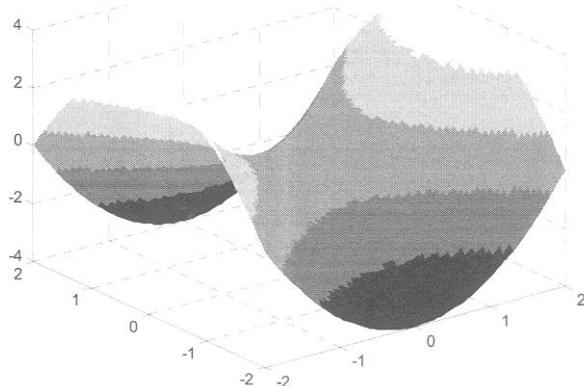


```

» [X,Y]=meshgrid(-2:0.05:2)
Z=X.^2-Y.^2;
subplot(2,2,1)
surf(X,Y,Z)
subplot(2,2,2)
surf(X,Y,Z), view(-90,0)
subplot(2,2,3)
surf(X,Y,Z), view(60,30)
subplot(2,2,4)
surf(X,Y,Z), view(-10,30)

```

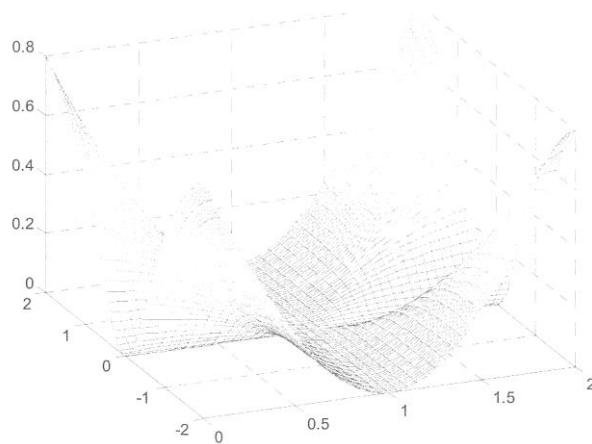




*Ejercicio 1.* Representar la superficie definida por la ecuación:

$$f(x, y) = \frac{(x-1)^2 y^2}{(x-1)^2 + y^2}$$

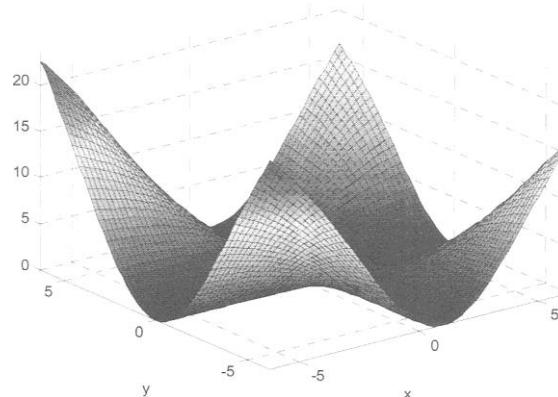
```
» [x, y]=meshgrid(0:0.05:2, -2:0.05:2);
» z=y.^2.* (x-1).^2./ (y.^2+(x-1).^2);
» mesh(x, y, z), view([-23, 30])
```



También podíamos haber representado la superficie de la forma siguiente;

```
>> ezsurf('y^2*(x-1)^2/(y^2+(x-1)^2)')
```

$$y^2(x-1)^2/(y^2+(x-1)^2)$$

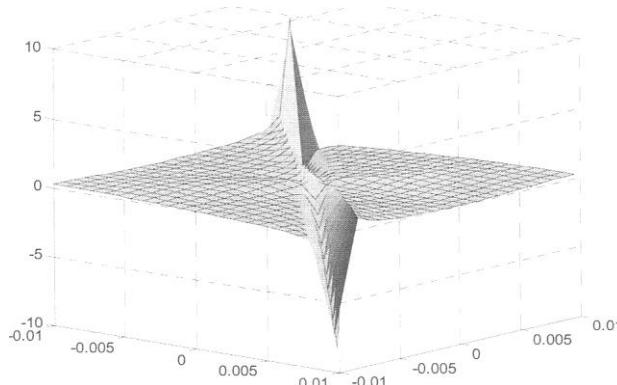


*Ejercicio 2. Sea  $f: \mathbb{R}^2 \rightarrow \mathbb{R}$  una función definida por:*

$$f(x, y) = \frac{[1-\cos(x)]\sin(y)}{x^3+y^3}$$

*Representarla gráficamente en un entorno de  $(0,0)$ .*

```
>> [x, y] = meshgrid(1/100:0.0009:1/100, -1/100:0.0009:1/100)
>> z=(1-cos(x)).*sin(y)./(x.^3+y.^3);
>> surf(x, y, z)
>> view([50, -15])
```

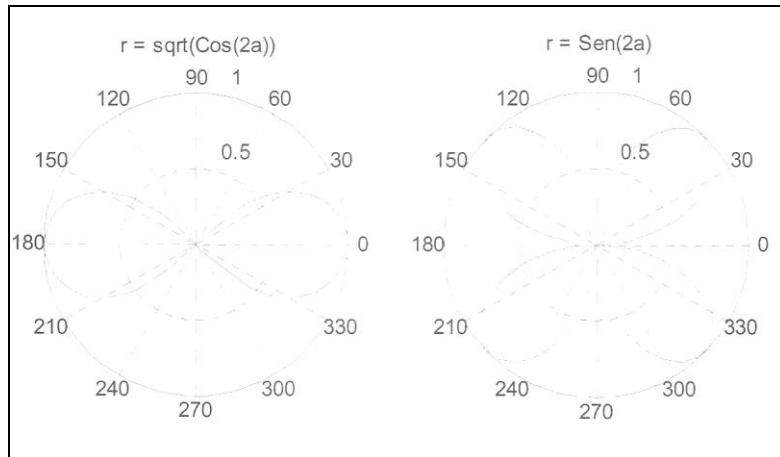


*Ejercicio 3. Representar sobre el mismo gráfico y de forma adyacente las curvas en polares:*

$$r = \sqrt{\cos(2a)} \text{ y } r = \sin(2a)$$

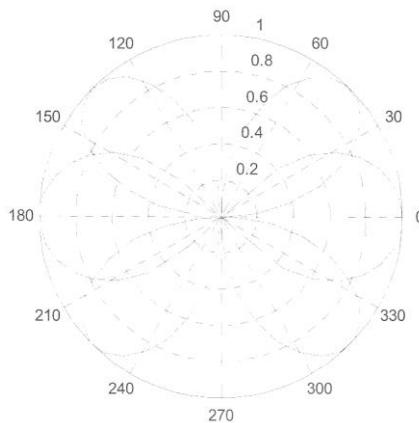
*Representar también su intersección.*

```
>> a=0:.1:2*pi;
>> subplot(1,2,1)
>> r=sqrt(cos(2*a));
>> polar(a,r)
>> title(' r = sqrt(Cos(2a)) ')
>> subplot(1,2,2)
>> r=sin(2*a);
>> polar(a,r)
>> title(' r = Sen(2a) ')
```



*Para ver la intersección de las dos curvas, las representamos ambas sobre los mismos ejes.*

```
>> a=0:.1:2*pi;
>> r=sqrt(cos(2*a));
>> polar(a,r)
>> hold on;
>> r=sin(2*a);
>> polar(a,r)
```



*Ejercicio 4. Representar el volumen engendrado al girar la cónica  $12x - 9x^2 + 2x^3$  alrededor del eje OX, comprendida entre  $x=0$  y  $x=5/2$*

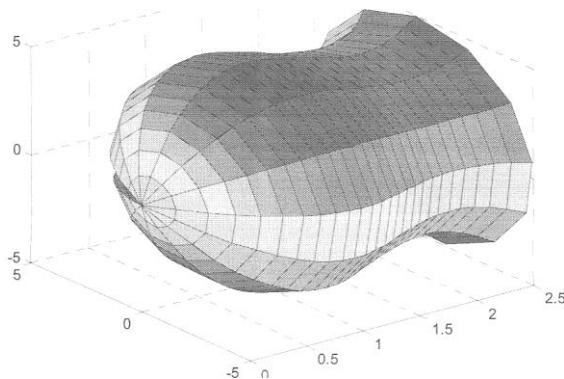
La superficie de revolución tendrá de ecuación  $y^2 + z^2 = (12x - 9x^2 + 2x^3)^2$ , y para graficarla puede parametrizarse como sigue:

$$x = t, \quad y = \cos(u) * (12t - 9t^2 + 2t^3), \quad z = \sin(u) * (12t - 9t^2 + 2t^3)$$

```

>> t=(0:.1:5/2);
>> u=(0:.5:2*pi);
>> x=ones(size(u))'*t;
>> y=cos(u)'*(12*t-9*t.^2+2*t.^3);
>> z=sin(u)'*(12*t-9*t.^2+2*t.^3);
>> surf(x,y,z)

```



*Ejercicio 5. Representar los volúmenes engendrados al girar la elipse:  $\frac{x^2}{4} + \frac{y^2}{9} = 1$*

*alrededor del eje OX y alrededor del eje OY.*

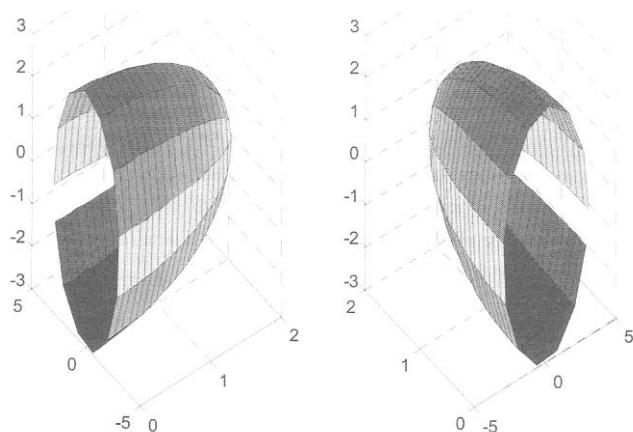
Representamos sobre el mismo gráfico las figuras generadas, pero solamente en su mitad positiva. El volumen final será el doble del representado. La ecuación de la figura de revolución alrededor del eje OX es  $y^2+z^2=9(1-x/4)$ , y se parametriza así:

$$x=t, y = 3 \operatorname{Cos}(u) (1-t^2/4)^{1/2} z = 3 \operatorname{Sen}(u) (1-t^2/4)^{1/2}$$

La ecuación de la figura de revolución alrededor del eje OY se escribe como  $x^2+z^2=4(1-y^2/4)$ , y se parametriza así:

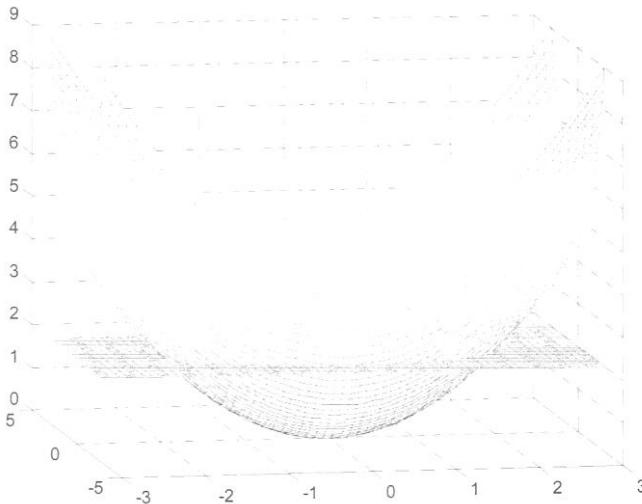
$$x = 3 \operatorname{Cos}(u) (1-t^2/9)^{1/2} y=t, z = 3 \operatorname{Sen}(u) (1-t^2/9)^{1/2}$$

```
» t=(0:.1:2);
» u=(0:.5:2*pi);
» x=ones(size(u))'*t;
» y=cos(u)'*3*(1-t.^2/4).^(1/2);
» z=sin(u)'*3*(1-t.^2/4).^(1/2);
» subplot(1,2,1)
» surf(x,y,z)
» subplot(1,2,2)
» x=cos(u)'*3*(1-t.^2/4).^(1/2);
» y=ones(size(u))'*t;
» z=sin(u)'*3*(1-t.^2/4).^(1/2);
» surf(x,y,z)
```



*Ejercicio 6. Representar la intersección del parabolóide  $x^2 + y^2 = 2z$  con el plano  $z=2$ .*

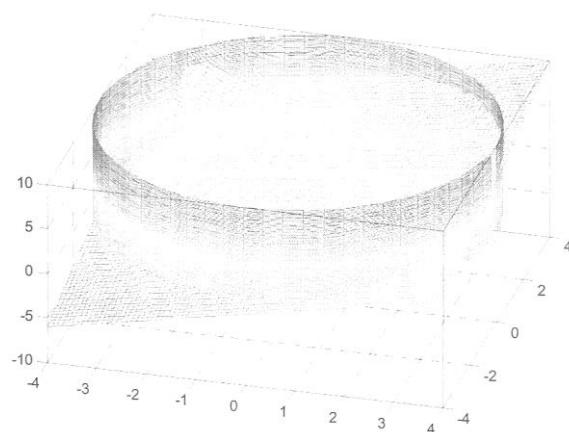
```
>> [x,y]=meshgrid(-3:.1:3);
>> z=(1/2)*(x.^2+y.^2);
>> mesh(x,y,z)
>> hold on;
>> z=2*ones(size(z));
>> mesh(x,y,z)
>> view(-10,10)
```



*Ejercicio 7. Representar el volumen en el primer octante comprendido entre el plano OXY, el plano  $z=x+y+2$  y el cilindro  $x^2 + y^2 = 16$ .*

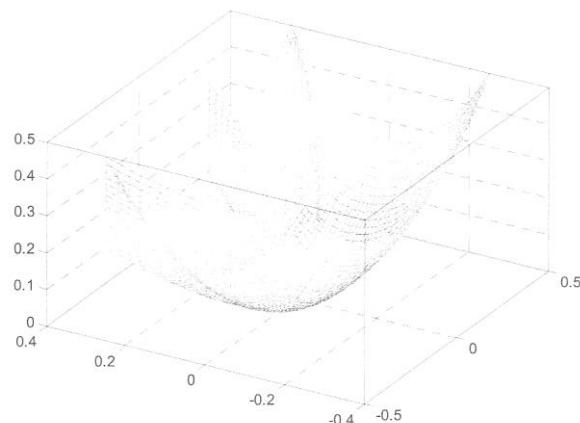
Hacemos una representación gráfica del recinto, con el plano en cartesianas y el cilindro parametrizado.

```
>> t=(0:.1:2*pi);
>> u=(0:.1:10);
>> x=4*cos(t)'.*ones(size(u));
>> y=4*sin(t)'.*ones(size(u));
>> z=ones(size(t))'*u;
>> mesh(x,y,z)
>> hold on;
>> [x,y]=meshgrid(-4:.1:4);
>> z=x+y+2;
>> mesh(x,y,z)
>> set(gca,'Box','on');
>> view(15,45)
```



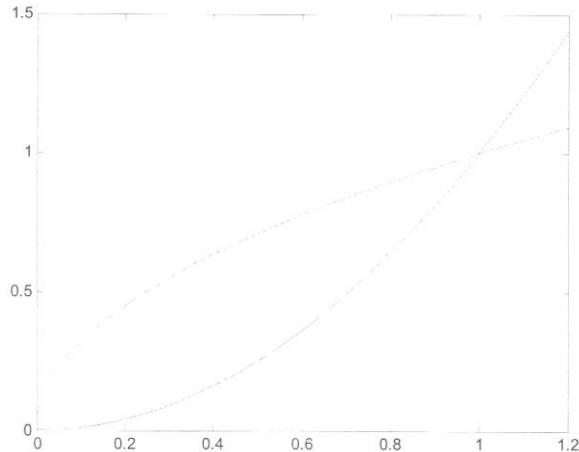
*Ejercicio 8. Representar el volumen limitado por el parabololoide  $x^2 + 4 = z$ , y lateralmente por los cilindros de ecuaciones  $y^2 = x$  y  $x^2 = y$ .*

```
» [x,y]=meshgrid(-1/2:.02:1/2,-1/4:.01:1/4);
» z=x.^2 + 4*y.^2;
» mesh(x,y,z)
» hold on;
» y=x.^2;
» mesh (x,y,z)
» hold on;
» x=y.^2;
» mesh (x,y,z)
» set (gca, 'Box', 'on')
» view(-60,40)
```

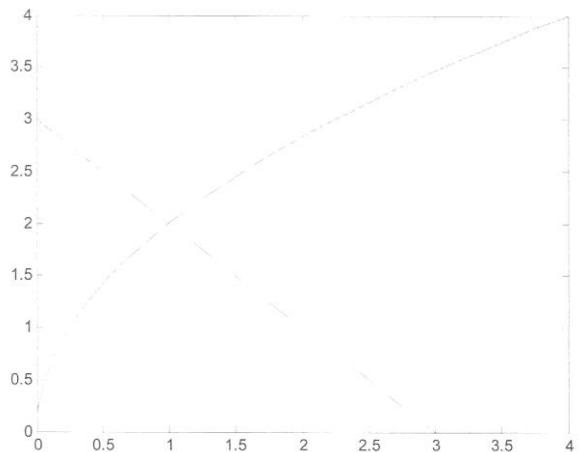


*Ejercicio 9. Representar sobre los mismos ejes las paráboles  $y^2 = x$  y  $x^2 = y$ .  
 Representar también sobre los mismos ejes la parábola  $y^2 = 4x$  y la recta  $x+y=3$ .*

```
» fpplot([x^2,sqrt(x)] [0,1.2])
```



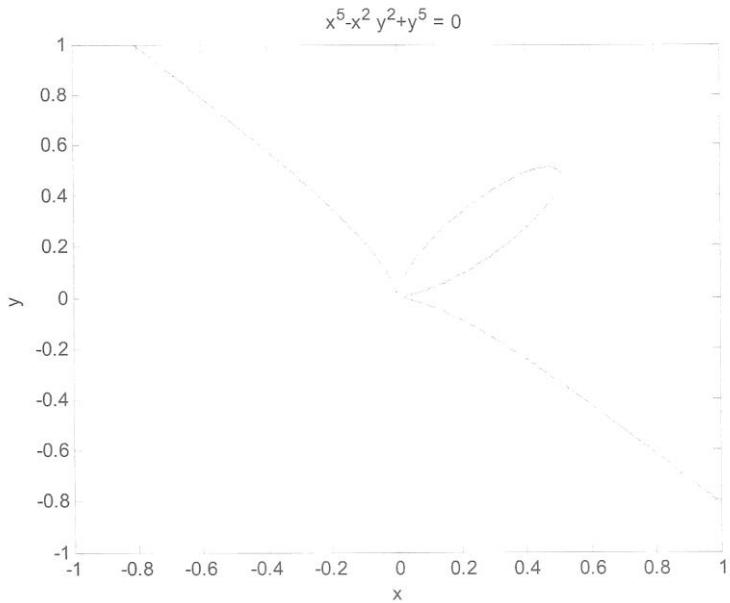
```
» fpplot([(4*x)^(1/2),3-x]^!, [0,4,0,4])
```



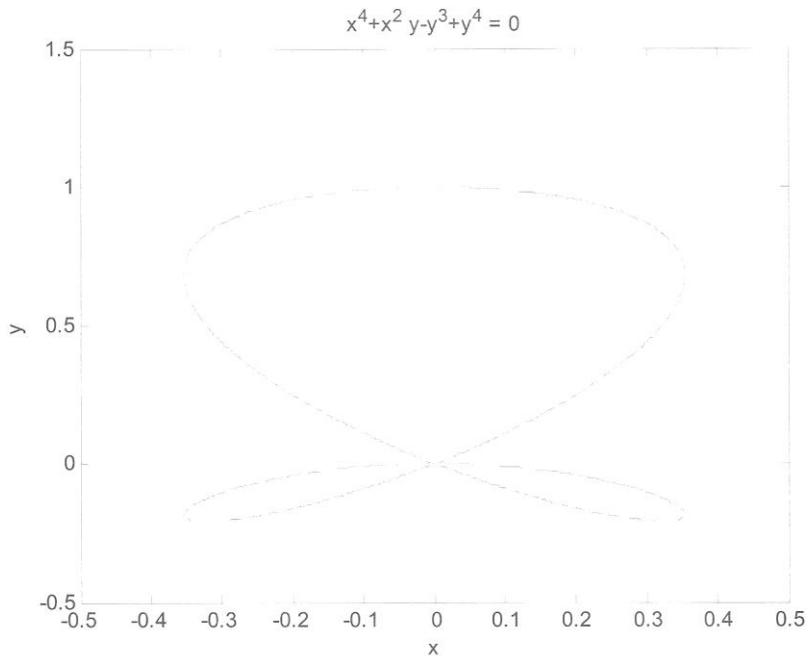
*Ejercicio 10. Representar la curvar en coordenadas implícitas de ecuaciones:*

$$\begin{aligned} x^5 - x^2y^2 + y^5 &= 0 \\ x^4 + x^2y - y^5 + y^4 &= 0 \end{aligned}$$

```
» ezplot ('x^5-x^2*y^2+y^5', [-1,1,-1,1])
```



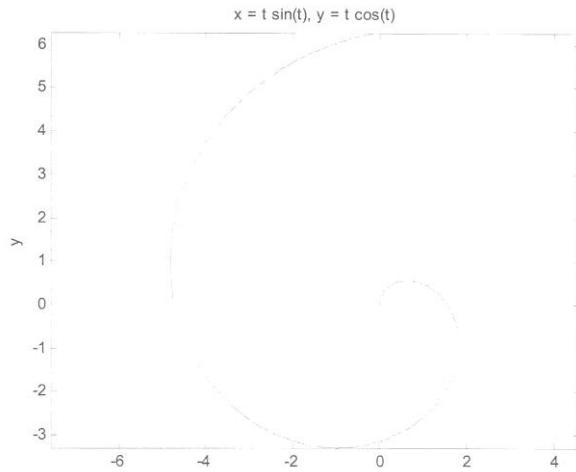
```
» ezplot (x^4+x^2*y-y^3+y^4', [-1/2,1/2,-1/2,3/2])
```



*Ejercicio 11. Representar la curva en coordenadas paramétricas:*

$$\begin{aligned}x(t) &= t \operatorname{Sen}(t) \\y(t) &= t \operatorname{Cos}(t)\end{aligned}$$

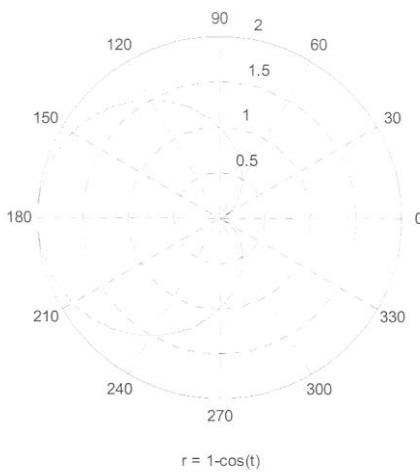
```
» ezplot('t*sin(t) t*cos(t)')
```



*Ejercicio 12. Representar la curva en coordenadas polares:*

$$\rho = 1 - \cos(\theta)$$

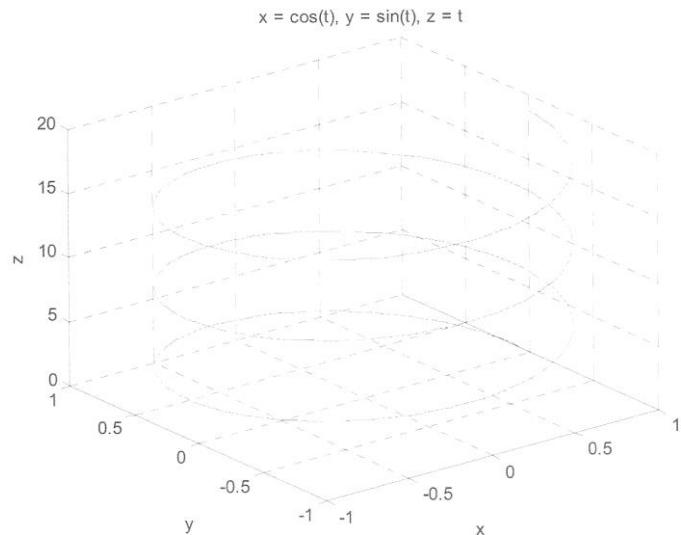
```
» ezpolar('1-cos(t)')
```



Ejercicio 13. Representar la curva alabeada de ecuaciones paramétricas:

$$x = \cos(t), y = \sin(t), z = t$$

```
>> ezplot3('cos(t)', 'sin(t)', 't', [0, 6*pi])
```



## CAPITULO 5

# EXPRESIONES ALGEBRAICAS, POLINOMIOS, ECUACIONES Y SISTEMAS

---

### EXPANSION, SIMPLIFICACION Y FACTORIZACION DE EXPRESIONES ALGEBRÁICAS

MATLAB incorpora una amplia gama de comandos que permiten trabajar con expresiones algebraicas. Es posible simplificar al máximo una expresión, expandirla o factorizarla adecuadamente. La tabla siguiente presenta los comandos más habituales para el trabajo con expresiones algebraicas.

<b>expand(expr)</b>	<p><i>Expande lo más posible una expresión algebraica, realizando totalmente los productos y potencias, hasta presentar el resultado como una suma de términos. Aplica reglas de ángulos múltiples para expresiones trigonométricas y aplica formalmente las propiedades de las funciones logarítmicas y exponenciales. También descompone fracciones algebraicas de numerador polinómico en sumas de fracciones</i></p> <pre>&gt;&gt; syms x y z t a b &gt;&gt; pretty(expand((x+2)^(x+3)))            2         x + 5 x + 6  &gt;&gt; pretty(expand((x+2)/(x+3)))            x         2            ----- + -----            x + 3     x + 3 &gt;&gt; pretty(expand(cos(x+y))) cos(x)cos(y) - sin(x) sin(y)</pre>
---------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>factor(expr)</b>	Escribe una expresión algebraica expandida como producto de factores (inversa de expand). La factorización se realiza por defecto en el cuerpo definido por los coeficientes de la expresión. Para fracciones algebraicas, factoriza numerador y denominador y simplifica factores comunes  <pre>&gt;&gt; syms x y &gt;&gt; pretty(factor(6*x^2+18*x-24)) 6 (x + 4) (x - 1) &gt;&gt; pretty (factor ( (x^3-y^3) / (x^4-y^4) ))       2      2 x  + x y + y -----       2      2 (x + y) (x + y ) &gt;&gt; pretty (factor (x^3+y^3))       2      2 (x + y) (x - x y + y )</pre>
<b>simplify(expr)</b>	Simplifica lo más posible una expresión algebraica realizando sumas finales y también ejecuta suma de fracciones algebraicas  <pre>&gt;&gt; syms x y a b c &gt;&gt; simplify (sin (x)^2 + cos (x)^2)*2 ans = 2 &gt;&gt; simplify(log(exp(a+log(exp(c)))))  ans = log(exp (a + c))</pre>
<b>simple(expr)</b>	Simplifica al máximo la expresión algebraica mostrando las reglas que se utilizan  <pre>&gt;&gt; syms a positive; f= (1/a^3 + 6/a^2 + 12/a + 8)^(1/3); &gt;&gt; simplify(f) ans = (8*a^3 + 12*a^2 + 6*a + 1)^(1/3)/a &gt;&gt; simple(f) simplify: (2*a + 1)/a</pre>

```
radsimp:  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
simplify(100):  
  
1/a + 2  
  
combine(sincos):  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
combine (sinhcosh):  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
combine (ln):  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
factor:  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
expand:  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
combine:  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
rewrite(exp);  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
rewrite(sincos):  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
rewrite(sinhcosh):  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)  
  
rewrite(tan):  
  
(12/a + 6/a^2 + 1/a^3 + 8)^(1/3)
```

	<pre> collect(a): (12/a + 6/a^2 + 1/a^3 + 8)^(1/3) ans = 1/a + 2 &gt;&gt; g=simple(f) g = 1/a + 2 </pre>
<b>collect(expr)</b>	<p>Presenta el resultado de la expresión agrupando términos en potencias de sus variables</p> <pre> &gt;&gt; syms x; f = x*(x*(x - 6) + 11) - 6; &gt;&gt; collect(f) ans = x^3 - 6*x^2 + 11*x - 6 &gt;&gt; f = (1+x)*t + x*t; &gt;&gt; collect(f) ans = (2*t)*x + t </pre>
<b>horner(expr)</b>	<p>Factoriza la expresión en su forma de Horner</p> <pre> &gt;&gt; syms x; f = x^3 - 6*x^2 + 11*x - 6; &gt;&gt; horner (f) ans = x*(x*(x - 6) + 11) - 6 </pre>

## POLINOMIOS

MATLAB implementa comandos específicos para el trabajo con polinomios en tareas como la búsqueda de sus raíces, evaluación, diferenciación e interpolación. La tabla siguiente presenta la sintaxis y ejemplos de los comandos más importantes que involucran polinomios.

<b>poly2sym(vector)</b>	<p>Convierte el vector en un polinomio simbólico cuyos coeficientes son las componentes del vector (de mayor a menor potencia)</p> <pre> &gt;&gt; poly2sym([3 5 0 8 9]) ans = 3*x^4 + 5*x^3 + 8*x + 9 </pre>
-------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>poly2sym(vector,'v')</b>	<p><i>Convierte el vector en un polinomio simbólico en la variable v cuyos coeficientes son las componentes del vector</i></p> <pre>&gt;&gt; poly2sym([3 5 0 8 9], 'z') ans = 3*z^4 + 5*z^3 + 8*z + 9</pre>
<b>sym2poli(polinomio)</b>	<p><i>Convierte el polinomio en un vector cuyas componentes son sus coeficientes</i></p> <pre>&gt;&gt; syms x &gt;&gt; sym2poly (x^5-3*x^4 + 2*x^2-7*x+12) ans = 1 -3 0 2 -7 12</pre>
<b>q=conv(u,v)</b>	<p><i>Da los coeficientes del polinomio producto de los dos polinomios cuyos coeficientes vienen dados por los vectores u y v</i></p> <pre>&gt;&gt; u=[3 -1 4 2];v=[2 14 6 8 3]; &gt;&gt; p=conv(u,v) P = 6 1 19 22 36 33 41 28 6 &gt;&gt; poly2sym(p) ans = 6*x^8 + x^7 + 19*x^6 + 22*x^5 + 36*x^4 + 33*x^3 + 41*x^2 + 28*x + 6</pre>
<b>[q,r] = deconv(v,u)</b>	<p><i>Da los polinomios cociente y resto de la división entre los polinomios u y v. Se tiene v=conv(u,q)+r</i></p> <pre>&gt;&gt; [q,r]=deconv(v,u) q = 0.6667 0.5556 0.6296 r = 0 0 0 3.0741 4.3704 1.7407 &gt;&gt; poly2sym(q) ans = (2*x^2)/3 + (5*x)/9 + 17/27 &gt;&gt; poly2sym(r) ans = (83*x^2)/27 + (118*x)/27 + 47/27</pre>

<b>p = poly(r)</b>	<p><i>Da los coeficientes del polinomio p cuyas raíces son el vector r</i></p> <pre>&gt;&gt; p=poly(u)  p =      1     -8     17      2    -24</pre> <p><i>&gt;&gt; poly2sym(p)</i></p> <pre>ans =  x^4 - 8*x^3 + 17*x^2 + 2*x - 24</pre>
<b>k = polyder(p)</b> <b>k = polyder(a,b)</b> <b>[q,d] = polyder(b,a)</b>	<p><i>Da los coeficientes del polinomio k derivada del polinomio p</i></p> <p><i>Da los coeficientes de la derivada del producto de a y b</i></p> <p><i>Da el numerador q y el denominador d de la derivada de a/b</i></p> <pre>&gt;&gt; polyder([1 -8 17 2 -24])  ans      4   -24   34   2  &gt;&gt; poly2sym([1 -8 17 2 -24])  ans =  x^4 - 8*x^3 + 17*x^2 + 2*x - 24  &gt;&gt; poly2sym(polyder([1 -8 17 2 -24]))  ans =  4*x^3 - 24*x^2 + 34*x + 2  &gt;&gt; u=[3 -1 4 2];v=[2 14 6 8 3];  &gt;&gt; k=polyder(u,v)  k =      48    7   114   110   144   99   82   28  &gt;&gt; poly2sym(k)  ans =  48*x^7 + 7*x^6 + 114*x^5 + 110*x^4 + 144*x^3 + 99*x^2 + 82*x + 28  &gt;&gt; [q,d]=polyder(u,v)  q =      -12        3       -30      -10        8       -29      -30      -4  d =      4        4       17      32      60      76     106     120     100      48         9</pre>

<p><b>p = polyfit(x,y,n)</b></p> <p><b>[p,S] = polyfit(x,y,n)</b></p> <p><b>[p,S,u] =</b></p> <p><b>polyfit(x,y,n)</b></p>	<p><i>Polinomio de grado n que ajusta la nube de puntos (x,y)</i></p> <p><i>Polinomio de grado n que ajusta los puntos (x,y) con varianza S</i></p> <p><i>Polinomio de grado n que ajusta los puntos (x,y) con varianza S y media u para el error del ajuste</i></p> <pre>&gt;&gt; u=[3 -1 4 2]; v=[2 1 4 6]; &gt;&gt; p=poly2sym(polyfit(u,v,3))  p = </pre> $(53*x^3)/60 - (99*x^2)/20 + (119*x)/30 + 54/5$ <pre>&gt;&gt; [p,S,u]=polyfit(u,V,3) p = </pre> $8.9050 \quad 1.6333 \quad -11.3053 \quad 6.0000$ <pre>s = </pre> <p>R: [4x4 double] df: 0 normr: 1.2686e-014</p> <pre>u = </pre> $\begin{matrix} 2.0000 \\ 2.1602 \end{matrix}$
<p><b>y = polyval(p,x)</b></p> <p><b>y = polyval(p,x,[ ],u)</b></p> <p><b>[y,delta] =</b></p> <p><b>polyval(p,x,S)</b></p> <p><b>[y,delta] =</b></p> <p><b>polyval(p,x,S,u)</b></p>	<p><i>Evalúa el polinomio p en x</i></p> <p><i>Evalúa el polinomio p en x con media u para el error</i></p> <p><i>Evalúa el polinomio p en x con varianza S para el error y con un error y±delta en el resultado</i></p> <p><i>Evalúa el polinomio p en x con media u y varianza S para el error y con un error y± delta en el resultado</i></p> <pre>&gt;&gt; p=[2 0 -1 7 9] p = </pre> $2 \quad 0 \quad -1 \quad 7 \quad 9$ <pre>&gt;&gt; poly2sym(p)  ans = </pre> $2*x^4 - x^2 + 7*x + 9$ <pre>&gt;&gt; polyval(p,10)  ans = </pre> $19979$
<p><b>Y = polyvalm(p,X)</b></p>	<p><i>Evalúa el polinomio p en la variable matricial X</i></p> <pre>&gt;&gt; x=[1 2 3; 4 5 6; 7 8 9]</pre>

	<pre> x = 1      2      3 4      5      6 7      8      9  &gt;&gt; p=[2    0    -1    7    9] p = 1      0      -1      7      9  &gt;&gt; A=polyval(p,x)  A = 17      51      183 533     1269     2607 4811     8193     13113 </pre>
<b>[r,p,k] = residue(b,a)</b> <b>[b,a] = residue(r,p,k)</b>	<p>Da residuos, polos y términos directos de la expansión racional de b/a</p> $\frac{b(s)}{a(s)} = \frac{r_1}{s-p_1} + \frac{r_2}{s-p_2} + \dots + \frac{r_n}{s-p_n} + k(s)$ <p>Convierte la expansión racional a polinomios de coeficientes b y a</p> <pre> &gt;&gt; u=[3 -1 4 2];v=[2 1 4 6 8 3]; &gt;&gt; [r, p, k]=residue(v,u)  r = 0.4751 - 0.6032i 0.4751 + 0.6032i 0.0745  p = 0.3705 + 1.2240i 0.3705 - 1.2240i -0.4076  k = 0.6667    0.5556    0.6296  &gt;&gt; [v, u] =residue (r,p,k) v = 0.6667    0.3333    1.3333    2.0000    2.6667    1.0000 </pre>

	$u =$  $1.0000 \quad -0.3333 \quad 1.3333 \quad 0.6667$
<b>r = roots(c)</b>	<p>Da el vector columna r de raíces del polinomio con coeficientes c</p> <pre>&gt;&gt; v=[ 0.6667    0.3333    1.3333    2.0000    2.6667 1.0000]; &gt;&gt; r=roots(v)</pre> <p>r =</p> <pre>0.6662 + 1.4813i 0.6662 - 1.4813i -0.6662 + 0.8326i -0.6662 - 0.8326i -0.5000</pre>

## INTERPOLACION POLINOMICA

MATLAB implementa comandos específicos, tanto algebraicos como gráficos, para el trabajo con interpolación polinómica. La tabla siguiente presenta la sintaxis y ejemplos de los comandos más importantes para la interpolación polinómica.

<b>Yi = interp1(X,Y,Xi)</b>	Da el vector Yi tal que (Xi, Yi) es el conjunto total de puntos hallados por interpolación unidimensional del conjunto de puntos dado (X, Y)
<b>Yi = interp1(Y,Xi)</b>	Supone que X=1:N, siendo N la longitud de Y
<b>Yi = interp1(X,Y,Xi,método)</b>	Realiza la interpolación mediante el método dado, que puede ser nearest (vecino más cercano), linear (lineal), cubic (cúbica de Hermite), vScubic (cúbica de MATLAB 5), spliney pchip (cúbica de Hermite)
<b>Yi = interp1(X,Y,Xi,método ,ext)</b>	Adicionalmente realiza extrapolación para el valor de x=ext externo al rango de variación de x usado para la interpolación

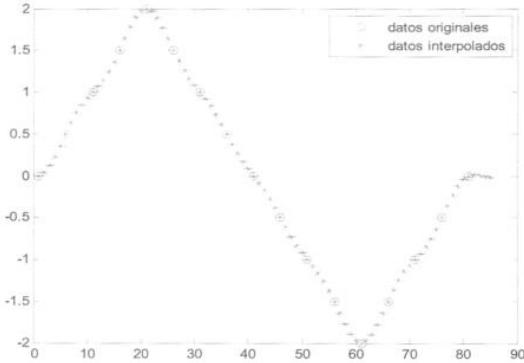
En el ejemplo siguiente se interpolan 20 puntos (x,y) según la función  $y=\sin(x)$  para valores de x igualmente espaciados entre 0 y 10

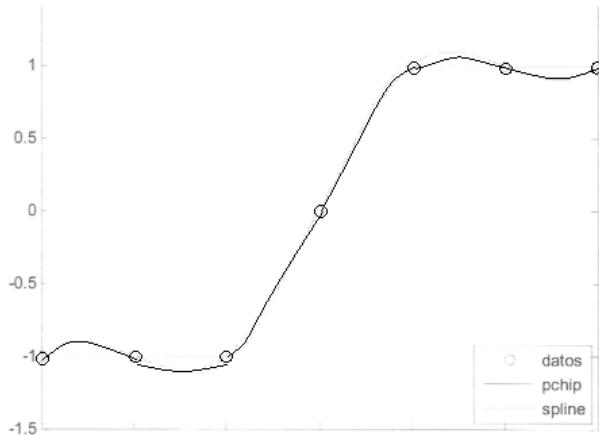
```
>> x = 0:10; y = sin(x); xi = 0;.5:10;yi =
interp1(x,y,xi); puntos=[xi',yi']
```

puntos =	
0	0
0.5000	0.4207
1.0000	0.8415
1.5000	0.8754
2.0000	0.9093

	<pre> 2.5000    0.5252 3.0000    0.1411 3.5000   -0.3078 4.0000   -0.7568 4.5000   -0.8579 5.0000   -0.9589 5.5000   -0.6192 6.0000   -0.2794 6.5000    0.1888 7.0000    0.6570 7.5000    0.8232 8.0000    0.9894 8.5000    0.7007 9.0000    0.4121 9.5000   -0.0660 10.0000   -0.5440 </pre> <p><i>Podemos representar los puntos de la siguiente forma:</i></p> <pre>plot(x,y, 'o', xi,yi)</pre>
<b>Zi =</b> <b>interp2(X,Y,Z,Xi,Yi)</b>  <b>Zi = interp2(Z,Xi,Yi)</b>  <b>Zi = interp2(Z,n)</b>  <b>Zi =</b> <b>interp2(X,Y,Z,Xi,Yi,</b> <b>method)</b>	<p><i>Da el vector Zi tal que (Xi, Yi, Zi) es el conjunto total de puntos hallados por interpolación bidimensional del conjunto de puntos dados (X,Y,Z)</i></p> <p><i>Supone que X=1:n e Y=1:m y (n,m) = tamaño de Z</i></p> <p><i>Interpolación recursiva bidimensional n veces</i></p> <p><i>Interpolación mediante los métodos nearest (vecino más cercano), linear (lineal), cubic (cúbica de Hermite) y spline</i></p> <p>En el ejemplo siguiente consideraremos un conjunto de años, tiempos de servicios y salarios e intentamos encontrar por interpolación el salario ganado en 1975 por un empleado con 15 años de servicios</p> <pre>&gt;&gt; anos = 1950:10:1990;</pre>

	<pre> servicios = 10:10:30; salarios = [150.697 199.592 187.625 179.323 195.072 250.287 203.212 179.092 322.767 226.505 153.706 426.730 249.633 120.281 598.243]; w = interp2 (servicios,anos,salarios,15,1975) w = 190.6288 </pre>
<b>Vi = interp3(X,Y,Z,V,Xi, Yi,Zi)</b>	<p><i>Da el vector Vi tal que <math>(Xi, Yi, Zi)</math> es el conjunto total de puntos hallados por interpolación tridimensional del conjunto de puntos resultante de aplicar la función tridimensional V a los puntos <math>(X, Y, Z)</math></i></p>
<b>Vi = interp3(V,Xi,Yi,Zi)</b>	<p><i>Supone que <math>X=1:n</math>, <math>Y=1:m</math>, <math>Z=1:p</math> (<math>n,m,p</math>) = tamaño de V</i></p>
<b>Vi = interp3(V,n) Vi = interp3(...,método)</b>	<p><i>Interpolación recursiva tridimensional n veces Interpolación mediante los métodos especificados</i></p> <p>En el ejemplo siguiente se calcula y representa la matriz de puntos volumétrica <math>w=f(x,y,z)</math> de interpolación tridimensional entre los puntos <math>(x,y,z,t)</math> dados por los valores que toma la función predefinida en MATLAB `flow` para valores de x entre 0.1 y 10 separados medio punto y para valores de y y z entre -3 y 3 separados también en medio punto.</p> <pre> &gt;&gt; [x, y, z, v] = flow(10); [xi, yi, zi] = meshgrid(.1:.5:10, -3:.5:3, -3:.5:3); vi = interp3 (x, y, z, v, xi, yi, zi) ; slice(xi, yi, zi, vi, [6 9.5], 2, [-2 .2]), shading flat </pre>

<p><b>Y = interpft(X,n)</b></p> <p><b>y = interpft(x,n,dim)</b></p>	<p><i>Interpolación unidimensional usando el método FFT. Da el vector Y que contiene los valores de la función periódica X muestreada en n puntos igualmente espaciados. El vector original X se transforma al dominio de frecuencias de Fourier mediante la transformada rápida de Fourier FFT</i></p> <p><i>Opera a lo largo de la dimensión especificada</i></p> <p>Se presenta un ejemplo donde se comparan los puntos originales con los puntos interpolados a través del método FFT</p> <pre>&gt;&gt; y = [0:.5:2 1.5:-.5:-2 -1.5:.5:0]; % Puntos igualmente espaciados     factor =5; % Interpolación con un factor de 5     m = length (y)*factor;     X = 1:factor:m;     xi = 1:m;     yi = interpft(y,m);     plot(x,y, 'o', xi,yi,'*')     legend('datos originales','datos interpolados')</pre> 
<p><b>Vi =</b>  <b>interpn(X,Y,Z,... V,Xi, Yi,Zi...)</b></p> <p><b>Vi =</b>  <b>interpn(V,Xi,Yi,Zi)</b></p> <p><b>Vi =</b>  <b>interpn(V,n)</b></p> <p><b>Vi =</b>  <b>interpn(...,método)</b></p>	<p><i>Da el vector Vi tal que (Xi, Yi, Zi,...) es el conjunto total de puntos hallados por interpolación n-dimensional del conjunto de puntos resultante de aplicar la función tridimensional V a los puntos (X,Y,Z,...)</i></p> <p><i>Supone X=1 :n, Y=1:m, Z=1:p,...y (n,m,p,...) = tamaño de V</i></p> <p><i>Interpolación recursiva n-dimensional n veces</i></p> <p><i>Interpolación mediante los métodos especificados</i></p>
<p><b>Yi = pchip(X,Y,Xi)</b></p>	<p><i>Da el vector Yi tal que (Xi, Yi) es el conjunto total de puntos</i></p>

<b>pp = pchip(X,Y)</b>	<i>hallados por interpolación cúbica polinomial de Hermite a trozos (PCHIP) del conjunto de puntos dado (X,Y) Realiza interpolación en puntos intermedios</i>
<b>Yi = spline(X,Y,Xi)</b>  <b>pp = spline(X,Y)</b>	<i>Da el vector Yi tal que (Xi, Yi) es el conjunto total de puntos hallados por interpolación cúbica spline del conjunto de puntos dado (X, Y) Realiza interpolación en puntos intermedios</i>  Se presenta un ejemplo donde se comparan los puntos originales con los puntos interpolados a través de método <i>pchip</i> y <i>spline</i> . <pre>&gt;&gt; x = -3:3; y = [-1-1-1 0 1 1 1]; t = -3:.01:3; plot (x,y,'o',t,[pchip(x,y,t); spline(x,y,t)]); legend('datos','pchip','spline',4)</pre> 
<b>Zi = griddata(X,Y,Z,Xi,Yi)</b>  <b>[Xi,Yi,Zi] = griddata(X,Y,Z,Xi,Yi)</b>  <b>[...] = griddata(...,método)</b>	<i>Ajusta una superficie de la forma Z=f(X, Y) en el espacio de los vectores (X,Y,Z). El vector Zi determina los puntos de interpolación (Xi, Yi, Zi) entre los puntos dados (X, Y, Z). Se utiliza un método de la distancia inversa para interpolar</i>  <i>Además de Xi, da los vectores fila y columna Yi y Zi</i>  <i>Interpolación mediante los métodos especificados</i>  A continuación se presenta un ejemplo de ajuste por interpolación de una superficie. <pre>x = rand(100,1)*4-2; y = rand (100, 1)*4-2 ; z = x.*exp (-x.^2-y.^2); ti = -2:.25:2;</pre>

	<pre>[xi,yi] = meshgrid(ti,ti); zi = griddata(x,y,z,xi,yi); mesh(xi,yi,zi), hold on, plot3(x,y,z, 'o') , hold off</pre>
<b>W =</b> <b>griddata3(X,Y,Z,V,Xi, Yi,Zi)</b>  <b>W =</b> <b>griddata3(...,'método')</b>	<p>Ajusta una hipersuperficie de la forma <math>W=f(X,Y,Z)</math> en el espacio de los vectores <math>(X,Y,Z,V)</math>. El vector W determina los puntos de interpolación <math>(Xi,Yi,Zi,Vi)</math> entre los puntos dados <math>(X,Y,Z,V)</math>. Se utiliza un método de la distancia inversa para interpolar. Interpolación mediante los métodos especificados</p> <p>A continuación se presenta un ejemplo de ajuste por interpolación de una hipersuperficie</p> <pre>&gt;&gt; x = 2*rand(5000,1)-1; y = 2*rand(5000,1)-1; z = 2*rand(5000,1)-1; V = x.^2 + y.^2 + z.^2; d = -0.8:0.05:0.8; [xi,yi,zi] = meshgrid (d,d,d); w = griddata3(X,y,z,v,xi,yi,zi); p = patch(isosurface(xi,yi,zi,w,0.8)); isonormals(xi,yi,zi,w,p); set(p,'FaceColor','blue','EdgeColor','none'); view (3), axis equal, axis off, camlight, lighting phong</pre>

## RESOLUCION DE ECUACIONES Y SISTEMAS

MATLAB contempla múltiples comandos para la resolución de ecuaciones y sistemas. En los apartados siguientes se presenta la sintaxis y características de estos métodos.

### *Métodos generales*

A continuación se presentan los comandos más habituales en MATLAB para resolver ecuaciones y sistemas

<b>solve('ecuación','x')</b>	<p><i>Resuelve la ecuación en la variable x</i></p> <p>Se resuelve la ecuación <math>p * \sin(x) = r</math></p> <pre>&gt;&gt; solve('p*sin(x) = r') ans =     asin(r/p) pi - asin(r/p)</pre>
<b>syms x ; solve(ecu(x),x)</b>	<p><i>Resuelve la ecuación <math>ecu(x)=0</math> en la variable x</i></p> <p>Se resuelve la ecuación <math>p * \cos(x) = r</math></p> <pre>&gt;&gt; syms x r; solve(p*cos(x)-r, x) ans = acos((8192*r)/1433639)</pre>
<b>solve('ec1,ec2,...,ecn', 'x1, x2,...,xn')</b>	<p><i>Resuelve n ecuaciones simultáneas <math>ec1,...,ecn</math> (sistema en las variables <math>x1,...,xn</math>)</i></p> <pre>&gt;&gt; [x,y] = solve('x^2 + x*y + y = 3', 'x^2 - 4*x + 3 = 0') x = 1/3 y = -1/2</pre>
<b>syms x1 x2...xn;</b> <b>solve(ec1,ec2,...,ecn, x1, x2,...,xn)</b>	<p><i>Resuelve n ecuaciones simultáneas <math>ec1,...,ecn</math> (sistema en las variables <math>x1,...,xn</math>)</i></p>

	<pre>&gt;&gt; syms x y; [x,y] = solve(x^2 + x*y + y - 3, x^2 - 4*x + 3)  x = 1 3  y = 1 -3/2</pre>
<b>X = linsolve(A,B)</b>	<p><i>Resuelve <math>A *X=B</math> para una matriz cuadrada A, siendo B y X matrices</i></p> <p>Resolvemos el sistema:</p> $\begin{aligned} 2x + y + z + t &= 1 \\ x + 2y + z + t &= 2 \\ x + y + 2z + t &= 3 \\ x + y + z + 2t &= 4 \end{aligned}$ <pre>&gt;&gt; A=[2,1,1,1;1,2,1,1;1,1,2,1;1,1,1,2] ;B=[1,2,3,4] linsolve(A,B)  ans = -1 0 1 2</pre>
<b>x = lsqcov(A,B)</b> <b>x = lsqcov(A,B,V)</b>	<p><i>Resuelve <math>A *x=B</math> en el sentido de mínimos cuadrados</i></p> <p><i>Resuelve <math>A *x=B</math> en el sentido de mínimos cuadrados con matriz de covarianzas del error proporcional a V. O sea, da el vector x tal que <math>Ax=b+e</math> con <math>e \sim N(0, V)</math>. Da x (vector) que minimiza <math>(A^*x-B)^T *inv(V) *(A^*x-B)</math>.</i></p> <pre>&gt;&gt; A=[2,1,1,1;1,2,1,1;1,1,2,1;1,1,1,2];B=[1,2,3,4]'; lsqcov(A,B)  ans = -1 0 1 2</pre>
<b>X = A\B</b>	<p><i>Resuelve el sistema <math>A *X=B</math></i></p> <pre>&gt;&gt; A=[2,1,1,1;1,2,1,1;1,1,2,1;1,1,1,2];B=[1,2,3,4]'; A\B  ans = -1.0000 -0.0000 1.0000 2.0000</pre>

<b>X = A/B</b>	<i>Resuelve el sistema X*A=B</i>
<b>roots(A)</b>	<p><i>Da las raíces del polinomio cuyos coeficientes construyen el vector A (de mayor a menor orden)</i></p> <p>Como ejemplo hallamos las raíces del polinomio <math>x^4+2x^3+3x+4</math></p> <pre>&gt;&gt; roots([1 2 3 4]) ans = -1.6506 -0.1747 + 1.5469i -0.1747 - 1.5469i</pre>
<b>poly(V)</b>	<p><i>Da el polinomio cuyas raíces son el vector V</i></p> <pre>&gt;&gt; poly([1 2 3 4]) ans = 1 -10 35 -50 24</pre>
<b>x=fzero(función,x0)</b>	<p><i>Halla un cero de la función cerca de x0</i></p> <pre>&gt;&gt; x = fzero('sin(x)-1', pi/2) x = 1.5708</pre>
<b>[x, feval]=fzero(fun,x0)</b>	<p><i>Da también el valor de la función objetivo en x</i></p> <pre>&gt;&gt; [Xf] =fzero('sin(x)-1', pi/2)  x = 1.5708 f = 0</pre>

### **Método de los gradientes biconjugados**

A continuación se presentan los comandos que aporta MATLAB para resolver ecuaciones y sistemas por el método de los gradientes biconjugados.

<b>x = bicg(A,b)</b>	<p><i>Intenta resolver el sistema Ax=b por el método de los gradientes biconjugados</i></p> <pre>&gt;&gt; A=[pi 2*pi 3*pi -pi; 1 0 -1 2; exp(1) exp(2) exp(3) exp(4); i 2i 3i -i]; &gt;&gt; B=[1 2 3 4]'; &gt;&gt; bicg (A,B) bicg stopped at iteration 4 without converging to the desired tolerance, le-006 because the maximum number of iterations was reached. The iterate returned (number 0) has relative residual 1</pre>
----------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<pre><b>bicg(A,b,tol)</b> <b>bicg(A,b,tol,maxit)</b></pre> <pre><b>bicg(A,b,tol,maxit,M)</b> <b>bicg(A,b,tol,maxit,M1 ,M2)</b> <b>bicg(A,b,tol,maxit,M1,M2,x0)</b> <b>[x,f] = bicg(A,b,...)</b></pre>	<pre>ans = 0 0 0 0 0</pre> <p><i>Resuelve <math>Ax=b</math> especificando la tolerancia</i></p> <p><i>Resuelve <math>Ax=b</math> especificando la tolerancia y el número máximo de iteraciones</i></p> <pre>&gt;&gt; bicg(A,B, 1e-009, 100)  ans = 1.0e+016 * 4.1449 - 0.7033i -7.1997 + 1.2216i 3.2729 - 0.5553i -0.4360 + 0.0740i</pre> <p><i>Resuelve el sistema <math>inv(M) *A *x = inv(M) *b</math></i></p> <p><i>Resuelve el sistema <math>inv(M) *A *x = inv(M) *b</math> con <math>M=M1*M2</math></i></p> <p><i>Resuelve el sistema <math>inv(M) *A *x = inv(M) *b</math> con <math>M=M1 *M2</math> y valor inicial <math>x0</math></i></p> <p><i>Resuelve el sistema y f indica el resultado (0=convergencia, 1=no convergencia, 2=convergencia condicionada, 3=estancamiento y 4=números muy extremos)</i></p> <pre>&gt;&gt; [x, f]=bicg(A,B, 1e-009, 100) x = 1.0e+016 * 4.1449 - 0.7033i -7.1997 + 1.2216i 3.2129 - 0.5553i -0.4360 + 0.0740i  f = 3</pre>
<b>x = bicgstab(A,b)</b>	<p><i>Intenta resolver el sistema <math>Ax=b</math> por el método de los gradientes biconjugados estabilizado</i></p> <pre>&gt;&gt; bicgstab (A,B) bicgstab stopped at iteration 4 without converging to the desired tolerance 1e-006 because the maximum number of iterations was reached. The iterate returned (number 4) has relative residual 0.88</pre>

<b>bicgstab(A,b,tol)</b>	ans = 1.0e+011 * 0.6696 - 0.4857i -1.1631 + 0.8437i 0.5287 - 0.3835i -0.0704 + 0.0511i
<b>bicgstab(A,b,tol,maxit)</b>	<i>Resuelve Ax=b especificando la tolerancia</i> <i>Resuelve Ax=b especificando la tolerancia y el número máximo de iteraciones</i> >> bicg(A,B, 1e-009, 100) ans =
<b>bicgstab(A,b,tol,maxit,M)</b>	1.0e+016 * 4.1449 - 0.7033i -7.1997 + 1.2216i 3.2729 - 0.5553i -0.4360 + 0.0740i
<b>bicgstab(A,b,tol,maxit,M1,M2)</b>	<i>Resuelve el sistema inv(M) *A *x = inv(M) *b</i> <i>Resuelve el sistema inv(M) *A *x = inv(M) *b con M=M1*M2</i> <i>Resuelve el sistema inv(M) *A = inv(M) *b con M=M1 *M2 y valor inicial x0</i> <i>Resuelve el sistema y f indica el resultado (0=convergencia, 1=no convergencia, 2=convergencia condicionada, 3=estancamiento y 4=números muy extremos)</i> <i>Devuelve también el residuo relativo norm(b-A *x)/norm(b)</i> <i>Devuelve también el número de iteraciones</i>
<b>[x,f] = bicgstab(A,b,...)</b>	>> [x,f,r,i]=bicg(A,B, 1e-006, 100) x = 1.0e+016 * 4.1449 - 0.7033i -7.1997 + 1.2216i 3.2729 - 0.5553i -0.4360 + 0.0740i
<b>[x,f,relres] = bicgstab(A,b,...)</b>	f = 3
<b>[x,f,relres,iter] = bicgstab(A,b,...)</b>	r = 26.0415
	i = 18

## Método de los gradientes conjugados

A continuación se presentan los comandos que aporta MATLAB para resolver ecuaciones y sistemas por el método de los gradientes conjugados.

<b>x = pcg(A,b)</b> <b>pcg(A,b,tol)</b> <b>pcg(A,b,toI,maxit)</b> <b>pcg(A,b,tol,maxit,M)</b> <b>pcg(A,b,tol,maxit,M1,M2)</b> <b>pcg(A,b,tol,maxit,M1,M2,x0)</b> <b>[x,f] = pcg(A,b,...)</b>  <b>[x,f,relres] = pcg(A,b,...)</b>  <b>[x,f,relres,iter] = pcg(A,b,...)</b>	<i>Intenta resolver el sistema <math>Ax=b</math> por el método del gradiente conjugado precondicionado</i> <i>Resuelve <math>Ax=b</math> especificando la tolerancia</i> <i>Resuelve <math>Ax=b</math> especificando la tolerancia y el número máximo de iteraciones</i> <i>Resuelve el sistema <math>\text{inv}(M) *A*x = \text{inv}(M)*b</math></i> <i>Resuelve el sistema <math>\text{inv}(M) *A*x = \text{inv}(M)*b</math> con <math>M=M1*M2</math></i> <i>Resuelve el sistema <math>\text{inv}(M) *A*x = \text{inv}(M)*b</math> con <math>M=M1*M2</math> y valor inicial <math>x0</math></i> <i>Resuelve el sistema y indica el resultado</i> <i>(0=convergencia, 1=no convergencia,</i> <i>2=convergencia condicionada,</i> <i>3=estancamiento y 4=números muy extremos)</i> <i>Devuelve también el residuo relativo</i> <i><math>\text{norm}(b-A*x)/\text{norm}(b)</math></i> <i>Devuelve también el número de iteraciones</i> <pre>&gt;&gt; A=[pi 2*pi 3*pi -pi; 1 0 -1 2; exp(1)       exp(2) exp(3) exp(4); i 2i 3i -i]; &gt;&gt; B=[1 2 3 4]'; &gt;&gt; [x,f,r,i]=pcg(A,B, 1e-006, 1000)</pre> <p>x =</p> <pre>0 0 0 0</pre> <p>f =</p> <pre>4</pre> <p>r =</p> <pre>1</pre> <p>i =</p> <pre>0</pre>
<b>x = lsqr(A,b)</b> <b>lsqr(A,b,tol)</b> <b>lsqr(A,b,tol,maxit)</b>	<i>Intenta resolver el sistema <math>Ax=b</math> por el método del gradiente conjugado en ecuaciones normales</i> <i>Resuelve <math>Ax=b</math> especificando la tolerancia</i> <i>Resuelve <math>Ax=b</math> especificando la tolerancia y</i>

<b>lsqr(A,b,tol,maxit,M)</b>	<i>el número máximo de iteraciones</i>
<b>lsqr(A,b,tol,maxit,M1,M2)</b>	<i>Resuelve el sistema <math>\text{inv}(M)*A*x = \text{inv}(M)*b</math></i>
<b>lsqr(A,b,tol,maxit,M1,M2,x0)</b>	<i>Resuelve el sistema <math>\text{inv}(M)*A*x = \text{inv}(M)*b</math> con <math>M=M1*M2</math></i>
<b>[x,f] = lsqr(A,b,...)</b>	<i>Resuelve el sistema y f indica el resultado (0=convergencia, 1=no convergencia, 2=convergencia condicionada, 3=estancamiento y 4=números muy extremos)</i>
<b>[x,f,relres] = lsqr(A,b,...)</b>	<i>Devuelve también el residuo relativo <math>\text{norm}(b-A*x)/\text{norm}(b)</math></i>
<b>[x,f,relres,iter] = lsqr(A,b,...)</b>	<i>Devuelve también el número de iteraciones</i>
	<pre>&gt;&gt; A=[pi 2*pi 3*pi -pi; 1 0 -1 2; exp(1) exp(2) exp(3) exp(4); i 2i 3i -i]; &gt;&gt; B=[1 2 3 4]'; &gt;&gt; [x,f,r,i]=lsqr(A,B, 1e-006, 1000)  x = 1.1869 - 0.0910i 0.4295 - 0.0705i -0.5402 - 0.0362i 0.1364 + 0.0274i  f = 0  r = 0.6981  i = 3</pre>

### ***Método residual***

A continuación se presentan los comandos que aporta MATLAB para resolver ecuaciones y sistemas por el método residual.

<b>x = qmr(A,b)</b>	<i>Intenta resolver el sistema <math>Ax=b</math> por el método residual cuasiminimal</i>
<b>qmr(A,b,tol)</b>	<i>Resuelve <math>Ax=b</math> especificando la tolerancia</i>
<b>qmr(A,b,tol,maxit)</b>	<i>Resuelve <math>Ax=b</math> especificando la tolerancia y el número máximo de iteraciones</i>

<b>qmr(A,b,tol,maxit,M)</b>	<i>Resuelve el sistema <math>inv(M) *A *x = inv(M) *b</math></i>
<b>qmr(A,b,tol,maxit,M1,M2)</b>	<i>Resuelve el sistema <math>inv(M) *A *x = inv(M) *b</math> con <math>M=M1*M2</math></i>
<b>qmr(A,b,tol,maxit,M1,M2,x0)</b>	<i>Resuelve el sistema <math>inv(M) *A *x = inv(M) *b</math> con <math>M=M1 *M2</math> y valor inicial <math>x0</math></i>
<b>[x,f] = qmr(A,b,...)</b>	<i>Resuelve el sistema y f indica el resultado (0=convergencia, 1=no convergencia, 2=convergencia condicionada, 3=estancamiento y 4=números muy extremos)</i>
<b>[x,f,relres] = qmr(A,b,...)</b>	<i>Da también el residuo relativo <math>norm(b-A *x)/norm(b)</math></i>
<b>[x,f,relres,iter] = qmr(A,b,...)</b>	<i>Devuelve también el número de iteraciones</i>
	<pre>&gt;&gt; A=[pi 2*pi 3*pi -pi; 1 0 -1 2; exp(1) exp(2) exp(3) exp(4); i 2i 3i -i]; &gt;&gt; B=[1 2 3 4]'; &gt;&gt; [x, f, r, i] =qmr (A,B, 1e-006, 1000)  x = 1.0e+016 * 0.4810 - 4.0071i -0.8356 + 6.9603i 0.3798 - 3.1640i -0.0506 + 0.4215i  f = 3  r = 19.5999  i = 11</pre>
<b>X = gmres(A,b)</b>	<i>Intenta resolver el sistema <math>Ax=b</math> por el método residual mínimo generalizado</i>
<b>gmres(A,b,tol)</b>	<i>Resuelve <math>Ax=b</math> especificando la tolerancia</i>
<b>gmres(A,b,tol,maxit)</b>	<i>Resuelve <math>Ax=b</math> especificando la tolerancia y el número máximo de iteraciones</i>
<b>gmres(A,b,tol,maxit,M)</b>	<i>Resuelve el sistema <math>inv(M) *A*x = inv(M)*b</math></i>
<b>gmres(A,b,tol,maxit,M1,M2)</b>	<i>Resuelve el sistema <math>inv(M) *A*x = inv(M)*b</math> con <math>M=M1*M2</math></i>
<b>gmres(A,b,tol,maxit,M1,M2,x0)</b>	<i>Resuelve el sistema <math>inv(M) *A*x = inv(M)*b</math> con <math>M=M1 *M2</math> y valor inicial <math>x0</math></i>
<b>[x,f] = gmres(A,b,...)</b>	<i>Resuelve el sistema y f indica el resultado (0=convergencia, 1=no convergencia, 2=convergencia condicionada, 3=estancamiento y 4=números muy extremos)</i>

<p><b>[x,f,relres] = gmres(A,b,...)</b></p> <p><b>[x,f,relres,iter] = gmres(A,b,...)</b></p>	<p>Devuelve también el residuo relativo  <math>norm(b-A *x)/norm(b)</math></p> <p>Devuelve también el número de iteraciones</p> <pre>&gt;&gt; A=[pi 2*pi 3*pi -pi; 1 0 -1 2; exp(1) exp(2) exp(3) exp(4); i 2i 3i -i]; &gt;&gt; B=[1 2 3 4];  &gt;&gt; [x, f, r, i] = gmres (A,B) x = 1.5504 + 0.0085i -0.2019 - 0.2433i -0.2532 + 0.0423i 0.0982 + 0.0169i  f = 3  r = 0.6981  i = 1 4</pre>
<p><b>x = minres(A,b)</b></p> <p><b>minres(A,b,tol)</b></p> <p><b>minres(A,b,tol,maxit)</b></p> <p><b>minres(A,b,toI,maxit,M)</b></p> <p><b>minres(A,b,tol,maxit,M1,M2)</b></p> <p><b>minres(A,b,tol,maxit,M1,M2,x0)</b></p> <p><b>[x,f] = minres(A,b,...)</b></p> <p><b>[x,f,relres] = minres(A,b,...)</b></p> <p><b>[x,f,relres,iter] = minres(A,b,...)</b></p>	<p>Intenta resolver el sistema <math>Ax=b</math> por el método residual mínimo</p> <p>Resuelve <math>Ax=b</math> especificando la tolerancia</p> <p>Resuelve <math>Ax=b</math> especificando la tolerancia y el número máximo de iteraciones</p> <p>Resuelve el sistema <math>inv(M)*A*x = inv(M)*b</math></p> <p>Resuelve el sistema <math>inv(M) *A *x = inv(M)*b</math> con <math>M=M1*M2</math></p> <p>Resuelve el sistema <math>inv(M) *A *x = inv(M) *b</math> con <math>M=M1 *M2</math> y valor inicial <math>x0</math></p> <p>Resuelve el sistema y f indica el resultado (0=convergencia, 1=no convergencia, 2=convergencia condicionada, 3=estancamiento y 4=números muy extremos)</p> <p>Devuelve también el residuo relativo  <math>norm(b-A *x)/norm(b)</math></p> <p>Devuelve también el número de iteraciones</p> <pre>&gt;&gt; A=[pi 2*pi 3*pi -pi; 1 0 -1 2; exp(1) exp(2) exp(3) exp(4); i 2i 3i -i]; &gt;&gt; B=[1 2 3 4'];  &gt;&gt; [x, f, r, i] = minres (A, B, 1e-006, 1000) x = 0.0748 - 0.0070i</pre>

	<pre> -0.0761 - 0.0001i 0.5934 - 0.1085i -0.1528 + 0.0380i  f = 1 r = 0.0592 i = 1000 </pre>
--	----------------------------------------------------------------------------------------------

### Método simétrico y de mínimos cuadrados no negativos

A continuación se presentan los comandos que aporta MATLAB para resolver ecuaciones y sistemas por el método simétrico y de mínimos cuadrados no negativos.

<b>x = symmlq(A,b)</b>	<i>Intenta resolver el sistema <math>Ax=b</math> por el método LQ simétrico</i>
<b>symmlq(A,b,tol)</b>	<i>Resuelve <math>Ax=b</math> especificando la tolerancia</i>
<b>symmlq(A,b,tol,maxit)</b>	<i>Resuelve <math>Ax=b</math> especificando la tolerancia y el número máximo de iteraciones</i>
<b>symmlq(A,b,tol,maxit,M)</b>	<i>Resuelve el sistema <math>\text{inv}(M) *A*x = \text{inv}(M)*b</math></i>
<b>symmlq(A,b,tol,maxit,M1,M2)</b>	<i>Resuelve el sistema <math>\text{inv}(M) *A*x = \text{inv}(M)*b</math> con <math>M=M1*M2</math></i>
<b>symmlq(A,b,tol,maxit,M1,M2,x0)</b>	<i>Resuelve el sistema <math>\text{inv}(M) *A*x = \text{inv}(M)*b</math> con <math>M=M1*M2</math> y valor inicial <math>x0</math></i>
<b>[x,flag] = symmlq(A,b,...)</b>	<i>Resuelve el sistema e indica el resultado (0=convergencia, 1=no convergencia, 2=convergencia condicionada, 3=estancamiento y 4=números muy extremos)</i>
<b>[x,flag,relres] = symmlq(A,b,...)</b>	<i>Devuelve también el residuo relativo <math>\text{norm}(b-A*x)/\text{norm}(b)</math></i>
<b>[x,flag,relres,iter] = symmlq(A,b,...)</b>	<i>Devuelve también el número de iteraciones</i>
	<pre> &gt;&gt; A=[pi 2*pi 3*pi -pi; 1 0 -1 2; exp(1) exp(2) exp(3) exp(4); i 2i 3i -i]; &gt;&gt; B=[1 2 3 4]'; &gt;&gt; [x,f,r,i] =syminlg(A,B, 1e-006,1000)  x= 0.0121 - 0.0004i 0.0035 - 0.0001i 0.1467 - 0.0061i 0.0001 + 0.0039i </pre>

	<pre>f = 1 r = 0.8325 i = 3</pre>
<b>x = lsqnonneg(C,d)</b>	<i>Da x que minimiza norm(C*x-d) sujeto a x≥0 por mínimos cuadrados (C y d reales)</i>
<b>x = lsqnonneg(C,d,x0)</b>	<i>Da x que minimiza norm(C*x-d) sujeto a x≥0, pero con x=x0≥0 como valor inicial</i>
<b>x = lsqnonneg(C,d,x0,opt)</b>	<i>Da x que minimiza norm(C*x-d) sujeto a x≥0, con x=x0≥0 como valor inicial y con la opción opt</i> <i>Las opciones son TolX para la tolerancia y Display para mostrar la salida ('off' no muestra la salida, 'final' muestra el final de la salida y 'notify' muestra la salida sólo si no hay convergencia)</i>
<b>[x,resnorm] = lsqnonneg(...)</b>	<i>Da la solución y el valor de la 2-norma cuadrada del residuo norm(C*x-d)^2</i>
<b>[x,resnorm,residual] = lsqnonneg(...)</b>	<i>Da la solución y el residuo C*x-d</i>
<b>[x,resnorm,residual,f] = lsqnonneg(...)</b>	<i>Da la solución, el residuo C*x-dy un valor f positivo o nulo según converja o no la solución</i>
<b>[x,resnorm,residual,f,out,lambda] = lsqnonneg(...)</b>	<i>Da la solución, el residuo C*x-d, el valor f, el algoritmo usado y el vector de los multiplicadores de Lagrange lambda</i>
	<pre>&gt;&gt; A=[1 2 3;5 7 1;2 3 6]; B= [1 3 5]'; lsqnonneg (A, B)  ans = 0.4857 0 0.5714</pre>

## RESOLUCION DE ECUACIONES Y SISTEMAS LINEALES

En los párrafos anteriores hemos estudiado las ecuaciones y sistemas en general. Ahora nos ocuparemos especialmente de las ecuaciones y sistemas lineales. Para trabajar en este campo es posible utilizar los comandos que hemos visto hasta ahora, pero MATLAB dedica una parcela especial a las ecuaciones y sistemas lineales implementando comandos específicos para su resolución. La tabla siguiente muestra estos comandos.

<b>X = linsolve(A,B)</b>	<p>Resuelve el sistema lineal <math>A*X=B</math></p> <p>Resolvemos el sistema:</p> $\begin{aligned} 2x + y + z + t &= 1 \\ x + 2y + z + t &= 2 \\ x + y + 2z + t &= 3 \\ x + y + z + 2t &= 4 \end{aligned}$ <pre>&gt;&gt; A=[2,1,1,1;1,2,1,1;1,1,2,1;1,1,1,2];B=[1,2,3,4] linsolve(A,B) ans = -1 0 1 2</pre>
<b>[X,R] = linsolve(A,B)</b>	<p>Resuelve el sistema lineal <math>A*X=B</math> y devuelve adicionalmente el inverso del número de condición de <math>A</math> si <math>A</math> es cuadrada y devuelve su rango si <math>A</math> no es cuadrada.</p> <pre>&gt;&gt; A=[2,1,1,1; 1,2,1,1; 1,1,2,1,-1,1,1,2]; B=[1,2,3,4]'; [X,R]=linsolve(A,B)  X = -1 0 1 2 R = 0.1429</pre>
<b>X = linsolve(A,B,opciones)</b>	<p>Resuelve el sistema lineal <math>A*X=B</math> utilizando distintas opciones para la matriz <math>A</math> (<i>UT</i> para triangular superior, <i>LT</i> para triangular inferior, <i>SYM</i> para simétrica real o compleja hermitiana, <i>RECT</i> para rectangular general, <i>POSDEF</i> para definida positiva, <i>UHESS</i> para Hessemberg superior y <i>TRANSA</i> para traspuesta)</p>
<b>rank(A)</b>	<p>Rango de la matriz <math>A</math></p> <pre>&gt;&gt; rank(A) ans = 4</pre>
<b>det (A)</b>	<p>Determinante de la matriz cuadrada <math>A</math></p> <pre>&gt;&gt; det (A) ans = 5</pre>
<b>Z=null (A,'r')</b>	<p>Base racional para el núcleo de <math>A</math></p>

Los sistemas de ecuaciones lineales pueden convertirse a forma matricial y resolverse utilizando el cálculo con matrices. Un sistema puede escribirse en la forma  $M \cdot X = B$ , siendo  $X$  el vector de variables,  $B$  el vector de términos independientes y  $M$  la matriz de coeficientes del sistema. Si  $M$  es una matriz cuadrada y el determinante de la matriz  $M$  es no nulo,  $M$  es invertible, y la solución única del sistema puede escribirse en la forma;  $X=M^{-1}B$ . En este caso, los comandos *solve*, *linsolve*, *lscov*, *bicg*, *pcg*, *lsqr*, *gmres*, *minres*, *symmlq* o *M\B*, ya descritos anteriormente, ofrecen la solución.

Si el determinante de  $M$  es nulo, el sistema tiene infinitas soluciones, con lo que hay filas o columnas en  $M$  que son linealmente dependientes. En este caso, se calculará el número de ecuaciones redundantes para saber cuántas variables tenemos que fijar para dar las infinitas soluciones. Si la matriz  $M$  es rectangular (no cuadrada), puede que el sistema sea indeterminado (número de ecuaciones menor que número de variables), sobredeterminado (número de ecuaciones mayor que el número de variables) o no singular (número de ecuaciones igual a número de variables y determinante de  $M$  no nulo). Un sistema indeterminado puede tener infinitas soluciones o ninguna, y lo mismo le pasa a un sistema sobredeterminado. Si no existe solución para un sistema, se llama inconsistente (incompatible), y si al menos existe una solución, se llama consistente (compatible). El sistema  $M \cdot X = B$  se llama *homogéneo* cuando el vector  $B$  es el vector nulo. El sistema quedará de la forma  $M \cdot X = 0$ . Si el determinante de  $M$  es no nulo, la única solución del sistema es el vector nulo (se obtiene con el comando *linsolve*). Si el determinante de  $M$  es nulo, el sistema tiene infinitas soluciones (es compatible indeterminado). Las soluciones se hallarán mediante los comandos *solve* o *linsolve*, *lsqr* u otros ya vistos anteriormente para los sistemas lineales generales.

Un instrumento fundamental para el análisis y solución de sistemas de ecuaciones es el *teorema de Ronché Frobenius*. Este teorema dice que un sistema de  $m$  ecuaciones con  $n$  incógnitas tiene alguna solución si, y solo si, el rango de la matriz de los coeficientes coincide con el rango de la matriz ampliada con el vector columna de los términos independientes del sistema. Si los dos rangos son iguales e iguales al número de incógnitas, el sistema tiene solución única. Si los dos rangos son iguales, pero menores que el número de incógnitas, el sistema tiene infinitas soluciones. Si son distintos, el sistema no tiene solución.

Sea  $A$  la matriz de coeficientes del sistema y  $B$  la matriz ampliada con el vector de términos independientes.

Si  $\text{rango}(A) \neq \text{rango}(B)$ , el sistema es incompatible (sin solución).

Si  $\text{rango}(A)=\text{rango}(B) < n$ , el sistema es compatible indeterminado (tiene infinitas soluciones).

Si  $\text{rango}(A)=\text{rango}(B) = n$ , el sistema es compatible determinado (tiene solución única).

Este teorema permite analizar las soluciones de un sistema de ecuaciones antes de resolverlo.

Ya sabemos que otro tipo de sistemas lineales especiales son los *sistemas homogéneos*. El sistema  $A \cdot X = B$  se dice homogéneo si el vector de términos independientes  $B$  es nulo, con lo que todo sistema homogéneo será de la forma  $A \cdot X = 0$ . En un sistema homogéneo, el rango de la matriz de los coeficientes y el rango de la matriz ampliada con la columna de los términos independientes siempre coinciden. Si aplicamos el teorema de Rouché-Frobenius, un sistema homogéneo tendrá una única solución cuando el determinante de la matriz  $A$  sea distinto de cero. Esta solución única es el vector nulo, ya que la solución nula siempre la tienen los sistemas homogéneos. Un sistema homogéneo tendrá infinitas soluciones cuando el determinante de la matriz  $A$  es cero. En este caso, las infinitas soluciones se calculan igual que en los sistemas generales (comando *solve*), o también utilizando la función *null* ( $A$ ).

Como primer ejemplo resolvemos el sistema:

$$\begin{aligned} 2x + y + z + t &= 1 \\ x + 2y + z + t &= 1 \\ x + y + 2z + t &= 1 \\ x + y + z + 2t &= 1 \end{aligned}$$

Buscaremos el rango de la matriz del sistema y de la matriz ampliada con la columna de términos independientes.

```
>> A=[2,1,1,1;1,2,1,1;1,1,2,1,-1,1,1,2];
>> B=[2, 1,1, 1,1;1,2,1,1;1,1,2,1,1;1,1,1,2,1];
>> [rank(A), rank(B)]
```

```
ans =
4           4
```

Observamos que los rangos de las dos matrices coinciden con el número de incógnitas. El teorema de Rouché-Frobenius asegura entonces que el sistema es compatible determinado con solución única. Podemos calcular la solución de la siguiente forma:

```
>> B=[1 1 1 1]';
>> linsolve(A,B)
ans =
0.2000
0.2000
0.2000
0.2000
```

También podría haberse resuelto el sistema con los siguientes comandos:

```
>> lscov(A,B)

ans =
0.2000
0.2000
0.2000
0.2000

>> bicg(A,B)
bicg converged at iteration 1 to a solution with relative residual 0

ans =
0.2000
0.2000
0.2000
0.2000

>> pcg(A,B)
pcg converged at iteration 1 to a solution with relative residual 0

ans =
0.2000
0.2000
0.2000
0.2000

>> lsqr(A,B)
lsqr converged at iteration 1 to a solution with relative residual 0

ans =
0.2000
0.2000
0.2000
0.2000

>> qmr(A,B)
qmr converged at iteration 1 to a solution with relative residual 0

ans =
0.2000
0.2000
0.2000
0.2000

>> gmres(A,B)
gmres converged at iteration 1 to a solution with relative residual
1.5e-016

ans =
```

```
0.2000
0.2000
0.2000
0.2000
0.2000

>> symmlq(A,B)
symmlq converged at iteration 1 to a solution with relative residual 0

ans =
0.2000
0.2000
0.2000
0.2000
```

Como segundo ejemplo resolvemos el sistema;

$$\begin{aligned}x + 2y + 3z &= 6 \\x + 3y + 8z &= 19 \\2x + 3y + z &= -1 \\5x + 6y + 4z &= 5\end{aligned}$$

Buscaremos el rango de la matriz del sistema y de la matriz ampliada con la columna de términos independientes.

```
>> [1, 2, 3; 1, 3, 8; 2, 3, 1; 5, 5, 4];
>> B=[1, 2, 3, 6; 1, 3, 8, 19, -2, 3, 1, -1, -5, 6, 4, 5];
>> [rank(A), rank(B)]
ans =
3      3
```

Observamos que los rangos de las dos matrices coinciden con le número de incógnitas. El teorema de Rouché-Frobenius asegura entonces que el sistema es compatible determinado con solución única. Podemos calcular la solución de la siguiente forma:

```
>> A=[1, 2, 3; 1, 3, 8; 2, 3, 1; 5, 6, 4];
>> B=[6 19 -1 5]';
>> linsolve(A,B)

ans =
1.0000
-2.0000
3.0000
```

Como tercer ejemplo resolvemos el sistema:

$$\begin{aligned}x + 2y - z &= 0 \\2x - y + z &= 0 \\3x + y &= 0\end{aligned}$$

Como estamos ante un sistema homogéneo calcularemos el determinante de la matriz de coeficientes del sistema.

```
>> A=[1,2,-1;2,-1,1;3,1,0];
>> det(A)
ans =
5.5511e-016
```

Como el determinante es prácticamente nulo, el sistema homogéneo tendrá infinitas soluciones, que se calculan con el comando *solve* tal y como se muestra a continuación.

```
>> [x,y,z]=solve('x+2*y-z', '2*x-y+z', '3*x+y', 'x,y,z')
x =
-z1/5
y =
(3*z1)/5
z =
z1
```

Se observa que las infinitas soluciones dependen del parámetro  $z_1$  y pueden escribirse de la siguiente forma;

$$\{(-z_1/5, 3z_1/5, z_1)\} z_1 \in \mathbb{R}$$

*Ejercicio 1. Expandir las expresiones algebraicas siguientes:*

$$(x + 1)(x + 2), \frac{x+1}{x+2}, \operatorname{Sen}(x + y), \operatorname{Cos}(2x), e^{a+Ln(b)}, \ln\left(\frac{x}{1-x^2}\right), (x+1)(y+z)$$

```
>> syms x y z t a b
>> pretty(expand((x+1)*(x+2)))
```

$$\begin{aligned}
 & \frac{2}{x + 3} x + 2 \\
 & \frac{x}{x + 2} + \frac{1}{x + 2}
 \end{aligned}$$

```

>> pretty(expand(sin(x+y)))
sin(x) cos(y) + cos(x) sin(y)

>> pretty(expand(cos(2*x)))
2
cos(x) - 1

>> pretty(expand(exp(a+log(b))))
exp(a) b

>> pretty(expand(log(x/(1-x)^2)))
log(x) - 2 log(1 - x)

>> pretty(expand((x+1)*(y+z)))
xy + xz + y + z

```

*Ejercicio 2. Factorizar las expresiones algebraicas siguientes:*

$$6x^2+18x \quad 24, \quad x^4 - y^4, \quad x^3 + y^3, \quad \frac{x^3-y^3}{x^4-y^4}$$

```

>> syms x y
>> pretty(factor(6*x^2+18*x-24))
6 (x + 4) (x - 1)

>> pretty(factor(x^4-y^4))
2 2
(x - y) (x + y) (x + y)

>> pretty(factor(x^3+y^3))
2 2
(x + y) (x - x y + y)

>> pretty(factor((x^3-y^3)/(x^4-y^4)))
2 2
x + x y + y
-----
2 2
(x + y) (x + y)

```

*Ejercicio 3. Simplificar las expresiones algebraicas siguientes:*

$$\sin^2(x) + \cos^2(x), \quad e^{a+Ln(be^c)}, \quad \cos(3a \cos(x)), \quad \frac{x^2-y^2}{(x-y)^3}$$

```

>> syms x y a b c
>> simplify (sin (x) ^2 + cos (x) ^2)
ans =
1

>> pretty(simplify(exp(a+log(b*exp(c)))))

b exp (a + c)

>> pretty(sym(simple(cos(3*acos(x)))))

3
4 x - 3 x
>> pretty (simple ( (x^2-y^2) / (x-y)^3 ))
x + y
-----
2
(x - y)

```

*Ejercicio 4. Agrupar términos en x en las expresiones algebraicas siguientes:*

$$f(x) = a^3 \cdot x + a^3 x + a, \quad p(x) = y/x + 2z/x + x^{1/3} - y*x^{1/3} \quad q(x) = (x+1)(x+2)$$

*Agrupar términos en sen(x) en la expresión: y(sen(x)+1)+sen(x)*

*Agrupar términos en Ln(x) en la expresión: f=aLn(x)-xLn(x)-x*

*Agrupar términos en (x,y) en la expresión: p=xy+zxy+yx^2+zxy^2+x+zx*

```

>> syms a x y z
>> pretty (collect (a^3*x-x+a^3+a, x))

3            3
(a - 1) x + a + a

>> pretty (collect (y/x+2*z/x+x^(1/3)-y*x^(1/3), x))

4            4
-            -
3            3
y + 2 z - x   y + x
-----
x

>> pretty(collect( (x+1)*(x+2) ))

2
x + 3 x + 2

>> p = x*y+z*x*y+y*x^2-z*y*x^2+x+z*x;

```

```
>> pretty(collect( p, [x,y] ))

$$(1 - z)^2 x^2 y + (z + 1) x^2 y + (z + 1) x^2$$

>> f = a*log(x)-log(x)*x-x;
>> pretty(collect(f,log(x)))

$$(a - x) \log(x) - x$$

```

*Ejercicio 5. Combinar términos lo más posible en la expresión siguiente:*

$$aln(x)+3ln(x)-ln(1-x)+ln(1+x)/2$$

*Realizar la simplificación suponiendo que a es real y  $x > 0$ .*

```
>> pretty(sym(simple(a*log(x)+3*log(x)-log(1-x)+log(1+x)/2)))

$$\frac{\log(x+1)}{2} - \log(1-x) + 3 \log(x) + a \log(x)$$

>> x = sym('x', 'positive')
x =
x
>> a = sym('a', 'real')
a =
a
>> pretty(sym(simple(a*log(x)+3*log(x)-log(1-x)+log(1+x)/2)))

$$-\log\left(\frac{x-1}{x^3 a^{\frac{1}{2}} (x+1)}\right)$$

```

*Ejercicio 6. Desarrollar y simplificar todo lo posible las siguientes expresiones trigonométricas:*

- a)  $\sin[3x]\cos[5x]$
- b)  $[(\cot[a])^2 + (\sec[a])^2 - (\csc[a])^2]$
- c)  $\sin[a]/(1+\cot[a]^2) - \sin[a]^3$

```
>> pretty(simple(expand(sym(sin(3*x)*cos(5*x)))))

$$\frac{\sin(8x) - \sin(2x)}{2}$$

```

```
>> pretty(simple(expand(((cot(a))^2+(sec(a))^2-(csc(a))^2))) )

$$\frac{1}{\cos^2(a)} - 1$$

>> pretty(simple(expand(sin(a)/(1+cot(a)^2)-sin(a)^3)))
0
```

*Ejercicio 7. Realizar las siguientes operaciones algebraicas simplificando al máximo el resultado:*

$$\frac{x}{x+y} - \frac{xy}{x-y} + \frac{2xy}{x^2-y^2}, \quad \frac{1+a^2}{b} + \frac{1-b^2}{a} - \frac{a^3-b^3}{ab}$$

```
>> pretty(simple(expand(x/(x+y)-y/(x-y)+2*x*y/(x^2-y^2))))
1
```

```
>> pretty(simple(expand((1+a^2)/b + (1-b^2)/a - (a^3-b^3)/(a*b))))

$$\frac{1}{a} + \frac{1}{b}$$

```

*Ejercicio 8. Simplificar lo más posible las siguientes fracciones algebraicas:*

$$\frac{a^3-a^2+c^2-c^2}{a^3+ac^2+a^2b+bc^2}, \quad \frac{(x^2-9)(x^2-2x+1)(x-3)}{(x^2-6x+9)(x^2-1)(x-1)}$$

```
>> pretty(simple(factor(a^3-a^2*b+a*c^2-b*c^2)/(a^3+a*c^2+a^2*b+b*c^2)))

$$\frac{a-b}{a+b}$$

>> pretty(simple(factor((x^2-9)*(x^2-2*x+1)*(x-3))/((x^2-6*x+9)*(x^2-1)*(x-1))))

$$\frac{2}{x+1} + 1$$

```

*Ejercicio 9. Calcular las raíces de los polinomios siguientes:*

$$x^3 - 6x^2 - 72x - 27, 2x^4 - 3x^3 + 4x^2 - 5x + 11, x^{11} - 1$$

*Evaluar el primer polinomio en la matriz unidad de orden 3, el segundo en la matriz de unos de orden 3 y el tercero en una matriz aleatoria uniforme de orden 3.*

*Hallar los coeficientes de los polinomios derivada de los polinomios dados, así como los propios polinomios.*

```
>> p1 = [1 -6 -72 -27]; r=roots(p)
r =
12.1229
-5.7345
-0.3884

>> p2 = [ 2 - 3   4   -5  11]; r=roots(p)
r =
1.2817 + 1.0040i
1.2817 - 1.0040i
-0.5317 + 1.3387i
-0.5317 - 1.3387i

>> p3 = [ 1  0  0  0  0  0  0  0  0  0  1]; r=roots(p)
r =
-1.0000
-0.8413 + 0.5406i
-0.8413 - 0.5406i
-0.4154 + 0.9096i
-0.4154 - 0.9096i
0.1423 + 0.9898i
0.1423 - 0.9898i
0.6549 + 0.7557i
0.6549 - 0.7557i
0.9595 + 0.2817i
0.9595 - 0.2817i

>> Y1=polyval(p1,eye(3))
Y1 =
-104   -27   -27
-27   -104   -27
-27   -27   -104

>> Y2=polyval(p2,ones(3))
Y2 =
```

```

9      9      9
9      9      9
9      9      9

>> Y3=polyval(p3,rand(3))
Y3 =
1.1050   1.3691   1.0000
1.3368   1.0065   1.0013
1.0          1.0000   1.6202

>> d1=polyder(p1)
d1 =
3      -12     -72

>> pretty(poly2sym(d1*x))
2
3      x - 12 x - 72

>> d2=polyder(p2)
d2 =
8      -9      8      -5

>> pretty(poly2sym(d2,'x'))
3      2
8x - 9x + 8 x - 5

>> d3=polyder(p3)
d3 =
1      1      0      0      0      0      0      0      0
0      0

>> pretty(poly2sym(d3,'x'))
10
11 x

```

*Ejercicio 10.* Consideramos los puntos del intervalo  $[0 \ 5]$  separados en una décima. Interolar la función error en estos puntos y ajustar un polinomio de grado 6 a los mismos. Representar la curva original y la interpolada en el mismo gráfico.

```

>> x = (0: 0.1: 5)';
y = erf(x);
f = polyval(p,x);
>> p = polyfit(x,y,6)

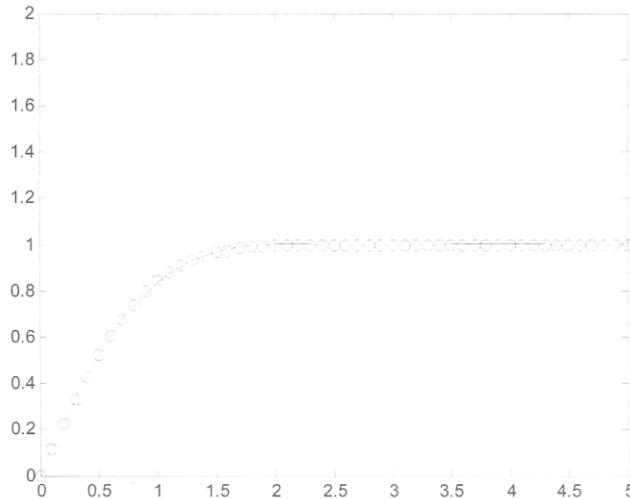
```

```
p =
```

```

0.0012   -0.0173   0.0812   -0.0791   -0.4495   1.3107   -0.0128
>> f = polyval(p,x);
plot(x,y, 'o',x,f, '-')
axis ([0 5 0 2])

```



*Ejercicio 11. Calcular el polinomio interpolador de segundo grado que pasa por los puntos (-1,4), (0,2), y (1,6) en el sentido de mínimos cuadrados.*

```

>> x=[-1, 0, 1]; y=[4, 2, 6]; p=poly2sym(polyfit(x,y,2))
P =
3*x^2+x+2

```

*Ejercicio 12. Representar 200 puntos de interpolación cúbica entre los puntos (x,y) dados por los valores que toma la función exponencial e para 20 valores de x igualmente espaciados entre 0 y 2. Utilizar interpolación cúbica.*

En primer lugar, definimos los 20 puntos dados  $(x,y)$ , igualmente espaciados entre 0 y 2:

```

>> x=0:0.1:2;
>> y=exp (x);

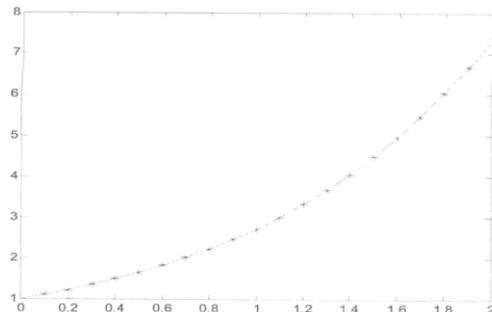
```

Ahora hallamos los 200 puntos de interpolación cúbica  $(xi,yi)$ , igualmente espaciados entre 0 y 2, y se representan en una gráfica, junto con los 20 puntos  $(x,y)$  iniciales (con asteriscos).

```

>> xi=0:0.01:2;
>> yi=interp1(x,y,xi,'cubic');
>> plot(x,y,'*',xi,yi)

```



*Ejercicio 13. Obtener 25 puntos de aproximación por interpolación de la función paramétrica  $X=\text{Cosh}(t)$ ,  $Y=\text{Senh}(t)$ ,  $Z=\text{Tanh}(t)$  para valores de  $t$  entre 0 y  $\pi/6$ , sobre el conjunto de puntos definido para valores de  $t$  en el intervalo  $(i \pi/6)$  con  $0 < i < 6$ .*

En primer lugar, definimos los 25 puntos dados  $(x,y,z)$ , igualmente espaciados entre 0 y  $\pi/6$ .

```
>> t=0:pi/150:pi/6;
>> x=cosh(t);y=sinh(t);z=tanh(t) ;
```

Ahora hallamos los 25 puntos de interpolación  $(x_i, y_i, z_i)$ , para valores del parámetro  $t$  igualmente espaciados entre 0 y  $\pi/6$ .

```
>> xi=cosh(t);yi=sinh(t);
>> zi=griddata(x,y,z,xi,yi) ;
>> puntos=[xi',yi',zi']
```

puntos =

1.0000	0	0
1.0002	0.0209	0.0209
1.0009	0.0419	0.0419
1.0020	0.0629	0.0627
1.0035	0.0839	0.0836
1.0055	0.1049	0.1043
1.0079	0.1260	0.1250
1.0108	0.1471	0.1456
1.0141	0.1683	0.1660
1.0178	0.1896	0.1863
1.0220	0.2110	0.2064
1.0267	0.2324	0.2264
1.0317	0.2540	0.2462
1.0373	0.2756	0.2657
1.0433	0.2974	0.2851
1.0498	0.3194	0.3042
1.0567	0.3414	0.3231
1.0641	0.3636	0.3417
1.0719	0.3860	0.3601
1.0802	0.4085	0.3782
1.0890	0.4312	0.3960
1.0983	0.4541	0.4135
1.1080	0.4772	0.4307
1.1183	0.5006	0.4476
1.1290	0.5241	0.4642
1.1402	0.5479	0.4805

*Ejercicio 14. Obtener 50 puntos  $(x_i, y_i)$  de aproximación a la función  $y=\text{Senh}(x)$  para valores de  $x$  igualmente espaciados, interpolándolos entre 20 valores de  $(x, y)$  dados por  $y=\text{Senh}(x)$  para valores de  $x$  uniformemente espaciados en el intervalo  $(0, 2\pi)$ , utilizando el método de interpolación basado en la transformada rápida de Fourier (FFT). Representar los puntos gráficamente.*

En primer lugar, definimos los 20 valores de  $x$  igualmente espaciados entre 0 y  $2\pi$ .

```
>> x=(0:pi/10:2*pi);
```

Ahora hallamos los 30 puntos de interpolación  $(x, y)$  pedidos.

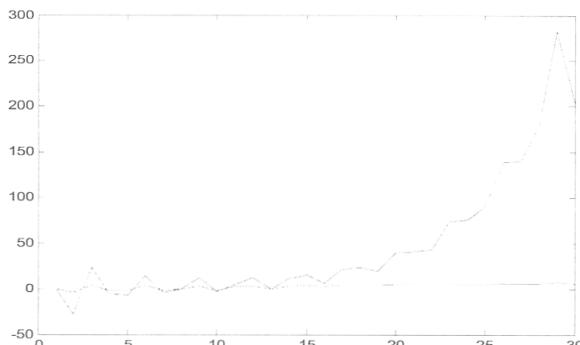
```
>> y=interpft(sinh(x), 30);
```

```
>> puntos=[y', (asinh(y))']
```

puntos =

-0.0000	-0.0000
-28.2506	-4.0346
23.3719	3.8451
-4.9711	-2.3067
-7.7918	-2.7503
14.0406	3.3364
-4.8129	-2.2751
-0.8717	-0.7877
11.5537	3.1420
-3.3804	-1.9323
4.4531	2.1991
11.8616	3.1682
-0.2121	-0.2105
10.9811	3.0914
15.1648	3.4132
6.1408	2.5147
21.2540	3.7502
23.3792	3.8455
18.5918	3.6166
39.4061	4.3672
40.6473	4.3982
42.8049	4.4499
73.2876	4.9876
74.8962	5.0093
89.7159	5.1898
139.0371	5.6279
139.3869	5.6304
180.2289	5.8874
282.4798	6.3368
201.7871	6.0004

```
>> plot(puntos)
```



*Ejercicio 15. Hallar el polinomio de grado 3 que mejor ajusta la nube de puntos  $(i, i^2)$  para  $1 \leq i \leq 7$ , en el sentido de mínimos cuadrados. Hallar su valor para  $x=10$  y representar gráficamente la curva de ajuste.*

```
>> x=(1:7);y=[1,4,9,15,25,36,49];p=vpa(poly2sym(polyfit(x,y,2)))
```

```
P =
x^2 - 0.000000000000009169181662272871686413366801652*x +
0.00000000000020761891472015924526365781895317
```

Ahora calculamos el valor numérico del polinomio  $p$  en  $x=10$ .

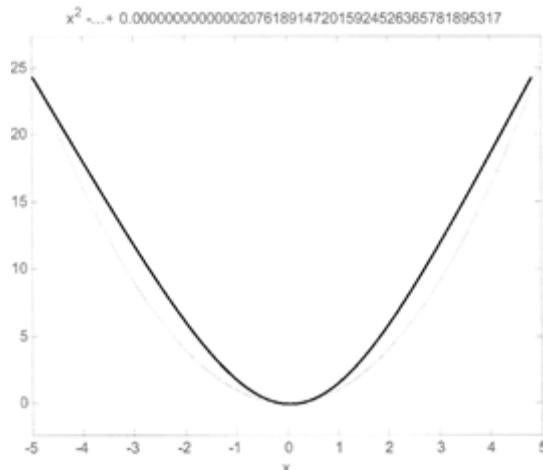
```
>> subs(p,10)
```

```
ans =
```

```
100.0000
```

A continuación se representa la gráfica de ajuste:

```
>> ezplot(p, [-5,5])
```



*Ejercicio 16. Hallar soluciones para las siguientes ecuaciones:*

$\operatorname{Sen}(x)\operatorname{Cos}(x)=0$ ,  $\operatorname{Sen}(x)=a\operatorname{Cos}(x)$ ,  $ax^2+bx+c=0$  y  $\operatorname{Sen}(x)+\operatorname{Cos}(x)=\sqrt{3}/2$

```
>> solve('sin(x)*cos(x)=0'
```

```
ans =
```

```
[      0]
[ 1/2*pi]
[-1/2*pi]
```

```
>> solve('sin(x)=a*cos(x)', 'x')
ans =
atan(a)

>> solve ('a*x^2+b*x+c=0', 'x')
ans =
[1/2/a*(-b+(b^2-4*a*c)^(1/2))] 
[1/2/a*(-b-(b^2-4*a*c)^(1/2))]

>> solve('sen(x)+cos(x)=sqrt(3)/2')
ans =
[1/2*3^(1/2)]
[1/2*3^(1/2)]
```

*Ejercicio 17. Hallar al menos dos soluciones para cada una de las dos ecuaciones trigonométrica y exponencial siguientes:*

$$x \operatorname{Sen}(x) = 1/2 \text{ y } 2^{x^3} = 4(2^{3x})$$

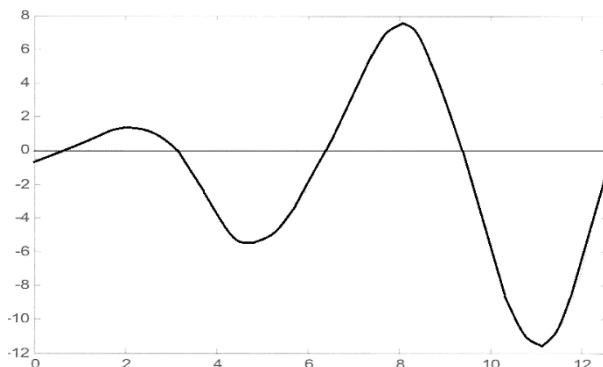
Inicialmente, utilizamos el comando *solve*:

```
>> vpa(solve('x*sin(x)=1/2', 'x'))
ans =
matrix([[-226.19688152398440474751335389781]])

>> vpa(solve('2^(x-3)=4*2^(3*x)', 'x'))
ans =
2.0
-1.0
-1.0
```

Para la primera ecuación no obtenemos soluciones, pero para la segunda sí. Para analizar mejor la primera ecuación, realizamos una representación gráfica para ver de forma aproximada los intervalos donde caen las posibles soluciones.

```
>> fplot ('[x*sin(x)-1/2,0]', [0,4*pi])
```



Se observa que hay una solución entre 0 y 2, otra entre 2 y 4, otra entre 4 y 8, y así sucesivamente. Vamos a calcular tres de ellas con el comando *fzero*.

```
>> s1=fzero('x*sin(x)-1/2',2)
s1 =
0.7408

>> s2=fzero('x*sin(x)-1/2',4)
s2 =
2.9726

>> s3=fzero('x*sin(x)-1/2',6)
s3 =
6.3619
```

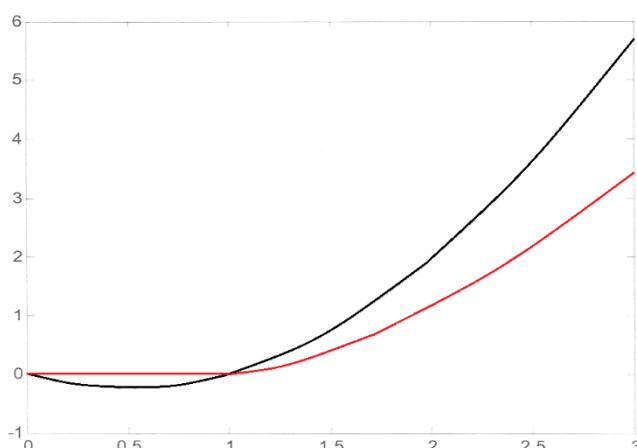
*Ejercicio 18.* Resolver cada una de las dos ecuaciones logarítmica e irracional siguientes:

$$x^{3/2} \log(x) = x \log(x^{3/2}), \sqrt{1-x} + \sqrt{1+x} = a$$

```
>> vpa(solve('x^(3/2)*log(x)=x*log(x)^(3/2)'))
ans =
1.0
0.318131505204764135312 55425158766 - 1.3372357014306894089011621431937*i
```

A continuación, realizamos la representación gráfica para poder intuir los intervalos en los que se encuentran las posibles soluciones. Se comprueba que  $x=1$  es la única solución real.

```
>> fplot('x^(3/2)*log(x),x*log(x)^(3/2)', [0,3,-1,6])
```



Ahora, vamos a resolver la ecuación irracional:

```
>> pretty(sym(solve('sqrt(1-x)+sqrt(1+x)=a' , 'x')))

+-      2 1/2      +-+
| a (4 - a )      |
| -----      |
|      2      |
| a (4 - a )      |
| - -----      |
|      2      +-
```

*Ejercicio 19. Resolver el sistema de dos ecuaciones siguiente:*

$$\begin{aligned} \cos(x/12)/\exp(x^2/16) &= y \\ -5/4 + y &= \sin(x^{3/2}) \end{aligned}$$

```
>> [x,y]=solve ('cos(x/12)/exp(x^2/16)=y',' -5/4+y=sin (x^(3/2))')
```

x =

0.34569744170126319331033283636228\*i - 0.18864189802267887925036526820236

y =

0.0086520715192230549621145978569268\*i + 1.0055146234480859930589058368368

*Ejercicio 20. Hallar la intersección de las hipérbolas de ecuaciones  $x^2-y^2=r^2$  y  $a^2x^2-b^2y^2=a^2b^2$  con la parábola  $z^2=2px$ .*

```
>>[x,y,z]=solve('a^2*x^2-b^2*y^2=a^2*b^2','x^2-y^2=r^2','z^2=2*p*x','x,y,z')
```

x =

```
((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/2)/(2*p)
((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/2)/(2*p)
((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/2)/(2*p)
- ((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/2)/(2*p)
((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/2)/(2*p)
- ((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/2)/(2*p)
- ((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/2)/(2*p)
- ((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/2)/(2*p)
```

y =

```
a*((b^2 - r^2)/(a^2 - b^2))^(1/2)
-a*((b^2 - r^2)/(a^2 - b^2))^(1/2)
a*((b^2 - r^2)/(a^2 - b^2))^(1/2)
a*((b^2 - r^2)/(a^2 - b^2))^(1/2)
-a*((b^2 - r^2)/(a^2 - b^2))^(1/2)
a*((b^2 - r^2)/(a^2 - b^2))^(1/2)
-a*((b^2 - r^2)/(a^2 - b^2))^(1/2)
```

```

z =
((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/4)
((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/4)
- ((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/4)
((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/4)*i
- ((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/4)*i
((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/4)*1
- ((4*a^2*b^2*p^2 - 4*b^2*p^2*r^2) / (a^2 - b^2))^(1/4)*1

```

*Ejercicio 21. Estudiar y resolver el sistema:*

$$\begin{aligned}x1 - x2 + x3 &= 1 \\4x1 + 5x2 - 5x3 &= 4 \\2x1 + x2 - x3 &= 2 \\x1 + 2x2 - 2x3 &= 1\end{aligned}$$

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

A =

$$\begin{matrix} 1 & -1 & 1 \\ 4 & 5 & -5 \\ 2 & 1 & -1 \\ 1 & 2 & -2 \end{matrix}$$

```
>>B=[1,-1,1,1;4,5,-5,4,-2,1,-1,2;1,2,-2,1]
```

B =

$$\begin{matrix} 1 & -1 & 1 & 1 \\ 4 & 5 & -5 & 4 \\ 2 & 1 & -1 & 2 \\ 1 & 2 & -2 & 1 \end{matrix}$$

```
>> [rank(A), rank(B)]
```

ans =

$$\begin{matrix} 2 & 2 \end{matrix}$$

Vemos que el rango de  $A$  y el de  $B$  coinciden y su valor es 2, que es menor que el número de incógnitas del sistema (3). Por lo tanto, el sistema tendrá infinitas soluciones (compatible indeterminado). Intentamos resolverlo con el comando *solve*:

```
>>[x1,x2,x3]=solve('x1-x2+x3=1', '4*x1 + 5*x2-5*x3=4' , '2*x1+x2-x3=2',
'x1+2*x2-2*x3=1', 'x1' , 'x2' , 'x3')
Warning: 4 equations in 3 variables.
```

x1 =

1

```
x2 =
```

```
z
```

```
x3 =
```

```
z
```

Las infinitas soluciones se obtienen al dar al parámetro  $x3$  los infinitos valores reales ( $\{I, z, z\} \quad \forall z \in R$ ). Obsérvese que la solución trivial  $\{1, 0, 0\}$  se obtiene para el valor del parámetro igual a cero.

*Ejercicio 22. Estudiar y resolver el sistema:*

$$\begin{aligned}x + 2y + 3z + t &= 6 \\x + 3y + 8z + t &= 19\end{aligned}$$

```
>> A=[1,2,3,1;1,3,8,1]
```

```
A =
```

$$\begin{matrix} 1 & 2 & 3 & 1 \\ 1 & 3 & 8 & 1 \end{matrix}$$

```
>> B=[1,2,3,1,6;1,3,8,1,19]
```

```
B =
```

$$\begin{matrix} 1 & 2 & 3 & 1 & 6 \\ 1 & 3 & 8 & 1 & 19 \end{matrix}$$

```
>> [rank(A), rank(B)]
```

```
ans =
```

$$\begin{matrix} 2 & 2 \end{matrix}$$

Vemos que el rango de  $A$  y el de  $B$  coinciden y su valor es 2, que es menor que el número de incógnitas del sistema (4). Por lo tanto, el sistema tiene infinitas soluciones (compatible indeterminado). Intentamos resolverlo:

```
>> [x,y,z,t]=solve('x+2*y+3*z+t=6','x+3*y+8*z+t=19','x','y','z','t')
Warning: 2 equations in 4 variables. New variables might be introduced.
```

```
x =
```

$$7z_1 - z_2 - 20$$

```
y =
```

$$z_2$$

```

z =
13 - 5*z1

```

```

t =
z1

```

Esta vez la solución depende de dos parámetros. Al hacer variar y / en los números reales, se obtienen las infinitas soluciones del sistema. Dichas soluciones forman un subespacio vectorial de dimensión dos del espacio vectorial real bidimensional que puede expresarse como sigue:

$$\{7z_1 - z_2 - 20, z_2, 13 - 5z_1, z_1\} \quad \forall z_1, z_2 \in R$$

*Ejercicio 23. Estudiar y resolver el sistema:*

$$\begin{aligned}
3x_1 + x_2 + x_3 - x_4 &= 0 \\
2x_1 + x_2 - x_3 + x_4 &= 0 \\
x_1 + 2x_2 + 4x_3 + 2x_4 &= 0 \\
2x_1 + x_2 - 2x_3 - x_4 &= 0
\end{aligned}$$

```

>> det([3,1,1,-1;2,1,-1,1,-1,2, 4, 2; 2,1,-
2,-1])

```

-30

Como el determinante de la matriz de los coeficientes es no nulo, el sistema sólo tiene la solución trivial:

```
>> [x1,x2,x3,x4]=solve('3*x1+x2+x3-x4=0','2*x1+x2-x3+x4=0','x1+2*x2-
4*x3-2*x4=0' , 'x1-x2-3*x3-5*x4=0','x1','x2','x3','x4')
```

```
x1 =

```

```
0

```

```
x2 =

```

```
0

```

```
x3 =

```

```
0

```

```
x4 =

```

```
0

```

*Ejercicio 24. Estudiar y resolver, según los valores de  $m$ , el sistema:*

$$\begin{aligned} mx + y + z &= 1 \\ x + my + z &= m \\ x + y + mz &= m^2 \end{aligned}$$

```
>> syms m
>> A= [m, 1,1; 1, m, 1; 1,
A =
[m, 1, 1]
[1, m, 1]
[1, 1, m]
>> det(A)
ans =
m^3 - 3*m + 2
>> solve('m^3 - 3*m + 2=0','m')
ans =
-2
1
1
```

Los valores de  $m$  que determinan el rango de la matriz son -2 y 1.

Consideramos ahora la matriz ampliada como una función de  $m$ :

```
>> B=[m,1,1,1;1,m,1,m;1,1,m,m^2]
B =
[m, 1, 1, 1]
[1, m, 1, m]
[1, 1, m, m^2]
```

Vamos a estudiar el caso  $m = -2$ :

```
>> rank(subs(A,{m},{-2}))
ans =
2
>> rank(subs(B,{m},{-2}))
ans =
3
```

Vemos que el rango de las dos matrices es distinto, con lo que el sistema es incompatible (no tiene solución) para  $m=-2$ .

Ahora estudiamos el caso  $m=1$ :

```
>> rank(subs(A,{m},{1}))
```

```
ans =
```

```
1
```

```
>> rank(subs(B,{m},{1}))
```

```
ans =
```

```
1
```

Ahora el rango de las dos matrices es 1, que es menor que el número de incógnitas. Luego, el sistema tiene infinitas soluciones (compatible indeterminado). Las hallamos sustituyendo  $m=1$  en el sistema inicial:

```
>> [x,y,z]=solve('x+y+z=1','x','y','z')
```

```
Warning: 1 equations in 3 variables. New variables might be introduced.
```

```
x =
```

```
1 - z2 - z1
```

```
y =
```

```
z2
```

```
z =
```

```
z1
```

Luego, las infinitas soluciones se obtienen al hacer variar los parámetros  $y$  y  $z$  en los números reales. El subespacio de soluciones de dimensión 2 es:

$$\{1-z2-z1, z2, z1\} \quad \forall z1, z2 \in \mathbb{R}$$

Si consideramos el caso en que  $m$  es distinto de -2 y de -1, el sistema tiene solución única, que nos viene dada por el comando *solve*:

```
>> [x,y,z]=solve('m*x+y+z=1','x+m*y+z=m','x+y+m*z=m^2','x','y','z')
```

```
x =
```

$$-(m + 1) / (m + 2)$$

```

y =
1/(m + 2)

z =
(m^2 + 2*m + 1)/(m + 2)

```

*Ejercicio 25. Estudiar y resolver, según los valores de m, el sistema:*

$$\begin{aligned} my &= m \\ (1+m)x - z &= m \\ y + z &= m \end{aligned}$$

```

>> syms m
>> A=[0,m,0;m+1,0,-1;0,1,1]
A =
[      0,  m,  0]
[m + 1,  0, -1]
[      0,  1,  1]
>> det(A)
ans =
-m^2 - m
>> solve('-m^2 - m=0','m')
ans =
-1
0

```

Vemos que los valores de  $m$  que determinan el rango de la matriz de los coeficientes del sistema son  $m=-1$  y  $m=0$ .

Consideramos ahora la matriz ampliada:

```

>> B=[0,m,0,m;m+1,0,-1,m;0, 1, 1,m]
B =
[      0,  m,  0,  m]
[m + 1,  0, -1,  m]
[      0,  1,  1,  m]
>> rank(subs(A,{m},{-1}))

```

```
>> rank(subs(B,{m}, {-1}))
```

```
ans =
```

```
3
```

En el caso  $m = -1$ , vemos que el sistema no tiene solución (es incompatible) porque el rango de la matriz de los coeficientes del sistema es 2 y el de la matriz ampliada es 3.

Ahora, analizamos el caso  $m=0$ :

Cuando  $m$  es cero el sistema es homogéneo, ya que los términos independientes son todos nulos. Para hacer su estudio, analizamos el determinante de la matriz de los coeficientes del sistema.

```
>> det(subs(A,{m},{0}))
```

```
ans =
```

```
0
```

Al ser el determinante nulo, el sistema tiene infinitas soluciones:

```
>> [x,y,z]=solve('x-z=0','y+z=0','x','y','z')
```

```
Warning: 2 equations in 3 variables. New variables might be introduced.
```

```
x =
```

```
z1
```

```
y =
```

```
-z1
```

```
z =
```

```
z1
```

Luego, las infinitas soluciones vendrán dadas cuando el parámetro  $z$  varía en los números reales. El subespacio de soluciones de dimensión 1 es:

$$\{z1, -z1, z1\} \quad \forall z1 \in \mathbb{R}$$

Si  $m$  es distinto de 0 y distinto de -1, el sistema es compatible determinado (solución única), ya que los rangos de la matriz del sistema y de la matriz ampliada coinciden. Se calculará la solución mediante la función *solve*.

```
>> [x,y,z]=solve('m*y=m','(1+m)*x-z=m','y+z=m','x','y','z')
```

```
X =  
(2*m - 1) / (m + 1)
```

```
Y =  
1
```

```
Z =  
m - 1
```

*Ejercicio 26. Estudiar y resolver el sistema:*

$$\begin{aligned}2x + y + z + t &= 1 \\x + 2y + z + t &= 1 \\x + y + 2z + t &= 1 \\x + y + z + 2t &= 1\end{aligned}$$

```
>> A=[2,1,1,1;1,2,1,1;1,1,2,1;1,1,1,2];  
>> B=[2,1,1,1,1;1,2,1,1;1,1,2,1;1,1,1,2,1];  
>> [rank(A), rank(B)]
```

```
ans =  
4 4
```

```
>> b=[1,1,1,1]';
```

Vemos que las matrices  $A$  y  $B$  (ampliada) tienen rango 4, que, además, coincide con el número de incógnitas. Luego, el sistema tiene solución única (es compatible determinado *con el mismo número de ecuaciones y de incógnitas*). Para calcular su solución se utiliza cualquiera de los múltiples comandos vistos.

```
>> X = nnls(A,b)
```

```
X =  
0.2000  
0.2000  
0.2000  
0.2000
```

```
>> X = bicg(A,b)
```

```
bicg converged at iteration 1 to a solution with relative residual 0
```

```
x =  
0.2000  
0.2000  
0.2000  
0.2000
```

```
>> x = bicgstab(A,b)
bicgstab converged at iteration 0.5 to a solution with relative
residual 0

x =
0.2000
0.2000
0.2000
0.2000

>> X = pcg(A,b)
pcg converged at iteration 1 to a solution with relative residual 0

x =
0.2000
0.2000
0.2000
0.2000

>> gmres(A,b)
gmres converged at iteration 1 to a solution with relative residual 0

ans =
0.2000
0.2000
0.2000
0.2000

>> x = lsqr(A,b)
lsqr converged at iteration 2 to a solution with relative residual 0

x =
0.2000
0.2000
0.2000
0.2000

>> A\b'

ans =
0.2000
0.2000
0.2000
0.2000
```

# MATRICES, ESPACIOS VECTORIALES Y APLICACIONES LINEALES

---

En el capítulo dedicado a variables vectoriales y matriciales ya estudiamos las formas de representar vectores y matrices en MATLAB. No obstante, recordamos aquí la notación.

La matriz:

$$A = (A_{ij}) = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \cdots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \cdots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \cdots & a_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & a_{m3} & \cdots & a_{mn} \end{bmatrix} \quad i = 1, 2, \dots, m \quad j = 1, 2, \dots, n$$

se puede introducir en Matlab de las siguientes formas:

$A = [a_{11}, a_{12}, \dots, a_{1n}; a_{21}, a_{22}, \dots, a_{2n}; a_{m1}, a_{m2}, \dots, a_{mn}]$

$A = [a_{11}, a_{12}, \dots, a_{1n}; a_{21}, a_{22}, \dots, a_{2n}; a_{m1}, a_{m2}, \dots, a_{mn}]$

Por otro lado, el vector  $V = (v_1, v_2, \dots, v_n)$  se introduce como caso particular de matriz de una sola fila (matriz de dimensión  $1 \times n$ ) o de la siguiente forma:

$V = [v_1, v_2, \dots, v_n]$

$V = [v_1, v_2, \dots, v_n]$

## OPERACIONES CON MATRICES

MATLAB incorpora comandos que permiten realizar las operaciones más habituales con matrices, tanto numéricas como simbólicas. La tabla siguiente muestra los más importantes.

<b>A+B</b>	<p><i>Suma de las matrices A y B</i></p> <pre>&gt;&gt; A=[-1 7 2;i -i 8; 3 1 4] A = -1.0000      7.0000      2.0000   0 + 1.0000i      0 - 1.0000i      8.0000   3.0000      1.0000      4.0000 &gt;&gt; B=[1 2 3; 1+i 1-2i 1-3i; -3 -5 -i] B =   1.0000      2.0000      3.0000   1.0000 + 1.0000i      1.0000 - 2.0000i      1.0000 - 3.0000i  -3.0000      -5.0000      0 - 1.0000i &gt;&gt; A+B ans =   0          9.0000      5.0000   1.0000 + 2.0000i      1.0000 - 3.0000i      9.0000 - 3.0000i   0          -4.0000      4.0000 - 1.0000i</pre>
<b>A-B</b>	<p><i>Diferencia de las matrices A y B</i></p> <pre>&gt;&gt; A-B ans = -2.0000      5.0000      -1.0000 -1.0000      -1.0000 + 1.0000i      7.0000 + 3.0000i   6.0000      6.0000      4.0000 + 1.0000i</pre>
<b>A*B</b>	<p><i>Producto de las matrices A y B</i></p> <pre>&gt;&gt; A*B ans =   0          + 7.0000i      -5.0000 -14.0000i      4.0000 -23.0000i  -23.0000      -42.0000 + 1.0000i      -3.0000 - 6.0000i  -8.0000 + 1.0000i      -13.0000 - 2.0000i      10.0000 - 7.0000i</pre>
<b>c*A</b>	<p><i>Producto de un escalar por una matriz</i></p> <pre>&gt;&gt; i*A ans =   0          - 1.0000i      0 + 7.0000i      0 + 2.0000i  -1.0000      1.0000      0 + 8.0000i   0          + 3.0000i      0 + 1.0000i      0 + 4.0000i</pre>

<b>expm(A)</b>	$e^A$ calculada a través de autovalores
	<pre>&gt;&gt; expm(A)  ans =  1.0e+003 * 0.5710 + 0.0674i  0.7081 - 0.0337i  1.7019 - 0.0338i 0.5710 + 0.0674i  0.7081 - 0.0337i  1.7018 - 0.0337i 0.5710 + 0.0675i  0.7082 - 0.0337i  1.7018 - 0.0337i</pre>
<b>expml(A)</b>	$e^A$ calculada a través de aproximantes de Padé
	<pre>&gt; expml(A)  ans =  1.0e+003 * -0.0006          1.0956          0.0064 -0.0005 + 0.0008i -0.0005 - 0.0008i  2.9800 0.0191           0.0017          0.0536</pre>
<b>logm(A)</b>	Logaritmo neperiano de la matriz A
	<pre>&gt;&gt; logm(A)  ans =  1.3756 + 0.0063i  2.4201 + 0.4906i -1.7163 - 0.4969i -0.9903 + 0.1445i  1.7284 - 0.2954i  1.3413 + 0.1509i 0.6464 - 0.0412i -0.6375 - 0.1572i  2.0705 + 0.1984i</pre>
<b>sqrtm(A)</b>	Raíz cuadrada de la matriz cuadrada A
	<pre>&gt;&gt; sqrtm(A)  ans =  1.1538 - 0.0556i  2.4904 + 0.3474i -0.8158 - 0.2918i -0.6332 + 0.2331i  1.5192 - 0.3779i  1.9424 + 0.1449i 0.8187 - 0.0083i -0.2678 - 0.0893i  2.2775 + 0.0975i</pre>
<b>funm(A, 'función')</b>	Aplica la función a la matriz cuadrada A. Las funciones que se pueden utilizar son EXP, LOG, COS, SIN, COSH, SINH, SQRTM y EXPM
	<pre>&gt;&gt; funm(A, 'log')  ans =  1.3756 + 0.0063i  2.4201 + 0.4906i -1.7163 - 0.4969i -0.9903 + 0.1445i  1.7284 - 0.2954i  1.3413 + 0.1509i 0.6464 - 0.0412i -0.6375 - 0.1572i  2.0705 + 0.1984i  &gt;&gt; funm(A, 'sinh')  ans =  1.0e+002 *  2.8840 + 0.3729i  3.4704 - 0.1953i  8.5505 - 0.1776i 2.8821 + 0.3256i  3.5574 - 0.1265i  8.4652 - 0.1992i 2.8359 + 0.3291i  3.5556 - 0.1738i  8.5133 - 0.1553i  &gt;&gt; funm(A, 'cos')</pre>

	<pre>ans = -22.0565 + 8.2759i 20.1371 -22.2895i 1.7740 +14.0136i 2.0152 +10.6436i -22.5464 + 2.7537i 20.3857 -13.3973i 6.7526 - 6.1638i 1.5253 + 5.1214i -8.4234 + 1.0424i</pre>
<b>transpose(A) o A'</b>	<p><i>Matriz transpuesta de A</i></p> <pre>&gt;&gt; transpose (A)  ans = -1.0000      0 + 1.0000i      3.0000 7.0000      0 - 1.0000i      1.0000 2.0000      8.0000          4.0000</pre>
<b>inv(A)</b>	<p><i>Matriz inversa de la matriz cuadrada A (<math>A^{-1}</math>)</i></p> <pre>&gt;&gt; Inv (A) ans = -0.0430 - 0.0266i  -0.1465 - 0.0133i  0.3145 + 0.0400i 0.1373 - 0.0102i  -0.0564 - 0.0051i  0.0441 + 0.0154i -0.0020 + 0.0225i  0.1240 + 0.0113i  0.0031 - 0.0338i  &gt;&gt; A*inv (A) ??? undefined function or method 'INV' for Input arguments of type 'double'.</pre> <p><i>Matriz inversa de la matriz cuadrada A (<math>A^{-1}</math>)</i></p> <pre>&gt;&gt; A*inv (A)  ans = 1.0000 + 0.0000i      0 + 0.0000i      -0.0000 0           1.0000 + 0.0000i      0.0000 + 0.0000i 0.0000 + 0.0000i      0.0000 + 0.0000i      1.0000</pre>
<b>det(A)</b>	<p><i>Determinante de la matriz cuadrada A</i></p> <pre>&gt;&gt; det (A) ans = 1.7600e+002 -1.6000e+001i</pre>
<b>rank(A)</b>	<p><i>Rango de la matriz A</i></p> <pre>&gt;&gt; rank (A) ans = 3</pre>
<b>trace(A)</b>	<p><i>Suma de los elementos de la diagonal de A</i></p> <pre>&gt;&gt; trace (A) ans = 3.0000 - 1.0000i</pre>
<b>svd(A)</b>	<p><i>Vector V de valores singulares de A, que son las raíces cuadradas de los autovalores de la matriz simétrica <math>A^T A</math></i></p>

	<pre>&gt;&gt; svd(A)  ans =  9.7269 5.6228 2.7434</pre>
[U,S,V]=svd(A)	<p><i>Da la matriz diagonal S de valores singulares de A (ordenados de mayor a menor) y las matrices U y V tales que</i></p> $A=U*S*V'$ <pre>&gt;&gt; [U, S, V]=svd(A)  U = -0.1891 - 0.4193i -0.6966 - 0.5354i -0.0595 + 0.1139i -0.5350 - 0.5583i 0.3725 + 0.1937i -0.4752 - 0.0029i -0.2989 - 0.3181i 0.2210 - 0.0543i 0.8621 - 0.1204i  S = 9.7269 0 0 0 6.6228 0 0 0 2.7434  V = -0.1301 0.2346 0.9634 -0.1094 - 0.3895i -0.7321 - 0.5179i 0.1635 + 0.0735i -0.6018 - 0.6762i 0.3731 + 0.0396i -0.1721 - 0.1010i</pre>
rcond(A)	<p><i>Recíproco de la condición de la matriz A</i></p> <pre>&gt;&gt; rcond(A)  ans = 0.1796</pre>
norm(A)	<p><i>Norma de A o 2-norma (mayor valor singular de la matriz A)</i></p> <pre>&gt;&gt; norm(A)  ans = 9.7269</pre>
norm(A,1)	<p><i>1-norma de A (mayor suma de las columnas de A)</i></p> <pre>&gt;&gt; norm(A,1)  ans = 14</pre>
norm(A,inf)	<p><i>Norma infinita de A (mayor suma de las filas de A)</i></p> <pre>&gt;&gt; norm(A,inf)  ans = 10</pre>

<b>norm(A,'fro')</b>	<i>F-norma de A, definida por <math>\sqrt{\text{sum}(\text{diag}(A \cdot A))}</math></i>
	<pre>&gt;&gt; norm(A,'fro') ans =     12.0830</pre>
<b>cond(A) o cond(A,2)</b>	<i>Da la condición de la matriz A (cociente entre el mayor y el menor valor singular de A). Condición según la 2-norma</i>
<b>cond(A,1)</b>	<i>Condición de A según la 1-norma</i>
<b>cond(A,inf)</b>	<i>Condición de A según la norma infinita</i>
<b>cond(A,fro)</b>	<i>Condición de A según la F-norma (norma de Frobenius)</i>
	<pre>&gt;&gt; cond(A) ans =     3.5456 &gt;&gt; cond(A,1) ans =     5.5676 &gt;&gt; cond(A,inf) ans =     5.1481 &gt;&gt; cond(A,'fro') ans =     4.9266</pre>
<b>Z=null(A)</b>	<i>Da una base ortonormal del núcleo de A (<math>Z'Z=I</math>). El número de columnas de Z es la nulidad de A</i>
	<pre>&gt;&gt; null ([1 2 3;4 5 6; 7 8 9]) ans = -0.4082   0.8165  -0.4082</pre>
<b>Z=null(A, 'r')</b>	<i>Da una base racional del núcleo de A</i>
	<pre>&gt;&gt; null([1 2 3;4 5 6;7 8 9], 'r') ans =   1  -2   1</pre>
<b>Q=orth(A)</b> <b>(<math>Q'Q=1</math>).</b>	<i>Da una base ortonormal para el rango de A. Las columnas de Q generan el mismo espacio que las columnas de A, y el número de columnas de Q es el rango de A</i>

	<pre>&gt;&gt; orth([1 2 3;4 5 6;7 8 9])  ans =  -0.2148 0.8872</pre>
<b>subspace(A,B)</b>	<p>Da el ángulo entre los subespacios especificados por las columnas de A y de B. Si A y B son vectores, da el ángulo formado por ambos</p> <pre>&gt;&gt; subspace(A,B)  ans =  1.1728e-015</pre>
<b>rref(A)</b>	<p>Da la matriz reducida escalonada de Gauss-Jordan por filas de A. El número de filas no nulas de rref(A) es el rango de A</p> <pre>&gt;&gt; rref ([1 2 3;4 5 6;7 8 9])  ans =  1 0 -1 0 1 2 0 0 0</pre>
<b>A^p</b>	<p>Matriz A elevada a la potencia escalar p</p> <pre>&gt;&gt; A^3  ans =  1.0e+002 * 1.8600 - 0.1200i 1.0400 + 0.5400i 2.2200 - 0.4200i 0.6400 - 0.1000i 2.1400 - 0.2000i 2.3400 + 0.3000i 0.8200 + 0.2400i 0.9200 - 0.1800i 3.3800 - 0.0600i</pre>
<b>p^A</b>	<p>Escalar p elevado a la matriz A</p> <pre>&gt;&gt; 3^A  ans =  1.0e+003 * 1.2568+ 0.1484i 1.5585 - 0.0742i 3.7457 - 0.0742i 1.2568+ 0.1484i 1.5586 - 0.0742i 3.7455 - 0.0742i 1.2568+ 0.1485i 1.5586 - 0.0742i 3.7456 - 0.0742i</pre>
<b>diag(V,k)</b>	<p>Construye una matriz cuadrada diagonal de orden <math>n+ k </math> con los n elementos del vector V en la késima diagonal. Si <math>k=0</math> la diagonal es la principal, si <math>k&gt;0</math> la diagonal está por encima de la principal k unidades. Si <math>k&lt;0</math> la diagonal está por debajo de la principal k unidades. Además <math>\text{diag}(V,0)=\text{diag}(V)</math></p> <pre>&gt;&gt; diag([1 2 3 4 5 6])  ans =</pre>

	<pre> 1      0      0      0      0      0 0      2      0      0      0      0 0      0      3      0      0      0 0      0      0      4      0      0 0      0      0      0      5      0 0      0      0      Á0      0      6  &gt;&gt; diag([1 2 3 4 5 6],1)  ans = 0      1      0      0      0      0      0      0 0      0      2      0      0      0      0      0 0      0      0      3      0      0      0      0 0      0      0      0      4      0      0      0 0      0      0      0      0      5      0      0 0      0      0      0      0      0      6      0 0      0      Á0      0      0      0      0      0  &gt;&gt; diag([1 2 3 4 5 6],-2)  ans = 0      0      0      0      0      0      0      0 0      0      0      0      0      0      0      0 1      0      0      0      0      0      0      0 0      2      0      0      0      0      0      0 0      0      3      0      0      0      0      0 0      0      0      4      0      0      0      0 </pre>
<b>triu(A,k)</b>	<p>Construye una matriz triangular superior con los elementos de A que están por encima de su késima diagonal. Si k=0 la diagonal es la principal, si k&gt;0 la diagonal está por encima de la principal k unidades. Si k&lt;0 la diagonal está por debajo de la principal k unidades. Además triu(A,0)=triu(V)</p> <pre> &gt;&gt; triu([1 2 3 ;4 5 6; 7 8 9],1)  ans = 0      2      3 0      0      6 0      0      0  &gt;&gt; triu( [1 2 3 ; 4 5 6; 7 8 9],1)  ans = 0      2      3 0      0      6 0      0      0  &gt;&gt; triu([1 2 3 ;4 5 6; 7 8 9],-1)  ans = 0      2      3 4      5      6 0      8      9 </pre>

<b>tril(A,k)</b>	<p>Construye una matriz triangular inferior con los elementos de A que están por debajo de su késima diagonal. Si k=0 la diagonal es la principal, si k&gt;0 la diagonal está por encima de la principal k unidades. Si k&lt;0 la diagonal está por debajo de la principal k unidades. Además <math>\text{tril}(A,0)=\text{tril}(V)</math></p> <pre>&gt;&gt; tril([1 2 3 ;4 5 6; 7 8 9])  ans =  1     0     0 4     5     0 7     8     9  &gt;&gt; tril([1 2 3 ; 4 5 6; 7 8 9],1)  ans =  1     2     0 4     5     6 7     8     9  &gt;&gt; tril([1 2 3 ;4 5 6; 7 8 9],-1)  ans =  0     0     0 4     0     0 7     8     0</pre>
<b>A=sym('f1;f2,...,fm')</b>	<p>Define la matriz simbólica mxn de filas f1 a fn donde <math>f_i=[a_{i1},a_{i2},...,a_{in}]</math>. Las operaciones y funciones para matrices simbólicas son similares a las ya vistas para matrices numéricas.</p> <pre>&gt;&gt; A=sym(' [a,a+1,a+2;a,a-1,a-2;1,2,3] ') A = [ a, a + 1, a + 2] [ a, a - 1, a - 2] [ 1,      2,      3] &gt;&gt; B=sym(' [b,b+1,b+2;b,b-1,b-2;1,2,3] ') B = [ b, b + 1, b + 2] [ b, b - 1, b - 2] [ 1,      2,      3]  &gt;&gt; A+B ans = [a + b, a + b + 2, a + b + 4] [a + b, a + b - 2, a + b - 4] [ 2,          4,          6]  &gt;&gt; det (A)</pre>

	<pre> ans = 0 &gt;&gt; rank(A) ans = 2 &gt;&gt; trace(A) ans = 2*a + 2 &gt;&gt; triu(A, 2) ans = [ 0,     0, a + 2] [ 0,     0,     0] [ 0,     0,     0] </pre>
<b>dot(A,B)</b>	<p><i>Producto escalar de los vectores A y B</i></p> <pre> &gt;&gt; a = [1 2 3] ; b = [4 5 6] ; c = dot(a,b) c = 32 </pre>
<b>cross(A,B)</b>	<p><i>Producto vectorial de los vectores A y B</i></p> <pre> &gt;&gt; a = [1 2 3] ; b = [4 5 6] ; c = cross(a,b) c = -3      6      -3 </pre>

## DESCOMPOSICIÓN DE MATRICES. VALORES PROPIOS Y VECTORES PROPIOS

MATLAB permite trabajar con especial cuidado en el campo de la descomposición matricial. Implementa comandos para la mayoría de las descomposiciones conocidas y para el trabajo con valores y vectores propios. La sintaxis de los comandos más habituales en este campo se presenta en la tabla siguiente.

<b>eig(A)</b>	<p><i>Halla los autovalores de la matriz cuadrada A</i></p> <pre> &gt;&gt; A=rand(4)  A = 0.8147    0.6324    0.9575    0.9572 0.9058    0.0975    0.9649    0.4854 0.1270    0.2785    0.1576    0.8003 0.9134    0.5469    0.9706    0.1419 </pre>
---------------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

	<pre> &gt;&gt; B=sym('[1, a, a^2;1, a, -a;0, 0, 1]') B = [ 1,     a, a-'2] [ 1,     a,    -a] [ 0,     0,     1] &gt;&gt; eig(A) ans = 2.4021 -0.0346 -0.7158 -0.4400  &gt;&gt; eig(B) ans = 0 1 a + 1 </pre>
<b>[V,D] = eig(A)</b>	<p>Halla la matriz diagonal D de autovalores de A y una matriz V cuyas columnas son los autovectores correspondientes, cumpliéndose que <math>A*V=V*D</math></p> <pre> &gt;&gt; [V, D]=eig(A)  V = -0.6621   -0.7149    0.1745    0.1821 -0.4819   -0.0292   -0.6291   -0.9288 -0.2766    0.6972    0.5995    0.3178 -0.5029   -0.0438   -0.4630    0.0570  D = 2.4021         0         0         0 0   -0.0346         0         0 0         0   -0.7158         0 0         0         0   -0.4400  &gt;&gt; [V1,D1]=eig(B) V1 = [ -a, a^2, 1] [ 1, -a, 1] [ 0, 1, 0] </pre>
<b>eig(A,B)</b>	<p>Devuelve un vector con los autovalores generalizados de las matrices cuadradas A y B. Los autovalores generalizados de A y B son las raíces del polinomio en X <math>\det(\lambda^*C-A)</math></p> <pre> &gt;&gt; C=randn(4) </pre>

	<pre>c = -0.1241    0.6715    0.4889    0.2939 1.4897   -1.2075    1.0347   -0.7873 1.4090    0.7172    0.7269    0.8884 1.4172    1.6302   -0.3034   -1.1471  &gt;&gt; eig(A,C)  ans = 0.6654 + 0.6533i 0.6654 - 0.6533i 0.0259 0.1997</pre>
[V,D]=eig(A,B)	<p>Halla la matriz diagonal D de autovalores generalizado de A y B y una matriz V cuyas columnas son los autovectores correspondientes, cumpliéndose <math>A^*V=B^*V*D</math></p> <pre>&gt;&gt; [V2,D2]=eig(A,C)  V2 = 0.2665 - 0.0340i    0.2665 + 0.0340i    1.0000 0.4002 -0.3908 + 0.0213i   -0.3908 - 0.0213i    0.0953 -1.0000 -0.8463 - 0.1537i   -0.8463 + 0.1537i   -0.9489 -0.1449 0.6062 - 0.2931i    0.6062 + 0.2931i    0.0211 0.3193  D2 = 0.6654 + 0.6533i    0                  0 0 0                  0.6654 - 0.6533i    0 0 0                  0                  0.0259 0 0                  0                  0 0.1997</pre>
[AA, BB, Q, Z,V] = qz(A,B)	<p>Calcula las matrices triangulares superiores AA y BB y las matrices Q y Z tales que <math>Q^*A^*Z = AA</math> y <math>Q^*B^*Z = BB</math>, y da la matriz V de autovectores generalizados de A y B. Los autovalores generalizados son los elementos de la diagonal de AA y BB, de tal modo que se tiene la igualdad siguiente: <math>A^*V*diag(BB) = B^*V*diag(AA)</math></p> <pre>&gt;&gt; [AA, CC, Q, Z, V]= qz(A,C)  AA = 6430 - 0.6312i    0.9129 - 1.0466i  1.2477 - 0.7149i -0.1102 + 0.2576i 0                  0.9818 + 0.9639i -0.1338 + 0.8139i  0.2874 + 0.0652i 0                  0                  0.0482            0.0822 0                  0                  0                0.4374  CC = 9663    0.1979 + 0.6630i    1.5008 - 0.7681i   -0.2960 + 0.0079i 0        1.4756           -0.0309 + 1.0436i   -0.1974 - 0.3071i 0        0                  1.8638            0.1951 0        0                  0                2.1900  Q =</pre>

	<pre> 0.4727 - 0.0698i 0.3875 - 0.1637i 0.1251 + 0.3290i 0.4287 - 0.5358i 0.2342 + 0.4500i 0.0410 + 0.2938i 0.5545 + 0.3640i -0.4572 + 0.0751i -0.5656 -0.0713 0.6541 0.4972 0.4410 -0.8544 0.0909 0.2594 z = -0.1939 - 0.1152i -0.0800 + 0.0321i 0.8930 -0.3798 0.2939 + 0.1469i -0.0424 - 0.0841i 0.4423 0.8292 0.7032 + 0.1644i 0.4323 - 0.4410i 0.0188 -0.3110 -0.3690 - 0.4283i 0.7481 - 0.2067i -0.0812 0.2673 v = -0.2235 - 0.1328i 0.1328 + 0.2235i 1.0000 + 0.0000i -0.4002 - 0.00000i 0.3388 + 0.1694i -0.1694 - 0.3388i 0.0953 - 0.0000i 1.0000 + 0.0000i 0.8106 + 0.1894i -0.1894 - 0.8106i -0.9489 - 0.0000i 0.1449 + 0.00001 -0.4253 - 0.4937i 0.4937 + 0.4253i 0.0211 - 0.0000i -0.3193 + 0.0000i </pre>																																
<b>[T,B] = balance(A)</b>	<p>Encuentra una matriz de transformación <math>T</math> tal que <math>B = T \setminus A^* T</math> tiene autovalores aproximados a los de <math>A</math>. La matriz <math>B</math> se llama matriz balanceada de la matriz <math>A</math></p> <pre> &gt;&gt; [T,B2]=balance (A)  T = </pre> <table style="margin-left: 20px;"> <tr><td>1.0000</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1.0000</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0.5000</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>1.0000</td></tr> </table> <pre> B2 = </pre> <table style="margin-left: 20px;"> <tr><td>0.8147</td><td>0.6324</td><td>0.4788</td><td>0.9572</td></tr> <tr><td>0.9058</td><td>0.0975</td><td>0.4824</td><td>0.4854</td></tr> <tr><td>0.2540</td><td>0.5570</td><td>0.1576</td><td>1.6006</td></tr> <tr><td>0.9134</td><td>0.5469</td><td>0.4853</td><td>0.1419</td></tr> </table>	1.0000	0	0	0	0	1.0000	0	0	0	0	0.5000	0	0	0	0	1.0000	0.8147	0.6324	0.4788	0.9572	0.9058	0.0975	0.4824	0.4854	0.2540	0.5570	0.1576	1.6006	0.9134	0.5469	0.4853	0.1419
1.0000	0	0	0																														
0	1.0000	0	0																														
0	0	0.5000	0																														
0	0	0	1.0000																														
0.8147	0.6324	0.4788	0.9572																														
0.9058	0.0975	0.4824	0.4854																														
0.2540	0.5570	0.1576	1.6006																														
0.9134	0.5469	0.4853	0.1419																														
<b>balance(A)</b>	<p>Calcula la matriz <math>B</math> balanceada de la matriz <math>A</math>. Su uso esencial es aproximar los autovalores de <math>A</math> cuando son difíciles de calcular. Se tiene que <math>eig(A)=eig(balance(A))</math></p> <pre> &gt;&gt; balance (A)  ans = </pre> <table style="margin-left: 20px;"> <tr><td>0.8147</td><td>0.6324</td><td>0.4788</td><td>0.9572</td></tr> <tr><td>0.9058</td><td>0.0975</td><td>0.4824</td><td>0.4854</td></tr> <tr><td>0.2540</td><td>0.5570</td><td>0.1576</td><td>1.6006</td></tr> <tr><td>0.9134</td><td>0.5469</td><td>0.4853</td><td>0.1419</td></tr> </table>	0.8147	0.6324	0.4788	0.9572	0.9058	0.0975	0.4824	0.4854	0.2540	0.5570	0.1576	1.6006	0.9134	0.5469	0.4853	0.1419																
0.8147	0.6324	0.4788	0.9572																														
0.9058	0.0975	0.4824	0.4854																														
0.2540	0.5570	0.1576	1.6006																														
0.9134	0.5469	0.4853	0.1419																														
<b>[V,D] cdf2rdf(V,D)</b>	<p>Transforma la salida compleja <math>[V,D]</math> del comando <math>eig</math> a forma real. Cada autovalor complejo en la diagonal de <math>D</math> origina una submatriz <math>2x2</math> en la forma real de la matriz <math>D</math></p> <pre> &gt;&gt; [V,D]=cdf2rdf (A,C)  V = </pre> <table style="margin-left: 20px;"> <tr><td>0.8147</td><td>0.6324</td><td>0.9575</td><td>0.9572</td></tr> <tr><td>0.9058</td><td>0.0975</td><td>0.9649</td><td>0.4854</td></tr> <tr><td>0.1270</td><td>0.2785</td><td>0.1576</td><td>0.8003</td></tr> <tr><td>0.9134</td><td>0.5469</td><td>0.9706</td><td>0.1419</td></tr> </table> <pre> D = </pre> <table style="margin-left: 20px;"> <tr><td>-0.1241</td><td>0.6715</td><td>0.4889</td><td>0.2939</td></tr> <tr><td>1.4897</td><td>-1.2075</td><td>1.0347</td><td>-0.7873</td></tr> <tr><td>1.4090</td><td>0.7172</td><td>0.7269</td><td>0.8884</td></tr> <tr><td>1.4172</td><td>1.6302</td><td>-0.3034</td><td>-1.1471</td></tr> </table>	0.8147	0.6324	0.9575	0.9572	0.9058	0.0975	0.9649	0.4854	0.1270	0.2785	0.1576	0.8003	0.9134	0.5469	0.9706	0.1419	-0.1241	0.6715	0.4889	0.2939	1.4897	-1.2075	1.0347	-0.7873	1.4090	0.7172	0.7269	0.8884	1.4172	1.6302	-0.3034	-1.1471
0.8147	0.6324	0.9575	0.9572																														
0.9058	0.0975	0.9649	0.4854																														
0.1270	0.2785	0.1576	0.8003																														
0.9134	0.5469	0.9706	0.1419																														
-0.1241	0.6715	0.4889	0.2939																														
1.4897	-1.2075	1.0347	-0.7873																														
1.4090	0.7172	0.7269	0.8884																														
1.4172	1.6302	-0.3034	-1.1471																														

<b>[U,T] = schur(A)</b>	<p>Halla una matriz <math>T</math> y una matriz unitaria <math>U</math> tales que <math>A = U*T*U'</math> and <math>U'*U = eye(U)</math>. Si <math>A</math> es compleja, <math>T</math> es una matriz triangular superior con los autovalores de <math>A</math> en la diagonal. Si <math>A</math> es real, la matriz <math>T</math> tiene los autovalores de <math>A</math> en la diagonal, y los correspondientes autovalores complejos se corresponden con los bloques diagonales <math>2x2</math> de la matriz <math>T</math></p> <pre>&gt;&gt; [U,T] = schur(A)  D = -0.6621   -0.5327    0.5206   -0.0825 -0.4819    0.1301   -0.5815   -0.6424 -0.2766    0.8273    0.4848   -0.0637 -0.5029    0.1217   -0.3947    0.7593  T = 1   -0.8133   -0.6225   -0.1304 0   -0.0346   -0.1940    0.2110 0     0       -0.7158    0.3496 0     0         0      -0.4400</pre>
<b>schur(A)</b>	<p>Devuelve la matriz <math>T</math> del apartado anterior solamente</p> <pre>&gt;&gt; schur(A) ans = 1   -0.8133   -0.6225   -0.1304 0   -0.0346   -0.1940    0.2110 0     0       -0.7158    0.3496 0     0         0      -0.4400</pre>
<b>[U,T] = rsf2csf(U,T)</b>	<p>Convierte a compleja la salida <math>[U, T]</math> del comando <i>schur</i></p> <pre>&gt;&gt; [DI, TI] = rsf2csf{A, C) U1 = 0.0394 + 0.2165i  -0.8561 + 0.0945i  -0.9360 1.1092 0.5415 + 0.2407i  -0.7067 - 0.0479i  -0.8523 0.6459 -0.1702 + 0.0337i  -0.2203 + 0.0689i  -0.1028 0.8155 0.1937 + 0.2427i  -0.8861 + 0.0600i  -1.0560 0.3086  T1 = -0.8576 + 0.4577i  -1.7102 + 0.4507i  0.1154 + 0.1647i 0.9728 - 0.0995i 0           -0.8576 - 0.4577i  0.7230 + 0.1166i 0.0766 + 0.2499i 1.4090 -0.9437 1.4172 -0.9901</pre>
<b>[P,H] = hess(A)</b>	<p>Devuelve la matriz unitaria <math>P</math> y la matriz de Hessenberg <math>H</math> tales que <math>A = P*H*P'</math> y <math>P'*P = eye(size(P))</math></p> <pre>&gt;&gt; [P,H] = hess (A) P =</pre>

	<pre> 1.0000    0        0        0 0        -0.7007   0.1180   -0.7036 0        -0.0982   -0.9928   -0.0687 0        -0.7066   0.0210    0.7073 H = 0.8147   -1.2135  -0.8559   0.1663 -1.2926   0.8399   1.2997   0.0668 0        0.6987   -0.0233   -0.3394 0        0        0.0323   -0.4196 </pre>																																																
<b>[L,U] = lu(A)</b>	<p><i>Descomponer la matriz A en el producto A=L*U (descomposición LU de A), siendo U una matriz triangular superior y L una matriz pseudotriangular inferior (triangularizable mediante permutación)</i></p> <pre> &gt;&gt; [L, U] = lu(A)  L = </pre> <table style="margin-left: 20px;"> <tr><td>0.8920</td><td>-0.3250</td><td>1.0000</td><td>0</td></tr> <tr><td>0.9917</td><td>1.0000</td><td>0</td><td>0</td></tr> <tr><td>0.1390</td><td>-0.4552</td><td>0.2567</td><td>1.0000</td></tr> <tr><td>1.0000</td><td>0</td><td>0</td><td>0</td></tr> </table> <pre> U = </pre> <table style="margin-left: 20px;"> <tr><td>0.9134</td><td>0.5469</td><td>0.9706</td><td>0.1419</td></tr> <tr><td>0</td><td>-0.4448</td><td>0.0024</td><td>0.3447</td></tr> <tr><td>0</td><td>0</td><td>0.0925</td><td>0.9426</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0.6955</td></tr> </table>	0.8920	-0.3250	1.0000	0	0.9917	1.0000	0	0	0.1390	-0.4552	0.2567	1.0000	1.0000	0	0	0	0.9134	0.5469	0.9706	0.1419	0	-0.4448	0.0024	0.3447	0	0	0.0925	0.9426	0	0	0	0.6955																
0.8920	-0.3250	1.0000	0																																														
0.9917	1.0000	0	0																																														
0.1390	-0.4552	0.2567	1.0000																																														
1.0000	0	0	0																																														
0.9134	0.5469	0.9706	0.1419																																														
0	-0.4448	0.0024	0.3447																																														
0	0	0.0925	0.9426																																														
0	0	0	0.6955																																														
<b>[L,U,P]=lu(A)</b>	<p><i>Da una matriz triangular inferior L, una matriz triangular superior U y una matriz de permutación P tales que P*A=L*U</i></p> <pre> &gt;&gt; [L, U, P] = lu(A)  L = </pre> <table style="margin-left: 20px;"> <tr><td>1.0000</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0.9917</td><td>1.0000</td><td>0</td><td>0</td></tr> <tr><td>0.8920</td><td>-0.3250</td><td>1.0000</td><td>0</td></tr> <tr><td>0.1390</td><td>-0.4552</td><td>0.2567</td><td>1.0000</td></tr> </table> <pre> U = </pre> <table style="margin-left: 20px;"> <tr><td>0.9134</td><td>0.5469</td><td>0.9706</td><td>0.1419</td></tr> <tr><td>0</td><td>-0.4448</td><td>0.0024</td><td>0.3447</td></tr> <tr><td>0</td><td>0</td><td>0.0925</td><td>0.9426</td></tr> <tr><td>0</td><td>0</td><td>0</td><td>0.6955</td></tr> </table> <pre> P = </pre> <table style="margin-left: 20px;"> <tr><td>0</td><td>0</td><td>0</td><td>1</td></tr> <tr><td>0</td><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>0</td><td>1</td><td>0</td></tr> </table>	1.0000	0	0	0	0.9917	1.0000	0	0	0.8920	-0.3250	1.0000	0	0.1390	-0.4552	0.2567	1.0000	0.9134	0.5469	0.9706	0.1419	0	-0.4448	0.0024	0.3447	0	0	0.0925	0.9426	0	0	0	0.6955	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0
1.0000	0	0	0																																														
0.9917	1.0000	0	0																																														
0.8920	-0.3250	1.0000	0																																														
0.1390	-0.4552	0.2567	1.0000																																														
0.9134	0.5469	0.9706	0.1419																																														
0	-0.4448	0.0024	0.3447																																														
0	0	0.0925	0.9426																																														
0	0	0	0.6955																																														
0	0	0	1																																														
0	1	0	0																																														
1	0	0	0																																														
0	0	1	0																																														

<b>R = chol(A)</b>	Devuelve la matriz triangular superior R tal que $R^*R=A$ (descomposición de Cholesky de A), en caso de que A sea definida positiva. Si A no es definida positiva devuelve un error
<b>[Q,R] = qr(A)</b>	Devuelve la matriz triangular superior R de la misma dimensión que A, y la matriz ortogonal Q tales que $A=Q*R$ (descomposición QR de A). Esta descomposición puede aplicarse a matrices no cuadradas <pre>&gt;&gt; [Q,R] = qr(A)  Q = 0.5332    0.4892    0.6519   -0.2267 0.5928   -0.7162    0.1668    0.3284 0.0831    0.4507   -0.0991    0.8833 0.5978    0.2112   -0.7331   -0.2462  R = 1.5279    0.7451    1.6759    0.9494 0         0.4805    0.0534    0.5113 0         0         0.0580    0.5216 0         0         0         0.6143</pre>
<b>[Q,R,E] = qr(A)</b>	Devuelve la matriz triangular superior R de la misma dimensión que A, la matriz permutación E y la matriz ortogonal Q tales que $A * E = Q * R$ <pre>&gt;&gt; [Q,R,E] = qr(A)  Q = 0.5707    0.4307    0.2564   -0.6504 0.5751   -0.0867   -0.8029    0.1308 0.0939    0.7694    0.0861    0.6259 0.5785   -0.4636    0.5313    0.4100  R = 1.6777    0.9827    0.7595    1.5263 0         0.9201    0.2246   -0.0534 0         0         0.3983   -0.0222 0         0         0         0.0425  E = 0         0         0         1 0         0         1         0 1         0         0         0 0         1         0         0</pre>
<b>jordan(A)</b>  <b>[V,J]=jordan(A)</b>	Halla la matriz canónica de Jordán J de la matriz A (J tiene los autovalores de A en la diagonal) Halla la matriz canónica de Jordán J de la matriz A y la matriz de paso V cuyas columnas son los autovectores de A cumpliéndose que $V^*A*V=J$

	<pre>&gt;&gt; jordan(A)  ans =  -0.0346 + 0.0000i      0           0           0 0           2.4021 - 0.0000i     0           0 0           0           -0.4400 + 0.0000i     0 0           0           0           -0.7158 - 0.0000i  &gt;&gt; [V,J]=jordan(A)  V =  16.3153 + 0.0000i 1.3164 + 0.0000i 3.1933 + 0.0000i -0.3768 - 0.0000i 0.6653 + 0.0000i 0.9583 + 0.0000i -16.2888 -0.0000i 1.3588 + 0.0000i -15.9101 - 0.0000i 0.5499 - 0.0000i 5.5734 - 0.0000i -1.2947 - 0.0000i 1.0000          1.0000          1.0000          1.0000  J =  -0.0346 + 0.0000i      0           0           0 0           2.4021 - 0.0000i     0           0 0           0           -0.4400 + 0.0000i     0 0           0           0           -0.7158 - 0.0000i</pre>
<b>condeig(A)</b>	<p>Vector con los números de condición para los autovalores de A</p> <pre>&gt;&gt; condeig(A)  ans =  1.0733 1.1338 1.7360 1.6539</pre>
<b>[V,D,s]=condeig(A)</b>	<p>Equivale a <math>[V,D] = \text{eig}(A)</math> y <math>s = \text{condeig}(A)</math></p> <pre>&gt;&gt; [V, D, s]=condeig(A)  V = -0.6621      -0.7149      0.1745      0.1821 -0.4819      -0.0292      -0.6291      -0.9288 -0.2766       0.6972      0.5995      0.3178 -0.5029      -0.0438      -0.4630      0.0570  D = 2.4021          0           0           0 0      -0.0346          0           0 0           0      -0.7158          0 0           0           0      -0.4400  s = 1.0733 1.1338 1.7360 1.6539</pre>
<b>[U,V,X,C,S]=gsvd(A,B)</b>	<p>Da las matrices unitarias U y V, la matriz cuadrada X y las matrices diagonales no negativas C y S tales que <math>A = U*C*X'</math>, <math>B = V*S*X'</math> y <math>C'*C + S'*S = I</math>. A es <math>(m,p)</math>, B es <math>(n,p)</math>, U es <math>(M,M)</math>, V es <math>(n,n)</math>, X es <math>(p,q)</math> y <math>q=\min(m+n,p)</math></p>
<b>[U,V,X,C,S]=gsvd(A,B,0)</b>	<p>Resultado anterior cuando m o n <math>\geq p</math>. U y V tienen a lo sumo p columnas y C y S tienen a lo sumo p filas</p>

<b>sigma = gsvd(A,B)</b>	<i>Devuelve el vector de valores singulares generalizados</i> $\text{sqrt}(\text{diag}(C'*C))./\text{diag}(S'*S)$
<b>X = pinv(A)</b>	<i>Devuelve la matriz X (pseudoinversa de A), de la misma dimensión que A' tal que <math>A*X*A=A</math> y <math>X*A*X=X</math> siendo <math>A*X</math> y <math>X*A</math> matrices hermitianas</i> <pre>&gt;&gt; X = pinv(A)  X = -15.2997    3.0761   14.7235    9.6445 -0.2088   -1.8442    1.0366   1.8711 14.5694   -1.9337   -14.6497  -9.0413 -0.3690    0.5345    1.4378  -0.4008</pre>
<b>hess(A)</b>	<i>Devuelve la matriz de Hessenberg H</i> <pre>&gt;&gt; hess(A)  ans = 0.8147   -1.2135   -0.8559    0.1663 -1.2926    0.8399    1.2997    0.0668 0         0.6987   -0.0233   -0.3394 0         0         0.0323   -0.4196</pre>
<b>poly(A)</b>	<i>Devuelve el polinomio característico de la matriz A</i> <pre>&gt;&gt; poly(A)  ans = 1.0000   -1.2118   -2.5045   -0.8416   -0.0261</pre>
<b>poly(V)</b>	<i>Devuelve un vector cuyas componentes son los coeficientes del polinomio cuyas raíces son los elementos del vector V</i> <pre>&gt;&gt; poly([1 2 3 4 5 6])  ans = 1624    -1764      1      -21      175     -735           720</pre>
<b>vander(C)</b>	<i>Devuelve la matriz de Vandermonde A tal que su j-ésima columna es el vector C. Además <math>A(:,j) = C^{n-j}</math></i> <pre>&gt;&gt; vander([1 -1 0 6 3])  ans = 1       1       1       1       1 1      -1       1      -1       1 0       0       0       0       1 1296   216     36      6      1 81     27      9      3      1</pre>
<b>eigs(A)</b> <b>eigs(A,B)</b> <b>eigs(A,k)</b> <b>eigs(A,B,k)</b> <b>eigs(A,k,sigma)</b>	<i>Vector con los seis mayores autovalores de A</i> <i>Resuelve la ecuación <math>A*V == B*V*D</math> con B simétrica</i> <i>Vector con los k mayores autovalores de A</i> <i>Devuelve las k mayores soluciones de <math>A*V == B*V*D</math></i> <i>Vector con los k mayores autovalores de A basados en <math>\sigma</math>, donde <math>\sigma</math> puede valer 'lm' (mayor magnitud), 'sm' (menor magnitud), 'lr' (mayor parte real), 'sr' (menor parte real), 'li' (mayor parte imaginaria) y 'si' (menor parte imaginaria)</i>

<b>eigs(A,B,k, a)</b> <b>[V,D] = eigs(A,...)</b>	<p>Da las <math>k</math> mayores soluciones de <math>A^*V == B^*V*D</math> según <math>a</math>      Da la matriz diagonal <math>D</math> de autovalores de <math>A</math> y la matriz <math>V</math>      cuyas columnas son los respectivos autovectores</p> <pre>&gt;&gt; [V, D] = eigs (A)  V =  -0.6621    0.1745    0.1821   -0.7149 -0.4819   -0.6291   -0.9288   -0.0292 -0.2766    0.5995    0.3178    0.6972 -0.5029   -0.4630    0.0570   -0.0438</pre> <p>D =</p> <pre>2.4021    0        0        0 0       -0.7158    0        0 0        0       -0.4400    0 0        0        0     -0.0346</pre>
<b>svds(A)</b> <b>svds(A,k)</b> <b>svds(A,k,0)</b> <b>[U,S,V] = svds(A,,,)</b>	<p>Da los 5 mayores valores singulares de <math>A</math>      Da los <math>k</math> mayores valores singulares de <math>A</math>      Da los <math>k</math> mayores valores singulares de <math>A</math> usando <i>eigs</i>      Da las matrices <math>U(m,k)</math> con columnas ortonormales, <math>S(k,k)</math>      diagonal y <math>V(n,k)</math> con columnas ortonormales</p> <pre>&gt;&gt; [U, S, V] = svds (A)  U =  -0.6380    0.3590    0.2155    0.6462 -0.5202   -0.2451   -0.8111   -0.1069 -0.2287    0.7358    0.0083   -0.6373 -0.5197   -0.5193    0.5436   -0.4059  S =  2.6201  0        0        0 0       0.8590    0        0 0        0       0.3796    0 0        0        0     0.0306  V =  -0.5705   -0.3613   -0.1621   -0.7196 -0.3061    0.1444    0.9401   -0.0416 -0.6310   -0.3268   -0.1247    0.6924 -0.4274    0.8613   -0.2729   -0.0321</pre>
<b>H=hadamard(n)</b>	<p>Matriz con valores 1 o -1 tal que <math>H'*H = n*eye(n)</math></p> <pre>&gt;&gt; H=hadamard(4)  H = 1     1     1     1 1    -1     1    -1 1     1    -1    -1 1    -1    -1     1</pre>
<b>hankel(V)</b>	<p>Matriz cuya primera columna es el vector <math>V</math> y cuyos elementos son cero por debajo de la primera antidiagonal. La matriz <math>Hankel(C,R)</math> tiene como primera columna el vector <math>C</math> y como última fila el vector <math>R</math></p>

	<pre>&gt;&gt; hankel([1 2 3 4 5]) ans = 1 2 3 4 5 2 3 4 5 0 3 4 5 0 0 4 5 0 0 0</pre>
<b>hilb(n)</b>	<p><i>Matriz de Hilbert de orden n tal que <math>A_{ij}=1/(i+j-1)</math></i></p> <pre>&gt;&gt; hilb(4) ans = 1.0000 0.5000 0.3333 0.2500 0.5000 0.3333 0.2500 0.2000 0.3333 0.2500 0.2000 0.1667 0.2500 0.2000 0.1667 0.1429</pre>
<b>invhilb(n)</b>	<p><i>Inversa de la matriz de Hilbert de orden n</i></p> <pre>&gt;&gt; invhilb(4) ans = 16 -120 240 -140 -120 1200 -2700 1680 240 -2700 6480 -4200 -140 1680 -4200 2800</pre>
<b>magic(n)</b>	<p><i>Matriz mágica de orden n. Sus elementos son los enteros desde 1 hasta n con iguales sumas de filas y columnas</i></p> <pre>&gt;&gt; magic(4) ans = 16 2 3 13 5 11 10 8 9 7 6 12 4 14 15 1</pre>
<b>pascal(n)</b>	<p><i>Matriz de Pascal de orden n (simétrica, definida positiva y basada en el triángulo de Pascal)</i></p> <pre>&gt;&gt; pascal(4) ans = 1 1 1 1 1 2 3 4 1 3 6 10 1 4 10 20</pre>
<b>rosser</b>	<p><i>Matriz 8x8 con un autovalor doble, otro nulo, tres casi iguales, otro muy pequeño y dos opuestos</i></p> <pre>&gt;&gt; rosser ans = 611 196 -192 407 -8 -52 -49 29 196 899 113 -192 -71 -43 -8 -44 -192 113 899 196 61 49 8 52 407 -192 196 611 8 44 59 -23 -8 -71 61 8 411 -599 208 208 -52 -43 49 44 -599 411 208 208 -49 -8 8 59 208 208 99 -911 29 -44 52 -23 208 208 -911 99</pre>

<b>toeplitz(C,R)</b>	<i>Matriz de Toeplitz (no simétrica con el vector C de primera columna y el vector R como primera fila)</i> <pre>&gt;&gt; toeplitz ([1 2 3 4 ],[1 6 7 8]) ans = 1     6     7     8 2     1     6     7 3     2     1     6 4     3     2     1</pre>
<b>wilkinson(n)</b>	<i>Matriz de Wilkinson (simétrica tridiagonal con pares de autovalores cercanos pero no iguales)</i> <pre>&gt;&gt; wilkinson(5) ans = 2     1     0     0     0 1     1     1     0     0 0     1     0     1     0 0     0     1     1     1 0     0     0     1     2</pre>
<b>compan(P)</b>	<i>Matriz del polinomio de coeficientes P</i> <pre>&gt;&gt; compan([1 2 3 4 5]) ans = -2    -3    -4    -5 1     0     0     0 0     1     0     0 0     0     1     0</pre>

## ESPACIOS VECTORIALES, APLICACIONES LINEALES Y FORMAS CUADRÁTICAS

Los comandos matriciales presentados anteriormente permiten trabajar con espacios vectoriales, aplicaciones lineales y formas cuadráticas. Será posible trabajar con dependencia e independencia lineal, bases, cambios de base y geometría vectorial en dos y tres dimensiones. A través de ejemplos se irán ilustrando estas afirmaciones.

Como primer ejemplo comprobamos si los vectores  $\{\{1,2,-3,4\}, \{3,-1,2,1\}, \{1,-5,8,-7\}, \{2,3,1,-1\}\}$  son linealmente independientes

```
>> A=[1,2,-3,4;3,-1,2,1;1,-5,8,-7;2,3,1,-1]
```

```
A =
```

```
1     2     -3     4
3    -1      2     1
1    -5      8    -7
2     3      1    -1
```

```
>> det(A)
```

```
ans =
```

```
0
```

Como el determinante de la matriz de sus coordenadas es cero, los vectores son linealmente independientes.

Como segundo ejemplo, comprobemos si el conjunto de vectores de  $\{\{1,2,2,1\}, \{3,4,4,3\}, \{1,0,0,1\}\}$  son linealmente independientes.

```
>> B=[1,2,2,1;3,4,4,3;1,0,0,1]
```

```
B =
```

1	2	2	1
3	4	4	3
1	0	0	1

```
>> rank(B)
```

```
ans =
```

```
2
```

Tenemos tres vectores en  $\mathbb{R}^4$ , serían independientes si el rango de la matriz de sus coordenadas es 3. Pero hemos visto que dicho rango vale 2, lo que indica que los vectores son linealmente dependientes.

Como tercer ejemplo vamos a obtener la dimensión y una base de la variedad lineal generada por los vectores  $\{\{2,3,4,-1,1\}, \{3,4,7,-2,-1\}, \{1,3,-1,1,8\}, \{0,5,5,-1,4\}\}$ .

Para hallar la dimensión de la variedad lineal calculamos el rango de la matriz formada por los vectores que la generan. Dicho rango será la dimensión de la variedad lineal.

```
>> A=[2,3,4,-1,1;3,4,7,-2,-1;1,3,-1,1,8;0,5,5,-1,4]
```

```
A =
```

2	3	4	-1	1
3	4	7	-2	-1
1	3	-1	1	8
0	5	5	-1	4

```
>> rank(A)
```

```
ans =
```

```
3
```

Tenemos entonces que la dimensión de la variedad lineal es 3 y una base estará formada por los vectores que contengan las componentes de cualquier menor de orden 3 no nulo de la matriz A.

```
>> det([3 4 7; 1 3 -1; 0 5 5])
```

```
ans =
```

```
75
```

Luego una base de la variedad lineal estará formada por los vectores  $\{\{3,4,7,-2,-1\}, \{1,3,-1,1,8\}, \{0,5,5,-1,4\}\}$

Como cuarto ejemplo comprobaremos si los vectores  $\{\{2,3,-1\}, \{0,0,1\}, \{2,1,0\}\}$  forman una base en y en caso positivo obtendremos las componentes del vector  $\{3,5,1\}$  en la citada base.

Los vectores dados son 3 vectores de un espacio de dimensión 3, luego la condición para que sean una base es que su determinante sea cero

```
>> det([2,3,-1;0,0,1;2,1,0])
```

```
ans =
```

```
4
```

Los vectores forman una base. Para hallar las componentes del vector  $\{3,5,1\}$  en dicha base se realizan las siguientes operaciones;

```
>> inv([2,0,2;3,0,1;-1,1,0])*[3,5,1]'
```

```
ans =
```

```
1.7500
2.7500
-0.2500
```

Como quinto ejemplo consideramos las bases de  $\mathbb{R}^3$  definidas como  $B=\{\{1,0,0\}, \{-1,1,0\}, \{0,1,-1\}\}$  y  $B1=\{\{1,0,-1\}, \{2,1,0\}, \{-1,1,1\}\}$  y hallamos la matriz de cambio de base de B a B1 calculando adicionalmente las componentes del vector  $\{2,1,3\}$  en base B en la base B1.

La operativa a llevar a cabo sería la siguiente:

```
>> B=[1,0,0;-1,1,0;0,1,-1]
```

```
B =
```

```
1          0
0
-1         1         0
0         1        -1
```

```
>> B1=[1,0,-1;2,1,0;-1, 1, 1]
```

```
B1 =
```

```
1      0      -1
 2      1       0
 -1     1       1

>> A=inv(B1')*B'

A =
-0.5000    1.5000    2.5000
 0.5000   -0.5000   -0.5000
 -0.5000    1.5000    1.5000
```

Para hallar las componentes del vector  $\{2,1,3\}$  en base B en la base B1 a partir de la matriz A de cambio de base, realizamos la siguiente operación:

```
>> inv(B1')*B'*[2,1,3]'

ans =
 8
-1
 5
```

Como sexto ejemplo hallamos el producto mixto de los vectores  $\{\{1,1,2\}, \{0,1,0\}, \{0,1,1\}\}$  y calculamos el área del triángulo cuyos vértices tienen como coordenadas los puntos  $(0,0)$ ,  $(5,1)$  y  $(3,7)$ .

```
>> dot([1,1,2],cross([0,1,0], [0,1,1]))

ans =
 1

>> (d/2)*det([0,0,1;5,1,1;3,7,1])

ans =
 16
```

Como séptimo ejemplo consideramos una aplicación lineal de en cuya matriz en las bases canónicas es la siguiente:

$$\begin{pmatrix} 0 & -3 & -1 & -3 & -1 \\ -3 & 3 & -3 & -3 & -1 \\ 2 & 2 & -1 & 1 & 2 \end{pmatrix}$$

Se trata de encontrar una base para su núcleo y la dimensión de dicho núcleo, así como la imagen de los vectores  $\{4,2,0,0,-6\}$  y  $\{1,2,-1,-2,3\}$  mediante dicha aplicación. Hallar también una base para la imagen de  $f$  y su dimensión.

```
>> A=[0,-3,-1,-3,-1;-3, 3,-3,-3,-1;2,2,-1,1, 2]
```

```
A =
0   -3   -1   -3   -1
-3   3   -3   -3   -1
2   2   -1   1   2

>> null(A)
ans =
-0.5397  -0.1251
-0.2787  -0.2942
-0.0266  -0.6948
 0.0230   0.6021
 0.7936  -0.2292
```

Los dos vectores de la salida anterior forman la base del núcleo y por lo tanto la dimensión del núcleo es 2.

Para hallar la imagen de cualquier vector  $X$  mediante la aplicación lineal  $f$  nos basamos en que  $f(X) = A^*X$ . Tenemos:

```
>> A*[4 2 0 0 -6]'
```

```
ans =
```

```
0
0
0
```

```
>> A*[1 2 -1 -2 3]'
```

```
ans =
```

```
-2
9
11
```

La dimensión de la imagen de  $f$  coincide con el rango de su matriz.

```
>> rank(A)
```

```
ans =
```

```
3
```

La dimensión de la imagen de  $f$  es 3 y una base estará formada por tres columnas contenidas en la matriz  $A$  que sean linealmente independientes.

```
>> det([0 -3 -2;-3 3 2; -1 -3 -1])
```

```
ans =
```

Por tanto, los vectores  $\{\{0 \ -3 \ -2\}; \{-3 \ 3 \ 2\}; \{-1 \ -3 \ -1\}\}$  forma una base de la imagen de  $f$ .

Como octavo ejemplo consideramos la aplicación lineal  $f$  entre dos subespacios vectoriales  $U$  y  $V$  del espacio real tridimensional, de tal forma que  $f(a,b,c) = (a+b, b+c, a+c)$ , siendo  $(a,b,c)$  cualquier punto de  $U$ . Se trata de hallar la matriz asociada a las aplicaciones  $f, f^5$  y  $e^f$ .

Para hallar la matriz de  $f$ , hay que considerar los vectores transformados por  $f$  de los de la base canónica:

```
>> f
f =
[ a + b, b + c, a + c]
>> subs(f,{a,b,c}, {1,0,0})
ans =
10          1
>> subs(f,{a,b,c}, {0,1,0})
ans =
1          1          0
>> subs(f,{a,b,c}, {0,0,1})
ans =
0          1          1
```

La matriz asociada a la aplicación lineal  $f$  tendrá como columnas los vectores anteriores transformados de la base canónica de  $R^3$ . Será entonces:

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$$

La matriz asociada a  $f^5$  será y la matriz asociada a  $e^f$  será  $e^A$ .

```
>> A= ([1 0 1; 1 1 0; 0 1 1])'
A =
1          1          0
0          1          1
1          0          1
```

```
>> A^5
ans =
11     10     11
11     11     10
10     11     11
>> expm(A)
ans =
3.1751    2.8321    1.3819
1.3819    3.1751    2.8321
2.8321    1.3819    3.1751
```

Como noveno ejemplo clasificamos la forma bilineal  $f:U \times V \rightarrow \mathbb{R}$  y la forma cuadrática  $g:U \rightarrow \mathbb{R}$  definidas de la siguiente forma:

$$f[(a, b, c), (d, e, f)] = (a, b, c) \begin{pmatrix} 1 & -2 & 0 \\ 0 & 0 & 4 \\ -1 & 0 & 3 \end{pmatrix} \begin{pmatrix} d \\ e \\ f \end{pmatrix}$$

$$g(a, b, c) = (a, b, c) \begin{pmatrix} 1 & -1 & 3 \\ -1 & 1 & 3 \\ 3 & 3/2 & 4 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

```
>> A=[1,-2,0;0,0,4;-1,0,-3]
A =
1     -2      0
0      0      4
-1      0     -3
>> det(A)
ans =
8
```

Como el determinante de la matriz de  $f$  es no nulo, la forma bilineal es regular no degenerada.

```
>> B=[1,-1,3;-1,1,-3/2;3,-3/2,4]
B =
```

```
1.0000    -1.0000    3.0000
-1.0000    1.0000   -1.5000
 3.0000   -1.5000    4.0000
```

Para clasificar la forma cuadrática calculamos sus determinantes diagonales.

```
>> det(B)
ans =
-2.2500
>> det([1,-1;-1,1])
ans =
0
```

Resulta que la forma cuadrática es semidefinida negativa.

Pero la clasificación también puede hacerse a través de los autovalores de la matriz de la forma cuadrática.

Una forma cuadrática es definida positiva si, y sólo si, todos sus autovalores son positivos estrictamente. Una forma cuadrática es definida negativa si, y sólo si, todos sus autovalores son negativos estrictamente.

Una forma cuadrática es semidefinida positiva si, y sólo si, todos sus autovalores son no negativos. Una forma cuadrática es semidefinida negativa si, y sólo si, todos sus autovalores son no positivos.

Una forma cuadrática es indefinida si existen autovalores positivos y negativos.

```
>> eig(B)
ans =
-0.8569
 0.4071
 6.4498
```

Hay autovalores positivos y negativos, con lo que la forma cuadrática es indefinida.

*Ejercicio 1. Consideramos la siguiente matriz:*

$$M = \begin{pmatrix} 1/3 & 1/4 & 1/5 \\ 1/4 & 1/5 & 1/6 \\ 1/5 & 1/6 & 1/7 \end{pmatrix}$$

*Hallar su transpuesta, su inversa, su determinante, su rango, su traza, sus valores singulares, su condición, su norma,  $M^3$ ,  $e^M$ ,  $\log(M)$  y  $\sqrt{M}$ :*

```
>> M=[1/3,1/4,1/5; 1/4,1/5,1/6;1/5,1/6,1/7]
```

```
M =
```

0.3333	0.2500	0.2000
0.2500	0.2000	0.1667
0.2000	0.1667	0.1429

```
>> transpuesta=M'
```

```
transpuesta =
```

0.3333	0.2500	0.2000
0.2500	0.2000	0.1667
0.2000	0.1667	0.1429

```
>> inversa=inv(M)
```

```
inversa =
```

1.0e+003 *		
0.3000	-0.9000	0.6300
-0.9000	2.8800	-2.1000
0.6300	-2.1000	1.5750

Para comprobar que la inversa está bien calculada, la multiplicamos por M y el resultado ha de ser la matriz identidad de orden 3:

```
>> M*inv(M)
```

```
ans =
```

1.0000	0.0000	0.0000
0.0000	1.0000	0.0000
0.0000	0.0000	1.0000

```
>> determinante=det(M)
```

```
determinante =
```

```
2.6455e-006
```

```
>> rango=rank(M)

rango =

3

>> traza=trace(M)

traza =

0.6762

>> vsingulares=svd(M)

vsingulares =

0.6571
0.0189
0.0002

>> condicion=cond(M)

condicion =

3.0886e+003

Para el cálculo de la norma, hallamos la norma, la 1-norma, la
norma infinita y la F-norma:

>> norm(M)

ans =

0.6571

>> norm(M,1)

ans =

0.7833

>> norm(M,inf)

ans =

0.7833

>> norm(M,'fro')

ans =

0.6573

>> M^3
```

```
ans =  
  
0.1403    0.1096    0.0901  
0.1096    0.0856    0.0704  
0.0901    0.0704    0.0578  
  
>> logm(M)  
  
ans =  
  
-2.4766    2.2200    0.5021  
2.2200    -5.6421    2.8954  
0.5021    2.8954    -4.7240  
  
>> sqrtm(M)  
  
ans =  
  
0.4631    0.2832    0.1966  
0.2832    0.2654    0.2221  
0.1966    0.2221    0.2342
```

Para calcular  $e^M$ , se usarán las variantes mediante autovalores, aproximantes de Padé, desarrollos de Taylor y condición de la matriz M:

```
>> expm(M)  
  
ans =  
  
1.4679    0.3550    0.2863  
0.3550    1.2821    0.2342  
0.2863    0.2342    1.1984  
  
>> expm1(M)  
  
ans =  
  
1.4679    0.3550    0.2863  
0.3550    1.2821    0.2342  
0.2863    0.2342    1.1984  
  
>> expm2(M)  
  
ans =  
  
1.4679    0.3550    0.2863  
0.3550    1.2821    0.2342  
0.2863    0.2342    1.1984  
  
>> expm3(M)  
  
ans =
```

```
1.4679  0.3550  0.2863
0.3550  1.2821  0.2342
0.2863  0.2342  1.1984
```

Como vemos, la matriz exponencial coincide para todos los métodos utilizados.

*Ejercicio 2. Consideramos la matriz siguiente:*

$$M = \begin{pmatrix} 1/3 & 1/4 & 1/5 \\ 1/4 & 1/5 & 1/6 \\ 1/5 & 1/6 & 1/7 \end{pmatrix}$$

*Hallar su transpuesta, su inversa, su determinante, su rango, su traza, sus valores singulares,  $M^3$ ,  $\log(M)$  y  $\sqrt{M}$ :*

```
>> M=sym('[1/3,1/4,1/5; 1/4,1/5,1/6;1/5,1/6,1/7]')

M =
[1/3, 1/4, 1/5]
[1/4, 1/5, 1/6]
[1/5, 1/6, 1/7]

>> transpuesta=M'

transpuesta =
[1/3, 1/4, 1/5]
[1/4, 1/5, 1/6]
[1/5, 1/6, 1/7]

>> inversa=inv(M)

inversa =
[300, -900, 630]
[-900, 2880, -2100]
[630, -2100, 1575]

>> determinante=det (M)

determinante =
1/378000

>> rango=rank (M)

rango =
3
```

```

>> traza=trace (M)

traza =
71/105

>> vsingulares=svd (M)

vsingulares =
(12703/88200 - (1045602865/351298031616 + (102103^(1/2)*i)/49787136)^(1/3)/2
 1030177/(99574272*(1045602865/351298031616 +
(102103^(1/2)*i)/49787136)^(1/3)) +
49787136*(1045602865/351298031616 +
(102103^(1/2)*i)/49787136)^(1/3) - ( (102103^(1/2)*i)/49787136 +
1045602865/351298031616)^(1/3)*i)/2)^(1/2)
(12703/88200 - (1045602865/351298031616 + (102103^(1/2)*i)/49787136)^(1/3)/2
 1030177/(99574272*(1045602865/351298031616 +
(102103^(1/2)*i)/49787136)^(1/3)) +
49787136*(1045602865/351298031616 +
(102103^(1/2)*i)/49787136)^(1/3) - ((102103^(1/2)*i)/49787136 +
1045602865/351298031616)^(1/3)*i)/2)^(1/2)

(1030177/(49787136*(1045602865/351298031616 +
(102103^(1/2)*i)/49787136)^(1/3)) + (1045602865/351298031616 +
(102103^(1/2)*i)/49787136)^(1/3) + 12703/88200)^(1/2)

>> M^3

ans =
[ 10603/75600, 1227/11200, 26477/294000]
[ 1227/11200, 10783/126000, 74461/1058400]
[ 26477/294000, 74461/1058400, 8927/154350]

>> log(M)

ans =
[ -log(3), -log(4), -log(5)]
[ -log(4), -log(5), -log(6)]
[ -log(5), -log(6), -log(7)]

>> sqrt(M)

ans =
[3^(1/2)/3, 1/2, 5^(1/2)/5]
[ 1/2, 5^(1/2)/5, 6^(1/2)/6]
[5^(1/2)/5, 6^(1/2)/6, 7^(1/2)/7]

```

*Ejercicio 3. Consideramos la matriz simbólica siguiente:*

$$A = \begin{bmatrix} a & b & c \\ 3c & a - 3c & b \\ 3b & -3b + 3c & a - 3c \end{bmatrix}$$

*Calcular  $A'$ ,  $A^{-1}$ , determinante( $A$ ), traza( $A$ ), rango( $A$ ), inv( $A$ ) y  $A^2$*

```
>> A=sym('[a,b,c; 3*c,a-3*c,b; 3*b,-3*b+3*c,a-3*c]')

A =
[a,           b,           c]
[3*c,         a - 3*c,       b]
[3*b, 3*c - 3*b, a - 3*c]

>> transpose(A)

ans =
[a,           3*c,         3*b]
[b, a - 3*c, 3*c - 3*b]
[c,           b,           a - 3*c]

>> pretty(det(A))

 3      2      2          2      3      2      3
a - 6 a c + 3 a b - 9 a b c + 9 a c + 3 b + 9 b c + 9 c

>> pretty(trace(A))

 3 a - 6 c

>> rank(A)

ans =
3

>> simplify(inv(A))
ans =
[(a^2 - 6*a*c + 3*b^2 - 3*b*c + 9*c^2)/(c^2*(9*a + 9*b) - c*(6*a^2 + 9*b*a) + 3*a*b^2 +
a^3 + 3*b^3 + 9*c^3) ,
 -(a*b - 3*c^2)/(c^2*(9*a + 9*b) - c*(6*a^2 + 9*b*a)) +
3*a*b^2 + a^3 + 3*b^3 + 9*c^3) ,
 (b^2 + 3*c^2 - a*c) / (c^2*(9*a + 9*b) - c*(6*a^2 + 9*b*a)) +
3*a*b^2 + a^3 + 3*b^3 + 9*c^3) ]
[(3*b^2 + 9*c^2 - 3*a*c)/(c^2*(9*a + 9*b) - c*(6*a^2 + 9*b*a) + 3*a*b^2 +
a^3 + 3*b^3 + 9*c^3) ,
 -(c*(3*a + 3*b) - a^2) / (9*b*c^2 - 6*a^2*c + a*(3*b^2 - 9*b*c) +
9*c^2) + a^3 + 3*b^3 + 9*c^3) ,
 -(a*b - 3*c^2)/(c^2*(9*a + 9*b) - c*(6*a^2 + 9*b*a)) +
3*a*b^2 + a^3 + 3*b^3 + 9*c^3) ]
[-(3*a*b - 9*c^2)/(c^2*(9*a + 9*b) - c*(6*a^2 + 9*b*a) + 3*a*b^2 +
a^3 + 3*b^3 + 9*c^3) ,
 (3*b^2 + 3*a*b - 3*a*c) / (c^2*(9*a + 9*b) - c*(6*a^2 + 9*b*a) +
3*a*b^2 + a^3 + 3*b^3 + 9*c^3) ,
 -(c*(3*a + 3*b) - a^2) / (9*b*c^2 - 6*a^2*c + a*(3*b^2 - 9*b*c) +
9*c^2) + a^3 + 3*b^3 + 9*c^3) ]
```

```

>> pretty (simplify(A^2))

+-+-----+-----+-----+
|   2           2           2           2           2
| a + 6 b c ,      3 c - 6 b c + 2 a b ,      b - 3 c + 2 a c
|   2           2           2           2           2
| 3 b - 9 c + 6 a c , 6 b c - 3 b + ( a - 3 c ) , 2 b ( a - 3 c ) + 3 c
|   2           2           2           2           2
| 9 c - 18 b c + 6 a b , 3 b - 2 ( a - 3 c ) ( 3 b - 3 c ) , 6 b c - 3 b +(a-3c)
+-+-----+-----+-----+

```

*Ejercicio 4. Dadas las matrices A y B siguientes:*

$$A = \begin{bmatrix} \cosh(a) & \sinh(a) \\ \sinh(a) & \cosh(a) \end{bmatrix} \quad B = \begin{bmatrix} \sinh(a) & \cosh(a) \\ \cosh(a) & \sinh(a) \end{bmatrix}$$

- a) Calcular  $M1=A^2+B^2$ ,  $M2=A^2-B^2$ ,  $A^n$ ,  $B^n$ ,  $e^A$ ,  $e^B$
- b) Hallar inversas, determinantes, trazas y normas de las matrices A y B

```

>> A=sym ('[cosh(a),sinh(a);sinh(a),cosh(a)]')

A =

[ cosh(a), sinh(a)]
[ sinh(a), cosh(a)]

>> B=sym ('[sinh(a),cosh(a) ;cosh(a),sinh(a)]')

B =

[ sinh(a), cosh(a)]
[ cosh(a), sinh(a)]

>> M1=A-'2+B''2

M1 =

[ 2*cosh(a)^2 + 2*sinh(a)^2,          4*cosh(a)*sinh(a) ]
[        4*cosh(a) *sinh(a) , 2*cosh(a)^2 + 2*sinh(a)^2]

>> pretty(simplify(M1))

+-+-----+-----+
|   2           |           2
| 4 sinh(a) + 2,      2 sinh(2 a)           |
|           |           2
|           2           |
|       2 sinh(2 a),      4 sinh(a) + 2 |
+-+-----+-----+-----+

```

```

>> M2=A^2-B^2

M2 =

```

```
[0, 0]
[0, 0]
```

Para calcular  $A^n$  y  $B^n$  hallamos sus potencias sucesivas para intentar ver la ley de formación:

```
> [simplify(A^2), simplify(A^3), simplify(A^4)]
ans =
[ cosh(2*a), sinh(2*a), cosh(3*a), sinh(3*a), cosh(4*a), sinh(4*a)]
[ sinh(2*a), cosh(2*a), sinh(3*a), cosh(3*a), sinh(4*a), cosh(4*a)]
>> [simplify(B^2), simplify(B^3), simplify(B^4)]
ans =
[ cosh(2*a), sinh(2*a), sinh(3*a), cosh(3*a), cosh(4*a), sinh(4*a)]
[ sinh(2*a), cosh(2*a), cosh(3*a), sinh(3*a), sinh(4*a), cosh(4*a)]
```

La ley de formación de las potencias enésimas es evidente, con lo que ya podemos escribir:

$$A^n = B^n = \begin{bmatrix} \cosh(na) & \sinh(na) \\ \sinh(2a) & \cosh(2a) \end{bmatrix}$$

```
>> simplify(inv(A))
ans =
[ cosh(a), -sinh(a)]
[ -sinh(a), cosh(a)]
>> simplify(inv(B))
ans =
[ -sinh(a), cosh(a)]
[ cosh(a), -sinh(a)]
>> simplify(det(A))
ans =
1
>> simplify(det(B))
ans =
-1
>> simplify(trace(A))
ans =
```

```

2*cosh(a)

>> simplify(trace(B))

ans =

2*sinh(a)

>> simplify(exp(A))

ans =

[ exp(cosh(a)), exp(sinh(a))]

[ exp(sinh(a)), exp(cosh(a))]

>> simplify(exp(B))

[ exp(sinh(a)), exp(cosh(a))]

[ exp(cosh(a)), exp(sinh(a))]

```

*Ejercicio 5. Considerar una matriz A aleatoria normal cuadrada de orden 3 y calcular la matriz diagonal D con los autovalores de A y la matriz V cuyas columnas son sus autovectores (si la salida es compleja, transformarla a real).*

*Hallar la matriz balanceada de Ay las formas real y compleja de su descomposición de Schur.*

*Encontrar los coeficientes del polinomio característico de la matriz A.*

*Calcular la matriz triangular superior R de la misma dimensión que la matriz A, la matriz permutación E y la matriz ortogonal Q tales que  $A^*E=Q^*R$  y comprobar el resultado.*

*Considerar la matriz B de Hessenberg de A y calcular la matriz diagonal D de autovalores generalizados de A y B, y una matriz V cuyas columnas son los autovectores correspondientes, cumpliéndose  $A^*V=B^*V*D$ . Asimismo, calcular el vector de valores singulares generalizados de A y B.*

```

>> A=randn(3)

A =

-0.4326    0.2877    1.1892
-1.6656   -1.1465   -0.0376
 0.1253    1.1909    0.3273

>> [V,D] = eig(A)

```

```
V =
0.2827      0.4094 - 0.3992i      0.4094 + 0.3992i
0.8191      -0.0950 + 0.5569i      0.0950 - 0.5569i
-0.4991      0.5948                  0.5948

D =
-1.6984          0                  0
    0      0.2233 + 1.0309i      0
    0                  0      0.2233 - 1.0309i

>> [V,D] = cdf2rdf(V,D)

V =
0.2827      0.4094      -0.3992
0.8191      -0.0950      0.5569
-0.4991      0.5948      0

D =
-1.6984          0                  0
    0      0.2233      1.0309
    0      -1.0309      0.2233

>> [T,B] = balance(A)

T =
1   0   0
0   1   0
0   0   1

B =
-0.4326      0.2877      1.1892
-1.6656     -1.1465     -0.0376
  0.1253      1.1909      0.3273

>> [U,T] = schur(A)

U =
0.2827      0.2924      0.9136
0.8191     -0.5691     -0.0713
-0.4991     -0.7685      0.4004

T =
-1.6984      0.2644      -1.2548
    0      0.2233      0.7223
    0     -1.4713      0.2233

>> [U,T] = rsf2csf(U,T)

U =
```

```

0.2827      -0.7482 + 0.1678i      0.2395 - 0.5242i
0.8191      0.0584 - 0.3266i      -0.4661 + 0.0409i
-0.4991      -0.3279 - 0.4410i      -0.6294 - 0.2298i

T =
-1.6984      1.0277 + 0.1517i      0.2165 + 0.7201i
          0      0.2233 + 1.0309i      0.7490 - 0.0000i
          0          0      0.2233 - 1.0309i

>> poly(A)
ans =
1.0000    1.2517    0.3540    1.8895

```

A continuación se calcula la matriz triangular superior  $R$  de la misma dimensión que la matriz  $A$  del ejemplo anterior, la matriz permutación  $E$  y la matriz ortogonal  $Q$  tales que  $A * E = Q * R$  y se comprueba el resultado.

```

>> [Q,R,E] = qr(A)
Q =
-0.2507    0.4556      -0.8542
-0.9653    -0.0514      0.2559
 0.0726    0.8887      0.4527

R =
1.7254      1.1211      -0.2380
          0      1.2484      0.8346
          0          0      -0.8772

E =
 1     0     0
 0     1     0
 0     0     1

```

```

>> A*E
ans =
-0.4326    0.2877    1.1892
-1.6656   -1.1465   -0.0376
 0.1253    1.1909    0.3273

```

```

>> Q*R
-0.4326    0.2877    1.1892
-1.6656   -1.1465   -0.0376
 0.1253    1.1909    0.3273

```

Se observa que se cumple  $A^*E=Q^*R$ .

Ahora consideramos la matriz  $B$  de Hessenberg de  $A$  y calculamos la matriz diagonal  $D$  de autovalores generalizados de  $A$  y  $B$ , y una matriz  $V$  cuyas columnas son los autovectores correspondientes, cumpliéndose  $A^*V=B^*V*D$ . Asimismo calculamos el vector de valores singulares generalizados de  $A$  y  $B$ .

```
>> B=hess(A)

B =
-0.4326   -0.1976   1.2074
 1.6703   -1.2245   0.1544
    0   -1.0741   0.4053

>> [V,D] = eig(A,B)

V =
 0.0567   1.0000   1.0000
 -0.0376   -0.4998   0.5297
 -1.0000   0.4172   0.3785

D =
 1.0000   0   0
 0   -0.4722   0
 0   0   -2.1176

>> A*V

ans =
 -1.2245   -0.0803   0.1699
 -0.0137   -1.1082   -2.2872
 -0.3649   -0.3334   0.8801

>> B*V*D

ans =
 -1.2245   -0.0803   0.1699
 -0.0137   -1.1082   -2.2872
 -0.3649   -0.3334   0.8801

>> sigma = gsvd(A,B)

sigma =
 0.2874
 1.0
 3.4799
```

*Ejercicio 6. Consideramos la matriz cuadrada 3x3 siguiente:*

$$\begin{pmatrix} 1 & 5 & -2 \\ -7 & 3 & 1 \\ 2 & 2 & -2 \end{pmatrix}$$

*Realizar las descomposiciones de Schur, LU, QR, Cholesky, Hessenberg y valores singulares comprobando que los resultados son correctos. Hallar también la matriz pseudo inversa de A.*

En primer lugar, hallamos la descomposición de Schur, comprobando que el resultado es correcto.

```
>> A=[1,5,-2;-7,3,1;2,2,-2];
>> [U,T] = schur(A)
```

U =

-0.0530	-0.8892	-0.4544
-0.9910	-0.0093	0.1337
0.1231	-0.4573	0.8807

T =

2.4475	-5.7952	-4.6361
5.7628	0.3689	2.4332
0	0	-0.8163

Ahora comprobamos que  $U^*T^*U'=A$  y que  $U^*U'=eye(3)$ :

```
>> [U*T*0', U*U']
```

ans =

1.0000	5.0000	-2.0000	1.0000	0.0000	0.0000
-7.0000	3.0000	1.0000	0.0000	1.0000	0.0000
2.0000	2.0000	2.0000	0.0000	0.0000	1.0000

Ahora hallamos las descomposiciones LU, QR, Cholesky, Hessenberg y valores singulares, comprobando los resultados para cada caso:

```
>> [L,U,P] = lu(A)
```

L =

1.0000	0	0	
-0.1429	1.0000	0	Matriz triangular inferior
-0.2857	0.5263	1.0000	

U =

```
-7.0000  3.0000  1.0000
 0      5.4286 -1.8571      Matriz triangular superior
 0      0      -0.7368

P =
 0      1      0
 1      0      0
 0      0      1

>> [P*A, L*U]

ans =
 -7      3      1      -7      3      1      Tenemos que P*A=L*U
  1      5     -2      1      5     -2
  2      2     -2      2      2     -2

>> [Q,R,E] = qr(A)

Q =
 -0.1361   -0.8785   -0.4579
 0.9526   -0.2430    0.1831
 -0.2722   -0.4112    0.8700

R =
 -7.3485   1.6330   1.7691      Matriz triangular superior
  0   -5.9442   2.3366
  0       0   -0.6410

E =
 1      0      0
 0      1      0
 0      0      1

>> [A*E,Q*R]

ans =
 1.0000   5.0000  -2.0000   1.0000   5.0000  -2.0000
 -7.0000   3.0000   1.0000  -7.0000   3.0000   1.0000
  2.0000   2.0000  -2.0000   2.0000   2.0000  -2.0000

Luego, A*E=Q*R.

>> R = chol(A)

??? Error using ==> chol
Matrix must be positiva definita.

Se obtiene mensaje de error porque la matriz no es definida positiva.

>> [P,H] = hess(A)
```

P =

1.0000	0	0
0	-0.9615	0.2747
0	0.2747	0.9615

H =

1.0000	-5.3571	-0.5494
7.2801	1.8302	-2.0943
0	-3.0943	-0.8302

>> [P\*H\*P', P'\*P]

ans =

1.0000	5.0000	-2.0000	1.0000	0	0
-7.0000	3.0000	1.0000	0	1.0000	0
2.0000	2.0000	-2.0000	0	0	1.0000

Luego  $PHP' = A$  y  $P'P = I$ .

>> [U, S, V] = svd(A)

U =

-0.1034	-0.8623	0.4957
-0.9808	0.0056	-0.1949
0.1653	-0.5064	-0.8463

S =

7.8306	0	0
0	6.2735	0
0	0	0.5700

Matriz diagonal

V =

0.9058	-0.3051	0.2940
-0.3996	-0.8460	0.3530
-0.1411	0.4372	0.8882

>> U\*S\*V

ans =

1.0000	5.0000	-2.0000
-7.0000	3.0000	1.0000
2.0000	2.0000	-2.0000

Vemos que  $USV' = A$

Ahora calculamos la matriz pseudoinversa de A:

>> x = pinv(A)

x =

0.2857 -0.2143 -0.3929

```

0.4286    -0.0714    -0.4643
0.7143    -0.2857    -1.3571

>> [A*X*A, X*A*X]

ans =

```

$$\begin{bmatrix} 1.0000 & 5.0000 & -2.0000 & 0.2857 & -0.2143 & -0.3929 \\ -7.0000 & 3.0000 & 1.0000 & 0.4286 & -0.0714 & -0.4643 \\ 2.0000 & 2.0000 & -2.0000 & 0.7143 & -0.2857 & -1.3571 \end{bmatrix}$$

Luego, se cumple que  $AXA=A$  y  $XAX=X$ .

*Ejercicio 7. Dada la matriz siguiente:*

$$A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix}$$

*Calcular sus autovalores, su polinomio característico, su forma canónica de Jordán y sus valores singulares.*

Comenzamos definiendo la matriz  $A$  como una matriz simbólica:

```

>> A=sym('[1 0 0; 0 cos(a) -sin(a); 0 sin(a) cos(a)]')

A =

```

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{bmatrix}$$

```

>> eigensys(A)

ans =

```

$$\begin{bmatrix} 1 \\ [\cos(a)+1/2*(-4*\sin(a)^2)^{(1/2)}] \\ [\cos(a)-1/2*(-4*\sin(a)^2)^{(1/2)}] \end{bmatrix}$$

```

>> pretty(simple(poly(A)))

```

$$x^3 - 2x^2\cos(a) + x^2 - x + 2x\cos(a) - 1$$

```

>> jordan(A)

ans =

```

$$\begin{bmatrix} 1, & 0, & 0 \\ 0, & \cos(a)+1/2*(-4*\sin(a)^2)^{(1/2)}, & 0 \\ 0, & 0, & \cos(a)-1/2*(-4*\sin(a)^2)^{(1/2)} \end{bmatrix}$$

```
>> simple(svd(A))

ans =
[
[ 1/2*(4*cos(a-conj(a))+2*(-2 + 2*cos(2*a-2*conj(a)))^(1/2))^(1/2) ]^1]
[ 1/2*(4*cos(a-conj(a))-2*(-2 + 2*cos(2*a-2*conj(a)))^(1/2))^(1/2) ]
```

*Ejercicio 8. Diagonalizar la matriz simétrica cuyas filas son los vectores:*

$$(3, -1, 0), (-1, 2, -1), (0, -1, 3)$$

*Hallar la matriz de paso V, comprobar el resultado y ver que los valores propios de la matriz inicial son los elementos de la diagonal de la matriz semejante a la dada.*

Calculamos la matriz diagonal  $J$  semejante a  $A$ , que tendrá los valores propios de  $A$  en la diagonal y la matriz de paso  $V$ . Para ello, usamos el comando  $[V,J]=jordan(A)$ :

```
>> A=[3,-1,0; -1,2,-1; 0,-1,3]
```

$A =$

$$\begin{pmatrix} 3 & -1 & 0 \\ -1 & 2 & -1 \\ 0 & -1 & 3 \end{pmatrix}$$

```
>> [V, J] =jordan (A)
```

$V =$

$$\begin{pmatrix} 1 & -1 & 1 \\ 2 & 0 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

$J =$

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 4 \end{pmatrix}$$

Ahora comprobamos que la matriz diagonal  $J$  semejante a  $A$  tiene los valores propios de  $A$  en la diagonal:

```
>> eig(A)
```

$ans =$

$$\begin{matrix} 1.0000 \\ 3.0000 \\ 4.0000 \end{matrix}$$

Las matrices  $A$  y  $J$  resultan ser semejantes, ya que existe la matriz de paso  $V$  que cumple la relación  $V^{-1} * A * V = J$ :

```
>> inv(V) * A * V
ans =
1.0000    0    -0.0000
0    3.0000    0
0    0    4.0000
```

*Ejercicio 9. Hallar una matriz diagonal semejante a cada una de las siguientes matrices:*

$$A = \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 1 & -\sin(a) \\ -1 & 0 & \cos(a) \\ -\sin(a) & \cos(a) & 0 \end{bmatrix} \quad C = \begin{bmatrix} \cos(a) & -\sin(a) \\ \sin(a) & \cos(a) \end{bmatrix}$$

*Hallar las matrices de paso y comprobar los resultados. Hallar el polinomio característico para las tres matrices.*

```
>> A=sym('[0,-r,q;r,0,-p;-q,p,0]');
>> [V,J]=jordan(A)

V =
[ (q*(- p^2 - q^2 - r^2)^(1/2))/(p^2 + q^2) - (p*r) / (p^2 + q^2), -(q*(- p^2 - q^2 - r^2)^(1/2))/(p^2 + q^2) - (p*r) / (p^2 + q^2), p/r]
[ -(p*(- p^2 - q^2 - r^2)^(1/2))/(p^2 + q^2) - (p*r) / (p^2 + q^2),
(p*(- p^2 - q^2 - r^2)^(1/2)) / (p^2+ q^2) - (q*r) / (p^2 + q^2), q/r]
[ 1, 1, 1]

J =
[ -(- p^2 - q^2 - r^2)^(1/2), 0, 0]
[ 0, (- p^2 - q^2 - r^2)^(1/2), 0]
[ 0, 0, 0]
```

Ya tenemos la matriz diagonal  $J$  semejante a la matriz y la matriz de paso  $V$ . Ahora, analizamos la matriz  $B$ :

```
>> B=sym ('[0,1,-sin(a); -1, 0, cos (a) ; -sin(a), cos(a),0]')
>> J=simple(jordan(B))
```

```
J =
[0, 0, 0]
[0, 0, 0]
[0, 0, 0]
```

Se observa que la matriz  $B$  tiene como único valor propio el cero y es de multiplicidad 3. Además, el núcleo de  $B - 0*eye(3) = B$  tiene dimensión menor que tres, ya que el determinante de  $B$  es nulo. Concretamente, tiene dimensión uno (se ve calculando una base con el comando *nullspace(B)*). Como la multiplicidad y la dimensión del núcleo difieren, se concluye que la matriz  $B$  no es diagonalizable:

```
>> null(B)
ans =
```

```
[cos(a)]
[sin(a)]
[    1]
```

Hemos calculado una base del núcleo de  $B$ , que está formada por un único vector, luego, la dimensión del núcleo de  $B$  es 1:

```
>> det (B)
ans =
```

```
0
```

Analizamos ahora la matriz C :

```
>> C=sym('[cos(a),-sin(a);sin(a),cos(a)]');
>> [V,J]=jordan(C)
V =
```

```
[1/2, 1/2]
[i/2, -i/2]
```

```
J =
```

```
[ cos(a) - sin(a)*i, 0
  0, cos (a) + sin(a)*i]
```

Podemos intentar simplificar la expresión de la matriz de Jordán  $J$ .

Ya tenemos la matriz diagonal  $J$  semejante a la matriz C y la matriz de paso V. Calculamos ahora los polinomios característicos de las tres matrices:

```
>> pretty(poly(A))
      3   2   2   2
      x +(p + q + r )x
>> pretty(simple(sym(poly(B))))
      3
      x
```

```
>> pretty (simple(sym(poly(C))))
```

$$x^2 - 2 \cos(a) x + 1$$

*Ejercicio 10.* Hallar los autovalores de la matriz de Wilkinson de orden 8, de la matriz mágica de orden 8 y déla matriz de Rosser.

```
>> [eig(wilkinson (8)) , eig(rosser), eig(magic (8)) ]
```

```
ans =
```

$$\begin{array}{ccc} 1.0e+003 * & & \\ \begin{matrix} -0.0010 & -1.0200 & 0.2600 \\ 0.0002 & 0.0000 & 0.0518 \\ 0.0011 & 0.0001 & -0.0518 \\ 0.0017 & 1.0000 & 0.0000 \\ 0.0026 & 1.0000 & -0.0000 + 0.0000i \\ 0.0028 & 1.0199 & -0.0000 - 0.0000i \\ 0.0042 & 1.0200 & -0.0000 + 0.0000i \\ 0.0043 & 1.0200 & -0.0000 - 0.0000i \end{matrix} & & \end{array}$$

Se observa que la matriz de Wilkinson tiene pares de autovalores cercanos, pero no iguales. La matriz de Rosser tiene un autovalor doble, otro nulo, tres casi iguales, otro muy pequeño y dos opuestos.

*Ejercicio 11.* Consideramos la aplicación lineal  $f$  entre dos subespacios vectoriales  $U$  (contenido en  $R^3$ ) y  $V$  (contenido en  $R^4$ ), de tal forma que para cualquier punto  $(a,b,c)$  de  $U$  se tiene:

$$f(ab,c) = (a,b,c) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}$$

Hallar el núcleo y las dimensiones del núcleo y la imagen de  $f$ .

```
>> A=[1, 0, 0; 0,0,0;0,0,1;0,0,0];
```

El núcleo es el conjunto de vectores de  $U$  con imagen nula en  $V$ :

```
>> nuil(A)
```

```
ans =
```

$$\begin{matrix} 0 \\ 0 \\ 0 \end{matrix}$$

Con lo que el núcleo será el conjunto de vectores  $\{0,b,0\}$  con  $b$  variando en  $U$ . Además, evidentemente el núcleo tiene de dimensión 1, ya que hemos visto que una base es el vector  $\{0,1,0\}$ .

```
>> rank(A)
```

```
ans =
```

```
2
```

La dimensión de la imagen de  $f$  será 2, ya que la suma de las dimensiones del núcleo y la imagen ha de ser la dimensión de  $U$  (que es 3, por ser un subespacio de  $R^3$ ). Por otra parte, la dimensión de la imagen de  $f$  ha de coincidir con el rango de la matriz de la aplicación lineal, que es 2. Los dos vectores columna que contienen a los elementos del menor no nulo que define el rango de la matriz formarán una base de la imagen de  $f$ .

```
>> det([1,0;0,1])
```

```
ans =
```

```
1
```

Luego una base de la imagen de  $f$  será  $\{\{1,0,0,0\}, \{0,0,1,0\}\}$ .

*Ejercicio 12. Dada la forma cuadrática  $g: \rightarrow R$  definida de la siguiente forma:*

$$G(a,b,c) = (a,b,c) \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 2 \\ 0 & 2 & 2 \end{pmatrix} \begin{pmatrix} a \\ b \\ c \end{pmatrix}$$

*clasificarla y hallar su ecuación reducida, su rango y su signatura. U*

Para clasificar la forma cuadrática calculamos sus determinantes diagonales.

```
>> G=[1,0,0;0,2,2;0,2,2]
```

```
G =
```

$$\begin{matrix} 1 & 0 & 0 \\ 0 & 2 & 2 \\ 0 & 2 & 2 \end{matrix}$$

```
>> det(G)
```

```
ans =
```

```
0
```

```
>> det([1,0;0,2])
ans =
2
```

La forma cuadrática es degenerada, semidefinida positiva.

Para hallar la ecuación reducida diagonalizamos su matriz y hallamos la expresión utilizando la matriz diagonal.

```
>> J=jordan(G)
```

```
J =
```

```
0      0      0
0      1      0
0      0      4
```

La ecuación reducida de la forma cuadrática será:

$$h(x, y, z) = (x, y, z) \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = x^2 + 4z^2$$

```
>> rank(J)
```

```
ans =
```

```
2
```

El rango de la forma cuadrática es 2, ya que el rango de su matriz es 2. La signatura también es 2, ya que el número de términos positivos en la diagonal de la matriz diagonalizada es 2.

## CAPITULO 7

# LIMITES DE SUCESIONES Y FUNCIONES. CONTINUIDAD. UNA Y VARIAS VARIABLES

---

### LIMITES

MATLAB incorpora funciones que permiten trabajar con límites y con límites laterales. Esto posibilita, además del cálculo de los propios límites de sucesiones y funciones, analizar la continuidad y derivabilidad de funciones, así como analizar el carácter de las series numéricas y series de potencias. La tabla siguiente presenta las funciones más habituales de MATLAB relativas a límites.

<b>limit(sucesión, n, inf)</b>	<i>Calcula el límite de la sucesión, indicada por su término general, cuando n tiende a infinito</i>  <pre>&gt;&gt; syms n &gt;&gt; limit (((2*n-3) / (3*n-7))^4, n,inf) ans = 16/81</pre>
<b>limit(sucesión, inf)</b>	<i>Calcula el límite de la sucesión, indicada por su término general, cuando n tiende a infinito</i>  <pre>&gt;&gt; limit (((2*n-3) / (3*n-7))^4, inf) ans = 16/81</pre>
<b>limit(función, x, a)</b>	<i>Calcula el límite de la función de variable x, indicada por su expresión analítica, cuando la variable x tiende hacia el valor a</i>

	<pre>&gt;&gt; syms x a limit((x-1)/(x^(1/2)-1),x,1) ans = 2</pre>
<b>limit(función,a)</b>	<p>Calcula el límite de la función de variable x, indicada por su expresión analítica, cuando la variable x tiende hacia el valor a</p> <pre>&gt;&gt; limit((x-1)/(x^(1/2)-1),1) ans = 2</pre>
<b>limit(función,x,a,'right')</b>	<p>Calcula el límite de la función de variable x, indicada por su expresión analítica, cuando la variable x tiende al valor a por la derecha</p> <pre>&gt;&gt; syms x limit((exp(1/x)),x,0, 'right') ans = Inf</pre>
<b>limit(función,x,a,'left')</b>	<p>Calcula el límite de la función de variable x, indicada por su expresión analítica, cuando la variable x tiende al valor a por la izquierda</p> <pre>&gt;&gt; limit((exp(1/x)),x,0, 'left') ans = 0</pre>

Como primer ejemplo calculamos los límites de las sucesiones siguientes:

$$\lim_{n \rightarrow \infty} \left( \frac{3+n}{-1+n} \right)^n, \lim_{n \rightarrow \infty} \left( \frac{3+n}{-1+n} \right)^n, \lim_{n \rightarrow \infty} \sqrt[n]{\frac{1}{n}}, \lim_{n \rightarrow \infty} \frac{-\sqrt[3]{n} + \sqrt[3]{1+n}}{-\sqrt{n} + \sqrt{1+n}}, \lim_{n \rightarrow \infty} \frac{n!}{n^n}$$

```
>> syms n
>> limit (((n+3)/(n-1))^n, inf)
ans =
exp(4)
>> limit ((1-2/(n+3))^n, inf)
ans =
1/exp(2)
>> limit((1/n)^(1/n), inf)
```

```

ans =
1
>> limit (((n+1)^(1/3)-n^(1/3))/((n+1)^(1/2)-n^(1/2)) , inf)
ans =
0
>> limit((n^n*exp(-n)*sqrt(2*pi*n))/n^n, n,inf)
ans =
0

```

En el último límite se ha utilizado la fórmula de Sterling para aproximar el factorial.

$$n! \approx e^{-n} \sqrt{2\pi n}$$

Como segundo ejemplo calculamos los límites de las funciones siguientes:

$$\lim_{x \rightarrow 1} \frac{|x|}{\sin(x)}, \lim_{x \rightarrow 3} |x^2 - x - 7|, \lim_{x \rightarrow 1} \frac{x-1}{x^{n-1}}, \lim_{x \rightarrow 0} \sqrt[n]{e}$$

```

>> limit(abs(x)/sin(x),x,0)
ans =
NaN
>> syms x
>> limit(abs(x)/sin(x),x,0)
ans =
NaN
>> limit(abs(x)/sin(x),x,0,'left')
ans =
-1
>> limit(abs(x)/sin(x),x,0,'right')
ans =
1
>> limit(abs(x^2-x-7), x,3)
ans =

```

```
1
>> limit((x-1)/(x^n-1),x,1)
ans =
1/n
>> limit(exp(1)^(1/x),x,0)
ans =
NaN
>> limit(exp(1)^(1/x),x,0,'left')
ans =
0
>> limit(exp(1)^(1/x),x,0,'right')
ans =
Inf
```

## SUCESIONES DE FUNCIONES

MATLAB permite utilizar los comandos sobre límites de sucesiones y funciones vistos en la tabla anterior para estudiar la convergencia de sucesiones de funciones.

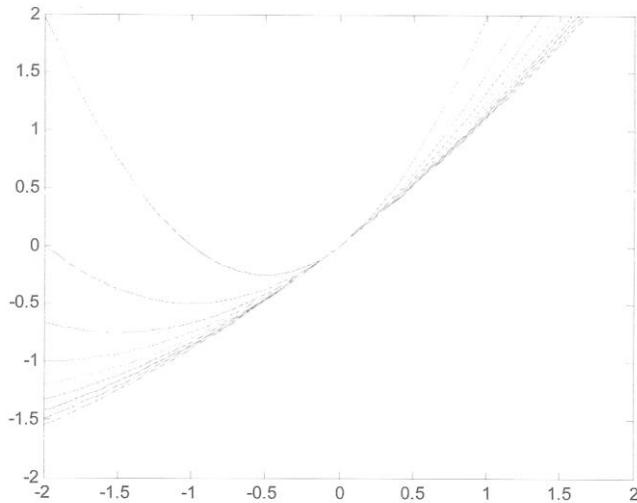
Como primer ejemplo consideraremos la sucesión de funciones la sucesión de funciones  $g_n(x) = (x^2 + nx)/n$  con  $x \in R$ .

Lo más adecuado será comenzar realizando una representación gráfica de las funciones de la sucesión para poder intuir cuál puede ser la función límite.

Para representar estas funciones podemos utilizar tanto el comando *plot* como el comando *fplot*.

Si utilizamos el comando *fplot*, la sintaxis para la representación gráfica simultánea sobre los mismos ejes de las 9 primeras funciones de la sucesión sería la siguiente:

```
>>
fplot ('[(x^2+x), (x^2+2*x)/2, (x^2+3*x)/3, (x^2+4*x)/4, (x^2+5*x)/5, (x^2+5*x)/5, (x^2+6*x)/6, (x^2+7*x)/7, (x^2+8*x)/8, (x^2+9*x)/9]', [-2,2,-2,2])
```



Gráficamente se observa que la sucesión de funciones converge hacia la función identidad  $f(x) = x$ .

Podemos corroborar este hecho utilizando el comando *limit* con la sintaxis siguiente:

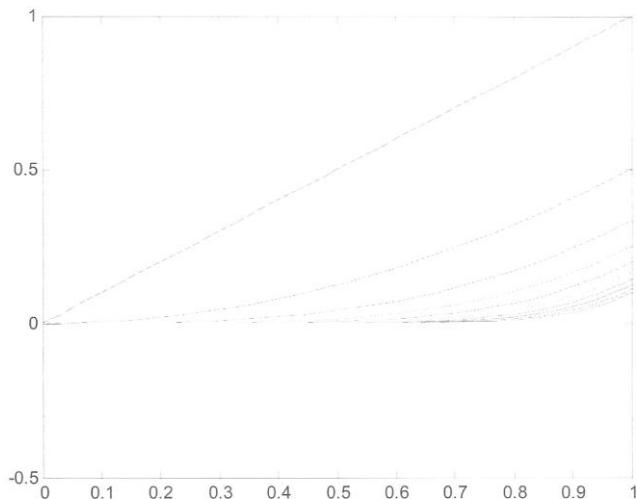
```
>> syms x n
>> limit((x^2+n*x)/n,n,inf)
ans =
x
```

Como segundo ejemplo consideramos la sucesión de funciones  $g_n(x) = (x^n)/n$  con  $x \in [0,1]$ .

```
>> limit (x.^n/n,n,inf);x(1)
ans =
0
```

A continuación se realiza la representación gráfica de la sucesión de funciones para corroborar el resultado.

```
>>
fplot ('[x,x^2/2,x^3/3,x^4/4,x^5/5,x^6/6,x^7/7,x^8/8,x^9/9,x^10/10]',[0,1,-1/2,1])
```



## CONTINUIDAD

El cálculo de límites es una herramienta necesaria para abordar el concepto de continuidad. Formalmente una función  $f$  se dice continua en el punto  $x=a$  si se verifica:

$$\lim_{x \rightarrow a} f(x) = f(a)$$

En otro caso, se dice discontinua en el punto. Es decir, que para que una función sea continua en  $x=a$ , la función ha de estar definida, ha de existir el límite (finito) de  $f$  en el punto  $x=a$  y el valor del límite ha de coincidir con el valor de la función en el punto. Si existe el límite de  $f(x)$  cuando  $x \rightarrow a$ , pero es distinto de  $f(a)$  (o  $f(a)$  no está definida),  $f$  es discontinua en  $x=a$ , y se dice que  $f$  presenta una discontinuidad evitable en  $x=a$ . La forma de evitar la discontinuidad es redefinir  $f(a)$  convenientemente.

Si existen los dos límites laterales de  $f$  en  $x=a$  (finitos o infinitos), pero son distintos, entonces la discontinuidad de  $f$  en  $x=a$  se denomina de primera especie. La diferencia entre los valores de los dos límites laterales distintos se llama salto. Si la diferencia es finita, se dice que la discontinuidad es de primera especie con salto finito. Si la diferencia es infinita, se dice que la discontinuidad es de primera especie de salto infinito. Si alguno de los límites laterales no existe, se dice que la función es discontinua de segunda especie.

Como primer ejemplo se comprueba la continuidad en  $R - \{0\}$  de la función  $f(x) = \operatorname{Sen}(x)/x$ . Para ello comprobaremos que  $\lim_{x \rightarrow a} f(x) = f(a)$ .

```
>> syms x a
>> limit(sin(x)/x,x,a)
```

ans =

$\sin(a)/a$

El problema se presenta en el punto  $x=0$ , en el que la función  $f$  no está definida. Por lo tanto, la función es discontinua en  $x=0$ . Esta discontinuidad se evitaría definiendo la función en  $x=0$  con un valor igual a  $\lim_{x \rightarrow 0} f(x)$ .

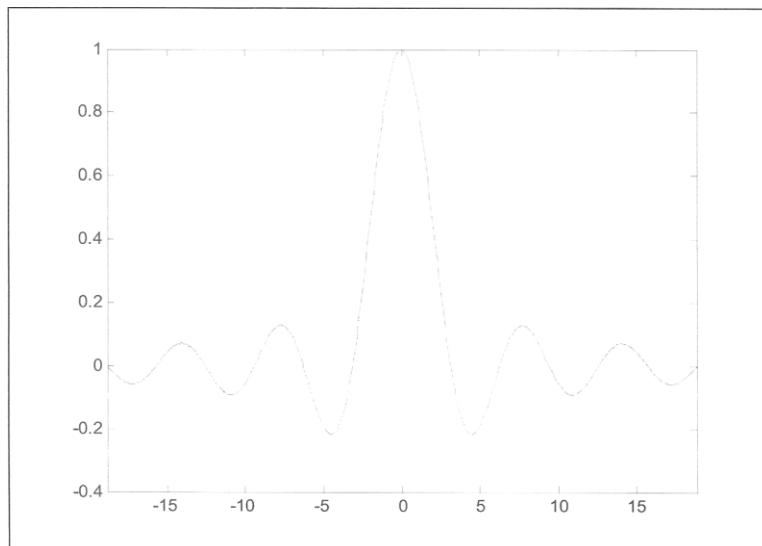
```
>> limit(sin(x)/x,x,0)
```

ans =

1

Por lo tanto, la función  $f(x)=\operatorname{Sen}(x)/x$  presenta una discontinuidad evitable en  $x=0$ , que se evita definiendo  $f(0)=1$ . En el resto de los puntos reales la función es continua. La gráfica de la función corrobora estos resultados.

```
>> fplot('sin(x)/x', [-6*pi, 6*pi])
```



Como segundo ejemplo se comprueba que la función  $f(x) = \sqrt[x]{e}$  no es continua en el punto  $x = 0$  ya que los límites laterales no coinciden (uno es cero y el otro infinito).

```
>> syms x  
  
>> limit((exp(1/x)),x,0,'right')  
  
ans =  
  
inf  
  
>> limit((exp(1/x)),x,0, 'left')  
  
ans =  
  
0
```

Como tercer ejemplo se estudia la continuidad de la función  $f(x)=\sin(1/x)$ . La función  $f$  es continua en cualquier punto  $x=a$  distinto de cero, ya que  $\lim_{x \rightarrow a} f(x) = f(a)$ :

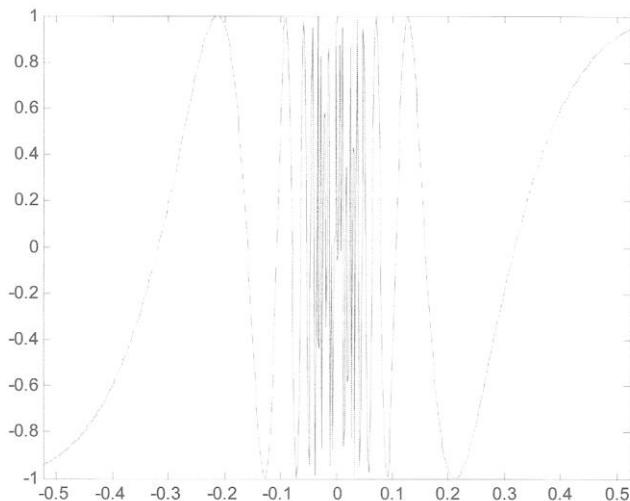
```
>> syms x a  
  
>> limit(sin(1/x),x,a)  
  
ans =  
  
sin(1/a)
```

Analicemos ahora el punto  $x=0$ , en el que la función  $f$  no está definida. Por lo tanto, la función es discontinua en  $x=0$ . Para intentar evitar la discontinuidad, calculamos:

```
>> limit(sin(1/x),x,0)  
  
ans =  
  
limit(sin(1/x), x = 0)  
>> limit(sin(1/x),x,0,'left')  
  
ans =  
  
NaN  
  
>> limit(sin(1/x),x,0,'right')  
  
ans =  
  
NaN
```

Vemos que en  $x=0$  no existe el límite, ni existe el límite por la izquierda, ni existe el límite por la derecha. Luego, la función presenta una discontinuidad de segunda especie en  $x=0$ . La representación gráfica corrobora este resultado.

```
>> fplot('sin(1/x)',[-pi/6,pi/6])
```



## LÍMITES EN VARIAS VARIABLES. LÍMITES ITERADOS Y DIRECCIONALES

La generalización del concepto de límite a varias variables es bastante sencilla. Se dice que la sucesión de puntos  $m$ -dimensionales  $\{(a_{1n}, a_{2n}, \dots, a_{mn})\}$  con  $n$  recorriendo los números naturales, tiene por límite el punto  $m$ -dimensional  $(a_1, a_2, \dots, a_m)$  si, y sólo si:

$$\lim_{n \rightarrow \infty} \{a_{1n}\} = a_1, \lim_{n \rightarrow \infty} \{a_{2n}\} = a_2, \dots, \lim_{n \rightarrow \infty} \{a_{mn}\} = a_m$$

Este teorema de caracterización permite calcular límites de sucesiones de puntos  $m$ -dimensionales.

Existe otro teorema semejante al anterior para la caracterización de los límites de funciones entre espacios de más de una dimensión. Este teorema se utilizará para posibilitar el cálculo de límites de funciones multivariadas.

Sea  $f: R^n \rightarrow R^m$  una función cuyas  $m$  componentes son  $(f_1, f_2, \dots, f_m)$ . Entonces se cumple que:

$\lim_{\substack{x_1 \rightarrow a_1, x_2 \rightarrow a_2, \dots, x_m \rightarrow a_m}} (f_1(x_1, x_2, \dots, x_m), f_2(x_1, x_2, \dots, x_m), \dots, f_n(x_1, x_2, \dots, x_n)) = (l_1, l_2, \dots, l_n)$  si y sólo si:

$\lim_{\substack{x_1 \rightarrow a_1, \dots, x_m \rightarrow a_m}} f_1(x_1, x_2, \dots, x_n) = l_1, \lim_{\substack{x_1 \rightarrow a_1, \dots, x_m \rightarrow a_m}} f_2(x_1, x_2, \dots, x_m) = l_2, \dots, \lim_{\substack{x_1 \rightarrow a_1, \dots, x_m \rightarrow a_m}} f_n(x_1, x_2, \dots, x_m) = l_n$

Como primer ejemplo calculamos el límite tridimensional siguiente:

$$\lim_{n \rightarrow \infty} \left[ \frac{1+n}{n}, \left(1 + \frac{1}{n}\right)^{2n}, \frac{n}{2n-1} \right]$$

```
>> syms n
>> V=[limit((n+1)/n,inf),limit((1+1/n)^(2*n),inf),limit(n/(2*n-1),inf)]
V =
[ 1, exp(2), 1/2]
```

Como segundo ejemplo calculamos el límite siguiente;

$$\lim_{n \rightarrow \infty} \left[ \sqrt[n]{\frac{n}{1+n^2}}, \sqrt[n]{\frac{1}{n}}, \sqrt[n]{5n}, \frac{1+n^2}{n^2} \right]$$

```
>> syms n
>> V1= [limit((n/(n^2+1))^(1/n),inf),limit((1/n)^(1/n),inf),
        limit((5*n)^(1/n),inf),limit((n^2+1)/n^2,inf)]
V1 =
[ 1, 1, 1, 1]
```

Como tercer ejemplo calculamos  $\lim_{x \rightarrow 0} f(x)$  para la función  $f: R \rightarrow R^2$  definida por:

```
f(x) = \frac{\sin(x)}{x}, \sqrt[x]{1+x}
>> syms x
>> V3= [limit(sin(x)/x,x,0),limit((1+x)^(1/x),x,0)]
V3 =
[1, exp(1)]
```

Como cuarto ejemplo calculamos  $\lim_{(x,y) \rightarrow (0,0)} f(x,y)$  para la función  $f: R \rightarrow R^2$  definida por:

```
f(x,y) = \left[ \frac{2(1-\cos(y))}{y^2} + \frac{\sin(x)}{x}, \sqrt[x]{1+x} - \frac{\tan(y)}{y} \right]
>> [limit(limit(sin(x)/x+2*(1-cos(y))/y^2,x,0),y,0)
      limit(limit(((1+x)^(1/x)-tan(y)/y),x,0),y,0)]
ans =
[ 2, exp(1) - 1 ]
```

Dada la función  $f: R^n \rightarrow R$ , se llama *límite iterado de  $f(x_1, x_2, \dots, x_n)$  en el punto  $(a_1, a_2, \dots, a_n)$*  al valor del límite, si existe:

$$\lim_{x_1 \rightarrow a_1} (\lim_{x_2 \rightarrow a_2} (\dots (\lim_{x_n \rightarrow a_n} f(x_1, x_2, \dots, x_n)) \dots))$$

o a cualquier otra ordenación de los límites en  $x_i$  con  $i=1, 2, \dots, n$ .

Se llama *límite direccional de  $f$  en el punto  $(a_1, a_2, \dots, a_n)$* , según la dirección de la curva  $h(t) = (h_1(t), h_2(t), \dots, h_n(t))$ , tal que  $h(t_0) = (a_1, a_2, \dots, a_n)$ , al valor:

$$\lim_{t \rightarrow t_0} (f(h(t))) = \lim_{(x_1, x_2, \dots, x_n) \rightarrow (a_1, a_2, \dots, a_n)} f(x_1, x_2, \dots, x_n)$$

Una condición necesaria para que una función de varias variables tenga límite en un punto, es que todos los límites iterados tengan el mismo valor (el valor del límite de la función, si éste existe). También puede ocurrir que la función tenga límite direccional según la dirección de varias curvas, que dicho límite sea distinto para distintas curvas, o bien que no exista para alguna de ellas.

Otra condición necesaria para que una función de varias variables tenga límite en un punto es que todos los límites direccionales, según las direcciones de todas las curvas, tengan el mismo valor (el valor del límite de la función, si éste existe). Por lo tanto, para demostrar que una función no tiene límite basta ver que los límites iterados no existen, o si existen, no son iguales. Tampoco la función tendrá límite si algún límite direccional no existe o si existen límites distintos para direcciones distintas. Otro procedimiento práctico para calcular el límite de una función de varias variables consiste en cambiar las variables cartesianas por variables polares, para facilitar así las operaciones.

Como primer ejemplo calculamos  $\lim f(x, y)$  para  $f: R^2 \rightarrow R$  definida por:

$$f(x, y) = \frac{xy}{x^2 + y^2}$$

```
>> syms x y
>> limit(limit((x*y) / (x^2+y^2), x, 0), y, 0)
ans =
0
>> limit(limit((x*y) / (x^2+y^2), y, 0), x, 0)
ans =
0
```

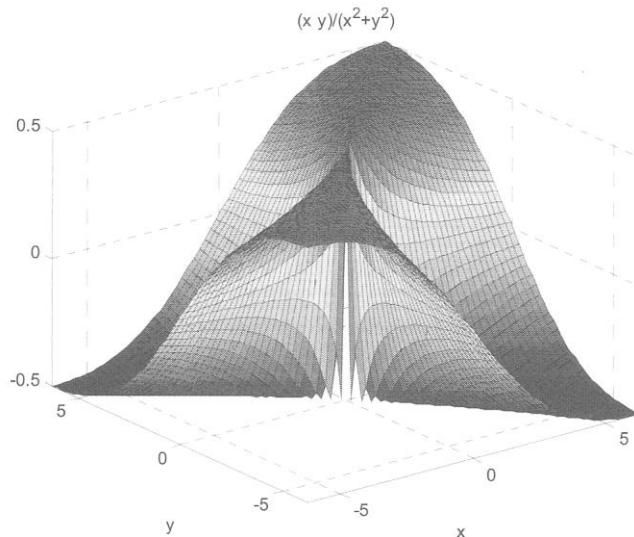
Se ha comprobado que los límites iterados son iguales. A continuación, calculamos los límites direccionales según la familia de rectas  $y=mx$ :

```
>> syms m
>> limit((m*x^2)/(x^2+m^2)*(x^2)), x, 0)
ans =
m/(1+m^2)
```

Se observa que los límites direccionales dependen del parámetro  $m$ , con lo que serán distintos para distintos valores de  $m$  (para las distintas rectas consideradas). Luego, se concluye que la función no tiene límite en  $(0,0)$ .

La representación gráfica de la superficie puede ilustrar el resultado.

```
>> ezsurf('(x*y)/(x^2+y^2)')
```



Se observa que en un entorno de  $(0,0)$  la función no tiene límite.

Como segundo ejemplo hallamos  $\lim_{(x,y) \rightarrow (0,0)} f(x,y)$  para la función  $f: R^2 \rightarrow R$  definida por:

$$f(x, y) = \frac{(y^2-x^2)^2}{x^2+y^4}$$

```
>> syms x y
>> limit(limit((y^2-x^2)^2/(y^4+x^2), x, 0), y, 0)
```

```

ans =
1
>> limit(limit((y^2-x^2)^2/(y^4+x^2),y,0),x,0)
ans =
0

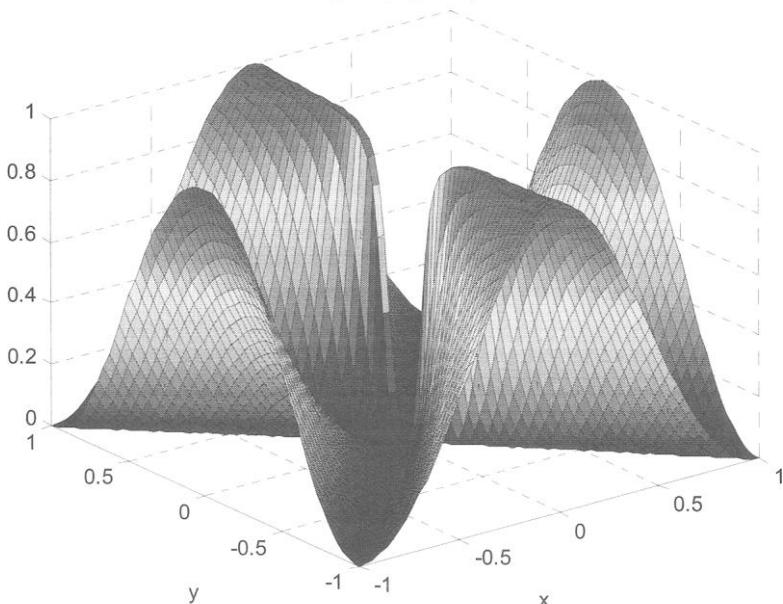
```

Como los límites iterados son distintos, se concluye que la función no tiene límite en el punto (0,0).

Si representamos la superficie en un entorno de (0,0) observamos su comportamiento en ese punto.

```
>> ezsurf ('(y^2-x^2)^2/(y^4+x^2)', [-1 1], [-1 1])
```

$$(y^2-x^2)^2/(y^4+x^2)$$



## CONTINUIDAD EN VARIAS VARIABLES

Generalizando la definición de continuidad en una variable podemos formalizar el concepto de continuidad en varias variables. Dada la función  $f: \mathbb{R}^n \rightarrow \mathbb{R}^m$ , se dice que es continua en el punto  $(a_1, a_2, \dots, a_n)$ , si se cumple que :

$$\lim_{\substack{x_1 \rightarrow a_1, x_2 \rightarrow a_2, \dots, x_n \rightarrow a_n}} f(x_1, x_2, \dots, x_n) = f(a_1, a_2, \dots, a_n)$$

Como primer ejemplo estudiamos la continuidad en el punto (1,2) de la función:

$$f(x, y) = x^2 + 2y \text{ si } (x, y) \neq (1, 2) \text{ y } f(1, 2) = 0$$

```
>> limit(limit((x^2+2*y),x,1),y,2)
ans =
5
>> limit(limit((x^2+2*y),y,2),x,1)
ans =
5
```

Vemos que el límite en  $(1,2)$ , de existir, ha de ser 5. Pero la función en el punto  $(1,2)$  vale 0. Luego, la función no es continua en el punto  $(1,2)$ .

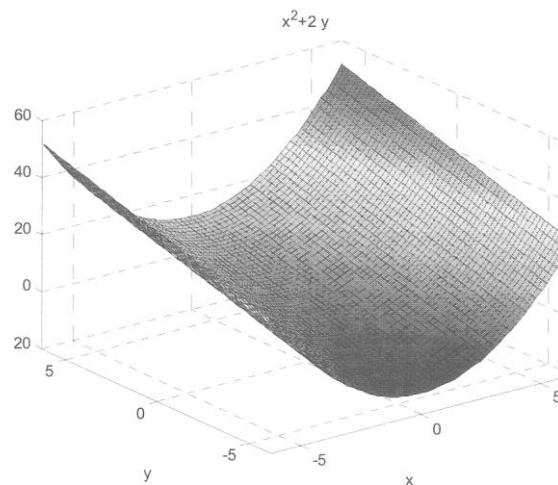
Como segundo ejemplo estudiamos la continuidad en todo de la función del ejemplo anterior.

```
>> syms x y a b
>> limit(limit((x^2+2*y),x,a),y,b)
ans =
a^2 + 2*b
>> limit(limit((x^2+2*y),y,b),x,a)
ans =
a^2 + 2*b
>> f='x^2+2*y'
f =
x^2+2*y
>> subs(f,{x,y},{a,b})
ans =
a^2 + 2*b
```

Hemos visto que los límites iterados coinciden y su valor coincide con el valor de la función en el punto genérico  $(a,b)$  de  $\mathbb{R}^2$ , es decir,  $\lim_{(x,y) \rightarrow (0,0)} f(x,y)$

Si realizamos la representación gráfica de la superficie, constatamos el hecho de que la función es continua en todo el campo de definición.

```
>> ezsurf ('x^2 + 2*y')
```



*Ejercicio 1. Calcular los siguientes límites:*

$$\lim_{n \rightarrow \infty} \frac{1+7n^2+3n^3}{5-8n+4n^3}, \quad \lim_{n \rightarrow \infty} \left[ \left( \frac{1+n}{2} \right)^4 \left( \frac{1+n}{n^5} \right) \right], \quad \lim_{n \rightarrow \infty} \sqrt[n]{\frac{1+n}{n^2}}$$

En el primer límite estamos ante la típica indeterminación dada por el cociente  $\infty/\infty$ :

```
>> syms n
```

```
>> limit(((2*n-3)/(3*n-7))^4, inf)
```

```
ans =
```

```
16/81
```

```
>> limit((3*n^3+7*n^2+1)/(4*n^3-8*n+5), n, inf)
```

```
ans =
```

```
3/4
```

Los dos últimos límites presentan indeterminaciones de tipos  $\infty \cdot 0$  e  $\infty^0$ :

```
>> limit(((n+1)/2)*((n^4+1)/n^5), inf)
```

```
ans =
```

```
1/2
```

```
>> limit(((n+1)/n^2)^(1/n), inf)
```

```
ans =
```

```
1
```

*Ejercicio 2. Calcular los siguientes límites:*

$$\lim_{x \rightarrow 2} \frac{x - \sqrt{2+x}}{-3 + \sqrt{1+4x}}, \lim_{x \rightarrow 0} \sqrt[x]{1+x}, \lim_{x \rightarrow 0} \frac{\lfloor \sin((ax)^2) \rfloor}{5 - 8n + 4n^3}, \lim_{x \rightarrow 0} \frac{e^x - 1}{\log(1+x)}$$

Inicialmente, tenemos dos indeterminaciones del tipo  $0/0$  y una de la forma  $1^\infty$ .

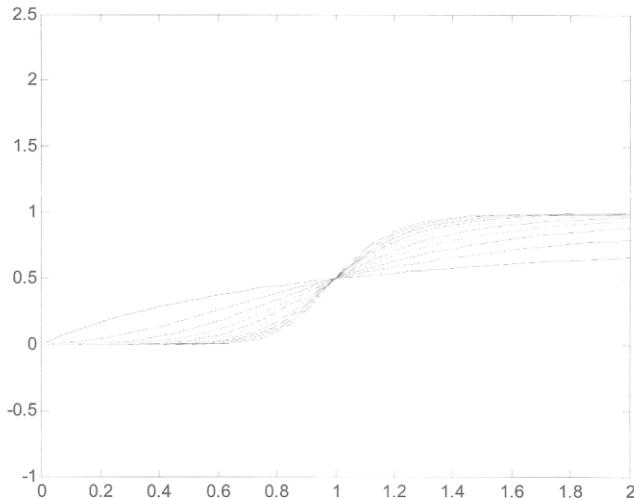
```
>> syms x
>> limit ((x - (x+2)^(1/2)) / ((4*x+1)^(1/2)-3), 2)
ans =
9/8
>> limit((1+x)^(1/x))
ans =
exp (1)
>> syms x a, limit(sin(a*x)^2/x^2, x, 0)
ans =
a^2
>> limit((exp(1)^x-1)/log(1+x))
ans =
log(3060513257434037/1125899906842624)
>> vpa(limit((exp(1)^x-1)/log(1+x)))
ans =
1.0000000000000001101889132838495
```

*Ejercicio 3. Calcular la función límite de la sucesión de funciones siguiente:*

$$g_n(x) = \frac{x^n}{(1+x^n)} \quad x > 0$$

Comenzaremos realizando una representación gráfica que de luz al posible resultado.

```
>> fplot ('[x/(1+x), x^2/(1+x^2), x^3/(1+x^3), x^4/(1+x^4), x^5/(1+x^5),
x^6/(1+x^6), x^7/(1+x^7), x^8/(1+x^8), x^9/(1+x^9), x^10/(1+x^10)]', [0, 1,
-1/2, 1])
```



Se intuye que la función límite es el eje OX (función  $f(x)=0$ ) en  $[0, 1]$ , el valor 0,5 en  $x=1$  y la función  $f(x)=1$  para  $x>1$ . Veamos:

```
>> vpa (simplify(limit(x^n/(1+x^n),n,inf)))
ans =
piecewise([x = 1.0, 0.5], [1.0 < x, 1.0], [x < 1.0, 0.0])
```

La salida de MATLAB nos dice que la función límite es:

$$f(x) = \begin{cases} 0.5 & \text{si } x = 1 \\ 1 & \text{si } x > 1 \\ 0 & \text{si } 0 < x < 1 \end{cases}$$

**Ejercicio 4.** Hallar  $\lim_{(x,y) \rightarrow (0,0)} f(x,y)$  para la función  $f: R^2 \rightarrow R$  definida por:

$$f(x,y) = \frac{(y^2-x)^2}{x^2+y^4}$$

```
>> syms x y m, limit(limit((y^2-x)^2/(y^4+x^2),y,0),x,0)
```

```
ans =
```

1

```
>> limit(limit((y^2-x)^2/(y^4+x^2),x,0),y,0)
ans =
1
```

Hemos visto que los límites iterados son iguales. A continuación, calculamos los límites direccionales según la familia de rectas  $y=mx$ :

```
>> limit(((m*x)^2-x)^2/((m*x)^4+x^2),x,0)
ans =
1
```

Los límites direccionales según la familia de rectas  $y=mx$  no dependen de  $m$  y coinciden con los límites iterados. Vamos a hallar el límite según las direcciones de la familia de parábolas  $y^2 = mx$ :

```
>> limit(((m*x)-x)^2/((m*x)^2+x^2),x,0)
ans =
(m - 1)^2/(m^2 + 1)
```

Los límites direccionales según las parábolas consideradas dependen del parámetro, luego, son distintos. Ello nos lleva a concluir que la función no tiene límite en  $(0,0)$ .

*Ejercicio 5. Hallar  $\lim f(x,y)$  para la función  $f:R^2 \rightarrow R$  definida por:*

$$f(x,y) = \frac{(x)^2y}{(x)^2+y^2}$$

```
>> syms x y, limit(limit((x^2*y)/(x^2+y^2),x,0),y,0)
ans =
0
>> limit(limit((x^2*y)/(x^2+y^2),x,0),y,0)
ans =
0
>> limit(((x^2)*(m*x))/(x^2+(m*x)^2),x,0)
ans =
0
>> limit (((m*y)^2)*y/((m*y)^2+y^2),y,0)
```

```
ans =
0
```

Vemos que los límites iterados y los límites direccionalas según las direcciones de rectas y paráolas coinciden y valen todos cero. Ello nos lleva a sospechar que el límite de la función puede ser cero. Para corroborarlo, transformamos la función a coordenadas polares y hallamos el límite en polares:

```
>> syms a r;
>> limit(limit(((r^2)*(cos(a)^2)*(r)*(sin(a)))/((r^2)*(cos(a)^2)+(r^2)
* (sin(a)^2)),r,0),a,0)
```

```
ans =
0
```

Se concluye por tanto que el límite es cero en el punto (0,0).

Este es un ejemplo en el que como último recurso se ha utilizado la transformación a coordenadas polares. Se han usado como direcciones familias de rectas y paráolas, pero podrían usarse otras. El cambio a polares resulta crucial a la hora de determinar límites de funciones de varias variables. Como se observa, hay criterios suficientes para ver que una función no tiene límite en un punto. Pero, sin embargo, no tenemos condiciones necesarias y suficientes para garantizar la existencia de dicho límite.

*Ejercicio 6. Hallar  $\lim f(x,y)$  para  $f:R^2 \rightarrow R$  definida por:*

$$f(x,y) = \frac{(x-1)^2y^2}{(x-1)^2+y^2}$$

```
>> syms x y m a r
>> limit (limit (y^2*(x-1)^2/(y^2+(x-1)^2),x,0),y,0)
ans =
0
>> limit(limit(y^2*(x-1)^2/(y^2+(x-1)^2),y,0),x,0)
ans =
0
>> limit((m*x)^2*(x-1)^2/((m*x)^2+(x-1)^2),x,0)
ans =
0
>> limit((m*x)*(x-1)^2/((m*x)+(x-1)^2),x,0)
ans =
0
```

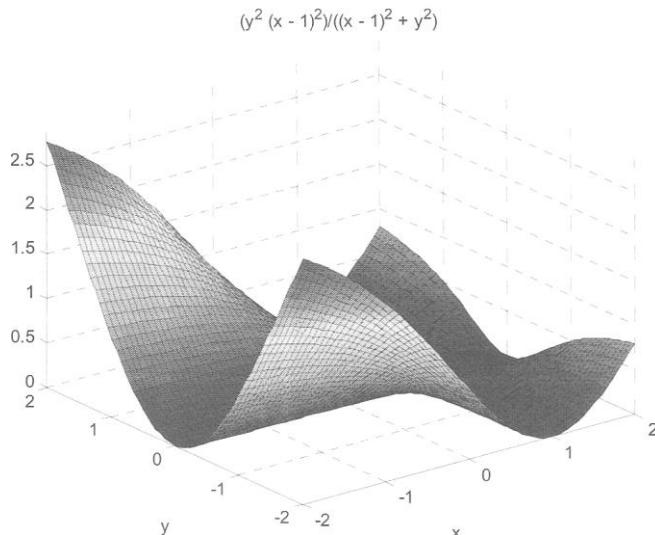
Vemos que los límites iterados y direccionales coinciden. Calculamos a continuación el límite en coordenadas polares:

```
>> limit(limit((r^2*sin(a)^2)*(r*cos(a)-1)^2/((r^2*sin(a)^2)
+ (r*cos(a)-1)^2), r, 1), a, 0)

ans =
0
```

El límite vale cero en el punto (1,0). A continuación se grafica la superficie, y se observa la tendencia a 0 en un entorno de (1,0):

```
>> ezsurf ('y^2*(x-1)^2/(y^2+(x-1)^2)', [-2,2,-2,2])
```



*Ejercicio 7. Estudiar la continuidad de la función de variable real:*

$$f(x, y) = \frac{1}{1+\sqrt[3]{e}} \text{ si } x \neq 0 \quad y \quad f(x) = 1 \text{ si } x = 0$$

El único punto problemático es  $x=0$ . Ahora, la función sí existe en  $x=0$  (vale 1).

Vamos a intentar hallar los límites laterales cuando  $x \rightarrow 0$ :

```
>> syms x

>> limit(1/(1+exp(1/x)), x, 0, 'right')

ans =

0

>> limit(1/(1+exp(1/x)), x, 0, 'left')
Warning: Could not attach the property of being close to the limit
point to limit variable
[limit]
```

```
ans =
```

```
1
```

Como los límites laterales son distintos, el límite de la función en  $x \rightarrow 0$  no existe. Pero como los límites laterales son finitos, la discontinuidad de primera especie en  $x=0$  es de salto finito. Se ilustra este resultado con la representación de la figura siguiente:

```
>> fplot('1/(1+exp(1/x))', [-5, 5])
```

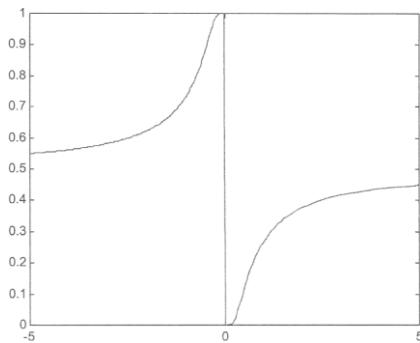


Figura 14-10

*Ejercicio 8. Sea  $f: R^2 \rightarrow R$  una función definida por:*

$$f(x, y) = \frac{[1 - \cos(x)]\sin(y)}{x^3 + y^3} \quad \text{si } (x, y) \neq (0, 0) \quad \text{y} \quad f(0, 0) = 0$$

*estudiar su continuidad en el punto  $(0, 0)$ .*

```
>> limit(limit((1-cos(x))*sin(y)/(x^3+y^3), x, 0), y, 0)
```

```
ans =
```

```
0
```

```
>> limit(limit((1-cos(x))*sin(y)/(x^3+y^3), y, 0), x, 0)
```

```
ans =
```

```
0
```

```
>> syms m
```

```
>> limit((1-cos(x))*sin(m*x)/(x^3+(m*x)^3), x, 0)
```

```
ans =
```

```
m/(2*(m^3 + 1))
```

Vemos que el límite en  $(0,0)$  no existe, pues hay límites direccionales distintos para direcciones distintas. Luego, la función no es continua en  $(0,0)$ .

Analicemos el resto de los puntos del plano.

```
>> syms x y a b m  
>> limit(limit((1-cos(x))*sin(y)/(x^3+y^3), x,a), y,b)  
ans =  
- (sin(b) * (cos(a) - 1))/(a^3 + b^3)  
>> limit(limit((1-cos(x)) *sin(y) / (x^3 + y^3) ,y,a) ,x,b)  
ans =  
- (sin(a)* (cos(b)- 1))/(a^3 + b^3)
```

Se observa que los límites iterados no coinciden, luego en un punto genérico  $(a,b)$  la función tampoco tiene límite y no es continua.

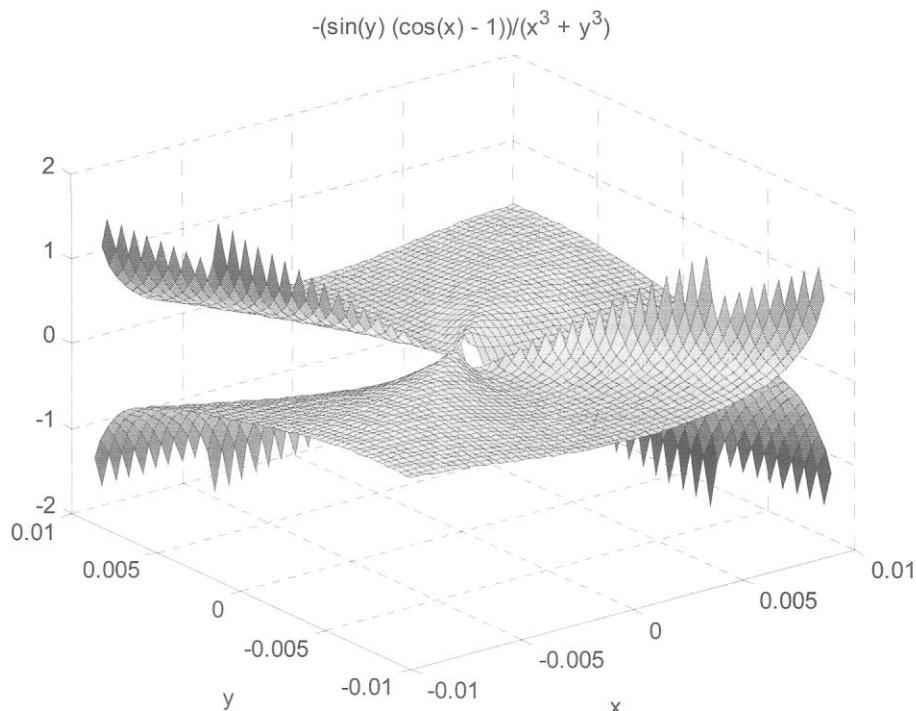
Los límites iterados serían iguales si  $\text{Sen}(a)\text{Cos}(b) = \text{Sen}(b)\text{Cos}(a)$ . Un caso particular que cumple esta condición es  $a=b$ , en cuyo caso los límites iterados coinciden y su valor coincide con el valor de la función en el punto.

```
>> limit(limit((1-cos(x))*sin(y)/(x^3+y^3) ,y,a),x,a)  
ans =  
- (sin(a) * (cos(a) - 1))/(2*a^3)  
>> limit(limit((1-cos(x))*sin(y)/(x^3+y^3) ,x,a),y,a)  
ans =  
- (sin(a)* (cos(a) - 1))/(2*a^3)  
>> f=(1-cos(x))*sin(y)/(x^3+y^3)  
f =  
- (sin(y)*(cos(x) - 1))/(x^3 + y^3)  
>> subs(f,{x,y},{a,a})  
ans =  
- (sin(a)* (cos(a) - 1))/(2*a^3)
```

La función puede ser continua en puntos del tipo  $(a,a)$ . Faltaría ver que los límites direccionales no dependen de ninguna dirección pasando la curva a coordenadas polares.

La representación gráfica de la función en un entorno del origen de radio una centésima que se presenta a continuación muestra que alrededor de (0,0) hay ramas de la superficie que se disparan al infinito, lo que provoca la no existencia del límite en el origen.

```
>> ezsurf ((1-cos(x))*sin(y)/(x^3+y^3), [-1/100,1/100,-1/100,1/100])
```

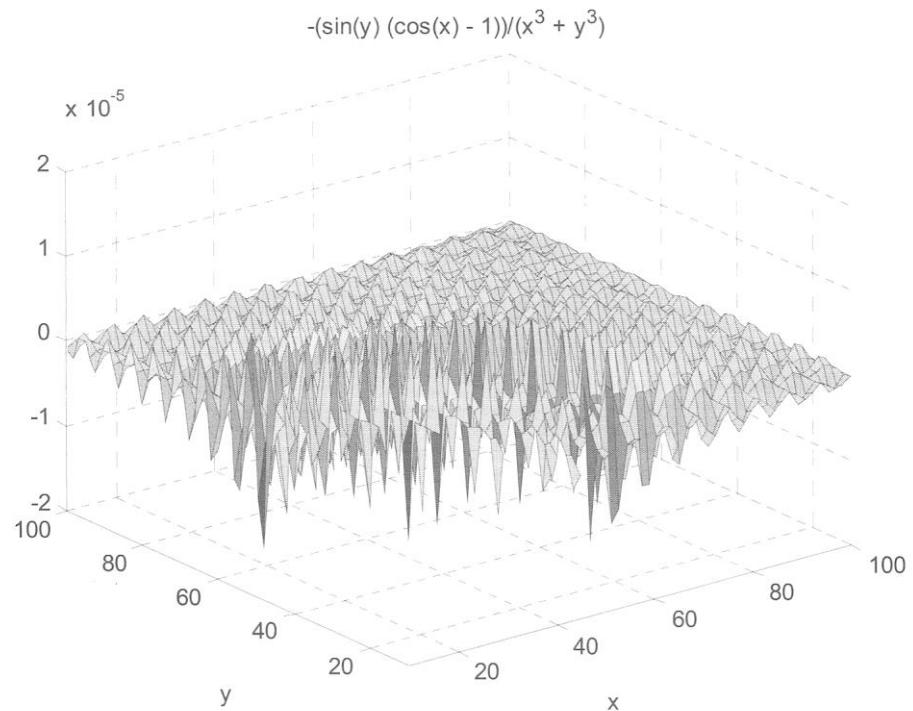


La representación gráfica de la función en una región que no contenga el origen se presenta a continuación y muestra que hay ramas de la superficie en las que no se puede asegurar la continuidad y otras zonas en las que sí.

Esto corrobora el resultado analítico que aseguraba la continuidad en puntos con coordenadas iguales y en otros puntos que cumplían la condición algebraica  $\text{Sen}(a)\text{Cos}(b) = \text{Sen}(b)\text{Cos}(a)$ .

A continuación se observa la representación gráfica de la superficie en la región [10,100]x[10,100].

```
>> ezsurf((1-cos(x))*sin(y)/(x^3+y^3), [10,100,10,100])
```



# SERIES NUMERICAS, SERIES DE POTENCIAS Y DESARROLLOS EN SERIE

---

## SERIES NUMERICAS DE TERMINOS NO NEGATIVOS

MATLAB permite trabajar con series numéricas, tanto de términos positivos como alternados. Por otra parte, los comandos relativos a límites nos permitirán trabajar con los diferentes criterios de convergencias de series numéricas.

Asimismo incorpora funciones para realizar su suma en caso de que sean convergentes. Tenemos los siguientes:

<b>symsum(S,v,a,b)</b>	<i>Suma la serie S para la variable v variando entre a y b</i>  syms x; symsum(x^k/sym('k!') , k, 0, inf)
<b>symsum(S,v)</b>	<i>Suma la serie S en la variable v variando de 0 a v-1</i>  » symsum(k^2,k) ans = k^3/S - k^2/2 + k/6
<b>r = symsum(S)</b>	<i>Suma la serie S respecto de su variable simbólica k (determinada por findsym) desde 0 hasta k-1</i>  » r=symsum(k^2) r = k^3/3 - k^2/2 + k/6

<pre><b>symsum(S, a,b)</b></pre>	<p>Suma la serie S respecto de su variable simbólica k (determinada por <code>findsym</code>) variando entre a y b</p> <pre>&gt;&gt; syms k n &gt;&gt; symsum(k, 0, n-1)  ans =</pre> $(n * (n - 1)) / 2$
----------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Criterios de convergencia: criterio del cociente

Existen varios criterios para determinar si una serie de términos positivos es convergente (suma finita). Entre los más comunes tenemos **el criterio del cociente o criterio de D'Alembert**, que dice lo siguiente:

$$\sum_{n=1}^{\infty} a(n) \text{ es convergente si } \lim_{n \rightarrow \infty} \frac{a(n+1)}{a(n)} < 1$$

$$\sum_{n=1}^{\infty} a(n) \text{ es divergente si } \lim_{n \rightarrow \infty} \frac{a(n+1)}{a(n)} > 1$$

En caso de que el límite sea 1, no se puede asegurar nada acerca del sentido de la convergencia de la serie.

Como primer ejemplo analizamos la convergencia de la serie  $\sum_{n=1}^{\infty} \frac{n}{2^n}$

Mediante el criterio del cociente calculamos  $\lim_{n \rightarrow \infty} \frac{a(n+1)}{a(n)}$

```
>> syms n
>> f='n/2^n'

f =

n/2^n

>> limit(subs(f,n+1)/subs(f,n),inf)

ans =

1/2
```

Como el límite es menor que la unidad, la serie es convergente y por lo tanto podremos calcular la suma de sus  $k$  primeros términos de la siguiente forma:

```
>> syms n k
>> symsum (n/2^n, 0, k-1)
ans =
2 - (2*(k + 1))/2^k
```

Si queremos obtener la suma infinita de la serie hacemos:

```
>> symsum(n/2^n, 0, inf)
ans =
2
```

Por lo tanto, ya podemos asegurar que:

$$\sum_{n=0}^{\infty} \frac{n}{2^n} = 2$$

Como segundo ejemplo analizamos la convergencia de la serie  $\sum_{n=1}^{\infty} \frac{n^n}{n!}$

Mediante el criterio del cociente calculamos  $\lim_{n \rightarrow \infty} \frac{a(n+1)}{a(n)}$

```
>> syms n
>> f=n^n/sym('n!')
f =
n^n/factorial(n)
>> limit(subs(f,n+1)/subs(f,n),inf)
ans =
exp(1)
```

Como el límite es mayor que la unidad, la serie es divergente y por lo tanto no es sumable.

### ***Criterio del Raabe***

El criterio de Raabe mejora el criterio de la raíz para analizar la convergencia de una serie. En caso de obtener límite 1 en el criterio del cociente u otro criterio semejante, suele utilizarse el *criterio de Raabe o Duhamel*, que dice lo siguiente:

$$\sum_{n=1}^{\infty} a(n) \text{ es convergente si } \lim_{n \rightarrow \infty} \left[ n \left( 1 - \frac{a(n+1)}{a(n)} \right) \right] > 1$$

$$\sum_{n=1}^{\infty} a(n) \text{ es divergente si } \lim_{n \rightarrow \infty} \left[ n \left( 1 - \frac{a(n+1)}{a(n)} \right) \right] < 1$$

En caso de que el límite sea 1, no se puede asegurar nada acerca del sentido de la convergencia de la serie.

Como ejemplo hallamos el carácter de la serie  $\sum_{n=1}^{\infty} \frac{1+n}{n(2+n)(3+n)}$

Como es lógico, inicialmente intentamos aplicar el criterio del cociente:

```
>> syms n  
» f='(1+n)/(n*(2+n)*(3+n))'  
f =  
(1+n)/(n*(2+n)*(3+n))  
>> limit(subs(f,n+1)/subs(f,n),inf)  
ans =  
1
```

Como el límite es 1 no podemos concluir nada respecto del carácter de la serie a través del criterio del cociente. Utilizamos entonces el criterio de Raabe.

```
>> limit(n*(1-subs(f,n+1))/subs(f,n),inf)  
ans =  
2
```

El límite es mayor que 1, luego la serie converge y será sumable. Hallamos su suma infinita de la siguiente forma:

```
>> symsum((1+n)/(n*(2+n)*(3+n)),1,inf)  
ans =
```

## Criterio de la raíz

El *criterio de la raíz* o *criterio de Cauchy* también mejora el criterio del cociente y a veces el criterio de Raabe para analizar la convergencia de una serie. En caso de obtener límite 1 en el criterio del cociente u otro criterio semejante, puede utilizarse el *criterio de la raíz*, que dice lo siguiente:

$$\sum_{n=1}^{\infty} a(n) \text{ es convergente si } \lim_{n \rightarrow \infty} \sqrt[n]{a(n)} < 1$$

$$\sum_{n=1}^{\infty} a(n) \text{ es divergente si } \lim_{n \rightarrow \infty} \sqrt[n]{a(n)} > 1$$

En caso de que el límite sea 1, no se puede asegurar nada acerca del sentido de la convergencia de la serie.

Como primer ejemplo analizamos el carácter de la serie  $\sum_{n=1}^{\infty} \frac{5}{2^n}$

Utilizamos el criterio de la raíz de la forma siguiente:

```
>> syms n
>> limit((5/2^n)^(1/n), inf)
ans =
1/2
```

Como el límite es menor que 1 la serie converge.

Si hubiésemos aplicado los criterios del cociente y Raabe tendríamos:

```
>> f='5/2^n'
f =
5/2^n
>> limit(subs(f,n+1)/subs(f,n), inf)
ans =
1/2
>> limit(n*(1-subs(f,n+1)/subs(f,n)), inf)
```

```
ans =  
Inf  
» limit((5/2^n)^(1/n),inf)  
ans =  
1/2
```

Se observa que mediante ambos criterios la serie resulta también convergente (límite menor que 1 para el criterio del cociente y límite mayor que 1 para el criterio de Raabe).

La serie sería sumable y su suma infinita se calcula mediante:

```
>> symsum(5/2^n,1,inf)  
ans =  
5
```

Como primer ejemplo analizamos el carácter de la serie:

$$\sum_{n=1}^{\infty} \tan^n(p + \frac{q}{n})$$

en función de los valores de los parámetros  $p$  y  $q$ .

Como su término general es una potencia enésima, es lógico intentar utilizar el criterio de la raíz. Tenemos:

```
>> syms n p q  
>> simplify(limit(tan(p+q/n),n,inf))  
ans =  
tan (p)
```

Luego, para valores de  $p$  tales que  $\tan(p) < 1$  la serie converge. Dichos valores cumplen  $0 < p < \pi/4$ . Y para valores de  $p$  tales que  $\tan(p) > 1$  la serie diverge. Dichos valores cumplen  $\pi/4 < p < \pi/2$ . MATLAB no ofrece el resultado exacto ni el aproximado de la suma de esta serie.

```
>> simplify(symsum(tan(p+q/n),n,1,inf))  
ans =  
sum(tan(p + q/n), n = 1..Inf)
```

## Otros criterios de convergencia

Existen otros criterios adicionales para estudiar la convergencia de series de términos positivos.

El *criterio de mayoración de Gauss* dice que toda serie mayorada por una convergente es convergente ( $a_n$  es mayorada por  $b_n$ , si  $a_n < b_n$  para todo  $n$ ). Además, toda serie minorada por una divergente es divergente ( $a_n$  es minorada por  $b_n$  si  $a_n > b_n$ , para todo  $n$ ).

El *criterio de comparación de segunda especie* asegura que si se compara en el límite por cociente una serie en estudio dada con una convergente o divergente conocida (la convergente o divergente en el denominador) y el límite de ese cociente es mayor que cero, las dos series tienen el mismo carácter.

Existe un tercer criterio que asegura que las series  $\sum_{n=1}^{\infty} a(n)$  y  $\sum_{n=1}^{\infty} 2^n a(2^n)$  tienen el mismo carácter.

Como primer ejemplo consideramos la serie siguiente:

$$\sum_{n=1}^{\infty} \frac{1}{(1 + \sqrt{n})^2}$$

Inicialmente intentamos aplicar los criterios de Raabe y del cociente:

```
>> f='1/(1+n^(1/2))-2'
f =
1/(1+n^(1/2))^2
>> limit(n*(1-subs(f,n+1)/subs(f,n)),inf)
ans =
1
>> limit(subs(f,n+1)/subs(f,n),inf)
ans =
1
```

Se observa que todos los límites son 1. Luego, de momento no puede asegurarse nada acerca de la convergencia de la serie.

Aplicamos ahora el criterio de comparación de segunda especie para comparar nuestra serie con la armónica divergente de término general  $1/n$ :

```
>> limit(subs(f,n)/(1/n),inf)
```

```
ans =
```

```
1
```

Como el límite es mayor que cero, la serie inicial es también divergente, y no es sumable.

Al mismo resultado se llega aplicando el criterio de la raíz.

```
>> limit(subs(f,n)^1/n,inf)
```

```
ans =
```

```
0
```

Al ser el límite menor que 1 en el criterio de la raíz, la serie diverge.

Como segundo ejemplo consideramos la serie siguiente:

$$\sum_{n=1}^{\infty} \frac{1}{n \log(n)}$$

Aplicamos el criterio que asegura que las series  $\sum_{n=1}^{\infty} a(n)$  y  $\sum_{n=1}^{\infty} 2^n a(2^n)$  tienen el mismo carácter.

```
>> f='1/(n*log(n))'
```

```
f =
```

```
1/(n*log(n))
```

```
>> 2^n*subs(f,2^n)
```

```
ans =
```

```
1/log(2^n)
```

Entonces nuestra serie tiene el mismo carácter que la serie de término general  $1/(n \log(2))$ , que está minorada por la serie armónica divergente de término general  $1/n$  ya que:

$$\frac{1}{n \log(2)} > \frac{1}{n}$$

Entonces la serie  $1/(n \log(2))$ , diverge y nuestra serie problema también.

## SERIES NUMÉRICAS ALTERNADAS. CRITERIOS DE DIRICHLET Y ABEL

Hasta ahora solamente hemos considerado series numéricas de términos positivos. A partir de ahora consideraremos series numéricas que tienen términos positivos y negativos. Habitualmente estas series se denominan series numéricas alternadas.

En el caso de las series alternadas es fundamental el concepto de convergencia absoluta. Se dice que la serie alternada  $\sum a(n)$  es absolutamente convergente si la serie de los módulos  $\sum |a(n)|$  es convergente. Como la serie de los módulos es una serie de términos positivos, ya sabemos analizarla. Por otra parte, toda serie absolutamente convergente es convergente.

Hay dos criterios clásicos que permiten analizar el carácter de series alternadas, y que nos permitirán salir de la mayoría de los apuros con series alternadas.

*El criterio de Dirichlet* dice que si  $\sum a(n)$  tiene la sucesión de sumas parciales acotadas y  $\{b(n)\}$  es una sucesión decreciente que tiene por límite 0, entonces la serie  $\sum a(n)b(n)$  es convergente.

*El criterio de Abel* dice que si  $\sum a(n)$  es convergente y  $\{b(n)\}$  es una sucesión monótona convergente, entonces la serie  $\sum a(n)b(n)$  es convergente.

Como primer ejemplo consideramos la serie alternada:

$$\sum_{n=1}^{\infty} \frac{(-1)^{1+n}}{1+2n^2}$$

Para analizar la convergencia de la serie vamos a estudiar si es o no absolutamente convergente, es decir, si es convergente la serie de términos positivos:

$$\sum_{n=1}^{\infty} \left| \frac{(-1)^{1+n}}{1+2n^2} \right|$$

Aplicamos a esta serie de términos positivos el criterio de comparación de segunda especie, comparándola con la armónica convergente de término general  $1/n^2$

```
>> f='abs((-1)^(1+n)/(1+2*n^2))'
```

```
f =
```

```
abs((-1)^(l+n)/(l+2*n^2))  
>> limit(subs(f,n)/(l/n^2),inf)  
ans =  
1/2
```

Como el límite es mayor que cero, la serie de términos positivos considerada es convergente, luego, la serie inicial es absolutamente convergente y, por lo tanto, convergente.

Como segundo ejemplo consideramos la serie:

$$\sum_{n=1}^{\infty} \frac{(-1)^{1+n}}{1+2n^2}$$

Haciendo  $a(n) = (-1)^{1+n}$  y  $b(n) = \frac{1}{n}$ , tenemos que  $\sum a(n)$  tiene las sumas parciales acotadas y que  $\{b(n)\}$  es monótona decreciente con límite 0.

Aplicando el criterio de Dirichlet se concluye que la serie alternada considerada es convergente.

## SERIES DE POTENCIAS

Una serie de potencias tiene la siguiente estructura:

$$\sum a(n)x^n$$

El objetivo fundamental es calcular el intervalo de convergencia de la serie, es decir, el intervalo de variación de  $x$  para el cual la serie numérica correspondiente es absolutamente convergente.

Al sustituir la variable  $x$  por sus valores de convergencia, la serie de potencias se convierte en una serie numérica. Los criterios a utilizar para hallar la convergencia son los ya utilizados para series numéricas. Los criterios más comunes a usar son los criterios de la raíz y del cociente, ya que para que una serie sea absolutamente convergente la serie de los módulos (valores absolutos) ha de ser convergente, y esta serie es de términos positivos.

Como primer ejemplo calculamos el intervalo de convergencia para la serie de potencias siguiente:

$$\sum_{n=1}^{\infty} \frac{(4)^{2n}}{n+2} (x-3)^n$$

A través del criterio del cociente vamos a intentar calcular el rango de variación de la variable  $x$  que hace convergente a la serie considerada.

```
>> f='(4^(2*n))*((x-3)^n)/(n+2)'

f =
(4^(2*n))*((x-3)^n)/(n+2)

>> limit(simplify(subs(f,n,n+1)/subs(f,n,n)), n, inf)

ans =
16*x - 48
```

La serie será convergente cuando  $|16x - 48| < 1$ . Para hallar los valores extremos del intervalo de convergencia resolvemos las ecuaciones siguientes;

```
>> [solve('16*x-48=1'), solve('16*x-48=-1')]

ans =
[ 49/16, 47/16]
```

Luego, la condición  $|16x - 48| < 1$  es equivalente a la siguiente:

$$47/16 < x < 49/16$$

Ya sabemos que en el intervalo anterior la serie es convergente. Ahora falta analizar el comportamiento de la serie en los extremos de ese intervalo. Hacemos ahora el análisis para  $x=49/16$ .

```
>> g1=simplify(subs(f,x,49/16))

g1 =
1/(n + 2)
```

Tenemos que analizar la serie numérica de términos positivos  $\sum_{n=1}^{\infty} \frac{1}{n+2}$

Observamos que los criterios del cociente y de Raabe dan límite 1 y por lo tanto hay que utilizar otro criterio alternativo

```
>> limit(simplify(subs(g1,n+1)/subs(g1,n)), n, inf)

ans =
```

```
1  
>> limit(n*(1-subs(g1,n+1)/subs(g1,n)),inf)  
ans =  
1
```

Vamos a aplicar el criterio de comparación de segunda especie, comparando la serie del problema con la armónica divergente de término general  $1/n$ :

```
>> limit(subs(g1,n)/(1/n),inf)  
ans =  
1
```

Como el límite obtenido es mayor que cero, la serie es divergente.

Analizamos ahora la serie del problema en el extremo  $x=47/16$ :

```
>> g2=simplify(subs(f,x,47/16))  
92 =  
(-1)^n/(n + 2)
```

Tenemos que analizar la serie alternada  $\sum_{n=1}^{\infty} \frac{-1^n}{n+2}$

Por el criterio de Dirichlet, la serie de término general  $(-1)^n$  tiene sus sumas parciales acotadas, y la sucesión de término general  $1/(n+2)$  es decreciente hacia 0. Luego, la serie alternada converge. El intervalo de convergencia de la serie de potencias ha resultado ser  $[47/16, 49/16]$ .

## DESARROLLOS EN SERIES DE POTENCIAS

MATLAB incorpora comandos que permiten abordar el problema de la aproximación local de una función real de variable real en un punto mediante la sustitución de la función inicial por otra más sencilla equivalente. Lo más frecuente es intentar sustituir una función arbitraria  $f(x)$  por un polinomio  $P_n(x)$ , llamado serie de potencias, tal que los valores de  $f(x)$  y  $P_n(x)$  sean lo más próximos posible para cada  $x$ . La serie de potencias polinómica  $P_n(x)$  se denomina *desarrollo en serie de potencias* de la función  $f$ .

La sintaxis de los comandos habilitados por MATLAB para trabajar con series de potencias se presenta en la tabla siguiente:

<b>taylor(f)</b>	Realiza el desarrollo de McLaurin de quinto orden para la función f. » syms x f = exp(x^2); » pretty(taylor(f)) $\frac{x^4}{2} + x^2 + 1$
<b>taylor(f, n)</b>	Realiza el desarrollo de McLaurin de orden n para la función f siendo n un número natural » pretty(taylor(f, 7)) $\frac{x^6}{6} + \frac{x^4}{2} + x^2 + 1$
<b>taylor(f, a)</b>	Realiza el desarrollo de Taylor de quinto orden para la función f centrado en el valor a (número real). Cuando a es natural es necesario utilizar la función taylor(f 5, a) » pretty(simplify(taylor(f, 1/2))) $\frac{\exp(-\frac{1}{4}) (2592x^5 - 2480x^4 + 2960x^3 + 1800x^2 + 250x + 2969)}{3840}$
<b>taylor(f, n, v)</b>	Realiza el desarrollo de McLaurin de orden n-1 para la función f en la variable v. » pretty(taylor(f, 3, p)) $\exp(p)^2 + \exp(p)^2 (2p^2 + 1) (p - x)^2 - 2p \exp(p)^2 (p - x)$
<b>taylor(f, n, v, a)</b>	Realiza el desarrollo de Taylor de orden n-1 para la función f en la variable v en un entorno del punto a. » pretty(taylor(f, 3, p, 2)) $\exp(4) + 4 \exp(4) (x - 2) + 9 \exp(4) (x - 2)^2$

Como primer ejemplo calculamos el polinomio de Taylor de grado 13 de la función  $\operatorname{senh}(x)$  en el punto  $x=0$  (polinomio de McLaurin).

```
>> syms x
>> f=sinh(x)
f =
sinh(x)
>> taylor(f,13)
ans =
```

```
x^11/39916800 + x^9/362880 + x^7/5040 + x^5/120 + x^3/6 + x
```

Como segundo ejemplo calculamos el polinomio de Taylor de grado 6 de la función  $1/(1+x)$  en el punto  $x=2$ .

```
>> syms x
>> f=1/(1+x)

f =
1/(x + 1)

>> pretty(taylor(f,6,2))


$$\frac{(x - 2)^2}{27} - \frac{x}{9} - \frac{(x - 2)^3}{81} + \frac{(x - 2)^4}{243} - \frac{(x - 2)^5}{729} + \frac{5}{9}$$

```

*Ejercicio 1. Estudiar la convergencia y hallar la suma, si es posible, para las siguientes series:*

$$\sum_{n=1}^{\infty} \frac{n^n}{3^n n!} \quad \sum_{n=1}^{\infty} \frac{(3+2n)}{7^n (1+n)}$$

Aplicamos el criterio del cociente para la primera serie:

```
>> syms n
>> f=n^n/(sym('n!')*(3^n))
f =
n^n/(3^n*factorial(n))
>> limit(subs(f,n+1)/subs(f,n),inf)
ans =
exp(1)/3
```

El límite es menor que 1, con lo que la serie resulta ser convergente. Por lo tanto podemos intentar calcular su suma exacta como sigue:

```
>> vpa(simplify(symsum(f,1,inf)))
ans =
1.6250941822939141659377675737628
```

Ahora aplicamos el criterio del cociente para la segunda serie:

```
>> f=(2*n+3)/(n*(n+1)*(7^n))
f =
```

```
(2*n + 3) / (7^n*n*(n + 1))
>> limit(subs(f,n+1)/subs(f,n),inf)
ans =
1/7
```

Como el límite es menor que 1, la serie es convergente y, por tanto, sumable. Vamos a calcular su suma. Matlab intenta devolver el resultado, por complicado que éste sea. En muchas ocasiones, el resultado se ofrece dependiendo de determinadas funciones del programa:

```
>> vpa(symsum(f,1,inf))
ans =
0.38339728069096678282849845975009
```

*Ejercicio 2. Estudiar la convergencia y hallar la suma, si es posible, para las siguientes series:*

$$\sum_{n=1}^{\infty} \frac{n}{p^n} \quad \sum_{n=1}^{\infty} \frac{(n+p)!}{p^n n! p!} \quad p = \text{parámetro real}$$

Aplicamos el criterio del cociente para la primera serie:

```
>> syms n;
>> p=sym('p','real');
>> f=n/p^n
f =
n/p^n
>> limit(subs(f,{n},{n+1})/subs(f,{n},{n}),n,inf)
ans =
1/p
```

Luego, si  $p > 1$ , la serie converge, si  $p < 1$ , la serie diverge y si  $p = 1$ , obtenemos la serie de término general  $n$ , que diverge. Cuando  $p$  es mayor que 1, hallamos la suma de la serie:

```
>> vpa(symsum(f,1,inf))
ans =
piecewise([1.0 < Re(n), n*zeta(n)])
```

Vamos a aplicar el criterio del cociente a la segunda serie:

```
>> f=sym('(n+p)!')/(sym('p!')*(sym('n!')*(p^n)))
f =
factorial(n + p) / (p^n * factorial(n) * factorial(p))
```

```
>> vpa(simplify(limit(subs(f,{n},{n+1})/subs(f,{n},{n}),n,inf)))  
ans =  
1/p
```

Luego, si  $p > 1$ , la serie converge, si  $p < 1$ , la serie diverge y si  $p=1$ , obtenemos la serie de término general  $n$ , que diverge. Cuando  $p$  es mayor que 1, hallamos la suma de la serie:

```
>> vpa(simplify(symsum(f,1,inf)))  
ans =  
numeric::sum(factorial(n + p) / (p^(1.0*n) * factorial(p)),  
p = 1..Inf)/factorial(n)
```

Se observa que Matlab no resuelve esta suma.

Luego, si  $p > 1$  las dos series convergen y si  $p < 1$  las series divergen. La primera serie tiene suma, pero la segunda no.

*Ejercicio 3. Estudiar la convergencia y halla la suma, si es posible para las siguientes series:*

$$\sum_{n=1}^{\infty} \left(1 + \frac{1}{n}\right)^{-n^2} \quad \sum_{n=1}^{\infty} \left[\left(\frac{1+n}{n}\right)^{1+n} - \frac{1+n}{n}\right]^{-n}$$

Para la primera serie aplicamos el criterio del cociente:

```
>> syms n  
>> f=(1+1/n)^(-n^2)  
f =  
1/(1/n + 1)^(-n^2)  
>> limit(subs(f,n+1)/subs(f,n),inf)  
ans =  
1/exp (1)
```

Como el límite es menor que 1, la serie converge y es sumable. Matlab puede sumar esta serie como sigue:

```
>> vpa(symsum(f,1,inf))  
ans =  
0.81741943329783346091309999437311  
Vamos a aplicar el criterio del cociente a la segunda serie:  
>> f=(((n+1)/n)^(n+1) - (n+1)/n)^(-n)
```

```
f =
1/(((n + 1)/n)^(n + 1) - (n + 1)/n)^n
>> limit(subs(f,n+1)/subs(f,n),inf)
ans =
1/(exp(1) - 1)
```

Como el límite es menor que 1, la serie converge y es sumable. La suma puede obtenerse con Matlab como sigue:

```
>> vpa(symsum(f,1,inf))
ans =
1.1745855750875866746226188496682
```

*Ejercicio 4. Estudiar la convergencia y hallar la suma, si es posible, para las siguientes series:*

$$\sum_{n=1}^{\infty} (\sqrt[n]{n-1})^n \quad \sum_{n=1}^{\infty} \left(\frac{n^2+2n+1}{n^2+n-1}\right)^{n^2}$$

Como el término general de la serie está elevado a una potencia de n, puede ser aplicable el criterio de la raíz en ambos casos.

Ahora aplicamos el criterio de la raíz a la primera serie:

```
>> f=(n^(1/n)-1)^n
f =
(n^(1/n) - 1)^n
>> limit (subs(f,n)^(1/n),inf)
ans =
0
```

Como el límite es menor que 1, la serie es convergente y sumable. La suma se calcula de la forma siguiente:

```
>> vpa(symsum(f,1,inf))
ans =
0.29759749220552960849982457076294
```

Ahora aplicamos el criterio de la raíz a la segunda serie:

```
>> syms n
>> f=((n^2+2*n+1)/(n^2+n-1))^(n^2)
f =
((n^2 + 2*n + 1)/(n^2 + n - 1))^(n^2)
```

```
>> limit(subs(f,n)^(1/n),inf)
ans =
exp(1)
```

Como el límite es mayor que 1, la serie diverge y no es sumable.

*Ejercicio 5. Estudiar la convergencia y hallar la suma, si es posible, para la siguiente serie:*

$$\sum_{n=1}^{\infty} \frac{(n+1)(n+2)}{n^5}$$

Intentamos aplicar los criterios de la raíz, cociente y Raabe:

```
>> f= (n+1)*(n+2)/n^5
f =
((n + 1)*(n + 2))/n^5
>> limit (subs (f,n)^(1/n),inf)
ans =
1
>> limit(subs(f,n+1)/subs(f,n),inf)
ans =
1
>> limit(n*(1-subs(f,n+1)/subs(f,n)),inf)
ans =
3
```

Los criterios de la raíz y del cociente no nos dicen nada, pero el de Raabe ya nos asegura la convergencia (límite menor que 1). Por lo tanto, la serie es sumable.

```
>> vpa(symsum(f,1,inf))
ans =
6.522882114579748712610480224049
```

También podemos analizar la serie del problema, aplicando directamente el criterio de comparación de segunda especie para comparar nuestra serie con la armónica convergente de término general  $1/n^3$ :

```
>> limit(f/(1/(n^3)),inf)
ans =
1
```

Como el límite es menor que 0, la serie es convergente.

*Ejercicio 6. Estudiar la convergencia y hallar la suma, si es posible, para la siguiente serie:*

$$\sum_{n=1}^{\infty} \frac{1}{n[\log(n)]^p} \quad p = \text{parámetro} > 0$$

Aplicamos el criterio que asegura que las series:

$\sum_{n=1}^{\infty} a(n)$  y  $\sum_{n=1}^{\infty} [2^n a(2^n)]$  tienen el mismo carácter.

Apliquemos ahora este criterio a nuestra serie:

```
> f=1/(n*(log(n))^p)
f =
1/(n*log(n)^p)
>> pretty((2^n)*subs(f,{n},{2^n}))
      1
-----
      n   p
log(2 )
```

Cuando  $p < 1$ , esta serie está minorada por la serie divergente armónica de término general  $= 1/n^p$ . Luego, diverge, y la inicial también.

Cuando  $p > 1$ , esta serie está mayorada por la serie convergente armónica de término general  $n^p = 1/n^p$ . Luego, converge, y la inicial, también.

Cuando  $p = 1$ , la serie se reduce a la serie:

$$\sum_{n=1}^{\infty} \frac{1}{n[\log(n)]}$$

que puede analizarse por el mismo criterio que la anterior.

```
>> f=1/(n*(log(n)))
f =
1/(n*log(n))
>> pretty((2^n)*subs(f,{n},{2^n}))
      1
-----
      n
log(2 )
```

Como esta serie está minorada por la armónica divergente de término general  $1/n$ , entonces diverge. Y la inicial también diverge.

*Ejercicio 7. Estudiar la convergencia de las siguiente serie:*

$$\sum_{n=1}^{\infty} \frac{-1^{1+n} n}{(1+n)^2}$$

Haciendo:

$$a(n) = -1^{1+n} \quad \text{y} \quad b(n) = \frac{n}{(1+n)^2}$$

tenemos que  $\sum a(n)$  tiene las sumas parciales acotadas y que  $\{b(n)\}$  es monótona decreciente con límite 0.

Aplicando el criterio de Dirichlet se concluye que la serie alternada considerada es convergente.

Alternativamente, podríamos haber considerado la serie de los módulos y estudiar su carácter. Observamos que los criterio de Raabe, cociente y raíz dan límite uno y no resuelven el problema.

```
>> f=n/(1+n)^2
f =
n/ (n + x)^2
>> limit(n*(1-subs(f,n+1)/subs(f,n)),inf)
ans =
1
>> limit(subs(f,n+1)/subs(f,n),inf)
ans =
1
>> limit (subs (f, n^(1/n)),inf)
ans =
1
```

Aplicamos ahora a esta serie de términos positivos el criterio de comparación de segunda especie, comparándola con la armónica convergente de término general  $1/n^2$

```
>> limit(subs(f,n)/(1/n^2),inf)
```

```
ans =
Inf
```

Como el límite es mayor que cero, la serie de términos positivos considerada es convergente, luego, la serie inicial es absolutamente convergente y, por lo tanto, convergente y sumable. Calculaos ahora su suma.

```
>> f=(-1)^n * (n/(n+1))^2
f =
((-1)^n*n) / (n + 1)^2
>> vpa(symsum(f,1,inf))
ans =
-0.12931985286416790881897546186484
```

*Ejercicio 8. Estudiar el intervalo de convergencia de la siguiente serie de potencias:*

$$\sum_{n=0}^{\infty} \frac{1}{(-5)^n} x^{2n-1}$$

Aplicamos el criterio de la raíz:

```
>> syms x n
>> f=x^(2*n+1)/(-5)^n
f =
x^(2*n + 1) / (-5)^n
>> limit(subs(f,{n},{n+1})/subs(f,{n},{n}),n,inf)
ans =
-x^2/5
```

La serie es absolutamente convergente cuando  $|x^2/5| < 1$ .

La condición  $|x^2/5| < 1$  es equivalente a  $-\sqrt{5} < x < \sqrt{5}$ . Luego, ya tenemos determinado el posible intervalo de convergencia de la serie de potencias. Vamos a analizar ahora los extremos:

```
>> pretty(simplify(subs(f,{x},(sqrt(5)))))
      1/2
      5
-----
      n
     (-1)
>> pretty(simplify(subs(f,{x},(-sqrt(5)))))
      n    1/2
     -(-1)   5
```

Ambas series son evidentemente series alternadas divergentes. Por lo tanto, el intervalo final de convergencia de la serie de potencias dada será el siguiente:  $-Sqrt(5) < x < Sqrt(5)$ .

*Ejercicio 9. Calcular el polinomio de McLaurin de grado 5 que desarrolla en serie de potencias la función  $\ln(x+1)$ . Hallar también el polinomio de McLaurin de grado 8.*

```
>> syms x
>> f=log(x+1)

f =
log(x + 1)

>> pretty(taylor(f))


$$\frac{x^5}{5} - \frac{x^4}{4} + \frac{x^3}{3} - \frac{x^2}{2}$$


>> pretty(taylor(f,8))


$$\frac{x^7}{7} - \frac{x^6}{6} + \frac{x^5}{5} - \frac{x^4}{4} + \frac{x^3}{3} - \frac{x^2}{2} + x$$

```

*Ejercicio 10. Calcular el polinomio de Tayor de grado 7 que desarrolla en serie de potencias la función  $1/(2-x)$  en el punto  $x=1$ . Hallar también el polinomio de Tayor de grado 8 que desarrolla en serie de potencias la función  $\operatorname{Sen}(x)$  en el punto  $x=2$ .*

```
>> syms x
>> f=1/(2-x)

f =
-1/(x - 2)

>> pretty(taylor(f,7,1))


$$x^2 + (x - 1)^3 + (x - 1)^4 + (x - 1)^5 + (x - 1)^6$$


>> f=sin(x);
>> pretty(taylor(f,8,pi))


$$\pi - x - \frac{(\pi - x)^3}{5} + \frac{(\pi - x)^5}{120} - \frac{(\pi - x)^7}{5040}$$

```

En ambos desarrollos se observa que el término al que correspondería el mayor grado, se anula.

# **DERIVACION EN UNA Y VARIAS VARIABLES. APLICACIONES**

---

## **DERIVADAS**

MATLAB permite trabajar con derivadas a través de un grupo de comandos que habilita especialmente para esta tarea tan importante en el mundo del cálculo y sus aplicaciones. La derivada de una función real en un punto mide el índice instantáneo de cambio de esa función en un entorno del punto; es decir, cómo cambia la variable dependiente como consecuencia de un pequeño cambio de la variable independiente. Geométricamente, la derivada de una función en un punto representa la pendiente de la tangente a dicha función en el punto. El origen de la idea de derivada proviene precisamente del intento de trazar la recta tangente en un punto dado a una curva dada.

La función  $f(x)$  definida en un entorno del punto  $x=a$  se dice derivable en  $a$  si existe:

$$\lim_{h \rightarrow 0} \frac{f(a+h)-f(a)}{h} = f'(a)$$

El valor del límite, si existe, se designa por  $f'(a)$ , y recibe el nombre de derivada de la función  $f$  en el punto  $a$ . Si  $f$  es derivable en todos los puntos de su dominio se dice simplemente que es derivable. La continuidad de una función es una condición necesaria para su derivabilidad, y toda función derivable es continua.

La tabla siguiente muestra los comandos básicos que habilita MATLAB para trabajar con derivadas.

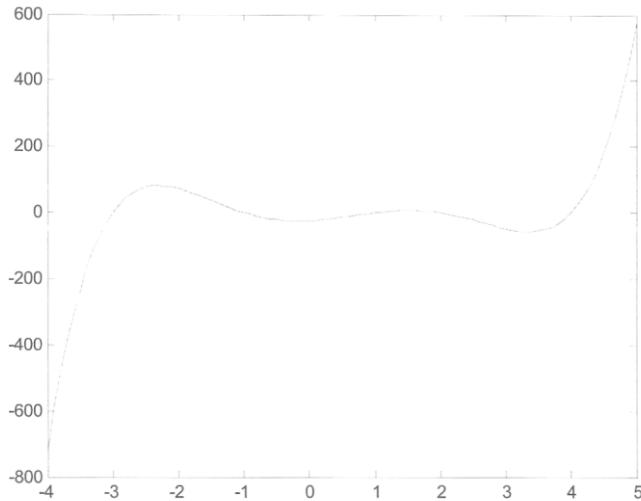
<b>diff('f', 'x')</b>	Halla la función derivada de $f$ respecto a $x$
-----------------------	-------------------------------------------------

	<pre>&gt;&gt; diff ('sin(x^2)', 'x') ans = 2*x*cos (x^2)</pre>
<b>syms x, diff(f,x)</b>	<p><i>Halla la función derivada de f respecto a x</i></p> <pre>&gt;&gt; syms x &gt;&gt; diff (sin (x^2) ,x) ans = 2*x*cos (x^2)</pre>
<b>diff('f', 'x', n)</b>	<p><i>Halla la función derivada enésima de f respecto a x</i></p> <pre>&gt;&gt; diff (sin(x^2)', 'x', 2) ans = 2*cos (x^2) - 4*x^2*sin (x^2)</pre>
<b>syms x, diff(f, x, n)</b>	<p><i>Halla la función derivada enésima de f respecto a x</i></p> <pre>&gt;&gt; syms x &gt;&gt; diff(sin(x^2),x,2) ans = 2*cos (x^2) - 4*x^2*sin (x^2)</pre>
<b>R = jacobian(w,v)</b>	<p><i>Halla la matriz jacobiana de w respecto de v</i></p> <pre>&gt;&gt; syms x y z &gt;&gt; jacobian([x*y*z; y; x+z], [x y z]) ans = [ y*z, x*z, x*y] [ 0, 1, 0] [ 1, 0, 1]</pre>
<b>Y = diff(X)</b>	<p><i>Calcula las diferencias entre elementos adyacentes del vector X: [X(2)-X(1), X(3)-X(2), X(n)-X(n-1)]. Si X es una matriz (m,n), diff(X) devuelve la matriz de diferencias por filas: [X(2:m,:)-X(1:m-1,:)]</i></p> <pre>x = [1 2 3 4 5] ; y = diff(x) y = 1      1      1      1</pre>
<b>Y = diff(X,n)</b>	<p><i>Hallar diferencias de orden n, por ejemplo: diff(X,2)=diff(diff(X))</i></p>

	<pre>x = [1 2 3 4 5] ; z = diff(x,2) z = 0     0     0</pre>
--	--------------------------------------------------------------

Como primer ejemplo consideramos la función  $f(x) = -3x^4 - 11x^3 + 27x^2 + 10x - 24$  y representamos su función derivada en el intervalo [-4,5].

```
>> x=-4:0.1:5;
>> f=x.^5-3*x.^4-11*x.^3+27*x.^2+10*x-24;
>> df=diff(f)./diff(x);
>> plot(x,f)
```



Como segundo ejemplo calculamos la derivada de la función  $\log(\sin(2x))$  y simplificamos totalmente el resultado.

```
>> pretty(simplify(diff('log(sin(2*x))','x')))

2 cot (2 x)
```

Como tercer ejemplo calculamos las cuatro primeras derivadas de la función  $f(x) = 1/x^2$

```
>> f='1/x^2'

f =
1/x^2
```

```
>> [diff(f),diff(f,2),diff(f,3),diff(f,4)]  
  
ans =  
[ -2/x^3, 6/x^4, -24/x^5, 120/x^6]
```

## DERIVADAS PARCIALES

Los comandos que aporta MATLAB para la derivabilidad vistos anteriormente, pueden utilizarse para realizar la derivación parcial.

Como ejemplo, dada la función  $f(x,y) = \operatorname{Sen}(xy) + \operatorname{Cos}(xy^2)$ , se calcula:

$\partial f / \partial x, \partial f / \partial y, \partial^2 f / \partial x^2, \partial^2 f / \partial y^2, \partial^2 f / \partial x \partial y, \partial^2 f / \partial y \partial x$  y  $\partial^4 f / \partial^2 x \partial^2 y$

```
>> syms x y  
>> f=sin(x*y)+cos(x*y^2)  
  
f =  
  
sin(x*y)+cos(x*y^2)  
  
>> diff(f,x)  
  
ans =  
  
cos (x*y)*y-sin(x*y^2)*y^2  
  
>> diff(f,y)  
  
ans =  
  
cos(x*y)*x-2*sin(x*y^2)*x*y  
  
>> diff{diff(f,x),x}  
  
ans =  
  
-sin(x*y)*y^2-cos(x*y^2)*y^4  
  
>> diff(diff(f,y),y)  
  
ans =  
  
-sin(x*y)*x^2-4*cos(x*y^2)*x^2*y^2-2*sin(x*y^2)*x  
  
>> diff(diff(f,x),y)  
  
ans =  
  
-sin(x*y)*x*y+cos(x*y) -2*cos(x*y^2)*x*y^3-2*sin(x*y^2)*y  
  
>> diff(diff(f,y),x)
```

```

ans =
-sin(x*y)*x*y+cos (x*y)-2*cos(x*y^2)*x*y^3-2*sin(x*y^2)*y
>> diff(diff(diff(f,x),x),y,y)
ans =
sin(x*y)*y^3*x-3*cos(x*y)*y^2 +2*cos(x*y^2)*y^7*x+6*sin(x*y^2)*y^5

```

## APLICACIONES DE LAS DERIVADAS. TANGENTES, ASÍNTOTAS, EXTREMOS Y PUNTOS DE INFLEXIÓN

Las aplicaciones directas de las derivadas permiten realizar el cálculo de tangentes a una función en un punto dado, hallar las asíntotas horizontales y oblicuas de una función, hallar los intervalos de crecimiento y concavidad, los máximos y mínimos y los puntos de inflexión.

Con toda esta información es posible estudiar y representar curvas de forma ordenada y completa.

Si  $f$  es una función para la cual existe  $f'(x_0)$ , entonces  $f'(x_0)$  es la pendiente de la línea tangente a la función  $f$  en el punto  $(x_0, f(x_0))$ . La ecuación de la tangente será  $y-f(x_0)=f'(x_0)(x-x_0)$ .

Las tangentes horizontales para las que  $x_0 \rightarrow \infty$  son las asíntotas horizontales de la curva  $y=f(x)$ . Su ecuación será  $y=f(x_0)$ , cuando  $x_0 \rightarrow \infty$ .

Las tangentes verticales para las que  $f(x_0) \rightarrow \infty$  son las asíntotas verticales de la curva  $y=f(x)$ . Su ecuación será  $x=x_0$ , siendo  $x_0$  un valor tal que  $\lim_{x \rightarrow x_0} f(x) = \infty$ .

Las asíntotas oblicuas de la curva  $y=f(x)$  en el punto  $x=x_0$  tienen la ecuación  $y=mx+n$ , donde

$$m = \lim_{x \rightarrow \infty} \frac{y}{x} \quad y \quad n = \lim_{x \rightarrow \infty} (y - mx)$$

Si  $f$  es una función para la cual existen  $f'(x_0)$  y  $f''(x_0)$ , entonces, si  $f'(x_0)=0$  y  $f''(x_0)<0$ , la función  $f$  presenta un máximo relativo en el punto  $(x_0, f(x_0))$ .

Si  $f$  es una función para la cual existen  $f'(x_0)$  y  $f''(x_0)$ , entonces, si  $f'(x_0)=0$  y  $f''(x_0)>0$ , la función  $f$  presenta un mínimo relativo en el punto  $(x_0, f(x_0))$ .

Si  $f$  es una función para la cual existen  $f'(x_0)$ ,  $f''(x_0)$  y  $f'''(x_0)$ , entonces, si  $f'(x_0)=0$  y  $f''(x_0)=0$  y  $f'''(x_0) \neq 0$ , la función  $f$  presenta un punto de inflexión en el punto  $(x_0, f(x_0))$ .

Supuesta la existencia de  $f'(x)$ , los valores de  $x$  para los cuales la función  $f$  es creciente, son aquellos para los cuales  $f'(x)$  es mayor que cero.

Supuesta la existencia de  $f'(x)$ , los valores de  $x$  para los cuales la función  $f$  es decreciente, son aquellos para los cuales  $f'(x)$  es menor que cero.

Supuesta la existencia de  $f''(x)$  y  $f'(x)$ , los valores de  $x$  para los cuales la función  $f$  es cóncava, son aquellos para los cuales  $f''(x)$  es mayor que cero.

Supuesta la existencia de  $f''(x)$  y  $f'(x)$ , los valores de  $x$  para los cuales la función  $f$  es convexa, son aquellos para los cuales  $f''(x)$  es menor que cero.

Como ejemplo realizamos un estudio completo de la función:

$$f(x) = \frac{x^3}{x^2 - 1}$$

Calculando las asíntotas, máximos, mínimos, puntos de inflexión, intervalos de crecimiento y decrecimiento e intervalos de concavidad y convexidad.

```
>> f = 'x ^ 3 / ( x ^ 2 - 1 ) ' ;  
f =  
x^3/(x^2-1)  
>> syms x, limit (x^3/(x^2-1), x,inf)  
ans =  
NaN
```

Por lo tanto, no hay asíntotas horizontales. Para ver si las hay verticales, analicemos los valores de  $x$  que hacen y infinito:

```
>> solve('x^2-1')  
ans =  
[ 1]  
[-1]
```

Las asíntotas verticales serán las rectas  $x=1$  y  $x=-1$ . Veamos ahora si existen asíntotas oblicuas:

```
>> limit (x^3/(x^2-1)/x, x,inf)  
ans =
```

```

1
>> limit(x^3/(x^2-1)-x, x, inf)
ans =
0

```

La recta  $y = x$  es la asíntota oblicua. Ahora, se analizarán los máximos y mínimos, puntos de inflexión e intervalos de crecimiento y concavidad:

```

>> solve(diff(f))
ans =
[      0]
[      0]
[ 3^(1/2)]
[ -3^(1/2)]

```

La primera derivada se anula en los puntos de abscisa  $x=0$ ,  $x=\sqrt{3}$  y  $x=-\sqrt{3}$ . Estos puntos son los candidatos a máximos y mínimos. Para corroborar si son máximos o mínimos, hallamos el valor de la segunda derivada en esos puntos:

```

>> [numeric(subs(diff(f,2),0)), numeric(subs(diff(f,2),sqrt(3))),
     numeric(subs(diff(f,2),-sqrt(3)))]
ans =
0      2.5981      -2.5981

```

Por lo tanto, en el punto de abscisa  $x=-\sqrt{3}$  hay un máximo y en el punto de abscisa  $x=\sqrt{3}$  hay un mínimo. En  $x=0$  no sabemos nada:

```

>> [numeric(subs(f,sqrt(3))), numeric(subs(f,-sqrt(3)))]
ans =
2.5981 -2.5981

```

Por lo tanto, el punto máximo es  $(-\sqrt{3}, -2.5981)$  y el punto mínimo es  $(\sqrt{3}, 2.5981)$ .

Vamos a analizar ahora los puntos de inflexión:

```

>> solve(diff(f,2))
ans =
[ 0]
[ i*3^(1/2)]
[-i*3^(1/2)]

```

El único punto de inflexión posible se presenta en  $x=0$ , y como  $f(0)=0$ , el punto de inflexión posible es  $(0,0)$ :

```
>> subs(diff(f,3),0)  
ans =  
-6
```

Como la tercera derivada en  $x=0$  no se anula, el origen es realmente un punto de inflexión:

```
>> pretty(simple(diff(f)))  
      2  
x ( x - 3 )  
-----  
      2      3  
(x - 1)
```

La curva será creciente cuando  $y' > 0$ , es decir, en los intervalos definidos por  $(-\infty, -\sqrt{3})$  y  $(\sqrt{3}, \infty)$ .

La curva será decreciente cuando  $y' < 0$ , es decir, en los intervalos definidos por  $(-\sqrt{3}, -1)$ ,  $(-1, 0)$ ,  $(0, 1)$  y  $(1, \sqrt{3})$ .

```
>> pretty(simple(diff(f,2)))  
      2  
x ( x + 3 )  
-----  
      2      3  
(x - 1)
```

La curva será cóncava cuando  $y'' > 0$ , es decir, en los intervalos  $(-1, 0)$  y  $(1, \infty)$ .

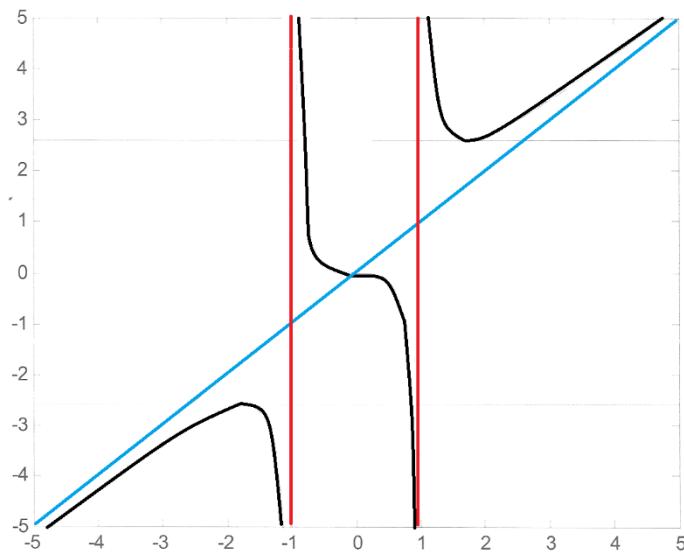
La curva será convexa cuando  $y'' < 0$ , es decir, en los intervalos  $(0, 1)$  y  $(-\infty, -1)$ .

La curva tiene tangentes horizontales en los tres puntos que anulan la primera derivada. Las ecuaciones de las tangentes horizontales son  $y=0$ ,  $y=2.5981$  e  $y=-2.5981$ .

La curva presenta tangentes verticales en los puntos que hacen infinita la primera derivada. Dichos puntos son  $x=1$  y  $x=-1$ . Por lo tanto, las tangentes verticales coinciden con las dos asíntotas verticales.

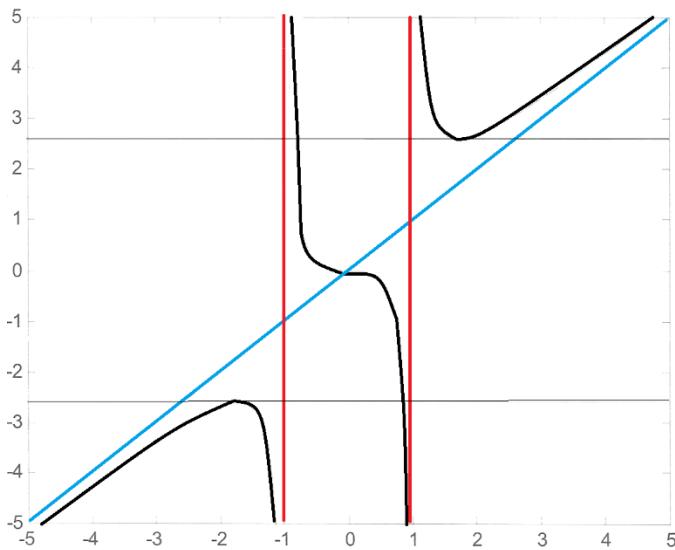
A continuación, representamos la curva junto con sus asíntotas:

```
>> fplot ('[x^3/(x^2-1), x]', [-5, 5, -5, 5])
```



También podemos representar la curva, sus asíntotas y sus tangentes horizontales y verticales en el mismo gráfico.

```
>> fplot('[x-3/(x^2-1), x, 2.5981, -2.5981]', [-5, 5, -5, 5])
```



## DERIVABILIDAD EN VARIAS VARIABLES

El concepto de derivada para funciones de una variable es generalizable a la derivabilidad en varias variables. A continuación se presenta el concepto de derivada parcial para el caso de dos variables.

Dada la función  $f: R^2 \rightarrow R$ , se define la *derivada parcial de f* con respecto a la variable  $x$  en el punto  $(a,b)$  de la siguiente forma:

$$\frac{\partial f}{\partial x}(a,b) = \lim_{h \rightarrow \infty} \frac{f(a+h,b) - f(a,b)}{h}$$

De la misma manera, se define la derivada parcial de  $f$  con respecto a la variable  $y$  en el punto  $(a,b)$  de la siguiente forma:

$$\frac{\partial f}{\partial y}(a,b) = \lim_{h \rightarrow \infty} \frac{f(a,b+h) - f(a,b)}{h}$$

Generalizando, podemos definir la derivada parcial respecto de cualquier variable para una función de  $n$  variables.

Dada la función  $f: R^n \rightarrow R$  se define la derivada parcial de  $f$  con respecto a la variable  $x_i$  ( $i=1,2,\dots,n$ ) en el punto  $(a_1,a_2,\dots,a_n)$  de la siguiente forma:

$$\frac{\partial f}{\partial x_i}(a_1,a_2,\dots,a_n) = \lim_{h \rightarrow \infty} \frac{f((a_1,a_2,\dots,a_i + h,\dots,a_n)) - f(a_1,a_2,\dots,a_n)}{h}$$

*La función f es diferenciable* si existen todas las derivadas parciales respecto de las  $x_i$  ( $i=1,2,\dots,n$ ) y son continuas.

Toda función diferenciable es continua, y si una función no es continua no puede ser diferenciable.

Se define la *derivada direccional de la función f* según el vector  $v=(v_1,v_2,\dots,v_n)$  como el siguiente producto escalar:

$$(Df)\mathbf{v} = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \cdot (v_1, v_2, \dots, v_n) = (\nabla f) \cdot v$$

$(\nabla f) = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right)$  se denomina vector gradiente de  $f$

La derivada direccional de la función  $f$  según el vector  $\mathbf{v}=(dx_1,dx_2,\dots,dx_n)$  se denomina *diferencial total de f*. Su valor será:

$$(Df)\mathbf{v} = \left( \frac{\partial f}{\partial x_1} dx_1 + \frac{\partial f}{\partial x_2} dx_2 + \cdots + \frac{\partial f}{\partial x_n} dx_n \right)$$

Toda función diferenciable es continua, y si una función no es continua no puede ser diferenciable.

A partir de los comandos de MATLAB para las derivadas que ya conocemos, se pueden calcular derivadas en varias variables de acuerdo a la siguiente tabla:

<b>diff(f(x,y,z,...),x)</b>	<i>Derivada parcial de f respecto a x</i>  <pre>&gt;&gt; syms x y z &gt;&gt; diff (x^2+y^2+z^2+x*y-x*z-y*z+1, z) ans = 2*z - v - x</pre>
<b>diff(f(x,y,z,...),x,n)</b>	<i>Derivada parcial enésima de f respecto a x</i>  <pre>&gt;&gt; diff (x^2+y^2+z^2+x*y-x*z-y*z+1, z, 2) ans =</pre>
<b>diff(f(x1,x2,x3,...),xj)</b>	<i>Derivada parcial de f respecto a xj</i>  <pre>&gt;&gt; diff (x^2+y^2 + z^2+x*y-x*z-y*z + 1, y) ans = y + 1 + x*y</pre>
<b>diff(f(x1,x2,x3,...),xj,n)</b>	<i>Derivada parcial enésima de f respecto a xj</i>  <pre>&gt;&gt; diff (x^2+y^2+z^2+x*y-x*z-y*z+1, y, 2) ans =</pre>
<b>diff(diff(f(x,y,z,...),x),y))</b>	<i>Derivada parcial de f respecto a x y a y sucesivamente</i>  <pre>&gt;&gt; diff (diff (x^2 + y^2+z^2+x*y-x*z-y*z+1, x), y) ans = 1</pre>

Como primer ejemplo estudiamos la diferenciabilidad y la continuidad de la función:

$$f(x, y) = \frac{2xy}{\sqrt{x^2+y^2}} \quad \text{si } (x, y) \neq (0,0) \text{ y } f(0,0) = 0$$

Para ver si la función es diferenciable es necesario comprobar si admite derivadas parciales continuas en todo punto. Consideramos cualquier punto distinto del origen y vamos a calcular la derivada parcial respecto a la variable  $x$ :

```
>> syms x y
>> pretty(simplify(diff((2*x*y)/(x^2+y^2)^(1/2), x)))
      3
      2 y
-----
      3
      -
      2
(x + y)^2
```

Ahora, vamos a ver si esta derivada parcial es continua en el origen. Al calcular los límites iterados en el origen observamos que no coinciden.

```
>> limit(limit((2*y-3/(x^2+y^2)^(3/2)), x, 0), y, 0)
ans =
NaN

>> limit(limit((2*y^3/(x^2+y^2)^(3/2)), y, 0), x, 0)
ans =
0
```

El límite de la derivada parcial no existe en  $(0,0)$ , luego se concluye que la función no es diferenciable.

Sin embargo, la función sí es continua, ya que el único punto problemático es el origen, y en él, el límite de la función vale  $0 = f(0,0)$ :

```
>> limit(limit((2*x*y)/(x^2+y^2)^(1/2), x, 0), y, 0)
ans =
0

>> limit(limit((2*x*y)/(x^2+y^2)^(1/2), y, 0), x, 0)
ans =
0
```

```

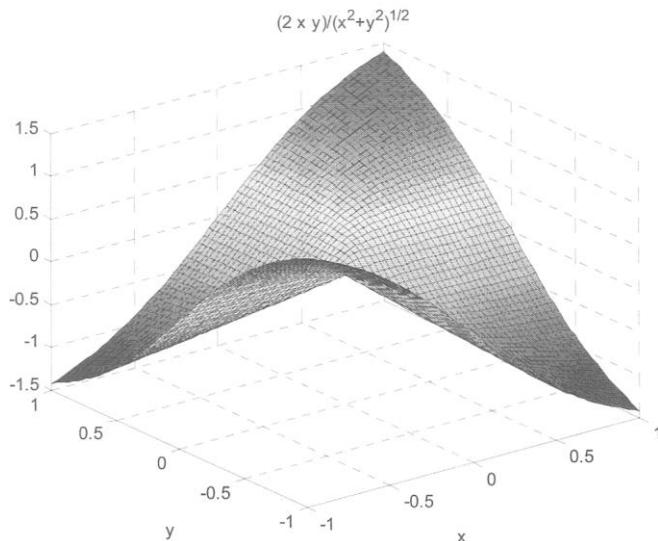
>> m = sym('m', 'positive')
>> limit((2*x*(m*x)) / (x-2+(m*x)-2)-(1/2),x,0)
ans =
0
>> a = sym('a', 'real');
>> f=(2*x*y) / (x-2+y-2)-(1/2);
>> limit(subs(f,{x,y},{r*cos(a),r*sin(a)}),r,0)
ans =

```

Los límites iterados y los direccionales valen todos cero, y al cambiar la función a coordenadas polares, el límite en el origen resulta ser cero, que coincide con el valor de la función en el origen.

Hemos visto un ejemplo de función continua no diferenciable.

La representación gráfica ayuda a la interpretación del resultado.



Los límites iterados y los direccionales valen todos cero, y al cambiar la función a coordenadas polares, el límite en el origen resulta ser cero, que coincide con el valor de la función en el origen.

Hemos visto un ejemplo de función continua no diferenciable.

Como segundo ejemplo comprobamos que la función:

$$f(x,y,z) = \frac{1}{\sqrt{x^2+y^2+z^2}}$$

verifica la ecuación:

$$\frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} + \frac{\partial^2 f}{\partial z^2} = 0$$

```
>> syms x y z
>> f=1/(x^2+y^2+z^2)^(1/2)
f =
1 / (x^2 + y^2 + z^2)^(1/2)

>> diff(f,x,2)+diff(f,y,2)+diff(f,z,2)

ans =
(3*x^2)/(x^2 + y^2 + z^2)^(5/2) - 3/(x^2 + y^2 + z^2)^(3/2) +
(3*y^2)/(x^2 + y^2 + z^2)^(5/2) + (3*z^2)/(x^2 + y^2 + z^2)^(5/2)

>> simplify(diff(f,x,2)+diff(f,y,2)+diff(f,z,2))

ans =
0
```

Como tercer ejemplo calculamos la derivada direccional de la función:

$$f(x,y,z) = \frac{1}{\sqrt{x^2+y^2+z^2}}$$

en el punto  $(2,1,1)$  según la dirección del vector  $v=(1,1,0)$ . Hallar también el vector gradiente de  $f$ .

Sabemos que se define la *derivada direccional de la función f* según el vector  $v=(v_1, v_2, \dots, v_n)$  como el siguiente producto escalar:

$$(\mathbf{D}f)v = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \cdot (v_1, v_2, \dots, v_n) = (\nabla f) \cdot v$$

$$(\nabla f) = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \text{ se denomina } \textit{vector gradiente de } f.$$

En primer lugar calculamos el gradiente de la función  $f$ .

```
>> syms x y z
>> f=1/(x^2+y^2+z^2)^(1/2)

f =
1/ (x^2 + y^2 + z^2)^(1/2)

>> Gradiente_f=simplify([diff(f,x),diff(f,y) ,diff(f,z)])

Gradiente_f =
[ -x/(x^2 + y^2 + z^2)^(3/2), -y/(x^2 + y^2 + z^2)^(3/2), -z/(x^2 +
y^2 + z^2)^(3/2)]
```

A continuación calculamos el vector gradient en el punto (2,1,1)

```
>> Gradiente_f_p=subs(Gradiente_f,{x,y,z},{2,1,1})

Gradiente_f_p =
-0.1361 -0.0680 -0.0680
```

Por último calculamos la derivada direccional.

```
>> Derivada_direccional_p=dot(Gradiente_f_p,[1,1,0])

Derivada_direccional_p =
-0.2041
```

## EXTREMOS EN VARIAS VARIABLES

MATLAB permite calcular de forma sencilla máximos y mínimos de funciones de varias variables.

Dada la función  $f: R^n \rightarrow R$ , que aplica el punto  $(x_1, x_2, \dots, x_n) \in R$  en el punto  $f(x_1, x_2, \dots, x_n) \in R$  se dice que presenta un extremo en el punto  $(a_1, a_2, \dots, a_n)$  si el vector gradiente:

$$(\nabla f) = \left( \frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \dots, \frac{\partial f}{\partial x_n} \right) \text{ se anula en el punto } (a_1, a_2, \dots, a_n).$$

Al igualar a cero todas las derivadas parciales y resolver el sistema resultante, se obtienen los posibles máximos y mínimos. Para saber de qué clase de extremo se trata, es necesario construir la **matriz Hessiana en el punto**, que tiene la siguiente estructura:

$$H = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \frac{\partial^2 f}{\partial x_2 \partial x_n} & \dots & \frac{\partial^2 f}{\partial x_n^2} \end{bmatrix}$$

Supongamos, en primer lugar, que el determinante de  $H$  es no nulo en el punto  $(a_1, a_2, \dots, a_n)$ . En este caso, se dice que el punto es no degenerado y, además, podemos dar las siguientes condiciones suficientes de extremo:

Si la matriz Hessiana en el punto  $(a_1, a_2, \dots, a_n)$  es definida positiva, entonces la función presenta un mínimo en ese punto. Si la matriz Hessiana en el punto  $(a_1, a_2, \dots, a_n)$  es definida negativa, entonces la función presenta un máximo en ese punto. En cualquier otro caso, la función presenta un punto silla en el punto  $(a_1, a_2, \dots, a_n)$ .

Si el determinante de  $H$  es nulo en el punto  $(a_1, a_2, \dots, a_n)$  se dice que el punto es no degenerado.

Como ejemplo hallamos y clasificamos los puntos extremos de la función:

$$f(x, y, z) = x^2 + xy + y^2 + z^2$$

Comenzamos hallando los posibles puntos extremos. Para ello, igualamos a cero las derivadas parciales respecto de todas las variables (componentes del vector gradiente de  $f$ ) y resolvemos el sistema resultante en las tres variables:

```
>> syms X Y z
>> f=x-2+y-2+z-2+x*y
f =
x-2 + x*y + y-2 + z-2
>> [x y z]=solve(diff(f,x), diff(f,y), diff(f,z), x, y, z)
x =
0
y =
0
z =
0
```

El Único punto extremo es el origen (0,0,0). Vamos a analizar de qué clase de extremo se trata. Para ello, calculamos la matriz hessiana y la expresamos como una función de  $x, y, z$ :

```
>> clear all
>> syms x y z
>>
f=x^2+y^2+z^2
+x*y

f =
x^2 + x*y + y^2 + z^2

>> diff(f,x)

ans =
2*x + y

>> H=simplify([diff(f,x,2),diff(diff(f,x),y),diff(diff(f,x),z);
diff(diff(f,y),x),diff(f,y,2),diff(diff(f,y),z);
diff(diff(f,z),x),diff(diff(f,z),y),diff(f,z,2)])
```

```
H =
[ 2, 1, 0]
[ 1, 2, 0]
[ 0, 0, 2]
```

```
>> def(H)
```

```
ans =
```

```
6
```

Hemos visto que la matriz hessiana es constante (no depende del punto en que se aplique), por lo tanto, su valor en el origen ya está hallado. El determinante es distinto de cero, con lo que no habrá extremos degenerados.

```
>> eig(H)

ans =
1
2
3
```

Se observa que la matriz hessiana en el origen es definida positiva, porque todos sus autovalores son positivos. Luego, podemos concluir que el origen es un mínimo de la función.

MATLAB incorpora adicionalmente comandos específicos para la optimización y la búsqueda de ceros de funciones de varias variables. La tabla siguiente presenta los más importantes.

<b>g = inline(expr)</b>	<i>Transforma en función la cadena expr</i> <pre>&gt;&gt; g = inline('t^2') g = Inline function: g(t) = t^2</pre>
<b>g = inline(expr,arg1,arg2,...)</b>	<i>Transforma en función la cadena expr con argumentos de entrada dados</i> <pre>&gt;&gt; g = inline('sin(2*pi*f + theta)', 'f', 'theta') g = Inline function: g(f, theta) = sin(2*pi*f + theta)</pre>
<b>g = inline(expr,n)</b>	<i>Transforma en función la cadena expr con n argumentos de entrada</i> <pre>&gt;&gt; g = inline('x^P1' , 1) g = Inline function: g(x, P1) = x^P1</pre>
<b>f = @función</b>	<i>Habilita la función para ser evaluada</i> <pre>&gt;&gt; f=@cos f = @cos &gt;&gt; ezplot(f, [-pi,pi])</pre>
<b>x = fminbnd(fun,x1,x2)</b>	<i>Minimiza la función en el intervalo (x1x2)</i> <pre>&gt;&gt; x=fminbnd(@cos,3,4) x = 3.1416</pre>
<b>x = fminbnd(fun,x1,x2,options)</b>	<i>Minimiza la función en el intervalo (x1x2) según la opción dada por optimset(...). Este último comando se explica más adelante</i> <pre>&gt;&gt; x = fminbnd(@cos,3,4,optimset('TolX-1e-12,'Display','off')) x = 3.1416</pre>
<b>X = fminbnd(fun,x1,x2,options,P1,P2,...)</b>	<i>Determina parámetros adicionales P1, P2, a pasar a la función objetivo fun(x,P1,P2, ...)</i>

<code>[x,fval] = fminbnd(...)</code>	<p><i>Da además el valor de la función objetivo en x</i></p> <pre>&gt;&gt; [x,fval] = fminbnd(@cos,3,4) x =     3.1416 fval =     -1.0000</pre>
<code>[x,fval,f] = fminbnd(...)</code>	<p><i>Da además un indicador de convergencia f (f&gt;0 indica convergencia a la solución, f&lt;0 no convergencia y f=0 n.º de pasos excedido)</i></p> <pre>&gt;&gt; [x,fval,f] = fminbnd (@cos,3,4)  x =     3.1416 fval =     -1.0000 f =     1</pre>
<code>[x,fval,f,output] = fminbnd(...)</code>	<p><i>Da además información sobre la optimización (output.algorithm da el algoritmo usado, output.funcCount da el número de evaluaciones de fun y output.iterations da el número de iteraciones)</i></p> <pre>&gt;&gt; [x,fval,f,output] = fminbnd(@cos,3,4) x =     3.1416 fval =     -1.0000 f =     1 output =     iterations: 7     funcCount: 8         algorithm: 'golden section search, parabolic interpolation'     message: [1x112 char]</pre>
<code>x = fminsearch(fun,x0)</code> <code>x = fminsearch(fun,x0,options)</code>	<p><i>Comandos idénticos a los anteriores para minimizar funciones de varias variables para</i></p>

<pre>x = fminsearch(fun,x0,options,P1,P2,...) [x,fval] = fminsearch(.,.) [x,fval,f] = fminsearch(... [x,fval,f,output] = fminsearch(...)</pre>	<p><i>Valores iniciales dados por <math>x_0</math>. El valor <math>x_0</math> pueda ser un intervalo <math>[a,b]</math> en el que se busca la solución. Luego, para minimizar <math>fun</math> en <math>[a,b]</math> se usa <math>x = fminsearch(fun,[a,b])</math></i></p> <pre>&gt;&gt; x=fminsearch(inline(' (100*(1-x^2)-2+(1- x)^2)'),3)  x = 1.0000 &gt;&gt;[x,feval]=fminsearch(inline(' (100*(1-x^2)^2 + (1-x)^2)'),3)  x = 1.0000 feval = 2.3901e-007 &gt;&gt;[x,feval,f]=fminsearch(inline(' (100*(1-x^2)^2 + (1-x)^2)'),3)  x = 1.0000 feval = 2.3901e-007 f = 1 &gt;&gt; [x,feval,f,output]=fminsearch(inline(' (100*(1- x^2)-2+(1-x)^2)'),3)  x = 1.0000 feval = 2.39010-007 f = 1 output = iterations: 18 funcCount: 36 algorithm: 'Nelder-Mead simplex direct search' message: [1x196 char]</pre>
------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

```
x = fzero(fun,x0)
x = fzero(fun,x0,options)
x = fzero(fun,x0,options,P1,P2,...)
[x,fval] = fzero(...)
[x,fval,exitflag] = fzero(...)
[x,fval,exitflag,output] = fzero(...)
```

Comandos idénticos a los anteriores para encontrar ceros de funciones. El valor  $x_0$  puede ser un intervalo  $[a,b]$  en el que se busca la solución. Luego, para hallar un cero de  $fun$  en  $[a, b]$  se puede usar  $x = fzero(fun,[a,b])$ , donde  $fun$  tiene signo contrario en  $a$  y  $b$ .

```
>> x = fzero (@cos, [1 2])
x =
1.5708

>> [x,feval] = fzero(@cos,[1 2])
x =
1.5708
feval =
6.12320-      017

>> [x,feval,exitflag] = fzero(@cos,[1 2])
x =
1.5708
feval =
6.1232e-017
exitflag =
1
>> [x,feval,exitflag,output] = fzero(@cos,[1 2])
x =
1.5708
feval =
6.12320-      017
exitflag =
1
output =
    iterations: 0
    iterations: 5
    funcCount: 7
    algorithm:'bisection,
    interpolation'message: 'Zero found in the
    interval [1, 2]'
```

<b>options = optimset('p1',v1,'p2',v2,...)</b>	<i>Crea opciones de optimización para los parámetros p1, p2, ... con valores v1, v2, ... Los parámetros posibles son Display (con valores posibles 'off'   'iter'   'final'   'notify' para no ver la salida, ver la salida de cada iteración, ver sólo la salida final y obtener mensaje si no hay convergencia); MaxFunEvals, cuyo valor es un entero que indica el máximo número de evaluaciones; MaxIter cuyo valor es un entero que indica el máximo número de iteraciones; TolFun, cuyo valor es un entero que indica la tolerancia en el valor de la función. y TolX, cuyo valor es un entero que indica la tolerancia en el valor de x</i>
<b>val = optimget(options,'param')</b>	<i>Devuelve el valor del parámetro especificado en la estructura de optimización options</i>

Como primer ejemplo minimizamos la función  $\cos(x)$  en el intervalo (3,4).

```
>> x = fminbnd(inline('cos(x)'), 3, 4)

x =
3.1416
```

En el ejemplo siguiente realizamos la minimización anterior con una tolerancia de 8 decimales y hallamos tanto el valor de x que minimiza el coseno en el intervalo dado como el valor mínimo de la función coseno en ese intervalo presentando información relativa a todas las iteraciones del proceso.

```
>> [x,fval,f]=fminbnd(@cos,3,4,optimset('TolX',1e-8, ...
'Display','iter'));

Func-count      x          f (x)      Procedure
    1      3.38197    -0.971249    initial
    2      3.61803    -0.888633    golden
    3      3.23607    -0.995541    golden
    4      3.13571    -0.999983    parabolic
    5      3.1413          -1        parabolic
    6      3.14159          -1        parabolic
    7      3.14159          -1        parabolic
    8      3.14159          -1        parabolic
    9      3.14159          -1        parabolic

Optimization terminated successfully:
the current x satisfies the termination criteria using OPTIONS.TolX
of 1.000000e-008
```

En el ejemplo siguiente, tomando como valores iniciales (-1,2;1), minimizamos y hallamos el valor objetivo de la función de dos variables;

$$f(x) = 100(x_2 - x_1^2)^2 + (1-x_1)^2$$

```
>> [x,fval] = fminsearch(inline('100*(x(2)-x(1)^2)^2+...
(1-x(1))^2'), [-1.2, 1])
x =
1.0000 1.0000
fval =
8.1777e-010
```

En el ejemplo siguiente se calcula un cero de la función seno cerca de 3 y un cero de la función coseno entre 1 y 2.

```
>> x = fzero(gsin,3)
x =
3.1416
>> x = fzero(@cos,[1 2])
X =
1.5708
```

## **EXTREMOS CONDICIONADOS. EL MÉTODO DE “LOS MULTIPLICADORES DE LAGRANGE”**

Supongamos que queremos optimizar (maximizar o minimizar) la función denominada función objetivo, pero sujeta a unas determinadas restricciones dadas por las ecuaciones:

$$\left| \begin{array}{l} g_1(x_1, x_2, \dots, x_n) = 0 \\ g_2(x_1, x_2, \dots, x_n) = 0 \\ G_4(x_1, x_2, \dots, x_n) = 0 \end{array} \right.$$

En este caso se plantea la función de Lagrange, que es una combinación lineal de la función objetivo y las restricciones, y que tiene la siguiente forma:

$$L(x_1, x_2, \dots, x_n, \lambda) = f(x_1, x_2, \dots, x_n) + \sum_{i=1}^k \lambda_i g_i(x_1, x_2, \dots, x_n)$$

Los posibles puntos extremos se obtienen de la resolución del sistema resultante de igualar a cero las componentes del vector gradiente de L, esto es,  $\nabla L(x_1, x_2, \dots, x_n, \lambda) = (0, 0, \dots, 0)$ . Lo que se traduce en:

$$\nabla L = \left( \frac{\partial L}{\partial x_1}, \frac{\partial L}{\partial x_2}, \frac{\partial L}{\partial x_n}, \frac{\partial L}{\partial \lambda_1}, \frac{\partial L}{\partial \lambda_2}, \dots, \frac{\partial L}{\partial \lambda_n} \right) = (0, 0, \dots, 0)$$

Al igualar a cero todas las derivadas parciales y resolver el sistema resultante, se obtienen los valores de  $x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_n$  correspondientes a posibles máximos y mínimos.

Para comprobar si los puntos obtenidos anteriormente  $(x_1, x_2, \dots, x_n)$  son una determinada clase de extremo, se utiliza la siguiente matriz:

$$\begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \dots & \frac{\partial^2 f}{\partial x_1 \partial x_n} & \frac{\partial g_i}{\partial x_1} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \dots & \frac{\partial^2 f}{\partial x_2 \partial x_n} & \frac{\partial g_i}{\partial x_2} \\ \dots & \dots & \dots & \dots & \dots \\ \frac{\partial^2 f}{\partial x_1 \partial x_n} & \frac{\partial^2 f}{\partial x_2 \partial x_n} & \dots & \frac{\partial^2 f}{\partial x_n^2} & \frac{\partial g_i}{\partial x_n} \\ \frac{\partial g_i}{\partial x_1} & \frac{\partial g_i}{\partial x_2} & \dots & \frac{\partial g_i}{\partial x_n} & 0 \end{bmatrix}$$

Sean:

$$H1 = \begin{bmatrix} \frac{\partial f}{\partial x_1^2} & \frac{\partial g_i}{\partial x_1} \\ \frac{\partial g_i}{\partial x_1} & 0 \end{bmatrix} \quad H2 = \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial g_i}{\partial x_1} \\ \frac{\partial^2 f}{\partial x_1 \partial x_2} & \frac{\partial^2 f}{\partial x_2^2} & \frac{\partial g_i}{\partial x_2} \\ \frac{\partial g_i}{\partial x_1} & \frac{\partial g_i}{\partial x_2} & 0 \end{bmatrix} \dots Hn = H$$

Si  $H1 < 0, H2 < 0, H3 < 0, \dots, H < 0$  tenemos un mínimo.

Si  $H1 < 0, H2 > 0, H3 < 0, \dots$ , tenemos un máximo.

Estas condiciones son válidas para cualquier restricción  $g_i(x_1, \dots, x_n)$  ( $i=1,2,\dots,k$ ) considerada. Cuando se consideran varias restricciones simultáneamente, es más cómodo determinar el carácter del punto extremo por simple inspección sustituyendo el punto en la función objetivo.

Como ejemplo hallamos y clasificamos los puntos extremos de la función:

$$f(x, y, z) = x + z$$

sujeta a la restricción:

$$x^2 + y^2 + z^2 = 1$$

En primer lugar planteamos la función de Lagrange  $L$ , que es una combinación lineal de la función objetivo y las restricciones:

```
>> syms x y z L p
>> f=x+z
f =
x + z
>> g=x^2+y^2+z^2-1
g =
x^2 + y^2 + z^2 - 1
>> L=f+p*g
L =
x + z + p*(x^2 + y^2 + z^2 - 1)
```

A continuación, los posibles puntos extremos se obtienen de la resolución del sistema resultante de igualar a cero las componentes del vector gradiente de  $L$ , esto es,  $\nabla L(x_1, x_2, \dots, x_n, \lambda) = (0, 0, \dots, 0)$ . Lo que se traduce en:

```
> [x,y,z,p]=solve(diff(L,x),diff(L,y),diff(L,z),diff(L,p),x,y,z,p)
x =
-2^(1/2)/2
2^(1/2)/2
y =
2^(1/2)/2
-2^(1/2)/2
```

```
z =
0
0

p =
2^(1/2)/2
-2^(1/2)/2
```

Al igualar a cero todas las derivadas parciales y resolver el sistema resultante, se obtienen los valores de  $x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_n$  correspondientes a posibles máximos y mínimos.

Ya tenemos que los posibles puntos extremos son:

$$(-\sqrt{2}/2, \sqrt{2}/2, 0) \text{ y } (\sqrt{2}/2, -\sqrt{2}/2, 0)$$

Ahora, vamos a ver qué tipo de extremos son. Para ello, los sustituimos en la función objetivo.

```
>> clear all
>> syms x y z
>> f=x+z

f =
x + z

>> subs(f, {x,y,z},{-sqrt(2)/2,sqrt(2)/2,0})

ans =
-0.7071

>> subs(f, {x,y,z},{sqrt(2)/2,-sqrt(2)/2,0})

ans =
0.7071
```

Luego, en el punto  $(-\sqrt{2}/2, \sqrt{2}/2, 0)$  la función presenta un máximo, y en el punto  $(\sqrt{2}/2, -\sqrt{2}/2, 0)$  presenta un mínimo.

## CÁLCULO DIFERENCIAL VECTORIAL

Nos ocuparemos aquí de cuatro teoremas clásicos en el cálculo diferencial en varias variables. Se trata de la regla de la cadena o teorema de la función compuesta, el teorema de la función implícita, el teorema de la función inversa y el teorema del cambio de variable.

Dada la función  $\bar{F} : \mathbb{R}^m \rightarrow \mathbb{R}^n$ :

$$(x_1, x_2, \dots, x_m) \rightarrow [F_1(x_1, x_2, \dots, x_m), \dots, F_n(x_1, x_2, \dots, x_m)]$$

se dice que la función vectorial  $\bar{F}$  es diferenciable en el punto  $a = (a_1, \dots, a_m)$  si lo son sus  $n$  componentes  $F_1, F_2, \dots, F_n$ .

Para la función anterior se define la matriz jacobiana como:

$$J = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \frac{\partial F_1}{\partial x_2} & \dots & \frac{\partial F_1}{\partial x_n} \\ \frac{\partial F_2}{\partial x_1} & \frac{\partial F_2}{\partial x_2} & \dots & \frac{\partial F_2}{\partial x_n} \\ \dots & \dots & \dots & \dots \\ \frac{\partial F_n}{\partial x_1} & \frac{\partial F_n}{\partial x_2} & \dots & \frac{\partial F_n}{\partial x_n} \end{bmatrix} = \frac{\partial(F_1, F_2, \dots, F_n)}{\partial(x_1, x_2, \dots, x_n)}$$

El jacobiano para una función vectorial es una extensión del concepto de derivadas parciales para una función de una sola componente. MATLAB habilita el comando **jacobian**, que calcula jacobianos y cuya sintaxis ya hemos visto al principio de este capítulo.

Como primer ejemplo calculamos el jacobiano de la función vectorial definida como  $(x^*y^*z, y, x + z)$  respecto del vector de variable  $(x, y, z)$ .

```
>> syms x y z
>> jacobian([x*y*z; y; x+z], [x y z])
ans =
[ y*z, x*z, x*y]
[ 0, 1, 0]
[ 1, 0, 1]
```

Como segundo ejemplo calculamos el jacobiano de la función vectorial  $f(x, y, z) = [e^x, \cos(y), \operatorname{Sen}(z)]$  en el punto  $(0, -\pi/2, 0)$ .

```
>> syms x y z
>> J=jacobian([exp(x), cos(y), sin(z)], [x, y, z])
J =
```

```
[ exp(x), 0, 0]
[ 0, -sin(y), 0]
[ 0, 0, cos(z)]

>> subs(J,{x,y,z},{0,-pi/2,0})

ans =

1      0      0
0      1      0
0      0      1
```

El jacobiano pedido resulta ser la matriz identidad.

### **Teorema de la función compuesta**

El teorema de la función compuesta o regla de la cadena permite la diferenciación de funciones vectoriales compuestas. La regla de la cadena es uno de los mitos del cálculo diferencial. Suele introducirse para funciones de una variable real, hasta ser generalizada para funciones vectoriales. Dice lo siguiente;

Sean las funciones vectoriales:

$$\bar{g}: U \subset \mathbb{R}^n \rightarrow \mathbb{R}^m \text{ y } \bar{f}: V \subset \mathbb{R}^m \rightarrow \mathbb{R}^p,$$

donde  $U$  y  $V$  son abiertos y existe la función compuesta  $(\bar{f} \circ \bar{g})$ ;  $\mathbb{R}^n \rightarrow \mathbb{R}^p$ .

Si  $g$  es diferenciable en  $\bar{x}_0$  y  $\bar{f}$ , es diferenciable en  $\bar{y}_0 = \bar{g}(\bar{x}_0)$ , entonces  $\bar{f} \circ \bar{g}$  es diferenciable en  $\bar{x}_0$  y además se cumple que: Escriba aquí la ecuación.

$$D(\bar{f} \circ \bar{g})(\bar{x}_0) = D\bar{f}(\bar{y}_0)D\bar{g}(\bar{x}_0)$$

MATLAB aplica directamente la regla de la cadena sólo con proponerle la diferenciación de la composición de funciones.

Como ejemplo sea  $f(x,y) = x^2 + y$ , y sea  $\bar{h}(u) = [Sen(3u), Cos(8u)]$ . Si  $g(x,y) = \bar{f}[\bar{h}(u)]$  calcular el jacobiano de  $g$  en  $(0,0)$ .

```
>> syms x y u
>> f=x^2+y

f =
x^2 + y
```

```

>> h=[sin(3*u), cos(8*u) ]

h =

[ sin(3*u), cos(8*u) ]

>> g=compose(h,f)
g =

[ sin(3*x^2 + 3*y), cos(8*x^2 + 8*y) ]

>> J=jacobian (g, [x, y])

J =

[ 6*x*cos(3*x^2 + 3*y), 3*cos(3*x^2 + 3*y) ]

[ -16*x*sin(8*x^2 + 8*y), -8*sin(8*x^2 + 8*y) ]

>> H=subs(J,{x,y},{0,0})

H =

```

0	3
0	0

### **Teorema de la función implícita**

Sea la función vectorial  $\bar{F} : A \subset \mathbb{R}^{n+m} \rightarrow \mathbb{R}^m$   $A=\text{Abierto de } \mathbb{R}^{n+m}$

$$(\bar{x}, \bar{y}) \rightarrow [F_1(\bar{x}, \bar{y}), \dots, F_m(\bar{x}, \bar{y})]$$

Si  $F_i$  ( $i=1,2,\dots,m$ ) son derivables con  $f$  derivada continua hasta el orden  $r$  y la matriz jacobiana  $J = \partial(F_1, \dots, F_m)/\partial(y_1, \dots, y_m)$  tiene determinante distinto de cero en un punto  $(\bar{x}_0, \bar{y}_0)$  tal que  $\bar{F}(\bar{x}_0, \bar{y}_0)=0$ , entonces existe un abierto  $U \subset \mathbb{R}^n$  que contiene a  $\bar{x}_0$  y un abierto  $V \subset \mathbb{R}^m$  que contiene a  $\bar{y}_0$  y una única función  $\bar{f} : U \rightarrow V$ , tal que  $\bar{F}[\bar{x}, f(\bar{x})] = 0 \forall x \in U$  y  $\bar{f}$  es diferenciable de orden  $r$  con derivadas continuas.

Este teorema permite asegurar la existencia de determinadas derivadas en funciones implícitas. MATLAB posibilita la diferenciación de funciones implícitas y ofrece los resultados en aquellos casos en que se cumplan las hipótesis del teorema.

Como ejemplo vamos a mostrar que cerca del punto  $(x, y, u, v) = (1, 1, 1, 1)$  se puede resolver el sistema:

$$xy + yvu^2 = 2$$

$$xu^3 + y^2v^4 = 2$$

de manera única para  $u$  y  $v$  como funciones de  $x$  e  $y$  ( $u=u(x, y)$ ,  $v=v(x, y)$ ).

En primer lugar, comprobamos si en el punto (1,1,1,1) se cumplen las hipótesis del teorema de la función implícita.

Las funciones son derivables y con derivada continua. Nos falta ver que el jacobiano correspondiente tiene determinante distinto de 0 en el punto (1,1,1,1).

```
>> clear all
>> syms x y u v
>> f=x*y +y*v*u^2-2
f =
v*y*u^2 + x*y - 2

>> g=x*u^3+y^2*v^4-2
g =
x*u^3 + v^4*y^2 - 2

>> J=simplify(jacobian([f,g],[u,v]))
J =
[ 2*u*v*y,           u^2*y]
[ 3*u^2*x,           4*v^3*y^2]

>> D=det(subs(J,{x,y,u,v},{1,1,1,1}))
D =
5
```

### ***Teorema de la función inversa***

Sea la función vectorial  $\bar{f}: U \subset \mathbb{R}^n \rightarrow \mathbb{R}^n$   $U=\text{Abierto de } \mathbb{R}^n$

$$(x_1, x_2, \dots, x_n) \rightarrow [f_1(x_1, x_2, \dots, x_n), \dots, f_n(x_1, x_2, \dots, x_n)]$$

de modo que  $f$  es derivable con derivada continua.

Si existe  $\bar{x}_0$  tal que  $|J| = |\partial(f_1, \dots, f_n) / \partial(x_1, \dots, x_n)| \neq 0$  en  $x_0$ , entonces existen un abierto  $A$  que contiene a  $\bar{x}_0$  y un abierto  $B$  que contiene a  $\bar{f}(\bar{x}_0)$ , tales que  $\bar{f}(A) = B$  y  $\bar{f}$  tiene una función inversa  $\bar{f}^{-1}: B \rightarrow A$  que es derivable con derivada continua. Además se cumple que:

$$D\bar{f}^{-1}(y) = [D\bar{f}(x)]^{-1} \text{ y si } J = \partial(f_1, \dots, f_n) / \partial(x_1, \dots, x_n) \text{ entonces } |J^{-1}| = 1/|J|.$$

MATLAB realiza automáticamente los cálculos correspondientes al teorema de la función inversa, siempre que se cumplan las hipótesis.

Como ejemplo, dada la función vectorial  $(u(x,y), v(x,y))$ , donde:

$$u(x, y) = \frac{x^4 + y^4}{x}, v(x, y) = \operatorname{Sen}(x) + \operatorname{Cos}(y)$$

Hallaremos las condiciones para que exista la función vectorial inversa  $(x(u,v), y(u,v))$  con  $x=x(u,v)$  e  $y=y(u,v)$  y hallar la derivada y el jacobiano de la transformación inversa. Hallaremos también su valor en el punto  $(\pi/4, -\pi/4)$ .

Las condiciones que tiene que cumplir serán las hipótesis del teorema de la función inversa. Las funciones son derivables con función derivada continua, salvo acaso en  $x=0$ . Ahora vamos a plantear el jacobiano de la transformación directa  $\partial(u(x,y), v(x,y))/\partial(x, y)$ :

```
>> syms x y
>> J=simple((jacobian([(x-4+y-4)/x,sin (x)+cos (y)], [x,y])))
J =
[ 3*x-2-1/x-2*y^4,          4*y-3/x]
[           cos(x),          -sin(y)]
>> pretty(det(J))
      4           4           3
      3 sin(y) x - sin(y) y + 4 y cos(x) x
----- x

```

Por lo tanto, en los puntos donde esta expresión no se anule, se puede resolver para  $x$  e  $y$  en términos de  $u$  y  $v$ . Además, también ha de cumplirse que  $x \neq 0$ .

Calculamos la derivada de la función inversa. Su valor será la matriz inversa de la matriz jacobiana inicial y el determinante de su jacobiano será el recíproco del determinante del jacobiano inicial:

```
>> I=simple(inv(J));
>> pretty(simple(det(I)))
      2
----- x
      4           4           3
      3 sin(y) x - sin(y) y + 4 y cos(x) x
```

Se observa que el determinante del jacobiano de la función vectorial inversa es el recíproco del determinante del jacobiano de la función directa.

Vamos a hallar los valores para el punto  $(\pi/4, -\pi/4)$ :

```
>> numeric(subs(subs(determ(I),pi/4,'x'),-pi/4,'y'))  
ans =  
0.38210611216717  
  
>> numeric(subs(subs(symdiv(1,determ(J)),pi/4,'x'),-pi/4,'y'))  
ans =  
0.38210611216717
```

Estos resultados corroboran que el determinante del jacobiano de la función inversa es el recíproco del determinante del jacobiano de la función.

### ***Teorema del cambio de variable***

El teorema del cambio de variable es otra de las herramientas fundamentales en el análisis diferencial de variable múltiple. Sus aplicaciones se extienden a cualquier problema en el que sea necesario transformar variables.

Supongamos que conocemos una función  $f(x,y)$  que depende de las variables originales  $x$  e  $y$ , y que cumple todas las condiciones de derivabilidad y continuidad necesarias. Introducimos las nuevas variables  $u$  y  $v$ , que se relacionan con las anteriores mediante las funciones  $u=u(x,y)$  y  $v=v(x,y)$ , de modo que las funciones  $u$  y  $v$  también cumplen las condiciones de derivabilidad y continuidad necesarias (teorema de la función inversa) para poder despejar  $x$  e  $y$  en función de  $u$  y  $v$ :  $x=x(u,v)$  e  $y=y(u,v)$ .

Bajo las condiciones anteriores, es posible expresar la función inicial  $f$  en función de las nuevas variables  $u$  y  $v$  mediante la expresión;

$$f(u,v) = f(x(u,v), y(u,v)) |J| \text{ siendo } J \text{ el jacobiano } \partial(x(u,v), y(u,v))/\partial(u,v)$$

El teorema es generalizable a funciones vectoriales de  $n$  componentes.

Como ejemplo consideraremos la función  $f(x,y) = f(x,y) = e^{-(x+y)}$  y la transformación  $u=u(x,y)=x+y$ ,  $v=v(x,y)=x$  para finalmente hallar  $f(u,v)$ .

Calculamos la transformación inversa y su jacobiano para aplicar el teorema del cambio de variable:

```

>> syms x y u v
>> [x,y]=solve ('u=x+y, v=x', 'x', 'y')
x =
v
y =
u-v

>> jacobian([v,u-v], [u,v])
ans =
[ 0,  1]
[ 1, -1]

>> f=exp(x-y);
>> pretty(simple(subs(f, {x,y}, {v,u-v})* abs(det(jacobian(
[v,u-v], [u,v])))))
exp (2 v - u)

```

La función pedida es  $f(u,v) = e^{2v-u}$ .

## DESARROLLOS EN SERIE PARA VARIAS VARIABLES

El concepto de desarrollo en serie de una función como una serie de potencias, visto ya para funciones de una variable, puede generalizarse a varias variables. El *teorema de Taylor para varias variables* dice lo siguiente;

Sea  $f: \mathbb{R}^n \rightarrow \mathbb{R}$  derivable  $k$  veces y con derivadas parciales continuas;

$$(x_1, \dots, x_n) \rightarrow f(x_1, \dots, x_n)$$

El desarrollo de Taylor de la función  $f(x)$  en el punto  $\bar{a} = (a_1, \dots, a_n)$  de orden  $k$  es el siguiente:

$$\begin{aligned}
f(\bar{x}) &= f(\bar{a}) + \sum_{i=1}^n \frac{\partial y}{\partial x_i}(\bar{a}) t_i + \frac{1}{2!} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 f}{\partial x_i \partial x_j}(\bar{a}) t_i t_j \\
&+ \frac{1}{3!} \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \frac{\partial^3 f}{\partial x_i \partial x_j \partial x_k}(\bar{a}) t_i t_j t_k + R(k+1)
\end{aligned}$$

$$\bar{x} = (x_1, x_2, \dots, x_n) \bar{a} = (a_1, a_2, \dots, a_n) t_i = x_i - a_i (i=1, 2, \dots, n)$$

R=resto

El teorema de Taylor permite desarrollar en forma polinómica una función cualquiera que cumpla sus hipótesis. Normalmente, los desarrollos se efectúan hasta el orden 2.

Como ejemplo obtendremos el desarrollo de Taylor hasta el orden 2 y en el punto (1,0) de la función:

$$f(x, y) = e^{(x-1)^2} \cos(y)$$

```
>> pretty(simplify(subs(f,{x,y},{1,0})+subs(diff(f,x),{x,y},{1,0})*(x-1)
+subs(diff(f,y),{x,y},{1,0})*(y)+1/2*(subs(diff(f,x,2),{x,y},{1,0})*
(x-1)^2+subs(diff(f,x,y),{x,y},{1,0})*(x-1)*(y) +
subs(diff(f,y,2),{x,y},{1,0})*(y)^2)) )
```

$$(x - 1)^2 - \frac{y^2}{2} + 1$$

## ROTACIONAL, DIVERGENCIA Y LAPLACIANO

Los conceptos típicos para el trabajo en campos vectoriales son tratables de modo directo por MATLAB. A continuación se presentan las definiciones de los conceptos más básicos en campos vectoriales.

**Gradiente:** Si  $h=f(x,y,z)$ , entonces el gradiente de  $f$ , que se denota mediante  $\nabla f(x,y,z)$ , es el vector:

$$\nabla f = \nabla f(x,y) = \frac{\partial f(x,y)}{\partial x} i + \frac{\partial f(x,y)}{\partial y} j$$

**Potencial escalar de un campo vectorial:** Un campo vectorial  $\bar{F}$  se llama conservativo si existe una función diferenciable  $f$  tal que  $\bar{F} = \nabla f$ . La función  $f$  se conoce como función potencial escalar para  $\bar{F}$ .

**Rotacional de un campo vectorial:** El rotacional del campo vectorial  $F(x,y,z)=M_i+N_j+P_k$  tiene la siguiente expresión:

$$\text{rot } F(x,y,z) = \Delta x F(x,y,z) = \left( \frac{\partial P}{\partial y} - \frac{\partial N}{\partial z} \right) i - \left( \frac{\partial P}{\partial x} - \frac{\partial M}{\partial z} \right) j + \left( \frac{\partial N}{\partial x} - \frac{\partial M}{\partial y} \right) k$$

**Potencial vector de un campo vectorial:** Un campo vectorial  $F$  tiene 1 como potencial vector a otro campo vectorial  $G$  si  $F = \text{rotacional}(G)$ .

**Divergencia de un campo vectorial:** La divergencia del campo vectorial  $F(x,y,z) = M\hat{i} + N\hat{j} + P\hat{k}$  tiene la siguiente expresión:

$$\text{dive } F(x, y) = \Delta \cdot F(x, y) = \left( \frac{\partial M}{\partial x} + \frac{\partial N}{\partial y} \right)$$

**Laplaciano:** El laplaciano es el operador diferencial definido por:

$$\text{laplacian} = \Delta^2 = \Delta \cdot \Delta = \left( \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} \right)$$

Como primer ejemplo calculamos gradiente y laplaciano de la función:

$$w = \frac{1}{\sqrt{1-x^2-y^2-z^2}}$$

```
>> gradiente=simplify([diff(f,x), diff(f,y), diff(f,z)])  
  
gradiente =  
[ x/(- x^2 - y^2 - z^2 + 1)^(3/2), y/(- x^2 - y^2 - z^2 + 1)^(3/2),  
z/(- x^2 - y^2 - z^2 + 1)^(3/2)]  
  
>> pretty(gradiente)  
+-  
| x  
| -----, -----, -----  
| | 3 | 3 | 3  
| |-- |-- |--  
| | 2 | 2 | 2  
| | (- x - y - z + 1 ) | { - x - y - z + 1 } | ( - x - y - z + 1 )  
+-  
  
>> laplaciano=simplify([diff(f,x,2)+diff(f,y,2)+diff(f,z,2)])  
  
laplaciano =  
3/(- x^2 - y^2 - z^2 + 1)^(5/2)  
  
>> pretty(laplaciano)  
3  
-----  
5  
-  
2  
2 2 2  
(- x - y - z + 1)
```

Como segundo ejemplo calculamos el rotacional y la divergencia del campo vectorial:

$$\bar{F}(x,y,z) = \arctan \frac{x}{y} \bar{i} + \ln \sqrt{x^2 + y^2} \bar{j} + \bar{k}$$

```
>> M=atan(x/y)
M =
atan(x/y)

>> N=log(sqrt(x^2+y^2))
N =
log((x-2 + y-2)^(1/2))

>> P=1
P =
1
>> Rotacional=simplify([diff(P,y)-diff(N,z),diff(P,x)-
diff(M,z),diff(N,x)-diff(M,y)])
Rotacional =
[ 0, 0, (2*x)/(x^2 + y^2)]

>> pretty(Rotacional)
+-          +-          -+
|           2x          |
| 0, 0,  -----  |
|           2   2      |
|           x + y      |
+-                      -+
+-          +-          -+
|           2x          |
|           2            |
| (2*y)/(x^2 + y^2)  |
+-          +-          -+
>> Divergencia=simplify(diff(M,x)+diff(N,y)+diff(P,z))
Divergencia =
(2*y)/(x^2 + y^2)

>> pretty(Divergencia)
2y
-----
2   2
x  + y
```

## COORDENADAS CILÍNDRICAS, ESFÉRICAS Y RECTANGULARES

MATLAB permite convertir fácilmente coordenadas cilíndricas y esféricas a rectangulares, cilíndricas a esféricas y las transformaciones inversas. En cuanto a las coordenadas cilíndricas y esféricas tenemos lo siguiente:

En un sistema de *coordenadas cilíndricas*, un punto  $P$  del espacio se representa por una tripleta ordenada  $(r, \theta, z)$ , donde:

$r$  es la distancia del origen ( $O$ ) a la proyección de  $P$  ( $P'$ ) en el plano  $XY$

$\theta$  es el ángulo entre el eje  $X$  y el segmento  $OP'$

$z$  es la distancia  $PP'$

En un sistema de *coordenadas esféricas*, un punto  $P$  del espacio se representa por una tripleta ordenada  $(\rho, \theta, (\phi))$ , donde:

$\rho$  es la distancia de  $P$  al origen

$\theta$  es el mismo ángulo que el usado en coordenadas cilíndricas

$\phi$  es el ángulo entre el eje  $Z$  positivo y el segmento  $OP$

Fácilmente se llega a las ecuaciones de conversión siguientes:

Cilíndricas a rectangulares:

$$x = r \cos \theta$$

$$y = r \sin \theta$$

$$z = z$$

Rectangulares a cilíndricas:

$$r = \sqrt{x^2 + y^2}$$

$$\theta = \arctg(y/x)$$

$$z = z$$

Esféricas a rectangulares:

$$x = \rho \sin \phi \cos \theta$$

$$y = \rho \sin \phi \sin \theta$$

$$z = \rho \cos \phi$$

Rectangulares a esféricas:

$$\rho = \sqrt{x^2 + y^2 + z^2}$$

$$\theta = \arctg(y/x)$$

$$\phi = \arccos \frac{z}{\sqrt{x^2 + y^2 + z^2}}$$

Como ejemplo expresamos las superficies de ecuaciones dadas por  $xz = 1$  y  $x^2 + y^2 + z^2 = 1$  en coordenadas esféricas.

```
>> clear all
>> syms x y z r t a
>> f=x*z-1

f=
x*z - 1
>>
ecuacion=simplify(subs(f,{x,y,z},{r*sin(a)*cos(t),r*sin(a)*sin(t),r*cos(a)}))

ecuación =
r^2*cos(a)*sin(a)*cos(t) - 1

>> pretty(ecuación)

 2
r cos(a) sin(a) cos(t) ~ 1

g =
x^2 + y^2 + z^2 - 1
>>
ecuacion1=simplify(subs(g,{x,y,z},{r*sin(a)*cos(t),r*sin(a)*sin(t),r*cos(a)}))

ecuacion1 =
r^2 - 1

>> pretty(ecuacion1)

 2
r - 1
```

No obstante, MATLAB ofrece comandos que permiten la transformación entre los diferentes sistemas de coordenadas.

A continuación se presenta la sintaxis de los comandos que ofrece MATLAB en su módulo básico para transformación de coordenadas.

<b>[THETA,RHO,Z] = cart2pol(X,Y,Z)</b>	Transforma cartesianas a cilíndricas
<b>[THETA,RHO] = cart2pol(X,Y)</b>	Transforma cartesianas a polares
<b>[THETA,PHI,R] = cart2sph(X,Y,Z)</b>	Transforma cartesianas a esféricas
<b>[X,Y,Z] = pol2cart(THETA,RHO,Z)</b>	Transforma cilíndricas a cartesianas
<b>[X,Y] = pol2cart(THETA,RHO)</b>	Transforma polares a cartesianas
<b>[x,y,z] = sph2cart(THETA,PHI,R)</b>	Transforma esféricas a cartesianas

En el ejemplo siguiente se transforma el punto  $(\pi, 1, 2)$  de coordenadas cilíndricas a cartesianas.

```
>> [X,Y,Z]=pol2cart(pi,1,2)
```

X =

-1

Y =

1.2246e-016

Z =

2

A continuación se transforma el punto  $(1,1,1)$  de coordenadas cartesianas a esféricas y cilíndricas.

```
>> [X,Y,Z]=cart2sph(1,1,1)
```

X =

0.7854

Y =

0.6155

Z =

1.7321

```
>> [X,Y,Z]=cart2pol(1,1,1)
```

X =

0.7854

Y =

1.4142

```
z =
1
En el ejemplo siguiente se transforma a cartesianas el punto (2,π/4) en polares.
>> [X,Y] = pol2cart(2,pi/4)
x =
-0.3268
y =
0.7142
```

*Ejercicio 1. Estudiar la derivabilidad de la función:*

$$f(x)=x^2 \operatorname{sen}(1/x) \text{ si } x \neq 0 \text{ y } f(x)=0 \text{ si } x=0$$

Podemos comenzar estudiando la continuidad de la función en el punto  $x=0$ .

```
>> syms x
>> f=x^2*sin(1/x)
f =
x^2*sin (1/x)
>> limit(f,x,0, 'right')
ans =
0
>> limit(f,x,0, 'left')
ans =
0
>>
>> limit(f,x,0)
ans =
0
```

Se observa que la función es continua en  $x=0$  porque el límite de la función cuando  $x$  tiende a cero coincide con el valor de la función en el punto cero. Por lo tanto, puede ser derivable en cero.

Estudiamos ahora la derivabilidad en el punto  $x=0$ :

```
>> syms h, limit((h^2*sin(1/h) - 0)/h,h,0)
ans =
0
```

Luego, se concluye que:

$$\lim_{h \rightarrow 0} \frac{f(0+h)-f(0)}{h} = f'(0) = 0$$

lo que indica la derivabilidad de la función  $f$  en el punto  $x=0$ .

Veamos lo que ocurre para un punto  $x=a$  distinto de cero:

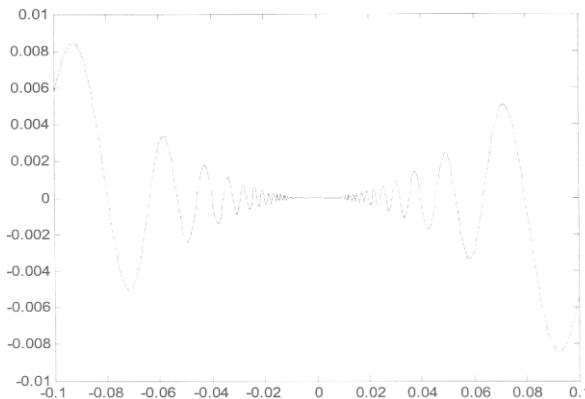
```
>> pretty(simple(limit((subs(f,[x],{a+h})-subs(f,[x],{a}))/h,h,a)))
/ 1 \ / 1 \
4 a sin| --- | - a sin| - |
\ 2a / \ a /
```

Luego, se concluye que:

$$\lim_{h \rightarrow 0} \frac{f(a+h)-f(a)}{h} = f'(a) = 4a \operatorname{sen}\left(\frac{1}{2a}\right) - \operatorname{sen}\left(\frac{1}{a}\right)$$

Luego, ya hemos hallado el valor de la derivada para cualquier punto  $x=a$  distinto de cero. En la figura siguiente representamos la función.

```
>> fplot('x^2*sin(1/x) [-1/10,1/10])
```



*Ejercicio 2. Calcular la derivada respecto de x de las siguientes funciones:*

$$\text{Log}[\text{Sen}(2x)], x^{\tan(x)}, \frac{4}{3} \sqrt{\frac{x^2+1}{x^2+2}}, \text{Log}[x + \sqrt{(x^2+1)}]$$

```
>> pretty(simple(diff('log(sin(2*x))','x')))

0 cot(2 x)

>> pretty(simple(diff('x^'tanx','x')))

tanx
x      tanx
-----
x

>> pretty(simple(diff('(4/3)*sqrt((x^2-1)/(x^2+2))','x')))

x
4-----
0      1/2      2      3/2
(x - 1)      (x + 2)

>> pretty(simple(diff('log(x+(x^2+1)^(1/2))','x')))

1
-----
2      1/2
(x + 1)
```

*Ejercicio 3. Calcular la derivada enésima de las siguientes funciones:*

$$1/x, e^{x/2}, (1+x)/(1-x)$$

```
>> f='1/x';

>> [diff(f),diff(f,2),diff(f,3),diff(f,4),diff(f,5)]

ans =
-1/x^2      2/x^3      -6/x^4      24/x^5      -120/x^6
```

Ya tenemos clara la formación de la derivada enésima, que será  $[-1]^n n! / x^{n+1}$

```
>> f='exp(x/2)';

>> [diff(f),diff(f,2),diff(f,3),diff(f,4),diff(f,5)]

ans =
1/2*exp(1/2*x) 1/4*exp(1/2*x) 1/8*exp(1/2*x) 1/16*exp(1/2*x) 1/32*exp(1/2*x)
```

Luego, la derivada enésima será  $(e^{x/2})/2^n$ .

```
>> f='(1+x)/(1-x)';
>> [simple(diff(f)),simple(diff(f,2)),simple(diff(f,3)),simple(diff(f,4))]

ans =
2/ (-1+X)^2          -4/ (-1+X)^3          12/ (-1+X)^4          -48/ (-1+X)^5
```

Luego, la expresión de la derivada enésima será  $2(n!)/(1-x)^{n+1}$ .

*Ejercicio 4. Encontrar la ecuación de la tangente a la curva:*

$$f(x) = 2x^3 + 12x^2 - 12x + 1 \text{ cuando } x = -1.$$

*Hallar también los x para los que las tangentes a la curva  $g(x) = (x^2 - x - 4)/(x - 1)$  sean horizontales y verticales. Hallar sus asíntotas.*

```
>> f='2*x^3+3*x^2-12*x+7';
>> g=diff(f)
```

```
g =
6*x^2+6*x-12
```

```
>> subs(g,-1)
```

```
ans =
```

```
-12
```

```
>> subs(f,-1)
```

```
ans =
```

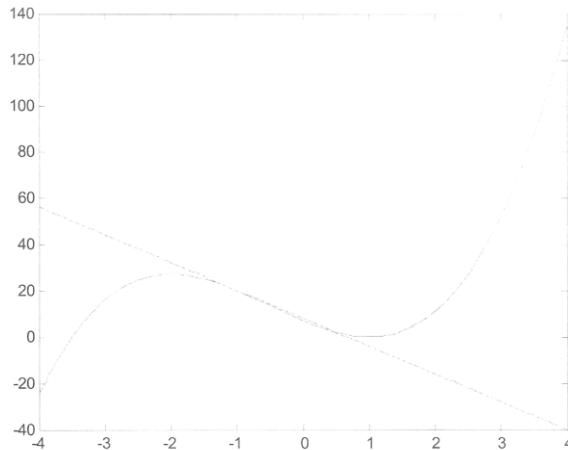
```
20
```

Hemos visto que la pendiente de la recta tangente en el punto  $x = -1$  es -12, y que la función en  $x = -1$  vale 20, luego, la ecuación de la tangente a la curva en el punto (-1,20) será:

$$y - 20 = -12(x - (-1))$$

Realizamos la representación gráfica de la curva y su tangente sobre los mismos ejes.

```
>> fplot('2*x^3+3*x^2-12*x+7,20-12*(x - (-1))', [-4, 4])
```



Para calcular las tangentes horizontales a la curva  $y=f(x)$  en  $x=x_0$ , hallamos los valores  $x_0$  que anulen la pendiente de la tangente ( $f'(x_0)=0$ ). Su ecuación será, por tanto,  $y=f(x_0)$ .

Para calcular las tangentes verticales a la curva  $y=f(x)$  en  $x=x_0$ , hallamos los valores  $x_0$  que hagan infinita la pendiente de la tangente ( $f'(x_0)=\infty$ ). Su ecuación será, por tanto,  $x=x_0$ :

```
>> g='(x^2-x+4)/(x-1)'
```

```
>> solve(diff(g))
```

```
ans =
```

```
[ 3]
```

```
[-1]
```

```
>> subs(g, 3)
```

```
ans =
```

```
5
```

```
>> subs(g, -1)
```

```
ans =
```

```
-3
```

Las dos tangentes horizontales tendrán de ecuaciones:

$$y = g'[-1](x+1) - 3, \text{ o sea, } y = -3$$

$$y = g'[3](x - 3) + 5, \text{ o sea, } y = 5$$

Las tangentes horizontales no son asíntotas porque los valores correspondientes de  $x_0$  son finitos (son -1 y 3).

Analizemos ahora las tangentes verticales. Para ello, calculamos los valores de  $x$  que hacen  $g'(x)$  infinito (valores que anulan el denominador de  $g'$  sin anular a la vez el numerador);

```
>> solve('x-1')
```

```
ans =
```

```
1
```

Por lo tanto, la tangente vertical tendrá de ecuación  $x=1$ .

Si para  $x=1$  el valor de  $g(x)$  es infinito, la tangente vertical será una asíntota vertical.

```
subs(g,1)
```

```
Error, división by zero
```

Efectivamente, la recta  $x=1$  es una asíntota vertical.

Como  $\lim_{x \rightarrow \infty} g(x) = \infty$ , no existen asíntotas horizontales.

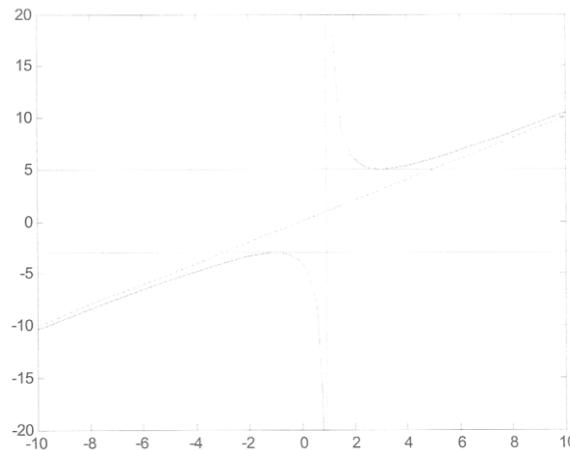
```
>> syms x, limit(((x^2-x+4)/(x-1))/x,x,inf)
ans =
1
>> syms x, limit(((x^2-x+4)/(x-1) - x)/x,x,inf)
ans =
0
```

Luego, existe la asíntota oblicua  $y = x$ .

A continuación, representamos la curva con sus asíntotas y tangentes;

Para ello, representamos en el mismo gráfico (ver la figura 16-4) la curva cuya ecuación es  $g(x) = (x^2-x+4)/(x-1)$ , las tangentes horizontales de ecuaciones  $a(x) = -3$  y  $b(x) = 5$ , la asíntota oblicua de ecuación  $c(x) = x$  y las asíntotas horizontales y verticales (que las representa por defecto el comando *plot*):

```
>> fplot ('[(x^2-x+4) / (x-1), -3, 5, x]', [-10,10,-20,20])
```



*Ejercicio 5. Descomponer el número positivo  $a$  en dos sumandos, tales que la suma de sus cubos sea mínima.*

Sea  $x$  uno de los sumandos. El otro será  $a-x$ . Se pide que la suma  $x^3+(a-x)^3$  sea mínima.

```
>> syms x a;
>> f='x^3+(a-x)^3'
f =
x^3+(a-x)^3
>> solve(diff(f,'x'))
ans =
1/2*a
```

El posible máximo o mínimo se encuentra en  $x=a/2$ . Utilizamos la segunda derivada para ver que efectivamente se trata de un mínimo:

```
>> subs(diff(f,'x',2), 'a/2')
ans =
3*a
```

Como  $a > 0$  (por hipótesis),  $4a > 0$ , lo que asegura la existencia de mínimo para  $x=a/2$ .

Los números pedidos serán  $x=a/2$  y  $a-x=a-a/2=a/2$ . O sea, que cuando los dos sumandos son iguales se obtiene el mínimo pedido.

*Ejercicio 6. Se quiere comprar un terreno rectangular de 1.600 metros cuadrados de superficie y posteriormente vallarlo. Sabiendo que la valla cuesta 200 pesetas por metro, ¿qué dimensiones deberá tener el terreno a fin de que su cercado sea lo más económico posible?*

Si la superficie es de 1.600 metros cuadrados y una de sus dimensiones, desconocida, es  $x$ , la otra será  $1.600/x$ .

El perímetro del rectángulo será  $p(x)=2x+2(1600/x)$ , y el coste estará dado por  $f(x)=200p(x)$ :

```
>>
f='200*(2*x+2*(1600/x))'
f =
200*(2*x+2*(1600/x))
```

Esta será la función a minimizar:

```
>> solve(diff(f))
ans =
[ 40]
[-40]
```

Los posibles máximos y mínimos se presentan para  $x=-40$  y para  $x=40$ . Usamos la segunda derivada para diferenciarlos:

```
>> [subs(diff(f,2),40), subs(diff(f,2),-40)]
ans =
20    -20
```

$x=40$  es un mínimo y  $x=-40$  es un máximo. Luego, uno de los lados del terreno rectangular pedido mide 40 metros, y el otro medirá  $1.600/40 =40$  metros. Luego, el rectángulo problema resulta ser un cuadrado de 40 metros de lado.

*Ejercicio 7. Dada la función real de variable real, definida por:*

$$f(x,y)=xy/(x^2+y^2) \quad \text{si } x^2+y^2 \neq 0 \quad \text{y} \quad f(x,y)=0 \quad \text{si } x^2+y^2=0$$

*Calcular las derivadas parciales de  $f$  en el origen. Estudiar la diferenciabilidad.*

Para hallar  $\partial f / \partial x$  y  $\partial f / \partial y$  en el punto (0,0), aplicamos directamente la definición de derivada parcial en un punto:

```
>> syms x y h k
>> limit((subs(f,{x,y},{h,0})-0)/h,h,0)
ans =
0
>> limit((subs(f,{x,y},{0,k})-0)/k,k,0)
ans =
0
```

Hemos visto que los límites de las dos expresiones anteriores cuando  $h \rightarrow 0$  y  $k \rightarrow 0$ , respectivamente, son ambos cero. Es decir:

$$\lim_{h \rightarrow 0} \frac{f(h, 0) - f(0, 0)}{h} = \frac{\partial f}{\partial x}(0, 0) = 0$$

$$\lim_{k \rightarrow 0} \frac{f(0, k) - f(0, 0)}{k} = \frac{\partial f}{\partial y}(0, 0) = 0$$

Vemos que las dos derivadas parciales en el origen son iguales y valen cero.

Pero la función no es diferenciable en el origen, porque no es continua en (0,0), ya que no existe su límite cuando  $(x, y) \rightarrow (0, 0)$ . Veamos:

```
>> syms m
>> limit((m*x)^2 / (x^2 + (m*x)^2), x, 0)
ans =
m^2 / (m^2 + 1)
```

El límite no existe en (0,0), porque al considerar los límites direccionales respecto de la familia de rectas  $y=mx$ , el resultado depende del parámetro  $m$ .

*Ejercicio 8. Dada la función:*

$$f(x, y) = e^{\frac{x^2+y^2}{8}} [\cos^2(x) + \sin^2(x)]$$

*calcular:*

$$\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial^2 f}{\partial x \partial y}, \frac{\partial^2 f}{\partial x^2}, \frac{\partial^2 f}{\partial y^2}, \frac{\partial^3 f}{\partial x \partial y^2}, \frac{\partial^4 f}{\partial x^2 \partial y^2} \text{ y } \frac{\partial^5 f}{\partial x^3 \partial y^2}$$

*Y hallar su valor en el punto  $(\pi/3, \pi/6)$ .*

```

>> f=exp(-(x-2+y^2)/8)*(cos(x)^2+sin(y)^2)

f =

(cos(x)^2 + sin(y)^2)/exp(x-2/8 + y-2/8)>>

>> pretty(simple(diff(f,x)))
      2          2
x cos (x) + x sin (y) + 4 sin (2 x)
-----
      / 2          2 \
      | x      y   |
4 exp | --- + --- |
      \ 8      8   /
>> pretty(simple(diff(f,y)))
      2          2
y eos (x) + y sin (y) - 4 sin (2 y)
      / 2          2 \
      | x      y   |
4 exp | - + - |
      \ 8      8   /
>> pretty(simple(diff(diff(f,x),y)))
      3          2
x y cos (x) + x y sin (y) - 4 x sin (2 y) + 4 y sin (2 x)
-----
      / 2          2 \
      | x      y   |
16 exp | - + - |
      \ 8      8   /
>> pretty(simple(diff(diff(f,x),x)))
      2          2
x cos (2 x) x cos (2 y)      2
2 cos (2 y) - 34 cos (2 x) + 8 x sin (2 x) + ----- -----
      2          2
      / 2          2 \
      | x      y   |
16 exp | - + - |
      \ 8      8   /
>> pretty(simple(diff(diff(f,y),y)))
      2          2
y cos(2x)      2
2 cos (2 x) - 34 cos (2 y) + 8 y sin (2 y) - ----- + ----- y + 4
      2          2
      / 2          2 \
      | x      y   |
16 exp | - + - |
      \ 8      8   /

```

```

>> pretty(simple(diff(diff(diff(f,x),y),y)))

$$(8x + 32 \sin(2x) + 4x \cos(2x) - 68 x \cos(2y) - 8 y \sin(2x) - 2 x^2 y +$$


$$16 x^2 y \sin(2y) - x y \cos(2x) + x y \cos(2y)) / 128 \exp(-\frac{x}{8})^2 //$$


>> pretty(simple(diff(diff(diff(f,x),x),y),y))

$$(272 \cos(2x) - 272 \cos(2y) + 2 x^2 y - 64 x \sin(2x) + 64 y \sin(2y) - 4 x \cos(2x) +$$


$$68 x^2 \cos(2y) - 68 y \cos(2x) + 4 y^2 \cos(2y) - 8 x^2 - 8 y^2 + x^2 y \cos(2x) -$$


$$x^2 y^2 \cos(2y) + 16 x^2 y \sin(2x) - 16 x^2 y \sin(2y) + 32) / 512 \exp(-\frac{x}{8})^2 //$$


>> pretty(simple(diff(diff(diff(f,x),x),x),y)))

$$-(96 x^3 + 2432 \sin(2x) + 2 x^2 y + 816 x \cos(2x) - 816 x \cos(2y) - 4 x \cos(2x) +$$


$$68 x^3 \cos(2y) - 96 x^2 \sin(2x) - 608 y \sin(2x) - 24 x^2 y - 8 x^3 + x^2 y \cos(2x) -$$


$$x^3 y \cos(2y) + 24 x^2 y \sin(2x) + 192 x^2 y \sin(2y) - 204 x^2 y \cos(2x) +$$


$$12 x^2 y \cos(2y) - 16 x^3 y \sin(2y)) / 2048 \exp(-\frac{x}{8})^3 //$$


```

Los valores de las derivadas parciales anteriores en el punto  $(n/3, n/6)$  se calcula como sigue (de última a primera):

```

>> subs(diff(diff(diff(f,x),x),y),{x,y},{pi/3,pi/6})
ans =
-0.5193

>> subs(diff(diff(diff(f,x),x),y),{x,y},{pi/3,pi/6})
ans =
-0.3856

>> subs(diff(diff(f,x),y),{x,y},{pi/3,pi/6})
ans =
0.0250

>> subs(diff(f,y),y),{x,y},{pi/3,pi/6})
ans =

```

```
0.5534
```

```
>> subs(diff(diff(f,x),x),{x,y},{pi/3,pi/6})
```

```
ans =
```

```
1.1481
```

```
>> subs(diff(diff(f,x),y),{x,y},{pi/3,pi/6})
```

```
ans =
```

```
-0.0811
```

```
>> subs(diff(f,y),{x,y},{pi/3,pi/6})
```

```
ans =
```

```
0.6745
```

```
>> subs(diff(f,x),{x,y},{pi/3,pi/6})
```

```
ans =
```

```
-0.8399
```

*Ejercicio 9. Hallar y clasificar los puntos extremos de la función:*

$$-120x^3 - 30x^4 + 18x^5 + 5x^6 + 30xy^2$$

Comenzamos hallando los posibles puntos extremos. Para ello, igualamos a cero las derivadas parciales respecto de todas las variables (componentes del vector gradiente de  $f$ ) y resolvemos el sistema resultante en las tres variables:

```
>> syms x y
>> f=-120*x^3-30*x^4+18*x^5+5*x^6+30*x*y^2
f =
5*x^6 + 18*x^5 - 30*x^4 - 120*x^3 + 30*x*y^2
>> [x y]=solve(diff(f,x),diff(f,y),x,y)
x =
0
2
-2
-3
y =
0
0
0
0
```

Luego, los posibles puntos extremos son: (0,0), (-2,0), (2,0) y (-3,0).

Vamos a analizar de qué clase de extremos se trata. Para ello, calculamos la matriz hessiana y la expresamos como una función de x, y.

```
>> clear all
>> syms x y

>> f=-120*x^3-30*x^4+18*x^5+5*x^6+30*x*y^2

f =
5*x^6+18*x^5-30*x^4-120*x^3+30*x*y^2
>>
H=simplify([diff(f,x,2),diff(diff(f,x),y);diff(diff(f,y),x),diff(f,y,
2) ] )

H =
[ -30*x* (-5*x^3 -12*x^2 + 12*x + 24), 60*y]
[ 60*y, 60*x]
```

Ahora calculamos el valor del determinante de la matriz hessiana en los posibles puntos extremos.

```
>> det(subs(H,{x,y},{0,0}))
```

```
ans =
```

```
0
```

El origen resulta ser un punto degenerado, pues el determinante de la matriz hessiana se anula en (0,0).

Analizaremos ahora el punto (-2,0).

```
>> det(subs(H,{x,y}, {-2,0}))
```

```
ans =
```

```
57600
```

```
>> eig(subs(H,{x,y}, {-2,0 }))
```

```
ans =
```

```
-480
-120
```

La matriz hessiana en el punto (-2,0) tiene determinante no nulo, y, además, es definida negativa, porque todos sus autovalores son negativos. Por lo tanto, el punto (-2,0) es un máximo de la función.

Analizaremos ahora el punto (2,0).

```
>> det(subs(H, {x,y}, {2,0}))
```

```
ans =
```

```
288000
```

```
>> eig(subs(H, {x,y}, {2,0}))
```

```
ans =
```

```
120  
2400
```

La matriz hessiana en el punto (2,0) tiene determinante no nulo, y, además, es definida positiva, porque todos sus autovalores son positivos. Por lo tanto, el punto (2,0) es un mínimo de la función.

Analizaremos ahora el punto (-3,0).

```
>> det(subs(H, {x,y}, {-3,0}))
```

```
ans =
```

```
-243000
```

```
>> eig(subs(H, {x,y}, {-3,0}))
```

```
ans =
```

```
-180  
1350
```

La matriz hessiana en el punto (-3,0) tiene determinante no nulo, y, además, ni es definida positiva ni definida negativa, porque sus autovalores ni son todos positivos ni son todos negativos. Por lo tanto, el punto (-3,0) es un punto silla de la función.

*Ejercicio 10. Hallar y clasificar los puntos extremos de la función:*

$$f(x,y,z) = \sqrt{[(x^2 + y^2) - z]}$$

*sujeta a las restricciones:  $x^2 + y^2 = 16$  y  $x+y+z=10$ .*

En primer lugar planteamos la función de Lagrange L, que es una combinación lineal de la función objetivo y las restricciones:

```
>> clear all  
>> syms x y z L p q
```

```
>> f=(x^2+y^2)^(1/2)-z
f =
(x^2+y^2)^(1/2) - z
>> g1=x^2+y^2-16, g2=x+y+z-10
g1 =
x^2 + y^2 - 16
g2 =
x + y + z - 10
>> L=f+p*g1+q*g2
L =
(x^2 + y^2)^(1/2) - z + q*(x + y + z - 10) + p*(x^2 + y^2 - 16)
```

A continuación, los posibles puntos extremos se obtienen de la resolución del sistema resultante de igualar a cero las componentes del vector gradiente de L, esto es,  $\nabla L(x_1, x_2, \dots, x_n, \lambda) = (0, 0, \dots, 0)$ . Lo que se traduce en:

```
>> [x,y,z,p,q]=solve(diff(L,x),diff(L,y),diff(L,z),diff(L,p),diff(L,q),x,y
,z,p,q)
x =
- 2^(1/2)/8 - 1/8
y =
1
z =
2*2^(1/2)
p =
2*2^(1/2)
q =
10 - 4*2^(1/2)
```

Al igualar a cero todas las derivadas parciales y resolver el sistema resultante, se obtienen los valores de  $x_1, x_2, \dots, x_n, \lambda_1, \lambda_2, \dots, \lambda_n$  correspondientes a posibles máximos y mínimos.

Ya tenemos que el posible punto extremo puede ser:

$$(-l+\sqrt{2})/8, 1, 2\sqrt{2}$$

Ahora, vamos a ver qué tipo de extremo es. Para ello, los sustituimos en la función objetivo.

```
>> syms x y z
>> vpa(subs(f,{x,y,z},{-2^(1/2)/8 - 1/8,1,2*2^(1/2)}))
ans =
-1.78388455796197398228741803905
```

Luego, en el punto  $(-l+\sqrt{2})/8, 1, 2\sqrt{2}$  la función presenta un máximo.

*Ejercicio 11. Dada la función  $f(x,y) = 10^{-(x+y)}$  y la transformación  $u=u(x,y)=2x+y$ ,  $v=v(x,y)=x-y$ , hallar  $f(u,v)$ .*

Calculamos la transformación inversa y su jacobiano para aplicar el teorema del cambio de variable:

```
>> [x,y]=solve('u=2*x+y,v=x-y','x','y')
x =
u/3 + v/3
y =
u/3 - (2*v)/3
>> jacobian([u/3 + v/3,u/3 - (2*v)/3],[u,v])
ans =
[ 1/3, 1/3]
[ 1/3,-2/3]
>> f=10^(x-y);
>> pretty(simple(subs(f,{x,y},{u/3 + v/3,u/3 - (2*v)/3})*
abs(det(jacobian([u/3 + v/3,u/3 - (2*v)/3],[u,v])))))
```

$$\frac{10^v}{3}$$

La función pedida es  $f(u,v) = 10^v/3$

*Ejercicio 12. Obtener el desarrollo de Taylor hasta el orden 2 en el origen de coordenadas de la función:*

$$f(x,y) = e^{x+y^2}$$

```
>>
```

```
f=exp (x+y^2)
```

```
>> pretty(simplify(subs(f,{x,y},{0,0})+subs(diff(f,x),{x,y},{0,0})*(x)
subs(diff(f,y),{x,y},{0,0})*(y)+1/2*(subs(diff(f,x,2),{x,y},{0,0})*
(x)^2+subs(diff(f,X,y),{x,y},{0,0})*(x)*(y) +
subs(diff(f,y,2),{x,y},{0,0})*(y)^2)))
```

$$\frac{x^2 + x^2 y^2 + 1}{2}$$

*Ejercicio 13. Expresar en coordenadas cartesianas la superficie que tiene de ecuación  $z=r^2(1+\sin(t))$  en coordenadas cilíndricas.*

```
>> syms X y z r t a
>> f=r^2*(1 + sin(t))

f =
r^2*(sin(t) + 1)

>> cartesianas=simplify(subs(f,{r,t},{sqrt(x^2+y^2),atan(y/x)}))

cartesianas =
(x^2 + y^2)*(y/(x*(y^2/x^2 + 1)^(1/2)) + 1)

>> pretty(cartesianas)
      2   2   /      y          \
      (x + y ) | -----+ 1 |
      | / 2      \1/2
      | | y      |
      | x | - + 1 |
      | | 2      |
      \ | x      /      /
```

*Ejercicio 14. Hallar los vectores tangente, normal principal y binormal (unitarios) de la cónica alabeada:  $x=t$ ,  $y=t^2$ ,  $z=t^3$ .*

Comenzamos trabajando con la variable  $t$  en el campo real mediante:

```
>> x=sym('x','real');
```

Definimos el vector simbólico V como sigue:

```
>> syms t, v= [t, t^2, t^3]
V =
[ t, t^2, t^3]
```

El vector tangente se calculará mediante;

```
>> tan=diff(V)
tan =
[ 1, 2*t, 3*t^2]
```

El vector tangente unitario será:

```
>> tu=simple(tan/sqrt(dot(tan,tan)))
tu =
[1/(1+4*t^2+9*t^4)^(1/2), 2*t/(1+4*t^2+9*t^4)^(1/2), 3*t^2/(1+4*t^2+9*t^4)^(1/2)]
```

Para hallar la normal unitaria de ecuación  $((v' \wedge v'') \wedge v') / (|v' \wedge v''| |v'|)$  se hace:

```
>> v1=cross(diff(V),diff(V,2));
>> nu=simple(cross(vi,tan)/(sqrt(dot(vi,vi))*sqrt(dot(tan,tan))))
nu =
[(-2*t-9*t^3)/(9*t^4+9*t^2+1)^(1/2)/(1+4*t^2+9*t^4)^(1/2),
(1-9*t^4)/(9*t^4+9*t^2+1)^(1/2)/(1+4*t^2+9*t^4)^(1/2),
(6*t^3+3*t)/(9*t^4+9*t^2+1)^(1/2)/(1+4*t^2+9*t^4)^(1/2)]
```

La binormal unitaria es el producto vectorial de la tangente y normal unitarias.

```
>> bu=simple(cross(tu,nu))
bu =
[3*t^2/(9*t^4+9*t^2+1)^(1/2), -3*t/(9*t^4+9*t^2+1)^(1/2), 1/(9*t^4+9*t^2+1)^(1/2)]
```

También puede calcularse la binormal unitaria mediante  $(v' \wedge v'') / |v' \wedge v''|$  como sigue:

```
>> bu=simple(v1/sqrt(dot(vi,vi)))
bu =
[3*t^2/(9*t^4+9*t^2+1)^(1/2),
3*t/(9*t^4+9*t^2+1)^(1/2), 1/(9*t^4+9*t^2+1)^(1/2)]
```

Hemos calculado los elementos del triángulo de Frenet para una curva alabada.

## CAPITULO 10

# INTEGRACION EN UNA Y VARIAS VARIABLES. APLICACIONES

## INTEGRALES

MATLAB permite trabajar con integrales en una y varias variables a través de un grupo de comandos que habilita especialmente para esta tarea. El programa intenta calcular las primitivas de las funciones siempre y cuando no sean muy complicadas algebraicamente. En caso de que el cálculo de la función integral no sea posible de forma simbólica porque no se puede encontrar la función primitiva correspondiente, MATLAB habilita un grupo de comandos que permiten calcular las integrales aproximadas mediante los algoritmos iterativos más comunes del cálculo numérico.

La tabla siguiente muestra los comandos más habituales que utiliza MATLAB para la integración en una y varias variables.

<code>syms x, int(f(x), x) o int('f(x)', 'x')</code>	<i>Calcula la integral indefinida <math>\int f(x)dx</math></i>
	<pre>&gt;&gt; syms x &gt;&gt; int((tan(x))^2,x) ans = tan(x) - x &gt;&gt; int ('(tan(x))^2', 'x') ans = tan(x) - x</pre>

<b>int(int('f(x,y)', 'x'), 'y'))</b>	<i>Calcula la integral doble</i> $\int \int f(x,y) dx dy$ <pre>&gt;&gt; simplify(int(int('(tan(x))^2+(tan(y))^2', 'x'), 'y')) ans = atan(tan(x))*tan(y) - 2*y*atan(tan(x)) + y*tan(x)</pre>
<b>syms x y, int(int(f(x,y), x), y))</b>	<i>Calcula la integral doble</i> $\int \int f(x,y) dx dy$ <pre>&gt;&gt; syms x y &gt;&gt; simplify(int(int((tan(x))^2+(tan(y))^2, x), y)) ans = atan(tan(x))*tan(y) - 2*y*atan(tan(x)) + y*tan(x)</pre>
<b>int(int(int(...int(f(x,y..z)^2, 'x^2), 'y^2)...), 'z^2)</b>	<i>Calcula</i> $\int \int \dots \int f(x,y,\dots,z) dx dy \dots dz$ <pre>&gt;&gt; int(int(int('sin(x+y+z)', 'x'), 'y'), 'z') ans = cos(x + y + z)</pre>
<b>syms x y z, int(int(int(...int(f(x,y..z), x), y)...), z)</b>	<i>Calcula</i> $\int \int \dots \int f(x,y,\dots,z) dx dy \dots dz$ <pre>&gt;&gt; syms x y z &gt;&gt; int (int (int (sin(x+y+z), x), y), z) ans = cos(x + y + z)</pre>
<b>syms x a b, int(f(x), x, a, b)</b>	<i>Calcula la integral definida</i> $\int f(x) dx$ <pre>&gt;&gt; syms x &gt;&gt; int ((cos(x))^2, x, -pi, pi) ans = pi</pre>
<b>int('f(x)', 'x', 'a', 'b')</b>	<i>Calcula la integral definida</i> $\int f(x) dx$ <pre>&gt;&gt; int ('(cos(x))^2', 'x', -pi, pi) ans = pi</pre>

<pre>int(int('f(x,y)', 'x', 'a', 'b'), 'y', 'c', 'd'))</pre>	<p>Calcula la integral <math>\int_a^b \int_c^d f(x,y) dx dy</math></p> <pre>&gt;&gt; int(int(' (cos(x+y))^2', 'x', -pi, pi), 'y', -pi/2, pi/2) ans = pi^2</pre>
<pre>syms X y a b c d, int(mt(f(x,y), x, a, b), y, c, d))</pre>	<p>Calcula <math>\iint f(x,y) dx dy</math></p> <pre>&gt;&gt; syms x y &gt;&gt; int (int ( (cos(x+y))^2, x, -pi, pi) , y, -pi/2, pi/2) ans = pi^2</pre>
<pre>int(int(...int('f(x,y,...)^', 'x', 'a', 'b'), 'y', 'c', 'd'),...), 'z', 'e', 'f')</pre>	<p>Halla <math>\iiint ... \int f(x,y,...,z) dx dy ... dz</math></p> <pre>&gt;&gt; int (int (int ( (cos(x+y+z))^2, x, -pi, pi) , y, -pi/2, pi/2), z, -2*pi, 2*pi) ans = 4*pi^3</pre>
<pre>syms x y z a b c d e f, int(int(int(...int(f(x,y,...,z), x, a, b), y, c, d),...), z, e, f))</pre>	<p>Halla <math>\iiint ... \int f(x,y,...,z) dx dy ... dz</math></p> <pre>&gt;&gt; syms x y z &gt;&gt; int(int(int((cos(x+y+z))^2, x, -pi, pi), y, -pi/2, pi/2), z, -2*pi, 2*pi) ans = 4*pi^3</pre>

## INTEGRALES INMEDIATAS, POR CAMBIO DE VARIABLE Y POR PARTES

Las integrales cuyas funciones primitivas pueden calcularse de forma inmediata, por cambio de variable y por partes son resolubles directamente con el comando *int* de MATLAB.

Como primer ejemplo calculamos las integrales inmediatas siguientes:

$$\int \frac{1}{\sqrt{x^2+1}} dx, \int \frac{\ln(x)}{\sqrt{x}} dx, \int \frac{1}{(2+x)\sqrt{x+1}} dx, \int x^3 \sqrt{1+x^4} dx$$

```
>> pretty(simple(int('1/sqrt(x^2+1)')))

asinh(x)

>> pretty(simple(int('log(x)/x^(1/2)')))

1/2
2 x  (log(x) - 2)

>> pretty(simple(int('1/((2+x)*(1+x)^(1/2))')))

1/2
2 atan( (x + 1) )

>> pretty(simple(int('x^3*sqrt(1+x^4)')))

3
-
2
(x + 1)
-----
6
```

Como segundo ejemplo se calculan las integrales resolubles por cambio de variable siguientes:

$$\int \frac{\arctan \frac{1}{2}x}{4+x^2} dx , \quad \int \frac{\cos^3(x)}{\sqrt{\sin(x)}} dx$$

```
>> pretty(simple(int(atan(x/2)/(4+x^2),x)))

/ x \2
atan| - |
\ 2 /
-----
4

>> pretty(simple(int(cos(x)^3/sin(x)^(1/2),x)))

1/2      2
2 sin(x) (cos(x) + 4)
-----
5
```

Como tercer ejemplo se calculan las integrales siguientes resolubles habitualmente por partes:

$$\int e^x \cos x \, dx, \quad \int (5x^2 - 3)4^{1+3x} \, dx$$

```
>> pretty(simple(int('exp(x)*cos(x)', 'x')))

exp(x) (cos(x) + sin(x))
-----
2

>> pretty(simple(int('(5*x^2-3)*4^(3*x+1)', 'x')))

3x      2      2
3 4   (45 log(4) x - 30 log(4) x - 27 log(4) + 10)
-----
3
27 log(4)
```

## INTEGRALES POR REDUCCIÓN Y CÍCLICAS

MATLAB calcula directamente el valor final de las integrales resolubles por el método de reducción y cíclicas.

En el método por reducción suele hacerse inicialmente una integral por partes que ofrece una parte ya integrada y otra parte similar a la integral inicial pero reducida de exponente hasta que es posible ser calculada.

En las integrales cíclicas se aplica el método de reducción varias veces consecutivas hasta que se obtiene una integral igual a la inicial, lo que permitirá que la integral inicial sea despejada algebraicamente.

Como ejemplo se resuelven las integrales siguientes;

$$\int \sin^{13}(x)\cos^{15}(x)dx, \int x^7(m+nx)^{1/2} dx, \int e^{12}x\cos^{11}(x) dx$$

```
>> I=int('sin(x)^13*cos(x)^15', 'x')

I =

cos(x)^18/6006 - (cos(x)^16*sin(x)^12)/28 - (3*cos(x)^16*sin(x)^10)/182 - 
(5*cos(x)^16*sin(x)^8)/728 - (5*cos(x)^16*sin(x)^6)/2002
(3*cos(x)^16*sin(x)^4)/4004 - (3*cos(x)^16)/16016

>> pretty(simple(I))

16      2      12      10      8      6
(cos(x) (8 cos(x) - 1716 sin(x) - 792 sin(x) - 330 sin(x) - 120 sin(x)
4
36 sin(x) - 9))/48048
>> syms x m n
>> pretty(simple(int(x^7*sqrt(m+n*x), x)))
```

```
/      3
|      -
|      2          7          6          5 2 2          3 4 4
\ 2 (m + n x) (- 2048 m + 3072 m n x - 3840 m n x + 4480 m n x - 5040 m n x +
           \           |
           2 5 5      6 6      7 7 |           8
5544 m n x - 6006 m n x + 6435 n x ) // (109395 n )
```

```
>> pretty(simple(int('exp(12*x)*cos(x)^11','x')))

           11          10          9
(exp(12 x) (2894432580 cos(x) + 2653229865 cos(x) sin(x) + 1415055928 cos(x) +
           8          7          6
1061291946 cos(x) sin(x) + 527896512 cos(x) + 307939632 cos(x) sin(x) +
           5          4          3
131193216 cos(x) + 54663840 cos(x) sin(x) + 17149440 cos(x) + 709632 cos(x) -
           3
4287360 sin(x) + 4346496 sin(x)) / 63918719475
```

## INTEGRALES RACIONALES, IRRACIONALES Y BINOMIAS

Las integrales cuyas primitivas son calculables tratándolas como funciones racionales, irracionales y binomias pueden ser calculables en MATLAB de modo directo con el comando *int*.

Como ejemplo calculamos las integrales siguientes:

$$\int \frac{3x^4+x^2+8}{x^4-2x^2+1} dx, \int \frac{\sqrt{9-x^2}}{x} dx, \int x^8(3+5x^3)^{1/4} dx$$

La primera integral es un caso típico de integral racional

```
>> I1=int((3*x^4+x^2+8)/(x^4-2*x^2+1),x)

I 1   =
2*x - (6*x)/(x^2 - 1) + atan(x*i)*i

>> pretty(simple(I1))

 6x
3x - ----- + atan(x i) i
      2
      x - 1
```

La segunda integral es un caso típico de integral irracional

```
>> I2=int(sqrt(9-4*x^2)/x,x)
```

```
I2 =
3*acosh(-(3*(1/x^2)^(1/2))/2) + 2*(9/4 - x^2)^(1/2)
>> pretty(simple(I2))
      /   / 1 \1/2 \
      |   3 | - |   |
      |   | 2 |   |
      |   \ x /   |
2 acosh | ----- | + (9 - 4 x )^2 1/2
      \ 2           /
La tercera integral es un caso típico de integral binomia
```

```
>> I3=int (x^8*(3+5*x^3)^(1/4),x)
I3 =
(5*x^3 + 3)^(1/4)*((4*x^9)/39 + (4*x^6)/585 - (32*x^3)/4875 + 128/8125)
>> pretty(simple(I3))
      5
      -
      4
      3      6      3
4 (5 x + 3) (375 x - 200 x + 96)
-----
73125
```

## **INTEGRAL DEFINIDA Y APLICACIONES. UNA VARIABLE**

La importancia de la integral definida radica en sus aplicaciones directas al cálculo de longitudes de arcos de curvas, áreas comprendidas entre curvas, áreas de superficies de revolución, volúmenes de revolución y otras áreas y volúmenes.

### ***Longitud de un arco de curva***

Una de las aplicaciones más comunes del cálculo integral es hallar longitudes de arcos de curva.

Para una curva plana de ecuación  $y=f(x)$ , la longitud del arco de curva comprendida entre los puntos de abscisas  $x=a$  y  $x=b$  viene dada por la expresión:

$$L = \int_a^b \sqrt{[1 + f'(x)]^2} dx$$

Para una curva plana con coordenadas paramétricas  $x=x(t)$   $y=y(t)$ , la longitud del arco de curva comprendida entre los puntos relativos a los valores  $t=t_0$  y  $t=t_1$  del parámetro viene dada por la expresión:

$$L = \int_0^{t_1} \sqrt{x't^2 + y't^2} dt$$

Para una curva en coordenadas polares de ecuación  $r=f(a)$ , la longitud del arco de curva comprendida entre los puntos relativos a los valores  $a=a_0$  y  $a=a_1$  del parámetro viene dada por la expresión:

$$L = \int_{a_0}^{a_1} \sqrt{r^2 + r'a^2} dt$$

Para una curva alabeada con coordenadas paramétricas  $x=x(t)$   $y=y(t)$   $z=z(t)$ , la longitud del arco de curva comprendida entre los puntos relativos a los valores  $t=t_0$  y  $t=t_1$  del parámetro viene dada por la expresión:

$$L = \int_0^{t_1} \sqrt{x't^2 + y't^2 + z't^2} dt$$

Para una curva alabeada en coordenadas cilíndricas de ecuaciones  $x=r\cdot\cos(a)$ ,  $y=r\cdot\sin(a)$ ,  $z=z$ , la longitud del arco de curva comprendida entre los puntos relativos a los valores  $a=a_0$  y  $a=a_1$  del parámetro viene dada por la expresión:

$$L = \int_{a_0}^{a_1} \sqrt{r^2 + y'^2 + z'^2} dr$$

Para una curva alabeada en coordenadas esféricas de ecuaciones dadas por  $x=r\cdot\sin(a)\cdot\cos(b)$ ,  $y=r\cdot\sin(a)\cdot\sin(b)$ ,  $z=r\cdot\cos(a)$ , se tiene que la longitud del arco de curva comprendido entre los puntos relativos a los valores  $a=a_0$  y  $a=a_1$  del parámetro viene dada por la expresión:

$$L = \int_{a_0}^{a_1} \sqrt{dr^2 + r^2 da^2 + r^2 \sin^2(a) db^2} dt$$

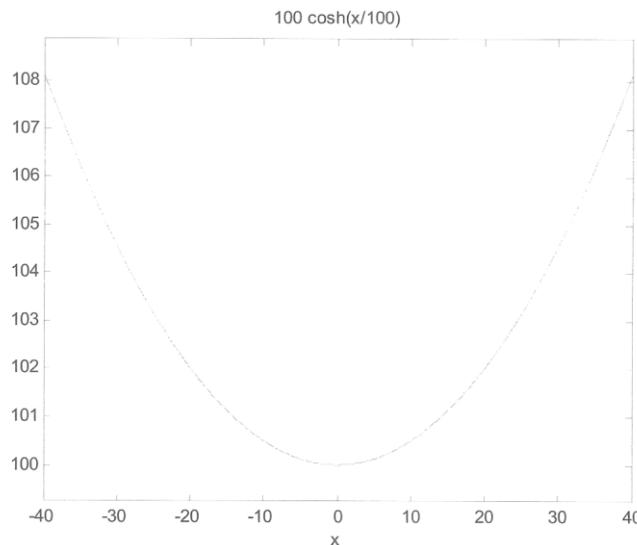
Como primer ejemplo consideramos un cable eléctrico que cuelga entre dos torres que están separadas 80 metros. El cable adopta la posición de una catenaria cuya ecuación es:

$$100 \cosh(x/100) - 1$$

Se trata de calcular la longitud del arco de cable entre las dos torres.

Comenzamos realizando la representación gráfica de la catenaria en el intervalo [-40,40].

```
>> syms x  
>> ezplot(100*cosh(x/100), [-40, 40])
```



La longitud de la catenaria se calcula como sigue:

```
>> f=100*cosh(x/100)
f =
100*cosh(x/100)
>> pretty(simple(int((1+diff(f,x)^2)^(1/2),x,-40,40)))
/ 2 \
200 sinh| - |
\ 5 /
```

Si queremos el resultado aproximado hacemos:

```
>> vpa(int((1+diff(f,x)^2)^(1/2),x,-40,40))
ans =
82.15046516056310170804200276894
```

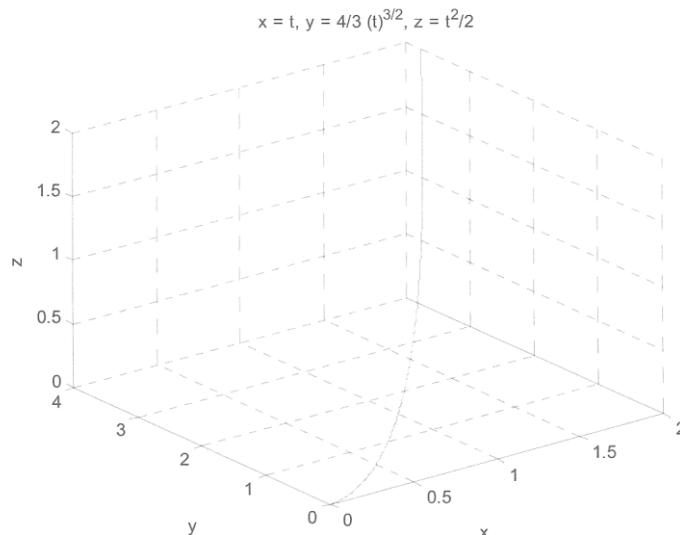
Como segundo ejemplo calculamos la longitud de la curva espacial representada por las ecuaciones paramétricas:

$$x = t, \quad y = 4/3 \cdot t^{3/2}, \quad z = t^2/2$$

desde  $t=0$  hasta  $t=2$ .

Comenzamos haciendo la representación gráfica de la curva paramétrica para hacemos una idea del problema.

```
>> ezplot3('t',' 4/3*(t)^(3/2)', 't^2/2', [0,2])
```



La longitud pedida se calcula de la forma siguiente:

```
>>
pretty(simple(int('(diff(t,t)^2+diff((4/3)*t^(3/2),t)^2+diff((1/2)*t^2,t)^2)^2/3,t,0,2)))
Log | /      3          \ |
| | - +----- - - | + 2 13 - 1
| | | 3          |   1/2
| | |----- - - |   3 | |
| | | | 1/2      1/2 \2 |   |
| | | | 4 3      39 |   |
| | |-----+----- |   |
\ \ 3          3 /   /
```

Que se aproxima mediante:

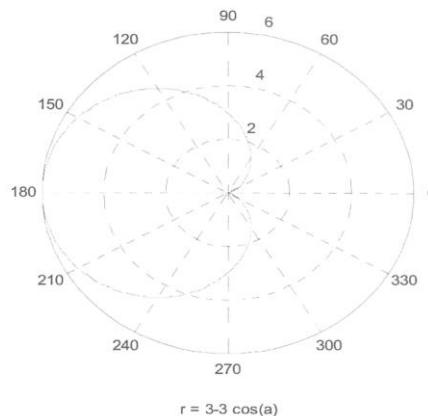
```
>>
vpa(simple(int('(diff(t,t)^2+diff((4/3)*t^(3/2),t)^2+diff((1/2)*t^2,t)^2)^2
^(1/2)'t',0,2)))
```

```
ans =
4.8157033654708774362587208533845
```

Como tercer ejemplo hallamos la longitud del arco definido para valores desde  $a=0$  hasta  $a=2\pi$  en la cardioide de ecuaciones polares  $r=3-3\cos(a)$ .

Comenzamos representando la curva para hacemos un idea gráfica de la longitud pedida.

```
>> ezpolar('3-3*cos(a)', [0, 2*pi])
```



Ahora calculamos la longitud pedida.

```
>> r='3-3*cos(a)';
>> diff(r,'a')
ans =
3*sin(a)
>> R=simple(int(((3-3*cos(a))^2+(3*sin(a))^2)(1/2)'a','0','2*pi'))
R =
24
```

### **Superficie entre dos curvas**

Otra de las aplicaciones habituales del cálculo integral es el cálculo de superficies comprendidas entre curvas.

El área comprendida entre una curva de ecuación  $y=f(x)$  y el eje  $x$  viene dada, de forma general, por la integral:

$$S = \left| \int_a^b f(x) dx \right|$$

siendo  $x=a$  y  $x=b$  las abscisas de los puntos de corte de la curva con el eje de abscisas.

Si la curva viene en coordenadas paramétricas  $x=x(t)$   $y=y(t)$ , el área viene dada por la integral:

$$S = \left| \int_a^b y(t) x'(t) dt \right|$$

para los valores correspondientes del parámetro  $t=a$  y  $t=b$ , en los puntos de corte.

Si la curva viene en coordenadas polares  $r=f(a)$ , el área viene dada por la integral:

$$S = \frac{1}{2} \int_a^b f(a)^2 da$$

para los correspondiente valores  $a_0$  y  $a_1$  del parámetro  $a$  en los puntos de corte.

Si se quiere calcular el área comprendida entre dos curvas de ecuaciones  $y=f(x)$  e  $y=g(x)$ , utilizaremos la integral:

$$S = \int_a^b |f(x) - g(x)| dx$$

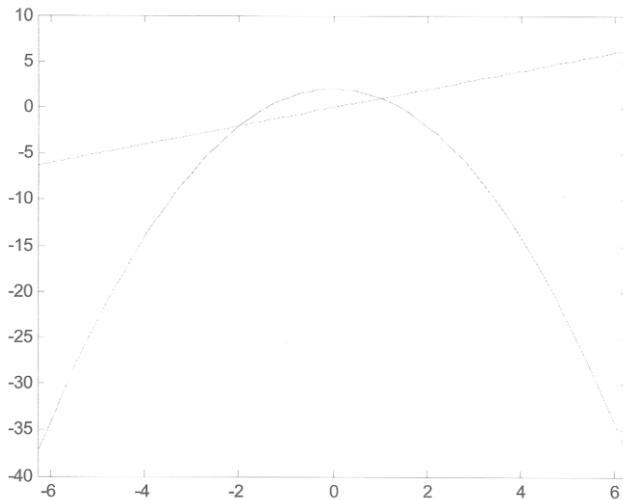
siendo  $x=a$  y  $x=b$  las abscisas de los puntos de corte de las dos curvas.

Es muy importante tener en cuenta el signo de las funciones a la hora de delimitar los recintos para el cálculo de las correspondientes áreas. Nunca se deben sumar valores negativos de áreas con valores positivos. Se dividirá la región de integración en las subregiones necesarias para que no se computen simultáneamente áreas positivas y negativas. Las áreas que resulten negativas se consideran en módulo.

Como primer ejemplo calculamos el área delimitada por las dos curvas siguientes:  $f(x)=2-x^2$  y  $g(x)=x^3$

Comenzamos realizando la representación gráfica.

```
>> fplot(' [2-x^2,x] ', [-2*pi,2*pi])
```



Será necesario conocer los puntos de corte de ambas curvas. Se calculan de la forma siguiente:

```
>> solve('2-x^2=x')
ans =
-2
1
```

Ya estamos en condiciones de calcular el área pedida.

```
>> int (' 2-x^2-x','x',-2,1)
ans =
9/2
```

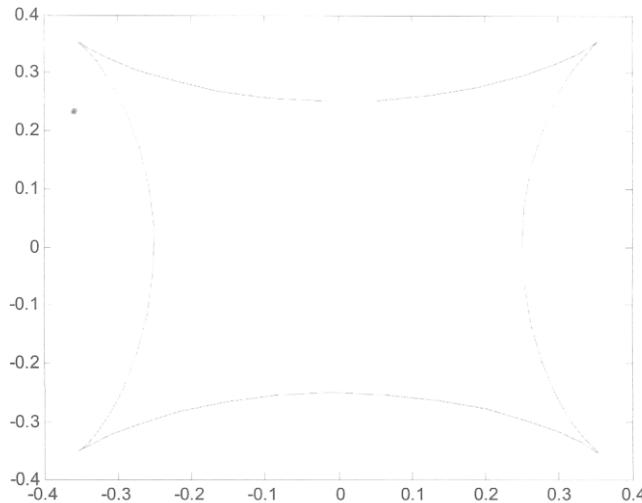
Como segundo ejemplo calculamos la longitud y el área comprendida por las curvas paramétricas:

$$x(t) = \frac{\cos(t)[2-\cos(2t)]}{4} \quad y(t) = \frac{\sin(t)[2+\cos(2t)]}{4}$$

Comenzamos realizando una representación gráfica del área pedida.

```
>> x=(0:.1:2*pi);
>> t=(0:.1:2*pi);
>> x=cos(t).*(2-cos (2*t))./4;
```

```
>> y=sin(t).*(2+cos(2*t))./4;
>> plot (x,y)
```



Se trata de una figura cerrada con variación del parámetro entre  $0$  y  $2\pi$  y tenemos que calcular su longitud y el área que encierra. Para evitar que se compensen áreas positivas con áreas negativas trabajaremos sólo con el primer medio cuadrante y multiplicaremos por  $8$  el resultado.

Para la longitud tenemos:

```
>> A=simple(diff('cos(t)*(2-cos(2*t))/4'))
A =
- (3*sin(t)*(2*sin(t)^2 - 1))/4
>> B=simple(diff('sin(t)*(2+cos(2*t))/4'))
B =
(3*cos(t)^3)/2 - (3*cos(t))/4
>> L=8*int('sqrt((-3*sin(t)*(2*sin(t)^2 - 1))/4)^2+( (3*cos(t)^3)/2
(3*cos(t))/4)^2)'t',0,pi/4)
L =
3
```

Para el área tenemos:

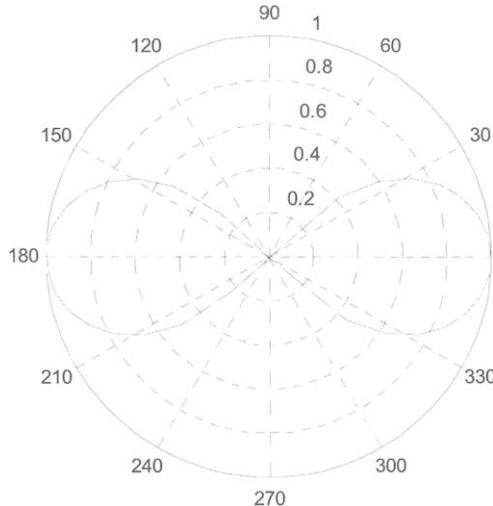
```
>> S=8*simple(int(' (sin(t)*(2+cos(2*t))/4)*(-3/8*sin(t)+3/8*sin(3*t)) ',  
't',0,pi/4))  
  
S =  
-(3*pi - 16)/32
```

Como tercer ejemplo calculamos la longitud y el área encerrada por la curva en polares de ecuación:

$$r = \sqrt{\cos(2\alpha)}$$

Comenzamos representando la curva en coordenadas polares para hacemos una idea de gráfica del problema.

```
>> a=0:.1/2*pi;  
>> r=sqrt(cos(2*a));  
>> polar(a,r)
```



Observamos que la curva repite 4 veces su estructura para  $\alpha$  entre 0 y  $\pi/4$ . Calculamos la superficie  $S$  encerrada por ella de la siguiente forma:

```
>> clear all  
>> syms a  
>> S=4*(1/2)*int((sqrt(cos(2*a)))^2,a,0,pi/4)  
  
S =  
1
```

## Superficies de revolución

El área engendrada por la curva de ecuación  $y=f(x)$ , al girar alrededor del eje x, viene dada por la integral:

$$S = 2\pi \int_a^b f(x) \sqrt{1 + f'(x)^2} dx \quad y \quad S = 2\pi \int_{t_0}^{t_1} y(t) \sqrt{x'(t)^2 + f'(t)^2} dt$$

en coordenadas paramétricas, siendo  $x=a$  y  $x=b$  los valores de las abscisas de los puntos que delimitan el arco de curva que gira ( $t_0$  y  $t_1$  son los valores del parámetro correspondiente a esos puntos).

El área engendrada por la curva de ecuación  $x=f(y)$ , al girar alrededor del eje y, viene dada por la integral:

$$S = \int_a^b f(y) \sqrt{1 + f'(y)^2} dy$$

siendo  $y=a$  e  $y=b$  los valores de las ordenadas de los puntos que delimitan el arco de curva que gira.

Como primer ejemplo calculamos el área engendrada al girar la cúbica de ecuación  $12x-9x^2+2x^3$  alrededor del eje OX, comprendida entre  $x=0$  y  $x=5/2$ .

```
>> I=2*pi*vpa(int('(2*x^3-9*x^2+12*x)*sqrt(1+(6*x^2-18*x+12)^2)', 'x', 0, 5/2))
Warning: Explicit integral could not be found.
```

```
I =
50.366979807381457286723333012298*pi
```

Como segundo ejemplo calculamos el área engendrada al girar la curva paramétrica dada por  $x(t) = t - \sin(t)$ ,  $y(t) = 1 - \cos(t)$  alrededor del eje OX, comprendida entre  $t=0$  y  $t=2\pi$ .

```
>> clear all
>> syms t
>> x=t-sin(t)

x =
t - sin(t)

>> y= 1-cos(t)
y =
1 - cos (t)
```

```
>> V=2*pi*int(y*(sqrt (diff (x,t)^2+diff(y,t)^2)),t,0,2*pi)
V =
(64*pi)/3
```

Aproximando el resultado tenemos:

```
>> V=vpa(2*pi*int(y*(sqrt(diff(x,t)^2+diff(y,t)^2)),t,0,2*pi))
V =
67.020643276582255753869725509963
```

### **Volúmenes de revolución**

El volumen engendrado por la curva de ecuación  $y=f(x)$ , al girar alrededor del eje  $x$ , viene dado por la integral:

$$V = \pi \int f(x)^2 dx$$

siendo  $x=a$  y  $x=b$  los valores de las abscisas de los puntos que delimitan el arco de curva que gira.

El volumen engendrado por la curva de ecuación  $x=f(y)$ , al girar alrededor del eje  $y$ , viene dado por la integral:

$$V = \pi \int f(y)^2 dy$$

siendo  $y=a$  e  $y=b$  los valores de las ordenadas de los puntos que delimitan el arco de curva que gira.

Si seccionamos una superficie por planos paralelos a uno de los tres planos coordenados (por ejemplo, al plano  $z=0$ ) y su ecuación, podemos ponerla en función de la distancia al origen (en este caso  $z$ ), es decir, la ecuación de la superficie se puede poner de la forma  $S(z)$ , entonces el volumen comprendido por los planos y la superficie viene dado por:

$$V = \int_{z_1}^{z_2} S(z) dz$$

Como primer ejemplo calculamos el volumen engendrado al girar la elipse:

$$x^2/4 + y^2/9 = 1$$

alrededor del eje OX y alrededor del eje OY.

Los volúmenes pedidos se calculan por integración de la siguiente forma:

```
>> V1=int('pi*9*(1-x^2/4)'x',-2,2)
```

V1 =

24\*pi

```
>> V2=int('pi*4*(1-y^2/9)'y',-3,3)
```

V2 =

16\*pi

Como segundo ejemplo calculamos el volumen engendrado al girar la curva en coordenadas polares  $r = 1 + \cos(a)$  alrededor del eje OX.

La curva viene en coordenadas polares, pero no hay ningún problema en calcular los valores de  $x(a)$  e  $y(a)$  necesarios para aplicar la fórmula conocida en coordenadas cartesianas. Tenemos:

$$x(a) = r(a)\cos(a) = (1 + \cos(a))\cos(a)$$

$$y(a) = r(a)\sin(a) = (1 + \cos(a))\sin(a)$$

$$\int \pi y(a)^2 dx(a) = \int \pi (1 + \cos(a))^2 (\sin(a))^2 [(1 + \cos(a))(-\sin(a)) - \sin(a)\cos(a)] da$$

El volumen pedido se calculará de la siguiente forma:

```
>> x='(1+cos(a))*cos(a)';
```

```
>> y='(1+cos(a))*sin(a)';
```

```
>> A=simple(diff(x))
```

A =

$$-\sin(2a) - \sin(a)$$

```
>> V=pi*abs(int('((1+cos(a))*sin(a))^2*(-sin(2*a)-sin(a))','a',0,pi))
```

V =

$$(8\pi)/3$$

## Integrales curvilíneas

Sea  $\bar{F}$  un campo vectorial en  $\mathbb{R}^3$  continuo sobre la trayectoria continua con derivada continua  $c:[a,b] \rightarrow \mathbb{R}^3$ . Definimos  $\int_c \bar{F} \cdot ds$ , y la denominamos la integral de

Línea de  $\bar{F}$  a lo largo de la curva  $c$ , como sigue:

$$\int_c \bar{F} \cdot ds = \int_c \bar{F}[c(t)] \cdot c'(t) dt$$

Como primer ejemplo consideramos la curva  $c(t) = [\sin(t), \cos(t), t]$  con  $0 < t < 2\pi$ , y sea el campo vectorial  $F(x, y, z) = \bar{x}i + \bar{y}j + \bar{z}k$ . Calcularemos la integral curvilínea:

$$\int_c \bar{F} \cdot ds$$

```
>> syms t
>> pretty(int(dot([sin(t),cos(t),t],[cos(t),-sin(t),1]),t,0,2*pi))
```

$2 \pi$

Como segundo ejemplo calculamos la integral:

$$\int_c [sen(z)dx + cos(z)dy - \sqrt[3]{xy}dz]$$

según la trayectoria de ecuaciones paramétricas:

$$x = \cos^3(a), \quad y = \sin^3(a), \quad z = a, \quad 0 < a < 7\pi/2$$

```
>> syms a
>> [diff(cos(a)^3), diff(sin(a)^3), diff(a)]
ans =
[ -3*cos(a)^2*sin(a), 3*cos(a)*sin(a)^2, 1]
>> int((dot([sin(a),cos(a),-sin(a)^3*cos(a)^3],[-3*cos(a)^2*sin(a),3*sin(a)^2*cos(a),1]))^(1/3),a,0,7*pi/2)
ans =
2* (-1)^(1/3) + 3/2
```

```
>>pretty(int(([sin(a),cos(a),-sin(a)^3*cos(a)^3],  
[-3*cos(a)^2*sin(a),3*sin(a)^2*cos(a),1]))^(1/3),a,0,7*pi/2))  
  
1  
-  
3 3  
2 (-1) + -  
2  
>>vpa(int(([sin(a),cos(a),-sin(a)^3*cos(a)^3],  
[-3*cos(a)^2*sin(a),3*sin(a)^2*cos(a),1]))^(1/3),a,0,7*pi/2))  
  
ans =  
1.7320508075688772935274463415059*i + 2.5
```

Como tercer ejemplo calculamos la integral:

$$\int_c x^3 dy - y^3 dx$$

sobre la trayectoria definida por la circunferencia  $x^2 + y^2 = a^2$

```
>>pretty(int('a*cos(t)^3*diff(a*sin(t),t)-a*sin(t)^3  
*diff(a*cos(t),t)' 't',0,2*pi))  
  
3 pi a  
-----  
2
```

## INTEGRALES IMPROPIAS

MATLAB trabaja con las integrales impropias considerándolas como cualquier otro tipo de integral definida. No vamos a tratar aquí temas teóricos de convergencia de integrales impropias, pero dentro de las integrales impropias distinguiremos dos tipos;

1) **Integrales con límites infinitos:** El campo de definición de la función integrando es el semieje cerrado  $[a, \infty)$  o el  $(-\infty, a]$  o el  $(-\infty, \infty)$ .

2) **Integrales de funciones discontinuas:** La función dada es continua en todo el intervalo  $[a, b]$ , a excepción de en un número finito de puntos aislados llamados singulares.

También pueden presentarse casos complicados, combinación de los dos anteriores. También se puede generalizar al caso en el que el campo de definición de la función (el recinto de integración) sea el conjunto de valores de una función (**integral de Stieljes**), pero esto sería materia de cursos completos de análisis matemático.

Como ejemplo, calculamos los valores de las siguientes integrales;

```

 $\int_0^b \frac{dx}{\sqrt{x}}, \quad \int_0^\infty \frac{e^{-x} \sin x}{x} dx$ 

>> syms x a b
>> pretty(limit(int(1/sqrt(x), x, a, b), a, 0))
1 / 2
2 b
>> pretty(simple(int(exp(-x)*sin(x)/x, x, 0, inf)))
pi
-- 
4

```

## INTEGRALES DEPENDIENTES DE UN PARÁMETRO

Dada la función de variable  $y$ .  $\int f(x, y) dx = F(y)$  que está definida en el intervalo  $c < y < e$ , siendo la función  $f(x, y)$  continua en el rectángulo  $[a, b] \times [c, e]$  con derivada parcial respecto de  $y$  continua en el rectángulo, entonces se cumple que para todo  $y$  del intervalo  $c < y < e$ :

$$\frac{d}{dy} \left[ \int_a^b f(x, y) dx \right] = \int_a^b \frac{d}{dy}(x, y) dx$$

Este resultado es muy importante, pues permite intercambiar derivada con integral, para poder derivar bajo el signo integral.

Las integrales dependientes de un parámetro también pueden ser impropias, y además también los límites de integración pueden depender del parámetro.

Si los límites de integración dependen del parámetro, podemos asegurar lo siguiente:

$$\frac{d}{dy} \left[ \int_a^b f(x, y) dx \right] + b'(y) \cdot f[b(y), y] - a'(y) \cdot f[a(y), y]$$

siempre y cuando  $a(y)$  y  $b(y)$  estén definidas en el intervalo  $[c, e]$  y tengan derivadas continuas  $a'(y)$  y  $b'(y)$ , estando también las curvas  $a(y)$  y  $b(y)$  contenidas en el rectángulo  $[a, b] \times [c, e]$ .

Además si la función  $\int_a^b f(x,y)dx = F(y)$  está definida en el intervalo  $[c,e]$  y  $f(x,y)$  es continua en el rectángulo  $[a,b] \times [c,d]$ , entonces se cumple la fórmula:

$$\int_c^e \int_a^b f(x,y)dydx = \int_a^b \int_c^e f(x,y)dydx$$

es decir, se permite la integración bajo el signo integral, y el orden de integración respecto de las variables es indiferente.

Como ejemplo, resolvemos por derivación respecto al parámetro  $a > 0$  las integrales:

$$\int_0^\infty \frac{\arctan(ax)}{x(1+x^2)} dx, \quad \int_0^\infty \frac{1-e^{-x^2}}{x^2} dx$$

Para la primera integral, como la función integrando cumple las condiciones para la derivación bajo el signo integral, partimos de la integral de la derivada respecto del parámetro  $a$  de la función integrando. El resultado final será la integral respecto del parámetro  $a$  del resultado de la integral de la derivada anterior. Si llamamos a la integral  $I(a)$ , calculamos que normalmente es más fácil de calcular que  $I(a)$ , y el resultado final será  $\int I'(a) da$ .

El principal apoyo nos lo presta el hecho de poder derivar la integral como la integral de la derivada, lo cual reduce la dificultad de la integración.

```
>> a = sym('a','positive')

a =

>> pretty(simple(sym((int(diff(atan(a*x)/(x*(1+x^2)),a),x,0,inf)))))

pi
-----
2 a + 2
```

Ahora, integramos esta función respecto de la variable  $a$ , para hallar  $I(a)$ :

```
>> pretty(simple(sym(int(pi/(2*a+2),a))))
```

```
pi log(a + 1)
-----
2
```

Para resolver la segunda integral, definimos la integral de parámetro  $a$ :

$$f(a) = \int_0^{\infty} \frac{1-e^{-ax^2}}{x^2} dx$$

Calculamos  $I'(a)$ . Posteriormente, integramos  $I'(a)$  derivando bajo el signo integral, ya que la función  $I(a)$  lo permite. A continuación, integramos en  $a$  el resultado anterior para obtener la función  $s(a)$ . La integral problema será  $s(1)$ :

```
>> Ia=simple(sym(int(diff((1-exp(-a*x^2))/x^2,a),x,0,inf)))

Ia =

(pi/a)^(1/2)/2

>> s=simple(sym(int(Ia,a)))

s =

(pi*a)^(1/2)
```

Al hacer  $a=1$ , tenemos que  $s(1) = I(a) = \sqrt{\pi}$ .

## INTEGRACIÓN NUMÉRICA APROXIMADA

MATLAB contiene funciones que permiten realizar integración numérica mediante los métodos de Simpson y Lobato. A continuación se presenta la sintaxis de estas funciones:

<b>q = quad(f,a,b)</b>	<i>Hallar la integral de f entre a y b por el método de Simpson con un error de <math>10^{-6}</math></i> $\gg F = @(x) 1./(x.^3-2*x-5) ;$ $Q = quad(F, 0, 2)$ $Q =$ $-0.4605$
<b>q = quad(f,a,b,tol)</b>	<i>Hallar la integral de f entre a y b por el método de Simpson con la tolerada tol en vez de con un error de <math>10^{-6}</math></i> $\gg Q = quad(F, 0, 2, 1.0e-20)$ $Q =$ $-0.4607$
<b>q = quad(f,a,b,tol,trace)</b>	<i>Hallar la integral de f entre a y b por el método de Simpson con la tolerancia tol y presentando la traza de las iteraciones</i>
<b>q = quad(f,a,b,tol,trace,p1,p2,...)</b>	<i>Pasa los argumentos adicionales p1, p2, ... a la función f,f(x,p1,p2, ...)</i>

<b>q = quadl(f,a,b)</b>	<i>Hallar la integral de f entre a y b por el método de la cuadratura de Lobato con un error de <math>10^{-6}</math></i> <pre>&gt;&gt; clear all &gt;&gt; syms x &gt;&gt; f=inline(1/sqrt(x^3+x^2+1)) f = Inline function: f(x) = 1.0./sqrt(x.^2+x.^3 + 1.0) &gt;&gt; Q = quadl(f,0,2) Q = 1.2326</pre>
<b>q = quadl(f,a,b,tol)</b>	<i>Hallar la integral def entre a y b por el método de la cuadratura de Lobato con la tolerancia tol en vez de con un error de <math>10^{-6}</math></i> <pre>&gt;&gt; Q = quadl(f,0,2,1.0e-25) Q = 1.2344</pre>
<b>q = quadl(f,a,b,tol,trace)</b>	<i>Hallar la integral de f entre a y b por el método de la cuadratura de Lobato con la tolerancia tol y presentando la traza de las iteraciones</i>
<b>q = quad(f,a,b,toI,trace,p1,p2,...)</b>	<i>Pasa los argumentos adicionales p1, p2, ... a la función ff(x,p1,p2, ...)</i>
<b>[q,fcnt] = quadl(f,a,b,...)</b>	<i>Devuelve adicionalmente el número de evaluaciones de f</i>
<b>q = dblquad(f,xmin,xmax,ymin,ymax)</b>	<i>Halla la integral doble de f(x,y) en el campo de variación de (x,y) especificado con un error de</i> <pre>&gt;&gt; clear all &gt;&gt; syms x y &gt;&gt; z = inline(y*sin(x)+x*cos(y)) z = Inline function: z(x,y) = x.*cos(y)+y.*sin (x) &gt;&gt; D=dblquad(z,-1,1,-1,1) D = 1.0637e-016</pre>
<b>q = dblquad(f,xmin,xmax,ymin,ymax,tol)</b>	<i>Halla la integral doble de f(x,y) en el campo de variación de (x,y) especificado con la tolerancia tol</i>
<b>q = dblquad(f,xmin,xmax,ymin,ymax,tol,@quadl)</b>	<i>Halla la integral doble de f(x,y) en el campo de variación de (x,y) especificado con la tolerancia tol y el método quadl</i>

**q=dblquad(f,xmin,xmax,  
ymin,ymax,tol,method,p1,p2,..)**

Pasa los argumentos adicionales  $p1, p2, \dots$  a la función  $f$

Como primer ejemplo calculamos  $\int_0^2 \frac{1}{x^3 - 2x - 5} dx$  por el método de Simpson.

```
>> F = inline('1./(x.^3-2*x-5)') ;
>> Q = quad(F, 0, 2)
```

```
Q =
-0.4605
```

A continuación se observa que se mantiene el valor de la integral aunque aumentemos la tolerancia a  $10^{-18}$ .

```
>> Q = quad(F, 0, 2, 1.0e-18)
```

```
Q =
-0.4605
```

En el ejemplo siguiente se evalúa la integral mediante el método de Lobato.

```
>> Q = quadl(F, 0, 2)
```

```
Q =
-0.4605
```

Después evaluamos la integral doble  $\int_{\pi}^{2\pi} \int_0^{\pi} (ysen(x) + xcos(y)) dy dx$ .

```
>> Q = dblquad(inline('y*sin(x)+x*cos(y)'), pi, 2*pi, 0, pi)
```

```
Q =
-9.8696
```

## INTEGRALES ESPECIALES

MATLAB dispone en su módulo básico de un grupo importante funciones integrales especializadas para facilitar la computación. Incluye funciones integrales gamma, beta, elípticas, etc. En la tabla siguiente se exponen las más importantes.

**gamma(A)**

Función gamma  $\Gamma(a) = \int_0^{\infty} e^{-t} t^{a-1} dt$

<b>gammainc(X,A)</b>	Función gamma incompleta $\Gamma(x, a) = \frac{1}{\Gamma(a)} \int_0^x e^{-t} t^{a-1} dt$
<b>gammaln (A y X vectores)</b>	Logaritmo de la función gamma Log $\Gamma(a)$ <i>Logaritmo de la función gamma Log <math>\Gamma(a)</math></i>
<b>beta(Z,W) betainc(X,Z,W) betaln(X,Z,W)</b>	Función beta $\beta(z, w) = \int_0^1 (1-t)^{w-1} t^{z-1} dt = \frac{\Gamma(z)\Gamma(w)}{\Gamma(z+w)}$ Beta incompleta $I_x(z, w) = \frac{1}{\beta(z,w)} \int_0^x (1-t)^{w-1} t^{z-1} dt$ <i>Logaritmo de la función beta Log <math>\beta(z,w)</math></i>
<b>[SN,CN,DN] = ellipj(U,M)</b>	Funciones elípticas de Jacobi $u = \int_0^\Phi \frac{1}{\sqrt{1-msin^2(\phi)}} d\phi$ $SN(u) = \sin((\phi)), CN(u) = \cos(\phi), DN(u) = \sqrt{1-msin^2(\Phi)}$
<b>K = ellipke(M) [K,E] = ellipke(M)</b>	Integrales elípticas completas de $I^a(K)$ y $2^a(M)$ clase $k(m) = \int_0^1 \sqrt{\frac{1-t^2}{1-mt^2}} dt = \int_0^{\pi/2} \frac{1}{\sqrt{1-msin^2(\phi)}} d\phi = F(\pi/2, m)$ $e(m) = \int_0^1 \sqrt{\frac{1-t^2}{1-mt^2}} dt = \int_0^{\pi/2} \sqrt{1-msin^2(\phi)} d\phi = E(\pi/2, m)$
<b>erf(X) = función de error erfc(X) = complementaria de función de error erfcx(X) = complementaria escalada de error erfinv(X) = inversa de error</b>	$erf(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = 2F_{N(0,1/2)}(x) \quad F = \text{distribución normal}$ $erfc(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt = 1 - erf(x)$ $erfcx(x) = e^{x^2} erfc(x) = \frac{1}{\sqrt{\pi}} \frac{1}{x}$ $x = erfinv(y) \leftrightarrow y = erf(x)$
<b>expint(X)</b>	Exponencial integral $expint(x) = \int_x^\infty \frac{e^{-t}}{t} dt$

Como primer ejemplo calculamos los valores de la función de distribución de una normal (0,1/2) en los números entre 0 y 1 distanciados 1/4.

```
>> erf(0:1/4:1)
```

```
ans =
0           0.28          0.52          0.71          0.84
```

A continuación calculamos los valores de la función gamma en los cuatro primeros números pares.

```
>> gamma([2 4 6 8])
ans =
1.0           6.00          120.00         5040.00
```

Teniendo presente el resultado anterior comprobamos que  $\Gamma(a) = (a-1)!$  para los cuatro primeros pares

```
>> [factorial(1),factorial(3),factorial(5),factorial(7)]
ans =
1.0           6.00          120.00         5040.00
```

A continuación, para  $z = 3$  y  $w = 5$ , comprobaremos que:

$$\begin{aligned} \text{beta}(z,w) &= \exp(\text{gammaln}(z)+\text{gammaln}(w)-\text{gammaln}(z+w)) \\ \text{betaln}(z,w) &= \text{gammaln}(z)+\text{gammaln}(w)-\text{gammaln}(z+w) \end{aligned}$$

```
>> beta{3,5}
ans =
0.01
>> exp(gammaln(3)+gammaln(5)-gammaln(3+5))
```

```
ans =
0.01
>> betaln(3,5)
```

```
ans =
-4 . 65
>> gammaln(3)+gammaln(5)-gammaln(3+5)
ans =
-4.65
```

También para los valores de  $z=3$  y  $w=5$  comprobaremos que:

```
beta(z,w)=Γ(z)Γ(w)/Γ(z+w)
>> gamma (3) *gamma (5) /gamma (3+5)
ans =
0.01
>> beta(3,5)
ans =
0.01
```

Como ejemplo adicional, supongamos que queremos calcular la longitud de un período completo de la sinusoide  $y=3\sin(2x)$ .

La longitud de esta curva vendrá dada por la fórmula;

$$4 \int_0^{\frac{1}{4}\pi} \sqrt{\left(1 - \left(\frac{\partial}{\partial x}y(x)\right)^2\right)} dt = \int_0^{\pi/2} \sqrt{1 + 36 \cdot \cos^2(2x)} dx = 2 \int_0^{\pi/2} \sqrt{37 - 36 \cdot \sin^2(t)} dt$$

En el último paso hemos hecho el cambio de variable  $2x=t$ , utilizando además que  $\cos^2(t)=1-\sin^2(t)$ . El valor de la integral puede calcularse ahora mediante;

```
>> [K,E]=ellipke (36/37)
K =
3.20677433446297
E =
1.03666851510702
>> 2*sqrt(37)*E
ans =
12.61161680006573
```

## INTEGRAL DEFINIDA Y APLICACIONES. VARIAS VARIABLES

Las aplicaciones de las integrales definidas en varias variables ocupan un lugar muy importante en el cálculo integral y en el análisis matemático en general. En los párrafos siguientes analizaremos el trabajo con MATLAB para el cálculo de áreas de figuras planas, superficies y volúmenes a partir de la integración múltiple.

## ÁREA DE FIGURAS PLANAS Y DOBLE INTEGRACIÓN

Si consideramos un recinto  $S$ , podemos hallar su área mediante el uso de integrales dobles. Si el recinto  $S$  está determinado por curvas cuyas ecuaciones vienen dadas en coordenadas cartesianas, su área  $A$  se halla mediante la fórmula:

$$A = \iint_S dx dy$$

Si, por ejemplo,  $S$  está determinado por  $a < x < b$  y  $f(x) < y < g(x)$ , el área valdrá:

$$A = \int_a^b dx \int_{f(x)}^{g(x)} dy$$

Si  $S$  está determinado por  $h(a) < x < k(b)$  y  $c < y < d$ , el área valdrá:

$$A = \int_c^d dy \int_{h(a)}^{k(b)} dx$$

Si el recinto  $S$  está determinado por curvas cuyas ecuaciones vienen dadas en coordenadas polares con radio vector  $r$  y ángulo  $\theta$ , su área  $A$  se hallará mediante la fórmula:

$$A = \iint_S r dr d\theta$$

Si, por ejemplo,  $S$  está determinado por  $s < \theta < t$  y  $f(\theta) < r < g(\theta)$ , el área valdrá:

$$A = \int_a^t d\theta \int_{f(\theta)}^{g(\theta)} r dr$$

Como primer ejemplo calculamos el área de la figura situada sobre el eje OX, limitada por este eje, la parábola  $y^2 = 4x$  y la recta  $x+y = 3$ .

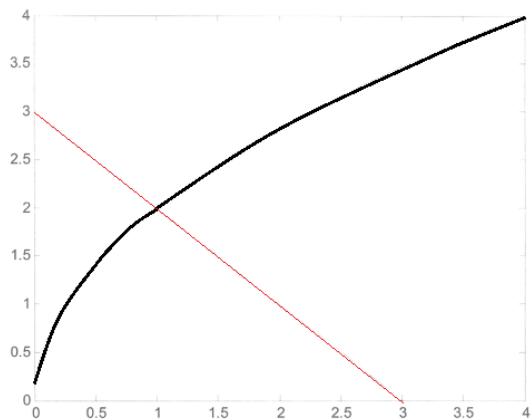
Es conveniente comenzar con la representación gráfica del recinto de integración.

Vemos que el recinto se puede limitar para  $y$  entre 0 y 2 ( $0 < y < 2$ ) y para  $x$  entre las curvas  $x = y^2/4$  y  $x = 3 - y$ . Luego, se puede calcular el área pedida de la siguiente forma:

```
>> A=int(int('1','x','y^2/4','3-y'),'y',0,2)
```

```
A =
```

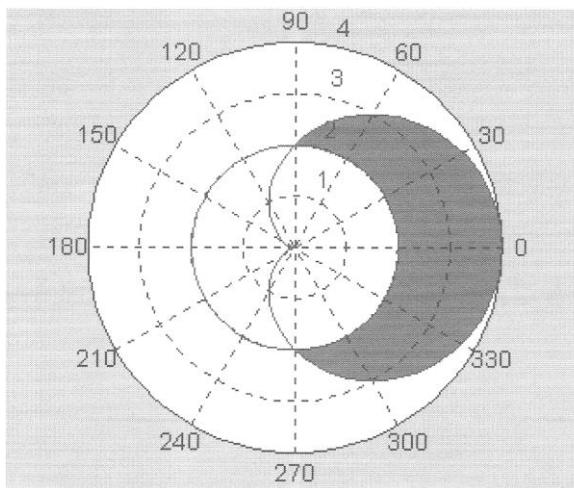
```
10/3
```



Como segundo ejemplo calculamos el área exterior al círculo de ecuación polar  $r = 2$ , e interior a la cardioide de ecuación polar  $r = 2(1+\cos(a))$ .

En primer lugar, representamos el área de integración.

```
>> a=0:0.1:2*pi;
>> r=2*(1+cos(a));
>> polar(a,r)
>> hold on;
>> r=2*ones(size(a));
>> polar(a,r)
```



Observando la gráfica, vemos que, por su simetría, podemos calcular la mitad del área pedida haciendo variar  $a$  entre  $0$  y  $\pi/2$  ( $0 < a < \pi/2$ ) y  $r$  entre las curvas  $r=2$  y  $r=2(1+\cos(a))$  ( $2 < r < 2(1+\cos(a))$ ):

```
>> pretty(int(int('r', 'r', 2, '2*(1 + cos(a))'), 'a', 0, pi/2))

pi
-- + 4
2
```

El área pedida será  $2(\pi/2+4) = \pi+8$  unidades cuadradas.

## Área de superficies por doble integración

El área  $S$  de una superficie curva expresada como  $z=f(x,y)$ , que tiene por proyección en el plano OXY un recinto  $R$ , es;

$$S = \iint \sqrt{1 + \left(\frac{\partial z}{\partial x}\right)^2 + \left(\frac{\partial z}{\partial y}\right)^2} dx dy$$

El área  $S$  de una superficie curva expresada como  $x=f(y,z)$ , que tiene por proyección en el plano OYZ un recinto  $R$ , es:

$$S = \iint \sqrt{1 + \left(\frac{\partial x}{\partial y}\right)^2 + \left(\frac{\partial x}{\partial z}\right)^2} dy dz$$

El área  $S$  de una superficie curva expresada como  $y=f(x,z)$ , que tiene por proyección en el plano OXZ un recinto  $R$ , es:

$$S = \iint \sqrt{1 + \left(\frac{\partial y}{\partial x}\right)^2 + \left(\frac{\partial y}{\partial z}\right)^2} dx dz$$

Como ejemplo calculamos el área de la superficie del cono de ecuación  $x^2 + y^2 = a z^2$ , limitada por encima del plano OXY y cortada por el cilindro  $x^2 + y^2 = by$ .

La proyección en el plano OXY es el recinto formado por el círculo de ecuación  $x^2 + y^2 = by$ , luego, el cálculo con MATLAB será como sigue:

```
>> clear all
>> syms x y z a b
>> z=sqrt((x^2+y^2)/a)
z =
```

```
((x^2 + y^2)/a)^(1/2)
>> pretty(int(int((1+diff(z,x)^2+diff(z,y)^2)^(1/2),x,-(b*y-y^2)^(1/2),
(b*y-y-2)^(1/2)),y,0,b))
      2/ 1      \ 1/2
pi b | - + 1 |
\ a      /
-----
4
```

### Cálculo de volúmenes por integrales dobles

El volumen  $V$  de un cilindroide limitado en su parte superior por la superficie de ecuación  $z=f(x,y)$ , en su parte inferior por el plano OXY y lateralmente por la superficie cilíndrica recta que corta al plano OXY, generando un recinto R, es:

$$V = \iint_R f(x, y) dx dy = \iint_R z dx dy$$

El volumen  $V$  de un cilindroide limitado en su parte superior por la superficie de ecuación  $x=f(y,z)$ , en su parte inferior por el plano OYZ y lateralmente por la superficie cilíndrica recta que corta al plano OYZ, generando un recinto R, es:

$$V = \iint_R f(y, z) dy dz = \iint_R x dy dz$$

El volumen  $V$  de un cilindroide limitado en su parte superior por la superficie  $y=f(x,z)$ ; en su parte inferior por el plano OXZ y lateralmente por la superficie cilíndrica recta que corta al plano OXZ, generando un recinto R, es:

$$V = \iint_R f(x, z) dx dz = \iint_R y dx dz$$

Como primer ejemplo calculamos el volumen en el primer octante comprendido entre el plano OXY, el plano  $z=x+y+2$  y el cilindro  $x^2 + y^2 = 16$

El volumen pedido se halla mediante la integral:

```
>> pretty(simple(int(int('x+y+2','y',0,'sqrt(16-x^2)'), 'x', 0, 4)))
      128
8 pi + ---
3
```

Como segundo ejemplo calculamos el volumen limitado por el paraboloide  $x^2 + 4y^2 = z$  y lateralmente por los cilindros de ecuaciones  $y^2 = x$  y  $x^2 = y$ .

El volumen se calculará mediante la integral siguiente:

```
>> pretty(int(int('x^2+4*y^2', 'y', 'x^2', 'sqrt(x)'), 'x', 0, 1))
```

$\frac{3}{7}$

### Cálculo de volúmenes e integrales triples

El volumen de un recinto R tridimensional, cuyas ecuaciones están expresadas en coordenadas cartesianas, viene dado por la integral triple:

$$\iiint_R dxdydz$$

El volumen de un recinto R tridimensional, cuyas ecuaciones están expresadas en coordenadas cilíndricas, viene dado por la integral triple:

$$\iiint_R rdzdrda$$

El volumen de un recinto R tridimensional, cuyas ecuaciones están expresadas en coordenadas esféricas, viene dado por la integral triple:

$$\iiint_R r^2 \sin(\theta) dr d\theta d\alpha$$

Como primer ejemplo calculamos el volumen limitado por el paraboloide a  $x^2 + y^2 = z$ , y el cilindro de ecuación  $z = a^2 - y^2$  para  $a > 0$ .

El volumen será cuatro veces la integral que se calcula como sigue:

```
>> a = sym('a', 'positive')

a =
a

>> pretty(simple(vpa(int(int(int('1', 'z', 'a*x^2+y^2', 'a^2-y^2'), 'y', 0,
'sqrt((a^2-a*x^2)/2)'), 'x', 0, 'sqrt(a)))))

7
-
2
```

0.27768018363489789043849256187879 a

O lo que es lo mismo:

```
>>V=4*(simple(int(int(int('1','z','a*x^2+y^2','a^2-y^2'),'y',0,
'sqrt((a^2-a*x^2)/2)'),'x',0,'sqrt(a'))))

V =
(pi*(2*a^7)^(1/2))/4
```

Como segundo ejemplo calculamos el volumen limitado por los cilindros de ecuaciones  $z = x^2$  y  $4 - y^2 = z$ .

Si resolvemos el sistema formado por las ecuaciones de las dos superficies nos hacemos una idea de sus puntos de corte:

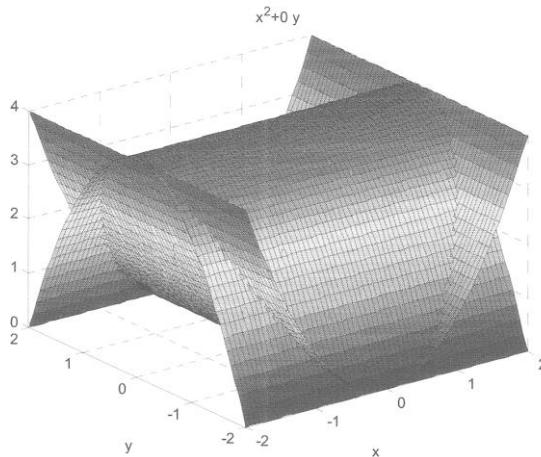
```
>> [x,y]=solve('x^2+0*y', '4-y^2+0*x')

x =
0
0

y =
2
-2
```

Ahora realizamos la representación gráfica del volumen a calcular.

```
>> ezsurf('4-y^2+0*x', [-2,2], [-2,2])
>> hold on;
>> ezsurf('x^2+0*y', [-2,2], [-2,2])
```



Por último calculamos el volumen pedido multiplicando por 4 una zona de volumen positivo.

```
>>V=4*int(int(int('1','z','x'^2,'4-y^2'),'y',0,'sqrt(4-x^2)'),  
'x',0,2)  
  
V =  
  
8*pi
```

## EL TEOREMA DE GREEN

El teorema de Green permite el cálculo de integrales en la frontera de regiones cerradas. Sea  $C$  una curva plana simple cerrada a trozos, y sea  $R$  la región consistente en  $C$  y su interior. Si  $f$  y  $g$  son funciones continuas con primeras derivadas parciales continuas en un abierto  $D$  que contiene a  $R$ , entonces:

$$\int_C m(x, y) dx + n(x, y) dy = \iint_R \left( \frac{\partial n}{\partial x} - \frac{\partial m}{\partial y} \right) dA$$

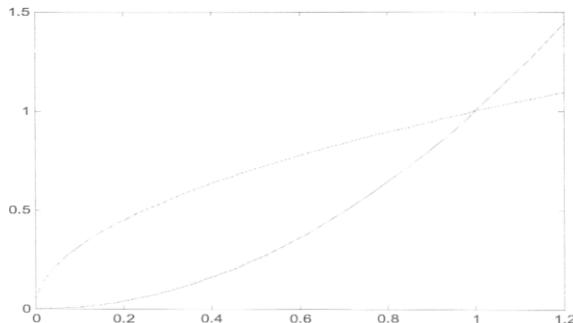
Como ejemplo calculamos, usando el teorema de Green, la integral:

$$\int_C (x + e^{iy}) dx + (2y + \operatorname{Cos}(x)) dy$$

donde  $C$  es la frontera de la región cerrada por las parábolas  $y=x^2$ ,  $x=y^2$ .

Las dos parábolas se interceptan en los puntos  $(0,0)$  y  $(1,1)$ . Al representarlas, se ven los límites de integración:

```
>> clear all  
>> fplot('[x^2,sqrt(x)] [0,1.2]')
```



Ahora podemos calcular la integral de la siguiente forma:

```
>> syms x y  
>> m=x+exp(sqrt(y))  
  
m =  
  
x + exp(y^(1/2))  
  
>> n=2*y+cos(x)  
  
n =  
  
2*y + cos(x)  
  
>> I=vpa(int(int(diff(n,x)-diff(m,y),y,x^2,sqrt(x)), x,0,1))  
  
I =  
  
-0.67644120041679251512532326651204
```

## TEOREMA DE LA DIVERGENCIA

El cálculo integral puede utilizarse para hallar el flujo de salida de un campo vectorial a través de una superficie. Sea  $Q$  un dominio con la propiedad de que cada linea recta que pasa por un punto interior al dominio corta a su frontera en exactamente dos puntos. Además, la frontera  $S$  del dominio  $Q$  es una superficie orientada cerrada a trozos con vector normal unitario exterior  $n$ . Si  $f$  es un campo vectorial que tiene derivadas parciales continuas en  $Q$ , entonces:

$$\iint_S f \cdot n \, dS = \iiint_Q \operatorname{Div}(f) \, dV$$

El primer miembro de la igualdad anterior se denomina flujo de salida del campo vectorial a través de la superficie  $S$ .

Como ejemplo utilizamos el teorema de la divergencia para hallar el flujo de salida del campo vectorial  $f = (xy+x^2 y z, yz+xy^2 z, xz+xyz^2)$  a través de la superficie del cubo delimitado en el primer octante por los planos  $x=2$ ,  $y=2$  y  $z=2$ .

```
>> clear all  
>> syms x y z  
>>  
M=x*y+x^2*z*y  
  
M =  
  
y*z*x^2 + y*x  
  
>> N=y*z+x*y^2*z  
  
N =
```

```

k*z*y^2 + z*y
>> P=x*z+x*y*z^2
P =
x*y*z^2 + x*z
>> Div=simple(diff(M,x)+diff(N,y)+diff(P,z))
Div =
x + y + z + 6*x*y*z
>> I=int(int(int(Div,x,0,2),y,0,2),z,0,2)
I =
72

```

## EL TEOREMA DE STOKES

El teorema de Stokes facilita el cálculo de integrales en línea. Sea  $S$  una superficie orientada de área finita definida por una función  $f(x,y)$ , con normal unitaria  $n$  y frontera  $C$ . Sea  $F$  un campo vectorial continuo definido en  $S$ , tal que las funciones componentes de  $F$  tienen derivadas parciales continuas en cada punto no frontera de  $S$ . Entonces se cumple:

$$\int_C F \cdot dr = \iint_S (\text{rot } F) \cdot n \, ds$$

Como ejemplo utilizamos el teorema de Stokes para evaluar la integral de línea:

$$\int_C -y^3 dx + x^3 dy - z^3 dz$$

donde  $C$  es la intersección del cilindro  $x^2 + y^2 = 1$  y el plano  $x+y+z = 1$ , y la orientación de  $C$  corresponde al movimiento en sentido contrario al que giran las manecillas del reloj en el plano OXY.

La curva  $C$  acota a la superficie  $S$  definida por  $z=1-x-y=f(x,y)$  para  $(x,y)$  en el dominio  $D=\{(x,y)/x^2+y^2=1\}$ .

Hacemos  $F = -y^3 i + x^3 j - z^3 k$ .

Ahora, calculamos el rotacional de  $F$  y se integra en la superficie  $S$ .

```
>> F=[-y^3,x^3,z^3]
```

```

F =
[-y^3,x^3,z^3]
>> clear all
>> syms x y z
>> M=-y^3

M =
-y^3
>> N=x^3

N =
x^3
>> P^z^3

P =
z^3

>> rotF=simple([diff(P,y)-diff(N,z),diff(P,x)-diff(M,z),diff(N,x)-diff(M,y)])
rotF =
[ 0, 0, 3*x^2 + 3*y^2]

```

Por lo tanto, hemos de calcular la integral  $\int_D (3x^2 + 3y^2) dx dy$ . Cambiando a coordenadas polares, esta integral se calcula de la forma:

```

>> pretty (simple (int(int('3*r^3','a',0,2*pi),'r',0,1)))
3 pi
-----
2

```

*Ejercicio 1. Calcular las integrales siguientes:*

$$\int \sec(x) \csc(x) dx, \int x \cos(x) dx, \int \arccos(2x) dx$$

```

>> pretty(simple(int('sec(x)*csc(x'))))
log(tan(x))
>> pretty(simple(int('x*cos(x'))))

cos(x) + x sin(x)
>> pretty(simple(int('acos(2*x'))))
2 1/2

```

$$\frac{x \cos(2x) - \frac{(1-4x)}{2}}{2}$$

*Ejercicio 2. Resolver las integrales siguientes:*

$$\int \frac{\sqrt{9-4x^2}}{x} dx, \int x^8(1+5x^3)^{\frac{1}{4}} dx$$

```
>> pretty(simple(Int('(9-4*x^2)(1/2)/x')))

      /      /1   \1/2   \
      |  3 | -- |       |
      |      | 2 |       |
      |      \ x /       |
3 acosh | - ----- | + (9 - 4 x )^2 1/2
      \        2         /
-----
```

```
>> pretty(simple(int('x^8*(3+5*x^3)^(1/4)')))

      5
      -
      4
3 (5 x + 3)  (375 x^6 - 200 x^3 + 96)
-----
```

*Ejercicio 3. Resolver las integrales:  $a = \int_0^\infty x^3 e^{-x} dx$  y  $b = \int_0^\infty x^2 e^{-x^3} dx$*

Se tiene que  $a = \Gamma(4)$ .

Por otra parte, para el cambio de variable  $x^3 = t \rightarrow b = \int_0^\infty x^2 e^{-x^3} dx = \Gamma(1)/3$ .

Por lo tanto, para el cálculo de  $a$  y  $b$  usaremos la siguiente sintaxis MATLAB:

```
>> a=gamma(4)
a =
6
>> b=(1/3)*gamma(1)
b =
0.3333
```

*Ejercicio 4. Resolver las siguientes integrales:*

$$\int_0^{\infty} \frac{x^3}{e^x} dx, \int_0^3 \frac{x^3}{e^x} dx$$

Como  $\Gamma(p) = \int_0^{\infty} x^{p-1} e^{-x} dx$ , la primera integral se resuelve de la forma:

```
>> gamma(4) 0
ans =
```

6

Como  $\Gamma(x, p) = \frac{1}{\Gamma(p)} \int_0^{\infty} t^{p-1} e^{-t} dt = gammaintc(x, p)$ , la segunda integral se resuelve de la forma siguiente:

se resuelve de la forma siguiente:

```
>> gamma(4)*gammaintc(5, 4)
ans =
```

4.4098

*Ejercicio 5. Resolver las integrales siguientes:*

$$\int_0^1 x^4 (1-x)^3 dx, \int_0^5 \frac{x^4}{(1-x)^3} dx, \int_0^8 \frac{\sqrt[4]{2-3\sqrt{x}}}{\sqrt{x}} dx$$

$$\beta(p,q) = \int_0^{\infty} x^{p-1} (1-x)^{q-1} dx \text{ y } betainc(z,w) = \frac{1}{\beta(p,q)} \int_0^1 t^{z-1} (1-t)^{w-1} dt$$

con lo que la primera integral se resuelve como:

```
>> beta(5, 4)*betainc(1/2, 5, 4)
```

```
ans =
```

0.0013

Para la segunda integral hacemos el cambio de variable  $x^{1/3} = 2t$ , con lo que la integral se convierte en la siguiente:

$$6 \int_0^1 \frac{\sqrt{t}(1-t)^{\frac{1}{4}}}{2^{\frac{4}{3}}} dt$$

cuyo valor se calcula mediante la expresión MATLAB:

```
>> 6*2^(3/4)*beta(3/2,5/4)
ans =
```

5.0397

*Ejercicio 6. Hallar el valor de las integrales siguientes:*

$$\int_3^\infty \frac{1}{\sqrt{6x^3 - 37x^2 + 37x - 45}} dx \quad \text{y} \quad \int_0^1 \frac{x^2 + 1}{\sqrt{x^4 - 5x^2 + 4}} dx$$

Las primitivas correspondientes a integrales elípticas son de difícil cálculo algebraico. El hecho de estandarizar su resolución permite hallar el valor de múltiples tipos de integrales, que por cambio de variable pueden reducirse a integrales elípticas. En los ejemplos veremos varios ejercicios de este tipo. MATLAB habilita funciones simbólicas que calculan los valores de las integrales elípticas de primer, segundo y tercer orden.

$$k(m) = \int_0^1 \sqrt{\frac{1-t^2}{1-mt^2}} dt = \int_0^{\pi/2} \frac{1}{\sqrt{1-msin^2(\Phi)}} d\Phi = F(\pi/2, m)$$

$$e(m) = \int_0^1 \sqrt{\frac{1-t^2}{1-mt^2}} dt = \int_0^{\pi/2} \sqrt{1-msin^2(\Phi)} d\Phi = E(\pi/2, m)$$

La función  $[K, E] = ellipke(m)$  calcula las dos integrales anteriores.

Para la primera integral (irracional) del problema, como el polinomio subradical es de tercer grado, se hace el cambio  $x = a +$  siendo  $a$  una de las raíces del polinomio subradical. Tomamos la raíz  $x = 3$  y hacemos el cambio  $x = 3 + t^2$  con lo que obtenemos la integral:

$$\frac{1}{3}\sqrt{6} \int_0^\infty \frac{1}{(t^2 + 4/3)(t^2 + 3/2)} dt$$

Hagamos ahora el cambio  $t=(2\sqrt{3}) \tan u$ , con lo que la integral se transforma en la elíptica completa de primer orden:

$$2 \int_0^{\frac{1}{2}\pi} \frac{1}{\sqrt{9 - \sin^2(u)}} du = \frac{2}{3} \int_0^{\frac{1}{2}\pi} \frac{1}{\sqrt{9 - \frac{1}{9}\sin^2(u)}} du$$

cuyo valor se calcula mediante la expresión:

```
>> (2/3) * ellipke(1/9)  
ans =  
1.07825782374982
```

Para la segunda integral hacemos el cambio  $x = \sin t$  y tendremos:

$$5 \int_0^2 \frac{1}{\sqrt{4 - \sin^2(t)}} dt - \int_0^2 \sqrt{4 - \sin^2(t)} dt = \frac{5}{2} \int_0^2 \frac{1}{\sqrt{4 - \frac{1}{4}\sin^2(t)}} dt - 2 \int_0^2 \sqrt{1 - \frac{1}{4}\sin^2(t)} dt =$$

Hemos reducido el problema a dos integrales elípticas, la primera completa de primera especie y la segunda completa de segunda especie, cuyo valor puede calcularse mediante la expresión:

```
>> [K,E]=ellipke(1/4)
```

```
K =  
1.68575035481260  
E =  
1.46746220933943  
>> I=(5/2)*K-2*E  
I =  
1.27945146835264
```

**Ejercicio 7.** Calcular la longitud de un período completo de la sinusoide  $y=3\sin(2x)$ .

La longitud de esta curva vendrá dada por la fórmula:

$$\int_0^{\frac{1}{2}\pi} \sqrt{1 - \left(\frac{\partial y}{\partial x} y(x)\right)^2} dt = \int_0^{\frac{\pi}{2}} \sqrt{1 + 36\cos^2(2x)} dx = \int_0^{\frac{\pi}{2}} \sqrt{37 + 36\sin^2(t)} dt$$

En el último paso hemos hecho el cambio de variable  $2x=t$ , utilizando además que  $\cos^2(t)=1-\sin^2(t)$ . El valor de la integral puede calcularse ahora mediante:

```
>> [K,E]=ellipke(36/37)
```

K =

3.20677433446297

E =

1.03666851510702

```
>> 2*sqrt(37)*E
```

ans =

12.61161680006573

*Ejercicio 8. Calcular la integral  $\int_{-2}^{\infty} 3 \frac{e^{-2t}}{t} dt$*

Se trata de una integral de tipo exponencial. Realizamos el cambio de variable  $v=2t$  y obtenemos la integral equivalente:

$$3 \int_{-4}^{\infty} \frac{e^{-t}}{t} dt$$

que se resuelve mediante la expresión MATLAB:

```
>> 3*expint(-4)
```

ans =

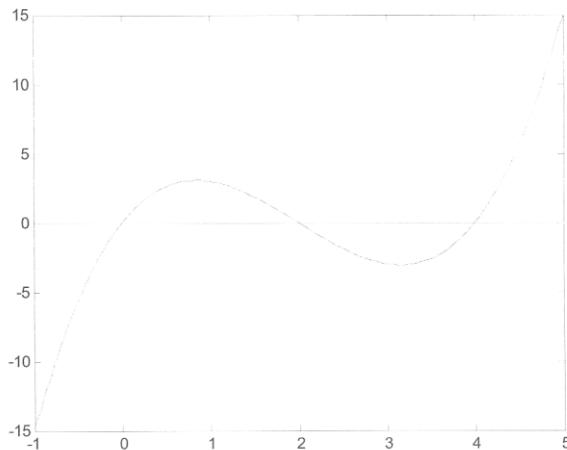
-58.89262341016866 - 9.42477796076938i

Suele utilizarse solamente la parte real del resultado anterior.

*Ejercicio 9. Calcular el área limitada por la curva  $y = x^3 - 6x^2 + 8x$  y el eje OX*

Comenzamos representando el área gráficamente.

```
>> fplot('x^3-6*x^2+8*x', [-1, 5])
```



Los puntos de corte de la curva con el eje OX son  $(0,0)$ ,  $(0,2)$  y  $(0,4)$ , tal y como se observa al resolver la ecuación:

```
>> solve ('x^3-6*x^2 + 8*x')
ans =
0
2
4
```

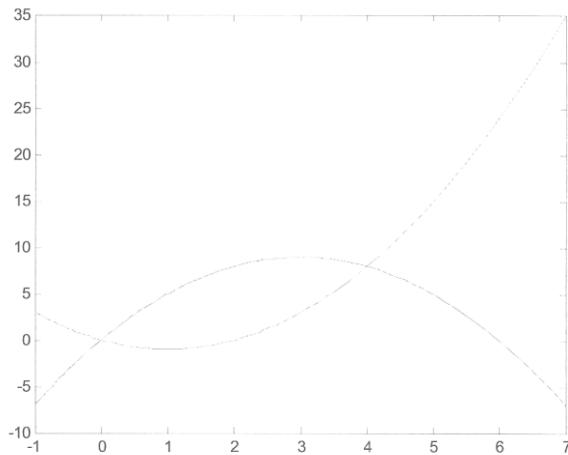
Como hay una región positiva y otra negativa, el área se calculará como sigue:

```
>> A=abs(int(x^3-6*x^2+8*x, 0, 2)) +abs(int(x^3-6*x^2+8*x, 2, 4))
A =
8
```

*Ejercicio 10. Calcular el área comprendida entre las parábolas  $y = 6x - x^2$  e  $y = x^2 - 2x$*

Comenzamos representando el área gráficamente.

```
>> fplot ('[6*x-x^2, x^2-2*x]', [-1, 7])
```



Buscamos los puntos de intersección de las curvas:

```
>> [x,y]=solve('y=6*x-x^2,y=x^2-2*x')
```

x =

0  
4

y =  
0  
8

Los puntos de corte son (0,0) y (4,8).

El área se calculará como sigue:

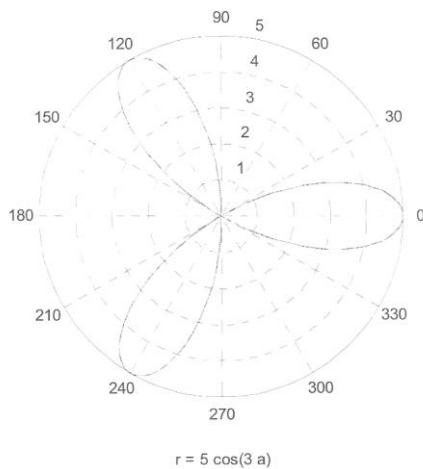
```
>> A=abs(int('(6*x-x^2)-(x^2-2*x)',0,4))
```

A =  
64/3

*Ejercicio 11. Calcular el área encerrada por la rosa de tres pétalos:  $r = 5 \cos(3\alpha)$*

Comenzamos representando la curva polar.

```
>> ezpolar('5*cos(3*a)')
```



El área total puede calcularse multiplicando por 6 el área entre 0 y  $\pi/6$  (medio pétalo). Tendremos entonces:

```
>> A=6*int((1/2)*5-'2*cos(3*a)^2,0,pi/6)
```

A =

$$(25\pi)/4$$

*Ejercicio 12. Calcular el área comprendida entre el eje OX y un arco de la cicloide de ecuaciones paramétricas:*

$$\begin{aligned}x &= 5(t - \operatorname{Sen}(t)) \\y &= 5(1 - \operatorname{Cos}(t))\end{aligned}$$

El arco completo de la cicloide comprende la variación de t entre 0 y  $2\pi$ . El área pedida vendrá dada por la integral:

```
>> A=int(5*(1-cos(t))*diff(5*(t-sin(t))),0,2*pi)
```

$$\begin{aligned}A &= \\&75\pi\end{aligned}$$

*Ejercicio 13. Calcular el volumen engendrado por la función  $y=\operatorname{Sen}(x)$  al girar alrededor del eje OX y alrededor del eje OY en el intervalo  $[0,\pi]$ .*

```
>> Vox=pi*int('sin(x)^2',0,pi)
```

```
Vox =
pi^2/2
>> Voy=2*pi*int('x*sin(x)',0,pi)
Voy =
2*pi^2
```

*Ejercicio 14. Calcular el volumen engendrado por revolución alrededor del eje OY de un arco de la cicloide de ecuaciones paramétricas:*

$$\begin{aligned}x &= (t - \operatorname{Sen}(t)) \\y &= (1 - \operatorname{Cos}(t))\end{aligned}$$

El arco completo de la cicloide comprende la variación de  $t$  entre 0 y  $2\pi$ . El volumen pedido vendrá dada por la integral:

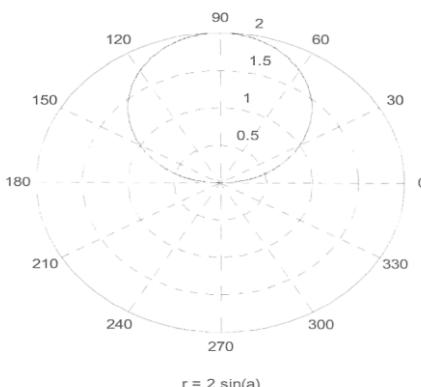
```
>> V=2*pi*int('(t-sin(t))*(1-cos(t))*(1-cos(t))',0,2*pi)
V =
6*pi^3
```

*Ejercicio 15. Calcular el volumen engendrado por revolución alrededor de su eje polar de la curva:*

$$r = 2 * \sin(a)$$

Representamos la curva polar:

```
>> ezpolar('2*sin(a)')
```



Observamos que la integral se extenderá a la variación del ángulo entre 0 y  $\pi$ . Para eliminar problemas de cálculo se puede considerar dos veces la variación del ángulo entre 0 y  $7\pi/2$ .

El volumen se calculará entonces mediante la siguiente integral:

```
>> V=2*2*pi/3*int((2*sin(a))^3*sin(a),0,pi/2)
V =
2*pi^2
```

*Ejercicio 16. Calcular la longitud del arco de curva:*

$$y = 2x\sqrt{x}$$

Para  $x$  variando entre 0 y 2

La longitud se calcula directamente mediante la integral siguiente:

```
>> L=int((1+(diff((2*x*sqrt(x)))^2),0,2)
L =
20
```

*Ejercicio 17. Calcular la longitud de la astroide:*

$$x^{2/3} + y^{2/3} = 2^{2/3}$$

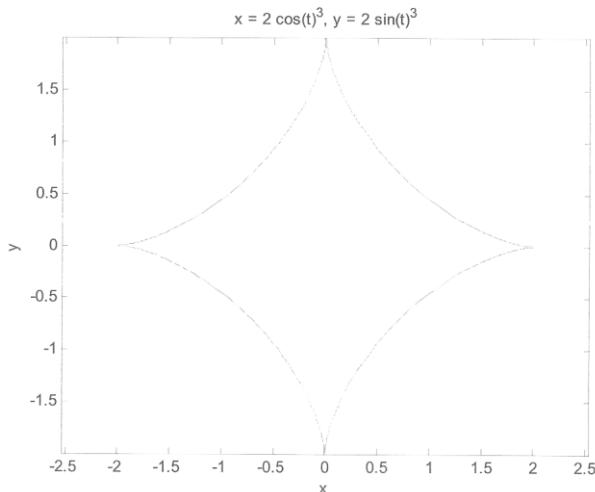
La curva se puede expresar en forma paramétrica como sigue:

$$\begin{aligned} x &= 2\cos(t)^3 \\ y &= 2\sin(t)^3 \end{aligned}$$

Ahora representamos el astroide para delimitar los límites de variación de la integral.

```
>> ezplot('2*cos(t)^3','2*sin(t)^3')
```

Observando la gráfica vemos que la integral puede hallarse multiplicando por 4 la variación entre 0 y  $\pi/2$



La longitud de la astroide se calculará mediante la integral siguiente:

```
>> L=4*int(((diff ('2*cos(t)^3'))^2+(diff('2*sin(t)^3'))^2)^(1/2),0,pi/2)
```

```
L =
12
```

*Ejercicio 18. Calcular la integral:*

$$\iint xe^{\frac{-x^2}{y}} dx dy$$

*Siendo S el recinto de plano limitado por  $x=0$ ,  $y=x^2$ ,  $y=1$  e  $y=2$ .*

El recinto S queda determinado por  $1 < y < 2$  y  $0 < x < y^{1/2}$

La integral se calculará de la forma siguiente:

```
>> int(int('x*exp(-x-'2/y)', 'x', 0, 'sqrt(y)'), 'y', 1, 2)

ans =
3/4 -3/4 (4*exp(1))
```

*Ejercicio 19. Calcular el volumen comprendido entre los dos cilindros:*

$$z=x^2 \text{ y } z=4-y^2.$$

El problema se resuelve mediante la integral:

```
>> 4*int(int(int(1,z,x^2, 4-y^2),y,0,sqrt(4-x^2)),x,0,2)
ans =
8* pi
```

*Ejercicio 20. Calcular la integral:*

$$\iiint \frac{1}{(x+y+z+1)^3}$$

*en el recinto V comprendido entre los tres planos coordenados y el plano  $x+y+z=1$ .*

El recinto V queda determinado por:

$$0 < x < 1, 0 < y < 1-x, 0 < z < 1-x-y$$

La integral que resuelve el problema es la siguiente:

```
>> x=sym('x','positive')
x=
x
>> vpa(int(int(int(1/(x+y+z+1)^3, z,0,1-x-y),y,0,1-x),x,0,1))
Warning: Explicit integral could not be found.

ans =
0.034073590279972654708616060729088
```