MATLAB

MATLAB (abreviatura de *MATrix LABoratory*, "laboratorio de matrices") es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Está disponible para las plataformas Unix, Windows, Mac OS X y GNU/Linux .

Entre sus prestaciones básicas se hallan: la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MATLAB dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Además, se pueden ampliar las capacidades de MATLAB con las *cajas de herramientas* (*toolboxes*); y las de Simulink con los *paquetes de bloques* (*blocksets*).

Es un software muy usado en universidades y centros de investigación y desarrollo. En los últimos años ha aumentado el número de prestaciones, como la de programar directamente procesadores digitales de señal o crear código VHDL.

MATLAB ® es un lenguaje de alto nivel y un entorno interactivo para el cálculo numérico, visualización y programación. Usando MATLAB, puede analizar los datos, desarrollar algoritmos y crear modelos y aplicaciones. El lenguaje, las herramientas y funciones matemáticas integradas que permiten explorar múltiples enfoques y llegar a una solución más rápida que con hojas de cálculo o lenguajes de programación tradicionales, como C / C + + o Java ™.

Usted puede utilizar MATLAB para una gama de aplicaciones, incluyendo el procesamiento de señales y comunicaciones, procesamiento de imágenes y vídeo, sistemas de control, prueba y medida, finanzas computacionales, y la biología computacional. Más de un millón de ingenieros y científicos en la industria y el mundo académico usan MATLAB, el lenguaje del cálculo técnico.

Fue creado por *Cleve Moler* en 1984, surgiendo la primera versión con la idea de emplear paquetes de subrutinas escritas en Fortran en los cursos de álgebra lineal y análisis numérico, sin necesidad de escribir programas en dicho lenguaje. El lenguaje de programación M fue creado en 1970 para proporcionar un sencillo acceso al software de matrices *LINPACK* y *EISPACK* sin tener que usar Fortran.

En 2004, se estimaba que MATLAB era empleado por más de un millón de personas en ámbitos académicos y empresariales

- MATLAB distingue entre mayúsculas y minúsculas.
- La comilla ' es la que, en un teclado estándar, se encuentra en la tecla de la interrogación.
- Los comentarios deben ir precedidos por % o, lo que es lo mismo, MATLAB ignora todo lo que vaya precedido por el símbolo %.
- La ayuda de MATLAB es bastante útil; para acceder a la misma basta teclear help. Es recomendable usarlo para obtener una información más precisa sobre la sintaxis y diversas posiblidades de uso de los comandos.

Kelly Chazaret Del Angel Rivera

Para escribir un programa con MATLAB habrá que crear un fichero que tenga extensión `.m` y contenga las instrucciones. Esto se puede hacer con cualquier editor de textos, pero tiene algunas ventajas usar el editor propio de MATLAB llamado `medit`.

MATLAB trabaja con memoria dinámica, por lo que no es necesario declarar las variables que se van a usar. Por esta misma razón, habrá que tener especial cuidado y cerciorarse de que entre las variables del espacio de trabajo no hay ninguna que se llame igual que las de nuestro programa (proveniente, por ejemplo, de un programa previamente ejecutado en la misma sesión), porque esto podría provocar conflictos. A menudo, es conveniente reservar memoria para las variables (por ejemplo, si se van a utilizar matrices muy grandes); para ello, basta con asignarles cualquier valor. Del mismo modo, si se está usando mucha memoria, puede ser conveniente liberar parte de ella borrando (clear) variables que no se vayan a usar más.

Un programa escrito en MATLAB admite la mayoría de las estructuras de programación al uso y su sintaxis es bastante estándar. En los siguientes ejemplos se muestra la sintaxis de algunas de estas estructuras (if, for, while,...).

Ejemplo 1: Calcular la suma de los n primeros términos de la sucesión 1, 2x, 3x^2, 4x^3, ...

```
n=input('¿Cuántos términos quieres sumar?  ');
x=input('Dame el valor del numero x  ');
suma=1;
for i=2:n
   suma=suma+i*x^(i-1);
end
disp('El valor pedido es')
disp(suma)
```

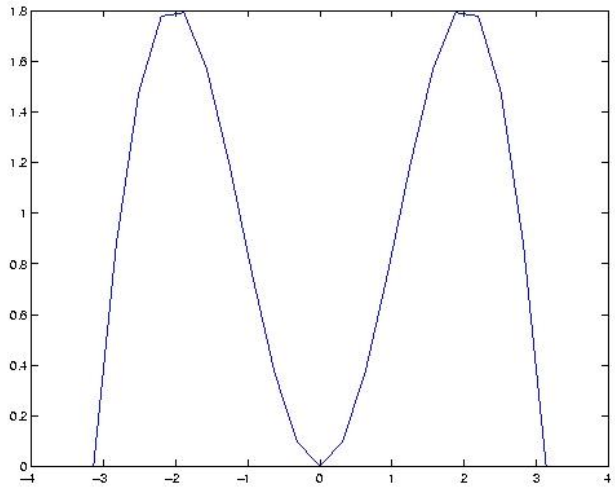
Ejemplo 2: Decidir si un número natural es primo.

```
n=input('Número natural que deseas saber si es primo ');
i=2;
primo=1;
while i<=sqrt(n)
 if rem(n,i)==0    % Resto de dividir n entre i
      primo=0;
   break
   end
   i=i+1;
end
```

MATLAB tiene un gran potencial de herramientas gráficas. Se pueden dibujar los valores de un vector frente a otro (de la misma longitud):

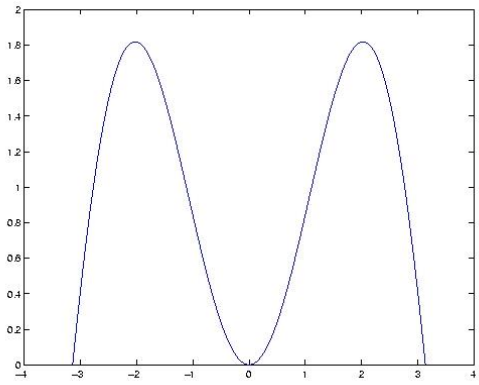
```
>>x=pi*(-1:0.1:1);
>>y=x.*sin(x);
>>plot(x,y)          % Por defecto une los puntos (x(i),y(i)) mediante una poligonal
```

Kelly Shazaret L... Rivera



Como se ve, con pocos puntos la gráfica tiene un aspecto demasiado lineal a trozos. Para "engañar" al ojo, basta tomar más puntos.

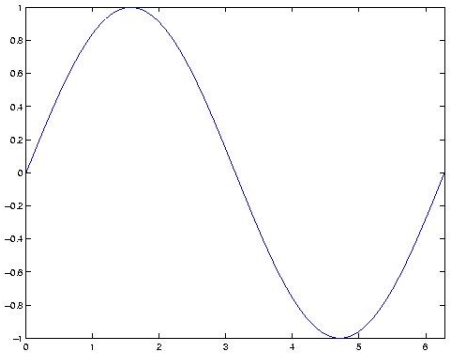
```
>>x=pi*(-1:0.01:1);
>>y=x.*sin(x);
>>plot(x,y)
```



También pueden dibujarse funciones. Así:

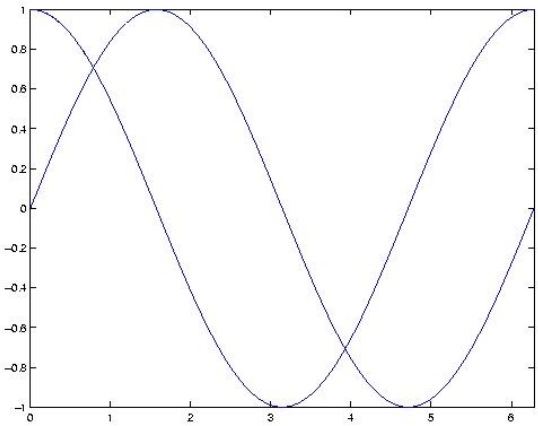
```
>>fplot('sin(x)',[0 2*pi])     % Dibuja la función seno en el intervalo [0,2*pi]
```

Del Angel Rivera



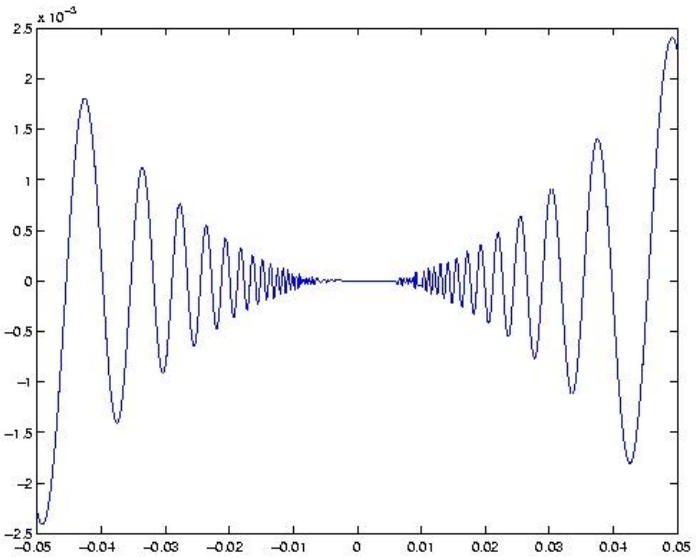
```
>>hold on                    % Mantiene en la ventana gráfica los dibujos anteriores

>>fplot('cos(x)',[0 2*pi])   % Dibuja sobre la gráfica anterior la función cos(x)
```



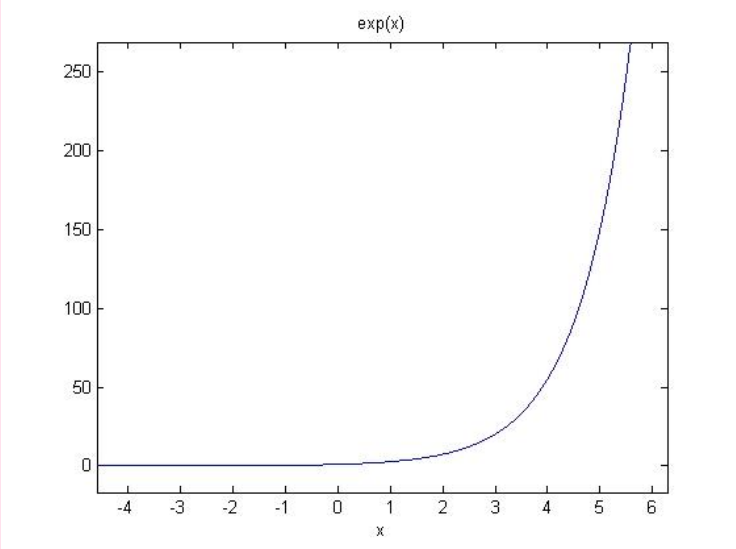
```
>>hold off                              % Con esto olvida los dibujos anteriores
                                        % y dibuja en una ventana nueva

>>fplot('x^2*sin(1/x)',[-0.05 0.05])   % Dibuja la función x^2*sin(1/x)
```



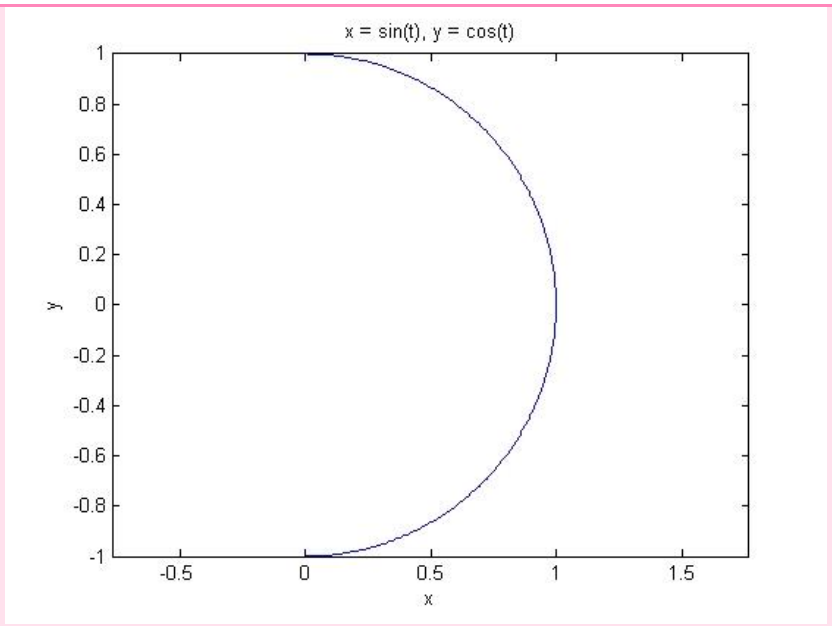
También puede usarse el versátil comando `ezplot` (se lee como *easy plot*) que permite dibujar funciones,

```
>>ezplot('exp(x)')      % Dibuja la función exponencial en un intervalo adecuado a la función
```



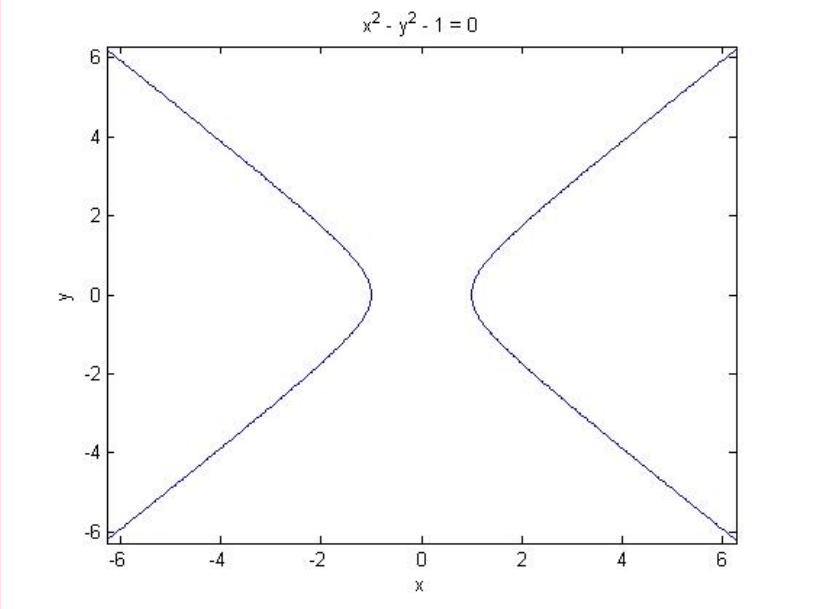
curvas en paramétricas,

```
>>ezplot('sin(t)','cos(t)',[0 pi])
```



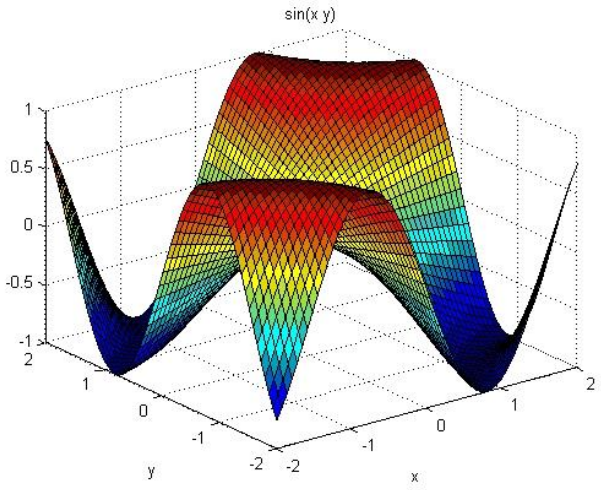
e implícitas

```
>>ezplot('x^2 - y^2 - 1')
```



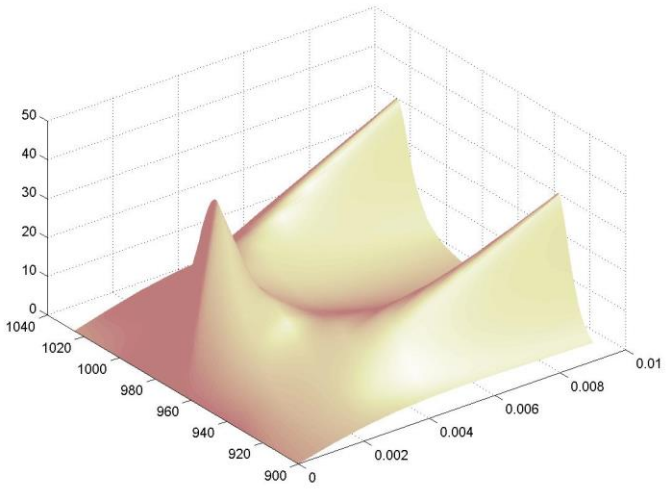
También permite dibujar superficies. La forma más sencilla es mediante el comando `ezsurf`,

```
>>ezsurf('sin(x*y)',[-2 2 -2 2])
```



aunque se pueden realizar gráficas más sofisticadas:

```
>>t=0:0.001:0.009;
>>v=900:1025;
>>[T V]=meshgrid(t,v);
>>aux1=16*pi^2*(T.^2).*((V-918).^2).*((V-1011).^2);
>>aux2=aux1+(2*V-1929).^2;
>>w=T./aux2;
>>z=35000000*w;
>>surfl(t,v,z);      % Este comando dibuja la superficie creada mediante las
>>shading interp;    % ordenes anteriores. Los siguientes sirven para modificar
>>colormap(pink);    % el dibujo obtenido
>>rotate3d;          % Sirve para girar la figura mediante el ratón
```



Kelly Sr[illegible]aret Del Angel Rivera