

Programación con Matlab

PRÁCTICA 1B: GRAFICANDO CON MATLAB

Prof. Guilmer González

Graficación 1D

Hacer gráficos en Matlab es muy sencillo. No necesitamos definir la ventana de despliegue ni el sistema coordenado a usar. Basta con utilizar la instrucción `plot` para obtener en pantalla una gráfica simple de datos.

Teclee en el entorno de Matlab las instrucciones:

```
>>x = [1 2 3 4 5 6 7 8 9 10];  
>>y = [2.1 2.1 3.2 3.5 4.1 6.2 5.2 6.3 7.1 4.3];  
>> plot(x,y)
```

Se observa en pantalla una ventana adicional donde son desplegados los datos en una gráfica. Como se puede apreciar, es una gráfica donde los puntos están unidos por un segmento de línea formando una especie de polígono. De igual manera, se observa una escala en x y en y en la que se grafica.

Intente ahora la instrucción

```
>>plot(x,y,'-o')
```

Ahora tiene un gráfico donde hemos indicado que se pinten los datos, además de unirlos con segmentos de línea, con una marca, la `o`. El comando `plot` permite modificadores de las opciones predeterminadas. Podemos cambiar el color y cambiar la forma de unir los puntos. Teclee en el entorno `help plot`, para observar distintas opciones.

Para identificar la gráfica y hacerla atractiva en reportes, muchas veces requerimos adicionar información de los datos representados. Para esto existen los comandos `xlabel`, `ylabel`, `title` con los cuales podemos añadir información adicional. Use las instrucciones

```
>> xlabel('Numero de observaciones ');
>> ylabel('Observaciones anuales ');
>> title('Grafica del comportamiento de los datos');
>> grid
```

El comando `grid` es muy útil, nos permite observar una regilla sobre el gráfico y con ello identificar mejor la posición de los datos.

Grafiquemos otra función, la función `sin(x)` sobre el intervalo $[0, 2\pi]$. Usemos 50 puntos igualmente espaciados entre 0 y π .

```
>> x=linspace(0,2*pi,50); y=sin(x);
>> plot(x,y)
>> title('Grafica de la funcion seno');
>> grid
```

Estamos usando una función de Matlab `linspace` la cual consta de 3 argumentos, que nos permite generar una colección de datos igualmente espaciados. Los dos primeros argumentos de `linspace` se refieren al extremo izquierdo y derecho del intervalo, el tercer argumento es para indicar el número de puntos a graficar.

Para cerrar las ventanas gráficas, use el comando `close all` o bien `close(4)` si 4 es el identificador o número de la ventana que quiere cerrar.

Debe tener mucho cuidado con interpretar los datos de una gráfica. Cierre la ventana o las ventanas y grafique:

```
>> y2 = sin(x)+0.5*sin(50*x);
>> plot(x,y2);
>> title('Grafica de la funcion seno con un ruido');
>> grid
```

Básicamente le estamos añadiendo un ruido a la función seno, pero este no se observa en la gráfica; vemos los puntos que corresponden a la función seno. Grafiquemos usando más puntos:

```
>> x=linspace(0,2*pi,150);
```

```
>> y2 = sin(x)+0.5*sin(50*x);
>> plot(x,y);
>> title('Grafica de la funcion seno con un ruido');
>> grid
```

La gráfica es distinta. Use la lupa del menú gráfico que aparece cerca del borde superior de la ventana para hacer un acercamiento y observe el ruido añadido.

Experimente un poco. Grafique las funciones

- 1) $f(x) = \sin(x) + \sin(2x) + \sin(3x)$, en $[0, 2\pi]$.
- 2) $f(x) = e^x \cos(x)$, en $[-2, 3]$.
- 3) $f(x) = 1/((x - .3)^2 + .01) + 1/((x - .9)^2 + .04)$ en $[0, 2]$.

Podemos combinar dos o mas datos a graficar en una misma instrucción `plot` de la siguiente manera:

```
>> x=2:.1:5;
>> f=1./x;
>> g=log(x-1);
>> plot(x,f,'-',x,g,'--');
>> legend('f(x)=1/x','g(x)=log(x-1)');
>> grid
```

Observe que hemos añadido una instrucción más `legend` que nos permite incorporar información de los datos graficados.

Graficación 2D

Las más de las veces estamos interesados en reproducir comportamientos a lo largo del tiempo, o simular flujos en las dos direcciones. En otras ocasiones nos interesa reconstruir superficies. Veamos cómo reproducir nuestros resultados si contamos con expresiones analíticas para la superficie.

Veamos algunos ejemplos típicos de Geometría Analítica para enfocar ideas de lo que podemos hacer en Matlab.

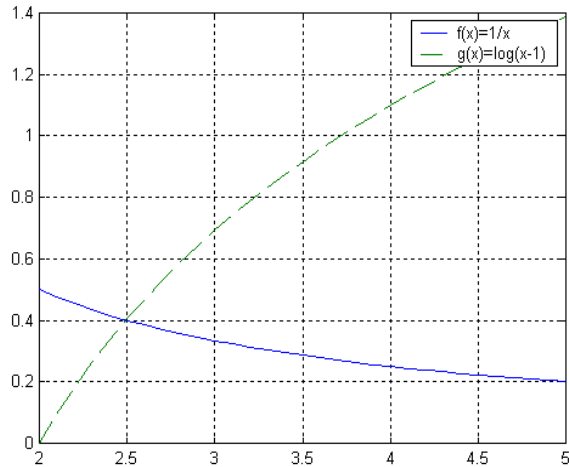


Figura 1: Dos gráficas.

Funciones cuadráticas en dos variables

Observe la expresión de una función en dos variables,

$$x^2 + 2y^2 + xy = 1$$

ecuación que representa una cónica. Esta curva puede representarse como una curva de nivel (o contorno de nivel) de $f(x, y) = x^2 + 2y^2 + xy$, cuando $f(x, y) = 1$. Es decir, una rebanada de la superficie $f(x, y)$.

Podemos graficar la curva de nivel para $z = 1$ usando la instrucción:

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

Usando Matlab, podemos observar esa curva, a través de los comandos `meshgrid` y `contour` en la forma

```
>>x = linspace(-6,6,150);
>>y = linspace(-6,6,150);
>>[X,Y]=meshgrid(x,y);
>>Z=X.^2+2*Y.^2+X.*Y;
>>contour(X,Y,Z,[1 1]);
```

Aquí aparecen dos instrucciones nuevas. Por una parte `mesgrid` nos permite llenar dos matrices `X` y `Y` como copias de los vectores `x` y `y`, y con esto, generamos sobre el cuadrado $[-6, 6] \times [-6, 6]$ una colección de nodos o puntos interiores sobre los cuales evaluamos la función.

Por otra parte, está la instrucción `contour` esta nos permite “rebanar” a la superficie $f(x, y)$ a la altura $z = 1$.

Experimente con distintos valores de corte; en el tercer parámetro de `contour` cambie a `[1 2 2.5 5 6]` y observe.

Graficación de superficies

Un ejercicio que trataremos en una de las prácticas será interpolar en 2D una función particular:

$$u(x, y) = \frac{\frac{5}{4} + \cos(5.4y)}{6 + 6(3x - 1)^2}.$$

```
>>nxy = 100;
>>xm = linspace(0,1,nxy);
>>ym = linspace(0,1,nxy);
>>[xm2d,ym2d] = meshgrid(xm,ym);
>>u=(5/4 + cos(5.4.*ym2d))./(6 + 6.*(3.*xm2d-1).^2);
>>mesh(xm2d,ym2d,u);
>>view(3);
```

el comando `view(3)` nos permite un punto desde el cual observar la superficie. Utilice la utilería para rotar gráficas que se encuentra en el menú gráfico de la ventana. La superficie la mostramos en la Figura 2.

El comando `mesh` no es el único para graficar superficies. Puede usar `surf(u)` en el ejemplo anterior para observar la superficie coloreada (cada celda está coloreada). Si utiliza el comando `surf(u)` obtendrá la anterior superficie con algunas curvas de nivel, experimente, para eso es Matlab: experimente, juegue y aprenda.

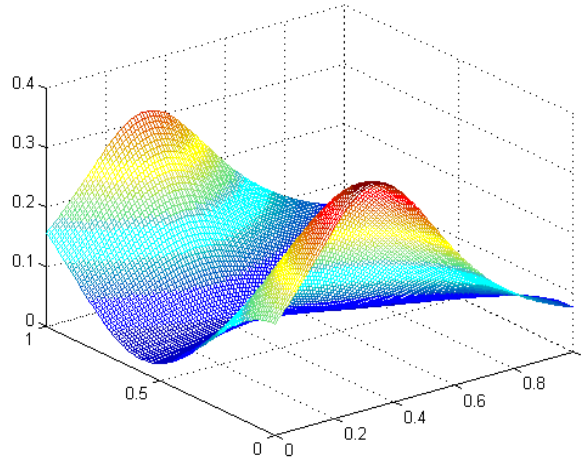


Figura 2: Una superficie simpática.

Ahora trate de graficar la expresión

$$\begin{aligned}
 u(x, y) = & 0.75 * \exp(-((9x - 2)^2 + (9 * x - 2)^2)/4.0) \\
 & + 0.75 * \exp(-(9 * x - 2). * (9 * x - 2)/49.0 - (9 * y - 2). * (9 * y - 2)/10.0) \\
 & + 0.5 * \exp(-((9 * x - 7). * (9 * x - 7) + (9 * y - 3). * (9 * y - 3))/4.0) \\
 & - 0.2 * \exp(-(9 * x - 4). * (9 * x - 4) - (9 * y - 7). * (9 * y - 7))
 \end{aligned}$$

definida de igual manera en $0 \leq x, y \leq 1$.

Forma cuadrática en 3 variables

Observemos algunas superficies en 3D, por ejemplo \mathcal{S} dada por

$$\mathcal{S} = \{(x, y, z) \mid x^2 + 2y^2 + 3z^2 + 4xy + 2xz + 6yz = 10\}$$

\mathcal{S} se puede ver como una superficie de una función cuadrática en tres variables

$$q(x, y, z) = x^2 + 2y^2 + 3z^2 + 4xy + 2xz + 6yz$$

para el nivel $q(x, y, z) = 1$. Usando Matlab, podemos observar su gráfica empleando las instrucciones `meshgrid` y `isosurface` en la forma:

```
>>x=linspace(-20,20,100);
>>y=linspace(-20,20,100);
>>z=linspace(-20,20,100);
>>[X,Y,Z]=meshgrid(x,y,z);
>>Q= X.^2 +2*(Y.^2) +3*(Z.^2) +4*(X.*Y) +2*(X.*Z) +6*(Y.*Z);
>>H=patch(isosurface(X,Y,Z,Q,11));
>>isonormals(X,Y,Z,Q, H);
>>set(H, 'FaceColor', 'red', 'EdgeColor', 'none');
>>daspect([1 1 1])
>>view(3)
>>camlight
>>lighting phong
>>axis equal
>>xlabel('x'); ylabel('y'); zlabel('z')
>>title ('Hiperboloide de una cara')
```

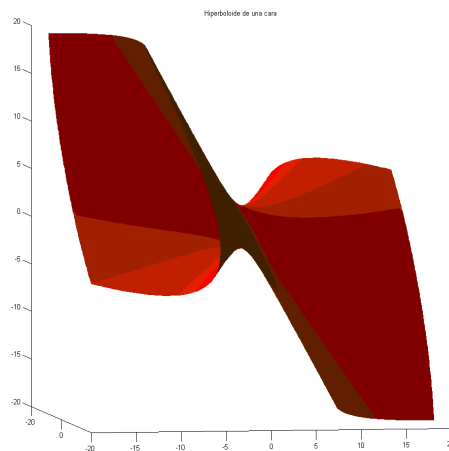


Figura 3: Un hiperboloide iluminado.

Aquí hemos utilizado más instrucciones para indicar a Matlab que hacer. Busque con el comando `help XXX` cada una de ellas, cambie algunos valores y observe.