Práctica 3

Gráficas.

3.1. Gráficas en el plano.

3.1.1. Polígonos y curvas planas.

Si deseamos dibujar en el plano un polígono con vértices (x_i, y_i) en Matlab, bastaría definir previamente los vectores $x = [x_1, x_2, \ldots, x_n]$, e $y = [y_1, y_2, \ldots, y_n]$, y utilizar el comando $plot(x, y)$.

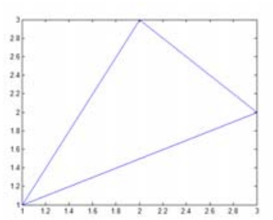
Nótese que, para que el polígono sea cerrado, el último vértice debe coincidir con el primero.

Ejemplo 3.1.1.–

Dibujar el triángulo de vértices $(1, 1), (3, 2), (2, 3), (1, 1)$:

$x = [1, 3, 2, 1];$
$y = [1, 2, 3, 1];$
$plot(x, y)$

El resultado será:



Si se desea hacer la gráfica de una función $f(x)$, en un intervalo $[a, b]$, se debe tener en cuenta que, el comando *plot*, dibuja poligonales, por lo que se deben definir dos vectores, x e y, de forma que en x tengamos representado el intervalo $[a, b]$, y en y, la imagen de f en dicho intervalo. Según realicemos dichas representaciones, más o menos finas serán las gráficas.

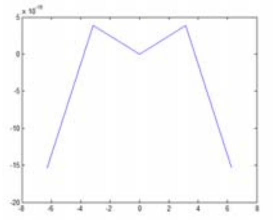
Ejemplo 3.1.2.– Queremos representar la función $f(x) = xsen(x)$ en el intervalo $[-2\pi, 2\pi]$.

Hagamos primero una representación basta (no deseable), para ello:

$x = -2 * pi : pi : 2 * pi;$
$y = x. * sin(x);$

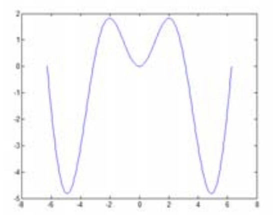
$plot(x, y)$

El resultado será:



Hagamos ahora, una representación más fina:

$x = -2 * pi : pi/100 : 2 * pi;$
$y = x. * sin(x);$
$plot(x, y)$



En la segunda gráfica, al haber hecho la partición del intervalo mucho más fina, parece que es una curva, aunque, en realidad, siguen siendo segmentos de rectas unidos.

Ejercicio 3.1.3.– Representar gráficamente las siguientes funciones:

1. $f(x) = \frac{cos(x)}{x}$, $x \in [\frac{pi}{4}, 3\frac{pi}{4}]$.
2. $f(x) = e^{x^2+1}$, $x \in [-5, 5]$
3. $f(x) = ln(x^2 + 1)$, $x \in [-5, 5]$

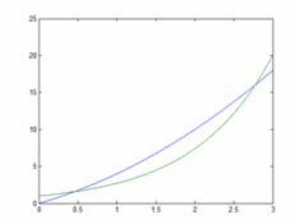
Es posible dibujar varias funciones en una misma gráfica, para ello, tenemos dos opciones:

- Usando únicamente el comando *plot*, evidentemente, todas las gráficas deben estar definidas en el mismo intervalo. Definiríamos las funciones $y1 = f(x)$, $y2 = g(x)$, ..., y escribiríamos $plot(x, y1, x, y2, \ldots)$.

  Ejemplo 3.1.4.– Dibujar en el intervalo $[0, 3]$ las funciones $f(x) = x^2 + 3x$ y $g(x) = e^x$.

  $x = 0 : 0{,}01 : 3;$
  $y = x.^2 + 3 * x;$
  $z = exp(x);$
  $plot(x, y, x, z)$

El resultado sería:



- Usando los comandos *hold on* y *hold off*. El comando *hold on* hace que la siguiente gráfica que deseemos dibujar, se realice sobre la última que tenemos activa, mientras que el comando *hold off*, nos devuelve a la situación previa. Si quisiésemos utilizar éste comando con el ejemplo anterior, haríamos:

  $x = 0 : 0{,}01 : 3;$
  $y = x.^2 + 3 * x;$
  $plot(x, y)$
  hold on
  $z = exp(x);$
  $plot(x, z)$

  El resultado es similar al anterior, la única diferencia es que, en esta segunda gráfica, ambas funciones se dibujan del mismo color. Con este comando, no es necesario que ambas funciones estén definidas en el mismo intervalo.

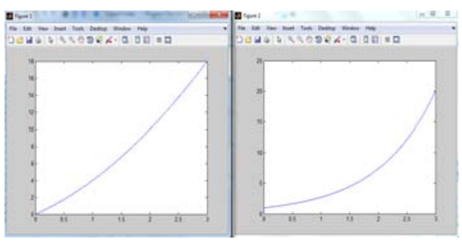
Ejercicio 3.1.5.– Representar en un mismo gráfico, las siguientes funciones:

1. $f(x) = sen(x)$, $g(x) = cos(x)$ y $h(x) = tan(x)$ en $[-2\pi, 2\pi]$
2.

Para tener distintas representaciones gráficas visibles simultáneamente, aunque en distintos gráficos, se usa el comando *figure*, dicho comando, abre una nueva ventana de gráfico para representar la siguiente función. Por defecto, sigue el orden numérico natural.

Ejemplo 3.1.6.– $x = 0 : 0{,}01 : 3;$
$y = x.^2 + 3 * x;$
$plot(x, y)$
figure
$z = exp(x);$
$plot(x, z)$

El resultado será:



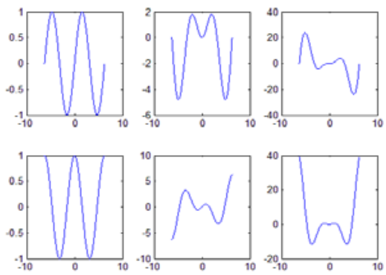
3.1.2. *Subplot*.

El comando *subplot*, nos permite realizar distintas gráficas y distribuirla según una matriz, dentro de una misma figura. Su terminología es $subplot(n, m, i)$, donde n nos indica el número de filas de la matriz, m, el número de columnas, e i, la situación dentro de la matriz, contando por filas y empezando por la izquierda.

Ejemplo 3.1.1.– Deseamos representar, dentro de la misma ventana, pero en gráficas diferentes, las funciones $f(x) = sen(x)$, $g(x) = cos(x)$, $h(x) = xsen(x)$, $t(x) = xcos(x)$, $u(x) = x^2sen(x)$ y $v(x) = x^2cos(x)$, todas ellas, en el intervalo $[-2\pi, 2\pi]$. Para que se vean comparativamente las funciones seno y coseno, pondremos en la primera fila, todas las que tienen seno, y en la segunda fila, todas las que tienen coseno. Es decir, queremos representarlas en una rejilla de 2×3 celdas. Para ello:

$x = -2 * pi : pi/100 : 2 * pi;$
$y1 = sin(x); y2 = x. * sin(x); y3 = (x.^2). * sin(x);$
$z1 = cos(x); z2 = x. * cos(x); z3 = (x.^2). * cos(x);$
$subplot(2, 3, 1); plot(x, y1)$
$subplot(2, 3, 2); plot(x, y2)$
$subplot(2, 3, 3); plot(x, y3)$
$subplot(2, 3, 4); plot(x, z1)$
$subplot(2, 3, 5); plot(x, z2)$
$subplot(2, 3, 6); plot(x, z3)$

El resultado es:



Ejercicio 3.1.2.– Representar las funciones $f(x) = x^2$, $g(x) = e^{x^2}$, $h(x) = sen(x^2)$ y $t(x) = log_10(1 + x^2)$, en una matriz de 2×2.

3.1.3. Aspectos del gráfico.

Si se desea, Matlab permite modificar el aspecto del gráfico con algunos comandos:

- $xlabel('\ldots')$, $ylabel('\ldots')$, permiten poner una etiqueta en los ejes x e y respectivamente. También tenemos el comando $gtext('\ldots')$, que nos permite poner la etiqueta donde deseemos usando el ratón sobre el gráfico.

- Para modificar el aspecto de una línea, se tienen, entre otras, las siguientes opciones: $+$, para poner símbolos de suma, $-$, para poner una línea discontinua, ., para poner una línea con puntos, x, $*$, ... Para usarlos, se pondría, por ejemplo, $plot(x, y, '+')$.

- Para cambiar el color de una gráfica, se añade (en general), la inicial del nombre del color que se desea (en inglés), por ejemplo r es para ponerlas en rojo, b, para azul, y para amarillo, g para verde, k para negro, $\ldots$, se usaría de manera análoga a los estilos de línea: $plot(x, y,' r')$
- Si se desean cambiar simultáneamente el estilo de la línea y el color, se ponen ambos comando juntos entre las comillas: $plot(x, y,' r+')$.
- Para modificar la amplitud de los ejes (siempre dentro de los límites en los que tenemos definidas las variables), se puede usar el comando $axis([xmin, xmax, ymin, ymax])$, donde $xmin$ simboliza la mínima x que deseamos ver, $xmax$, la máxima, y análogamente, con $ymin$ e $ymax$. Otras opciones son $axis\ square$, para que los ejes definan un cuadrado, o $axis('equal')$, si queremos que en ambos ejes tengamos la misma escala.
- Para que nos aparezca la cuadrícula (o no), tenemos el comando $grid\ on$ ($grid\ off$).

3.1.4. Coloreado.

El comando para colorear en Matlab, es el comando $fill$.

Ejemplo 3.1.1.– Queremos colorear el pentágono irregular definido por los vértices $(1, 1)$, $(3, 2)$,$(4, 5)$,$(3, 7)$ y $(2, 5)$.
$x = [1, 3, 4, 3, 2, 1];$
$y = [1, 2, 5, 7, 5, 1];$
$fill(x, y,' r')$

A la hora de usar el comando $fill$, hay que tener en cuenta que colorea recintos cerrados, de tal manera que si el recinto que le damos no es, en principio, cerrado, unirá el último punto con el primero para realizar el coloreado, es decir, habríamos obtenido el mismo resultado si en el ejemplo anterior hubiésemos puesto:
$x = [1, 3, 4, 3, 2];$
$y = [1, 2, 5, 7, 5];$
$fill(x, y,' r')$

Ejercicio 3.1.2.– Colorear de color verde, el recinto delimitado por los puntos $(1, 2)$,$(4, 1)$,$(3, 3)$,$(5, 5)$ y $(5, 2)$.

3.2. Dibujar en el espacio.

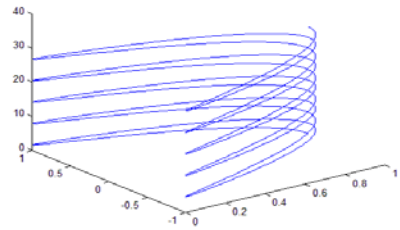
3.2.1. Curvas en el espacio.

Si la curva que queremos dibujar, está escrita en modo paramétrico, se puede visualizar utilizando el comando $plot3$.

Ejemplo 3.2.1.– Queremos dibujar la curva de ecuaciones implícitas $(cos^2(t), sen(t), t)$, con $t \in [0, 10\pi]$, para ello:
$t = 0 : 0{,}01 : 10 * pi;$
$x = cos(t). \wedge 2; y = sin(t); z = t;$

$plot3(x, y, z)$

El resultado es:



Ejercicio 3.2.2.– Representar las siguientes curvas en el espacio:

1. $f(t) = (2cos^3(t), 2sen^3(t), t)$, para $-4 < t < 3$.
2. $f(t) = (e^t 4sen(2t), e^t 4cos(2t), \frac{t}{4})$, $-10 < t < 5$.
3. $f(t) = (sen(2t) + sen(t), -cos(2t) - cos(t), \frac{t}{6})$, $-9 < t < 10$.

3.2.2. Superficies en el espacio.

Para dibujar una superficie en el espacio, hay que tener en cuenta que necesitamos una cuadrícula para valorar la función, no dos simples vectores lineales. Matlab permite generar dicha cuadrícula en base a dos vectores que definamos como ejes X e Y utilizando el comando *meshgrid*.

Ejemplo 3.2.1.– $a = 0 : 3; b = 2 : 5;$
$[x, y] = meshgrid(a, b)$

Nos generaría una cuadrícula con los puntos $(0, 2), (0, 3), (0, 4), (0, 5), (1, 2), (1, 3), \ldots (3, 5)$

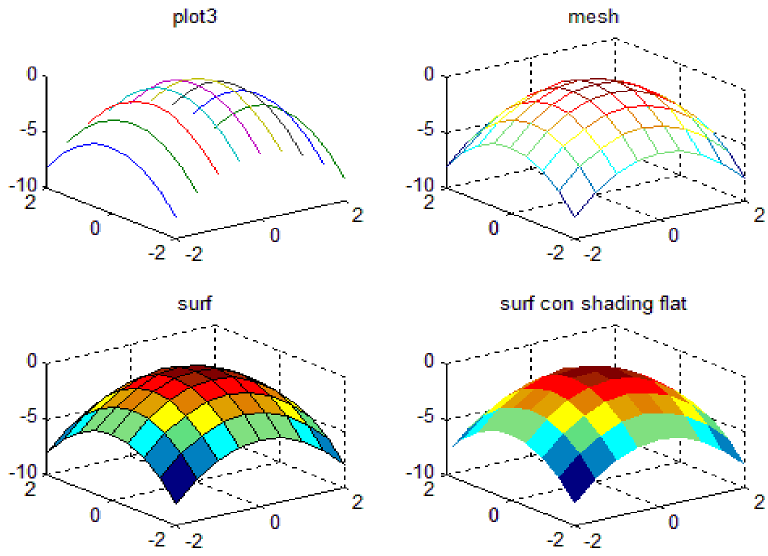
Si ponemos sólo un vector x en $meshgrid$, nos generaría la misma cuadrícula que si pusiéramos $meshgrid(x, x)$.Por ejemplo, $meshgrid(0 : 3)$, nos generaría los puntos $(0, 0), (0, 1), (0, 2), (0, 3), (1, 0), (1, 1), \ldots, (3, 3)$

Para dibujar una superficie, una vez generada la rejilla, y con ecuación $z = f(x, y)$, se pueden usar los comandos $plot3$, $mesh$ y $surf$, a éste último, además se le puede añadir el comando $shading\ flat$, que nos muestra el gráfico con el sombreado distinto.

Ejemplo 3.2.2.– Queremos dibujar $z = -x^2 - y^2$, para $x, y \in [-2, 2]$, usando los tres comandos, para ello, vamos a utilizar, además, el comando *subplot*.

$[x, y] = meshgrid(-2 : 0{,}5 : 2);$
$z = -x.^2 - y.^2;$
$subplot(2, 2, 1),\ plot3(x, y, z),\ title('plot3')$
$subplot(2, 2, 2),\ mesh(x, y, z),\ title('mesh')$
$subplot(2, 2, 3),\ surf(x, y, z),\ title('surf')$
$subplot(2, 2, 1),\ surf(x, y, z),\ shadingflat,\ title('surf\ con\ shading\ flat')$

El resultado es:



Ejercicio 3.2.3.– Representar las siguientes superficies:

1. $z = -\sqrt{|xy|}$, $x, y \in [-2, 2]$ (Haced el salto de 0.01).

2. $z = \frac{cos(\frac{x^2+y^2}{4})}{3+x^2+y^2}$, $x, y \in [-2\pi, 2\pi]$

3. $z = \frac{y^2}{5} - 3|x|$, $x, y \in [-3, 3]$

4. $z = \sqrt{4 - x^2 + y^2}$, $x, y \in [-2, 2]$

Matlab también permite dibujar las superficies más sencillas con comandos preestablecidos:

- Para dibujar una esfera de radio 1, se utiliza el comando *sphere*, el argumento que se le pone, es el número de meridianos que deseamos ver, por defecto, si no se pone nada, considera 20. Para dibujar cualquier otra esfera o elipsoide, se utiliza el comando *ellipsoid*, en este comando, hay que indicar el centro y la longitud de cada eje dividida entre 2 (el radio, en el caso de una esfera), por ejemplo, si queremos dibujar una esfera centrada en el $(1, 1, 1)$ y con radio 3, escribiríamos $ellipsoid(1, 1, 1, 3, 3, 3)$, si queremos un elipsoide centrado en el $(1, 1, 1)$, y con semiejes $(2, 4, 1)$, escribiríamos $ellipsoid(1, 1, 1, 2, 4, 1)$

- Para dibujar un cilindro de radio r, usamos el comando $cylinder(r, n)$, donde n es el número de líneas que queremos que aparezcan. Por ejemplo, el cilindro de radio 3, puede ser $cylinder(3, 20)$ Este comando permite radios variables, por lo que permitiría dibujar superficies de revolución, por ejemplo, si queremos dibujar la superficie obtenida al girar la parábola $y = \sqrt{x}$, bastaría escribir:

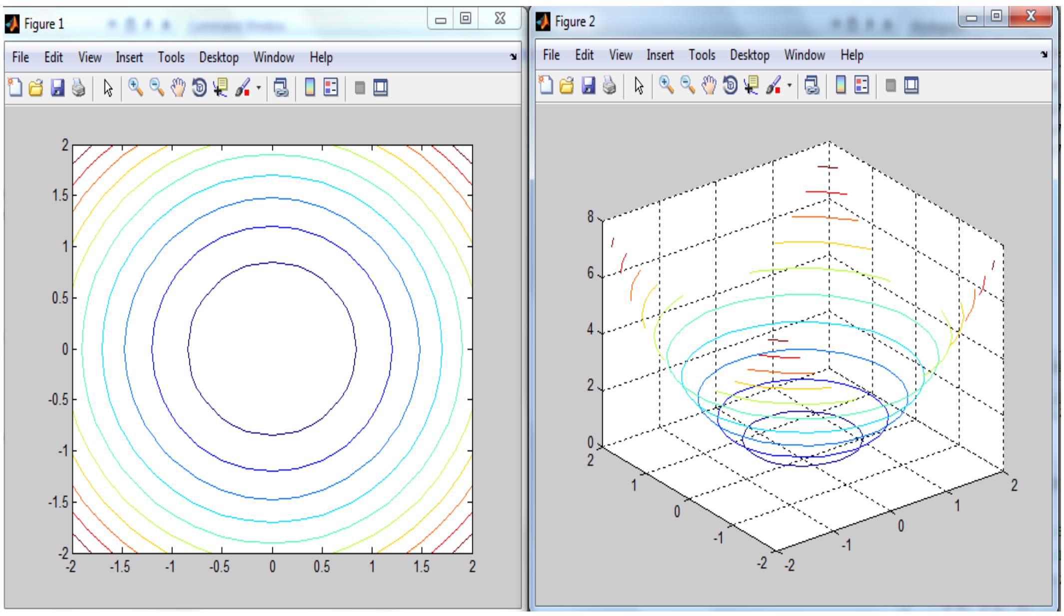
  $x = 0 : 0{,}25 : 4;$
  $cylynder(x.^(1/2), 20)$

Por último, el comando *rotate3d*, nos permite rotar el gráfico una vez realizado utilizando el ratón.

3.2.3. Curvas de nivel.

Las curvas de nivel se obtienen cuando una función $f(x, y)$ se iguala a una constante. Para obtener las curvas de nivel de un determinado gráfico, se usa el comando *contour*, si deseamos verlas en el plano, o *contour3* si las queremos ver en el espacio. El argumento será: las coordenadas (x, y, z) y el número de curvas que deseamos ver.

Ejemplo 3.2.1.–
Vamos a dibujar las curvas de nivel de la superficie generada por $f(x, y) = x^2 + y^2$.
$[x, y] = meshgrid(-2 : 0{,}25 : 2);$
$z = x.^2 + y.^2$
$contour(x, y, z, 10)$
$figure$
$contour3(x, y, z, 10)$



Ejercicio 3.2.2.– Dar las curvas de nivel de las siguientes gráficas:

1. $z = -\sqrt{|xy|}$, $x, y \in [-2, 2]$.

2. $z = \frac{cos(\frac{x^2+y^2}{4})}{3+x^2+y^2}$, $x, y \in [-2\pi, 2\pi]$

3. $z = \frac{y^2}{5} - 3|x|$, $x, y \in [-3, 3]$

4. $z = \sqrt{4 - x^2 + y^2}$, $x, y \in [-2, 2]$