Tema II: Funciones de una variable

1 Introducción

Una de las utilidades más importantes de MatLab desde la perspectiva de los objetivos de este texto es su adecuación para representar gráficamente la variación de una magnitud circuital (tensión, corriente, potencia, etcétera) con el tiempo (análisis temporal, análisis en régimen transitorio) o con la frecuencia de operación del circuito (respuesta en frecuencia). Formalmente hablando, ambos tipos de variaciones pueden tratarse de forma unificada haciendo referencia a la representación gráfica de funciones de una sola variable. En consecuencia, y a menos que sea imprescindible, en lo sucesivo designaremos la variable independiente mediante la letra x y la función, mediante $y(x)$ o simplemente y.

En este tema presentaremos algunos aspectos relevantes de tal tipo de representación. Mientras sea posible no indicaremos explícitamente si nos referimos al tiempo o a la frecuencia como variable independiente (es decir, la correspondiente al eje de abscisas de la representación; los correspondientes valores que toma la magnitud cuya variación deseamos conocer se muestran en el eje de ordenadas). Debe tenerse en cuenta que las opciones que mencionamos en este tema son únicamente algunas de las posibles, ya que una de las grandes ventajas de MatLab es que se trata de un paquete de software sumamente versátil, con el que es posible resolver problemas por diversos procedimientos alternativos. Es la experiencia previa la que dicta, en cada caso concreto, cuál de estas alternativas es la más conveniente.

2 Definición de la variable independiente

La variable independiente se define como un vector que está constituido por todos los valores de x para los que queremos obtener los correspondientes valores de y. Así, si, por ejemplo, queremos representar la variación de una magnitud en el intervalo de tiempos comprendido entre -10 s y 10 s, los valores inicial y final del vector serían precisamente los que acabamos de indicar. Si introducimos este vector en un programa destinado a calcular y para cualquier valor de x, el programa va dando sucesivamente a la variable independiente los valores del vector y obteniendo los correspondientes a la función.

Idealmente, querríamos que el vector correspondiente a x fuera un conjunto infinito que incluyera cualquier posible valor de x comprendido entre -10 s y 10 s. De esa forma podríamos obtener, al menos en teoría, una representación absolutamente precisa de la variación de la función con el tiempo. En otras palabras, tendríamos una representación continua de $y(x)$. Sin embargo, MatLab no permite semejante posibilidad, ya que es una herramienta discreta. Es decir, el vector representativo de la variable independiente no puede ser continuo, o, expresado de otro modo, ha de ser un conjunto finito de valores (puede ser todo lo grande que se quiera -dentro de las limitaciones que imponen MatLab y el ordenador en el que corre el paquete-, pero no infinito).

MatLab dispone de varias posibilidades para definir el vector que constituye la variable independiente. Las más relevantes desde el punto de vista de este texto son las que se mencionan seguidamente.

- Variable independiente como vector lineal. El vector x queda constituido por un conjunto de puntos equiespaciados entre los valores inicial y final del rango, los cuales deben ser computados en dicho conjunto. La instrucción para utilizar esta posibilidad es la mostrada[1] en la figura II.1

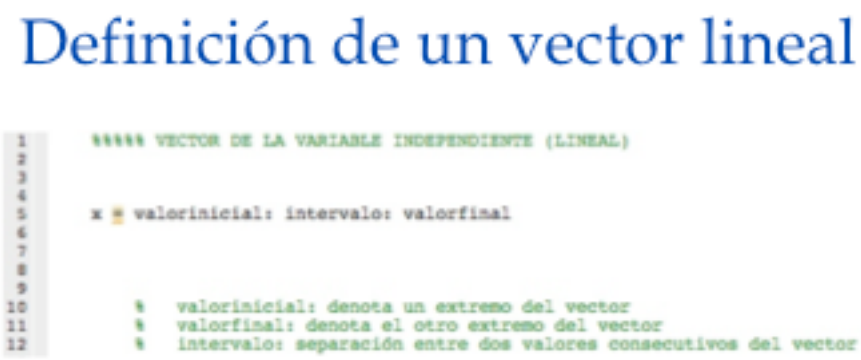

Figura II.1. Instrucción correspondiente a la definición de un vector lineal.

Por ejemplo, si la instrucción es `x = 1: 1: 10`, estamos indicando que los valores inicial y final del rango de variación de `x` son `1` y `10`, respectivamente, y que la diferencia entre dos valores consecutivos es `1`. Es decir, el vector de la variable independiente está constituido por los valores `1, 2,..., 9, 10`.

- Variable independiente como vector lineal (instrucción alternativa). Una forma alternativa de la que se acaba de mencionar para definir el vector correspondiente a la variable independiente es la mostrada en la figura II.2.

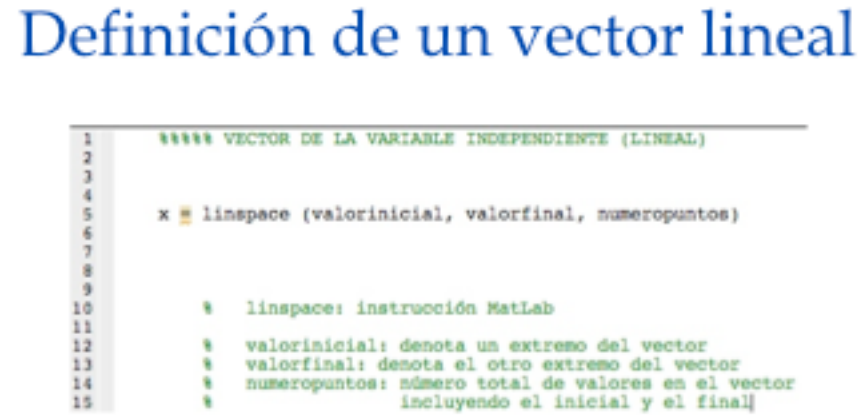

Figura II.2. Instrucción correspondiente a la definición de un vector lineal.

[1] En lo sucesivo utilizaremos las siguientes reglas para la representación de instrucciones: la instrucción propiamente dicha va en negro, mientras que los comentarios adicionales (no tienen por qué ser utilizados en programas reales) van en un tono más claro; indicaciones como `valorinicial`, por ejemplo, denotan valores numéricos que han de ser proporcionados por el usuario antes de la ejecución de la instrucción, y las funciones propias de MatLab, que han de ser utilizadas exactamente en la forma indicada, son indicadas explícitamente.

Por ejemplo, si la instrucción es `x = linspace (1, 10, 100)`, estamos indicando que el vector correspondiente a la variable independiente contiene `100` puntos (entre los que se incluyen el inicial y el final) equiespaciados. Si el número de puntos es, precisamente, `100`, puede omitirse su indicación explícita en la instrucción (en cuyo caso también hay que prescindir del símbolo `,` que lo precede).

- Variable independiente como vector logarítmico. Si se utiliza este recurso, el equiespaciado entre los puntos del vector que define la variable independiente no es lineal (como ocurría en los dos casos anteriores), sino logarítmico. La definición del vector se hace como se indica en la figura II.3.

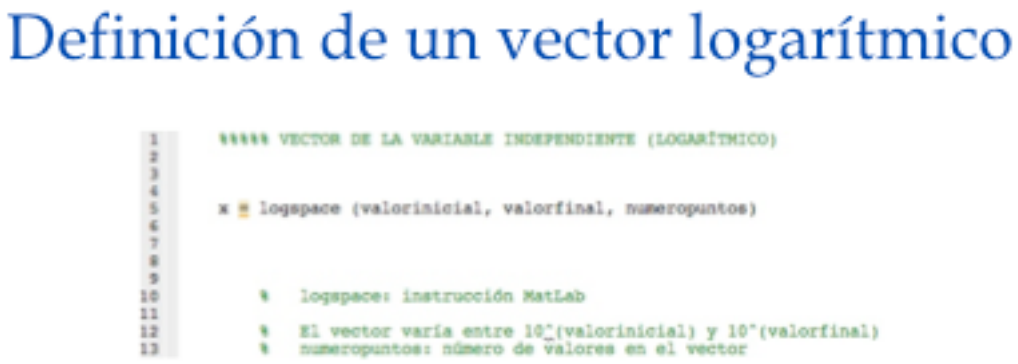

Figura II.3. Instrucción correspondiente a la definición de un vector con variación logarítmica.

Por ejemplo, si la instrucción es `x = logspace (3, 9, 1000)`, estamos indicando que los valores inicial y final del vector son, respectivamente, 10^3 y 10^9 y que aquél incluye un total de `1000` puntos entre uno y otro (ambos inclusive). Si el número de puntos es `50`, pueden omitirse su indicación y el símbolo `,` que la precede).

Hay tres circunstancias que conviene tener en cuenta en relación con la definición del vector que constituye la variable independiente.

- La elección de los valores que han de tener parámetros como `intervalo` o `numeropuntos` está condicionada por dos requisitos contrapuestos. Por una parte, cuanto más pequeños sean tales parámetros, más precisa será la representación de la función. En otras palabras, una representación gráfica de ésta tendrá un aspecto más *continuo*, mientras que, cuanto mayores sean aquéllos, más se parecerá la representación a un conjunto de puntos unidos por líneas generadas automáticamente por MatLab. Además, cuanto menores sean los valores de los parámetros citados, más tiempo tardará MatLab en completar la definición del vector y mayor espacio de memoria se ocupará en el ordenador; de hecho, para unos parámetros muy pequeños, que den origen a un gran número de valores en el vector de la variable independiente, la instrucción puede ser rechazada por MatLab, ya que éste impone un límite máximo (que depende de la versión concreta de MatLab de la que se trate) al número de elementos que puede contener un vector.
- Como podrá deducirse a raíz de lo expuesto hasta el momento, los distintos parámetros que aparecen en las instrucciones son números reales. Adicionalmente,

los parámetros `intervalo` y `numeropuntos` han de ser positivos. Finalmente, el segundo de los citados ha de tener, además, un valor entero.

- Obsérvese que la opción de la variación logarítmica es particularmente adecuada para la representación de la respuesta en frecuencia de un circuito, mientras que las dos correspondientes a variación lineal son más idóneas para representar variaciones de magnitudes con el tiempo.

3 Aspectos relacionados con vectores

El tratamiento de la variable independiente como un vector sugiere añadir algunas consideraciones adicionales sobre el manejo de vectores en MatLab, que pueden ser de utilidad a la hora de elaborar programas que incluyan tales elementos. Tales consideraciones aparecen resumidas en la figura II.4.

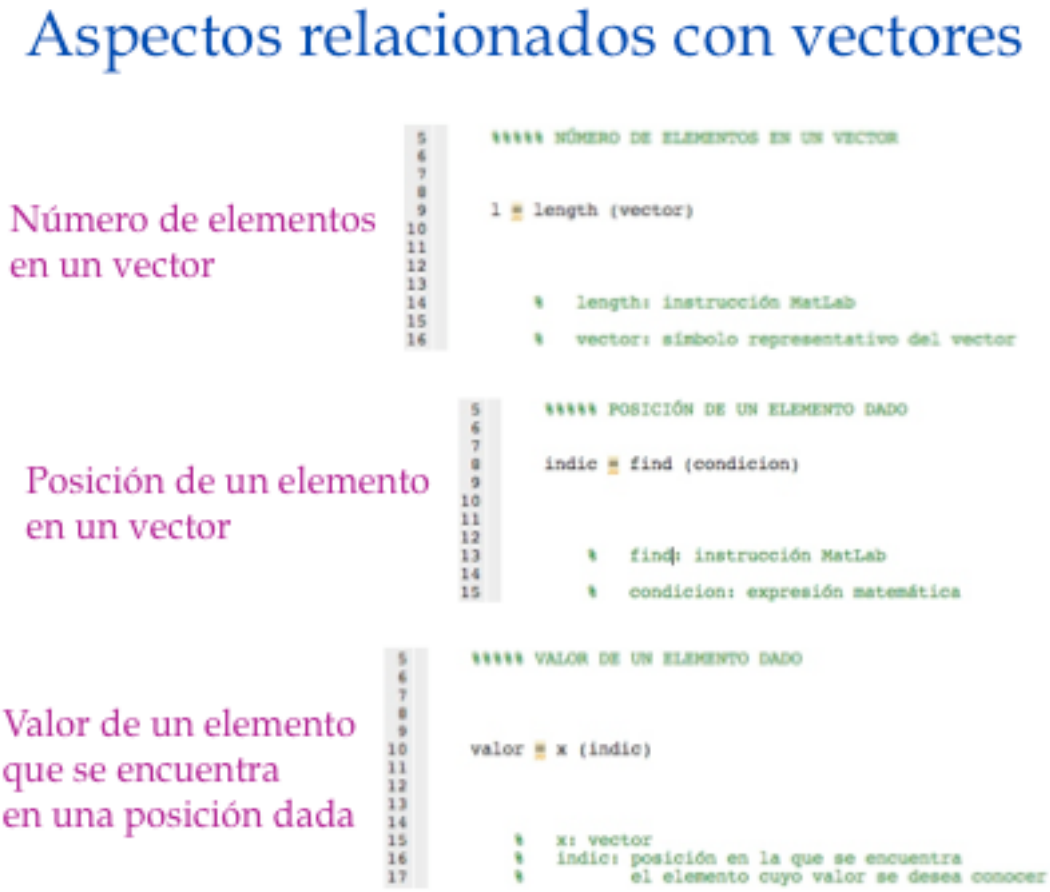

Figura II.4. Instrucciones para el manejo de vectores en MatLab.

En la figura II.5 pueden verse algunos ejemplos de utilización de los conceptos mostrados en la figura II.4.

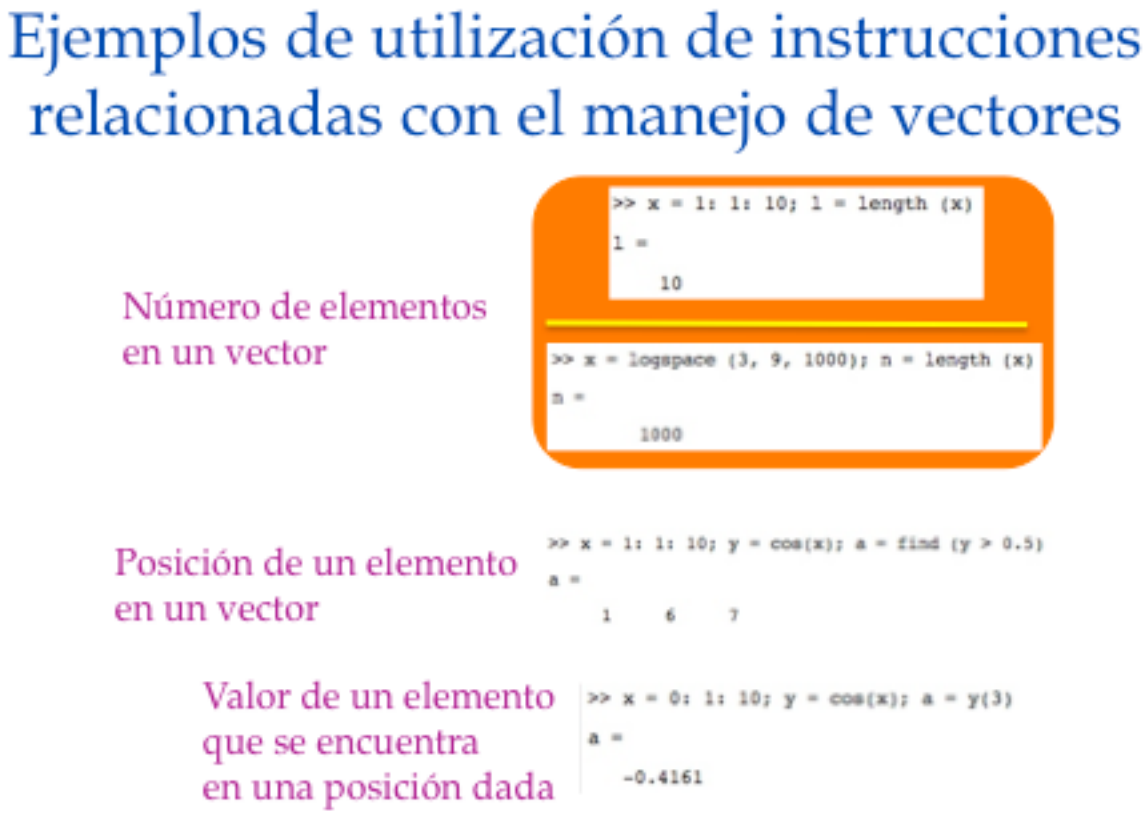

Figura II.5. Ejemplos de utilización de los conceptos presentados en la figura anterior.

La instrucción `length` permite determinar el número de componentes de un vector. En la figura II.5 se muestran dos ejemplos de su utilización. Obsérvese que, en ambos, el número de elementos del vector coincide con lo que se indicó anteriormente en relación con este aspecto.

La instrucción `find` permite determinar las posiciones dentro del vector que ocupan los elementos que satisfacen una condición determinada, que ha de ser establecida por el usuario; en el caso representado en la figura II.5 esa condición es que los elementos del vector tengan un valor superior a `0.5`. Con relación a este ejemplo conviene resaltar dos aspectos. El primero es que, si `x` es un vector, también lo será una función, como `y=cos(x)`, definida a partir de aquél. El segundo aspecto es que el usuario no debe olvidar que el resultado de la instrucción `find` es una (o más) posición dentro del vector, no el valor del elemento que ocupa dicha posición. Así, a partir de los resultados mostrados en la figura, puede deducirse que los elementos que ocupan las posiciones `1`, `6` y `7` tienen valores superiores a `0.5`. Incidentalmente, también puede observarse que MatLab trata los ángulos como si estuvieran expresados en radianes. La conversión de estas unidades a grados sexagesimales puede realizarse empleando la expresión

$$\alpha\,[^\circ] = \frac{180}{\pi}\alpha\,[\text{rad}]$$

El tercero de los aspectos mencionados en la figura II.4 representa una alternativa a la determinación del valor que toma una función para un valor dado de la variable independiente. En este caso se pregunta por el valor del elemento que ocupa una posición dada en el vector que define la función sin necesidad de conocer previamente el valor del elemento que ocupa la misma posición en el vector que define la variable dependiente. Es importante destacar la

circunstancia de que, mientras que `x`, por ejemplo, representa un vector completo, `x(k)` denota el valor del elemento que ocupa la posición `k` en el vector.

4 Representación gráfica de una función

El caso más sencillo de representación gráfica se presenta cuando se hace referencia a un vector del que se conocen los valores de todos los elementos. Es el caso indicado en la figura II.6, en la que se muestra la forma de proceder cuando se da esta situación. Obsérvese que `plot` es la instrucción de MatLab más utilizada en tareas de representación gráfica.

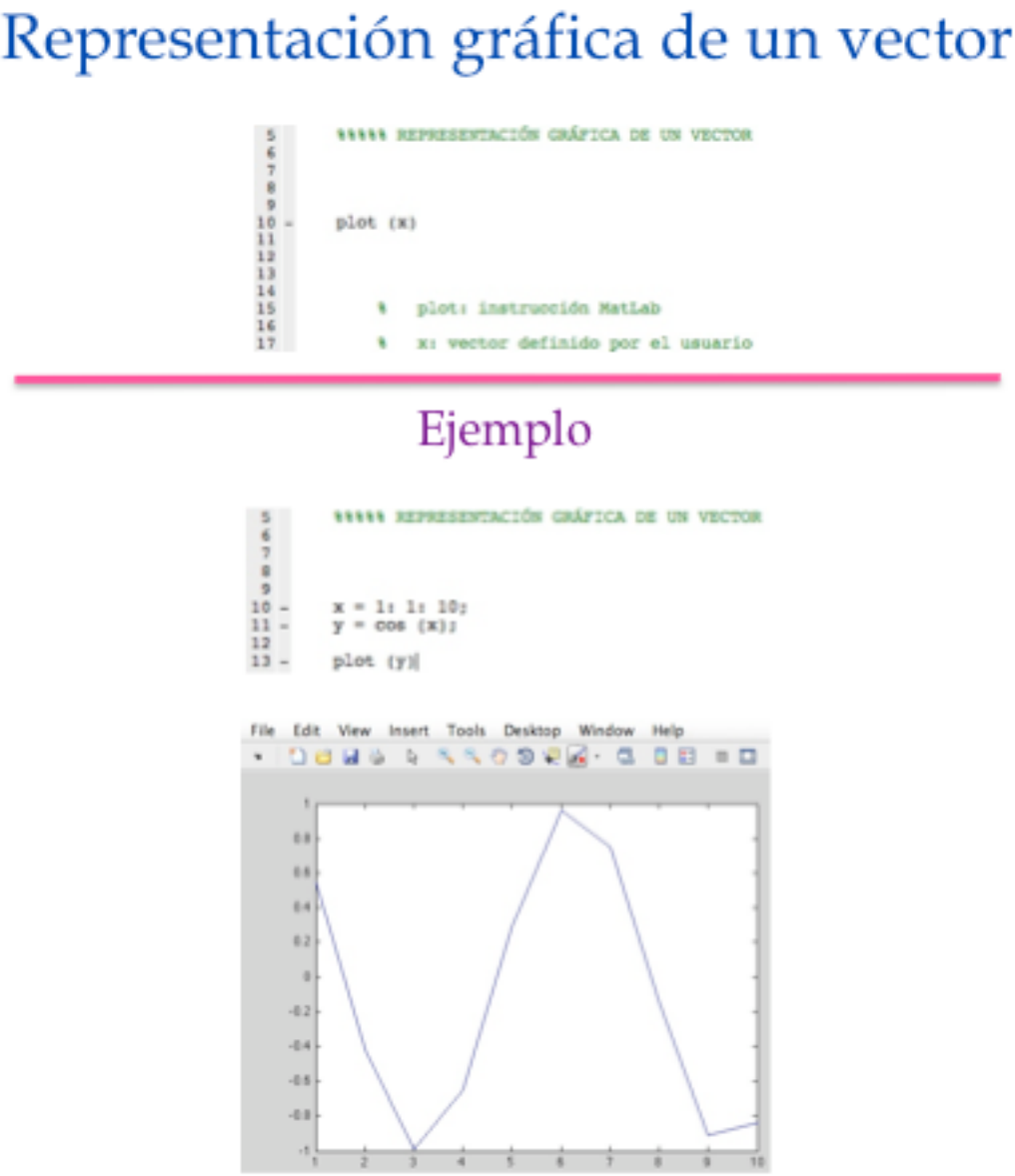

Figura II.6. Representación gráfica de una función definida por un vector.

Refiriéndonos al ejemplo, el resultado del programa es la figura mostrada. Si se desea modificar su presentación (no su contenido), puede utilizarse, como se indicó anteriormente, la opción `Figure Properties...`, incluida en el menú `Edit`. Esta modificación no afecta en ningún caso al rango de valores de `x`, ya que éste ha sido establecido por el programa. Obsérvese que la representación (corresponde a una función coseno) es bastante defectuosa, lo cual se debe a la gran separación entre los elementos del vector que define el eje de abscisas. Se recomienda al lector que repita el ejercicio mostrado en el ejemplo, pero estableciendo una mayor proximidad entre tales elementos.

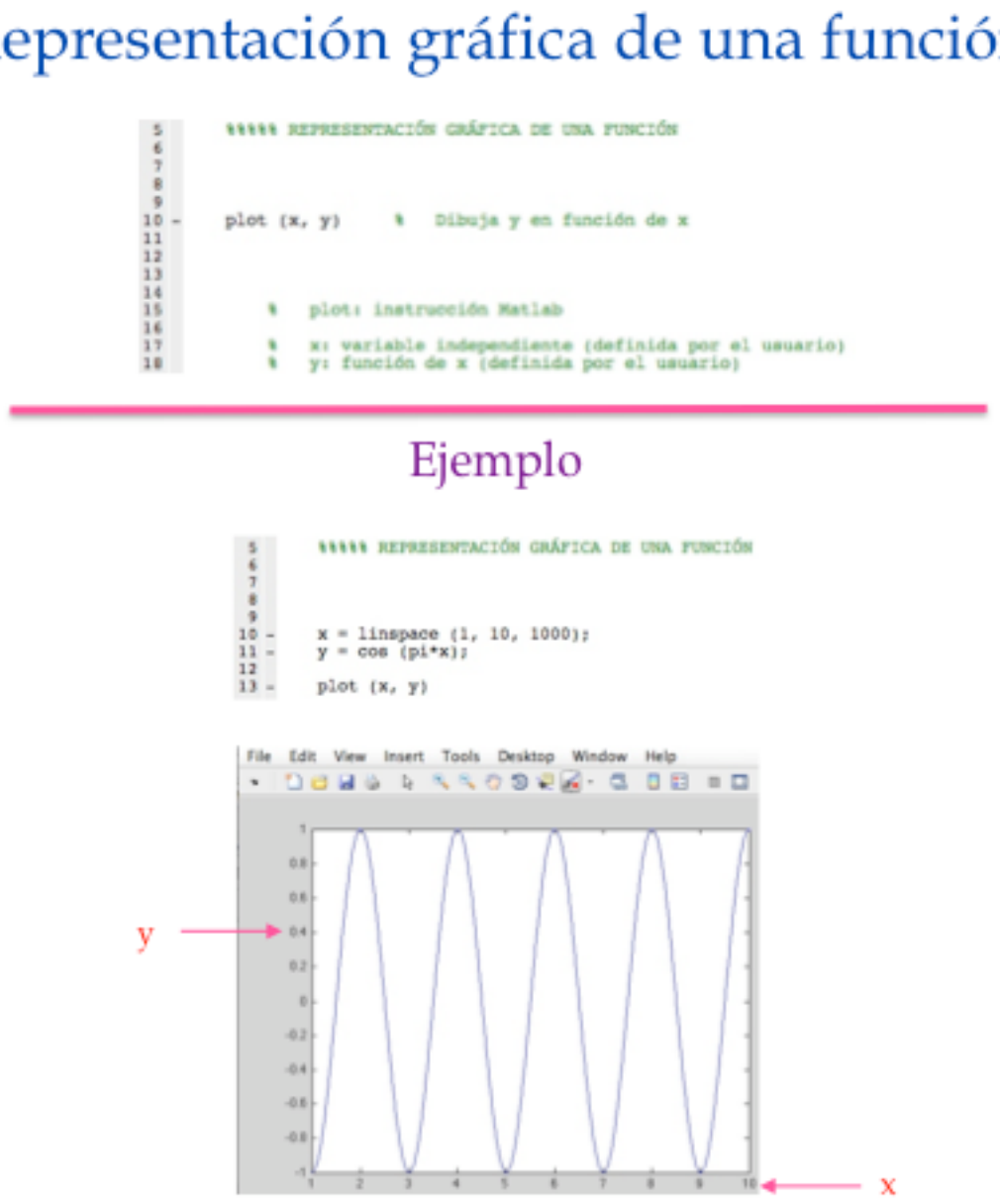

Figura II.7. Representación gráfica de una función.

La situación más habitual en la práctica es la resumida en la figura II.7. Como puede verse, la ejecución de la instrucción `plot` requiere la definición previa del vector que constituye el eje

de abscisas, así como la de la función cuyos valores han de ser representados en el eje de ordenadas[1].

A la hora de utilizar la instrucción `plot` es preciso tener en cuenta que no es indiferente el orden en el que se disponen la variable y la función en el paréntesis utilizado para definirla. Se recomienda al lector que repita el ejemplo contenido en la figura II.7, pero intercambiando las posiciones de `x` e `y` en la definición de tal instrucción. Obsérvese incidentalmente que el símbolo π se representa en MatLab como `pi`.

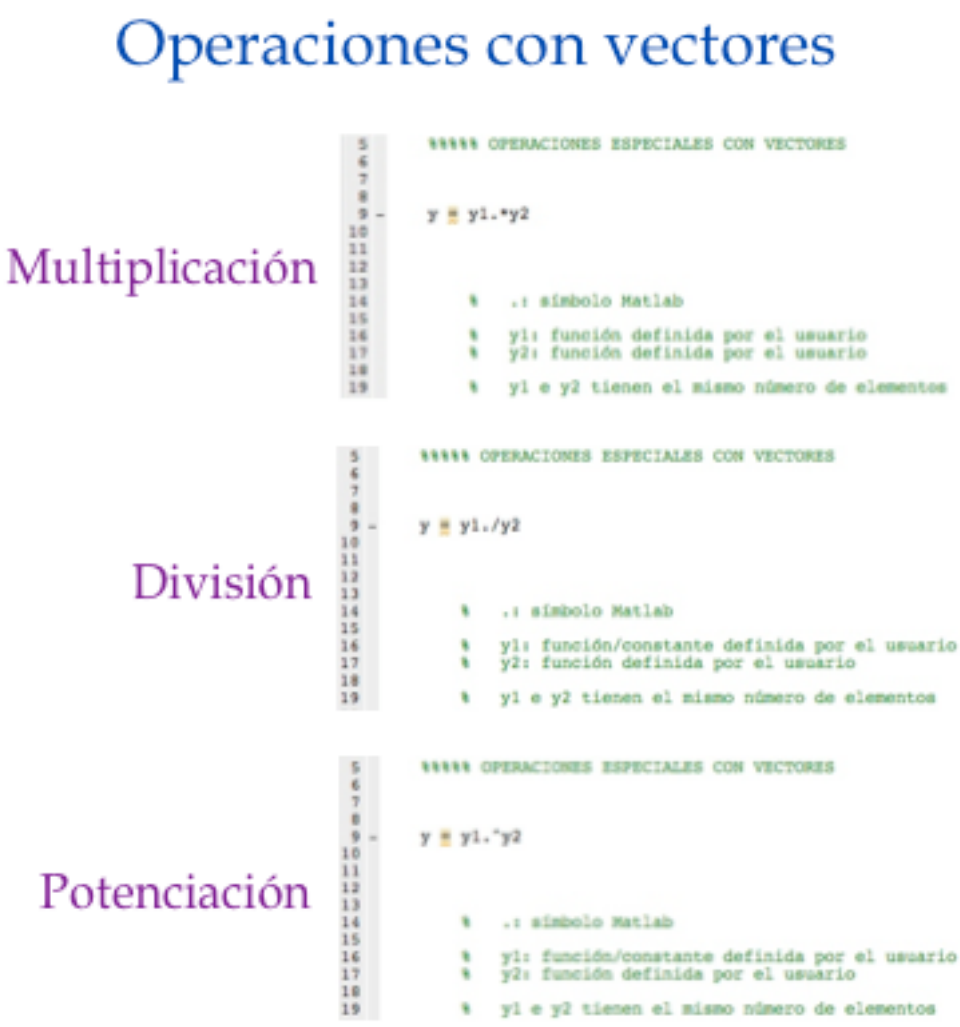

Figura II.8. Notación para operaciones con vectores.

Hay un aspecto muy importante que debe ser tenido en cuenta en la definición de funciones (aunque se trate, como es el caso, de funciones de una variable). Mientras una función sea el resultado de una suma o una diferencia de funciones (en ambos casos, los vectores que definen las funciones han de tener el mismo número de elementos), el resultado de multiplicar una constante por una función (es el caso de la función cosenoidal utilizada en las figuras II.6 y II.7), o el resultado de dividir una función por una constante, no es preciso tomar ninguna

[1] Estamos haciendo referencia a una representación bidimensional porque, como se indicó con anterioridad, en este texto tratamos únicamente el caso de funciones que dependen de una sola variable.

precaución especial. Ahora bien, cuando se trata de multiplicar dos funciones definidas por vectores, o de dividir algo (una constante u otra función) por una función, o de elevar algo (una constante o una función) a otra función, es preciso utilizar el símbolo `.` para indicar que se trata de operaciones con vectores y que se desea obtener como resultado un vector con el mismo número de elementos que los que participan en la operación. En la figura II.8 se resumen estos casos *especiales* y se indica la forma de tratarlos.

Retornando al tema específico de la representación gráfica, puede decirse que hay distintas instrucciones MatLab relacionadas más o menos directamente con `plot`. Se recomienda al lector que haga uso del menú `Help` para ver las características de algunas de tales instrucciones. Entre ellas es interesante mencionar las siguientes:

- `semilogx (x, y(x))`: los valores de la variable independiente son presentados en una escala logarítmica, manteniéndose la escala lineal para la representación de la función.
- `semilogy (x, y(x))`: los valores de la función son presentados en una escala logarítmica, manteniéndose la escala lineal para la representación de la variable independiente.

Cabe señalar que el cambio de escalas también puede realizarse con la opción `Figure Properties...`, seleccionable en el menú `Edit`. La ventaja que tiene la posibilidad que acabamos de presentar es que ya se obtiene directamente la representación en el formato deseado, mientras que en la opción convencional hay que proceder a una manipulación de los ejes.

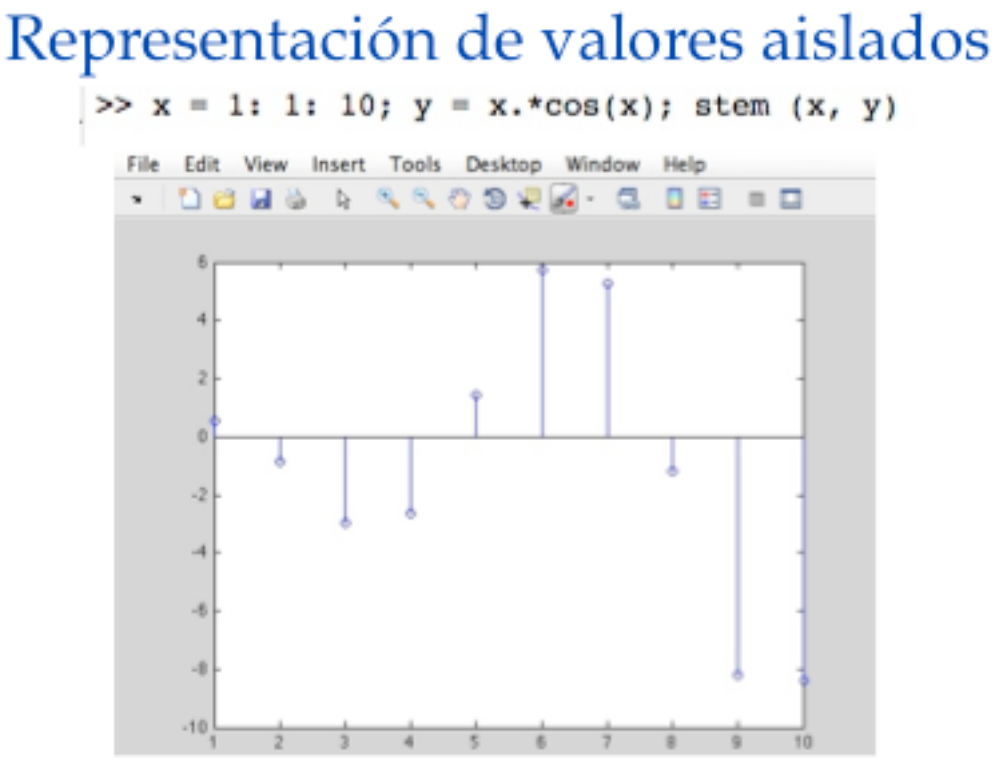

Figura II.9. Ejemplo de utilización de la instrucción `stem`.

- `stem`: representa únicamente los valores que toma la función en los puntos definidos por los elementos representativos del vector que constituye la variable independiente. La figura II.9 muestra un ejemplo del tipo de representación que se obtiene utilizando la instrucción `stem`. La operación se realiza directamente en la ventana de comandos de la pantalla principal de MatLab.

5 Combinación de funciones en una sola representación

En las secciones anteriores se ha tratado el caso de funciones, cada una de las cuales se definía por medio de una única expresión matemática. En la presente sección nos referiremos a funciones que requieren más de una función matemática para quedar completamente definidas. Es el caso, por ejemplo, de la función considerada en la figura II.10.

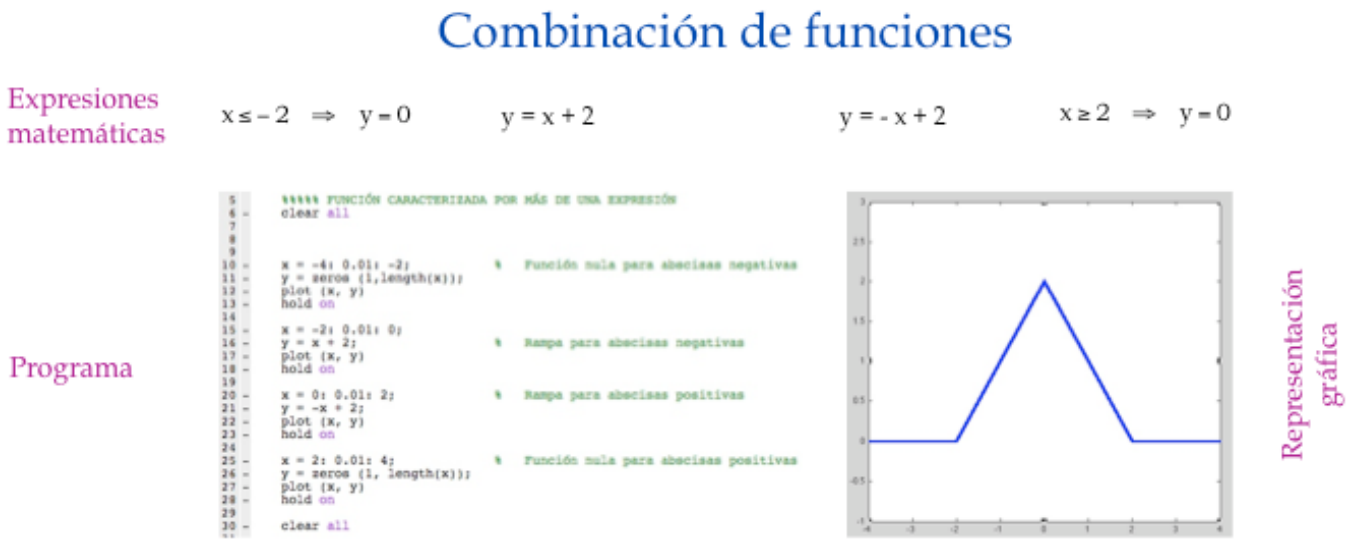

Figura II.10. Ejemplo de función caracterizada por cuatro expresiones matemáticas.

Una posible forma (no la única) de abordar el problema de obtener una representación gráfica de dicha función consiste en ir representando sucesivamente las expresiones que definen la función global, tal y como se hace con el programa indicado en la misma figura.

La representación se divide en tantas zonas como expresiones matemáticas definen la función (cuatro en este caso) y se establece la representación de cada función parcial en la zona en la que está definida.

Empezando por la zona que se encuentra más a la izquierda se define un vector que representa el eje de abscisas en ella; en teoría dicho vector comenzaría en $x = -\infty$, pero el rango elegido es más que suficiente para ilustrar lo que se pretende mostrar. A continuación, y porque así lo establece la expresión matemática correspondiente a esa zona, hay que anular el valor de la función en ella. Hay distintas formas para anular una función en un determinado intervalo de valores de la variable independiente; en el presente caso hemos optado por la solución mostrada en la figura II.11. Obsérvese que el extremo final del intervalo en el que se anula la función ha sido definido directamente (evidentemente, coincide con el número que da la

longitud del vector representativo de la variable independiente) en lugar de calcularlo previamente y luego sustituir el valor obtenido en el formato general de la instrucción `zeros`, representado en la figura II.11.

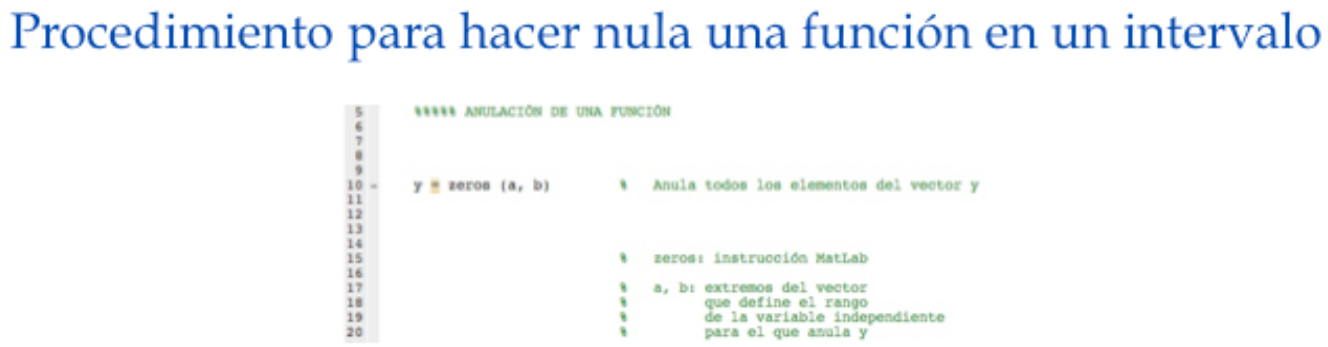

Figura II.11. Un procedimiento para hacer nulos los valores del vector representativo de una función en un intervalo dado de valores de la variable independiente.

Con lo indicado hasta el momento obtendríamos el segmento horizontal situado más a la izquierda en la representación de la figura II.11. A continuación podríamos aplicar el mismo procedimiento para obtener los otros tres segmentos de la representación. Ahora bien, proceder de esta forma sin tomar ninguna precaución adicional es inviable, porque MatLab, al tiempo que dibuja un nuevo segmento, borra el que existía hasta ese momento. Para evitar que ocurra esto es por lo que utilizamos la instrucción `hold on` una vez concluido el dibujo de una parte de la representación; tal instrucción mantiene sin cambios lo que ya existe en una figura cuando se le añade algo nuevo.

Otra posibilidad para representar funciones caracterizadas por más de una expresión matemática consiste en combinar, por un lado, los rangos de la variable independiente y, por otro, las expresiones matemáticas en un rango y una expresión únicos, tal y como se indica en la figura II.12, en la que también se muestra, a título de ejemplo, la aplicación de este procedimiento al caso tratado en la figura II.10. Obsérvese que, con la nueva metodología, se reducen a una las instrucciones `plot` y se prescinde de las instrucciones `hold on`.

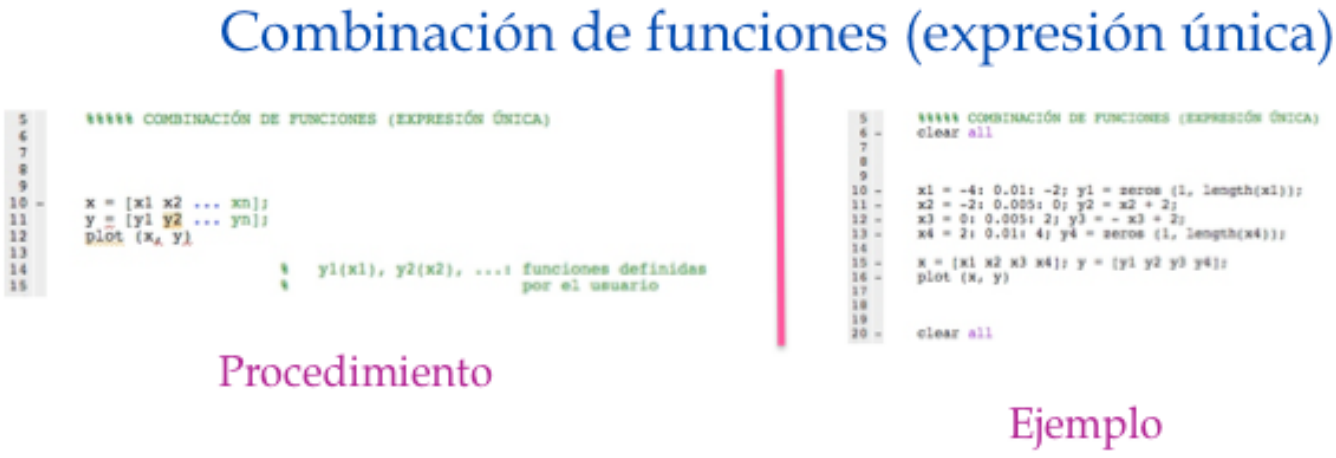

Figura II.12. Combinación de funciones para obtener una representación gráfica única.

6 Ejercicios propuestos

El lector puede determinar el grado en el que ha asimilado los conceptos presentados en este tema intentando resolver los ejercicios que se proponen a continuación (figura II.13). Las soluciones de los mismos pueden ser encontradas en el apéndice 1.

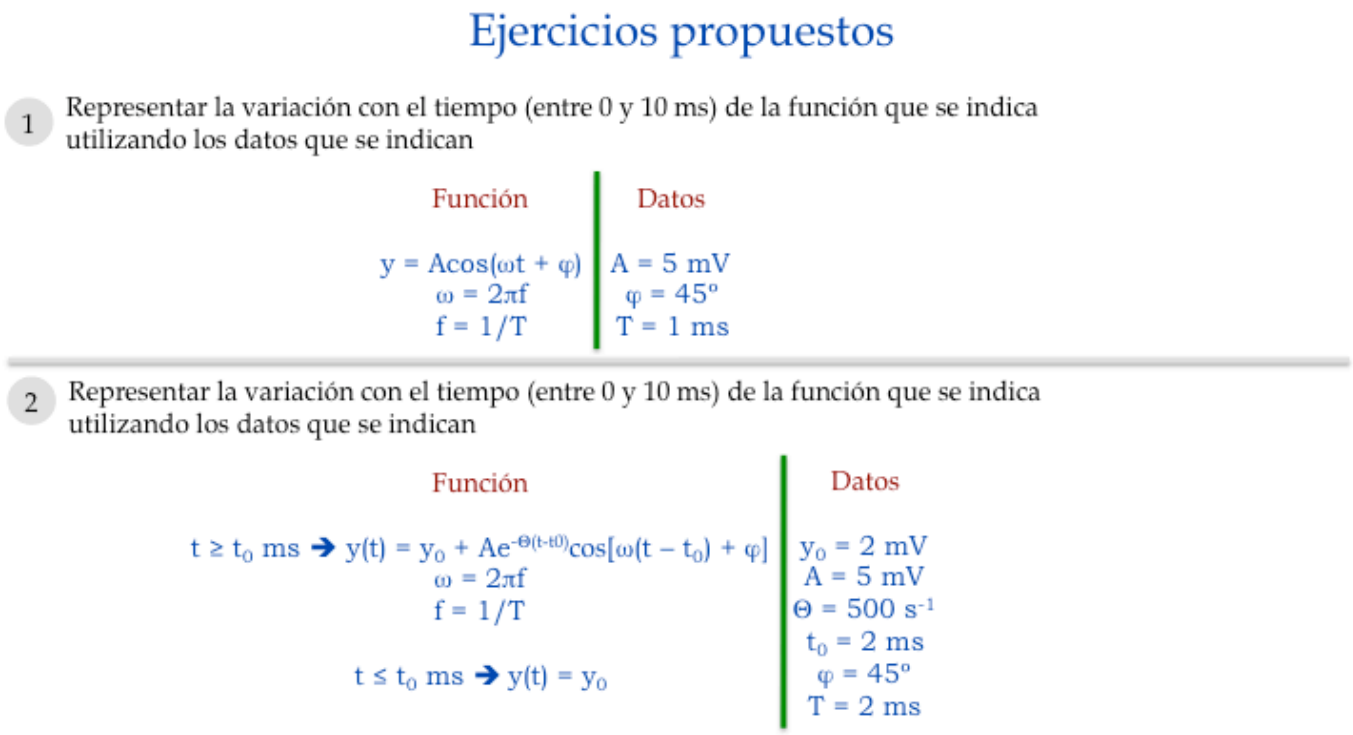

Figura II.13. Ejercicios propuestos relativos a la representación gráfica de funciones.