

Tutorial Básico de Programación en Matlab

(Autor: Armando Pezzente)

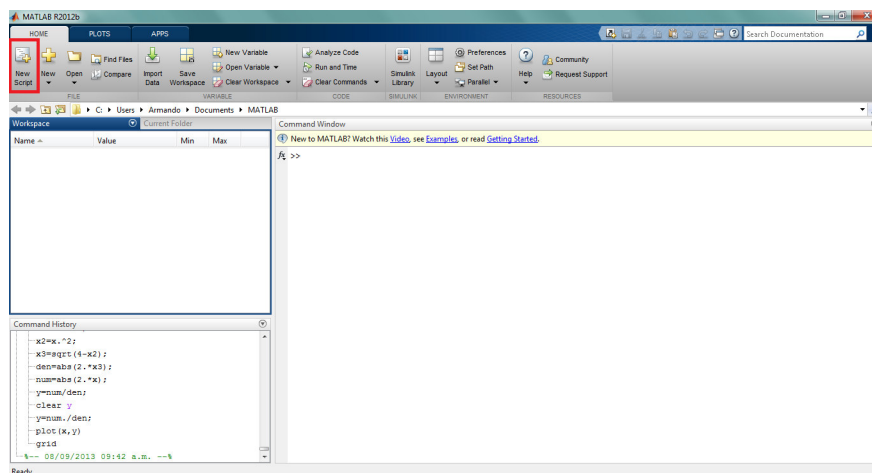
Para este tutorial se utiliza la versión R2012b. Los comandos son los mismos que para versiones anteriores. Cambia el aspecto gráfico.

En este tutorial veremos cómo resolver un problema en particular mediante el uso de funciones.

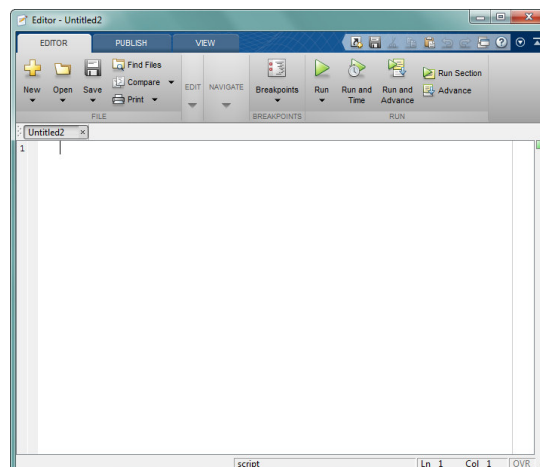
El problema es el Ejercicio 10.a

Programar (usando bucles *for*) una función que retorna un vector *v* con la suma, elemento a elemento, de dos vectores *a* y *b* de la misma longitud. Detectar si los vectores no tienen la misma longitud y dar un error en caso contrario. (Usar la función *length* para determinar la longitud de los vectores) Aplicar a los vectores *a*=[1 2 3], *b*=[2 3 8] y a los vectores *a*=[1 2 3], *b*=[2 8].

Primero abrimos el Matlab y creamos un nuevo script o archivo “.m”. En Matlab 6.5 puede recurrirse al menú “File --> New --> M File”.



La ventana que se abre es el editor de código



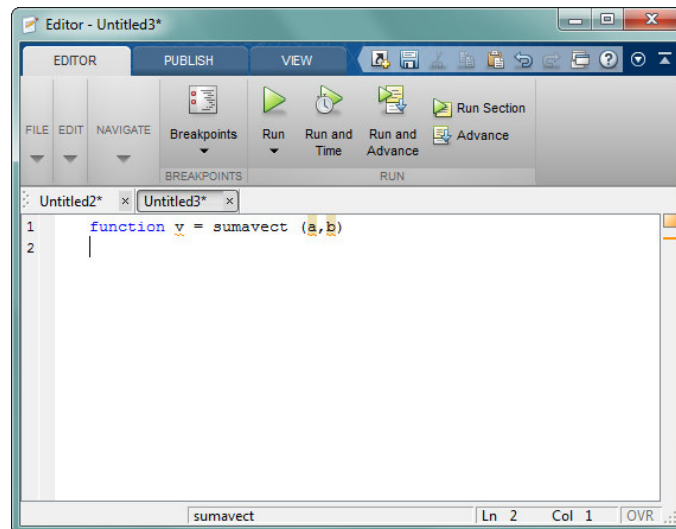
En esta ventana escribimos el programa. El mismo debe seguir la siguiente estructura:

function [variables de salida] = nombre de la función (variables de entrada)

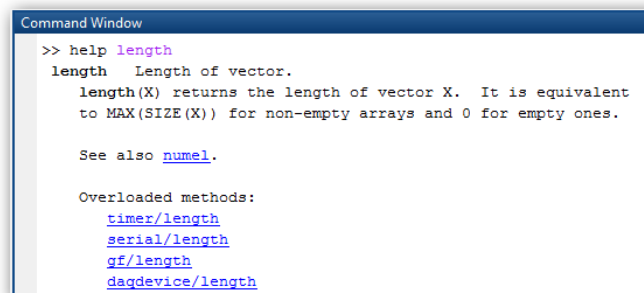
En este caso tenemos una única variable de salida: el vector “v” que contiene la suma. Por lo tanto podemos omitir el uso de los corchetes. Los corchetes se utilizan en el caso de tener salidas múltiples. Por ejemplo, para un programa con salidas s1, s2 y s3, y entradas e1 y e2:

function [s1,s2,s3] = nombre de la función (e1,e2)

Para el ejemplo, llamaremos a la función con el nombre “sumavect”, a las entradas “a” y “b” y a la salida “v” (como especifica el problema)

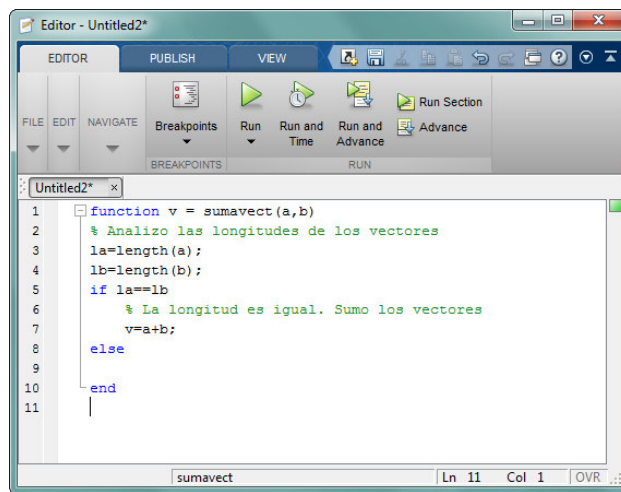


Luego de esto, debemos analizar las condiciones del problema. Como nos pide la suma vectorial, debemos considerar que la longitud de ambos vectores sea la misma. Esto se hace con el comando *length*. Veamos cómo hay que usar este comando. Para esto, vamos a la ventana de comandos de Matlab (la ventana principal) y escribimos en el prompt: *help length*



El comando devuelve la longitud del vector dado como argumento. En este caso es simple, pero hay funciones que requieren más de un parámetro de entrada o devuelven más de un parámetro de salida, en cuyo caso es una buena costumbre recurrir al comando *help* para ver cómo hay que ingresar estos datos.

Entonces, el programa debe evaluar las longitudes de los vectores “a” y “b”. Si las longitudes son iguales procede a hacer la suma, caso contrario devuelve un error y termina la ejecución del programa. Esto puede hacerse con una sentencia *if*.

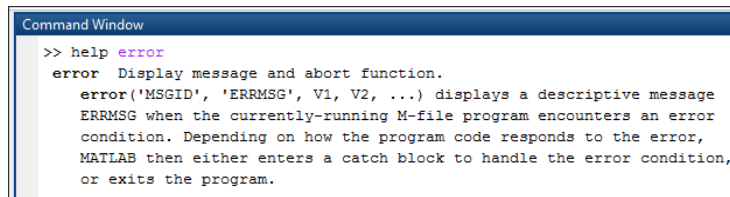


En la figura anterior puede verse el uso de comentarios. Los comentarios del programador son útiles para conocer qué es lo que pretende hacerse en cada punto del programa. Se hacen con el símbolo %. Todo lo que siga a este símbolo hasta el final de línea es ignorado por el compilador del programa, y por lo tanto, no se ejecuta ni se analiza.

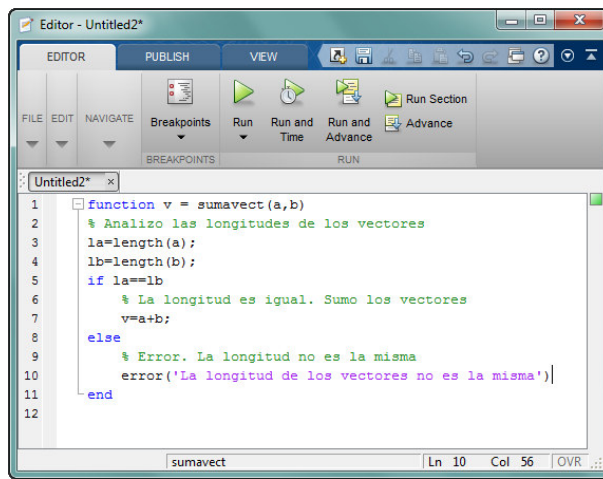
En las líneas 3 y 4 se calculan las longitudes de los vectores de entrada. Luego se evalúa si son iguales mediante el operador lógico “==” (*equal*). Este devuelve un “1” o “True” si los argumentos de entrada son iguales o un “0” o “False” en caso contrario.

Si los vectores son iguales, hago la suma vectorial de los mismos y la guardo en el vector de salida “v”.

Ahora veamos qué pasa si los vectores no tienen la misma longitud. El programa debe devolver un error y terminar su ejecución. Esto se logra con el comando “error”. Nuevamente recurrimos al help de Matlab

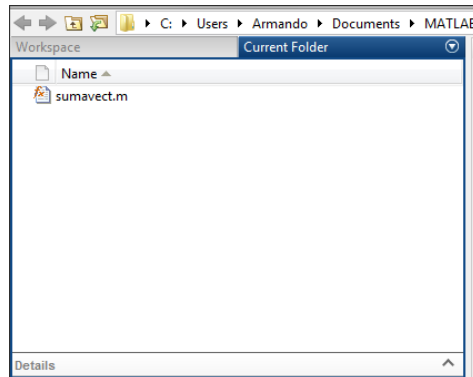


Para describir el error, debemos pasarle a la función error un string describiendo el mismo. El comando terminará la ejecución del programa y mostrará el string dado como argumento en la ventana de comandos

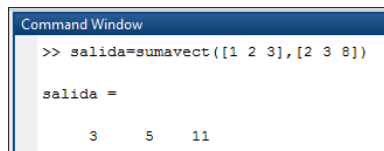


Con esto tendremos terminado nuestro programa. Para ejecutarlo es necesario guardarlo primero. El nombre del archivo debe ser el mismo que el de la función. MATLAB lo hace por defecto, pero siempre es responsabilidad del programador corroborar que los nombres sean iguales.

Una vez que lo guardamos nos fijamos que la función que acabamos de crear (“sumavect”) esté presente en el *Current Directory* de MATLAB.



Ahora procedemos a llamar la función desde la ventana de comandos de MATLAB. Para esto, debemos darle a la función los argumentos de entrada y guardar el resultado en la variable de salida. Los nombres de las variables no deben ser los mismos, pero sí hay que respetar el orden.



Pero si la longitud de los vectores no es la misma, el programa debe dar un error

