Uso de bucles en Matlab

Ismael Ignacio Mendoza
UNMSM

1. **Introducción**

   Una de las cosas que seguramente desearemos hacer al escribir un programa es repetir una operación o un conjunto de operaciones muchas veces hasta que una condición se satisfaga. Principalmente, hay dos tipos de estructuras básicas que se deben conocer. Una es el bucle **for** y la otra es la condición **if**.

2. **for**

   El bucle

   for [condición]

   [secuencia de comandos]

   end

   realiza toda la secuencia de comandos contenidas entre las líneas for y end siempre que la condición no se vea satisfecha. Una vez que la condición se satisface entonces MatLab salta a ejecutar todas las líneas de código posteriores a end. Un ejemplo sería un código con las siguientes líneas:

   %Este programa usa el comando for para generar una secuencia autorregresiva

   clear

   T = 50;

   k0 = 0;

   phi = 0.7;

   k(1) = k0;

   for t=2:T

   k(t)=phi*k(t-1)+randn;

   end

   plot(k)

   Este programa comienza con una breve documentación y el necesario comando clear. Después define una serie de parámetros que son la longitud de la secuencia T, el valor inicial de la serie k y un parámetro phi. Después asigna el valor de k0 al primer valor de la secuencia para entrar en el bucle for donde se le dice a MatLab que ejecute en el interior del bucle unos comandos siempre que t sea inferior a T, es decir, la línea que dice for t=2:T, quiere decir "desde t igual a 2 hasta T" haz lo que hay escrito hasta end. El comando end indica a MatLab hasta dónde tiene que hacer la evaluación para el valor de t. Una vez que llega a end MatLab acude a la línea for, suma una unidad a t, comprueba si t es T y, en caso de ser inferior, vuelve a ejecutar los comandos de dentro del bucle. Una vez que lo ha hecho tantas veces que t=T se verifica la condición y salta por debajo de end, encontrándose con el comando plot que indica a MatLab que debe sacar por pantalla un gráfico con la secuencia de k.

Los bucles for pueden estar anidados unos dentro de otros en cuyo caso los va resolviendo desde el más interior al más exterior. Por ejemplo en el siguiente programa construimos una matriz cuyos elementos aij dependen de la posición en la que el elemento se encuentra:

```
%Este programa ilustra otro uso del comando for
clear
n=10;
for i=1:n
for j=1:n
A(i,j)=i*j*abs(i-j);
end
end
mesh(A)
AZ = -50; EL = 45;
view(AZ, EL)
```

Como se puede ver este programa genera una matriz cuyos valores aij son más pequeños a medida que nos acercamos a la diagonal (con 0 exactamente en la diagonal), y más grande a medida que nos alejamos de ella. Para hacer una gráfico de una matriz ejecutamos el comando mesh que lo que hace es asignar en el espacio (x; y) los índices (i; j) y la elevación viene dada por el valor aij : El comando view toma dos argumentos, que son el ázimut y la elevación y modifican la perspectiva con la que se ve el gráfico. Como pueden ver queda un corazon. El resultado de anidar dos bucles for es el siguiente: el programa llega por primera vez a for i=1:n asignando a i el valor 1; a continuación se encuentra con for j=1:n asignando a j el valor 1, y opera A(i,j)=i*j*abs(i-j); (que es igual a 0). LLega al end del bucle j, y comprueba que j todavía no es n, por tanto a j le suma una unidad y por tanto j=2. Opera la expresión A(i,j)=i*j*abs(i-j); (que es igual a 2). MatLab hará esto hasta que el valor de j sea igual a n, entonces se encuentra con el end del bucle i. Compara el valor de i (que era igual a 1) con n, y al ver que i todavía no es n, le suma una unidad y se encuentra de nuevo (por segunda vez ahora) con el bucle for j=1:n. A j le asigna el valor 1 otra vez (pero ahora con i valiendo 2) y opera A(i,j)=i*j*abs(i-j); (que de nuevo es 2). Esto se hace hasta que i sea igual a n, provocando el salto hasta el end asociado al bucle de i.

3. **if**

La estructura

```
if [condición]
[secuencia de comandos]
else
[secuencia de comandos]
end
```

comprueba la primera condición y si el resultado de la condición es verdadero entonces ejecuta la secuencia de comandos que está entre los comandos if y else, en caso contrario (es falsa) ejecuta la secuencia de comandos que está entre else y end. Ejemplos:

```
%Este programa simula T lanzamientos de una moneda posiblemente desequilibrada.
clear
T = 1000;
equil = 0.5;
for i=1:T
a = rand;
if a<=equil
lanzamiento(i) = 0;
else
lanzamiento(i) = 1;
end
end
hist(lanzamiento)
frecunos = sum(lanzamiento)/T;
```

En este programa vemos varias cosas interesantes. En primer lugar dentro del bucle for hemos definido una variable a que toma un valor aleatorio en una distribución uniforme sobre el intervalo [0; 1]: En la estructura if decimos que si el valor de a es menor o igual al parámetro equil, que hemos fijado en 0:5 para hacer la moneda equilibrada pero que podía haber tomado otro valor distinto, entonces asigne a la variable lanzamiento el valor 0; y en caso contrario (a>equil) asigne el valor 1: Finalmente el programa realiza un histograma a través del comando hist (ver help hist) donde vemos las frecuencias relativas en un gráfico. Finalmente, inventamos la variable frecunos que es la suma de los componentes del vector lanzamiento dividido por el número de lanzamientos. Como el vector lanzamiento consta de ceros y unos, al pedir la suma (ejecutar »help sum) de los elementos del vector en realidad encontramos exáctamente el número de ocurrencias del evento 1 y, al dividirlo entre el número total de elementos T, encontramos la frecuencia con la que el uno apareció. Es conveniente notar dos cosas. La primera es que los nombres de las variables nos dicen algo sobre su naturaleza y sus nombres son largos. Imaginen el mismo programa en el que las variables en vez de llamarse equil, lanzamiento y frecunos se llamaran a, b, y c. El resultado es que los programas serían completamente ilegibles por un humano, cansarían al tener que estar recordando quién es quién y muestra un estilo de programación nefasto. Además pueden ver que los bucles anidados y las estructuras del programa están tabuladas, de modo que hay líneas de comandos que empiezan en la columna 1 de mi editor y otras que comienzan en la columna 7 y otras en la 14: Esto hace que los programas sean más fáciles de leer porque si además hacemos que un for comience en la misma columna que su correspondiente end, y hemos metido los comandos de dentro del bucle hacia adentro, entonces es muy fácil ver el alcance de los bucles for y de las estructuras condicionales if.