

### 3.2 Estructura repetitiva *for*

Una estructura repetitiva o bucle se utiliza cuando se quiere repetir un conjunto de sentencias un número determinado de veces o mientras se mantenga el cumplimiento de una condición.

El bucle `for` utiliza la primera opción usando una variable numérica capaz de controlar el número de iteraciones. Esta variable es conocida como variable de control.

Su sintaxis es la siguiente:

```
for variable = inicio:fin
    Sentencias
end
```

La utilización del operador (`:`) en la cabecera del bucle es la conocida hasta el momento. La variable de la cabecera va tomando sucesivamente cada uno de los valores asignados (no es un vector), y para cada uno de ellos se repiten las sentencias asociadas al bucle. En el caso anterior el incremento de la variable que controla el bucle es 1. Para incrementos distintos se debe utilizar:

```
for variable = inicio:incremento:fin
    Sentencias
end
```

También puede darse a la variable una sucesión de valores cualesquiera, sin necesidad de utilizar el operador (`:`). La variable que controla el bucle irá tomando cada uno de los valores indicados efectuando para cada uno una iteración.

```
for variable=[valor1,valor2,valor3,.....]
    Sentencias
end
```

Véanse a continuación unos ejemplos en los que se utilizan bucles ordinarios.

En el siguiente programa se inicializan al valor 1 las  $n$  primeras componentes de un vector.

```
n=input('Introduce un número natural');  
  
for i=1:n  
    v(i)=1;  
  
end
```

En el siguiente programa se crea la matriz de Hilbert de tamaño  $m \times n$ .

```
m=input('Introduce el número de filas');  
n=input('Introduce el número de columnas');  
  
for i=1:m  
    for j=1:n  
        A(i,j)=1/(i+j-1);  
    end  
  
end
```

El siguiente programa pide por teclado cada elemento de una matriz  $4 \times 3$  con un mensaje adecuado a cada uno.

```
for i=1:4  
    for j=1:3  
        fprintf('Elemento [%d,%d]\n',i,j);  
        A(i,j)=input('Introduce dato');  
    end  
  
end
```

En los dos últimos programas aparecen bucles anidados. Expliquemos con detalle el último de ellos. Para cada valor de la variable  $i$ , se ejecuta el bucle  $j$  para los valores 1, 2, 3. En cada iteración se genera un texto con `fprintf` que sirve como mensaje previo a la petición del dato en la sentencia posterior, en la que se lee un escalar que se guarda en  $A(i, j)$ . Por el orden en el que están dispuestos los bucles los datos leídos se guardan en el orden de las filas de  $A$ . El orden de lectura sería:  $A(1,1)$ ,  $A(1,2)$ ,  $A(1,3)$ ,  $A(2,1)$ ,  $A(2,2)$ ,  $A(2,3)$ ,  $A(3,1)$ ,  $A(3,2)$ ,  $A(3,3)$ ,  $A(4,1)$ ,  $A(4,2)$ ,  $A(4,3)$ .