Máster en Materiales y Sistemas Sensores para Tecnologías Medioambientales (Erasmus Mundus)

PRÁCTICAS DE CÁLCULO NUMÉRICO

Damián Ginestar Peiró

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA DEL DISEÑO

UNIVERSIDAD POLITÉCNICA DE VALENCIA

Práctica 1

Bucles y funciones

1.1. Bucles

La utilización de bucles permite realizar múltiples operaciones que se repiten, mediante instrucciones muy sencillas. El bucle más sencillo que se puede escribir es un bucle tipo `for` y tiene la forma

```
for i =1:n
      instrucciones
end
```

Por ejemplo, si se desea calcular la suma de las diez primeras potencias del número 2, es decir, si se pretende calcular

$$\sum_{i=1}^{10} 2^i,$$

se pueden utilizar las siguientes instrucciones:

```
suma=0;
for i=1:10
    suma=suma+2^i;
end
suma
```

Si lo que se pretende es calcular

$$\prod_{j=1}^{10} \frac{j}{j+1} \ ,$$

las instrucciones a utilizar serían

```
produc=1;
for j=1:10
   produc=produc*j/(j+1);
end
produc
```

En algunas ocasiones se necesita que el contador no varíe de uno en uno. Para ello, sólo tendremos que indicar el tamaño de paso en la instrucción **for.** Por ejemplo, las siguientes instrucciones

```
x=[ ];
for i=10:-1:1
      x=[x,i^2];
end
x
```

construyen un vector cuyas componentes son los cuadrados de los primeros diez números naturales en orden inverso.

Hay que tener en cuenta que MatLab dispone de operaciones matriciales y vectoriales optimizadas y hay que evitar el uso de bucles en lo posible, ya que hacen que los programas funcionen más lentamente.

Otro tipo de bucles utilizan el comando `while`. La estructura de estos bucles es la siguiente

```
while relacion
    instrucciones
end
```

Por ejemplo, si se quiere calcular el mayor número entero n tal que $2^n < 3000$, se puede hacer utilizando las siguientes instrucciones

```
n=0
while  2^n < 3000
    n=n+1;
end
n-1
```

Otra estructura útil es la estructura `if` que se utiliza del siguiente modo:

```
if  relacion
   instrucciones
end
```

Además se pueden hacer ramificaciones como se muestra en el siguiente ejemplo.

```
if  n<0
    paridad=0;
elseif rem(n,2)==0
  paridad=2;
else
  paridad=1;
end
```

Los operadores que permiten establecer relaciones son:

$<$, $>$, $<=$, $>=$, $==$,
$\sim =$ significa no igual,
& significa y
$\sim$ significa no
| significa o

Por ejemplo, la negación $A \sim = B$ donde A y B son matrices, sólo se cumplirá cuando todos los elementos de A sean distintos de los de B.

Otra estructura de interés es la estructura `switch`, que se utiliza como se muestra en el siguiente ejemplo:

```
switch method
case {1,2}
   disp('metodo lineal')
```

```
case 3:
   disp('metodo cubico}
otherwise:
   disp('metodo de orden superior')
end
```

1.2. Ficheros.m

Matlab dispone, en general de un editor que permite crear ficheros de texto. Estos ficheros suelen llevar la extensión **.m** y permite crear programas, que no son más que un conjunto de instrucciones y funciones. Para ejecutar estos programas bastará incluir el directorio donde se tiene el fichero en el **path** del MatLab, haciendo uso de la función `path()` y teclear el nombre del fichero sin extensión.

Las funciones tienen una estructura especial, como se muestra en el siguiente ejemplo:

```
function s=suma5(x)
% suma5 es una funcion que  suma 5 a  x
      s=x+5;
```

Esta estructura empieza con la palabra `function`. La frase comentada que hay en la línea siguiente a la cabecera es la que aparece al utilizarse el comando de ayuda, es decir, `help suma5`. Una vez definida la función, deberá guardarse en un fichero de nombre igual al nombre de la función con la extensión **.m**, para nuestro ejemplo el nombre del fichero sería `suma5.m`.

Otro ejemplo de función sencilla es el siguiente:

```
function [media,desv]=estad(x)
[m,n]=size(x);
if  m==1
m=n;
end
media =sum(x)/m;
desv =sqrt(sum(x.^2)/m-media^2);
```

Para escribir la desviación típica se ha tenido en cuenta que

$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})^2 = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \overline{x}^2.$$

Además el paquete MatLab tiene muchas funciones propias, tanto en el núcleo del programa como en una serie de librerias especializadas llamadas toolboxes, como ejemplo, mostramos algunos ejemplos de funciones que permiten escribir mensajes al usuario.

La instrucción `disp('mensaje')` muestra un mensaje al ususario.

La instrucción `error('mensaje')` devuelve un mensaje y para la ejecución de la función.

La instruccción

```
iter=input('introducir el numero de iteraciones')
```

muestra un mensaje y espera a que se introduzca un valor mediante el teclado que se asigna a la variable `iter`.

1.3. Ejercicios

1. Calcular, utilizando la instrucción `sum( )`

$$\sum_{i=1}^{10} 2^i \ .$$

Y, utilizando la instrucción `prod( )`

$$\prod_{j=1}^{10} \frac{j}{j+1} \ .$$

2. *a*) Escribir un bucle para calcular la suma de los cuadrados de los 25 primeros números naturales.

   b) Calcular el mayor número natural que satisface que

$$3^n + \ln(n) < 2000.$$

   c) Calcular el producto de los 10 primeros números impares

3. Construir una función que calcule la nota media del expediente académico de un alumno de primer curso, a partir de un vector donde estén almacenadas las notas.

4. Si suponemos una versión simplificada del sistema de suspensión de un automovil, el comportamiento entrada-salida, considerando el movimiento del cuerpo sólo en la dirección vertical, viene representado por la función de transferencia

$$F(s) = \frac{\frac{b}{m}s + \frac{k}{m}}{s^2 + \frac{b}{m}s + \frac{k}{m}} ,$$

donde b es el amortiguamiento, k la elasticidad del muelle y m la masa. Utilizando MatLab, definir esta función de transferencia. Calcular los valores de la función de transferencia si tomamos $b = 1$, $m = 2$, $k = 3$ y se hace variar la variable s de 0 a 100 de una unidad en una unidad, guardando el resultado en un vector.

5. Escribe un pequeño programa que pida un número por el teclado y compruebe su tamaño. Si el número es menor que 100, el programa debe escribir "numero pequeño", y si el número es mayor o igual que 100, el programa debe escribir algo apropiado. Introduce el código en un bucle para que se puedan introducir varios números, uno detrás de otro.

6. La siguiente expresión

$$\log(n!) \approx n\log(n) - n ,$$

constituye la aproximación de Stirling de $\log(n!)$. Escribe un programa para comprobar la validez de esta aproximación para $n = 100$, $n = 1000$ y $n = 5000$, teniendo en cuenta que 5000! es un número demasiado grande para poder ser tratado de forma standard por el ordenador.

7. La ecuación de Van der Waals para los gases es una generalización de la ecuación

$$PV = nRT ,$$

que tiene en cuenta la desviación del comportamiento ideal de los gases. Esta ecuación es

$$\left(P + \frac{n^2 a}{V^2}\right)(V - nb) = nRT ,$$

donde n es el número de moles del gas, P es la presión en (Pa), R es la constante de los gases (8.314 J/ (mol K)), y a y b son constantes que miden la desviación de la idealidad en el comportamiento del gas. Reordena la ecuación de forma que P se exprese como una función de V, y escribe un pequeño programa que genere una tabla con la presión correspondiente a 10 volúmenes igualmente espaciados entre V_1 y V_2. Las variables a, b, T, V_1 y V_2 han de introducirse por el teclado. Suponed $n = 1$.

8. Dada una matriz A $n \times m$, escribe una función que permute las filas k e i de la matriz A cuya llamada sea `B = permuta(A,k,i)`.

9. Calcula aproximadamente el valor de π utilizando el siguienete resultado

$$\frac{\pi^2 - 8}{16} = \sum_{n=1}^{\infty} \frac{1}{(2n-1)^2(2n+1)^2} \ .$$

   ¿ Cúantos términos de la serie hace falta sumar para obtener una precisión de $1\,10^{-12}$?.

10. Los números de Fibonacci se calculan haciendo uso de la relación

$$F_n = F_{n-1} + F_{n-2} \ ,$$

   con $F_0 = F_1 = 1$

   a) Calcula los 10 primeros números de Fibonacci.

   b) Para los 50 primeros números de Fibonacci calcula el cociente F_n/F_{n-1}. Este cociente se aproxima a valor del *número aureo* $(1+\sqrt{5})/2)$. ¿ Qué puedes decir de tus resultados?