Laboratorio de Algorítmica y **MATLAB**

Práctica 3ª: *Funciones en MATLAB. Variables lógicas. Estructuras repetitivas condicionales. Estructuras de ramificación.*

Febrero 2019

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

C. Conde, A. Fidalgo, R. Gómez, A. López, Mª Pilar Martínez de la Calle

Ejercicio: Interpolando resolviendo un sistema

Escribe un programa MATLAB en el que dado un vector de n abscisas distintas $\{s_1, s_2, \ldots, s_n\}$ y otro vector con n ordenadas $\{f_1, f_2, \ldots, f_n\}$ calcule los n coeficientes $\{c_1, c_2, \ldots, c_n\}$ del polinomio interpolador de Lagrange sobre el soporte de abscisas dado y con las ordenadas dadas, sabiendo que son solución del sistema de ecuaciones:

$$\begin{pmatrix} 1 & s_1 & \cdots & s_1^{n-1} \\ 1 & s_2 & \cdots & s_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & s_n & \cdots & s_n^{n-1} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

Verifica el funcionamiento del programa escrito con el vector de abscisas {0, 1, 2, 3, 4, 5} y el de ordenadas {0, 1, 0, 0, 0, 0}

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Ejercicio: Interpolando resolviendo un sistema

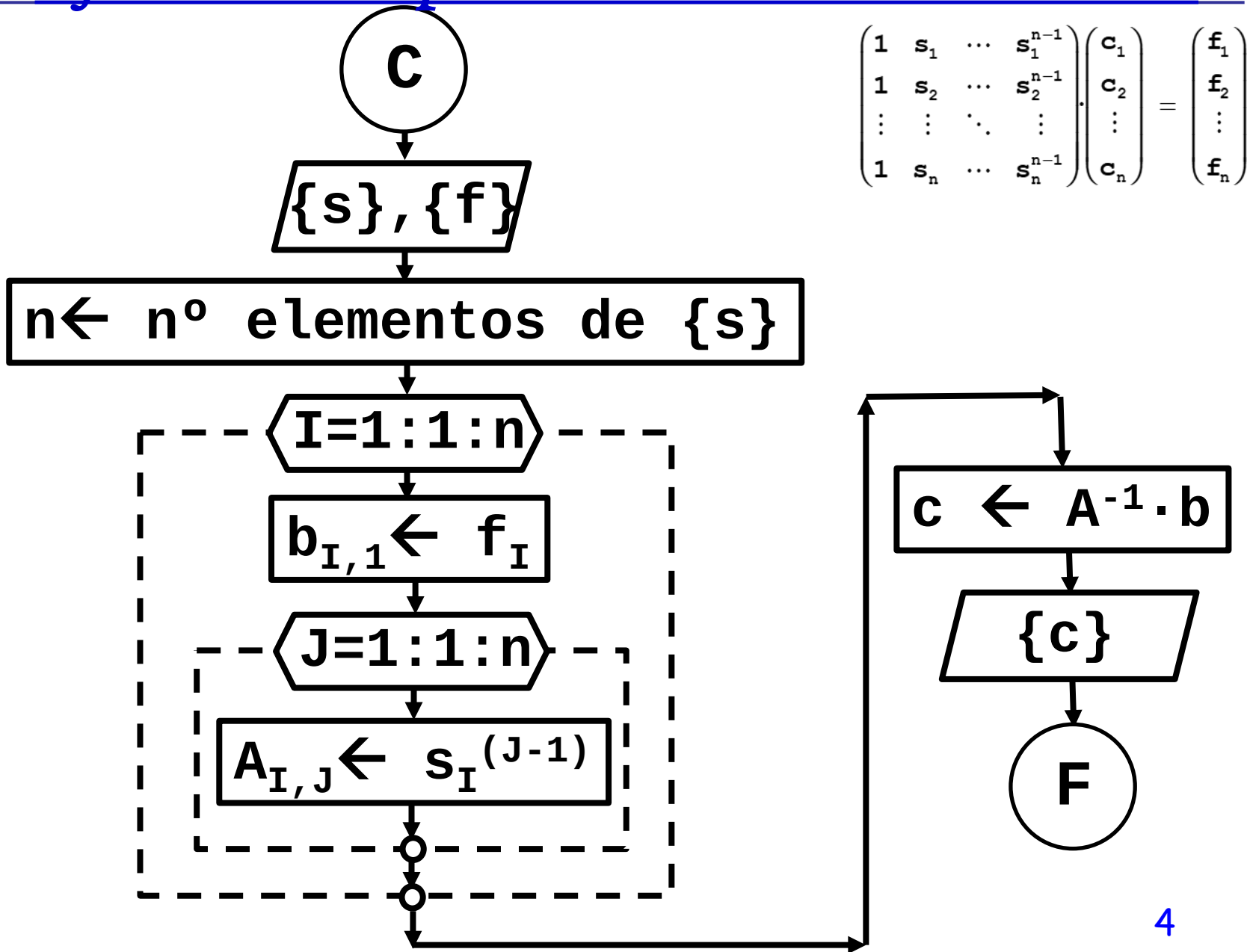
$$\begin{pmatrix} 1 & s_1 & \cdots & s_1^{n-1} \\ 1 & s_2 & \cdots & s_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & s_n & \cdots & s_n^{n-1} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$

```
INICIO
  Leer el vector s
  Leer el vector f
  n ← nº elementos de s
  Para I, desde 1, con paso 1, hasta n,
  hacer:
     b_{I,1} ← f_I
     Para J, desde 1, con paso 1,
     hasta n, hacer:
          A_{I,J} ← (s_I)^(J-1)
     Fin bucle en J
  Fin bucle en I
  c ← A^-1·b
  Escribir c
FIN
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Ejercicio: Interpolando resolviendo un sistema

$$\begin{pmatrix} 1 & s_1 & \cdots & s_1^{n-1} \\ 1 & s_2 & \cdots & s_2^{n-1} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & s_n & \cdots & s_n^{n-1} \end{pmatrix} \cdot \begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_n \end{pmatrix} = \begin{pmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{pmatrix}$$



Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Ejercicio: Interpolando resolviendo un sistema

```
clear all                    % LIMPIO MEMORIA
s=input('SOPORTE? ');% LEO DATOS
f=input('Valores en el soporte?');
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Ejercicio: Interpolando resolviendo un sistema

```
>> EP1_GITM18_19
SOPORTE? [0,1,2,3,4,5]
Valores en el soporte? [0,1,0,0,0,0]
   c =
             0
        5.0000
       -6.4167
        2.9583
       -0.5833
        0.0417
```

Vectores proporcionados por el usuario al ejecutar el programa

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Ejercicio: Interpolando resolviendo un sistema

Otra forma de programar lo mismo:

```
clear all                          % LIMPIO MEMORIA
s=input('SOPORTE? ');              % LEO DATOS
f=input('Valores en el soporte? ');
n=length(s); % EVALÚO nº ABSCISAS SOPORTE
S(1:n,1)=s(1:n); % ASEGURO S COMO COLUMNA
b(1:n,1)=f(1:n);                   % IDEM CON b
A(1:n,1)=1;          % INICIO 1ª COLUMNA DE A
for J=2:1:n        % PARA COLUMNAS 2 A N ...
    A=[A,A(:,J-1).*S]    % ... CONCATENO
   % LA ANTERIOR MULTIPLICADA ELEM. A ELEM.
end     % POR EL VECTOR COLUMNA DE ABSCISAS
c=inv(A)*b % RESUELVO A·c = b Y MUESTRO c
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1. Las funciones en MATLAB

C. Conde, A. Fidalgo, R. Gómez, A. López, Mª Pilar Martínez de la Calle

1. Las funciones en MATLAB

En **MATLAB** una función es un ***programa independiente***, que puede ser utilizado desde **Command window** o en otros programas, asignando valor a uno o más **argumentos de entrada** (*variables independientes*) para calcular el valor de uno o más **argumentos de salida** (*variables dependientes*).

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1. Las funciones en MATLAB

En **MATLAB** una función es un **SUBPROGRAMA**, que puede ser utilizado desde **Command window** o en otros programas, asignando valor a uno o más **argumentos de entrada** (*variables independientes*) para calcular el valor de uno o más **argumentos de salida** (*variables dependientes*).

Numerosas funciones forman parte de la plataforma MATLAB. Se utilizan escribiendo su nombre y tras él, encerradas entre paréntesis, los valores (o los nombres de variables) que dan valor a los argumentos de entrada que se utilizan.

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1. Las funciones en MATLAB

EJEMPLOS:

1º) **MATLAB** incorpora las funciones trigonométricas. Entre ellas la función **coseno**, que tiene asignado el nombre **cos** y que utiliza un único argumento de entrada. En cualquier momento el usuario puede escribir:

```
>> v=cos(2.56)
```

En ese momento **MATLAB** ejecuta las instrucciones que evalúan el desarrollo en serie del coseno asignando al **argumento de entrada** el **parámetro de entrada** dado por el valor **2.56** y, una vez calculado, el valor correspondiente se almacena en el **argumento de salida**, que a su vez asigna ese valor al **parámetro de salida** (aquí llamado variable **v**).

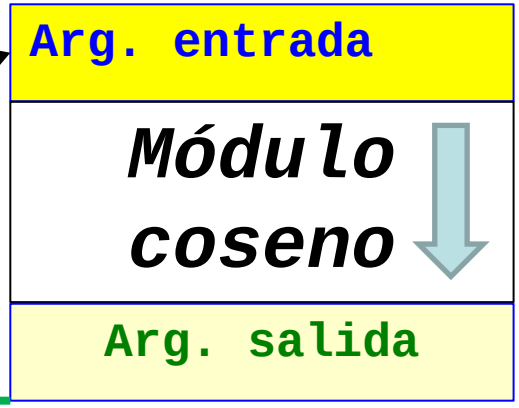
Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1. Las funciones en MATLAB

PROGRAMA

```
A= input('A');
B=log(A)+A^2;
C= cos(B)
D=max(abs(C),A/4)
```

MÓDULO



1º) El **argumento de entrada** *del módulo coseno toma el valor que tenga el* **parámetro de entrada** *(variable* **B***)*

2º) Se calcula el valor del **coseno** *almacenando su resultado en el* **argumento de salida**.

3º) El valor del **argumento de salida** *se asigna a la variable* **parámetro de salida** *(variable* **C***)*

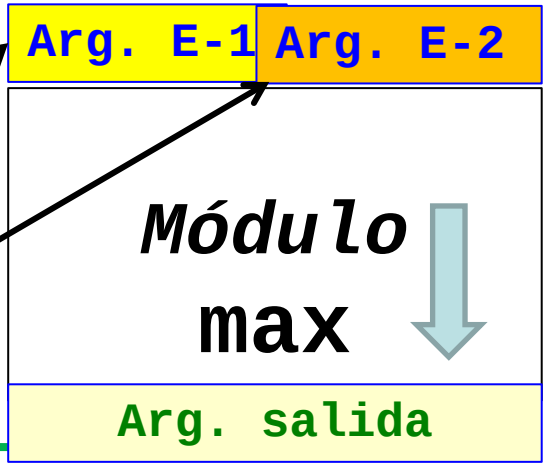
Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1. Las funciones en MATLAB

PROGRAMA

```
A= input('A');
B=log(A)+A^2;
C= cos(B)
D=max(abs(C),A/4)
```

MÓDULO



1º) El **argumento de entrada 1º** *del módulo* **max** *toma el valor* **|C|** *y el* **argumento de entrada 2º** *el de* **A/4**.

2º) Se asigna al **argumento de salida** *del módulo* **max** *el mayor de los valores de los* **argumentos de entrada**.

3º) El valor del **argumento de salida** *se asigna al* **parámetro de salida** *(la variable* **D***).*

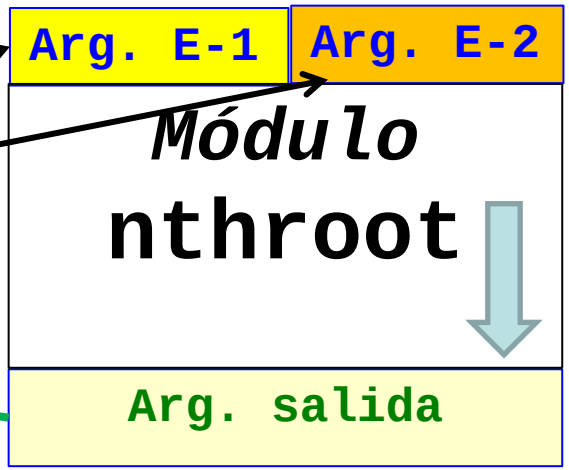
Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1. Las funciones en MATLAB

PROGRAMA

```
A= input('A');
N=3
R=nthroot(A,N);
```

MÓDULO



1º) El **argumento de entrada** *1 del módulo* **nthroot** *toma el valor* **A** *y el* **argumento de entrada 2º** *el de* **N**.

2º) Se asigna al argumento de salida del módulo **nthroot** *el valor de la raíz real* **N**-*ésima de* **A**.

3º) El valor del **argumento de salida** *se asigna al* **parámetro de salida** *(la variable* **R***).*

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1. Las funciones en MATLAB

ALGUNA OBSERVACIÓN:

1ª) Las variables que dan valor a los **argumentos de entrada** no deben ver modificado su valor tras ejecutarse la función.

2ª) El usuario no tiene por qué conocer el nombre dado a los **argumentos de entrada** ni a los de **salida**, ni tampoco el proceso que se realiza en el módulo. "*Lo que pase en el módulo se queda en el módulo*", sin transmitirse al programa salvo los valores de los **argumentos de salida**. **Se utilizan variables locales**.

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1. Las funciones en MATLAB

MATLAB permite el uso de **funciones definidas por el usuario** de 4 maneras distintas:

1ª) Escribiéndolas en una única sentencia (funciones handle).

2ª) Escribiéndolas como un subprograma function (En un script DIFERENTE)

3ª) Funciones "especiales": polinomios *o definidas por tramos polinómicos, …*

4ª) Utilizando variables simbólicas previamente declaradas como tales. (Más adelante)

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.1. Las funciones handle

Se denominan **funciones handle** a aquellas que no son escritas en ficheros diferenciados ni están predefinidas en **MATLAB**.

Son funciones "sencillas" que se definen con una sola instrucción en el espacio de trabajo. Se las denomina también *funciones anónimas y funciones online*.

Su **estructura** es la siguiente:

```
nombre = @(argumentos) expresión
```

Se **utilizan** escribiendo el **nombre** y, entre paréntesis y separados por comas, los valores que se quiera asignar a los argumentos.

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

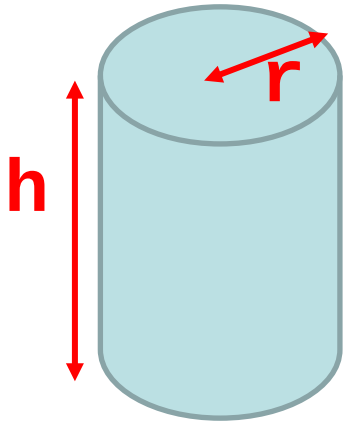
1.1. Las funciones handle

EJEMPLO: Función calculando el volumen cilíndrico

```
>> fun1=@(r,h) h*pi*r^2;

>> v=fun1(1.56,2)
```

$2 \cdot \pi \cdot 1.56^2$




```
>> fun1=@(r,h) h*pi*r^2;

>> radio=3; altura=4;
>> volumen=fun1(radio,altura)
```

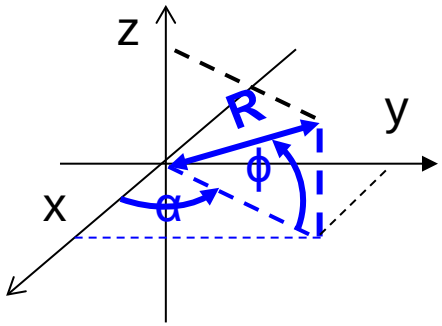
$4 \cdot \pi \cdot 3^2$

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.1. Las funciones handle

EJEMPLO:

Función que transforma coordenadas esféricas a cartesianas




```
>> esf2car=@(R,alpha,phi) ...
          [R*cos(alpha)*cos(phi),...
           R*sin(alpha)*cos(phi),...
           R*          sin(phi)]

>>C=esf2car(2,-pi/3,-pi/4)
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.1. Las funciones handle

OBSERVACIONES:

1ª) Las funciones handle sólo se pueden utilizar mientras su nombre está activo en **`Workspace`**. Tras borrar la memoria (p.ej.: con **`clear`**) es necesario volver a definir la función antes de usarla

2ª) Las variables utilizadas en la definición son **`variables locales`** (su valor no aparece en **`Workspace`**).

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

Otra forma de desarrollar funciones consiste en escribir un script con la siguiente estructura

```
function [ARG_SAL.]=nombre(ARG_ENT.)
  % Comentarios (opcionales)
   Instrucciones (se recomienda
      terminarlas en ";" para evitar
      "ecos" en pantalla)
end (opcional)
```

con: **[ARG_SAL.]** lista de **argumentos de salida** separados por comas "," y entre **corchetes** y: **(ARG_ENT.)** lista de **argumentos de entrada** separados por comas "," y entre **paréntesis**

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

Una vez escrito un script **function** se debe salvar en el directorio de trabajo en un fichero **con el mismo nombre que la función** y extensión **.m**.

Para utilizar un script **function** debes:

1º) Asegurarte de que el fichero que la contiene está en el directorio de trabajo (aparece en la ventana **Current folder**).

y:

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

2º) Escribir

[PARÁMETROS SALIDA]=nombre(PARÁMETROS ENT.)

con: **[PARÁMETROS SALIDA]** lista de variables, encerradas entre corchetes y separadas por comas “,””, a las que la función transferirá los valores de los *argumentos de salida*. Debe haber tantos parámetros como argumentos

y: **(PARÁMETROS ENT.)** lista de variables y/o constantes, encerradas entre paréntesis y separadas por comas “,””, de las que la función tomará los valores de los *argumentos de entrada*. Debe haber tantos parámetros como argumentos

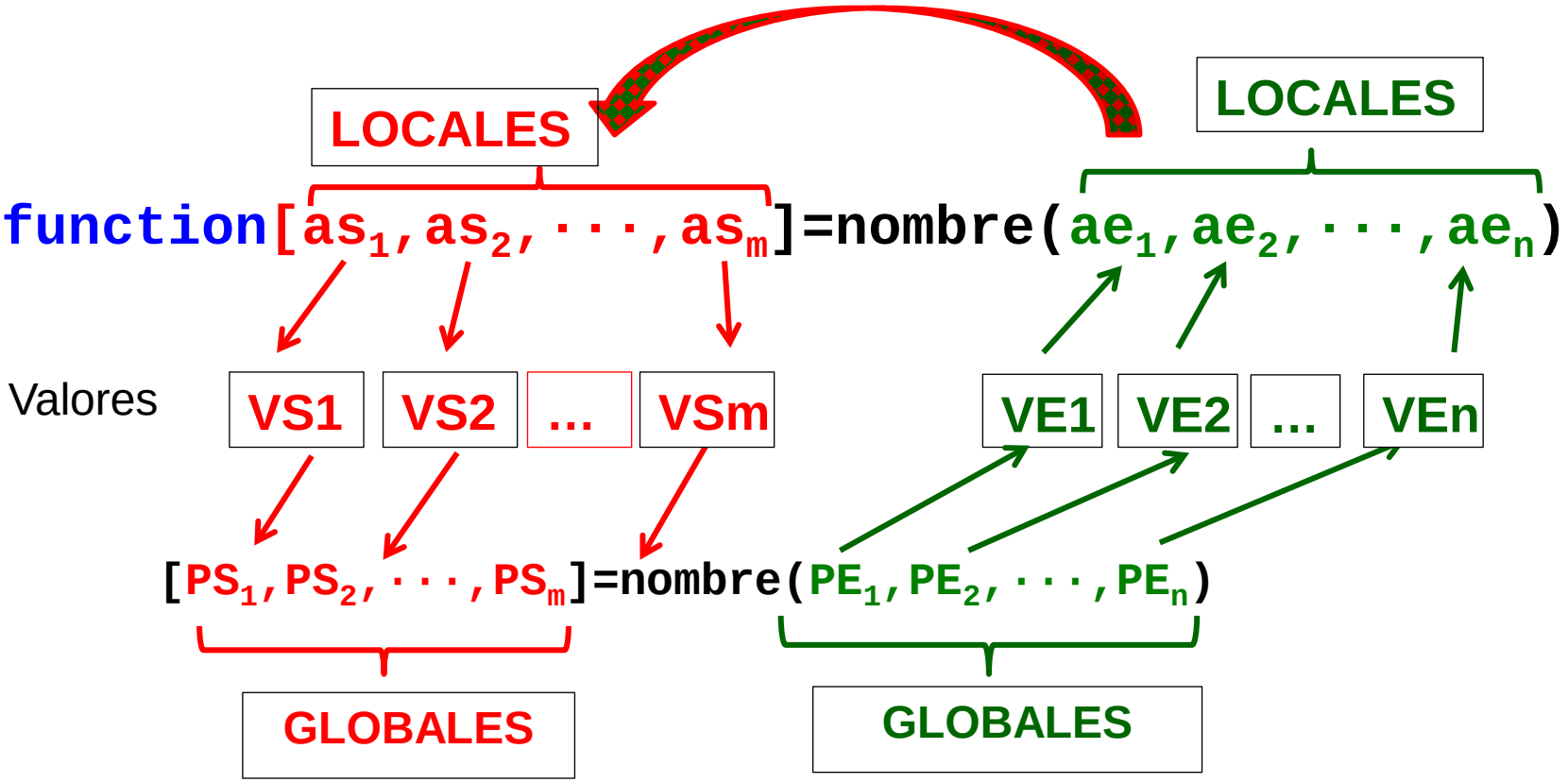
Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas `function`

- Los **argumentos de entrada y de salida son nombres de variables** simples, vectores y/o matrices (según les den valor los parámetros de entrada). .
- Las instrucciones pueden ser cualesquiera, y en ellas pueden emplearse otras variables.
- Todas **las variables usadas en una función son, por defecto, locales**, es decir no aparecen en la ventana **`Workspace`** y no se "mezclan" con las de los programas que las utilicen aunque tengan el mismo nombre.
- No utilices **`clear`** o similares instrucciones.

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function



Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

EJEMPLO: 1º) Construyamos un subprograma function, llamado CCPIL (Coeficientes Canónicos del Polinomio Interpolador de Lagrange) , que usando como argumentos de entrada un vector de abscisas de soporte (s) y un vector de valores (f) ambos con el mismo número de elementos, calcule como argumento de salida el vector de coeficientes del polinomio interpolador de Lagrange

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas `function`

2º Escribir un programa en MATLAB que:

2-1) Solicite al usuario un vector de abscisas de soporte (SOP) y otro con valores de una función (VAL).

2-2) Calcule los coeficientes del polinomio interpolador usando el subprograma `CCPIL`.

2-3) Construya un vector con 201 abscisas uniformemente repartidas entre el menor y el mayor valor de los elementos del vector `SOP`.

2-4) Represente gráficamente el polinomio interpolador.

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

~~clear all~~ No borramos (pues perderíamos argumentos)

~~s=input('SOPORTE? '); % LEO DATOS~~

~~f=input('Valores en el soporte? ');~~

Los "datos" serán los argumentos. No se leen

```
n=length(s); % EVALÚO nº ABSCISAS SOPORTE
S(1:n,1)=s(1:n); % ASEGURO S COMO COLUMNA
b(1:n,1)=F(1:n); % IDEM CON b
A(1:n,1)=1;      % INICIO 1ª COLUMNA DE A
for J=2:1:n % Para J desde 2 hasta n ...
    A=[A,A(:,J-1).*S] % ...COLUMNA
                      % J-ésima DE A
end
c=inv(A)*b % RESUELVO A·c = b Y MUESTRO c
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

```
function [c]=CPIL(s,f)
n=length(s); % EVALÚO nº ABSCISAS SOPORTE
S(1:n,1)=s(1:n); % ASEGURO S COMO COLUMNA
b(1:n,1)=F(1:n); % IDEM CON b
A(1:n,1)=1;      % INICIO 1ª COLUMNA DE A
for J=2:1:n % Para J desde 2 hasta n ...
    A=[A,A(:,J-1).*S] % ...COLUMNA
                      % J-ésima DE A
end
c=inv(A)*b % RESUELVO A·c = b Y MUESTRO c
end
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

```
function [c]=CCPIL(s,f)% DECLARO LA FUNCIÓN
  n=length(s);% EVALÚO nº ABSCISAS SOPORTE
  S(1:n,1)=s(1:n); % ASEGURO S COLUMNA
  b(1:n,1)=F(1:n); % IDEM CON b
  A(1:n,1)=1; % INICIO 1ª COLUMNA DE A
  for J=2:1:n % Para J de 2 a n ...
    A=[A,A(:,J-1).*S] % ...COLUMNA
                      % J-ésima DE A
  end
  c=inv(A)*b        % RESUELVO A·c = b
end                 % FINALIZO EL SUBPROGRAMA
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

```
clear all                    % LIMPIO MEMORIA
SOP=input('SOPORTE? '); % LEO DATOS
VAL=input('Valores en el soporte? ');
N=length(SOP); % EVALÚO nº ABSC. SOPORTE
[COEF]=CCPIL(SOP,VAL); % CALCULO COEFS
% EN LA SENTENCIA SIGUIENTE CALCULO
% 201 ABSC. UNIFORMEMENTE REPARTIDAS
% ENTRE EL MENOR Y MAYOR VALOR DEL
% SOPORTE
X=linspace(min(SOP),max(SOP),201);
```

(CONTINÚA EN PÁG SIG.)

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

(VIENE DE PÁG ANT.)

```
AUX=X; % EN AUX ALMACENO LAS ABSCISAS DE
% DIBUJO. A CONTINUACIÓN INICIALIZO VALORES
% DEL POLINOMIO (yJ = c1 + c2·xJ)
Y(1:201)=COEF(1)+COEF(2)*X;
for J=3:1:N
    AUX=AUX.*X; % MULTIPLICO AUX POR
    % ABSCISAS Y SE LO SUMO AL VALOR
    % MULTIPLICADO POR EL COEFICIENTE
    % CORRRESPONDIENTE
    Y=Y+COEF(J)*AUX;
end
plot(X,Y,'-b') % DIBUJO
grid on
title('TERCER POLINOMIO DE BASE DE LAGRANGE')
```

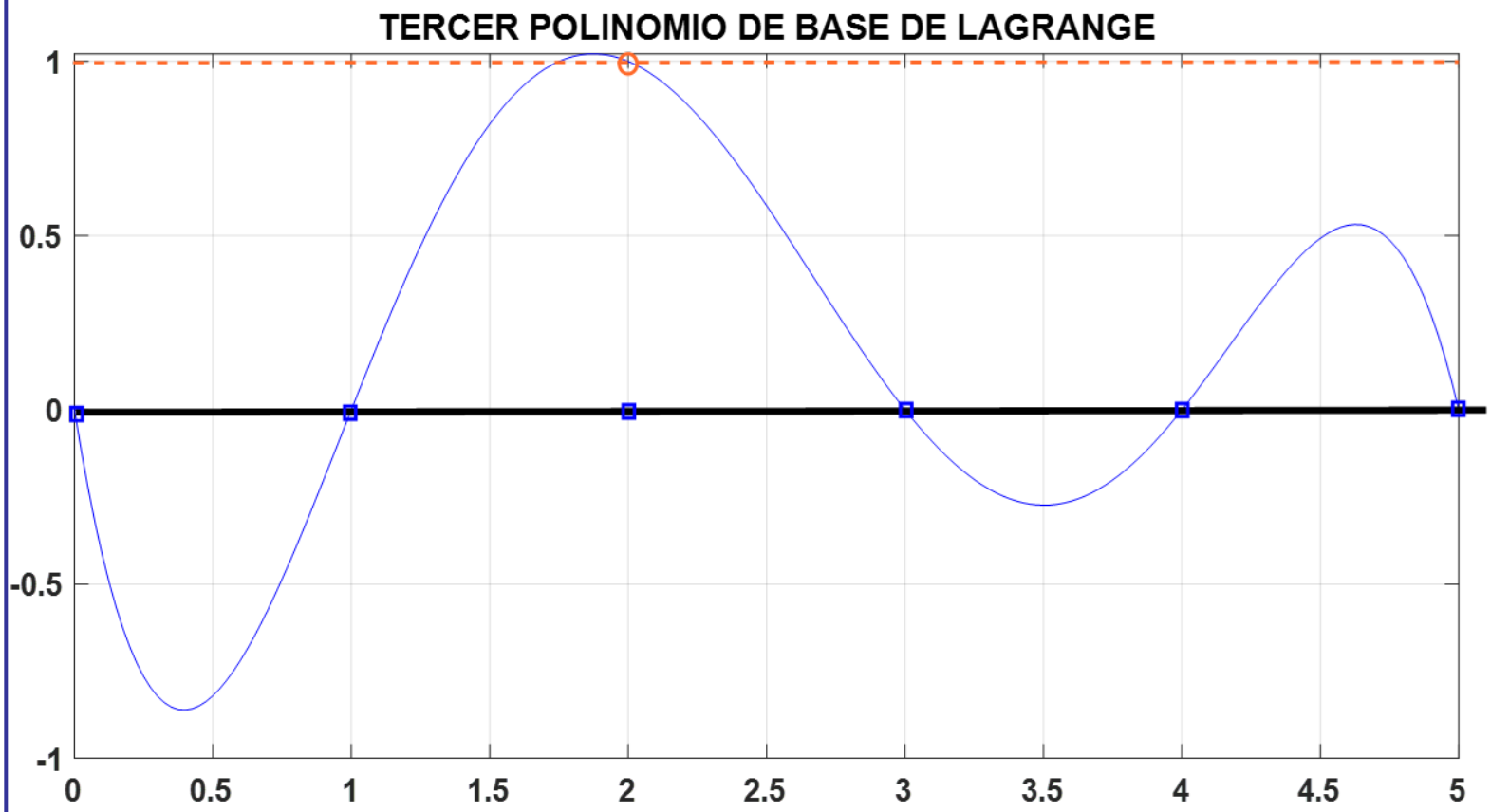
Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function

```
>> PR_INTERP_SIST
 SOPORTE? [0,1,2,3,4,5]
 Valores en el soporte? [0,0,1,0,0,0]
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

1.2. Los subprogramas function



Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2. Variables lógicas Operadores de relación y Operadores lógicos

C. Conde, A. Fidalgo, R. Gómez, A. López, Mª Pilar Martínez de la Calle

2.1. Variables Lógicas

VARIABLES LÓGICAS (O BOOLEANAS):

Son aquellas que sólo pueden tomar los valores `true` o `false`.

Estos valores en MATLAB se representan por:

`1` (`true`) y `0` (`false`)

```
>> alpha=true,   beta=false

   alpha =
        1                    (true)
   beta =
        0                    (false)
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2.2. Operadores de Relación

Los **OPERADORES DE RELACIÓN** (también llamados ***OPERADORES DE COMPARACIÓN***) permiten **comparar** dos a dos, los valores de variables, constantes y resultados de expresiones, produciendo un **resultado lógico**.

Comparación	*Operador*	*Comparación*	*Operador*
Ser igual:	==	Ser distinto:	~=
Ser menor que:	<	Ser menor o igual que:	<=
Ser mayor que:	>	Ser mayor o igual que:	>=

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2.2. Operadores de Relación

```
>> a=sqrt(17.35*12); b=11;

>> a>=b
      ans =
                    1
```

true

```
>> a<2*b
    ans =
                    1
```

true

```
>> b-a>0
    ans =
                    0
```

false

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2.2. Operadores de Relación

Cuando se comparan variables alfanuméricas se sigue el código UNICODE, por lo que:

- Los caracteres de dígitos numéricos son menores que las letras

```
>> '9'<'a'
   ans =
        1
```

* Las mayúsculas son menores que las minúsculas

```
>> 'a' > ' A'
ans =
      1
```

true

```
>> 'Z' < 'a'
ans =
      1
```

true

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2.2. Operadores de Relación

Cuando se comparan variables alfanuméricas se sigue el código UNICODE, por lo que:

- Entre dígitos numéricos de menor a mayor según valor

```
>> '9'< '6'
    ans =
              0
```

false

* Entre letras del mismo tipo se sigue orden alfabético

```
>> 'a'>'b'
   ans =
          0
>> 'J'< 'S'
   ans =
          1
```

false

true

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2.3. Operadores Lógicos

Las variables y constantes lógicas (como son los resultados de operaciones de relación) se pueden operar mediante los **operadores lógicos**, construyendo **EXPRESIONES LÓGICAS** cuyo resultado es un valor lógico.

Los **operadores lógicos** en MATLAB son:

&
(May+6)

(Operador AND, o "y lógico"): el resultado de su aplicación es **1** si ambos operandos son **1** y es **0** en caso contrario.

|
(Alt Gr+1)

(Operador OR, u "o lógico"): el resultado de su aplicación es **0** si ambos operandos son **0** y es **1** en caso contrario.

~
(Alt Gr+4)

(Operador NOT, o "negación lógica"): el resultado de su aplicación a un único operando es el contra-rio a su valor.

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2.3. Operadores Lógicos

AND (&)

A	B	A & B
T	T	T
T	F	F
F	T	F
F	F	F

OR (|)

A	B	A | B
T	T	T
T	F	T
F	T	T
F	F	F

NOT (~)

A	~A
T	F
F	T

```
>> a=5; b=-2; c=4; d=-2;
>> ((a*b>c*d)&(a>=d))
      false     true
   ans =
              0   (false)
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2.3. Operadores Lógicos

AND (&)

A	B	A & B
T	T	T
T	F	F
F	T	F
F	F	F

OR (|)

A	B	A | B
T	T	T
T	F	T
F	T	T
F	F	F

NOT (~)

A	~A
T	F
F	T

```
>> a=5; b=-2; c=4; d=-2;
>> ((a*b>c*d)|(a>=d))
      false     true
   ans =
              1   (true)
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2.3. Operadores Lógicos

AND (&)

A	B	A & B
T	T	T
T	F	F
F	T	F
F	F	F

OR (|)

A	B	A | B
T	T	T
T	F	T
F	T	T
F	F	F

NOT (~)

A	~A
T	F
F	T

```
>> a=5; b=-2; c=4; d=-2;
>> ((a*b>c*d)|(a<=d))
        false    false
   ans =
          0        (false)
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

2.3. Operadores Lógicos

AND (&)

A	B	A & B
T	T	T
T	F	F
F	T	F
F	F	F

OR (|)

A	B	A | B
T	T	T
T	F	T
F	T	T
F	F	F

NOT (~)

A	~A
T	F
F	T

```
>> a=5; b=-2; c=4; d=-2;
>> ~(a>0)
      true
   ans =
            0   (false)
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Operadores de Relación y Lógicos

RESUMEN

OPERADORES DE RELACIÓN

<	menor que
<=	menor o igual que
>	mayor que
>=	mayor o igual que
==	igual que
~=	distinto que

OPERADORES LÓGICOS

&	y / and
|	o / or
~	no / not

NOT

A	Not A	When A is		The Resultant is
T	F	NOT TRUE	IS	FALSE
F	T	NOT FALSE	IS	TRUE

AND

A	B	A AND B	When A is		When B is		The Resultant is
T	T	T	TRUE	AND	TRUE	IS	TRUE
T	F	F	TRUE	AND	FALSE	IS	FALSE
F	T	F	FALSE	AND	TRUE	IS	FALSE
F	F	F	FALSE	AND	FALSE	IS	FALSE

OR

A	B	A OR B	When A is		When B is		The Resultant is
T	T	T	TRUE	OR	TRUE	IS	TRUE
T	F	T	TRUE	OR	FALSE	IS	TRUE
F	T	T	FALSE	OR	TRUE	IS	TRUE
F	F	F	FALSE	OR	FALSE	IS	FALSE

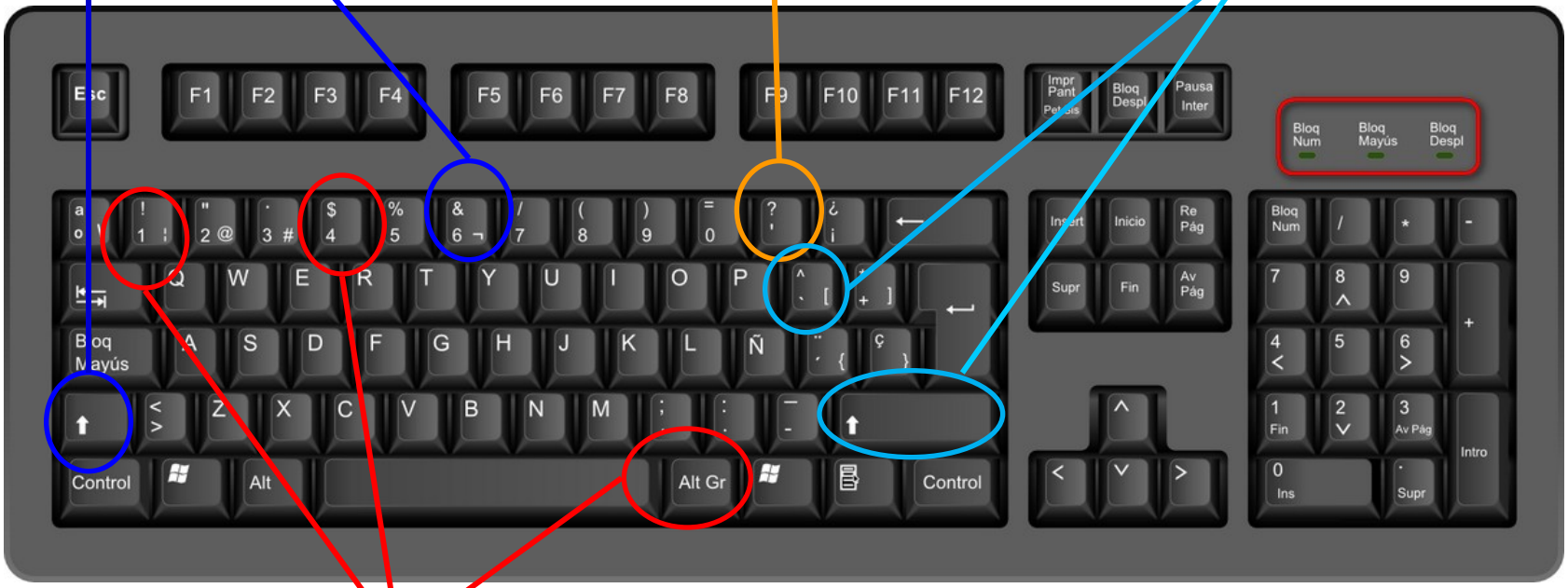
Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Operadores de Relación y Lógicos

May 6: **&** “and”

' comilla simple

May ^: **^** “exponente”



Alt Gr 1: **|** barra vertical “or”

Alt Gr 4: **~** “not”

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Operadores de Relación y Lógicos

Aplicación a matrices con el mismo número de filas y columnas

Los operadores de relación, generan una matriz de elementos lógicos operando elemento a elemento

```
>> A=[1,-2,3;-4,5,-6;7,-8,9]
   B=[1, 2,3; 4,5, 6;7, 8,9]
A =                                B =
     1    -2     3                     1     2     3
    -4     5    -6                     4     5     6
     7    -8     9                     7     8     9
>> B>A
ans =
     0     1     0
     1     0     1
     0     1     0
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Operadores de Relación y Lógicos

Aplicación a matrices con el mismo número de filas y columnas

Los operadores de relación, generan una matriz de elementos lógicos operando elemento a elemento

```
>> A=[1,-2,3;-4,5,-6;7,-8,9]
   B=[1, 2,3; 4,5, 6;7, 8,9]
A =                                B =
     1    -2     3                   1     2     3
    -4     5    -6                   4     5     6
     7    -8     9                   7     8     9
>> ((B-A)<B)&((A+B)>=0)
ans =
     1     0     1
     0     1     0
     1     0     1
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles Condicionales

C. Conde, A. Fidalgo, R. Gómez, A. López, Mª Pilar Martínez de la Calle

3. Los Bucles Condicionales

Los bucles condicionales son estructuras algorítmicas en las que se repite un grupo de instrucciones mientras una expresión de relación o lógica tenga como resultado `1` (`true`) y dejan de ejecutarse cuando el resultado sea `0` (`false`).

ATENCIÓN: Entre las instrucciones a realizar debe haber alguna que modifique el resultado de la expresión de relación o lógica *(Para que el bucle no se repita eternamente)*

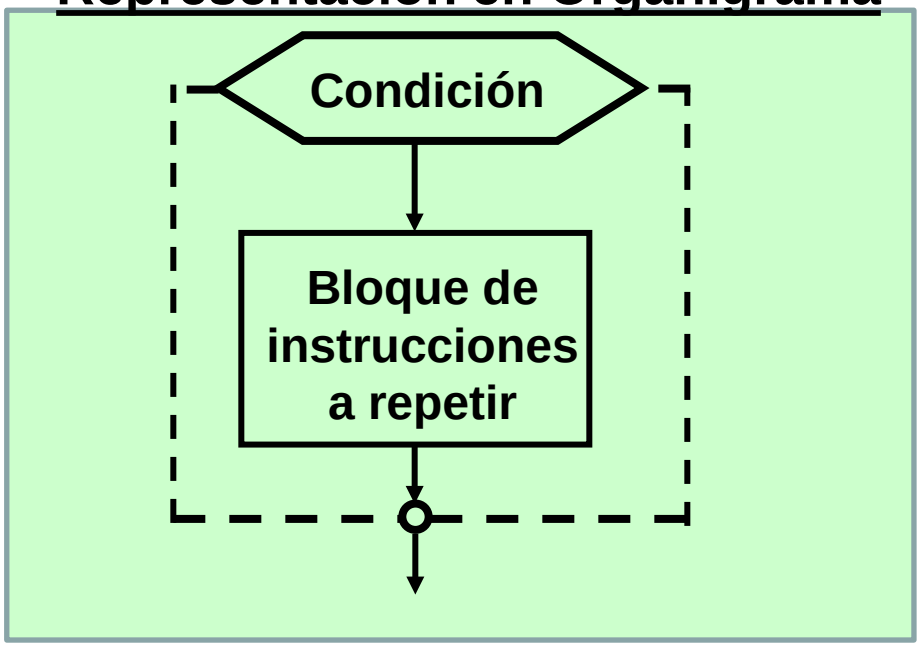
Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles Condicionales

Representación en pseudocódigo

```
Mientras (expresión de relación o
    lógica) hacer
 Grupo de sentencias que se repiten
Fin del bucle
```

Representación en Organigrama



Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles Condicionales

Bucle condicional en MATLAB:

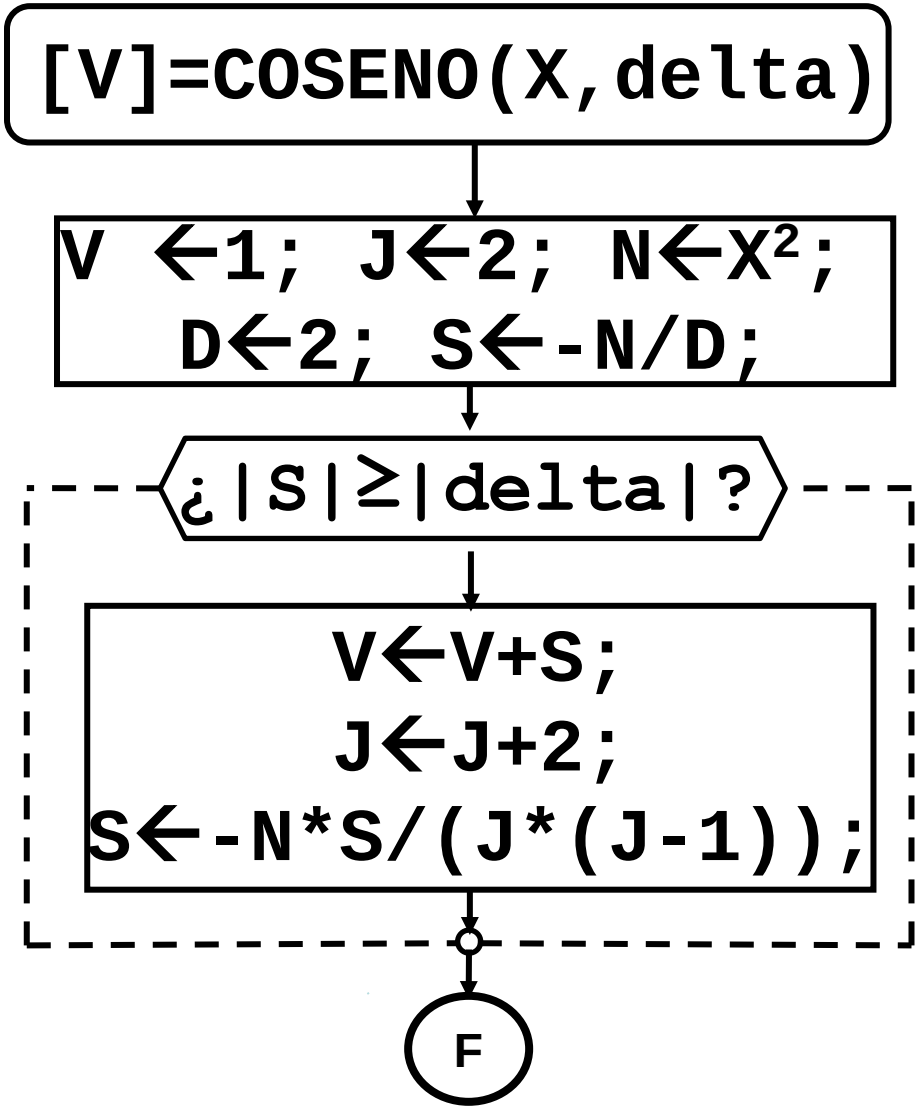
```
> while (condición)
      Sentencias MATLAB que se repiten
  end
```

Ejercicio: *Escribir un algoritmo y una función MATLAB llamada* **COSENO** *que, teniendo por argumentos de entrada una variable* (**X**) *y otra variable* (**delta**) *sume los primeros términos del desarrollo de Taylor de cos(X) hasta obtener dos resultados que difieran en menos de* **|delta|**.

$$\cos(X) = 1 - \frac{X^2}{2!} + \frac{X^4}{4!} - \frac{X^6}{6!} + \ldots = \sum_{J=0}^{M} \frac{(-1)^J \cdot x^{(2 \cdot J)}}{(2 \cdot J)!}$$

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles Condicionales



Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles Condicionales

```
function [V]=COSENO(X,delta)
% FUNCIÓN PARA APROXIMAR EL VALOR (V) DEL
% COSENO DE UN ÁNGULO (X) CON PRECISIÓN
% DADA POR delta
   V=1; J=2; N=X*X; D=2; S=-N/D;
   adelta=abs(delta);
   while ( abs(S) >= adelta )
       V=V+S
       J=J+2;
       S=-N*S/(J*(J-1));
   end
end
```

```
save as  COSENO.m
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles Condicionales

```
>> vcos=COSENO(pi,10^(-6));
      V =-3.9348
      V = 0.1239
      V =-1.2114
      V =-0.9760
      V =-1.0018
      V =-0.9999
      V =-1.0000
      V =-1.0000

>> format long
>> vcos
      vcos =-0.999999864739555
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles con variable de control indicada

También pueden definirse bucles, similares a los determinados, en los que **se especifican, uno a uno, los valores que debe tomar la variable de control**.

Su representación en **pseudocódigo** es:

```
Para Vble. Control = [v1, v2, …, vn], hacer:
    Grupo de sentencias que se repiten.
Fin bucle
```

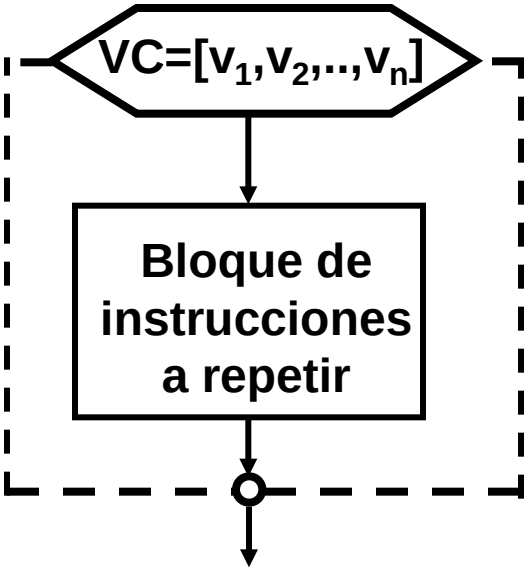
La variable de control toma el valor v_1 y se ejecuta el grupo de sentencias que se repiten. Tras ello, la variable de control toma el valor v_2 y se vuelve a repetir el grupo de sentencias. Así sucesivamente hasta que la variable de control toma el valor v_n, se ejecuta el grupo de sentencias y se finaliza el bucle continuando el algoritmo.

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles con variable de control indicada

<u>OBSERVACIÓN</u>: El conjunto de valores de la variable de control puede estar almacenado en un vector.

Su representación mediante **ORGANIGRAMA** es:



Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles con variable de control indicada

En **MATLAB** estos bucles se escriben en la forma:

```
for vble.control = vector fila
      Grupo de sentencias a repetir
end
```

El **`vector fila`** *puede especificarse elemento a elemento tras el signo igual (=), o puede ser el nombre de un vector fila previamente definido.*

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles con variable de control indicada

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

EJERCICIO

Dada una matriz **A**, de dimensiones (n, n), con n $\geq$ 6, se quieren elevar al cuadrado sus elementos (2,2), (2, 4), (2,5), (4, 2), (4,4), (4, 5), (5,2), (5,4) y (5,5) (**es decir los elementos que están en las filas 2, 4 y 5 y en las columnas 2, 4 y 5**). Escríbase un subprograma **function MATLAB** que realice esta tarea, almacenando la matriz resultante en una matriz llamada **B**.

3. Los Bucles con variable de control indicada

```
function [B]=ejemploX(A)
% Esta función eleva al cuadrado
% los elementos de una matriz A
% que estén en la fila y la columna
% especificada por alguno de los
% elementos del vector v dado más
% abajo, devolviendo el resultado
% en la matriz B.
   v=[2,4,5];
   B=A;
   for I=v
       for J=v
           B(I,J)=A(I,J)^2;
       end
   end
end
```

save as ejemploX.m

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles con variable de control indicada

```
>> MATRIZ_A=rand(6,6)
   MATRIZ_B=ejemploX(MATRIZ_A);
   MATRIZ_B

MATRIZ_A =
```

0.6443	0.8759	0.4709	0.2277	0.9049	0.5949
0.3786	0.5502	0.2305	0.4357	0.9797	0.2622
0.8116	0.6225	0.8443	0.3111	0.4389	0.6028
0.5328	0.5870	0.1948	0.9234	0.1111	0.7112
0.3507	0.2077	0.2259	0.4302	0.2581	0.2217
0.9390	0.3012	0.1707	0.1848	0.4087	0.1174

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

3. Los Bucles con variable de control indicada

```
MATRIZ_B =

    0.6443    0.8759    0.4709    0.2277    0.9049    0.5949
    0.3786    0.3027    0.2305    0.1898    0.9599    0.2622
    0.8116    0.6225    0.8443    0.3111    0.4389    0.6028
    0.5328    0.3446    0.1948    0.8526    0.0123    0.7112
    0.3507    0.0432    0.2259    0.1851    0.0666    0.2217
    0.9390    0.3012    0.1707    0.1848    0.4087    0.1174
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de ramificación (o bloque IF)

4. Estructuras de ramificación (o bloque IF)

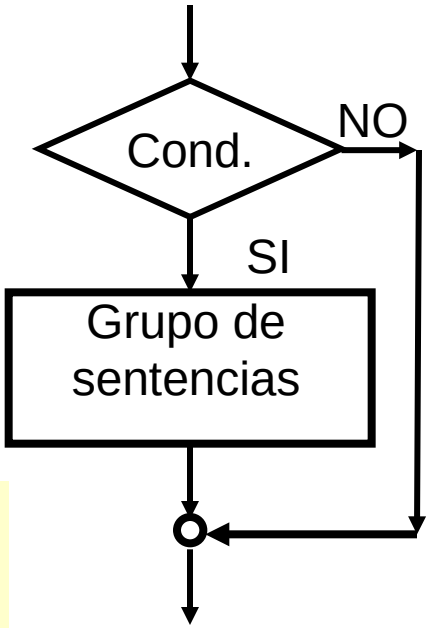
Las expresiones de relación y lógicas pueden utilizarse para optar por realizar unas operaciones (si el resultado fuese `1` `(true)`) u otras (si el resultado fuese `0``(false)`).

A su vez cada una de las opciones anteriores, puede bifurcarse con otras "condiciones" (expresiones relacionales o lógicas).

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

La estructura de bifurcación más simple es aquella en la que se ejecutan unas sentencias si el resultado de la condición es `1(true)` y no se hace nada en caso contrario:




```
Si (condición) hacer:
   Grupo de sentencias
Fin condición.
```

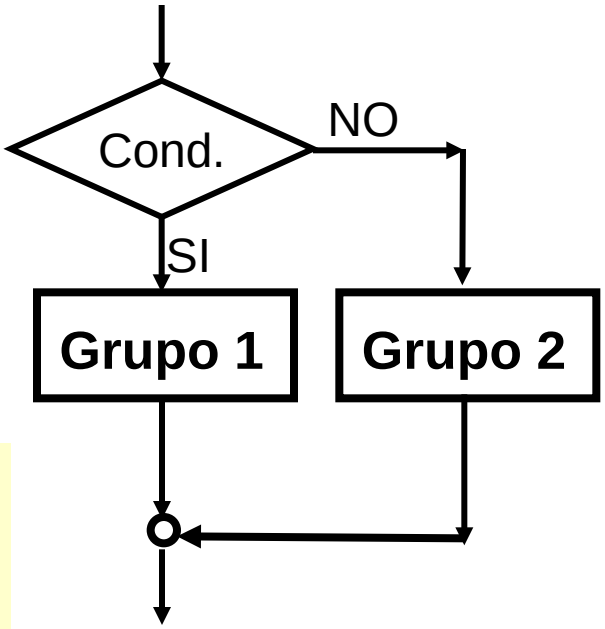
```
if (condición)
   Grupo de sentencias
end
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

Aunque también se pueden realizar unas sentencias si el resultado de la condición es `true` y otras en caso contrario:

`Si` (condición) `hacer`:
Grupo de sentencias 1
`si no:`
Grupo de sentencias 2
`Fin condición`.




```
if (condición)
   Grupo de sentencias 1
else
    Grupo de sentencias 2
end
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

Aunque también se pueden realizar unas sentencias si el resultado de la condición 1 es `true` y en el caso de ser false realizar otras sólo si una segunda condición fuese cierta:

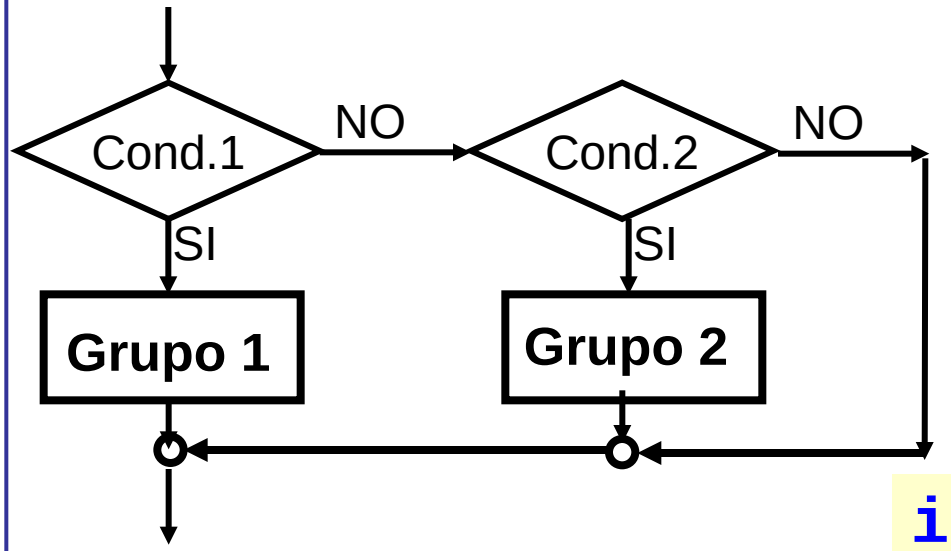
```
Si (condición 1) hacer:
   Grupo de sentencias 1
si no, si (condición 2) hacer:
   Grupo de sentencias 2
Fin condición.
```

```
if (condición 1)
   Grupo de sentencias 1
elseif  (condición 2)
   Grupo de sentencias 2
end
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

Aunque también se pueden realizar unas sentencias si el resultado de la condición 1 es `true` y en el caso de ser false realizar otras sólo si una segunda condición fuese cierta:




```
if (condición 1)
   Grupo de sentencias 1
elseif  (condición 2)
   Grupo de sentencias 2
end
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

Aunque si el el resultado de la condición 2 fuese `false` se podría realizar un grupo de sentencias 3 …. :

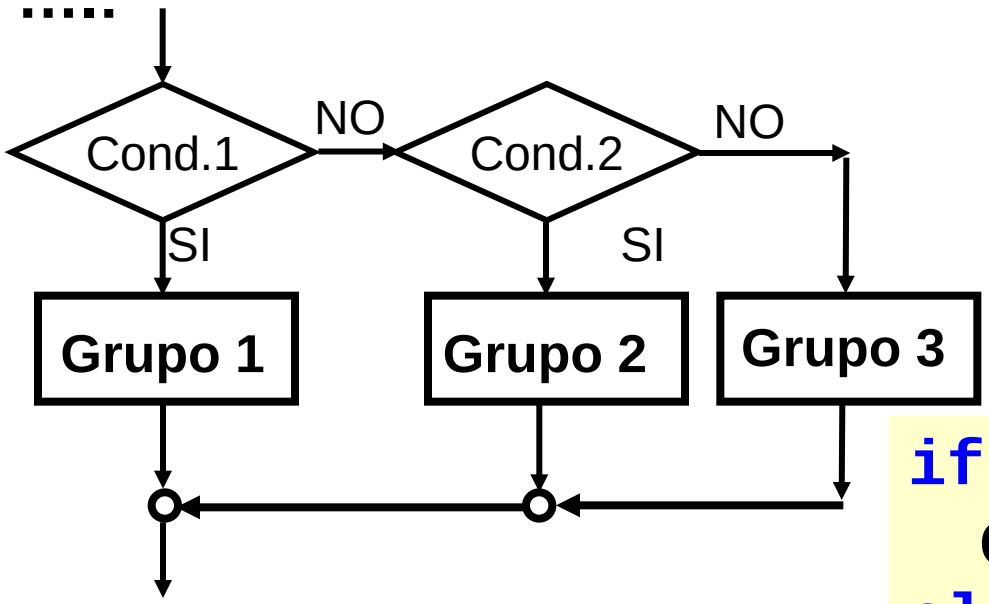
```
Si (condición 1) hacer:
   Grupo de sentencias 1
si no, si (condición 2) hacer:
    Grupo de sentencias 2
si no:
    Grupo de sentencias 3
Fin condición.
```

```
if (condición 1)
   Grupo de sentencias 1
elseif (condición 2)
    Grupo de sentencias 2
else
    Grupo de sentencias 3
end
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

Aunque si el el resultado de la condición 2 fuese `false` se podría realizar un grupo de sentencias 3 ....:




```
if (condición 1)
   Grupo de sentencias 1
elseif  (condición 2)
   Grupo de sentencias 2
else
   Grupo de sentencias 3
end
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

Aunque si el el resultado de la condición 2 fuese `false` se podría analizar otra condición 3 y si su resultado fuese `true` ejecutar un grupo de sentencias 3 ….:

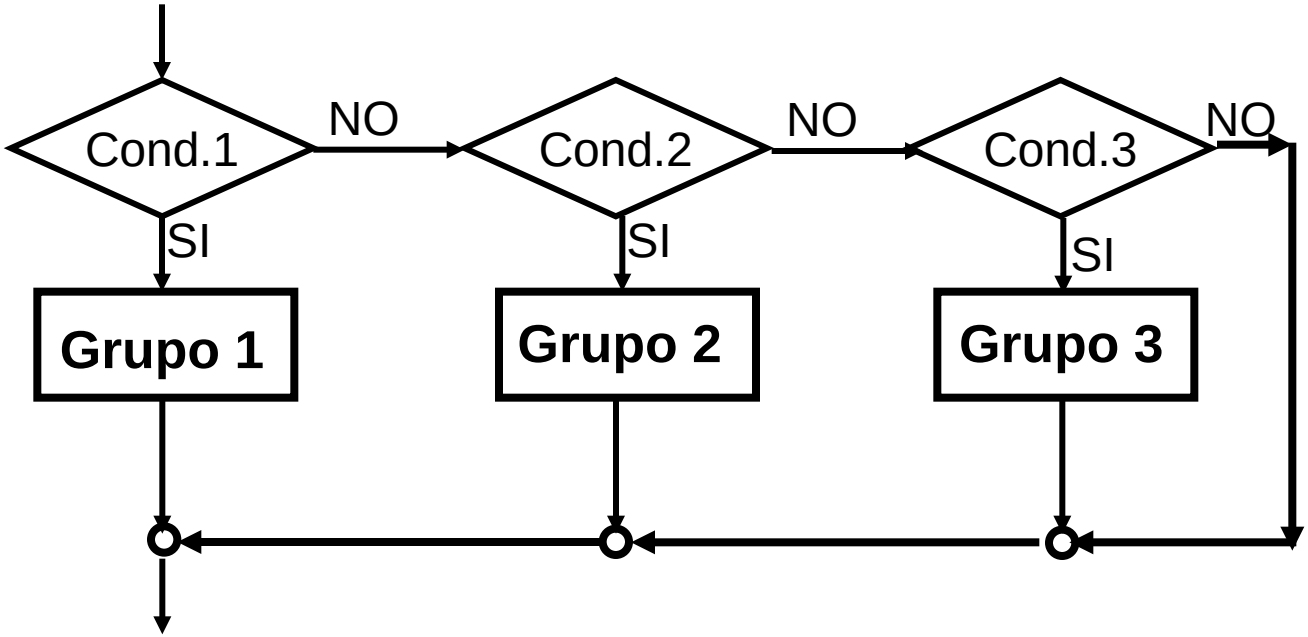
```
Si (condición 1) hacer:
   Grupo de sentencias 1
si no,si (condición 2) hacer:
   Grupo de sentencias 2
si no, si (condición 3) hacer:
   Grupo de sentencias 3
Fin condición.
```

```
if (condición 1)
   Grupo de sentencias 1
elseif (condición 2)
   Grupo de sentencias 2
elseif (condición 3)
   Grupo de sentencias 3
end
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

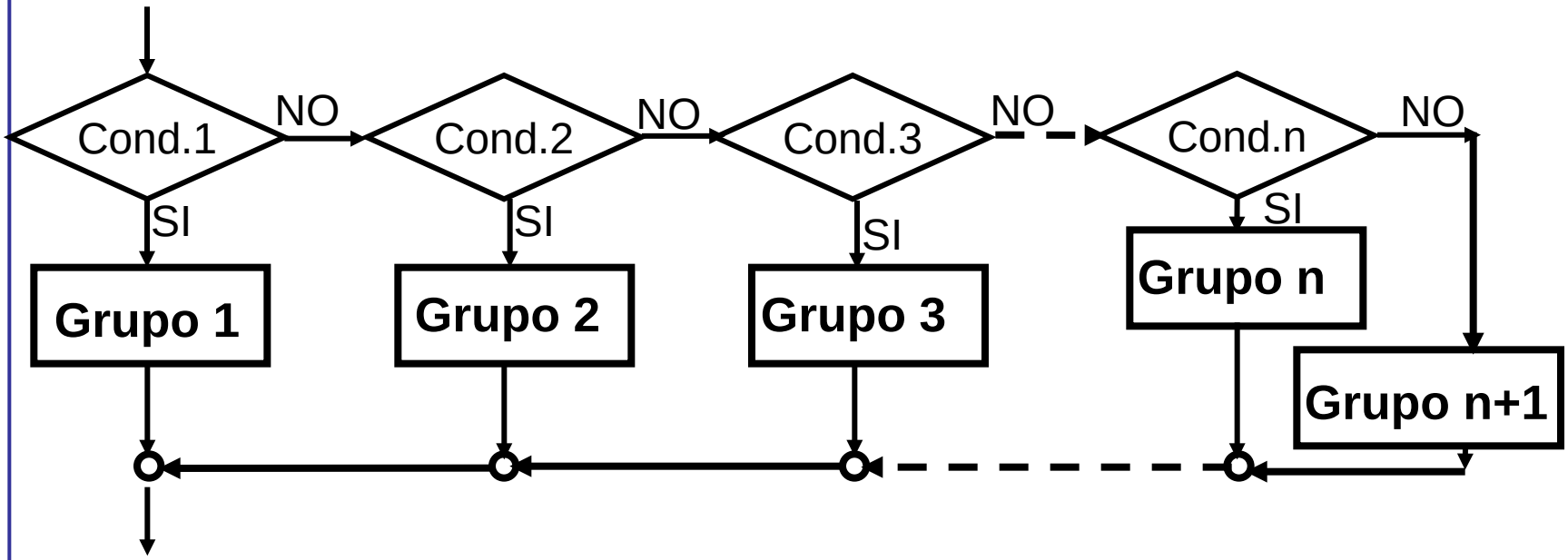
Aunque si el el resultado de la condición 2 fuese `false` se podría analizar otra condición 3 y si su resultado fuese `true` ejecutar un grupo de sentencias 3 ….:



Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

El proceso puede continuarse con tantas condiciones como se requiera:



Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

EJEMPLO:

A) Escríbase un subprograma **function**, llamado **VPLJ**, en el que sean **argumentos de entrada** un vector (**S**), un valor (**X**), y un entero positivo (**J**), y en el que se evalúe como **argumento de salida un valor** (**V**) correspondiendo al valor del **J**-ésimo polinomio de base de Lagrange $L_J(x)$ en la abscisa **X**:

$$v = L_J(x) = \prod_{\substack{K=1 \\ K \neq J}}^{n} \frac{(x - S_K)}{(S_J - S_K)}$$

donde **n** es el número de elementos del vector **S**.

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Subprogramas function en MATLAB

EJEMPLO:

```
function [v]=VPLJ(S,x,J)
% ESTA FUNCIÓN EVALÚA EN x EL J-ÉSIMO
% POLINOMIO DE BASE DE LAGRANGE
% VINCULADO AL SOPORTE DE ABSCISAS
% DADAS EN EL VECTOR S
n=length(S);
aux = 1;
for K=1:1:n
    if (K~=J)
        aux = aux*(x-S(K))/(S(J)-S(K));
    end
end
v=aux;
end
```

Escrita en el editor.
Guardada en fichero PBLAGR.m

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

EJEMPLO:

B) Escríbase un subprograma **function**, llamado **VPL**, en el que sean **argumentos de entrada** un vector (**S**) y otro vector de valores (**X**), no necesariamente con los mismos elementos, y en el que se evalúe como **argumento de salida** una matriz (**V**) con tantas filas como elementos tenga **X** y tantas columnas como elementos tenga **S** en el que el elemento **V(I,J)** sea el valor del **J**-ésimo polinomio de base de Lagrange $L_J(x)$ en la abscisa **X(I)**.

$$V(I,J) = L_J(X_I) = \prod_{\substack{K=1 \\ K \neq K}}^{n} \frac{(X_I - S_K)}{(S_J - S_K)}$$

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

```
function [V]=VPL(S,X)
m=length(S);
n=length(X);
for I=1:1:n
    for J=1:1:m
        [V(I,J)]=VPLJ(S,X(I),J);
    end
end
end
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

EJEMPLO:

C) Escríbase un programa, llamado **EP3_PBL**, en el que se proporcione un vector (**S**) y otro vector de valores (**X**), no necesariamente con los mismos elementos, y se calcule la matriz (**V**) en la que el elemento **V(I,J)** sea el valor del **J**-ésimo polinomio de base de Lagrange $L_J(x)$ en la abscisa **X(I)**.

Verifíquese el funcionamiento del programa con el soporte **S={0, 1, 2, 3}** y el vector **X={0,0.5,1,1.5,2,2.5,3}**

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

4. Estructuras de bifurcación (o bloque IF)

```
clear
S=[0,1,2,3];
X=[0,0.5,1,1.5,2,2.5,3];
[V]=VPL(S,X);
V
```

```
>> EP3_PBL
   V =
    1.0000         0         0         0
    0.3125    0.9375   -0.3125    0.0625
         0    1.0000         0         0
   -0.0625    0.5625    0.5625   -0.0625
         0         0    1.0000         0
    0.0625   -0.3125    0.9375    0.3125
         0         0         0    1.0000
```

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Estructuras de ramificación: Ejercicio 1 (3/8)

$$v \leftarrow \sum_{I=1}^{m} a_J$$

Sumatorio

En los **sumatorios** (sumas de m números) se **acumulan los sumandos**, añadiendo cada sumando a la suma de los que le preceden. Por ello, la variable en la que se acumule la suma, se inicializa:

- Con el valor **0** y se le va añadiendo sucesivamente cada uno de los sumandos, o
- Con el valor del primer sumando y se le van añadiendo sucesivamente cada uno de los sumandos a partir del 2º (inclusive)

Muchos lenguajes tienen instrucciones que lo hacen automáticamente

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Estructuras de ramificación: Ejercicio 1 (4/8)

$$v \leftarrow \prod_{I=1}^{m} a_I$$

Productorio

En los **productorios** (producto de m números) se **acumulan el producto de los factores**, multiplicando cada factor por el producto de los que le preceden. Por ello, la variable en la que se acumule el producto, se inicializa:

- Con el valor **1** y se la va multiplicando sucesivamente por cada uno de los factores, o
- Con el valor del primer factor y se le va multiplicando sucesivamente por cada uno de los factores a partir del 2º (inclusive)

Muchos lenguajes tienen instrucciones que lo hacen automáticamente

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Estructuras de ramificación: Ejercicio (5/8)

$$v = \prod_{\substack{K=1 \\ K \neq J}}^{n} \frac{(x - S_K)}{(S_J - S_K)}$$

```
INICIO
  Leer los valores de n, J, x
  Para K, desde 1, hasta n, con paso 1,
  hacer:
        Leer S_K
  Fin bucle
  v ← 1
  Para K, desde 1, hasta n, con paso 1,
  hacer:
        Si (K≠J)
              v ← v·(x-S_K)/(S_J-S_K)
        Fin condición
  Fin bucle
```

(Continúa en pág. siguiente)

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Estructuras de ramificación: Ejercicio 1 (6/8)

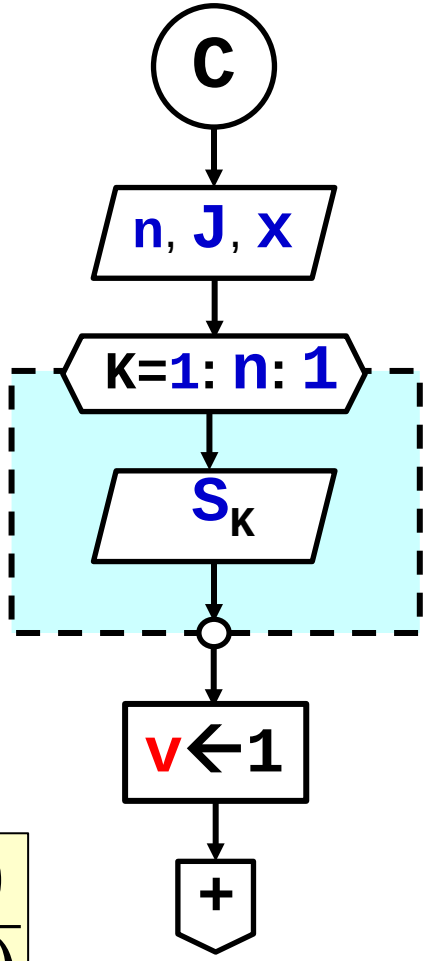
(Viene de pág. anterior)

Escribir el valor de **v**

FIN

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Estructuras de ramificación: Ejercicio 1 (7/8)



(INICIO)

(Lectura de los valores naturales **n** y **J**, del valor real **x** y del vector **S**)

(Inicialización del productorio **v**)

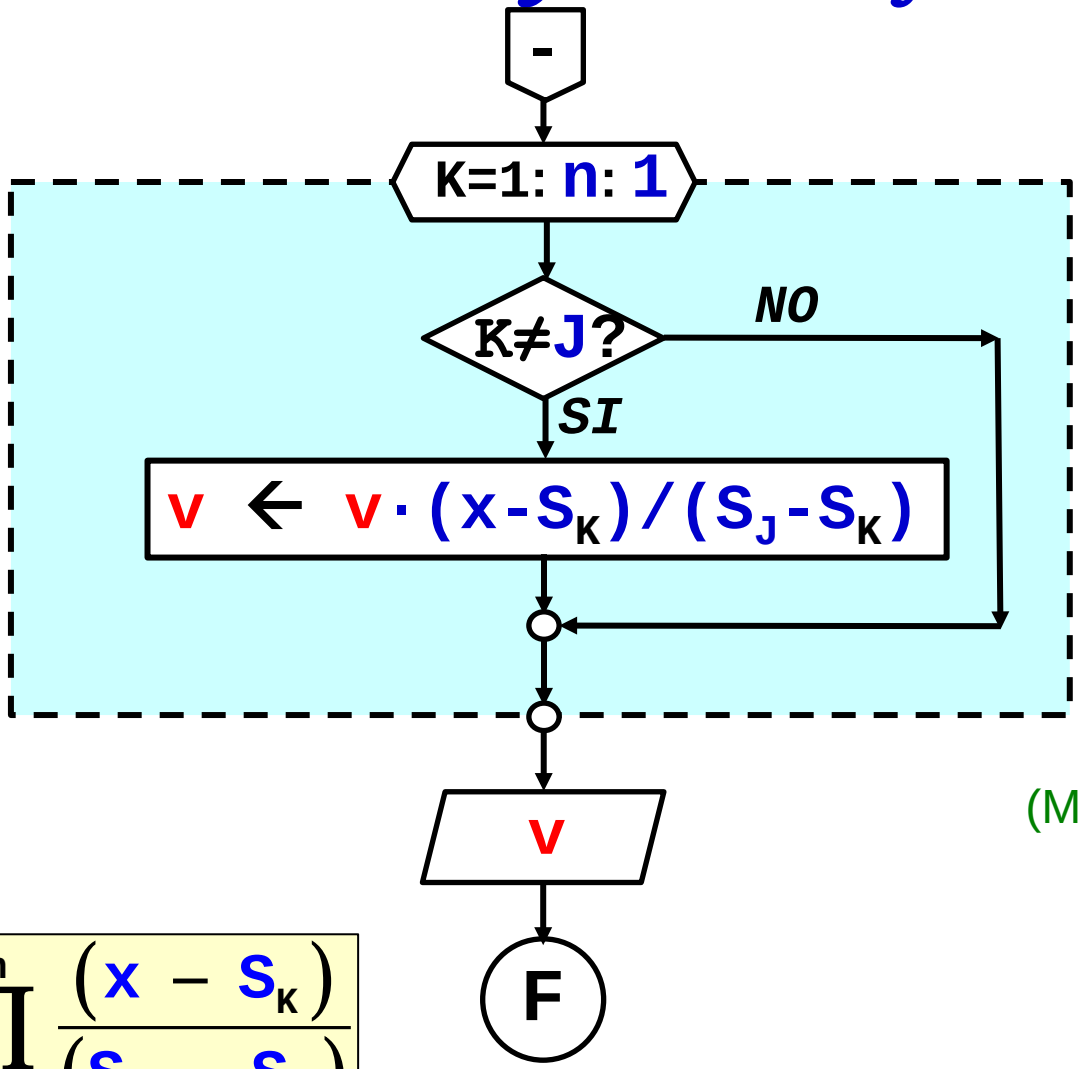
(Continúa en pág. siguiente)

$$v = \prod_{\substack{K=1 \\ K \neq J}}^{n} \frac{(x - S_K)}{(S_J - S_K)}$$

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Estructuras de ramificación: Ejercicio 1 (8/8)

(Viene de pág. anterior)



(Si K es distinto a J multiplico el valor anterior del productorio por el factor, y almaceno el resultado en la misma variable)

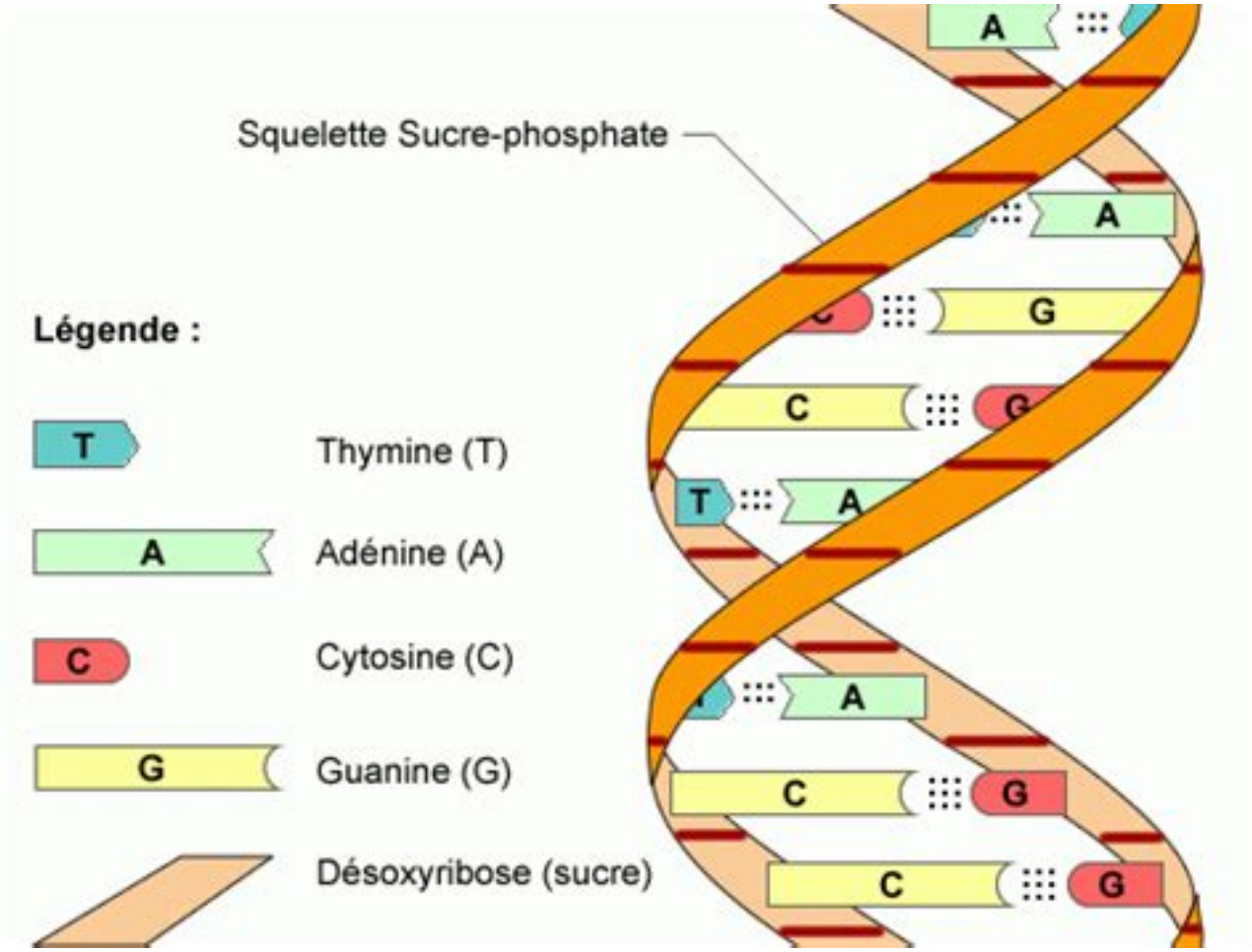
(Muestro el resultado)

(FINAL)

$$v = \prod_{\substack{K=1 \\ K \neq J}}^{n} \frac{(x - S_K)}{(S_J - S_K)}$$

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

EJERCICIO PROPUESTO (Entregable)



Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

EJERCICIO PROPUESTO (Entregable)

1º) Escribir un subprograma **function**, llamado **ADN**, en el que sea **argumento de entrada** un entero positivo llamado **N** y en el que sea argumento de salida una variable llamada **CADENA** generada como sigue:

a) Con ayuda de los comandos **floor()** y **rand()** se generará un vector fila llamado **AUX** conteniendo **N** valores aleatorios pertenecientes al conjunto {**0,1,2,3,4**}

b) Cada elemento del vector **CADENA** tendrá asignado el carácter:

- **'A'** si el elemento con el mismo subíndice de **AUX** vale **0**
- **'C'** si el elemento con el mismo subíndice de **AUX** vale **1**
- **'G'** si el elemento con el mismo subíndice de **AUX** vale **2**
- **'T'** si el elemento con el mismo subíndice de **AUX** vale **3** o **4**

(Continúa en pág. sig.)

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

EJERCICIO PROPUESTO (Entregable)

2º) Escribir un subprograma **`function`**, llamado **ACGT**, en el que sea **argumento de entrada** un vector **V** con **n** elementos cuyo valor sólo puede ser uno de los 4 caracteres siguientes **'A'**, **'C'**, **'G'** o **'T'** (es decir una *cadena de ADN*) y proporcione como **argumentos de salida** las variables **QA**, **QC**, **QG** y **QT** conteniendo el valor del número de elementos de **V** que coinciden con cada uno de los caracteres anteriores.

(Continúa en pág. sig.)

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

EJERCICIO PROPUESTO (Entregable)

3º) Escribir un subprograma **`function`**, llamado **BASES**, en el que sea **argumento de entrada** un vector **V** conteniendo una *cadena de ADN* y proporcione como **argumentos de salida** las variables **`PUR`** y **`PIR`** conteniendo respectivamente el valor del porcentaje de *bases púricas* (elementos iguales a **`'A'`** más elementos iguales a **`'G'`**) y el porcentaje de *bases pirimidínicas* (elementos iguales a **`'C'`** más elementos iguales a **`'T'`**) presentes en la cadena.

Para contar el número de elementos iguales a **`"A"`**, **`"C"`**, **`"G"`** o **`"T"`** se utilizará la función **`ACGT`** escrita en el apartado 1º.

(Continúa en pág. sig.)

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

EJERCICIO PROPUESTO (Entregable)

4º) Escribir un programa en **MATLAB**, en el que:

a) Se le pida al usuario un valor para la variable **N**

b) Con ayuda de la función **ADN** se genere una cadena aleatoria de N elementos pertenecientes al conjunto {**'A'**, **'C'**, **'G'**, **'T'**}

c) Se utilice la función **ACGT** para contar el número de elementos de cada tipo de la cadena generada.

d) Se utilice la función **BASES** para calcular el porcentaje de bases púricas y bases pirimidínicas de la cadena.

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid

Depto. de Ingeniería Geológica y Minera
E.T.S. de Ingenieros de Minas y Energía
Universidad Politécnica de Madrid