Bifurcaciones y Bucles

if : Ejecuta sentencias si se cumple una condición.

```
if expr1
  sentencias1
elseif expr2
  sentencias2
...
else
  sentencias
end
```

for: Repite sentencias un número predeterminado de veces.

```
for   i=valor inicial : incremento : valor final
sentencias
End
```

while: Repite sentencias mientras se cumpla una condición.

```
while expr
   sentencias
end
```

break: interrumpe la ejecución de un bucle for

Ejemplo 1: Realizar la sumatoria del vector u:

```
s=0;
for i=1:1:length(u)
   s=s+u(i);
end
```

Ejemplo 2: Encontrar la posición del primer elemento de valor 0 en el vector w:

```
w=[1 3 5 0 9 18];
n= length(w);
for i=1:n
   if w(i)==0
      posicion=i
      break
   end
end
```

```
w=[1 3 5 0 9 18];
i=1;
while w(i)~=0
   i=i+1;
end
posicion=i;
```


Funciones útiles

[m p]=max(v); [m p]=min(v); Son funciones cuyo primer valor de retorno corresponde al máximo (o mínimo) del vector (o matriz) ingresado y el segundo indica su posición.

```
--> A=[1 6 32; 12 23 87; 57 12 15];
-->[m p]=max(A)
 p  =
   2.   3.
 m  =
   87.
```

linspace(x1,x2,n); Genera un vector fila de n valores equiespaciados que van de x1 a x2.

logspace(x1,x2,n); Devuelve un vector fila de n puntos logarítmicamente equiespaciados entre 10^{x1} y 10^{x2}.

sum(v); Suma todos los elementos del vector

```
--> v=[1 6 32];
--> sum(v)
 ans=
    39.
```

abs(v); Valor absoluto del argumento.

```
--> A=[1 -6;-3 7];
--> abs(A)
 ans=
      1.   6.
      3.   7.
```

Barra izquierda "\"; x=A\b es la solución al sistema de ecuaciones Ax=b.

- Si A es cuadrada y no singular x = A\b es equivalente a x=inv (A)*b pero de manera mas eficiente.
- Si A no es cuadrada, x = A\b es la solución de mínimos cuadrados.

Ejemplo:

```
--> A=[9 -36 30;-36 192 -180;30 -180 180];
--> b=[3;-24;30];
--> x=A\b
 x  =
   1.
   1.
   1.
```

MSA

Scilab

Pocket Guide

Matrices y Vectores

Se definen por filas; los elementos de una misma fila están separados por espacios o comas y las filas por Entre o punto y coma. Por ejemplo, si deseamos ingresar la matriz A, el vector v y el vector u:

$$A=\begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\end{pmatrix} \quad v=\begin{pmatrix}3 & 7 & 2\end{pmatrix} \quad u=\begin{pmatrix}1\\ 3\\ 7\end{pmatrix}$$

En Scilab:

```
--> A=[1 2 3; 4 5 6; 7 8 9];
--> v=[3 7 2];
--> u=[1;3;7];
```

Nota: Las variables escalares pueden ser ingresadas como vectores de un solo elemento. Ejemplo: --> p= 3.14

Manipulación de elementos

En Scilab se accede a los elementos de un vector o de una matriz poniendo la posición entre paréntesis a la derecha del nombre.
Ejemplo:

```
--> v(2)
anos=
      7.
--> m=A(1,3)
m=
      3.
```

Operaciones con matrices

Los operadores matriciales:

+ Adición o suma; – Sustracción o resta
* Multiplicación; .* Producto elemento a elemento
^ Potenciación; .^ Potencia elemento a elemento
/ y \ División derecha e izquierda;
./ y .\ División elemento a elemento;
' Traspuesta

Nota: Se aplican también a las variables o valores escalares, aunque con algunas diferencias. Todos estos operadores son coherentes con las correspondientes operaciones matriciales: no se puede por ejemplo sumar matrices que no sean del mismo tamaño. Si no se usan de modo correcto se obtiene un mensaje de error.

Creación de funciones

Se definen bajo la siguiente sentencia:

función [valores de retorno]=ñame)
endfunction

name corresponde al nombre de la función. Entre corchetes y separados por comas se definen los valores de retorno (salida/s de la función). Los argumentos (entadra/s de la función) se definen entre paréntesis separados por comas.
Ejemplo: Creamos la función "promedio" que encuentra la media aritmética de un vector.

```
function [x]=promedio(a)
x=sum(a)/length(a);
endfunction
```

Al ejecutar las sentencias anteriores, la función promedio se activa para utilizarse como cualquier otra función de Scilab.

```
--> v=[3 7 2];
--> promedio(v)
ans=
        4.
```

Ejemplo: Deseamos obtener la densidad del aire en función de la presión y temperatura, considerándolo como ideal.

```
function [densidad]=dens_ideal(P, T)
// P en atmosfera
// T en ºC
// PM en g/mol
// R en atm.L/mol.ºK
// densidad en g/L
Tk=T+273.15;
PM=29;
R=0.082;
densidad=(P*PM)/(R*Tk);
endfunction
```

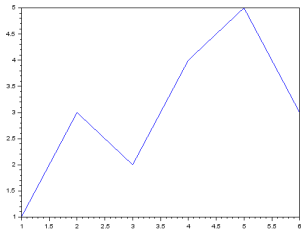
```
-->  dens_ideal(1,25)
ans=
          1.186176
```


Gráficos bidimensionales

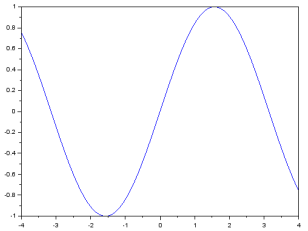
Scilab dispone de varias funciones básicas para crear gráficos 2-D. Algunas son:
plot() crea un gráfico a partir de vectores y/o columnas de matrices, con escalas lineales sobre ambos ejes.
plot2d() similar a plot pero cuenta con funciones de cambio de escala en los ejes.
fplot2d() grafico 2D de una curva definida por una función
polarplot() crea una gráfica de coordenadas polares del ángulo theta versus el radio rho
Nota: Para mayor información consultar el comando help seguido de cualquiera de las anteriores funciones.

Ejemplos:

```
-->  x=[1 3 2 4 5 3];
--> plot(x)
```



```
--> x=-4:0.01:4;
--> y=sin(x);
--> plot(x,y)
```



```
--> Re=(1e4:100:1e6);
--> f=(-2*log10(0.001/3.7 + 5.1286./(Re.^0.89))).^(-2);
--> plot2d("ll",Re,f)
```

Doble escala logarítmica