

**OCTAVE: FUNCIONS**  
**UNIVERSITAT POLITÈCNICA DE CATALUNYA**  
**ALGEBRA LINEAL, ENGINYERIA QUÍMICA**

JAUME AMORÓS, UPC, BARCELONA

RESUM. Qué són les funcions en Octave, com fer-les servir i com crear-ne de noves.

Guia breu per a fer les pràctiques d'Octave de l'assignatura d'Àlgebra Lineal, Enginyeria Química, de la UPC.

ÍNDIX

1. Qué és una funció en Octave?	1
2. Com s'usa una funció en Octave?	2
3. Com es fa una funció en Octave?	3
4. Més informació sobre funcions	4

1. QUÉ ÉS UNA FUNCIO EN OCTAVE?

Una funció és un programa que es pot executar dins de l'Octave de la mateixa manera que hom crida una instrucció del llenguatge.

És un conjunt, llarg o breu, d'instruccions d'Octave, que s'executen de cop quan cridem a la funció. Agrupar-les en una funció evita haver d'introduir-les una a una de manera repetitiva.

Les funcions en Octave (i en la majoria de llenguatges de programació) són com les funcions i aplicacions matemàtiques:

- Quan es crida una funció cal passar-li uns *arguments*, que poden ser nombres, matrius . . . i que són la informació que la funció farà anar pel seu càlcul.
- La funció produeix uns *valors de retorn* (un o varis), que són el resultat dels càlculs.

*Exemple 1.* Guardem el nombre 43 en una variable a:

a=43

Ara li calculem l'arrel quadrada, que guardem en una variable b:

b=sqrt(a)

La variable b conté ara l'arrel quadrada de 43 (6.5574 aprox.).

En aquest exemple:

---

*Date:* 2003/12/21.

- la *funció* era `sqrt`,
- l'*argument* passat a la funció era `a`
- el *valor de retorn* de la funció ha estat guardat a la variable `b`.

## 2. COM S'USA UNA FUNCIO EN OCTAVE?

Cada funció té el seu nom, una llista d'arguments que demana, i una llista de valors que retorna.

Aquestes llistes d'arguments i de valors de retorn poden tenir un sol element, més d'un, o fins i tot ser buides.

En l'exemple 1 de la secció anterior hem usat la funció `sqrt`, que rep com argument un nombre i retorna un altre nombre.

Quan la funció demana més d'un argument, aquests se li han de donar al cridar-la, posant-los entre parèntesi, separats per comes, i en l'ordre en que la funció els demana.

Quan la funció retorna més d'un valor els hem de guardar posant-los en una llista, separats per comes i entre claus quadrades `[ ]` com s'indica a l'exemple 3.

*Exemple 2.* En Octave podem operar amb els polinomis si els guardem com a vectors fila, escrivint els monomis en ordre decreixent. La funció `conv` multiplica dos polinomis.

Declarem el polinomi  $p_1(x) = 3x^3 - 2x^2 + 5x - 2$

```
p1=[3,-2,5,-2]
```

Ara declarem el polinomi  $p_2(x) = x^2 - 6$

```
p2=[1,0,-6]
```

Multipliquem  $p_1(x)$  amb  $p_2(x)$  i guardem el resultat en  $pr(x)$

```
pr=conv(p1,p2)
```

En aquest exemple cridem la funció `conv` donant-li dos arguments, els vectors `p1` i `p2`. La funció interpreta els vectors com a polinomis, fa la multiplicació, i retorna un únic vector, que conté els coeficients del polinomi producte. Hem guardat aquest producte en una variable `pr`, però si només el volem veure per pantalla n'hi ha prou amb escriure `conv(p1,p2)`.

*Exemple 3.* La funció `deconv` divideix polinomis. Aquesta funció demana dos arguments: el vectors que contenen el polinomi dividend i el polinomi divisor, i retorna dos valors: el quocient i el reste (de nou com a vectors).

Declarem el polinomi dividend  $p(x) = 3x^3 - 2x^2 + 5x - 2$

```
p=[3,-2,5,-2]
```

Declarem el polinomi divisor  $d(x) = x^2 - x + 2$

```
d=[1,-1,2]
```

Fem la divisió de  $p(x)$  per  $d(x)$ : guardem el quocient en el vector `q`, i el reste en el vector `r`:

```
[q,r]=deconv(p,d)
```

Notem que aquí l'ordre dels valors de retorn i dels arguments és molt important!:

- Si fem  $[r, q] = \text{deconv}(p, d)$  es guardarà a  $r$  el quocient i a  $q$  el reste.
- Si fem  $[q, r] = \text{deconv}(d, p)$  es farà la divisió de  $d(x)$  per  $p(x)$ .

### 3. COM ES FA UNA FUNCIO EN OCTAVE?

Una funció d'Octave es posa en un fitxer de text, i es pot escriure i editar en qualsevol editor per a fitxers de text. Recordeu que el Microsoft Word NO és un editor de text. En canvi el *Cuaderno de Notas* o el *Wordpad* sí ho són.

El nom del fitxer de text ha de ser el de la funció, i la seva terminació ha de ser `.m`.

Quan creeu una funció, vigileu que el vostre editor de text o sistema operatiu no afegixi alguna terminació tipus `' .txt'` al final del nom; si ho fa, la funció no funcionarà.

La funció té una primera línia i una darrera línia que tenen el format fixat, i entremig va el cos de la funció.

La **primera línia** de l'arxiu de text que conté la funció ha de ser així:

```
function [valor 1,valor 2,...]=nom(argument 1,...)
```

on

- entre les claus `[ ]` van els valors que ha de retornar la funció, per ordre i separats per comes,
- **nom** és el nom que ha de tenir la funció
- després del nom de la funció van els arguments que se li passen, ordenats i separats per comes.

Després de la primera línia ve el **cos** de la funció, que són instruccions standard d'Octave i/o crides a altres funcions.

En el cos de la funció hi podeu posar comentaris. Per a indicar que la línia es de comentari i no s'ha d'executar poseu al començament el símbol `%`.

Finalment, la **darrera línia** de la funció és

```
return;
```

per a indicar a Octave que ha acabat la funció.

Per a **executar una funció** que heu creat, heu de posar l'arxiu nom de la funció.m en el *directori de treball* d'Octave.

El directori de treball per Octave 2.1.36 en Windows és `C:\Archivos de programa\GNU Octave 2.1.36\octave_files`

El directori de treball per Octave en Linux és aquell des d'on el crideu.

Un cop teniu la funció en el directori de treball (ho podeu comprovar fent `dir` dins d'Octave, aleshores ha d'apareixer el fitxer de la vostra funció acabat en `.m`), podeu cridar-la com qualsevol altra instrucció del llenguatge. És a dir, que heu de tenir ja definits els arguments que demana la funció, cridar-la amb aquests arguments i guardar en variables els valors que retorna.

*Exemple 4.* Creem una funció `eqs2base` per a que a partir de les equacions que defineixen un subespai ens calculi la dimensió i una base del subespai. Aquesta funció va dins d'un fitxer de text anomenat `eqs2base.m`

El contingut del fitxer `eqs2base.m` és

```
function [base,dim]=eqs2base(eqs)

% Funcio per calcular base i dimensio d'un subespai
% a partir de les seves equacions.
% Calcul de rangs habitual d'Octave.
% INPUT: matriu eqs amb files un sistema d'equacions del
% subespai (sense terme independent zero)
% OUTPUT: matriu base amb columnes una base del subespai,
% dimensio dim del subespai

% Jaume Amoros, UPC, Barcelona

% 2003/12/10

base=null(eqs);
dim=rank(base);
return;
```

Observeu com la primera i la darrera linea segueixen els formats d'inici i final de funció respectivament, com hi han linees de comentari que expliquen que són els arguments i els valors que retorna, i entremig hi van les instruccions Octave que fan la feina (esencialment la instrucció `null`).

Si ara volem fer anar aquesta funció per a calcular una base i dimensió del subespai  $F = \{(x, y, z) \in \mathbb{R}^3 \mid x - 2y - 5z = 0, 2x + 7y - z = 0\}$  hem de fer el següent:

- (i) Posem el fitxer de text `eqs2base.m` en el directori de treball d'Octave.
- (ii) Arrenquem Octave i introduïm a ma la matriu de coeficients de les equacions del subespai:

$$A = [1, -2, 5; 2, 7, -1]$$

- (iii) Cridem a la funcio i guardem el resultat:

$$[\text{BasF}, \text{dimF}] = \text{eqs2base}(A)$$

Ara la variable `BasF` és una matriu  $1 \times 3$  que conté en columna un vector base de  $F$ , i la variable `dimF` conté la dimensió del subespai.

#### 4. MÉS INFORMACIÓ SOBRE FUNCIONS

Per a tenir informació més detallada podeu mirar el capítol 13 del manual de l'Octave, o be les seccions 3.2 i 7.3.2 del manual *Aprenda MATLAB como si estuviera en primero*, de l'Universidad de Navarra. Podeu trobar-los tots dos a la pàgina web sobre Octave de l'assignatura

<http://www-ma1.upc.es/~amoros/octave.html>

i si us heu instal·lat l'Octave al PC de casa, els teniu en la seva carpeta.