MATLAB e Octave

Seconda Parte

Programmare in MATLAB

- Gli m-file sono macro di comandi MATLAB/Octave; sono memorizzati come file di testo con l'estensione ".m", cioè filename.m

- Ci sono due tipi di m-file, gli script e le function

script file

```
%  script file
p=[5 3 2];
r=sqrt(p);
disp(r)
```

function file

```
% function file
function [y]=fun(x)
y=x^2+3*x^2+2;
return
```

m-file – continua

- Script
  - Consiste di una lista di comandi e istruzioni che saranno eseguiti sequenzialmente

- Function
  - Definisce una funzione che può essere chiamata
  - In Matlab/Octave, una function è simile ad una function C/Fortran
  - Ci sono diversi tipi di function Matlab/Octave:
    - Function anonime, che non richiedono un m-file ma solo una singola espressione Matlab/Octave
    - Function primarie e secondarie

Script vs function

Script file

- Lista di istruzioni MATLAB/Octave
- Le variabili sono globali
- Si esegue digitando il nome del file

Function file

- Inizia con **function**
- Lista di istruzioni MATLAB/Octave
- Termina con **end**, ritorna con **return**
- Le variabili sono locali
- Si esegue digitando il nome del file e dando i parametri

Function

- Funzioni primarie e secondarie
  - Funzioni primarie danno il nome al file
  - Altre function secondarie possono essere definite nello stesso file
  - Funzioni primarie possono essere richiamate da un qualsiasi punto (di uno script o altra function) mentre le function secondarie possono essere richiamate solo dalla corrispondente function primaria.

Function

- Sintassi

  function [out1, out2, ...] = funname(in1, in2, ...)

  definisce una function di nome funname che accetta in ingresso i parametri/valori in1, in2, … etc. e restituisce in uscita i parametri/valori out1, out2, … etc.

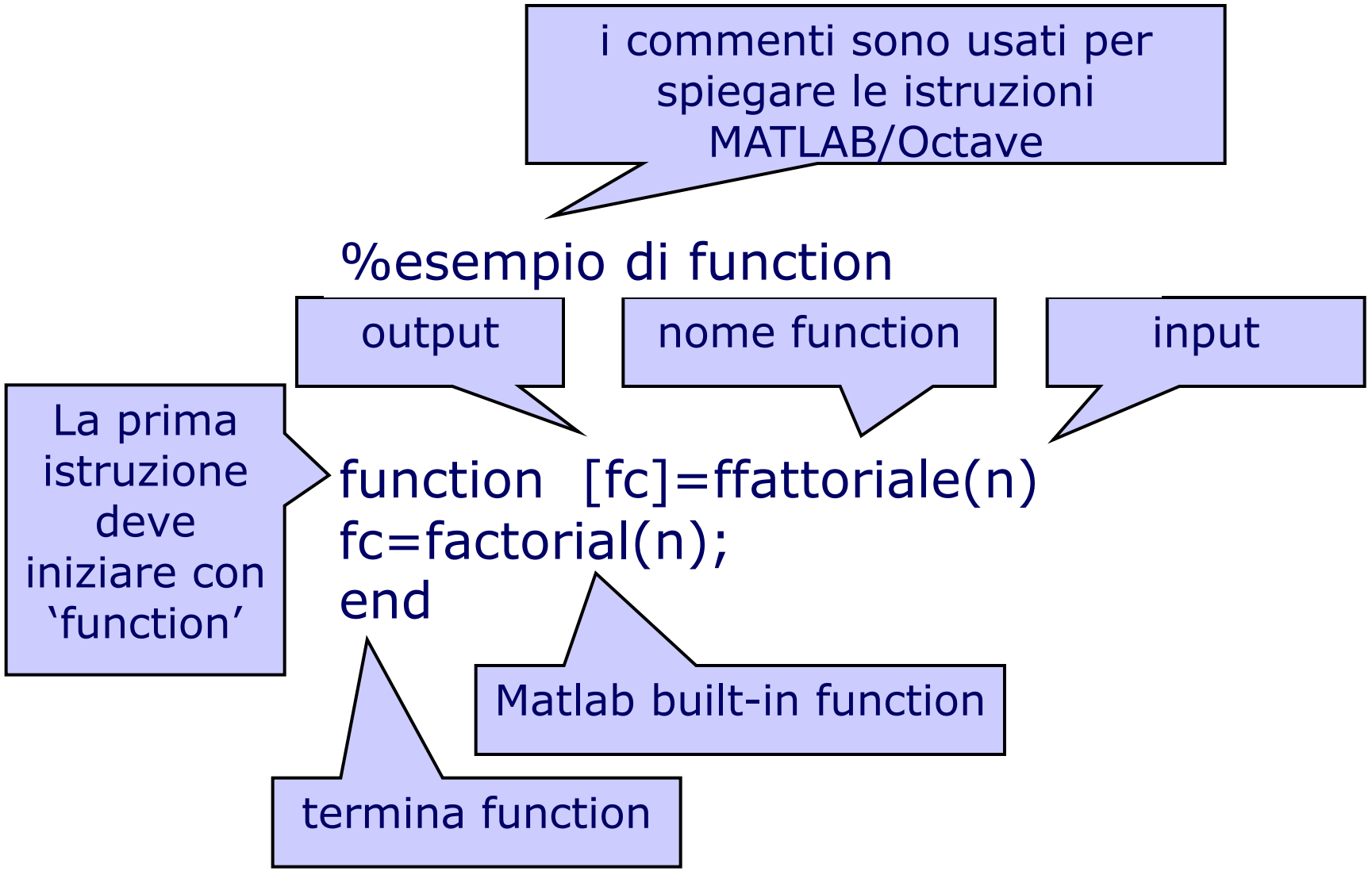
Creare una function

Usare l'Editor per scrivere la seguente function:

```
%esempio di function
function  [fc]=ffattoriale(n)
  fc=factorial(n);
end
```

- Salvare il programma usando 'salva con nome' e dare 'ffattoriale.m' come nome
- Se si è usato un altro editor per creare il file usare il nome 'ffattoriale.m'
- Salvare il file in una directory riconosciuta da MATLAB/Octave

Function utente



Eseguire uno script

Dalla Command Window dare il comando:

```
>>fc=ffattoriale(7)
```

- MATLAB/Octave cerca nelle directory corrente se esiste un m-file con questo nome
- Se lo trova lo carica in memoria e lo esegue, cioè esegue istruzione per istruzione il contenuto della function dopo avergli passato come argomento il valore 7; alla fine dell'esecuzione il risultato viene memorizzato in fc e visualizzato perché non c'è il ";" alla fine dell'istruzione.

```
5040
>>
```

Function per calcolare cos(x)

```
% function per  calcolare una stima di cos(x)
% con x scalare in [0,1]
% Sviluppo di Taylor di grado 4:
% cos(x) ≈ 1 - x^2/2! + x^4/4!
function [Sum]=fcos(x)
Sum=1;
n=2;
fc2=factorial(n);
Sum=Sum-x^2/fc2;
n=4;
fc4=factorial(n);
Sum=Sum+x^4/fc4;
```

Function factorial

Function factorial

Eseguire lo script

Dalla Command Window dare il comando:

```
>>cs=fcos(pi/2)
```

- Octave/MATLAB cerca nelle cartella corrente se esiste un m-file con questo nome
- Se lo trova lo carica in memoria e lo esegue, cioè esegue istruzione per istruzione il contenuto della function dopo avergli passato come argomento il valore pi/2; alla fine dell'esecuzione il risultato viene memorizzato in Sum e copiato in cs come valore di ritorno della function.

```
>>cs
0.019969
```

Functions secondarie

```
% function per  calcolare una stima di cos(x)
% con x scalare in [0,1]
% Sviluppo di Taylor di grado 4:
% cos(x) ≈ 1 - x^2/2! + x^4/4!
function [Sum]=fcos(x)
Sum=1;
n=2;
fc2=factorial(n);
Sum=Sum-fpower(x,2)/fc2;
n=4;
fc4=factorial(n);
Sum=Sum+fpower(x,4)/fc4;
end
function y=fpower(x,n)
y=x.^n;
end
```

Function secondaria

Esercitiamoci nel fare una function

Seguendo il foglio dell’ Esercitazione 2: script, function e grafici in Matlab, realizzare una function per gli Esercizi A

Funzioni e Script-File

Riassumiamo brevemente quanto detto finora

e

vediamo le strutture di controllo più utilizzate in linguaggio Matlab/Octave

Script-File

- Uno script file è un file contenente una qualunque sequenza di comandi.
  - Esso viene letto e valutato esattamente come se ogni comando fosse digitato al prompt.
  - Permette un modo per memorizzare una sequenza di comandi che non starebbero logicamente dentro ad una funzione.
  - A differenza di una funzione, uno script file non deve cominciare con la parola chiave function.
  - Variabili usate in uno script file non sono variabili locali, ma sono nello stesso scope delle altre variabili entrate al prompt.

Function-File

- Programmi complessi possone essere semplificati mediante la definizione di funzioni.
- Le funzioni possono essere definite direttamente sulla linea comandi durante una sessione interattiva.
- Alternativamente, le funzioni possono essere create come file esterni e possono essere chiamate proprio come le funzioni predefinite.

Definizione di funzioni

- Nella sua forma più semplice, la definizione di una funzione chiamata fname è del tipo:

```
function fname
  body;
end
```

- Un nome valido di funzione segue le stesse regole per il nome di variabile.
- Il corpo della funzione consiste di espressioni e istruzioni di controllo.

Passaggio informazioni a funzioni

- Solitamente, si vogliono passare alcune informazioni alle funzioni che si definiscono.

```
function fname(arg_list)
  body;
end
```

- dove **`arg_list`** è una lista di argomenti separati da virgole. Quando la funzione viene chiamata, i nomi degli argomenti valgono i valori dati nella chiamata.

Ritorno di informazioni

- Nella maggioranza dei casi, si vogliono anche ottenere certe informazioni di ritorno dalla funzione che si definisce.

```
function ret_var=fname(arg_list)
  body;
  ret_var = …;
end
```

- Il simbolo **`ret_var`** è il nome della variabile, definita dentro la funzione, che varrà il valore che deve essere ritornato.

Ritorno di informazioni

- Le funzioni possono ritornare più di un valore
-
```
function [ret_list]=fname(arg_list)
  body;
  …
end
```

- dove **`ret_list`** è una lista di nomi di variabili separati da virgole che varranno i valori ritornati dalla funzione. Si noti che **`ret_list`** è un vettore racchiuso in parentesi quadre.

Istruzioni di Controllo

- Le istruzioni di controllo, controllano il flusso dell'esecuzione nei programmi (script e function)
  - Tutte le istruzioni di controllo iniziano con una parole chiave iniziale
  - Ogni istruzione di controllo ha una corrispondente parola chiave finale
  - La lista delle istruzioni contenute fra la parola chiave iniziale e quella finale viene chiamato il corpo dell'istruzione di controllo

Strutture di Controllo

- **Matlab while statement**

```
while (condition)
  body;
end
```

- **Matlab for statement**

```
for var = expression
  body;
end
```

Strutture di Controllo

- **Matlab if statement**
  - Le clausole else ed elseif sono opzionali. Può esistere un qualunque numero di clausole elseif.

```
if (condition)
  then-body;
elseif (condition)
  elseif-body;
else
  else-body;
end
```

Strutture di Controllo

- **Matlab switch statement**
  - Sono ammessi un qualunque numero di "case label"

```
switch expression
  case label
    command_list;
  case label
    command_list;
  ...
  otherwise
    command_list;
end
```

Strutture di Controllo

- **L'istruzione `break`**
  - Esce dal ciclo for o while in cui è inclusa. L'istruzione **`break`** può essere usata solo nel corpo di un ciclo

- **L'istruzione `continue`**
  - Come **`break`**, viene usato solo dentro cicli **`for`** o **`while`**. Salta il resto del corpo del ciclo, provocando il passaggio alla successiva iterazione del ciclo

Caricare e salvare Dati

Durante il lavoro con MATLAB/Octave, si può avere la necessità di salvare vettori e matrici definite nel programma. Per salvare il file dati nella directory di lavoro, digitare

save filename

dove "filename" è un nome scelto dall'utente.

save('ofile.txt','-ascii','ndata'); il nome del file salvato sarà ofile.txt e conterrà i valori dell'array ndata

save ndata; il nome del file che viene salvato è ndata.mat

Per recuperare i dati nel seguito, digitare

load filename

>>load ndata; il nome del file letto sarà ndata.mat

>>load('ndata'); il nome del file letto sarà ndata.mat

>>mydata = load('ofile.txt'); il nome del file letto sarà ofile.txt e l'array sarà mydata

Caricare e salvare Dati

In alternativa si possono caricare e salvare dati utilizzando i comandi **fopen**, **fscanf**, **fprintf** ed **fclose**

```
fid = fopen('input.txt', 'r');
%legge da file un dato alla volta con formato intero
  im=fscanf(fid,'%d',1);
  ai=fscanf(fid,'%d',1);
  af=fscanf(fid,'%d',1);
fclose(fid);

fid = fopen('output.txt', 'wt');
  fprintf(fid,' Intestazione File \n');
%stampa su file valori numerici con il formato specificato
  fprintf(fid,'%5d %7.3f \n', num, float);
fclose(fid);
```

Grafici in MATLAB

MATLAB permette di fare dei grafici/disegni in una finestra, aperta con il comando figure. Per fare un disegno/grafico è necessario riferirsi al sistema di assi cartesiani associato alla figure; il sistema è orientate in modo classico e le coordinate sono numeri floating point.

Comando	Descrizione
figure(k)	Apre una nuova figura e gli assegna l'id k
plot(x,y)	Fa un grafico nella finestra corrente
title('text')	Inserisce il titolo nel grafico
xlabel('text')	Aggiunge un'etichetta all'asse orizzontale
ylabel('text')	Aggiunge un'etichetta all'asse verticale
legend('text')	Inserisce una legenda
hold on	Mantiene quanto disegnato nella finestra

Grafici 2D

Per il disegno 2D la funzione Matlab più importante è la funzione o comando

plot (x , y)

Il comando plot (x, y) fa un disegno/grafico dei punti del piano le cui ascisse sono definite nel vettore x e le ordinate nel vettore y, ed eventualmente li connette con dei segmenti retti (disegna nell'ordine i punti del piano di coordinate (x(i),y(i))).

Funzione plot: sintassi

plot(vettore1, vettore2, opzioni)

vettore1 e vettore2 sono i vettori di dati (ascisse e ordinate dei punti)

opzioni è una stringa di al più 3 caratteri che definisce le 3 seguenti caratteristiche:

– colore

– marker (simbolo usato per il disegno di punti)

– line specification (simbolo usato per il tipo di tratto fra i punti)

un carattere per ogni caratteristica secondo la seguente convenzione:

Funzione plot: opzioni

Colore		Marker		Line Specif.	
b	blue	.	point	-	solid
g	green	o	circle	:	dotted
r	red	x	x-mark	-.	dashdot
c	cyan	+	plus	--	dashed
m	magenta	*	star	(none)	no line
y	yellow	s	square		
k	black	d	diamond		
		v	triangle (down)		
		^	triangle (up)		
		<	triangle (left)		
		>	triangle (right)		
		p	pentagram		
		h	hexagram		

Funzioni utili

axis: controlla gli assi di Matlab e il loro scaling e appearance (digitare help axis per le opzioni)

grid: disegna delle linee griglia

cla: cancella gli assi e resetta il default

Esercitiamoci con script e function

Seguendo il foglio dell'
Esercitazione 2: script, function e grafici in Matlab, realizzare script e function per la risoluzione degli Esercizi B.