Matlab

The

Matrix Laboratory

Oliviero Giannini

Laboratorio di progettazione strutturale meccanica
Esercitazione 1

Sommario dell'esercitazione

- L'ambiente Matlab
- Gli help di Matlab
- Nozioni di sintassi base
- Istruzioni FOR e IF
- PLOT
- campionamento
- Script e function
- Programmazione pragmatica

Ambiente Matlab

- Il Workspace
  - Allocazione dinamica della memoria
  - L'istruzione clear
- Le librerie Matlab
- Gli M-files: Script e function

  - Matlab è case sentive

Gli Help di Matlab

- Help [function]
  - Da informazioni compatte sulla sintassi, presuppone la conoscenza di cosa fa la data funzione
- Doc [function]
  - Apre una finestra di help in html con sintassi dettagliate e informazioni sulla matematica che sta dietro alle funzioni
- È inoltre possibile fare ricerche su tutto il testo degli help.
- i demos presentano il funzionamento dei tollbox: gruppi di funzioni

Le directory

- Matlab accede a tutti i file nella work directory
- la work directory può essere cambiata
- Si possono definire delle directory su cui mettere function o script che sono sempre accessibili
- La directory di lavoro può essere cambiata all'interno di uno scrip con l'istruzione "cd"

Sintassi base:

Assegnazione variabili

```
>> a=1

a =
    1
>> b=[1 2 3]

b =
    1    2    3

>> e=[1 2 3;4 5 6;6 7 8]

e =
    1    2    3
    4    5    6
    6    7    8
```

```
>> c=[0.1;0.2;0.3]
c =
   0.1000
   0.2000
   0.3000

>>k=c'
k=
    0.1000   0.2000   0.3000

>> d=0:0.1:1

d =
        0   0.1000   0.2000   0.3000   0.4000
    0.5000   0.6000   0.7000   0.8000
    0.9000   1.0000
```

Sintassi base:

Operazioni base

```
>>f=a+5
f =
     6
>> g=f*2
g =
    12
>> h=g/f
h =
     2
>>i=e*h
i =
     2     4     6
     8    10    12
    12    14    16
```

(analogamente somme e divisioni per scalari)

```
>> clear all
a =
     1     2     3
>> b=[1;2;3]
b =
     1
     2
     3
>> c=a*b
c =
    14
>> d=b*a
d =
     1     2     3
     2     4     6
     3     6     9
```

```
>> clear b
b=[2 3 4]
b =
     2     3     4
>> c=a.*b
c =
     2     6    12
```

(analogamente per le divisioni)

```
>> d=c.^2
d =
     4    36   144
>>k=d(2:3)
k=
     36 144
```

Il ciclo FOR e l'istruzione IF

```
FOR variable = expr, statement, ...,
     statement
END

IF expression
        statements
END
IF expression
     statements
ELSE
        statements
END
```

Operatori logici:

Uguale **==** diverso **~=**

Maggiore **>** minore **<** minoreuguale **<=**

NOT **~** AND **&** OR **|**

```
for i=1:10
     a(i)=i;
     %qualunque istruzione
end

for i=1:10
     a(i)=i^2;
     if i==3  %inizio if
              i %scrive i
              b=a(i)
     end %fine if
     %qualunque  altra istruzione
end
i =
    3
b =
    9
```

Funzioni grafiche:PLOT

Figure crea una figura

Plot(x,y,'str') plotta un grafico cartesiano del vettore x in fuzione del vettore y

NB: x ed y devono avere la stessa lunghezza

'str' è una stringa che setta lo stile del plottaggio

r=rosso k=nero b=blu * = plotta i punti come asterischi

o =plotta i punti come pallini -- = plotta la linea tratteggiata etc.....

Funzioni grafiche:PLOT

Semilogy (x,y,'str') plotta un grafico su scala semilogaritmica di x in fuzione di y (y ha scala logaritmica)

Semilogx (x,y,'str') plotta un grafico su scala semilogaritmica di x in fuzione di y (x ha scala logaritmica)

loglog (x,y,'str') plotta un grafico su scala logaritmica di x in fuzione di y

Hold on sovrappone i grafici succesivi
Hold off smette di sovrapporre i grafici succesivi
Clf pulisce la finestra
Axis([xmin xmax ymin ymax]) setta i limiti del grafico

esercizio

Scrivere una funzione che:

- Riceva in ingresso dt,Tfin , ω e φ
- Restituisca in uscita un vettore con 2 colonne di cui la prima contenga i tempi e la seconda la funzione sin(ω*t + φ)
- Plotti il grafico

- Opzionale
  - Se ω e φ sono vettori plotta n grafici e restituisce un vetore con n+1 colonne

Script o Function?

script

- Lavorano sul workspace di matlab
- Possono essere eseguiti a blocchi, suddivisi in più file e riuniti facilmente.
- Non possono essere compilati
- Non incapsulano i dati su cui lavorano
- Tendono a creare variabili spazzatura nel workspace
- Tendono a crescere

Script o Function?

Function

- Isolano parti di codice
- Lavorano su un workspace proprio
- Sono compilabili
- Aiutano a risparmiare memoria

- Più difficile il debug

Script o Function?

Gli script si usano:

- Per la parte portante del codice
- Per piccoli calcoli “usa e getta”
- Per la prima stesura di parti codici complessi

Function

- Ogni volta che si ha un gruppo di operazioni definite
- Per descrivere funzioni matematiche

Programmazione pragmatica

Il principio DRY: Don’t Repeat Yourself

- Usa le variabili
- Non duplicare variabili
- Non duplicare parti funzionali
- Incapsula parti di codici in funzioni o script separati
- Separa la parte di modello dagli script che ci lavorano sopra
- Commentare il codice ad alto livello

Programmazione pragmatica

Mantieni chiarezza e semplicità del codice

- Cerca di non superare 100 righe per script
- Quando fai modifiche assicurati di modificare eventuali commenti
- Metti blocchi di righe bianche per separare parti di codice con funzioni diverse
- Usa nomi di variabili descrittivi ma non troppo lunghi
- Ricordati che il codice è autoesplicativo ed i commenti possono quindi essere riservati ad osservazioni di alto livello