Guida Rapida a MATLAB per Analisi Dati

Indice

1. Introduzione a MATLAB

MATLAB (MATrix LABoratory) è un ambiente di calcolo numerico e un linguaggio di programmazione che permette di manipolare matrici, visualizzare funzioni e dati, implementare algoritmi e creare interfacce utente.

Punti di forza per una tesi ingegneristica:

- Elaborazione rapida di grandi set di dati
- Potenti strumenti di visualizzazione
- Ampia libreria di funzioni matematiche e statistiche
- Possibilità di creare script riutilizzabili

2. Concetti di Base

Interfaccia MATLAB

- **Command Window**: dove inserire i comandi uno alla volta
- **Editor**: dove scrivere e salvare script completi (.m)
- **Workspace**: dove visualizzare le variabili in memoria
- **Current Folder**: dove navigare tra i file
- **Command History**: storico dei comandi eseguiti

Avvio e Chiusura di MATLAB

```
% Per avviare MATLAB: fare doppio clic sull'icona
% Per chiudere MATLAB: digitare exit o quit
exit  % oppure
quit
```

Comandi Base

```
clc     % Pulisce la Command Window
clear   % Rimuove tutte le variabili dal workspace
clear x % Rimuove solo la variabile x
close   % Chiude la figura corrente
close all % Chiude tutte le figure
help funzione % Mostra l'aiuto per una funzione
doc funzione  % Apre la documentazione completa
```

Creare e Salvare Script

1. Nell'Editor, crea un nuovo script: File > New > Script
2. Scrivi il codice MATLAB
3. Salva con File > Save As... (nomescript.m)
4. Esegui con il pulsante "Run" o premendo F5

3. Strutture Dati Principali

Vettori e Matrici

```
% Vettore riga
v = [1, 2, 3, 4]     % o anche v = [1 2 3 4]

% Vettore colonna
v_col = [1; 2; 3; 4]  % o anche v_col = v'

% Matrice 2x3
A = [1, 2, 3; 4, 5, 6]

% Matrice con valori specificati
zeros(3)     % matrice 3x3 di zeri
ones(2,4)    % matrice 2x4 di uni
eye(3)       % matrice identità 3x3
rand(3,2)    % matrice 3x2 di numeri casuali tra 0 e 1
```

Sequenze e Spaziature

```
% Creare sequenze
1:5         % da 1 a 5 con passo 1: [1 2 3 4 5]
```

```
1:0.5:3     % da 1 a 3 con passo 0.5: [1 1.5 2 2.5 3]
linspace(0, 10, 5)  % 5 valori equidistanti da 0 a 10
```

Accesso agli Elementi

```
A = [1, 2, 3; 4, 5, 6; 7, 8, 9];

% Accedere agli elementi
A(2,3)      % elemento in riga 2, colonna 3 (= 6)
A(2,:)      % tutta la riga 2 (= [4 5 6])
A(:,1)      % tutta la prima colonna (= [1; 4; 7])
A(1:2,2:3)  % sottomatrice (righe 1-2, colonne 2-3)
```

Tipi di Dati Speciali

```
% NaN (Not a Number) per valori mancanti
dati = [1, 2, NaN, 4]

% Stringhe e Caratteri
s = 'testo'
s = "testo"  % stringhe (dalla versione R2016b)

% Celle (per dati eterogenei)
C = {1, 'testo', [1,2,3]}
C{1}   % accede al contenuto della prima cella (= 1)
```

4. Operazioni Fondamentali

Operazioni Matematiche

```
% Operazioni elementari
a = 5 + 3   % addizione
b = 5 - 3   % sottrazione
c = 5 * 3   % moltiplicazione
d = 5 / 3   % divisione
e = 5 ^ 3   % potenza
f = 5 \ 3   % divisione inversa (3/5)

% Operazioni su matrici
A = [1, 2; 3, 4];
B = [5, 6; 7, 8];

C = A + B   % addizione elemento per elemento
D = A * B   % moltiplicazione matriciale
E = A .* B  % moltiplicazione elemento per elemento
```

```
F = A .^ 2  % elevamento a potenza elemento per elemento
G = A'      % trasposta di A
```

Operazioni Logiche

```
% Confronti
a = 5 > 3   % maggiore (true)
b = 5 < 3   % minore (false)
c = 5 == 3  % uguale (false)
d = 5 ~= 3  % diverso (true)
e = 5 >= 3  % maggiore o uguale (true)
f = 5 <= 3  % minore o uguale (false)

% Operatori logici
and(a, b)   % AND logico (o a & b)
or(a, b)    % OR logico (o a | b)
not(a)      % NOT logico (o ~a)
```

Strutture di Controllo

```
% If-Else
if condizione
    % codice se condizione è vera
elseif altra_condizione
    % codice se altra_condizione è vera
else
    % codice se nessuna condizione è vera
end

% For
for i = 1:5
    % codice da ripetere 5 volte
end

% While
while condizione
    % codice da ripetere finché la condizione è vera
end
```

5. Importazione ed Esportazione Dati

Importare Dati

```
% Da file CSV
dati = readtable('file.csv');  % importa come tabella
dati = csvread('file.csv');    % importa come matrice (solo numeri)
```

```
% Da file Excel
dati = readtable('file.xlsx'); % importa come tabella
[dati, headers] = xlsread('file.xlsx'); % importa come matrice
```

Esportare Dati

```
% In CSV
writetable(tabella, 'output.csv');
csvwrite('output.csv', matrice);

% In Excel
writetable(tabella, 'output.xlsx');
xlswrite('output.xlsx', matrice);
```

Inserire Dati Manualmente

```
% Inserimento diretto nel codice
dati = [1, 2, 3; 4, 5, 6];

% Input dell'utente
x = input('Inserisci un valore: ');
```

6. Visualizzazione dei Dati

Grafici di Base

```
% Grafico a linee
x = 0:0.1:10;
y = sin(x);
plot(x, y);
title('Grafico del seno');
xlabel('x');
ylabel('sin(x)');
grid on;

% Aggiungere più curve
hold on;
plot(x, cos(x), 'r--');  % linea rossa tratteggiata
legend('sin(x)', 'cos(x)');
hold off;
```

Tipi di Grafici Comuni

```
% Grafico a barre
bar(dati);

% Grafico a barre raggruppate
bar(dati, 'grouped');

% Grafico a dispersione
scatter(x, y);

% Istogramma
histogram(dati);

% Grafico a torta
pie(dati);

% Grafico 3D
mesh(X, Y, Z);
surf(X, Y, Z);
```

Personalizzazione dei Grafici

```
% Colori e stili delle linee
plot(x, y, 'r--', 'LineWidth', 2);  % linea rossa tratteggiata spessa

% Aggiungere testo
text(x_pos, y_pos, 'Etichetta');

% Aggiungere frecce o linee
annotation('arrow', [x1 x2], [y1 y2]);

% Sottografici
subplot(2, 1, 1);  % divide la figura in 2 righe, 1 colonna, seleziona il
primo grafico
plot(x1, y1);
subplot(2, 1, 2);  % seleziona il secondo grafico
plot(x2, y2);
```

Salvare Grafici

```
% Salvare come immagine
saveas(gcf, 'figura.png');  % gcf = current figure handle
saveas(gcf, 'figura.pdf');
saveas(gcf, 'figura.fig');  % formato MATLAB per riaprire

% Esportazione ad alta risoluzione
print('figura', '-dpng', '-r300');  % 300 dpi
```

7. Analisi Statistica

Statistiche Descrittive

```
media = mean(dati);          % media
mediana = median(dati);      % mediana
moda = mode(dati);           % moda
dev_std = std(dati);         % deviazione standard
varianza = var(dati);        % varianza
min_val = min(dati);         % valore minimo
max_val = max(dati);         % valore massimo
range_val = range(dati);     % range (max - min)
iqr_val = iqr(dati);         % range interquartile
```

Gestione di Valori NaN

```
% Calcolare statistiche ignorando NaN
media = nanmean(dati);
dev_std = nanstd(dati);

% Rimuovere NaN da un vettore
dati_filtrati = dati(~isnan(dati));
```

Coefficiente di Variazione

```
% Formula: (deviazione standard / media) * 100
cv = (std(dati) / mean(dati)) * 100;
```

8. Risoluzione dei Problemi Comuni

Errori Comuni e Soluzioni

1. **Undefined function or variable**: Controlla il nome della variabile o funzione, potrebbe essere scritto male.
2. **Index exceeds matrix dimensions**: Stai cercando di accedere a un elemento che non esiste nella matrice.
3. **Matrix dimensions must agree**: Le dimensioni delle matrici non sono compatibili per l'operazione.

Tecniche di Debug

```
% Visualizzare il valore di una variabile
disp(variabile);
```

```
% Stampare un messaggio
fprintf('Il valore di x è: %.2f\n', x);

% Inserire un breakpoint (cliccando a sinistra del numero di riga)
% poi eseguire lo script e usare Step, Continue, ecc.

% Verificare dimensioni di una matrice
size(A)
```

9. Consigli Pratici per la Tesi

Organizzazione del Codice

1. **Struttura in sezioni**: Usa `%%` per dividere lo script in sezioni.
2. **Commenti chiari**: Documenta bene ogni passaggio con `%`.
3. **Script modulari**: Dividi il codice in più file per funzionalità diverse.
4. **Funzioni**: Crea funzioni riutilizzabili per operazioni ripetitive.

Esempio di Struttura di un File

```
%% TITOLO E DESCRIZIONE DEL PROGRAMMA
% Autore: Nome Cognome
% Data: 17/05/2025
% Descrizione: Questo script analizza...

%% INIZIALIZZAZIONE
clc; clear; close all;
% Definizione costanti e parametri

%% CARICAMENTO DATI
% Lettura dei dati...

%% ELABORAZIONE
% Algoritmi di calcolo...

%% VISUALIZZAZIONE RISULTATI
% Grafici e output...

%% SALVATAGGIO RISULTATI
% Export dei risultati...
```

Ottimizzazione

1. **Pre-allocazione**: Allocare array prima di cicli `for`.
2. **Vettorizzazione**: Usare operazioni vettoriali invece di cicli quando possibile.
3. **Profilazione**: Usare `tic` / `toc` o `profile` per identificare colli di bottiglia.

10. Risorse Utili

Documentazione Ufficiale

- Documentazione MATLAB
- File Exchange (librerie condivise dalla community)

Riferimenti Rapidi

- Operatori MATLAB
- Funzioni grafiche
- Funzioni statistiche

Community e Supporto

- MATLAB Answers
- Stack Overflow - Tag MATLAB

Appendice: Esempio di Codice Completo

```
%% ANALISI DI DATI SPERIMENTALI
% Questo esempio mostra come importare, analizzare e visualizzare dati

% Inizializzazione
clc; clear; close all;

% Generazione di dati di esempio (sostituire con l'importazione reale)
x = linspace(0, 10, 100);
y_teorico = 2 * x + 1;
y_misurato = 2 * x + 1 + 0.5 * randn(1, 100);  % aggiunge rumore

% Calcolo statistiche
errore = y_misurato - y_teorico;
media_errore = mean(errore);
dev_std_errore = std(errore);
rmse = sqrt(mean(errore.^2));

% Visualizzazione dei risultati
figure('Name', 'Analisi Dati', 'NumberTitle', 'off', 'Position', [100, 100,
800, 600]);

% Plot dei dati
subplot(2, 1, 1);
plot(x, y_teorico, 'b-', 'LineWidth', 2);
hold on;
```

```
plot(x, y_misurato, 'r.', 'MarkerSize', 8);
xlabel('x');
ylabel('y');
title('Confronto dati teorici e misurati');
legend('Modello teorico', 'Dati misurati', 'Location', 'northwest');
grid on;

% Plot dell'errore
subplot(2, 1, 2);
stem(x, errore);
xlabel('x');
ylabel('Errore');
title(sprintf('Distribuzione dell''errore (RMSE = %.4f)', rmse));
grid on;
yline(media_errore, 'r--', 'Media', 'LineWidth', 1.5);
yline(media_errore + dev_std_errore, 'g:', 'Media + σ', 'LineWidth', 1.5);
yline(media_errore - dev_std_errore, 'g:', 'Media - σ', 'LineWidth', 1.5);

% Output testuale
fprintf('Risultati dell''analisi:\n');
fprintf('  Media dell''errore: %.4f\n', media_errore);
fprintf('  Deviazione standard: %.4f\n', dev_std_errore);
fprintf('  RMSE: %.4f\n', rmse);

% Salvataggio
saveas(gcf, 'analisi_dati.png');
```