



HELMUT SCHMIDT
UNIVERSITÄT

Universität der Bundeswehr Hamburg

MATLAB - Grundlagen für Ingenieurwissenschaften

Inhaltsverzeichnis

1	Einführung	2
1.1	Was ist MATLAB?	2
1.2	Anwendungsgebiete in den Ingenieurwissenschaften	2
1.3	Die Benutzeroberfläche	2
2	Grundlegende Operationen	5
2.1	Variablendeklaration	5
2.2	Mathematische Grundoperationen	6
2.3	Komplexe Zahlen	7
2.4	Kommentare	7
3	Vektoren und Matrizen	7
3.1	Erstellen von Vektoren und Matrizen	7
3.2	Zugriff auf Elemente und Indizierung	7
3.3	Matrixoperationen	7
3.4	nuetzliche MATLAB Funktionen	7
4	Programmiergrundlagen	7
4.1	Skripte	7
4.2	Funktionen	7
4.3	Schleifen	7
5	Arbeiten mit Dateien und Daten	7
5.1	Speichern und Laden von Daten	7
5.2	Importieren von Messdaten	7
5.3	Analyse und Verarbeitung von Daten	7
6	Visualisierung von Daten	7
6.1	Einfache Diagramme	7
6.2	Mehrere Kurven in einem Diagramm	7
6.3	Mehrere Diagramme in einer UEbersicht	7
6.4	Grafische Anpassungen	7
7	Anhang	7
7.1	Dokumentation in MATLAB	7
7.2	Uebersicht wichtiger MATLAB Befehle	7

1 Einführung

1.1 Was ist MATLAB?

MATLAB ist die Abkürzung für MATrix LABoratory. Zudem ist es ein interaktives, integriertes System zur Berechnung, Visualisierung oder Programmierung mathematischer Problemstellungen. Es bietet eine einfache Skriptsprache welche auf die Verarbeitung von Matrizen ausgelegt ist.

1.2 Anwendungsgebiete in den Ingenieurwissenschaften

MATLAB bietet in vielen Ingenieurwissenschaftlichen Betätigungsfeldern weitreichende Vorteile.

- Signalverarbeitung
- Regelungstechnik
- FEM-Simulation
- Schaltungsanalyse
- Bildverarbeitung
- Datenanalyse

1.3 Die Benutzeroberfläche

Command Window

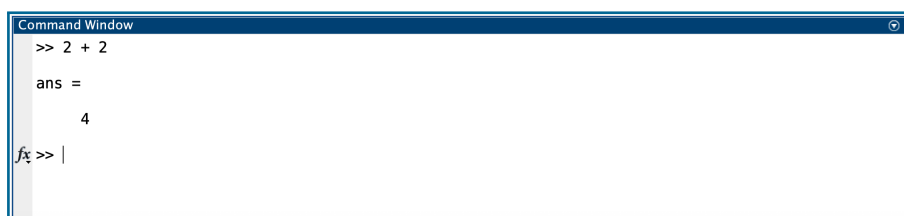
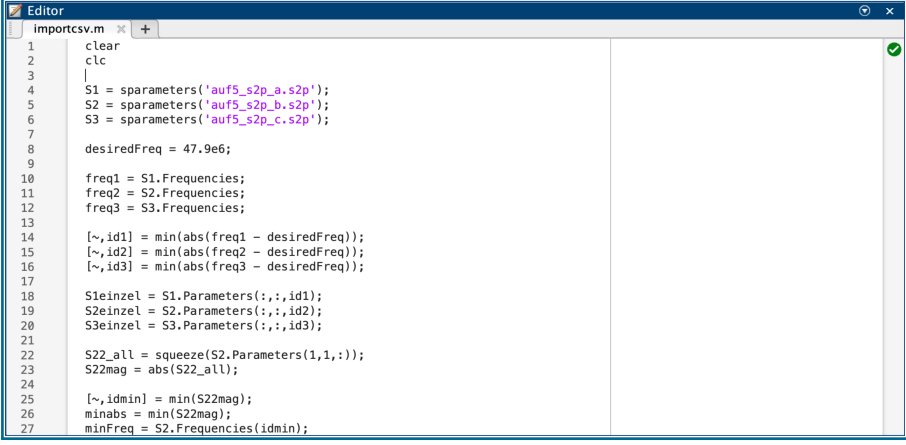


Figure 1: Command Window in MATLAB

Im Command Window können Befehle direkt eingegeben werden. Da Ergebnisse von Berechnungen unverzüglich angezeigt werden, können hier einzelne Befehle idealerweise getestet werden.

Editor

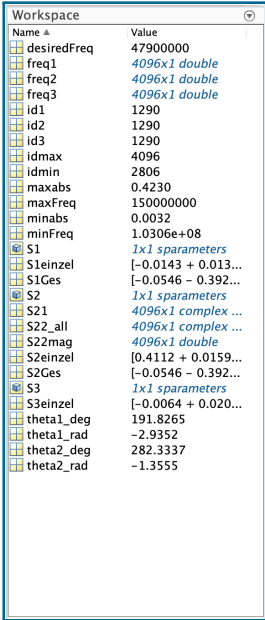


```
1 clear
2 clc
3 |
4 S1 = sparameters('auf5_s2p_a.s2p');
5 S2 = sparameters('auf5_s2p_b.s2p');
6 S3 = sparameters('auf5_s2p_c.s2p');
7
8 desiredFreq = 47.9e6;
9
10 freq1 = S1.Frequencies;
11 freq2 = S2.Frequencies;
12 freq3 = S3.Frequencies;
13
14 [~,id1] = min(abs(freq1 - desiredFreq));
15 [~,id2] = min(abs(freq2 - desiredFreq));
16 [~,id3] = min(abs(freq3 - desiredFreq));
17
18 S1einzel = S1.Parameters(:,id1);
19 S2einzel = S2.Parameters(:,id2);
20 S3einzel = S3.Parameters(:,id3);
21
22 S22_all = squeeze(S2.Parameters(1,1,:));
23 S22mag = abs(S22_all);
24
25 [~,idmin] = min(S22mag);
26 minabs = min(S22mag);
27 minFreq = S2.Frequencies(idmin);
```

Figure 2: Editor in MATLAB

Im Editor können komplette Skripte und Funktionen geschrieben, gespeichert und ausgeführt werden. Er unterstützt das Debugging mittels Breakpoints und Schritt-für-Schritt Ausführung.

Workspace



Name	Value
desiredFreq	47900000
freq1	4096x1 double
freq2	4096x1 double
freq3	4096x1 double
id1	1290
id2	1290
id3	1290
idmax	4096
idmin	2806
maxabs	0.4230
maxFreq	150000000
minabs	0.0032
minFreq	1.0306e+08
S1	1x1 sparameters
S1einzel	[-0.0143 + 0.013...
S1Ges	[-0.0546 - 0.392...
S2	1x1 sparameters
S21	4096x1 complex ...
S22_all	4096x1 complex ...
S22mag	4096x1 double
S2einzel	[0.4112 + 0.0159...
S2Ges	[-0.0546 - 0.392...
S3	1x1 sparameters
S3einzel	[-0.0064 + 0.020...
theta1_deg	191.8265
theta1_rad	-2.9352
theta2_deg	282.3337
theta2_rad	-1.3555

Figure 3: Workspace in MATLAB

Im Workspace werden alle aktuellen Variablen inklusive ihres Inhalts angezeigt. Weiterhin ist es möglich diese Variablen hier manuell anzupassen oder zu löschen.

Current Folder

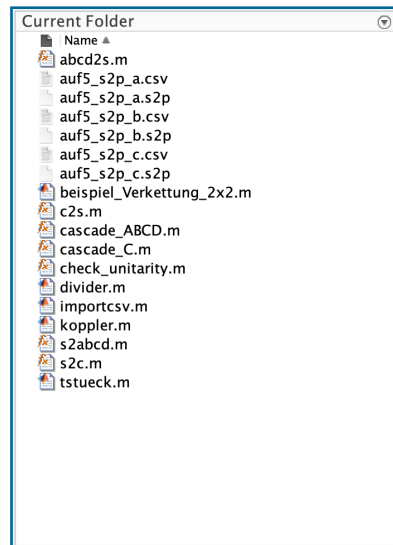


Figure 4: Current Folder in MATLAB

Im Current Folder findet man alle Dateien des Projektordners. Diese können durch Doppelklick oder das Ziehen in den Editor geöffnet und bearbeitet werden.

2 Grundlegende Operationen

2.1 Variablendeklaration

Einfache Wertzuweisung

```
a = 3;
```

Der Variable **a** wird der Wert 3 zugewiesen.

Eine Zuweisung ohne ein Semikolon am Ende der Zeile bewirkt eine direkte Rückgabe des Variablenwertes.

Fließkommazahl

```
a = 4.5;
```

Der Variable **a** wird der Wert 4.5 zugewiesen. Als Trennzeichen in MATLAB wird der Punkt an Stelle eines Kommas verwendet.

Zeichenkette

```
name = "Peter";
```

Der Variable **name** wird der String **Peter** zugewiesen.

Logischer Wert

```
isValid = true;
```

Der Variable **isValid** wird der boolsche Wert **true** zugewiesen.

Automatische Typzuweisung

```
a = pi;
```

Der Variable **a** wird die, in MATLAB vordefinierte Variable π zugewiesen.

Neben **pi** gibt es weitere vordefinierte Variablen. Diesen kann zwar ebenfalls ein selbst definierter Wert zugewiesen werden, jedoch ist es nicht empfehlenswert.

Variable	Bedeutung	Wert
inf	Unendlich	$\frac{1}{0}$ ergibt inf
i	Imaginäre Einheit	$\sqrt{-1}$
j	Alternative imaginäre Einheit	$\sqrt{-1}$
NaN	"Not a Number" - ungültiger Wert	$\frac{0}{0}$ ergibt NaN
ans	Ergebnis der letzten berechneten Zeile	z.B. ans = 42
true/false	Boolsche Werte	1 bzw. 0

2.2 Mathematische Grundoperationen

Addition

`c = a + b;`

In der Variable `c` wird die Summe aus `a` und `b` gespeichert.

Subtraktion

`c = a - b;`

In der Variable `c` wird die Differenz aus `a` und `b` gespeichert.

Multiplikation

`c = a * b;`

In der Variable `c` wird das Produkt aus `a` und `b` gespeichert.

Division

`c = a / b;`

In der Variable `c` wird der Quotient aus `a` und `b` gespeichert.

Ganzzahlige Division

`c = floor(a / b);`

In der Variable `c` wird das Ergebnis der Division von `a` und `b` gerundet zur nächsten Ganzzahl gespeichert.

Modulo

`c = mod(a / b);`

In der Variable `c` wird der Rest der Division von `a` und `b` gespeichert.

Potenzieren

`c = a ^ 2;`

In der Variable `c` wird das Ergebnis der zweiten Potenz von `a` gespeichert.

Wurzeln

`c = sqrt(a);`

In der Variable `c` wird die Wurzel von `a` gespeichert.

Betrag

`c = abs(-a);`

In der Variable `c` wird der Betrag von `-a` gespeichert.

2.3 Komplexe Zahlen

2.4 Kommentare

3 Vektoren und Matrizen

3.1 Erstellen von Vektoren und Matrizen

3.2 Zugriff auf Elemente und Indizierung

3.3 Matrixoperationen

3.4 nuetzliche MATLAB Funktionen

4 Programmiergrundlagen

4.1 Skripte

4.2 Funktionen

4.3 Schleifen

5 Arbeiten mit Dateien und Daten

5.1 Speichern und Laden von Daten

5.2 Importieren von Messdaten

5.3 Analyse und Verarbeitung von Daten

6 Visualisierung von Daten

6.1 Einfache Diagramme

6.2 Mehrere Kurven in einem Diagramm

6.3 Mehrere Diagramme in einer UEbersicht

6.4 Grafische Anpassungen

7 Anhang

7.1 Dokumentation in MATLAB

7.2 Uebersicht wichtiger MATLAB Befehle