Allgemeine Befehle

`ans`	Variable mit letztem Ergebnis
`whos x`	Eigenschaften der Variable `x` ausgeben
`clc`	Kommandoliste leeren
`clear`	Variablen löschen
`clear x`	Variable `x` löschen
`close all`	Alle Abbildungen schließen
`save Dateiname`	Alle Variablen in `Dateiname.mat` speichern
`save Datei x,y`	Variablen `x,y` in `Dateiname.mat` speichern
`load Dateiname`	Alle Variablen aus `Dateiname.mat` laden
`ver`	MATLAB Version ausgeben
`beep`	Piepen ausgeben
`help Fkt`	Gibt die Kurzhilfe für die Funktion `Fkt` aus
`doc Fkt`	Öffnet die Hilfe/Dokumentation für die Funktion `Fkt`
`tic`	Startet eine interne Stoppuhr
`toc`	Beendet eine interne Stoppuhr und gibt die Zeitdifferenz aus
`memory`	Gibt Informationen zum Speicher aus

Konstanten

`pi`	Kreiszahl π
`inf`	Aliasname für Unendlich ∞
`eps`	Aliasname für die Gleitkommazahlgenauigkeit
`2e-9`	Kurzschreibweise für 2×10^{-9}

Allgemeine Funktionen

`abs(x)`	Betrag von `x`
`sum(x)`	Summe aller Elemente in `x`
`cumsum(x)`	Kumulative Summe aller Elemente in `x`
`mean(x)`	Mittelwert aller Elemente in `x`
`diff(x)`	Differenz aller Elemente in `x`
`round(x),ceil(x), floor(x)`	Funktionen zum Runden
`sqrt(x)`	Wurzelfunktion
`log(x),log2(x), log10(x)`	Logarithmusfunktionen
`exp(x)`	Exponentialfunktion
`min(x), max(x)`	Minimum und Maximum aller Elemente in `x`
`sin(x), cos(x), tan(x)`	Winkelfunktionen
`asin(x), acos(x), atan(x)`	inverse Winkelfunktionen

Erzeugung von Variablen

`j:k`	Zeilenvektor `[j,j+1,...,k]`
`j:i:k`	Zeilenvektor `[j,j+i,...,k]`
`linspace(a,b,n)`	`n` Werte mit identischem Abstand innerhalb und einschließlich `a` und `b`.
`zeros(m,n)`	`m`×`n` Matrix mit Werten 0
`ones(m,n)`	`m`×`n` Matrix mit Werten 1
`NaN(m,n)`	`m`×`n` Matrix mit Werten NaN
`eye(n)`	Identitätsmatrix der Größe `n`×`n`
`rand(m,n)`	`m`×`n` Matrix mit gleichverteilten Zufallszahlen
`randn(m,n)`	`m`×`n` Matrix mit normalverteilten Zufallszahlen

Matrix- und Vektoroperationen

`x=[1, 2, 3]` `x=[1 2 3]`	1×3 Zeilenvektor (engl. row)
`x=[1; 2; 3]`	3×1 Spaltenvektor (engl. column)
`x=[1,2;3,4]`	2×2 Matrix
`x(2) = 4`	Element No. 2 auf 4 setzen
`x(:)`	Alle Elemente von `x`
`x(2:5)`	Elemente 2 bis 5
`x(2:2:6)`	Geradzahlige Elemente 2 bis 6
`x(j:end)`	Elemente ab `j` bis zum Ende
`x(j,:)`	Alle Zeilenelemente für Spalte `j`
`x(:,k)`	Alle Spaltenelemente für Zeile `k`
`A+B`	Elemenweise Addition
`A-B`	Elemenweise Subtraktion
`A.*B`	Elemenweise Multiplikation
`A./B`	Elemenweise Division
`A.^n`	Elemenweise Potenz
`A'`	Transponierte Matrix
`A*B`	Matrixmultiplikation
`x=A\B`	Löst das Gleichungssystem `Ax=B`
`x=A/B`	Löst das Gleichungssystem `xA=B`
`inv(A)`	Inverse Matrix
`size(A)`	Größe der Matrix
`length(A)`	Größte Dimension der Matrix
`sort(x)`	Sortiert aufsteigend
`fliplr(A)`	Matrix zeilenweise spiegeln
`flipud(A)`	Matrix spaltenweise spiegeln
`x(x>5)`	Alle Elemente größer 5 auswählen
`find(x>5)`	Indices aller Elemente größer 5
`[A,B]`	Horizontal verbinden
`[A;B]`	Vertikal verbinden

Graphische Darstellung

`figure`	Neues Abbildungsfenster
`subplot(a,b,c)`	Mehrere Abbildungen in einem Fenster
`fh=plot(x,y)`	2D Verlauf von `y` gegen `x`
`plotyy(x1,y1,x2,y2)`	2D Verlauf mit zwei y-Achsen
`loglog(x,y)`	2D Verlauf mit logarithmischen Achsen
`semilogx(x,y)` `semilogy(x,y)`	2D Verlauf mit halb-logarithmischen Achsen
`hold on`	Neue 2D Verläufe zu vorhandenen Daten hinzufügen
`hold off`	Vorhandene Verläufe beim nächsten Darstellungsbefehl ersetzen
`title('exp. data')`	Titel der Abbildung

`xlabel('time (s)')`
`ylabel('pos. (m)')`
`zlabel('force (N)')`
	Bezeichnung der Achsen

`xlim([a b])`
`ylim([a b])`
`zlim([a b])`
	Achsenbegrenzungen

`legend('text')`	Abbildungslegende

Darstellungsparameter

Können als weitere Parameter- Wertpaare z.B. mit den Funktionen `plot`, `plotyy` oder `loglog` verwendet, oder per `set(fh, ...)` nachträglich gesetzt werden.

`'LineWidth', 2`	Linienbreite auf 2
`'LineStyle', '-'`	Linientyp. Mögliche Werte: `-`, `--`, `-.`, `.`, `:`
`'Marker', '.'`	Markertyp. Mögliche Werte: `'`, `+`, `*`, `x`, `o`, `square`
`'color', 'red'`	Linienfarbe. Mögliche Werte: `red`, `blue`, `green`, `yellow`, `black` oder Vektor mit RGB Werten
`'MarkerSize', 10`	Markergröße zu `10` setzen
`'FontSize', 14`	Schriftgröße zu `14` wählen

Datenimport und -export

`xlsread(f), readtable(f)`	Einlesen von Tabellendaten (.xls)
`xlswrite(f), writetable(f)`	Schreiben von Tabellendaten (.xls)
`dlmread(f)`	Einlesen von Textdateien
`dlmwrite(f)`	Schreiben von Textdateien
`importdata(f)`	Einlesen von Dateien mit vielfältiger Konfiguration
`load(f, '-ascii')`	Einlesen von einfachen Textdateien
`save(f, 'x', '-ascii')`	Schreiben von einfachen Textdateien

Befehle zur Ausgabe

`format short`	Ausgabe mit vier Nachkommastellen
`format long`	Ausgabe mit 15 Nachkommastellen
`disp(s)`	Gibt die Zeichenkette `s` aus
`num2str(x)`	Wandelt die Zahl `x` in eine Zeichenkette um
`int2str(x)`	Wandelt den ganzzahligen Wert `x` in eine Zeichenkette um
`mat2str(x)`	Wandelt die Matrix `x` in eine Zeichenkette um
`sprintf(f, a,...)`	Wandelt Daten `a,...` gemäß der Formatvorgabe `f` in eine Zeichenkette um

Programmierung

```
function res = myfunction(a)
  ...
end
```

Definition der Funktion `myfunction` mit Parameter `a` und Rückgabewert `res`. Die Funktion kann als Datei mit dem Namen `myfunction.m` gespeichert werden. Der Funktionsname muss mit einem Buchstaben beginnen.

```
for i=1:10
  ...
end
```

Schleife zum Durchlaufen der Werte `i`= 1 bis 10

```
while(Kriterium)
  ...
end
```

Schleife zur wiederholten Ausführung solange `Kriterium` wahr ist.

```
if(Kriterium1)
  ...
elseif(Kriterium2)
  ...
else
  ...
end
```

Fallunterscheidung mit verschiedenen Kriterien.