

# Inhaltsverzeichnis

<b>Über die Autoren</b>	<b>7</b>
<b>Einleitung</b>	<b>17</b>
Über dieses Buch	17
Törichte Annahmen über die Leser	18
Die Symbole in diesem Buch	18
Über das Buch hinaus	19
Wie es weitergeht	20
<b>Teil I</b>	
<b>Erste Schritte mit MATLAB</b>	<b>21</b>
<b>Kapitel 1</b>	
<b>MATLAB mit seinen vielen Möglichkeiten kennen lernen</b>	<b>23</b>
MATLAB einordnen	23
Verstehen, wie sich MATLAB zu einer Turing-Maschine verhält	24
Über MATLAB als Taschenrechner hinausgehen	26
Herausfinden, warum Sie MATLAB brauchen	27
Die anderen MATLAB-Anwender kennenlernen	29
Das meiste aus MATLAB herausholen	30
Grundlegende Fähigkeiten am Computer erwerben	30
Die mathematischen Anforderungen festlegen	31
Anwenden, was Sie schon von anderen prozeduralen Programmiersprachen wissen	31
Verstehen, wie Ihnen dieses Buch hilft	32
Die Lernkurve erklimmen	32
<b>Kapitel 2</b>	
<b>MATLAB installieren und verwenden</b>	<b>33</b>
MATLAB installieren	33
Herausfinden, welche Plattformen MATLAB unterstützt	33
Ihr persönliches Exemplar von MATLAB erhalten	34
Die Installation durchführen	35
MATLAB aktivieren	35
Die Benutzeroberfläche kennenlernen	36
MATLAB das erste Mal starten	36
Das Anweisungsfenster verwenden	38
Das Aktuelles-Verzeichnis-Fenster verwenden	40
Das Fenster mit dem aktuellen Verzeichnis verwenden	42
Das Layout von MATLAB ändern	49

## **Kapitel 3**

### **Mit MATLAB arbeiten**

**53**

MATLAB als Taschenrechner verwenden	54
Informationen an der Eingabeaufforderung eingeben	54
Eine Formel eingeben	56
Formeln kopieren und einfügen	57
Die Formatierung des Anweisungsfensters ändern	59
Ausgaben des Anweisungsfensters unterdrücken	60
Die mathematische Syntax von MATLAB verstehen	61
Addieren, Subtrahieren, Multiplizieren und Dividieren	61
Mit Exponenten arbeiten	63
Aufbewahrungsboxen verwenden	64
Die Variable ans verwenden – die Standardaufbewahrungsbox	64
Ihre eigenen Aufbewahrungsboxen erzeugen	64
Über MATLAB als Taschenrechner hinausgehen	66
Die Wahrheit lernen	67
Eingebaute Funktionen verwenden	68
Den Funktions-Browser benutzen	68
Sich von Fehlern erholen	70
Die Fehlermeldungen von MATLAB verstehen	70
MATLAB anhalten, wenn es sich aufgehängt hat	71
Hilfe erhalten	72
Die Dokumentation erforschen	73
Sich durch die Beispiele arbeiten	73
Sich auf Mitstreiter verlassen	73
Schulungen erhalten	74
Unterstützung durch MathWorks anfragen	74

## **Kapitel 4**

### **MATLAB-Dateien erzeugen, bearbeiten und speichern**

**75**

Die Dateistruktur von MATLAB untersuchen	76
MATLAB-Dateien und ihre Aufgaben verstehen	76
Verzeichnisse mit der Benutzeroberfläche (GUI) erforschen	77
Verzeichnisse mit der Kommandozeile erforschen	81
In MATLAB mit Dateien arbeiten	86
Auf MATLAB-Dateien zugreifen und sie teilen	88
Dateien öffnen	89
Dateien importieren	90
Dateien exportieren	92
Ihre Arbeit sichern	93
Variablen mit der Benutzeroberfläche speichern	93
Variablen mit Anweisungen speichern	94
Anweisungen mithilfe der Benutzeroberfläche speichern	94
Anweisungen mithilfe von Anweisungen speichern	94

## **Teil II**

### **Daten in MATLAB bearbeiten und grafisch darstellen**

**97**

#### **Kapitel 5**

##### **Vektoren, Matrizen und höhere Dimensionen verarbeiten**

**99**

Mit Vektoren und Matrizen arbeiten	99
MATLABs Sicht auf lineare Algebra verstehen	100
Daten eingeben	101
Addieren und Subtrahieren	107
Die vielen Wege der Multiplikation und Division verstehen	108
Skalare Multiplikation und Division betreiben	108
Matrixmultiplikation anwenden	109
Matrizen dividieren	113
Potenzen von Matrizen berechnen	115
Elementweise arbeiten	116
Komplexe Zahlen verwenden	117
Mit der Exponentialfunktion arbeiten	119
In höheren Dimensionen arbeiten	120
Eine mehrdimensionale Matrix erzeugen	120
Auf eine mehrdimensionale Matrix zugreifen	122
Einzelne Elemente einer Matrix ersetzen	124
Mehrere Elemente einer Matrix als Bereich ersetzen	124
Die Matrixgröße verändern	126
Die Datenstrukturen cell array und structure verwenden	127
Die Matrix-Hilfe verwenden	131

#### **Kapitel 6**

##### **Die Grundlagen der Grafik verstehen**

**135**

Grafik in MATLAB verstehen	135
Was Sie mit Grafiken tun können	136
MATLAB-Grafik mit den Diagrammen einer Tabellenkalkulation vergleichen	136
Eine Grafik mithilfe von Anweisungen erstellen	137
Eine Grafik mithilfe des Arbeitsbereichsfensters erstellen	138
Eine Grafik mithilfe des Reiters Plots erstellen	140
Die Funktion plot() verwenden	142
Mit Linienfarbe, Symbolen und Linientyp arbeiten	143
Mehrere Grafiken mit einer Anweisung erzeugen	144
Grafiken verändern	144
Einfache Veränderungen vornehmen	145
Einer Grafik eine weitere hinzufügen	147
Eine Grafik löschen	148
Mit Untergrafiken arbeiten	149
Zweidimensionale Informationen darstellen	153

<b>Kapitel 7</b>	
<b>Komplexe Grafikfunktionen verwenden</b>	<b>155</b>
Dreidimensionale Informationen darstellen	155
Mit der Funktion <code>bar()</code> ein flaches 3D-Balkendiagramm erzeugen	156
Mit der Funktion <code>bar3()</code> ein räumliches 3D-Diagramm erzeugen	160
Die Funktion <code>barh()</code> und weitere verwenden	162
Ihre Grafiken aufwerten	164
Einen Zeiger auf die Achsen erhalten	164
Achsenbeschriftungen verändern	165
Eine Überschrift hinzufügen	166
Beschriftungen drehen	168
Anmerkungen hinzufügen	169
Ihre Grafik ausdrucken	171
<b>Teil III</b>	
<b>MATLAB effizient nutzen</b>	<b>173</b>
<b>Kapitel 8</b>	
<b>Abläufe automatisieren</b>	<b>175</b>
Verstehen, was Skripte tun	176
Ihr Arbeitspensum reduzieren	176
Festlegen, wann Sie ein Skript verwenden	177
Ein Skript erstellen	177
Ihr erstes Skript schreiben	177
Anweisungen für Benutzereingaben verwenden	180
Kopieren und in ein Skript einfügen	181
Die Anweisungshistorie in ein Skript umwandeln	182
Lange Zeichenketten fortsetzen	182
Ihren Skripten Kommentare hinzufügen	184
Skripte überarbeiten	188
Skripte aufrufen	190
Skripte schneller machen	190
Skripte nach Fehlern durchsuchen	191
<b>Kapitel 9</b>	
<b>MATLAB mit Funktionen weiter ausreizen</b>	<b>195</b>
Mit eingebauten Funktionen arbeiten	196
Etwas über eingebaute Funktionen lernen	196
Daten übergeben und herausbekommen	201
Eine Funktion erzeugen	202
Den Unterschied zwischen einem Skript und einer Funktion verstehen	203
Die Unterschiede zwischen eingebauten und eigenen Funktionen verstehen	204

Ihre erste Funktion schreiben	204
Die neue Funktion verwenden	207
Daten an die Funktion übergeben	208
Daten von der Funktion erhalten	210
Globale Variablen erzeugen und verwenden	212
Unterfunktionen verwenden	214
Funktionen verschachteln	215
Weitere Funktionstypen verwenden	216
Inline-Funktionen verwenden	217
Anonyme Funktionen verwenden	218

## ***Kapitel 10***

### ***Skripte strukturieren*** **219**

Entscheidungen treffen	219
Die if-Anweisung verwenden	220
Die switch-Anweisung verwenden	225
Das besondere an switch verstehen	226
Sich zwischen if und switch entscheiden	227
Rekursive Funktionen erzeugen	227
Aufgaben wiederholt ausführen	231
Die for-Anweisung verwenden	232
Die while-Anweisung verwenden	232
Die Verarbeitung mit break beenden	233
Die Verarbeitung mit return beenden	234
Die richtige Schleife aussuchen	236
Menüs erzeugen	236

## ***Teil IV***

### ***Fortgeschrittene Methoden in MATLAB anwenden*** **239**

#### ***Kapitel 11***

##### ***Daten importieren und exportieren*** **241**

Daten importieren	242
Grundlegende Schritte beim Import	242
Gemischte Zahlen und Texte importieren	247
Das Feldtrennzeichen festlegen	250
Nur ausgewählte Zeilen und Spalten importieren	251
Daten exportieren	252
Grundlegende Schritte beim Export	252
Skripte und Funktionen exportieren	256
Mit Bildern arbeiten	257
Bilder exportieren	257
Bilder importieren	258

<b>Kapitel 12</b>	
<b><i>Ihre Ergebnisse drucken und publizieren</i></b>	<b>261</b>
Mit Anweisungen Texte formatieren	261
Das Schriftbild beeinflussen	262
Sonderzeichen verwenden	270
Mathematische Symbole verwenden	272
Ihre MATLAB-Daten publizieren	276
Fortgeschrittene Techniken beim Publizieren von Skripten und Funktionen	277
Abbildungen speichern	279
Ihre Arbeit drucken	282
Die Ausgabeseite konfigurieren	283
Die Daten drucken	284
<b>Kapitel 13</b>	
<b><i>Sich von Fehlern erholen</i></b>	<b>285</b>
Mit Fehlermeldungen umgehen	286
Auf Fehlermeldungen reagieren	286
Die Klasse MException verstehen	288
Fehler- und Warnmeldungen einsetzen	290
Modi für Warnmeldungen einstellen	292
Fehlermeldungen per E-Mail verschicken	293
Hilfe für bekannte Fehlermeldungen suchen	294
Ihre eigenen Fehlermeldungen schreiben	296
Die maßgeschneiderte Fehlermeldung entwerfen	296
Nützliche Fehlermeldungen erstellen	301
Guten Programmierstil anwenden	302
<b>Teil V</b>	
<b><i>Spezialeinsätze für MATLAB</i></b>	<b>305</b>
<b>Kapitel 14</b>	
<b><i>Gleichungen lösen und Nullstellen finden</i></b>	<b>307</b>
Mit der Symbolic Math Toolbox arbeiten	307
Eine Version der Toolbox erhalten	308
Die Symbolic Math Toolbox installieren	310
Mit der grafischen Benutzeroberfläche arbeiten	315
Eine einfache Anweisung im Anweisungsfenster eingeben	318
Algebraische Aufgaben lösen	319
Zwischen numerischer und symbolischer Algebra unterscheiden	320
Quadratische Gleichungen lösen	322
Mit kubischen und anderen nichtlinearen Gleichungen arbeiten	323
Interpolation verstehen	323

Mit Statistik umgehen	325
Deskriptive Statistik verstehen	326
Robuste Statistik verstehen	330
Die Methode der kleinsten Quadrate anwenden	331

## ***Kapitel 15***

### ***Analysis und lineare Algebra einsetzen*** **337**

Mit linearer Algebra arbeiten	337
Mit Determinanten arbeiten	338
Matrizen reduzieren	338
Eigenwerte berechnen und verwenden	340
Faktorzerlegungen berechnen und verstehen	341
Mit Analysis arbeiten	342
Differentialrechnung betreiben	342
Integralrechnung betreiben	343
Analysis mit mehreren Variablen betreiben	344
Differentialgleichungen lösen	346
Numerische Methoden anwenden	346
Geschlossene Lösungen finden	347

## ***Kapitel 16***

### ***Richtig schicke Diagramme erzeugen*** **349**

Verstehen, was eine Supergrafik ausmacht	350
Eine Grafik mit Extras aufwerten	351
Die Funktion <code>grid()</code> verwenden	351
Die aktuelle Achse mithilfe des Zeigers <code>gca</code> ermitteln	351
Achsen mit der Funktion <code>datetick()</code> beschriften	352
Diagramme mit der Funktion <code>colorbar()</code> aufwerten	356
Mit <code>daspect</code> arbeiten	359
Mit <code>pbaspect</code> arbeiten	362
Grafik-Routinen verwenden	364
Fehlerbalken mit der Funktion <code>errorbar()</code> erzeugen	364
Mit <code>pareto()</code> ein Paretodiagramm erstellen	365
Mit <code>stairs()</code> ein Treppendiagramm darstellen	366
Mit <code>stem()</code> Daten als Stiel von einer Grundlinie darstellen	367
Mit <code>fill()</code> Bilder zeichnen	368
Mit <code>quiver()</code> Geschwindigkeitsfelder darstellen	370
Mit <code>feather()</code> Geschwindigkeitsfelder darstellen	371
Mit <code>compass()</code> Kompassnadeln zeichnen	372
Mit <code>polar()</code> Polarkoordinaten darstellen	372
Mit <code>rose()</code> Winkelverteilungen darstellen	373
Mit <code>spy()</code> die Besetzung von dünnbesetzten Matrizen darstellen	375
Animationen einbauen	376
Mit kleinen Filmen arbeiten	377
Mit Objekten umgehen	378
Daten aktualisieren	380

<b>Teil VI</b>	
<b>Der Top-Ten-Teil</b>	<b>381</b>
<b>Kapitel 17</b>	
<b>Die Top Ten der Anwendungsgebiete von MATLAB</b>	<b>383</b>
MATLAB im Ingenieurwesen	383
MATLAB in der Ausbildung	384
MATLAB in der linearen Algebra	385
MATLAB in der angewandten Mathematik	385
MATLAB in der Wissenschaft	386
MATLAB und die Freude an der Mathematik	386
MATLAB und die Forschung	386
MATLAB und Simulationen	387
MATLAB und Bildverarbeitung	388
MATLAB und Programmentwicklung	388
<b>Kapitel 18</b>	
<b>Zehn Wege, um von MATLAB zu leben</b>	<b>391</b>
Grüne Technologien umsetzen	392
Blindgänger finden	392
Spracherkennungssoftware entwickeln	393
Krankheiten unter Kontrolle bekommen	393
Designer für Computerchips werden	394
Lastkraftwagen entwickeln	394
In die Produktentwicklung gehen	395
Ausrüstung für Einsätze unter freiem Himmel entwickeln	395
Die Familienplanung unterstützen	395
Mit Simulation Risiken reduzieren	396
<b>Anhang A MATLAB-Funktionen</b>	<b>397</b>
<b>Anhang B Grafikroutinen in MATLAB</b>	<b>407</b>
<b>Stichwortverzeichnis</b>	<b>413</b>