

# MATLAB

für Studierende und Professionals  
der Sozial- und Wirtschaftswissenschaften

PROF. DR. THORSTEN PODDIG  
PROF. DR. ARMIN VARMAZ  
PD DR. CHRISTIAN FIEBERG  
BENJAMIN M. ABDEL-KARIM

# **MATLAB für Studierende und Professionals der Sozial- und Wirtschaftswissenschaften**



# **MATLAB für Studierende und Professionals der Sozial- und Wirtschaftswissenschaften**

**Thorsten Poddig    Armin Varmaz    Christian Fieberg  
Benjamin M. Abdel-Karim**

1. Auflage

**BOD – Books on Demand**

**Bibliografische Information der Deutschen Nationalbibliothek:**

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.dnb.de> abrufbar.

©2020 Thorsten Poddig, Armin Varmaz, Christian Fieberg, Benjamin M. Abdel-Karim

Herstellung und Verlag: BoD – [Books on Demand](#), Norderstedt

Gestaltung des Buchcovers: Timo Grahl, grow W erbeagentur GmbH

ISBN: 978-3-752666-39-7

# Über das Buch

Viele einführende Lehrbücher in MATLAB decken ähnliche Inhalte ab und dieses Buch stellt keine Ausnahme dar. Ein Blick in die Inhaltsangabe verdeutlicht diesen Punkt: dieses Buch stimmt inhaltlich weitestgehend mit traditionellen Lehrbüchern überein. Das besondere Merkmal dieses Buches sind nicht die (universellen) Inhalte. Das Buch zeichnet sich durch die innovative Art und Weise aus, wie die Inhalte der Leserin vermittelt werden. Wir, das Autorenteam, haben diesen innovativen Ansatz der Lehrvermittlung jahrelang an Hochschulen und in Praxisseminaren erfolgreich eingesetzt und sind überzeugt, einen ausgezeichneten Ansatz entwickelt zu haben, den Leserinnen<sup>1</sup> die Grundlagen der Programmierung mit MATLAB zu vermitteln. Viele Erfahrungen und Ansätze aus unseren Veranstaltungen ohne einen wöchentlichen Termin („Lernen auf Distanz“) sind in Erstellung dieses Lehrbuches geflossen.

## Innovationen im Ansatz

Jede interessierte Leserin kann die Grundlagen der Programmierung erlernen. Programmieren mit MATLAB ist nicht schwieriger als der gewöhnliche Lehrinhalt, der in verschiedenen fachbezogenen Einführungskursen an Hochschulen oder in zahlreichen Workshops angeboten wird. Die notwendigen Vorkenntnisse gehen nicht weiter als die Nutzung einer Tabellenkalkulationssoftware. Bei der Programmierung wird eine gedankliche Zerlegung der anstehenden Aufgabe in Teilschritte vorgenommen. Analog haben wir das Erlernen der Programmierung mit MATLAB in Teilaufgaben gegliedert, die graduell die Komplexität des Themas, und damit auch die Fähigkeiten der Leserin, steigern. Dabei haben wir den Fokus auf die konkrete Anwendung in typischen Fallstudien gelegt. Die Fallstudien sind nicht fachbezogen, bedürfen keiner Vorkenntnisse und sind häufig aus dem „Leben“ (z.B. den Zeitschriften) entnommen. Erst dann haben wir, falls überhaupt notwendig, die abstrakten Theorien erläutert.

Das Lehrbuch startet mit Themen und Problemstellungen, die sich einfach lösen lassen. Darauf aufbauend schreitet das Buch zu komplexeren, und realitätsnahen, Themen und Problemstellungen voran, deren Lösung komplexere Programmierschritte erfordert. Der Lernfortschritt folgt einem logischen Ablauf und berücksichtigt die Herausforderungen, vor denen eine typische Leserin steht. Beispielsweise beginnt das Buch mit einer detaillierten Anleitung der MATLAB-Installation und geht über zu einer Beschreibung der grafischen Benutzeroberfläche. Gerade diese ersten Schritte können viele Probleme erzeugen, die sich einfach beheben lassen. In dieser Struktur des Buches bauen die Kapitel auf den erlernten Konzepten der Vorkapitel auf. Diese stetige Steigerung der Komplexität erlaubt der Leserin, das Erlernte wieder (und immer wieder) zu

---

<sup>1</sup> Sämtliche Personenbezeichnungen gelten für alle Geschlechter. Aus Gründen der besseren Lesbarkeit verzichten wir auf die gleichzeitige Verwendung unterschiedlicher Sprachformen.

benutzen, die Kompetenzen zu festigen und die Auswirkungen von neuen Konzepten besser zu verstehen.

Nach unserer Erfahrung lernen die Leserinnen am besten, wenn sie einem konkreten Fallbeispiel folgen. Auch wir selbst versuchen neue Ideen zunächst in vereinfachten Beispielen zu verstehen. Dabei gilt: je einfacher ein Beispiel, desto besser. Diesem einfachen Prinzip folgen wir im Buch. Anstelle einer „Vogelperspektive“, die mit der Darstellung der Theorie beginnt, erläutern wir das zu lösende (vereinfachte) Problem zuerst. Diese vereinfachten Beispiele überführen wir im nächsten Schritt zu übergeordneten Konzepten und wieder im nächsten Schritt zu komplexeren Beispielen usw. Jeder einzelne Schritt ist an sich einfach zu verstehen. Die vielen einfachen Schritte führen in der Summe zum sicheren Umgang mit MATLAB.

Das Lösen von vereinfachten Fragestellungen mithilfe der Fallbeispiele dient der Vermittlung von Kompetenzen in der Programmierung. Wir versuchen den Leserinnen zu zeigen, wie sie sich einer Problemstellung annähern und diese lösen können. Das Lösen einer Fragestellung führt zum besseren Verständnis über die Methoden der Programmierung – und diese Methoden können universell zur Lösung neuer, bisher unbekannter Probleme angewendet werden. Viele Leserinnen werden unterschiedliche Bildungspfade und somit unterschiedliches Hintergrundwissen entlang ihres Ausbildungsweges erworben haben. Aus unserer Erfahrung mit anderen Lehrmaterialien werden beim Erlernen von MATLAB unnötig Lücken im Fachwissen der Leserinnen durch die Fallstudien der Lehrenden sichtbar, die zu Frustrationen führen. Die Frustrationen haben zwar an sich nichts mit MATLAB zu tun, bewirken aber oft ein Aufgeben der Leserinnen. Daher ist dieses Buch in sich abgeschlossen. Ein besonderes, fachliches bzw. themenbezogenes Vorwissen in Sozialwissenschaften und/oder Wirtschaftswissenschaften ist nicht notwendig.

Neben der besonderen Art der Beschreibung durch die vereinfachten Fallbeispiele zeichnet sich dieses Buch auch durch die begleitenden Lehrvideos aus. Bei den Lehrvideos handelt es sich nicht um Aufnahmen von Frontalvorlesungen. Vielmehr haben wir die (kurzen) Lehrvideos speziell für das Lehrbuch an einem Rechner aufgenommen. Dadurch sehen, hören und beobachten die Leserinnen, wie wir verschiedene Problemstellungen in MATLAB angehen. Der Zweck der Lehrvideos ist vielfältig. Erstens, zahlreiche Möglichkeiten und Limitationen von MATLAB lassen sich viel eindrucksvoller zeigen als in einem Text beschreiben. Beispielsweise ist es deutlich lehrreicher (und weniger frustrierend), eine Fehlermeldung, ihre Ursache und ihre Behebung in einem Lehrvideo zu sehen, als darüber zu lesen. Weil Fehlermeldungen so frustrierend sind, haben wir ihnen immer wieder Lehrvideos gewidmet. Zweitens, die Lehrvideos greifen ausgewählte Aspekte aus dem Buch auf und vertiefen diese. In diesem Sinne sind die Lehrvideos eine *Ergänzung* zum Buch. Drittens, beinhalten die Lehrvideos auch eigenständige Inhalte (seltener) und eigenständige Fallstudien (häufiger), die über die Inhalte und Fallstudien des Buches hinausgehen. In diesem Sinne sind diese Lehrvideos *Erweiterungen* des Buches. Die erweiterten Fallstudien gehen teilweise weit über die Fallstudien aus dem Buch hinaus. An der einen oder anderen Stelle werden die Videos als schwer, gerade im Vergleich zum Buch, empfunden. Das ist Absicht und zielt auf die Leserinnen ab, die ihre Kenntnisse in MATLAB im Rahmen von komplexeren Fallstudien vertiefen wollen. Viertens, aus unserer Erfahrung gelingt das Erreichen der Lernergebnisse deutlich besser, wenn die Leserinnen die Inhalte nicht nur lesen, sondern auch beobachten, sehen und hören sowie zusätzlich selbst ausführen. Frei nach dem chinesischen Sprichwort: „Ich höre und vergesse, ich sehe und behalte, ich handle und verstehe.“

Fünftens, die Lehrvideos sind permanent. Sie können wiederholt und jederzeit, losgelöst von den zeitlichen und räumlichen Limitationen einer gewöhnlichen Vorlesung, durchgearbeitet werden.

Ein integraler, erweiternder Bestandteil der Lehrvideos ist die Darstellung der „typischen Fehler“. Aus unserer Erfahrung stehen 99% der Leserinnen (und wir sind dabei keine Ausnahme gewesen) immer wieder vor denselben Problemen. Für den reibungslosen Einstieg sorgen wir durch das explizite Hervorrufen dieser Fehler in den Lehrvideos, durch das Interpretieren der Fehlermeldungen und schließlich durch das Beheben der Fehler. Gerade zu Beginn handelt es sich um Kleinigkeiten, z.B. falsche Schreibweisen, falsche aktuelle Ordner etc., die aus Unerfahrenheit der Leserinnen zu Frustrationen und manchmal zum Aufgeben führen. Die Fehler passieren allen und sind nicht auf die Person der Leserin zurückzuführen!

Wir glauben fest an das Sprichwort: „Übung macht den Meister“. Daher beinhaltet das Buch (und die Online-Lernumgebung) eine Vielzahl von Aufgaben, die didaktischen Überlegungen folgen, das Lernen der Programmierung mit MATLAB zu erleichtern. Wir haben vier Arten von Aufgaben:

**Online-Quizzes** sind einfache, kurze Aufgaben zu Definitionen, die der Leserin helfen, das faktische Wissen in den Ebenen Erinnern und Verstehen zu sichern. Die Rückmeldung zu den Lernergebnissen erfolgt unmittelbar in der Online-Lernumgebung.

**Aufgaben im Text** innerhalb eines Kapitels sind eher einfache, kurze Aufgaben, in denen wir das faktische Wissen der Leserin in den Ebenen Erinnern, Verstehen und Anwenden fördern. Die beispielhaften Lösungen der Aufgaben im Text werden am Ende des jeweiligen Kapitels dargestellt.

**„End-of-Chapter-Aufgaben“** am Ende eines Kapitels sind kompliziertere und etwas längere Aufgaben, die das konzeptionelle Wissen in den Ebenen Verstehen, Anwenden und Analysieren testen. Die beispielhaften Lösungen zu diesen Aufgaben erhalten nur Dozenten für die einführenden MATLAB-Kurse.

**Online-Aufgaben** umfassen auch (aber nicht ausschließlich) längere Aufgaben, die das prozedurale Wissen der Leserin in den Ebenen Verstehen, Anwenden, Analysieren und Evaluieren unterstützen. Die Rückmeldung zu den Lernergebnissen erfolgt unmittelbar in der Online-Lernumgebung.

Die graduell steigende Komplexität, verbunden mit dem Lernen in inhaltlich kleinen Schritten, das Sehen der Umsetzung von Fallstudien in Lehrvideos, das eigenständige Umsetzen der Fallstudien begleitet durch Lehrvideos sowie das Lösen von vielen Aufgaben mit praktischem Einsatz von MATLAB sind nach unserer Meinung die Kernelemente in der Kompetenzvermittlung. Dieser innovative Ansatz ist das Besondere unseres Buches, der unser Buch von anderen einführenden Lehrbücher maßgeblich unterscheidet.

## Lernumgebung

Für die begleitenden Lehrvideos, die Online-Quizzes, die Online-Aufgaben sowie Lernbegleitmaterialien bedürfen wir einer Online-Umgebung, in der wir zusätzlich und reibungslos mit



den Leserinnen kommunizieren können. Neben vielen „Non-Profit“- und privaten „For-Profit“-Organisationen haben wir uns mit „ILIAS“ (bzw. „AULIS“) für eine „Open-Source“-Lösung entschieden. Unter

<https://bit.ly/2UtZKf5>

haben wir einen Online-Kurs eingerichtet und alle Lernmaterialien (Online-Quizzes, Online-Aufgaben, MATLAB-Codes aus dem Buch, MATLAB-Codes als Lösungen der Aufgaben, frei verfügbare Daten aus dem Buch, Folien zu den Lehrvideos sowie Vorlesungsfolien für die Dozenten) dort zusammengetragen. Die Lernmaterialien haben wir kapitelweise und übersichtlich in Online-Modulen organisiert. Zusätzlich haben wir im Anhang A eine Tabelle mit der Zuordnung von MATLAB-Dateien aus den Begleitmaterialien und den Quellcodes aus dem Buch erstellt. Aus unserer Erfahrung erzielen die Leserinnen die besten Lernergebnisse, wenn sie diesem idealisierten Verlauf folgen:

1. Lesen eines Buchkapitels
2. Selbst-Ausprobieren der mitgelieferten MATLAB-Codes aus dem Buchtext
3. Lösen der Aufgaben im Text
4. Anschauen und Durcharbeiten der ergänzenden Online-Lehrvideos
5. Lösen der Online-Quiz-Aufgaben
6. Lösen der Online-Aufgaben
7. Lösen der „End-of-Chapter-Aufgaben“

Zur Unterstützung der Leserinnen haben wir in der Lernumgebung Online-Module eingerichtet, die das Lernen im idealtypischen Verlauf fördern. Natürlich lernen alle individuell. Aus diesem Grund achten wir durchgängig auf die identische Bezeichnung der Lernelemente im Buch, in den Lehrvideos und in der Online-Umgebung. Daher kann jede Leserin ohne Nachteile von dem idealtypischen Verlauf abweichen und wird jederzeit wissen, worauf im Buch und in den Lehrvideos verwiesen wird.

Der Zugang zur Lernumgebung mit allen Lernmaterialien ist für alle frei. In der Lernumgebung wird sogar eine Online-Version des Buches verfügbar sein. Wir sind überzeugt, durch unsere Art der Lehrvermittlung deutlich mehr Leserinnen von den Vorzügen von MATLAB überzeugen zu können. Dazu sind wir gerne bereit, neben der klassischen Print-Variante auch eine Online-Version anzubieten. Die Online-Variante des Buches stellen wir unter

[www.matlab-intro.de](http://www.matlab-intro.de)

zum Online-Lesen bereit.

## Organisation des Buches

Dieses Buch ist kein reines Referenzbegleitbuch zu MATLAB oder ein einfaches MATLAB-Tutorial. Es ist eher ein einführendes Lehrbuch in Programmierung, das MATLAB nutzt, um die unterschiedlichen Konzepte und Methoden zu illustrieren. Daher wird die Leserin nicht nur eine solide Kenntnis von MATLAB erwerben, sondern auch Grundlagen der Programmierung.

Die Lernmaterialien dieses Buches haben wir in vielen Seminaren an der Universität Bremen, Universität Oldenburg, an der Hochschule Bremen und in zahlreichen Seminaren in Kooperation mit Uhlenbruch erprobt. Auch aufgrund der dortigen Diskussionen gelangen wir zu der Einsicht, dass nicht die Inhalte per se unverständlich sind, sondern die Art der Kompetenzvermittlung. Daher stellt der nachfolgende Streifzug durch die Inhalte des Buches keine Überraschung im Vergleich zu anderen Lehrbüchern dar. Das Buch beginnt mit der Installation und der Beschreibung der MATLAB-Benutzeroberfläche und endet mit dem Datenimport und -export und einer abschließenden empirischen Studie:

**Installation und Programmeinstieg** Das Kapitel erläutert den Installationsprozess und gibt einen Schnellüberblick über die wichtigsten Elemente des MATLAB-Systems und der MATLAB-Programmierung.

**Matrizenoperationen und Indexierung** Das Kapitel führt in die Grundlagen der Nutzung von Matrizen in MATLAB ein. Neben der obligatorischen linearen Algebra werden in diesem Kapitel auch die Möglichkeiten von Indexierung aufgezeigt, um auf Teilbereiche einer Matrix zugreifen zu können.

**Die Datentypen** Die grundlegenden Datentypen wie Double, String, Boolean und Arrays werden in diesem Kapitel eingeführt und ihre Nutzung und Einsatzgebiete erarbeitet.

**Grafik erstellen** Dieses Kapitel beschäftigt sich mit der Visualisierung von Daten und Ergebnissen. Dort werden die häufig genutzten Grafik-Typen wie Liniendiagramme, Häufigkeitsdiagramme, Punktwolken sowie Kuchen- und Säulendiagramme vorgestellt.

**Datenstrukturen** Dieses Kapitel stellt spezielle Datentypen vor. Diese Datenstrukturen sind für den effizienten Einsatz in spezialisierten Einsatzgebieten verfügbar.

**Kontrollstrukturen** Die Nutzung von if- und for-Strukturen (und einige weitere), sicherlich einige der bekanntesten Konzepte der Programmierung, werden in diesem Kapitel eingeführt.

**Datenimport und Datenexport** Das Kapitel beschreibt den Import empirischer Daten, die die Grundlage jeder empirischen Analyse darstellen. Der Schwerpunkt liegt auf dem Import von Excel-Dateien.

**Anreizwirkungen in der Nachfrage nach Gesundheitsvorsorge** Das Kapitel führt keine neuen Konzepte ein. Es stellt vielmehr dar, wie die bisherigen Kompetenzen und Fähigkeiten auf eine umfangreiche Fallstudie übertragen werden können. Dazu wird auf einen empirischen Datensatz aus der Literatur zurückgegriffen, mit dem die Ergebnisse der Literatur repliziert werden.

## Das Begleitprogramm

Unser Lehrbuch behandelt die notwendigsten Grundlagen für einen einführenden Kurs zu MATLAB. Das hält das Buch angenehm kurz. Alle anderen Materialien, je nach Vorkenntnis mehr oder weniger bedeutend für die eigenen Bedürfnisse, haben wir in der Online-Umgebung aufbereitet. Die zentralen Bestandteile der Online-Umgebung sind die Online-Quizzes, Online-Aufgaben sowie die Online-Videos.

Bei den Online-Videos haben wir mehrheitlich darauf geachtet, die Doppelung der Inhalte auf ein Minimum zu reduzieren. Die Online-Videos sind eine Ergänzung des Buches und können dieses, gerade für die Einsteigerinnen, nicht ersetzen. Vielmehr sollen die Online-Videos das Buch ergänzen und erweitern. Die Online-Videos behandeln in der Regel komplexere Themen als das Buch. Als Beispiel seien die Videos begleitend zum Kapitel 2 genannt. Diese Online-Videos haben den Charakter eines Tutorials „MATLAB in 90 Minuten lernen“. Dort werden die MATLAB-Elemente eingehender beschrieben und in herausfordernden Fallstudien im Vergleich zum Buchkapitel eingesetzt. Wenn Sie Probleme haben die Videos zum Kapitel 2 vollständig nachzuvollziehen, liegt es nicht an Ihnen, sondern am Charakter der Videos. Dort wird ein Schnelleinstieg beschrieben, um die Möglichkeiten von MATLAB, im Sinne von einem „Appetizer“, aufzuzeigen. Die dortigen Inhalte werden in den nachfolgenden Kapiteln vertieft.

Bremen, Frankfurt am Main, Dezember 2020

Thorsten Poddig

Armin Varmaz

Christian Fieberg

Benjamin M. Abdel-Karim

# Inhaltsverzeichnis

Abbildungsverzeichnis	V
Tabellenverzeichnis	VII
Verzeichnis der Quellcodes	VIII
<b>I. Installation, Kennenlernen, erste Grundlagen</b>	<b>1</b>
<b>1. Einleitung</b>	<b>2</b>
1.1. Gegenstandsbereich dieses Buches . . . . .	2
1.2. Aufbau und Zielsetzung . . . . .	3
1.2.1. Matrizenorientierte Programmiersprachen . . . . .	4
1.2.2. Warum MATLAB? . . . . .	5
1.2.3. Warum Octave? . . . . .	6
<b>2. Die Installation und der Programmeinstieg</b>	<b>8</b>
2.1. Die Studenten-Version von MATLAB . . . . .	8
2.2. Die Installation von MATLAB . . . . .	10
2.2.1. Erster Schritt: Account-Erstellung und Lizenzerwerb . . . . .	10
2.2.2. Zweiter Schritt: Aktivierung von MATLAB . . . . .	10
2.3. Die graphische Benutzeroberfläche . . . . .	13
2.3.1. Menüleiste . . . . .	14
2.3.2. Current Folder . . . . .	14
2.3.3. Workspace . . . . .	15
2.3.4. Konsole . . . . .	15
2.4. Befehle und Ausdrücke . . . . .	15
2.5. Das Skript . . . . .	17
2.6. Einfache Benutzer-Funktionen . . . . .	20
2.7. Exkurs: Codeausführung in Matlab . . . . .	25
2.8. Fallstudie Wetterdaten . . . . .	26
2.9. Zusammenfassung - Installation und Programmeinstieg . . . . .	28
<b>3. Matrizenoperationen und Indexierung</b>	<b>31</b>
3.1. Was ist eine Matrix? . . . . .	31
3.2. Indexierung von Matrizen . . . . .	34

3.3.	Elementare Matrizenoperationen . . . . .	36
3.3.1.	Voraussetzungen . . . . .	36
3.3.2.	Die Matrizenaddition . . . . .	38
3.3.3.	Die Matrizensubtraktion . . . . .	38
3.3.4.	Elementenweise Matrizenmultiplikationen . . . . .	39
3.3.5.	Elementenweise Matrizendivisionen . . . . .	40
3.3.6.	Reguläre Matrizenmultiplikation . . . . .	41
3.3.7.	Lösen von Gleichungssystemen: Die „Matrizendivision“ . . . . .	42
3.4.	Elementare Matrizen-Funktionen . . . . .	44
3.4.1.	Nullmatrix . . . . .	44
3.4.2.	Ones-Befehl . . . . .	45
3.4.3.	Not-a-Number . . . . .	45
3.4.4.	Zufallszahlen . . . . .	46
3.5.	Arrays . . . . .	48
3.6.	Fallstudie Wetterdaten . . . . .	49
3.7.	Zusammenfassung - Matrizen . . . . .	52
3.8.	End-of-Chapter-Aufgaben . . . . .	57

## II. Programmierung in MATLAB 59

4.	Die Datentypen <span style="float: right;">60</span>
4.1.	Zahlen in der Programmierung . . . . . <span style="float: right;">61</span>
4.2.	Der Zahlendatentyp Double . . . . . <span style="float: right;">62</span>
4.2.1.	Repräsentation von Zahlen durch Zahlendatentypen . . . . . <span style="float: right;">62</span>
4.2.2.	Der Aufbau des Double . . . . . <span style="float: right;">63</span>
4.3.	Zeichen und Zeichenketten in der Programmierung . . . . . <span style="float: right;">66</span>
4.4.	Die Wahrheitswerte . . . . . <span style="float: right;">75</span>
4.5.	Fallstudie Wetterdaten . . . . . <span style="float: right;">81</span>
4.6.	Zusammenfassung - Datentypen . . . . . <span style="float: right;">81</span>
4.7.	End-of-Chapter-Aufgaben . . . . . <span style="float: right;">87</span>
5.	Grafiken erstellen <span style="float: right;">89</span>
5.1.	Grundlagen der graphischen Darstellung . . . . . <span style="float: right;">89</span>
5.1.1.	Diagramme erstellen . . . . . <span style="float: right;">90</span>
5.1.2.	Diagramme individuell anpassen . . . . . <span style="float: right;">92</span>
5.2.	Weitere Visualisierungswerkzeuge in MATLAB . . . . . <span style="float: right;">96</span>
5.2.1.	Häufigkeitsdiagramm . . . . . <span style="float: right;">96</span>
5.2.2.	Scatterplot . . . . . <span style="float: right;">98</span>
5.2.3.	Kreisdiagramme . . . . . <span style="float: right;">100</span>
5.2.4.	Das Säulendiagramm . . . . . <span style="float: right;">101</span>
5.3.	Zusammenfassung - Grundlagen der Visualisierung . . . . . <span style="float: right;">103</span>
5.4.	End-of-Chapter-Aufgaben . . . . . <span style="float: right;">104</span>

<b>6. Datenstrukturen</b>	<b>106</b>
6.1. Cell-Arrays	106
6.1.1. Eindimensionales Cell-Array	107
6.1.2. Mehrdimensionales Cell-Array	109
6.2. String-Arrays	111
6.3. Struct	114
6.3.1. Einstufiges Struct	114
6.3.2. Mehrstufiges Struct	115
6.4. Struct-Arrays	117
6.5. Tabellen	118
6.5.1. Gewöhnliche Tabellen	118
6.5.2. Zeittabellen (Time Tables)	124
6.6. Fallstudie Wetterdaten	128
6.7. Zusammenfassung - Datenstrukturen	131
6.8. End-of-Chapter-Aufgaben	135
<b>7. Kontrollstrukturen</b>	<b>137</b>
7.1. Die Verzweigungen	138
7.1.1. Die if-then-else-Bedingungen	138
7.1.2. Verschachtelte if-then-else-Bedingungen	140
7.1.3. Die switch-case-otherwise-Anweisung	142
7.2. Die Schleifen	144
7.2.1. Die for-Schleife	144
7.2.2. Die while-Schleife	146
7.2.3. Die while-Schleife mit einem Break	148
7.2.4. Die verschachtelte Schleife	150
7.3. Die try-catch-Anweisung	152
7.4. Die Aufwandsbetrachtung und Programmcodeoptimierung	156
7.5. Fallstudie Wetterdaten	159
7.5.1. Daten zerlegen	159
7.5.2. Analysedatensatz zusammenstellen	161
7.5.3. Metadaten zusammenstellen	162
7.6. Zusammenfassung - Kontrollstrukturen	163
7.7. End-of-Chapter-Aufgaben	168
<b>8. Datenimport und Datenexport</b>	<b>172</b>
8.1. Datenimport und Datenexport im Matlab-Format	172
8.2. Der grundlegende Datenimport	173
8.3. Der grundlegende Datenexport	180
8.4. Datenimport und Datenexport von Tabellen	181
8.5. Datenimport und Datenexport von Zeittabellen	182
8.6. Exkurs: Datenimport und Datenexport mit benutzerdefinierten Funktionen	184
8.7. Tipps für das Verfassen eines Quellcodes	186
8.7.1. Benennungskonventionen in der Software-Entwicklung	187

---

8.7.2. Quellcode-Dokumentation . . . . .	188
8.7.3. Die Ordnerstruktur anlegen . . . . .	191
8.8. Fallstudie Wetterdaten . . . . .	192
8.9. Zusammenfassung - Datenimport und Datenexport . . . . .	194
8.10. End-of-Chapter-Aufgaben . . . . .	196
<b>9. Anreizwirkungen in der Nachfrage nach Gesundheitsvorsorge</b>	<b>198</b>
9.1. Die Einführung in die Fallstudie . . . . .	198
9.2. Daten importieren . . . . .	200
9.3. Arzt- und Krankenhausbesuche: Deskriptive Statistiken . . . . .	202
9.4. Deskriptive Statistiken für abhängige und unabhängige Variablen . . . . .	205
9.5. Bedingte Erwartungswerte der Arzt- und Krankenhausbesuche . . . . .	208
9.6. Korrelation vs. Kausalität . . . . .	211
9.7. Zusammenfassung - Die Fallstudie . . . . .	230
<b>Literaturverzeichnis</b>	<b>232</b>
<b>A. Tabelle der Dateien in den Begleitmaterialien</b>	<b>235</b>
<b>B. Octave</b>	<b>239</b>
B.1. Installation . . . . .	239
B.2. Bedienoberfläche . . . . .	240
<b>C. Die Befehlssammlung</b>	<b>241</b>
<b>Das Register</b>	<b>244</b>

# Abbildungsverzeichnis

1.1. Der Aufbau des Buches . . . . .	3
2.1. Der Aktivierungsprozess-Schritt 1. . . . .	10
2.2. Der Aktivierungsprozess-Schritt 2. . . . .	11
2.3. Der Aktivierungsprozess-Schritt 3. . . . .	11
2.4. Der Aktivierungsprozess-Schritt 4. . . . .	12
2.5. Der Aktivierungsprozess-Schritt 5. . . . .	12
2.6. Die Benutzeroberfläche von MATLAB R2020a unter Mac OS X . . . . .	13
2.7. Die Menüleiste . . . . .	14
2.8. Der Current Folder . . . . .	14
2.9. Die Konsole . . . . .	15
2.10. Der MATLAB-Editor - Skripterstellung im Editor. . . . .	18
2.11. Die Speicherung des Skripts . . . . .	19
2.12. Die Ausführung des Skripts . . . . .	20
3.1. Die Grundregel für das Rechnen mit vollständigen Matrizen . . . . .	38
4.1. Die Datentypen in der Übersicht . . . . .	61
4.2. Der 64 Bit Datentyp Double. . . . .	64
5.1. Linienchart des Temperaturverlaufs an der Wetterstation 72. . . . .	92
5.2. Linienchart des Temperaturverlaufs an der Wetterstation 72. . . . .	94
5.3. Häufigkeitsdiagramm der Lufttemperatur an der Wetterstation 72, 200 Monate, mittlere Monatstemperaturen. . . . .	97
5.4. Scatterplot der Lufttemperatur (abhängige Variable) gegen die Sonnenstunden (unabhängige Variable) an der Wetterstation 72. . . . .	99
5.5. Scatterplot der Lufttemperatur (abhängige Variable) gegen die Sonnenstunden (unabhängige Variable) an der Wetterstation 72 mit der zusätzlichen Darstellung der Regressionsgerade. . . . .	100
5.6. Anteil von Monaten mit Minusgraden vs. Monaten mit Plusgraden an der Wetterstation 72 als Kreisdiagramm. . . . .	102
5.7. Verlauf der mittleren Monatstemperaturen in den ersten 12 Monaten als Säulendiagramm. . . . .	103
6.1. Das eindimensionale Array . . . . .	107
6.2. Das mehrdimensionale Array . . . . .	110



7.1. Die Kontrollstrukturen in der Übersicht . . . . .	138
7.2. Das Funktionsprinzip der <i>if-then-else</i> -Anweisung . . . . .	139
7.3. Das Funktionsprinzip der verschachtelten <i>if-then-else</i> -Anweisung . . . . .	141
7.4. Das Funktionsprinzip der <i>for</i> -Schleife . . . . .	145
7.5. Das Funktionsprinzip der <i>while</i> -Schleife . . . . .	147
7.6. Das Funktionsprinzip der <i>try-catch</i> -Anweisung . . . . .	153
7.7. Ausschnitt der Profiler-Übersicht für das MATLAB-Skript . . . . .	158
7.8. Die MATLAB-Quellcodestruktur . . . . .	159
8.1. Fiktive Kursdaten (genauer: Zufallszahlen) für den deutschen Aktienindex DAX. . . . .	175
8.2. CSV-Datei im Matlab-Editor. . . . .	178
8.3. Das Ergebnis der <i>PUBLISH</i> Funktion. . . . .	191
8.4. Eine anfängliche Ordnerstruktur . . . . .	192
9.1. Anzahl der Beobachtungen (links) und Zufriedenheit mit der Gesundheit (rechts) vor und nach dem Event sowie im Jahr des Events. Oben ist der Typ 2 („Typ Scheidung“) und unten der Typ 1 („Typ Heirat“) dargestellt. . . . .	227
9.2. Abnormale Gesundheitszufriedenheit vor und nach dem Event sowie im Jahr des Events. Oben ist der Typ 2 und unten der Typ 1 dargestellt. . . . .	229

# Tabellenverzeichnis

2.1. Übersicht über einige ausgewählte MATLAB-Grundfunktionen und Operatoren. . . . .	24
3.1. Syntax der Indexierung bei Matrizen . . . . .	35
3.2. Übersicht gängiger Matrixoperationen, der Dimensionen der zu verrechnenden Matrizen sowie die Dimension der resultierenden Ergebnismatrix. . . . .	37
4.1. Beispiel der Transistoren-Konfiguration zur Darstellung der ersten vier natürlichen Zahlen . . . . .	62
4.2. Übersicht der Vergleichsoperatoren . . . . .	79
4.3. Übersicht der logischen Operatoren . . . . .	79
5.1. Zusammenstellung einiger Standardbefehle für Diagramme . . . . .	91
5.2. Zusammenstellung einiger Zusatzbefehle für Diagramme . . . . .	93
9.1. Übersicht der Variablenbezeichnung und –definition aus der Studie von RIPHahn et al. (2003). Die Bezeichnungen und Beschreibungen sind aus der Datei „Readme file“ von RIPHahn et al. (2003) unverändert entnommen. . . . .	201
9.2. Replizierte Ergebnisse aus der Table I aus RIPHahn et al. (2003). Der Aufbau der Tabellen ist identisch. . . . .	204
9.3. Replizierte Ergebnisse aus der Table II aus RIPHahn et al. (2003). Der Aufbau der Tabellen ist analog. . . . .	207
9.4. Replizierte Ergebnisse aus der Table III aus RIPHahn et al. (2003). Der Aufbau der Tabellen ist identisch. . . . .	210
9.5. Tabelle gibt die Anzahl der Beobachtungen sowie die mittlere Zufriedenheit mit der eigenen Gesundheit von ledigen und verheirateten Personen. . . . .	212
9.6. Auszug aus der Outputvariable mErg nach der Ausführung der Funktion fIdentChange. . . . .	221

# Verzeichnis der Quellcodes

2.1. Die erste Konsoleneingabe . . . . .	16
2.2. Die erste Matlab-Berechnung . . . . .	16
2.3. Multiplizieren zweier Matrizen . . . . .	18
2.4. Funktionen erstellen - Teil I . . . . .	21
2.5. Funktionen erstellen - Teil II . . . . .	22
2.6. Multiplizieren zweier Matrizen als Funktion . . . . .	22
2.7. Multiplizieren zweier Matrizen mit Funktion . . . . .	23
2.8. Die vordefinierten MATLAB-Funktionen . . . . .	24
2.9. Temperaturdaten Station 72 laden . . . . .	27
2.10. Mittlere Temperatur der Station 72 berechnen . . . . .	27
3.1. Das Anlegen einer 4 x 3 Matrix mit dem Namen mC . . . . .	32
3.2. Das Ergebnis des Anlegens der 4 x 3 Matrix mC . . . . .	32
3.3. Das Anlegen eines 4 x 1 Spaltenvektors mit dem Namen vD . . . . .	33
3.4. Das Transponieren einer 4 x 3 Matrix mit dem Namen mC . . . . .	33
3.5. Zugriff auf Zeile einer Matrix . . . . .	34
3.6. Zugriff auf Elemente einer Matrix mittels des linearen Indexierens . . . . .	35
3.7. Die Matrizenaddition . . . . .	38
3.8. Die Matrixsubtraktion . . . . .	39
3.9. Elementenweise Matrizenmultiplikationen . . . . .	39
3.10. Elementenweise Matrixdivisionen . . . . .	40
3.11. Reguläre Matrizenmultiplikation . . . . .	41
3.12. Die „Matrixdivision“ . . . . .	42
3.13. Die „Matrixlinksdivision“ . . . . .	42
3.14. Die Nullmatrix . . . . .	44
3.15. Der Ones-Befehl . . . . .	45
3.16. Der Ones-Befehl und die Skalarmultiplikation . . . . .	45
3.17. Not-a-Number . . . . .	45
3.18. Die Erstellung von Zufallszahlen zwischen 0 und 1 . . . . .	46
3.19. Die Erstellung von gleichverteilten Zufallszahlen . . . . .	47
3.20. Zufällige Erzeugung einer Korrelationsmatrix . . . . .	48
3.21. Ein 3-dimensionales Array . . . . .	48
3.22. Die Untersuchung der Schwankungsbreite der Temperaturen . . . . .	51
4.1. Wertebereich eines Doubles, Double und Single . . . . .	65
4.2. String erstellen . . . . .	68
4.3. Operationen auf Strings . . . . .	68

4.4. Indexierung auf Strings . . . . .	69
4.5. Zusammenfügen von Strings . . . . .	69
4.6. Konvertierung von Zahlen in Strings . . . . .	70
4.7. Die universelle Funktion sprintf . . . . .	70
4.8. Die Funktion fprintf . . . . .	71
4.9. Vergleichen von Strings . . . . .	71
4.10. Suchen in Strings . . . . .	72
4.11. Datumsstring in Zahlenwert konvertieren . . . . .	73
4.12. Datumswert in Datumsstring konvertieren . . . . .	73
4.13. Datumsvergleich . . . . .	74
4.14. Datumsaddition . . . . .	74
4.15. Operationen mit Logicals in MATLAB . . . . .	76
4.16. Verknüpfungsoperatoren mit Logicals in MATLAB . . . . .	76
4.17. Einfach logischer Und-Operator in MATLAB . . . . .	77
4.18. Einfach logischer Oder-Operator in MATLAB . . . . .	77
4.19. Rechnen auf logischen Matrizen . . . . .	78
4.20. Negations-Operator . . . . .	78
4.21. Indizierung mit logischen Werten . . . . .	79
4.22. Indizierung mit logischen Werten - zweite Beispiel . . . . .	80
4.23. Die Untersuchung der Schwankungsbreite der Temperaturen ohne ungültige Werte . . . . .	81
4.24. Installation von Packages in Octave . . . . .	87
5.1. Die Erstellung eines Liniendiagramms . . . . .	91
5.2. Die Erstellung eines individuell angepassten Liniendiagramms . . . . .	94
5.3. Die Erstellung eines Häufigkeitsdiagramms . . . . .	97
5.4. Die Erstellung eines Scatterplots . . . . .	98
5.5. Die Erstellung eines Kreisdiagramms . . . . .	101
5.6. Die Erstellung eines Säulendiagramms . . . . .	101
6.1. Anlegen eines Cell-Arrays . . . . .	108
6.2. Cell-Array und Matrix . . . . .	109
6.3. Das mehrdimensionale Array in der MATLAB-Darstellung . . . . .	110
6.4. Strings und skalare String-Arrays . . . . .	112
6.5. String-Array und Zugriff im String-Array . . . . .	113
6.6. Cell-Arrays zur Nachbildung von String-Arrays . . . . .	113
6.7. Das einstufige Struct . . . . .	114
6.8. Erzeugungsfunktion struct . . . . .	115
6.9. Zugriff im Struct . . . . .	115
6.10. Das mehrstufige Struct . . . . .	115
6.11. Struct-Array . . . . .	117
6.12. Anlegen einer gewöhnlichen Tabelle . . . . .	119
6.13. Zugriff in Tabelle mit Feldnamen . . . . .	119
6.14. Zugriff in Tabelle mit Index - runde Klammern . . . . .	120
6.15. Zugriff in Tabelle mit Index - geschweifte Klammern . . . . .	120

6.16. Fehlerhafte Addition von Tabellen . . . . .	121
6.17. Addition in Tabellen . . . . .	121
6.18. Addition in Tabellen mit Konvertierungsfunktionen . . . . .	121
6.19. Mittelwertberechnung in Tabellen . . . . .	122
6.20. Zeittabelle mittels Cell-Array . . . . .	124
6.21. Zeittabelle mittels Time-Table-Objekt . . . . .	124
6.22. Zugriff auf eine Zeittabelle . . . . .	125
6.23. Horizontale Verkettung . . . . .	126
6.24. Vertikale Verkettung . . . . .	126
6.25. Synchronisierung von Zeittabellen . . . . .	127
6.26. Bereinigen von Zeittabellen . . . . .	127
6.27. Konstruktion von Cell-Arrays und Tables aus Wetterdaten . . . . .	129
6.28. Datumsumwandlungen bei den Wetterdaten . . . . .	129
6.29. Datumsumwandlungen bei den Wetterdaten, Fortsetzung . . . . .	130
6.30. Datumsformat im Cell-Array ersetzen . . . . .	130
6.31. Konvertierung in time table . . . . .	130
6.32. Zusammenführen zweier Zeittabellen für die Stationen 3 und 72 . . . . .	131
7.1. Beispielskript für eine if-then-else-Kontrollstruktur . . . . .	139
7.2. Ausführung des if-then-else-Skripts . . . . .	140
7.3. Beispielskript für eine if-then-else-Kontrollstruktur . . . . .	140
7.4. Beispielskript für eine mehrfach verschachtelte Kontrollstruktur . . . . .	141
7.5. Einfaches Beispiel für eine case-Struktur . . . . .	143
7.6. Einfaches Beispiel für eine for-Schleife . . . . .	145
7.7. Einfaches Beispiel für eine while-Schleife . . . . .	147
7.8. Einfaches Beispiel für eine while-Schleife mit break . . . . .	149
7.9. Kombination von Schleifen . . . . .	150
7.10. Ausgabe der kombinierten Schleifen . . . . .	151
7.11. Try and Catch - Anweisung . . . . .	154
7.12. Try and Catch - Anweisung - Fehlermeldungen abfangen . . . . .	154
7.13. Quellcode Analyse mit dem Profiler . . . . .	157
7.14. Daten nach Stationsnummern zerlegen . . . . .	160
7.15. Analysedatensatz erstellen . . . . .	161
7.16. Metadaten ermitteln . . . . .	162
8.1. Einfaches Beispiel für den Datenimport aus Excel . . . . .	176
8.2. Einlesen und Konvertieren einer Zeittabelle . . . . .	183
8.3. Export eines Cell-Arrays als CSV-Datei . . . . .	184
8.4. Demoskript zum Export eines Cell-Arrays als CSV-Datei . . . . .	185
8.5. Datenimport aus .txt-Datei . . . . .	185
8.6. Datenexport nach .txt-Datei . . . . .	185
8.7. Demoskript für den Datenimport aus und Datenexport nach .txt-Datei . . . . .	186
8.8. Professionelle Quellcode-Dokumentation . . . . .	190
8.9. Einlesen der Wetterdaten Zugspitze . . . . .	193

8.10. Einlesen der Wetterdaten Zugspitze - Alternative . . . . .	193
9.1. Einlesen der Fallstudien­daten . . . . .	201
9.2. Skript zur Nachbildung der Tabelle 1 aus RIPHahn et al. (2003) . . . . .	202
9.3. Skript zur Nachbildung der Tabelle 2 aus RIPHahn et al. (2003) . . . . .	206
9.4. Skript zur Nachbildung der Tabelle 3 aus RIPHahn et al. (2003) . . . . .	208
9.5. Skript zur Nachbildung der Tabelle 9.5 . . . . .	212
9.6. Familienstand von Person mit der ID 6, 7 und 21 . . . . .	217
9.7. Logischer Index des Familienstandes der drei Personen aus der Stichprobe . . .	218
9.8. Ergebnis der diff-Funktion . . . . .	219
9.9. Zuordnung des Typen 1 . . . . .	220
9.10. Zuordnung des Typen 2 . . . . .	220
9.11. Funktion fIdentChange . . . . .	221
9.12. Bildung eines Event-Vektors für ein Event, Schritt 1 . . . . .	222
9.13. Bildung eines Event-Vektors für ein Event, Schritt 2 . . . . .	223
9.14. Bildung eines Event-Vektors, Schritt 3 . . . . .	223
9.15. Bildung einer Event-Matrix . . . . .	224
9.16. Funktion zur Erstellung der Event-Matrix fEventMatrix . . . . .	226
9.17. Skript zur Umsetzung der Fallstudie . . . . .	226
9.18. Skript zur Bestimmung der abnormalen Zufriedenheit in der Fallstudie . . . . .	228



## **Teil I.**

# **Installation, Kennenlernen, erste Grundlagen**



# 1. Einleitung

## 1.1. Gegenstandsbereich dieses Buches

Der Einsatz von Software ist in vielen Wissenschaftsdisziplinen, nicht nur aufgrund der kontinuierlich steigenden Menge und Komplexität an zu verarbeitenden Daten in Forschung und Lehre, kein Novum mehr. Darüber hinaus ist der Umgang mit Software in der beruflichen Praxis (z.B. in der Industrie, dem Dienstleistungssektor oder der Finanzwirtschaft) nicht mehr wegzudenken. Dieser Umstand erfordert, dass sich auch Studierende und Berufsanfängerinnen frühzeitig mit den elementaren Softwarekenntnissen vertraut machen.

Das Lehrbuch richtet sich prinzipiell an Studierende aller Fachrichtungen, die sich Grundlagen des Programms MATLAB oder Octave aneignen wollen. Das Buch adressiert vorzugsweise, aber nicht ausschließlich, Wirtschafts- und Sozialwissenschaftler im Bachelor- und Masterstudium (z.B. Wirtschaftswissenschaften, Soziologie, Politologie, Psychologie, aber auch z.B. Biologie, Medizin, etc.).

Darüber hinaus ist das Lehrbuch auch für Berufspraktikerinnen gedacht, die in ihrer Tätigkeit eine moderne und flexibel einzusetzende Software zur Analyse von Daten benötigen. Dieses Buch möchte in verständlicher Weise Studierende der Sozial- und Naturwissenschaften, Wissenschaftlerinnen und Berufspraktikerinnen ermutigen, professionelle Software zur Datenerhebung und Datenauswertung für Evaluationen, wissenschaftliche oder berufspraktische Studien, Bachelor- und Masterarbeiten o.Ä. zu nutzen. Das Erlernen der Software erfordert zwar Basiswissen in der Programmierung, was jedoch keine Hürde für die Nutzung der Software darstellen soll, da dieses Buch den Anspruch erhebt, den Umgang mit der Software fallstudienorientiert zu vermitteln. Dadurch kann das nötige Handwerkszeug, um die eigenen Vorhaben realisieren zu können, erworben werden. In diesem Buch werden zwei matrizenorientierte Programmsysteme parallel verwendet, die in Wissenschaft und Praxis sehr weit verbreitet sind: MATLAB und Octave. In diesem Buch wird vornehmlich ein Einblick in das Programm MATLAB und an geeigneter Stelle zum kostenlosen Open Source Klone Octave<sup>1</sup> vermittelt.

Dieses Lehrbuch ist als Einführungswerk zu verstehen und entstand aus der Überlegung heraus, einen möglichst einfachen Einstieg zum Fachbuch Computational Finance von PÖDTER et al. (2015) zu liefern, gleichzeitig aber auch, einen breiteren Kreis von Leserinnen anzusprechen. Im Rahmen einer ersten Einführung gibt dieses Buch einen Einblick in elementare Grundlagen der angewandten Programmierung und Statistik. Diese Kenntnisse werden anhand verschiedener Fallstudien aus unterschiedlichen thematischen Bereichen vertieft und durch zahlreichen Übungsaufgaben intensiv trainiert, da empirische Forschung mittels Software keine „Buchgelehr-

---

<sup>1</sup> Aus Gründen der Leseergonomie wird im Folgenden von MATLAB gesprochen. Sofern im Text nicht explizit darauf verwiesen wurde, ist es möglich, die Codes analog für Octave zu verwenden.

samkeit“ ist, sondern nur durch praktisches Tun erlernt werden kann. Dafür ist die eigenständige Bearbeitung aller Übungs- und Testaufgaben dringlich angeraten.

## 1.2. Aufbau und Zielsetzung

Als Grundlagenwerk hat dieses Buch den primären Anspruch, die Umsetzung von grundlegenden Methoden im Umgang mit den Programmen zu vermitteln. Daraus ergibt sich der grundlegende Aufbau dieses Buches (Abb. 1.1).

Kapitel 1: Einleitung
Kapitel 2: Installation und der Programmeinstieg
Kapitel 3: Matrizenoperationen und Indexierung
Kapitel 4: Datentypen
Kapitel 5: Grafiken erstellen
Kapitel 6: Datenstrukturen
Kapitel 7: Kontrollstrukturen
Kapitel 8: Datenimport und Datenexport
Kapitel 9: Anreizwirkungen in der Nachfrage nach Gesundheitsvorsorge
Anhang

Abbildung 1.1.: Der Aufbau des Buches. Das Buch besteht aus insgesamt neun Kapiteln und einem Anhang.

Das Lehrbuch besteht insgesamt aus 9 Kapiteln und einem Anhang. Jedes Kapitel endet mit einer Schlussbetrachtung und fasst die wesentlichen Inhalte zusammen. Im ersten Kapitel erfolgt die Einleitung. Anschließend wird der Installationsprozess von MATLAB beschrieben. In der Folge beginnt Teil I „Grundlagen der Programmierung“. In diesem Teil werden im dritten Kapitel Grundkenntnisse zu Matrizenoperationen vermittelt (siehe Kapitel 3), welche den Ausgangspunkt für das Arbeiten mit MATLAB bilden. In Kapitel 4 werden die Datentypen und in Kapitel 6 die Datenstrukturen dargestellt. Zwischen diesen beiden Kapiteln ist eine erste Einführung in die Möglichkeiten der grafischen Darstellung und Visualisierung von Datenanalysen eingefügt (siehe Kapitel 5). Es soll bereits frühzeitig illustrieren, wie einfach derartige Analysen sind und welche interessanten Darstellungsmöglichkeiten sich daraus ergeben. Daran knüpft das Kapitel 7 an, welches die verschiedenen Kontrollstrukturen vorstellt. In Kapitel 8 werden Im- und Exportverfahren für empirische Daten vorgestellt. Den Abschluss des Buches bildet das Kapitel 9, in dem die vermittelten MATLAB-Konzepte aus den vorangegangenen Kapiteln im Rahmen

einer umfangreichen Fallstudie mit der Replikation der Ergebnisse aus der Literatur angewendet werden.

Einfache Handgriffe, die im Rahmen der alltäglichen Computerhandhabung vorausgesetzt werden, können nicht explizit erklärt werden. Jedoch sind scheinbar einfache Sachverhalte der Programmierung an einigen Stellen ausführlicher erklärt und in Online-Videos gezeigt, denn eben jene Grundkenntnisse sind elementarer Bestandteil im Umgang mit MATLAB und in der Folge auch essenziell für den Praxiseinsatz.

### 1.2.1. Matrizenorientierte Programmiersprachen

In den meisten Fällen besteht der erste Ansatz im empirischen Arbeiten für das Studium in der Verwendung eines Tabellenkalkulationsprogrammes wie Microsoft Excel<sup>2</sup>. Die Gründe dafür werden vermutlich primär in der Vertrautheit und der Einfachheit im Umgang liegen. Im privaten und schulischen Bereich ist derartige Software durch ihre Verbreitung als Standardsoftwareausstattung auf den meisten PCs bekannt, zumal die eingebauten (statistischen) Funktionen für einfache Datenanalysen ausreichend erscheinen. Dazu können einfache Optimierungsprobleme durch den mitgelieferten *Solver*, einem allgemeinen Optimierungstool, gelöst werden. Als Erweiterung bietet die integrierte Programmiersprache Visual Basic for Applications (VBA) weitreichende Möglichkeiten. Mit diesem Tabellenkalkulationstool lassen sich wahrscheinlich mit ausreichend Zeit, einer großen Frustrationstoleranz und entsprechendem Ehrgeiz, eine Vielzahl an Problemen lösen. Insbesondere im Studium können die Grenzen eines Tabellenkalkulationssystems aber schnell erreicht werden, weil komplexe Probleme in annehmbarer Zeit mit möglichst großer Effizienz gelöst werden müssen. Vor diesem Hintergrund wird in diesem Werk auf den Einsatz von beispielsweise Excel und VBA konsequent verzichtet.

Allerdings stellt sich somit die Frage nach geeigneten Alternativen, beispielsweise zur Auswertung von empirischen Daten im Rahmen akademischer oder berufspraktischer Fragestellungen in unterschiedlichsten Disziplinen. Aus der Vielzahl an verfügbaren Softwarelösungen z. B. MATLAB, R, Gauss, etc. lässt sich unschwer folgern, dass die Entscheidung für eine bestimmte Software durch persönliche Vorlieben des Anwenders bestimmt wird. Bei aller Vielseitigkeit dieser Systeme lassen sich jedoch einige zentrale Gemeinsamkeiten derartiger matrizenorientierter Programmierwerkzeuge identifizieren:

- Grundlegend werden Vektoren und Matrizen als Datenstrukturen verwendet, um komplexe Rechenoperationen effizient auszuführen. Dies ist gerade für die Analyse großer Mengen empirischer Daten von Vorteil, da sich diese zumeist in natürlicher Weise in Form von Vektoren oder Matrizen organisieren und bearbeiten lassen.
- Umfangreiche Bibliotheken mit vordefinierten Funktionen und Algorithmen werden durch

---

<sup>2</sup> An dieser Stelle sei ausdrücklich angemerkt, dass die Tabellenkalkulation Microsoft Excel in Verbindung mit der eingebauten Programmiersprache Visual Basic for Applications nur als Stellvertreter für Tabellenkalkulationen mit integrierter Programmiersprache allgemein dient. Die dargestellten Überlegungen beziehen sich damit auf die gesamte Klasse dieser Werkzeuge und gelten nicht ausschließlich für Microsoft Excel.

den Hersteller und eine aktive Community<sup>3</sup> bereitgestellt. Für Anwender bedeutet dies eine erhebliche Zeit- und Arbeitsersparnis.

- Geschwindigkeitsoptimierte Basisfunktionen für größtmögliche Recheneffizienz helfen gerade bei der Verarbeitung großer Datenmengen oder aufwendigen Simulationen von (wissenschaftlichen) Modellen.
- Viele Schnittstellen für die Dateneingabe, Datenausgabe, Datenvisualisierung und teilweise auch zu Datenbanksystemen stehen zur Verfügung, sodass automatisierte Datenbankabrufe leicht implementiert werden können. Gerade immer wiederkehrende Datenanalysen können dadurch erheblich vereinfacht und beschleunigt werden.
- Zahlreiche Schnittstellen stehen bereit, um mit anderen Programmiersprachen zu kommunizieren. Dies ist von Vorteil, falls doch bestimmte Spezialprobleme nicht oder nicht effizient mit den eigenen Mitteln des Werkzeugs gelöst werden können. Außerdem werden spezielle Hardwarezugriffe bereitgestellt. Dies ermöglicht z.B. den Zugriff auf die Grafikkarten als zusätzliche Recheneinheiten, um z.B. besonders intensive Datenanalysen oder Simulationen zu beschleunigen, ohne dass dafür besondere Programmierkenntnisse von Nöten wären.

Zusammenfassend liegen die Vorteile solcher matrizenorientierter Programme in (i) den sehr viel umfangreicheren und elaborierteren vordefinierten Funktionen, (ii) den darauf aufbauenden Funktionsbibliotheken (auch Toolboxen oder Packages genannt), (iii) der hohen Flexibilität und dem Umgang mit großen Datenmengen, (iv) den mitunter einfacher erlernbaren internen Programmiersprachen im Vergleich zu denen der Tabellenkalkulationen (z.B. VBA), (v) den geschwindigkeitsoptimierten Matrizenoperationen, sowie nicht zuletzt (vi) den herausragenden Möglichkeiten zur Datenvisualisierung, die weit über die von Tabellenkalkulationen bekannten Möglichkeiten hinausgehen.

### 1.2.2. Warum MATLAB?

In Anbetracht der möglichen Alternativen zu Matlab (und Octave) erscheint eine kurze Auseinandersetzung mit den Vorzügen sinnvoll.

- MATLAB<sup>4</sup> ist eine weit verbreitete Standardsoftware in Forschung und Praxis. Gerade in vielen Wirtschaftsunternehmen sind Grundkenntnisse im Umgang mit MATLAB für die berufliche Zukunft von Vorteil. Die Verwendung dieser Software ist im Hinblick auf die angesprochene Zielgruppe eine folgerichtige Konsequenz. Dieses Werk vermittelt die hierfür erforderliche Grundlagenexpertise.

---

<sup>3</sup> Beispielhaft sei vorab der MATLAB-Fileexchange-Sever URL: <http://www.mathworks.com/matlabcentral/fileexchange/> genannt.

<sup>4</sup> Das Programm MATLAB® ist eingetragenes Warenzeichen von The Mathworks, Inc. Im weiteren Verlauf dieses Buches wird zur Vereinfachung „MATLAB“ ohne Hinweis auf das eingetragene Warenzeichen verwendet.