

Einsatz der Computer im Grundpraktikum Physik

1 Zum Umgang mit den Computern im Grundpraktikum

Die Computer im Grundpraktikum können und sollen von den Studierenden für alle Aufgaben genutzt werden, die im Zusammenhang mit den Praktikumsversuchen stehen. Eine kurze Einführung in ihre Benutzung wird während des ersten Praktikumstermins gegeben. Studierenden ohne ausreichende Kenntnisse im Umgang mit Computern wird empfohlen, möglichst bald an entsprechenden Kursen teilzunehmen (Windows, Textverarbeitung (Word oder LaTeX), Tabellenkalkulation (Excel), Präsentation (Powerpoint), eine Programmiersprache).

1.1 Anmelden am Computer

Die Computer im Grundpraktikum sind mit dem Betriebssystem Windows 10 ausgerüstet. Zur Anmeldung ist die Angabe eines *lokalen Benutzernamens* und des zugehörigen *Passworts* notwendig. Beides wird den Studierenden am Anfang des Kurses vor Ort mitgeteilt. Arbeitsverzeichnis auf den PCs im Praktikum ist eine externe Cloud, die unter dem Netzwerklaufwerk O:\ zur Verfügung gestellt wird. Im Windows-Explorer (Dateimanager) erscheint dieses Arbeitsverzeichnis unter dem Eintrag

`https://cloud.uol.de/remote.php/webdav (O:)`

Dort sind neben dem Ordner MATLAB, in dem entsprechende Matlab-Skripte¹ zur Verfügung gestellt werden, auch die Gruppenverzeichnisse GPR01, GPR02,...etc angelegt. Jedes 2er-Team (also z. B. Müller und Meier) legt dort unter dem entsprechenden Gruppenverzeichnis bei der ersten Anmeldung sein eigenes Unterverzeichnis an. Als Verzeichnisname werden die Nachnamen gewählt, also z. B. O:\Mueller_Meier.

➤ *Eigene Daten dürfen nur in diesem persönlichen Verzeichnis gespeichert werden!*

In diesem Verzeichnis muss im Laufe des Semesters eine Struktur mit Unterordnern angelegt werden, in denen später die Daten zu einzelnen Versuchen abgelegt werden, also:

- O:\Mueller_Meier
 - Uebungen_Origin
 - Oszilloskop
 - Fehlerrechnung
 - Widerstaende
 - Kapazitaeten
 - ...

Vermeiden Sie Umlaute und Leerzeichen in den Namen von Ordnern und Dokumenten. Dadurch verhindern Sie, dass es zu Problemen kommt, wenn Sie auf ein anderes Betriebssystem wechseln.

Alle Mitglieder einer Praktikumsgruppe „nn“ haben im gesamten Verzeichnis O:\ die Berechtigung zum Lesen und Schreiben und damit auch zum Löschen von Daten. Dauerhafte Datensicherheit kann also nicht gewährleistet werden. Um persönliche Daten dauerhaft zu sichern, sollten sie deshalb auf einem eigenen USB-Speicherstick oder in einem persönlichen Verzeichnis im Hochschulnetz gespeichert werden. Ein solches Verzeichnis wird von der Abteilung IT-Dienste² der Universität für alle Studierenden automatisch angelegt. Näheres dazu erfährt man bei der Abteilung IT-Dienste und über die Person, die die Computer im CIP-Raum³ des Instituts für Physik betreut.

¹ Die Matlab-Skripte werden Laufe des Praktikums bei verschiedenen Versuchen benötigt. Einzelheiten dazu sind den entsprechenden Versuchsanleitungen der Teilversuche zu entnehmen.

² <http://www.uni-oldenburg.de/itdienste/>

³ Im CIP-Raum des Instituts für Physik (W2 2-249) stehen den Studierenden mehrere Computer zur Verfügung. Einzelheiten dazu unter <https://uol.de/physik/cip/>

1.2 Drucker im Praktikum

Im Grundpraktikum steht ein Netzwerkdrucker (HP LaserJet Pro MFP M426fdw, schwarz/weiß) zur Verfügung. Der Drucker kann zum Ausdruck der Dokumente genutzt werden, die im Rahmen des Grundpraktikums erstellt werden.

1.3 Schutz vor Computer-Viren

Die Studierenden im Grundpraktikum können für die Sicherung ihrer eigenen Daten USB-Speichersticks verwenden. Dabei ist sicherzustellen, dass keine Computer-Viren auf die PCs übertragen werden. Im Zweifelsfall muss der Datenträger vor Verwendung mit der Antivirensoftware Sophos überprüft werden.

2 Auswahl der zur Verfügung stehenden Software

Neben den Windows-Standardprogrammen sind übliche Programme zur Textverarbeitung (Word), zur Tabellenkalkulation (Excel), zur Präsentation (Powerpoint) und Internetbrowser (Firefox) auf den Computern im Grundpraktikum verfügbar. Darüber hinaus stehen die Programme Origin und Matlab zur Verfügung. Sie sind für die Datenaufnahme, Datenanalyse und Datenvisualisierung sowie für allgemeine Funktionsberechnungen und Darstellungen von Funktionsgraphen besonders geeignet und im technisch-wissenschaftlichen Bereich weit verbreitet. Beide Programme sind auch auf den Computern im CIP-Raum des Instituts für Physik verfügbar. Eine Nutzung auf dem eigenen PC ist ebenfalls möglich. Informationen dazu finden sich hier:

<https://uol.de/en/physics/laboratory-courses/origin> (Origin)

<https://uol.de/physik/praktika/matlab> (Matlab).

Die folgenden Kurzanleitungen können und sollen keine Handbücher ersetzen, sondern lediglich Einstiegshinweise geben, die für die Lösung der jeweiligen exemplarischen Aufgaben ausreichend sind. Weitere Hinweise werden vor Ort gegeben. Bei den folgenden Beschreibungen wird vorausgesetzt, dass grundlegende Kenntnisse im Umgang mit Windows-Programmen vorhanden sind.

3 Origin

Das Programm Origin (Version 2019) wird im Praktikum eingesetzt, um Messdaten in Tabellen einzugeben, Berechnungen mit den Daten durchzuführen, grafische Darstellungen der Daten zu erzeugen, Parameter von Ausgleichsgeraden durch Messwerte zu berechnen (*lineare Regression*) und nichtlineare Funktionsfits durchzuführen. Die in Kap. 3.3.3 beschriebenen Punkte werden erst im späteren Verlauf des Praktikums benötigt. Es wird daher empfohlen, zu gegebener Zeit erneut einen Blick in diesen Text zu werfen.⁴

3.1 Start von Origin, Grundeinstellungen

Nach dem Start von Origin erscheint eine Bildschirmoberfläche ähnlich wie in Abb. 1 dargestellt. Die Anzahl und die Position geöffneter Fenster und Symbolleisten hängen von den persönlichen Einstellungen ab. Über

- Ansicht bzw.
- Ansicht → Symbolleisten

lassen sich die Einstellungen den individuellen Bedürfnissen anpassen.⁵

In dem Startfenster erscheint, ähnlich wie beim Programm EXCEL, ein Fenster mit einem leeren Arbeitsblatt (*Sheet1*) einer Arbeitsmappe (*Book1*). In dieses Arbeitsblatt werden die Messdaten eingetragen, aus denen anschließend Diagramme erzeugt werden. Möglicherweise werden später weitere Arbeitsblätter und Diagramme, Notizen, Berechnungen usw. ergänzt. All diese Daten werden von Origin zu einem *Projekt* zusammengefasst, das als Ganzes in einer Datei mit der Endung **.opj (origin project)** oder mit der Endung **.opju** abgespeichert werden kann:

⁴ Weitere Unterlagen zu Origin (Getting Started, Tutorials, Help,...) finden sich im Download-Bereich der Seite <http://www.originlab.com/>.

⁵ Die Lage einer Symbolleiste (oben, unten, seitlich) kann, wie bei Windows-Programmen üblich, verändert werden, indem die Symbolleiste bei gedrückter linker Maustaste an die gewünschte Position gezogen wird.

→ Datei → Projekt speichern.

Es wird empfohlen, die Datei mit der Endung .opj zu speichern, um die Kompatibilität mit älteren Origin-Versionen zu gewährleisten.

Das in Abb. 1 unten dargestellte Fenster des *Projekt-Explorers* enthält eine Übersicht aller zu einem *Projekt* gehörenden Daten. Es lässt sich wie folgt sichtbar machen:

→ Ansicht → Projekt Explorer

Die Lage und Größe des Fensters kann wie üblich eingestellt werden.

Über das Menü

→ Hilfe → Sprache ändern bzw.
→ Help → Change Language

kann zwischen der deutschen und englischen Sprachversion von Origin umgeschaltet werden.

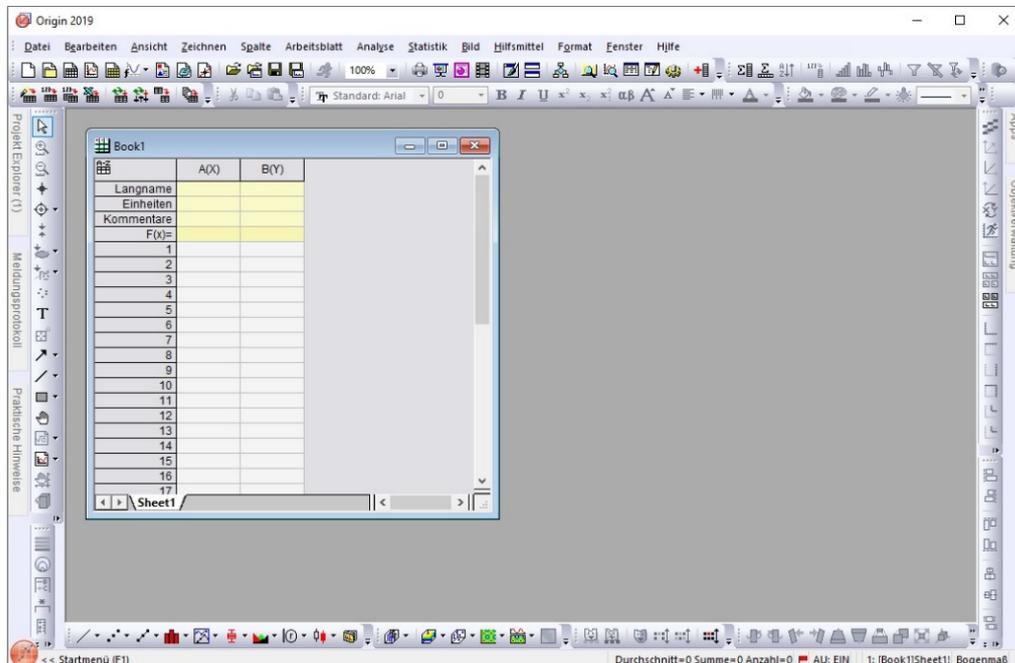


Abb. 1: Ausschnitt der Bildschirmoberfläche nach dem Starten des Programms Origin (Version 2019).

3.2 Einstellung des Dezimalzeichens

Version 2019 von Origin erlaubt die Umschaltung zwischen einer deutschen und englischen Sprachversion. Dadurch kann es zu Mehrdeutigkeiten bei der Interpretation des Dezimaltrennzeichens (Dezimalkomma bzw. Dezimalpunkt) kommen, die u.U. zu scheinbar unerklärlichen Fehlern führen. Um solche Probleme zu vermeiden, dürfen bei Verwendung von Origin nicht die *Regions- und Sprachoptionen* aus Windows übernommen werden, sondern es muss explizit der *Dezimalpunkt* als Trennzeichen eingestellt werden. Diese Einstellung erfolgt gem. Abb. 2 über

→ Hilfsmittel → Optionen → Zahlenformat → Trennzeichen → 1,000.0
Trennzeichen für ASCII-Import → 1,000.0

Der Punkt in der Angabe 1,000.0 ist das Dezimaltrennzeichen, das Komma nur eine visuelle Hilfe zur Hervorhebung von Tausender-Blöcken (englische Notation).

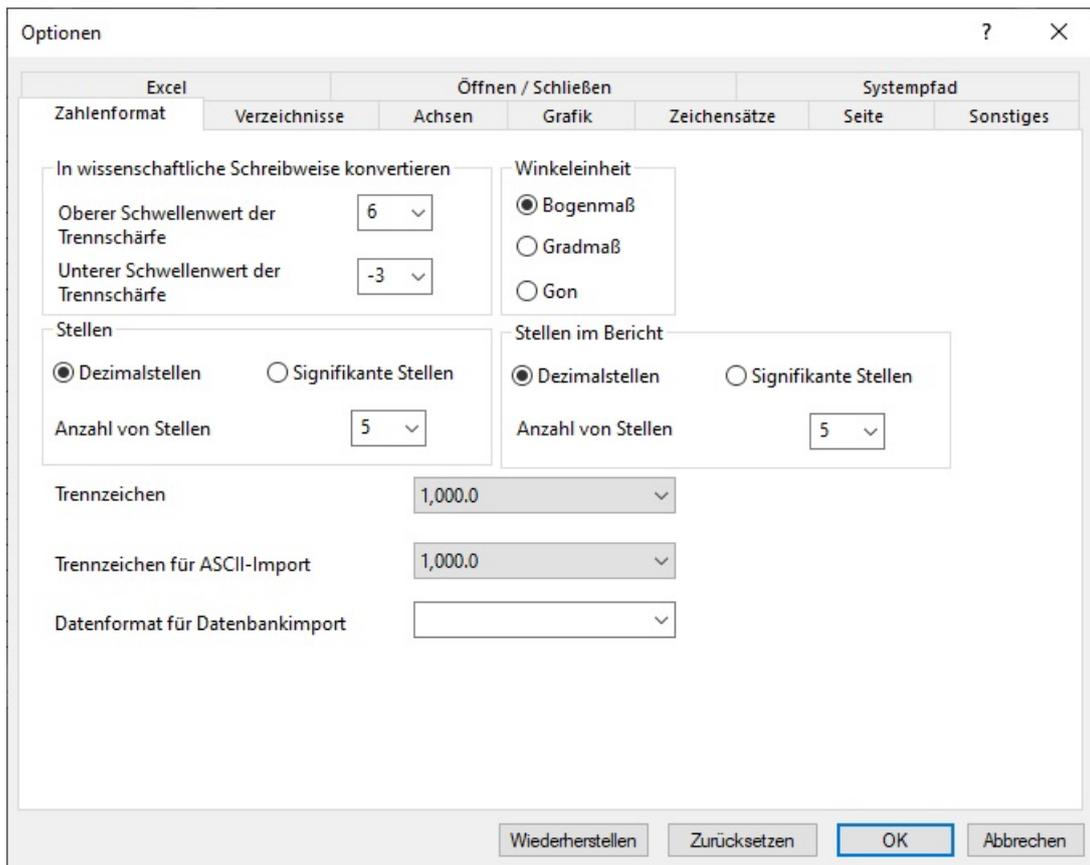


Abb. 2: Einstellung des Dezimalpunktes als Dezimaltrennzeichen.

3.3 Beispielaufgaben

Kap. 3.3.3 widmet sich der linearen Regression. Es kann zunächst übersprungen werden. Es wird bei der Behandlung der „Fehler- und Ausgleichsrechnung“ benötigt.

3.3.1 Grafische Darstellung von Messdaten

Ziel: Eingabe von X-Werten (X), einem dazugehörigen Satz von Y-Werten (Y1), Fehlern zu diesen Y1-Werten (FY1) und einem zweiten Satz von Y-Werten (Y2) zu denselben X-Werten. Anschließend grafische Darstellung dieser Werte inkl. Fehlerbalken für die Y1-Werte.

- (1) Nach dem Start von Origin erscheint die Oberfläche eines neuen *Projektes* (Abb. 1). Auf der Projektoberfläche ist ein Arbeitsblatt (*Worksheet*) der Arbeitsmappe *Book1* geöffnet, in das die Daten eingegeben werden können (analog zu einer Excel-Tabelle).
- (2) Statt manueller Eingabe in das Arbeitsblatt können Daten auch aus Dateien mit fremden Formaten (Excel, ASCII,...) importiert oder per *cut&paste* eingefügt werden.
- (3) Die Arbeitsblatt-Tabelle hat zunächst 2 Spalten⁶: A(X) und B(Y), wobei A und B die Bezeichnungen der Spalten sind und die Buchstaben in Klammern angeben, ob es sich um Abszissen-Werte (X) oder Ordinaten-Werte (Y) handelt. Weitere Spalten für die Fehlerangabe zu Y (FY1) und den zweiten Datensatz mit Y-Werten (Y2) erhält man durch: → Spalte → Spalten hinzufügen.
- (4) Die Dateneingabe erfolgt unter Verwendung des Dezimalpunktes (s. Kap. 3.2). In den gelb unterlegten Zellen der Zeile Langname wird die Beschriftung für die Daten der jeweiligen Spalte eingetragen. Abb. 3 (links) zeigt das Arbeitsblatt nach Eintrag der Daten.

⁶ englisch *column*, in Origin abgekürzt mit *col*.

	A(X)	B(Y)	C(Y)	D(Y)
Langname	X	Y1	FY1	Y2
Einheiten				
Kommentare				
F(x)=				
1	1	3	1.3	1
2	2	5	1.5	4
3	3	7	1.7	9
4	4	9	1.7	16
5	5	11	1.1	26
6	6	13.5	1.1	36
7	7	15	1.9	54
8				
9				
10				
11				

Abb. 3: Origin-Arbeitsblatt in der Arbeitsmappe *Book1* nach Eintrag der Daten (links) und nach Festlegung des Datentyps in den Spalten (rechts).

- (5) Zur Erstellung eines Diagramms aus den eingegebenen Daten ist festzulegen, welche Spalte welchen Datentyp enthält. Dazu jeweils die gesamte Spalte markieren (Mausklick auf den Kopf der Spalte), dann rechter Mausklick, danach → *Setzen Als*. Zur Auswahl stehen:

- Als X setzen
- Als Y setzen
- Y-Fehlerbalken
- X-Fehlerbalken.

Durch diese Festlegung ändert sich die Spaltenbeschriftung gem. Abb. 3 (rechts): $C(yEr\pm)$ ⁷ bedeutet z. B., dass in Spalte C Y-Fehlerwerte stehen.

- (6) Anschließend die Spalten mit den zu zeichnenden Daten markieren. Da hier alle Daten gezeichnet werden sollen, müssen alle Spalten markiert werden. Danach → *Zeichnen* → *Punkt* → *Punktplot*. Durch Wahl von *Punktplot* werden nur Datenpunkte gezeichnet, ohne Verbindungslinien, die in der Regel physikalisch unsinnig sind.
- (7) Das Diagramm wird in ein neues Fenster (*Graph1*) gezeichnet, das damit zum aktiven Fenster wird. Dadurch ergeben sich z.T. andere Einstellungen in der Hauptmenüleiste (Abb. 4), als wenn das Arbeitsblatt das aktive Fenster ist (Abb. 1).
- (8) Die Symbole für die Datenpunkte lassen sich nach Doppelklick auf die Symbole ändern. Im sich öffnenden Fenster muss zunächst unter → *Gruppe* → *Modus Bearbeiten* der Wert *Unabhängig* eingestellt werden. Danach können unter → *Symbole* diverse Eigenschaften (Größe, Form, Farbe) für die einzelnen Datensätze unabhängig voneinander eingestellt werden. Es wird empfohlen, möglichst *offene* statt gefüllter Symbole zu verwenden, da dadurch z. B. kleine Fehlerbalken besser erkannt werden können.
- (9) Alle Textfenster in einem Diagramm sind nach Anklicken frei verschiebbar.
- (10) Die Achsenbeschriftung, die Achsenkalierung, die Art der Achse (linear, logarithmisch, ...), Gitterlinien usw. lassen sich über ein Fenster einstellen, das sich nach einem Doppelklick auf die entsprechende Achse öffnet.
- (11) Funktionsgraphen können einem Diagramm wie folgt hinzugefügt werden: Klick auf das Diagrammfenster, das dadurch zum aktiven Fenster wird. Dann → *Grafik* → *Funktionsgraph* hinzufügen. Im sich öffnenden Fenster kann die Funktion eingegeben werden.

⁷ „Er“ von „error“ (Fehler).

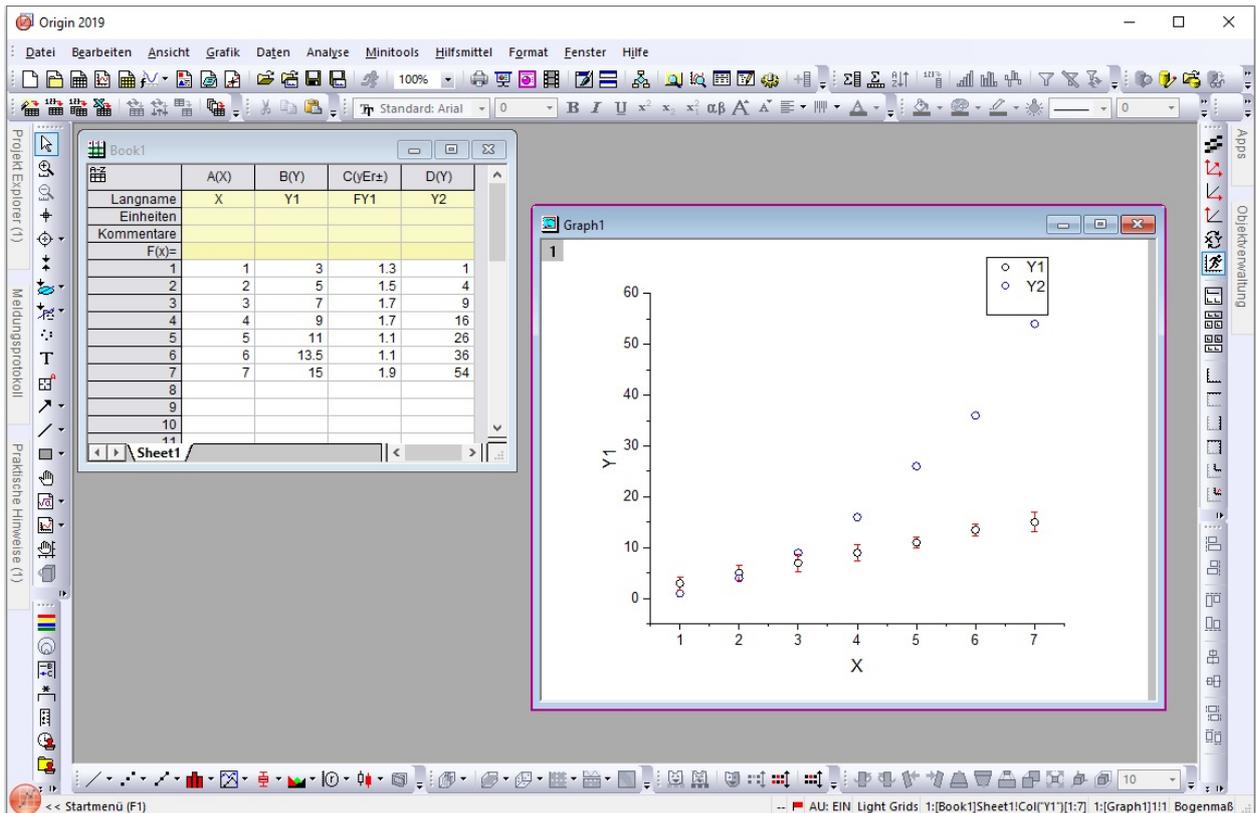


Abb. 4: Origin-Fenster nach Zeichnen eines Diagramms (*Graph1*). Die Fehler zu den Y1-Werten (schwarze Kreise) werden als Fehlerbalken (rote Linien) dargestellt.

3.3.2 Berechnungen mit Tabellendaten

Berechnungen mit den eingegebenen Daten können wie folgt durchgeführt werden: Arbeitsblatt durch Anklicken zum aktiven Fenster machen, dann eine leere Spalte (in den folgenden Beispielen E, F oder G) markieren, anschließend

→ Spalte → Spaltenwerte errechnen

In das nun erscheinende obere Textfenster (Abb. 5 links) wird die gewünschte Rechenoperation eingetragen.

1. Beispiel: In Spalte E soll das (elementweise) Produkt der Spalten A und B erscheinen. In das Textfenster trägt man demnach $\text{col}(A) * \text{col}(B)$ ein (Abb. 5 links), anschließend → OK.

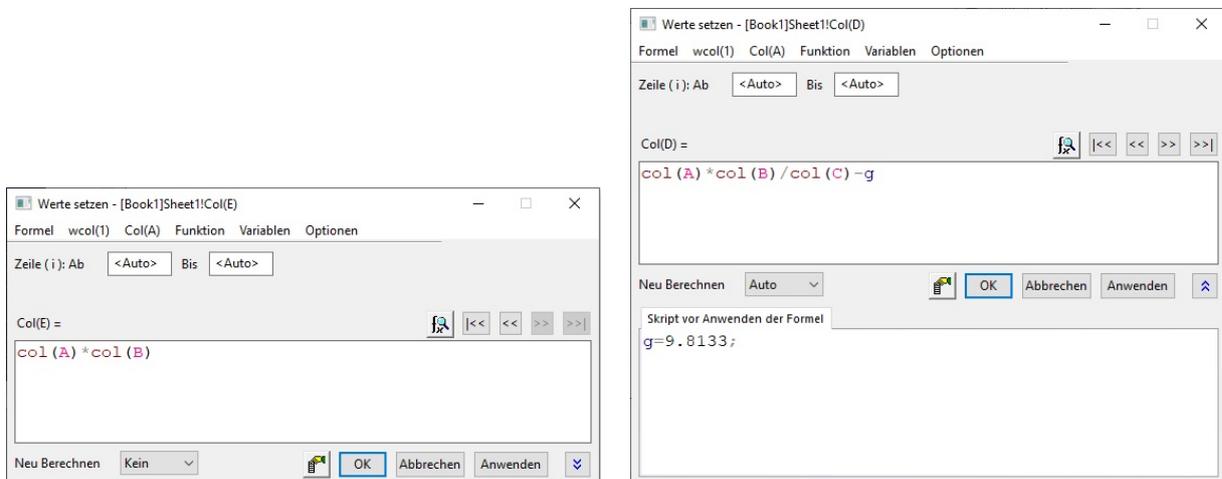


Abb. 5: Textfenster zum Eintrag von Rechenoperationen (links) und zusätzliches Skriptfenster (rechts, unten) zur Definition von Parametern.

Häufig müssen Berechnungen durchgeführt werden, bei denen neben den Zahlenwerten aus einzelnen Spalten auch Zahlenwerte physikalischer Größen wie z.B. die Erdbeschleunigung g , ein Widerstand R , eine Kapazität C usw. benötigt werden. Dann ist es praktisch, wenn diese Zahlenwerte einmal definiert (festgelegt) werden können und für alle späteren Berechnungen innerhalb desselben Projektes zur Verfügung stehen. Solche Definitionen können in einem *Skriptfenster* vorgenommen werden. Dazu wird dieses Fenster zunächst durch Klick auf den nach unten gerichteten Doppelpfeil (Abb. 5 links, unten rechts) sichtbar gemacht, sofern das Skriptfenster noch nicht geöffnet ist. Danach trägt man im unteren *Skriptfenster* (Abb. 5 rechts) die gewünschten Definitionen ein. Jede Definition wird mit einem Semikolon und der Eingabetaste abgeschlossen. Dazu folgendes Beispiel:

2. Beispiel: In Spalte F soll die Differenz aus dem Produkt der Spalten A und B, dividiert durch Spalte C, und der Erdbeschleunigung g erscheinen, die im Skriptfenster mit $g=9.8133$ definiert wird. Der Eintrag im Textfenster muss dann lauten: `col(A)*col(B)/col(C) - g, → OK.`
3. Beispiel: In Spalte G soll der Sinus von der Differenz der Werte in den Spalten B und A erscheinen. Der Eintrag im Textfenster muss dann lauten: `sin(col(B)-col(A)), → OK.`

3.3.3 Lineare Regression

Eine lineare Regression („linearer Fit“), d. h. die Berechnung einer Ausgleichsgeraden $y = a + bx$ durch Datenpunkte, die in einem Diagramm dargestellt wurden, wird wie folgt durchgeführt: Klick auf das Diagrammfenster, das dadurch zum aktiven Fenster wird. Anschließend Klick auf einen Datenpunkt, der zu dem Datensatz gehört, für den eine Ausgleichsgerade berechnet werden soll (hier ein Punkt aus dem Datensatz (X, Y_1)). Danach \rightarrow Analyse \rightarrow Anpassen \rightarrow Linearer Fit⁸. Es öffnet sich ein Fenster *Lineare Anpassung*, in dem eine Vielzahl von Parametern eingestellt werden kann. Nur die Wichtigsten werden hier erwähnt:

- (1) Der Wert im Fenster *Neu berechnen* sollte auf *Auto* gesetzt werden. Dadurch erfolgt eine automatische Neuberechnung aller Parameter der linearen Regression nach Änderung von Daten.
- (2) Im Feld *Fit Steuerung* \rightarrow Fehler als Gewichtung \rightarrow Keine Gewichtung (vorerst).
- (3) Für eine Berechnung der Steigung b und des Ordinatenabschnitts a der Ausgleichsgeraden im Feld *Fit Steuerung* keine Haken setzen.

Hinweis:

Für eine Berechnung *nur* der Steigung b der Ausgleichsgeraden (wenn der Ordinatenabschnitt a fest vorgegeben ist): Haken bei *Fester Schnittpunkt mit der Y-Achse* setzen, Ordinatenabschnitt a festlegen. Für eine Berechnung *nur* des Ordinatenabschnitts a der Ausgleichsgeraden (wenn also die Steigung der Geraden fest vorgegeben ist): Haken bei *Feste Steigung* setzen, Steigungswert b festlegen.

- (4) Im Feld *Angepasstes Kurvendiagramm* unter \rightarrow X-Datentyp \rightarrow Bereich die Einstellung *Ausweiten auf gesamten Achsenbereich* wählen. Die übrigen Einstellungen können beibehalten werden.
- (5) Nach Klick auf *OK* wird der lineare Fit durchgeführt. Im Diagrammfenster erscheinen die Ausgleichsgerade und eine Tabelle mit den Ergebnissen des linearen Fits (Abb. 6). Nach Doppelklick auf die Tabelle kann sie, wie üblich, den eigenen Bedürfnissen angepasst werden.

⁸ Wenn bereits einmal ein linearer Fit durchgeführt wurde, kann gewählt werden, ob alte Einstellungen übernommen werden sollen <zuletzt verwendet>, oder ob ein neues Dialogfenster geöffnet werden soll <standard>.

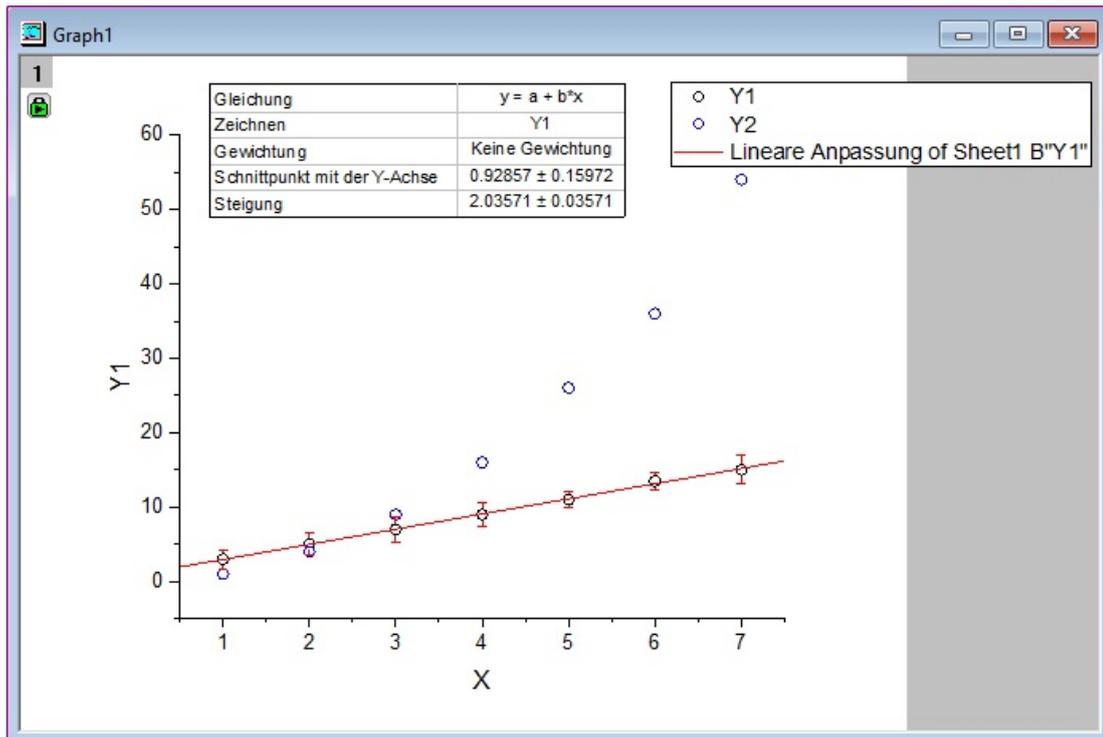


Abb. 6: Origin-Diagrammfenster nach Durchführung eines linearen Fits durch die X/Y1-Wertepaare.

- (6) Nach Durchführung des linearen Fits werden in der Arbeitsmappe *Book1* automatisch zwei weitere Arbeitsblätter ergänzt, in denen die Ergebnisse des linearen Fits (Blatt *FitLinear1*) und eine Wertetabelle zu der Ausgleichsgeraden (Blatt *FitLinearCurve1*) ausgegeben werden (Abb. 7).

	A(X)	B(Y)	C(yErz)	D(Y)
Langname	X	Y1	FY1	Y2
Einheiten				
Kommentare				
F(x)=				
1	1	3	1.3	1
2	2	5	1.5	4
3	3	7	1.7	9
4	4	9	1.7	16
5	5	11	1.1	26
6	6	13.5	1.1	36
7	7	15	1.9	54
8				
9				
10				
11				

Abb. 7: Origin-Arbeitsmappe nach Durchführung eines linearen Fits.

- (7) Neben den Größen a und/oder b werden im Ergebnissenfenster die Zahl N der für den Fit benutzten Datenpunkte (Wertepaare) und der Korrelationskoeffizient R ausgegeben. Die übrigen Parameter sind im Grundpraktikum vorerst unbedeutend.

Andere Verfahren zur Berechnung von Ausgleichskurven („Fit“) durch Datenpunkte, z. B. nichtlineare Funktionsfits, wie sie unter \rightarrow Analyse \rightarrow Anpassen zur Verfügung stehen, werden im Grundpraktikum erst später benötigt. Auf sie wird zu gegebener Zeit eingegangen.

3.3.4 Lineare Regression in logarithmischen Diagrammen

Durch Logarithmierung lassen sich bestimmte nichtlineare Zusammenhänge linearisieren. Beispielsweise wird aus dem Exponentialzusammenhang

$$y = a e^{bx}$$

durch Anwendung des natürlichen Logarithmus (Basis e)

$$\ln y = \ln a + bx$$

Trägt man also in einem *halblogarithmischen* Diagramm y über x auf (Ordinate logarithmisch), so ergibt sich eine lineare Darstellung: eine Gerade mit der Steigung b und dem Ordinatenabschnitt $\ln a$.

Ebenso wird aus dem Potenzgesetz

$$y = a x^b$$

durch Logarithmieren (Basis 10)

$$\log y = \log a + b \log x$$

Trägt man demnach in einem *doppeltlogarithmischen* Diagramm y über x auf (Ordinate und Abszisse logarithmisch), so ergibt sich ebenfalls eine lineare Darstellung: eine Gerade mit der Steigung b und dem Ordinatenabschnitt $\log a$.

Für eine lineare Regression durch derart linearisierte Daten muss unter \rightarrow Analyse \rightarrow Anpassen \rightarrow Linearer Fit unter Fit Optionen der Haken bei Scheinbarer Fit gesetzt werden. Der Fit erfolgt dann durch die linearisierten Daten, wie sie im Diagramm erscheinen.

4 Matlab

Das Programm Matlab wird im Praktikum eingesetzt, um Zahlenwerte mathematischer Funktionen zu berechnen und grafisch darzustellen, Berechnungen mit vektoriellen Größen durchzuführen (z. B. beim Versuch „Impuls- und Energieerhaltungssatz“) und Messdaten über geeignete Hardware in den Computer einzulesen und darzustellen. Dies stellt nur einen sehr bescheidenen Ausschnitt der vielfältigen Möglichkeiten dar, die Matlab bietet, der für die Arbeit im Grundpraktikum jedoch zunächst ausreichend ist.⁹

Die folgenden Bemerkungen sind wichtig, um die Lösung der Aufgabe aus Kap. 4.1 verstehen zu können. Matlab steht für *matrix laboratory*. Damit wird deutlich, dass Matlab ein Werkzeug zur Rechnung mit *Matrizen* (zur *Matrizenalgebra*) ist. Eine Matrix besteht aus $m \times n$ Zahlen, die in einem zweidimensionalen Schema mit m Zeilen und n Spalten angeordnet sind; man spricht auch von einer (m,n) Matrix (Abb. 8). Im einfachsten Fall ist $m = n = 1$, die Matrix enthält dann nur ein Element und stellt somit nur eine einzelne Zahl (einen *Skalar*) dar. Im nächst einfachen Fall enthält die Matrix nur eine Zeile oder nur eine Spalte, sie stellt dann einen *Zeilen-* oder *Spaltenvektor* dar.

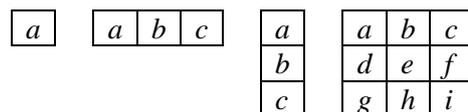


Abb. 8: V. l. n. r. schematische Darstellung von Skalar, Zeilenvektor mit 3 Komponenten, Spaltenvektor mit 3 Komponenten, Matrix mit 3×3 Elementen.

⁹ Eine Kurzanleitung für Matlab („Primer“) ist zu finden unter:
http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/getstart.pdf

In einer Programmiersprache stellen üblicherweise z. B. A und B zwei Variablen dar, die jeweils *eine Zahl* repräsentieren und deren Produkt $C = A \times B$ *eine neue Zahl* C ergibt. In Matlab dagegen stellen die Variablen A und B im Allgemeinen *Matrizen* dar, deren Produkt $A \times B$ eine *neue Matrix* C ergibt. Sollen demnach z. B. zwei Zeilenvektoren $A = [1 \ 2 \ 3]$ und $B = [4 \ 5 \ 6]$ *elementweise* miteinander multipliziert werden, um den Zeilenvektor $C = [1 \times 4 \ 2 \times 5 \ 3 \times 6]$ zu erzeugen, so darf man in Matlab *nicht* schreiben $C = A \times B$. Das ergäbe nach den Gesetzen der Matrizenalgebra auch keinen Sinn, denn es lassen sich nur Matrizen miteinander multiplizieren, bei denen die Spaltenzahl der ersten Matrix mit der Zeilenzahl der zweiten identisch ist. Für eine elementweise Multiplikation muss man vielmehr schreiben: $C = A . \times B$. Aus dem Operator „ \times “ wird also der Operator „ $. \times$ “, wenn elementweise Multiplikation gemeint ist, die im genannten Beispiel zu dem Ergebnis $C = [4 \ 10 \ 18]$ führt. Analog dazu gibt es die Operatoren für die elementweise Division („ $./$ “) und das elementweise Potenzieren („ $.^$ “).

Dazu ein einfaches Beispiel. Für die x -Werte 1, 2, 3, 4 sollen die Funktionen

$$y_1 = \sin(x) \quad \text{und} \quad y_2 = x \sin(x)$$

berechnet werden. Dazu schreibt man die x -Werte zunächst in einen Zeilenvektor x . In Matlab-Syntax lautet der entsprechende Befehl:

$$x = [1, 2, 3, 4]$$

Mit Hilfe von

$$y1 = \sin(x)$$

wird aus dem Zeilenvektor x der Zeilenvektor $y1$ berechnet, der vier Funktionswerte für die vier x -Werte enthält:

$$y1 = 0.8415 \quad 0.9093 \quad 0.1411 \quad -0.7568$$

Für die Berechnung der Funktion y_2 schreibt man:

$$y2 = x .* y1 \quad \text{oder direkt} \quad y2 = x .* \sin(x)$$

Der Punkt vor dem Multiplikationszeichen ($*$) sorgt dafür, dass eine *elementweise* Multiplikation der Zeilenvektoren x und $y1$ bzw. x und $\sin(x)$ stattfindet und das Ergebnis einen neuen Zeilenvektor $y2$ mit ebenfalls vier Elementen ergibt:

$$y2 = 0.8415 \quad 1.8186 \quad 0.4234 \quad -3.0272$$

Alles Weitere im Umgang mit Matlab ist „learning by doing“, wie aus der Lösung der folgenden Aufgabe hoffentlich deutlich wird.

4.1 Exemplarische Aufgabe

Zeichnen Sie ein Spannungssignal U als Funktion der Zeit t , das eine gedämpfte harmonische Schwingung darstellt:

$$(1) \quad U(t) = U_0 \sin(\omega t + \varphi) e^{-\alpha t}$$

mit der Amplitude $U_0 = 2 \text{ V}$, der Kreisfrequenz $\omega = 2\pi f$, der Frequenz $f = 0,5 \text{ Hz}$, der Anfangsphase $\varphi = \pi/5$ und der Dämpfungskonstanten $\alpha = 1/(4T)$ ($T = 2\pi/\omega$ ist die Periodendauer der Schwingung) im Zeitbereich zwischen $t = 0 \text{ s}$ und $t = 15 \text{ s}$.

4.2 Hinweise zur Lösung

- (1) Starten des Programms durch Doppelklick auf das Matlab-Icon (zurzeit Version 2017b).
- (2) Nach dem Start erscheint ein Fenster gem. Abb. 9. Sollte das Startfenster anders aussehen, kann es durch → Layout → Default in die Form aus Abb. 9 gebracht werden.
- (3) Das Fenster enthält in der Mitte das „Command Window“, in das Matlab-Befehle direkt eingegeben werden können. Rechts oben erscheint das Fenster „Workspace“, in dem alle benutzten Variablen aufgelistet werden. In dem darunter liegenden Fenster „Command History“ erscheinen alle Befehle, die in das „Command Windows“ eingegeben wurden. Das linke Fenster „Current Folder“ zeigt den Inhalt des aktuellen Arbeitsverzeichnisses an, das in der Pfad-Zeile ausgewählt wurde.
- (4) Statt Matlab-Befehle direkt in das „Command Window“ einzugeben, ist es praktischer, sie zunächst in eine Textdatei, das so genannte „m-File“ (oder „Matlab-Script-File“, Endung „.m“) zu schreiben. Durch → New Script öffnet sich das Fenster des Matlab Editors, in das die Befehle zur Lösung der Aufgabe eingetragen werden können. Abb. 10 zeigt das Editor-Fenster mit diesen Befehlen und zusätzlichen Kommentaren.
- (5) Nach Ende der Eingabe erfolgt ein Klick auf das Symbol  (Save and Run) in der "Toolbar-Zeile" des Editor-Fensters. Das m-File wird dadurch zunächst unter einem einzugebenden Namen, z. B.

O:\Mueller_Meier\Uebungen_Matlab\Gedaempfte_Schwingung.m

gespeichert. Anschließend werden die Befehle aus dem Script-File `Gedaempfte_Schwingung.m` nacheinander ausgeführt. In diesem Beispiel wird die gesuchte Funktion berechnet und in einem separaten Fenster „Figure 1“ grafisch dargestellt (Abb. 11).

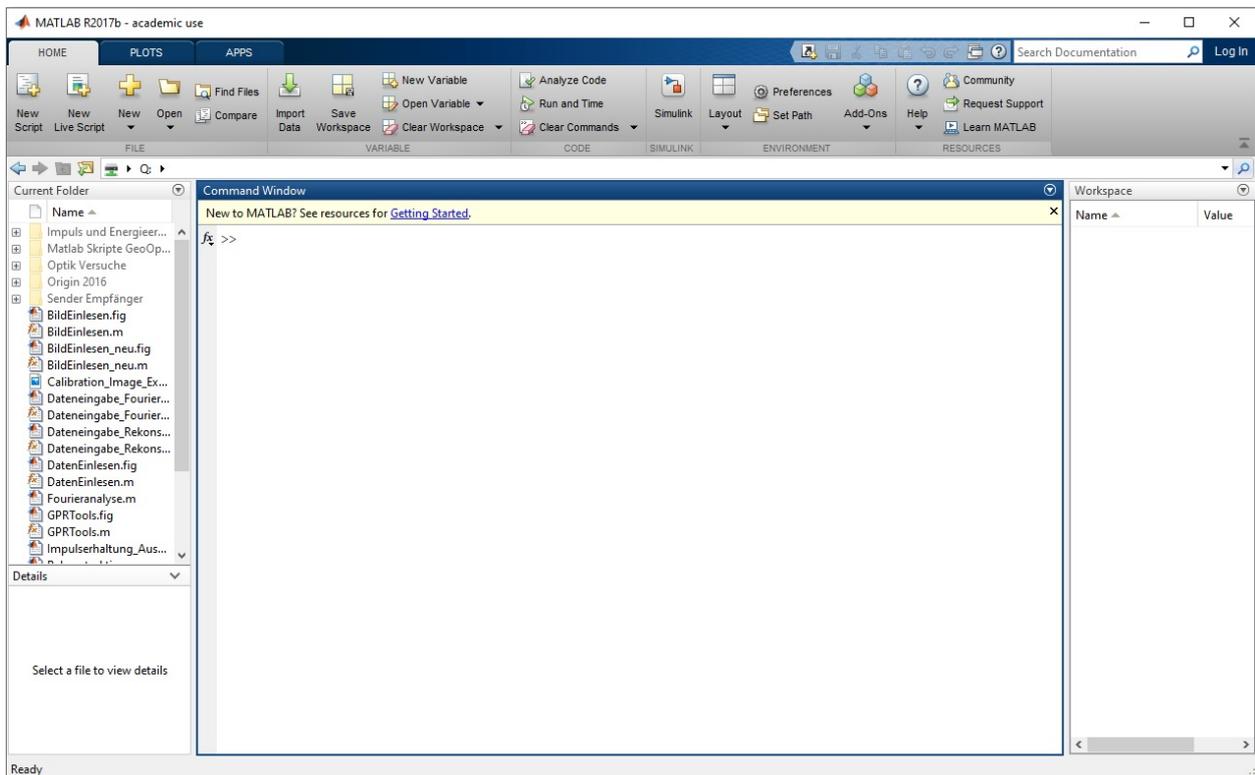
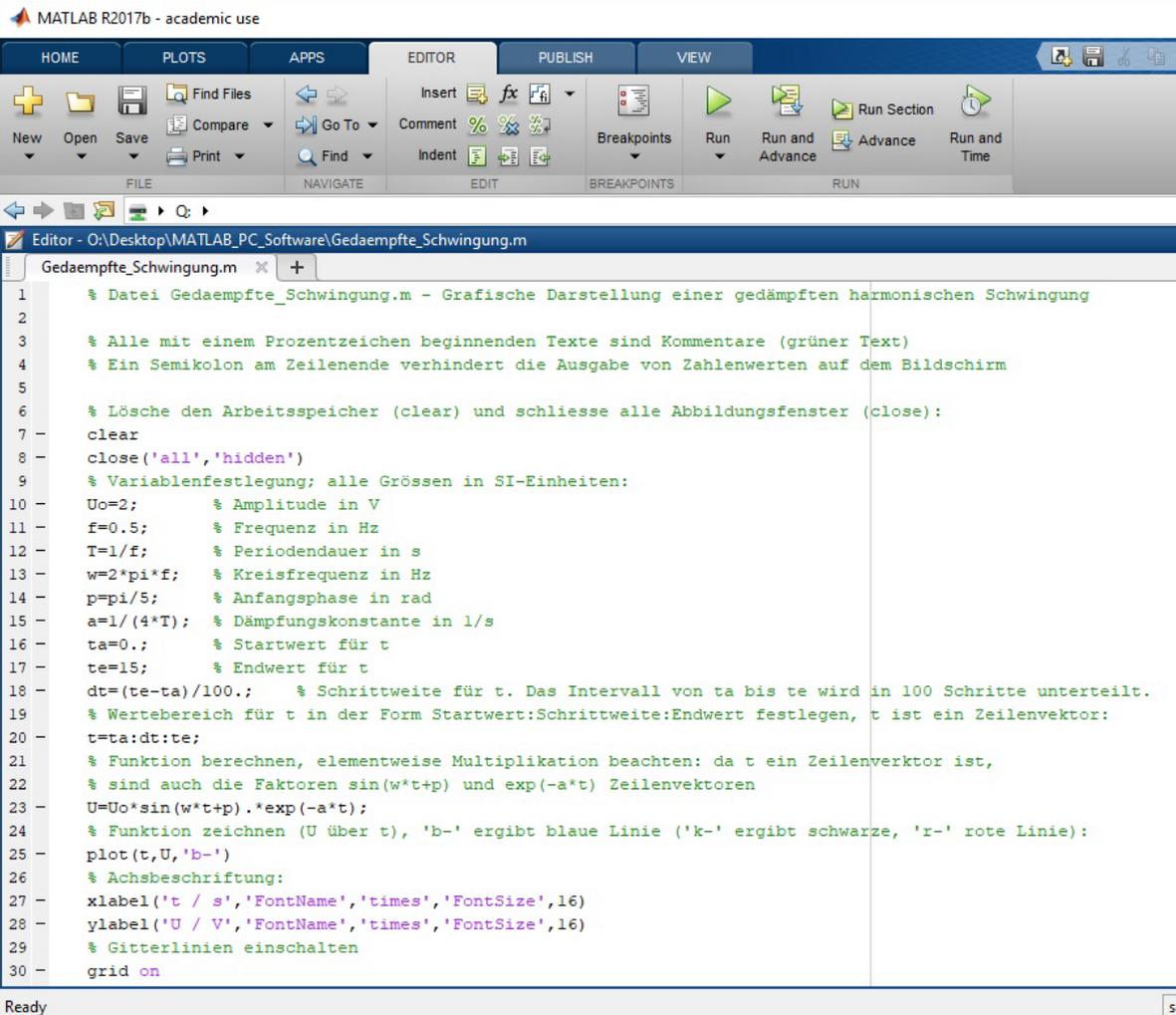


Abb. 9: Bildschirmoberfläche nach Starten des Programms Matlab (Version 2017b).



```

MATLAB R2017b - academic use
HOME PLOTS APPS EDITOR PUBLISH VIEW
New Open Save Find Files Compare Go To Insert Comment % Indent Breakpoints Run Run and Advance Run and Time
FILE NAVIGATE EDIT BREAKPOINTS RUN
Editor - O:\Desktop\MATLAB_PC_Software\Gedaempfte_Schwingung.m
Gedaempfte_Schwingung.m
1 % Datei Gedaempfte_Schwingung.m - Grafische Darstellung einer gedämpften harmonischen Schwingung
2
3 % Alle mit einem Prozentzeichen beginnenden Texte sind Kommentare (grüner Text)
4 % Ein Semikolon am Zeilenende verhindert die Ausgabe von Zahlenwerten auf dem Bildschirm
5
6 % Lösche den Arbeitsspeicher (clear) und schliesse alle Abbildungsfenster (close):
7 clear
8 close('all','hidden')
9 % Variablenfestlegung; alle Größen in SI-Einheiten:
10 Uo=2; % Amplitude in V
11 f=0.5; % Frequenz in Hz
12 T=1/f; % Periodendauer in s
13 w=2*pi*f; % Kreisfrequenz in Hz
14 p=pi/5; % Anfangsphase in rad
15 a=1/(4*T); % Dämpfungskonstante in 1/s
16 ta=0.; % Startwert für t
17 te=15; % Endwert für t
18 dt=(te-ta)/100.; % Schrittweite für t. Das Intervall von ta bis te wird in 100 Schritte unterteilt.
19 % Wertebereich für t in der Form Startwert:Schrittweite:Endwert festlegen, t ist ein Zeilenvektor:
20 t=ta:dt:te;
21 % Funktion berechnen, elementweise Multiplikation beachten: da t ein Zeilenvektor ist,
22 % sind auch die Faktoren sin(w*t+p) und exp(-a*t) Zeilenvektoren
23 U=Uo*sin(w*t+p).*exp(-a*t);
24 % Funktion zeichnen (U über t), 'b-' ergibt blaue Linie ('k-' ergibt schwarze, 'r-' rote Linie):
25 plot(t,U,'b-')
26 % Achsbeschriftung:
27 xlabel('t / s','FontName','times','FontSize',16)
28 ylabel('U / V','FontName','times','FontSize',16)
29 % Gitterlinien einschalten
30 grid on

```

Abb. 10: Matlab Editor Fenster mit Befehlen für das Berechnen und Zeichnen der Funktion aus Gl. (1).

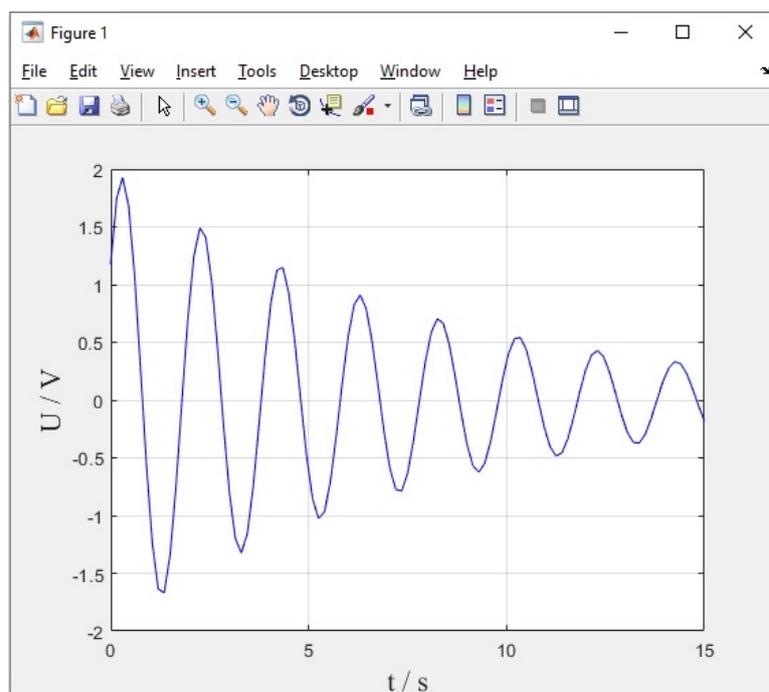


Abb. 11: Grafische Darstellung der Funktion aus Gl. (1) im „Figure“ Fenster. Auf die kursive Darstellung der physikalischen Größen U und t wird in diesem Beispiel verzichtet.