Usage of computers in the Introductory Laboratory Course Physics

1 Usage of Computers in the Introductory Laboratory Course

The students may and should use computers for all tasks related to the experiments that are to be performed during the introductory course. A short introduction will be given during the first hour of the course. Students without sufficient knowledge of computers are recommended to participate in corresponding courses (`Windows`, text processing (`Word` or `LaTeX`), spread sheet calculation (`Excel`), presentation (`Powerpoint`), one programming language).

1.1 Logging onto a Computer

The computers used in the introductory laboratory course are equipped with the operating system `Windows 10`). To login, you must enter a *local username* and *password*. Both will be communicated to the students at the beginning of the course on the spot. Working directory on the PCs in the basic laboratory course is an external cloud which is provided under the network drive "O:\". In the `Windows Explorer` (file manager) this working directory appears as

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

There, in addition to the `Matlab` folder, in which `Matlab` scripts[1] are provided, the group directories GPR01, GPR02,….etc. are also created. At first login, every team of two (e.g. Müller und Meier) creates a personal folder in this directory, choosing the respective surnames, e.g. O:\Mueller_Meier as name of the folder.

➢ *Personal data may only be saved in this personal directory!*

In the course of the semester, a structure of subfolders has to be created in this folder, where the data of specific experiments are stored later on, so:

- O:\Mueller_Meier
    - Origin_Excercises
    - Oscilloscope
    - Error_Theory
    - Resistances
    - Capacities
    - …

Avoid umlaut and blanks when naming folders and documents. Thereby avoiding problems when changing to another operating system.

Every member of the working group nn has a read/write access to the entire directory O:\ and can therefore also delete files. Thus, permanent security of data cannot be guaranteed. In order to store personal data permanently, data should be stored on a USB stick or in a personal directory in the university network. Such a directory is created automatically by the IT Services[2] division of the university for all students. For details please refer to the division IT Services and the person in charge of the computers in the CIP room[3] of the Institute of Physics.

---

1 Various `Matlab`-Skripts are required in various experiments during the lab course. Details can be found in the corresponding instructions of the experiments.

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

3 Several computers are available for student use in the CIP room of the Institute of Physics (W2 2-249). For details please refer to the information on https://uol.de/physik/cip/.

1.2 Printer in the Laboratory

A network printer (HP LaserJet Pro MFP M426fdw, black/white) is available in the laboratory. The printer can be used to print out documents which are prepared in the context of the laboratory course.

1.3 Protection Against Computer Viruses

The students in the laboratory course can use memory sticks to backup their own data. Care has to be taken that no computer viruses are transferred to PCs. In case of doubt the storage medium must be checked with the anti-virus software Sophos prior to usage.

2 Selection of the Available Software

In addition to the standard Windows programs, common text processing programs (Word), spread sheet calculation (Excel), presentation (Powerpoint), and internet browser (Firefox) are available on the computers used in the introductory laboratory course. Furthermore, the programs Origin and Matlab are also available. These programs are well suited for data acquisition, data analysis, and data visualization as well as for general functional calculations and representations of functional graphs, and are commonly used by the technical and scientific community. Both programs are also available on the computers in the CIP room of the Institute of Physics. Use of one of these programs on one's own personal PC is also possible. More information about can be found here:

http://www.uni-oldenburg.de/en/physics/teaching/laboratory-courses/origin/ (Origin)
https://uol.de/en/physics/laboratory-courses/matlab (Matlab).

The following brief instructions cannot make up for a manual, but can only give some introductory information sufficient for performing the different exemplary tasks. Furthermore information will be provided during the course. For the following description it is taken for granted that the users have already gained basic knowledge in handling Windows programs.

3 Origin

The software Origin (version 2019) is used in the laboratory course to feed data into tables, to carry out calculations with these data, to present the data graphically, to calculate parameters of regression lines through measured data (*linear regression*), and to perform nonlinear fits. The points described in chapter 3.3.3 are needed only later in the laboratory course. It is therefore recommended to take a look at this text once more at the proper time.[4]

3.1 Start of Origin, Default Settings

After starting Origin, a screen similar to that in Fig. 1 appears. The number and position of opened windows and toolbars depend on the personal settings. By

→ View, or
→ View → Toolbars

the settings of the toolbars can be adjusted to fit personal preferences.[5]

On the start screen, a window with an empty worksheet (*Sheet1*) of a workbook (*Book1*) appears, as in the case of the start of Excel. Measured data to be represented graphically are entered into this worksheet. This will possibly be complemented later by adding further worksheets and diagrams, notes, calculations, etc. Origin bundles all this information into a *Project*, which is saved to a single file with the extension .opj (**o**rigin **p**ro**j**ect) or .opju:

→ File → Save project.

It is strongly recommended to safe the file with the extension .opj in order to assure the compatibility with older versions of Origin.

[4] Further documents for Origin (Getting Started, Tutorials, Help,…) can be found in the download area under http://www.originlab.com/.

[5] The position of a menu bar (top, bottom, on the side) may be adjusted, as usual for Windows programs, by dragging the menu bar to the desired position while holding down the left mouse button.

The window shown at the bottom of Fig. 1 contains an overview of all data being part of a project. It can be made visible by performing the following steps:

→ View → Project Explorer

The position and size of the windows can be adjusted as usual.

By selecting the menu

→ Help → Change Language or
→ Hilfe → Sprache ändern

it is possible to switch between the German and English language versions of `Origin`.

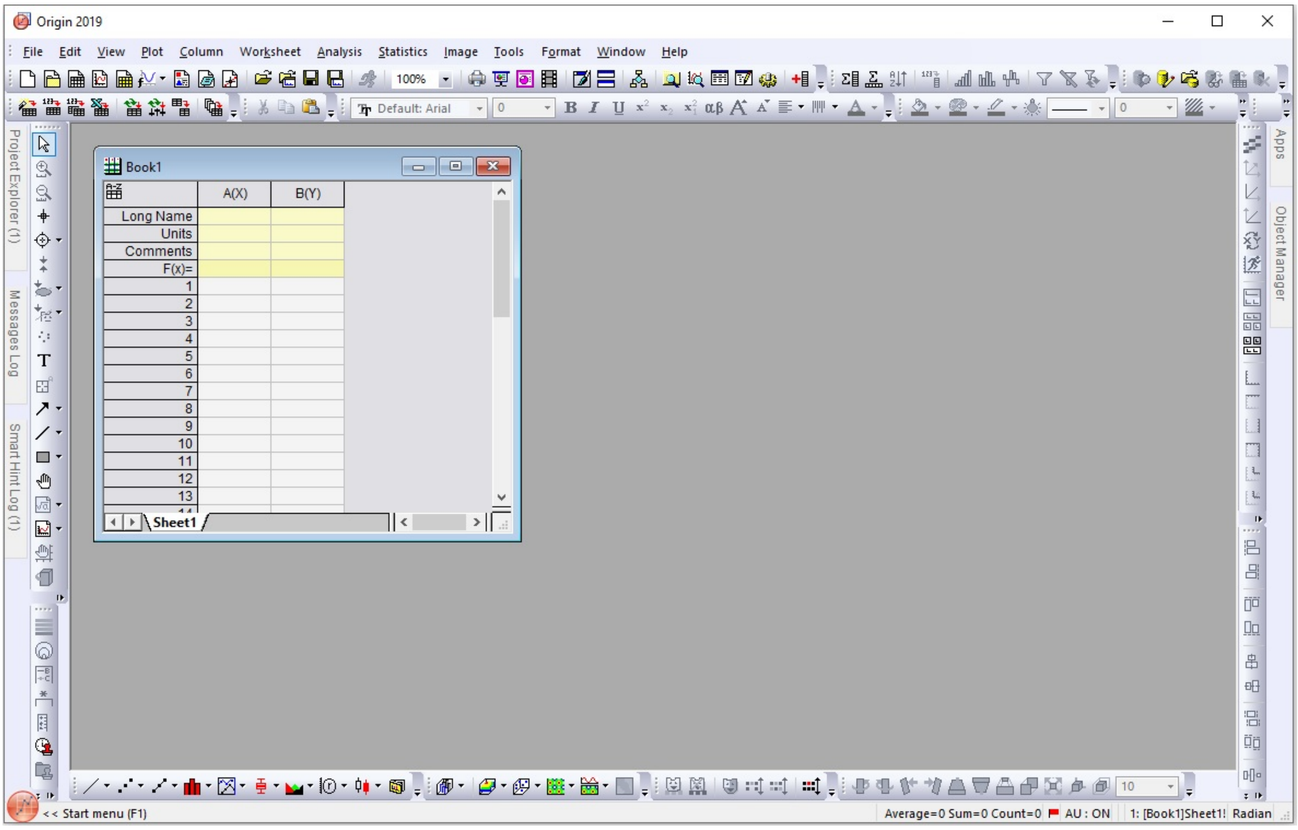


Fig. 1: Screen interface of the program `Origin (version 2019)`

3.2 Setting the Decimal Delimiter

Version 2019 of Origin allows switching between the English and German language versions. This can cause ambiguity in the interpretation of the decimal delimiter (decimal point vs. decimal comma), and possibly lead to inexplicable errors. In order to avoid such problems the *regional and language options* of `Windows` must not be imported; instead, the decimal *point* must be set explicitly as the decimal delimiter. This setting is done according to Fig. 2 via

Tools → Options → Numeric Format → Separators → 1,000.0
Separator for ASCII Import → 1,000.0

The point in the expression `1,000.0` is the decimal delimiter, while the comma serves solely as a visual aid to highlight blocks of thousands (English notation).

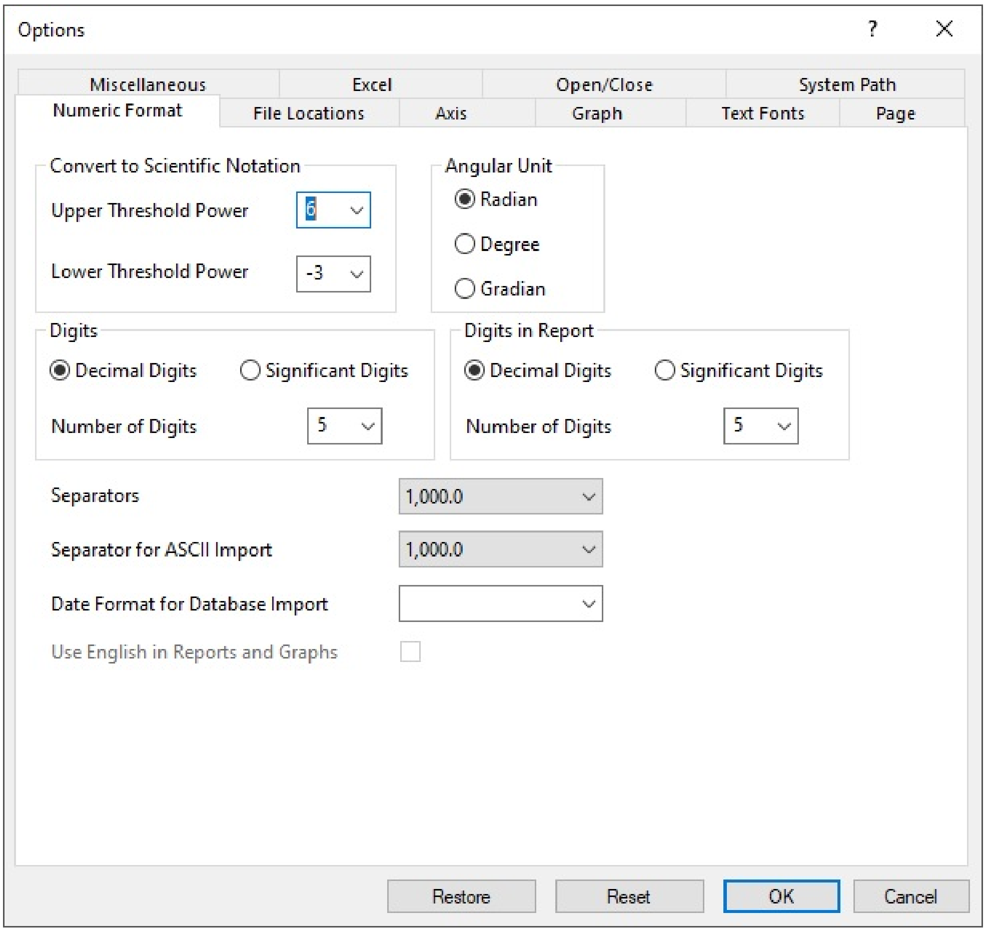


Fig. 2: Setting the decimal-point as the decimal delimiter

3.3 Sample Tasks

Linear regression is introduced in chapter 3.3.3. It may be skipped for the moment and will be used later in *"Error theory and regression analysis"*.

3.3.1 Graphical Representation of Measured Data

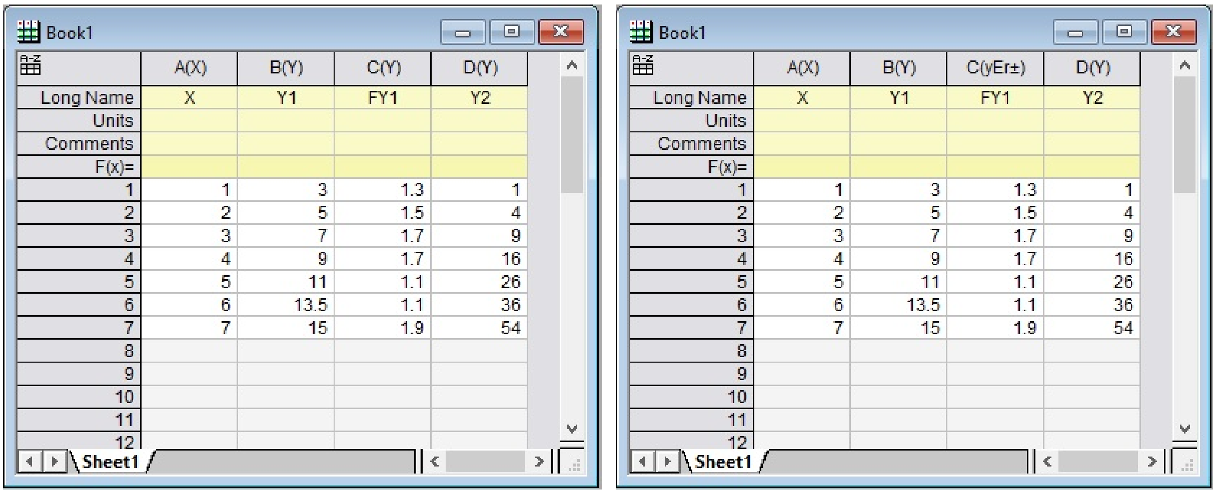
Goal: Feed in X values (X), a related set of Y values (Y1), errors of these Y1 values (FY1), and a second set of Y values (Y2) related to the same X values. These values are to be presented graphically, including error bars for the Y1 values.

(1) After starting `Origin` a screen of a new *Project* appears (Fig. 1). Within the project window, a *worksheet* of the workbook *Book1* is opened into which the data can be fed (just like an `Excel` table).

(2) Instead of a manual input, data may also be imported from files with other formats (`Excel`, ASCII,...), or inserted into the worksheet by *cut&paste*.

(3) In the beginning the worksheet table has two columns: A(X) and B(Y), A and B denoting the two columns and the letters in parentheses stating, whether they are values of the abscissa (X) or the ordinate (Y). Further columns for the errors of Y (FY1) and for the second set of Y values (Y2) are obtained by: → Column → Add New Columns.

(4) Data is entered using the decimal point (cf. Chap. 3.2). In the yellow underlayed cells of the row "Long Name" the label of the data in the corresponding columns is typed in Fig. 3 (left) shows the worksheet after entering the data.



Book1

	A(X)	B(Y)	C(Y)	D(Y)
Long Name	X	Y1	FY1	Y2
Units				
Comments				
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				
12				

Sheet1

Book1

	A(X)	B(Y)	C(yEr±)	D(Y)
Long Name	X	Y1	FY1	Y2
Units				
Comments				
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				
12				

Sheet1

Fig. 3: `Origin`-worksheet of the workbook *Book1* after entering the data (left) and after setting the data type (right).

(5) To create a diagram from the entered data, it has to be determined which column contains which type of data. For this purpose, mark the whole column (mouse click on head of column), then right click and → Set As. The following options are presented:

→ Set as X
→ Set as Y
→ Set as Y Error
→ Set as X Error.

Due to this setting the column label changes (Fig. 3, right): "C(yEr±)"[6] means e.g. that column C contains Y-Errors.

(6) Subsequently, mark the columns with the data to be drawn. Since all data shall be drawn here, all columns must be marked. Then → Plot → Symbol → Scatter. By selecting „Scatter", only data points without connecting lines are shown, which is generally nonsensical in physics.

(7) The diagram is drawn in a new window (*Graph1*) which becomes the active window. This changes partly the main menu (cf. Fig. 4) as compared to the case where the active window is the worksheet (Fig. 1).

(8) The symbols for data points may be changed by double clicking on the symbols. For this purpose, first → Group → Edit Mode "independent" has to be set in the appearing window. Afterwards, several properties (size, form, and colour) can be set under → Symbol (for the individual data sets independent of each other). It is recommended to use *open* symbols rather than filled symbols, since this increases the visibility (e.g. of small error bars).

(9) All text windows of a diagram may be moved around after a click.

(10) Axes labels, the scaling, the type of the axes (linear, logarithmic,…), and grid lines etc. may be changed in a windows which appears after a double click on the respective axis.

(11) Function graphs can be added to a diagram as follows: Click on the diagram window which thereby becomes the active window. Then → Graph → Add Function Graph. The function can be entered in the appearing window.

6 "Er" stands for "error".

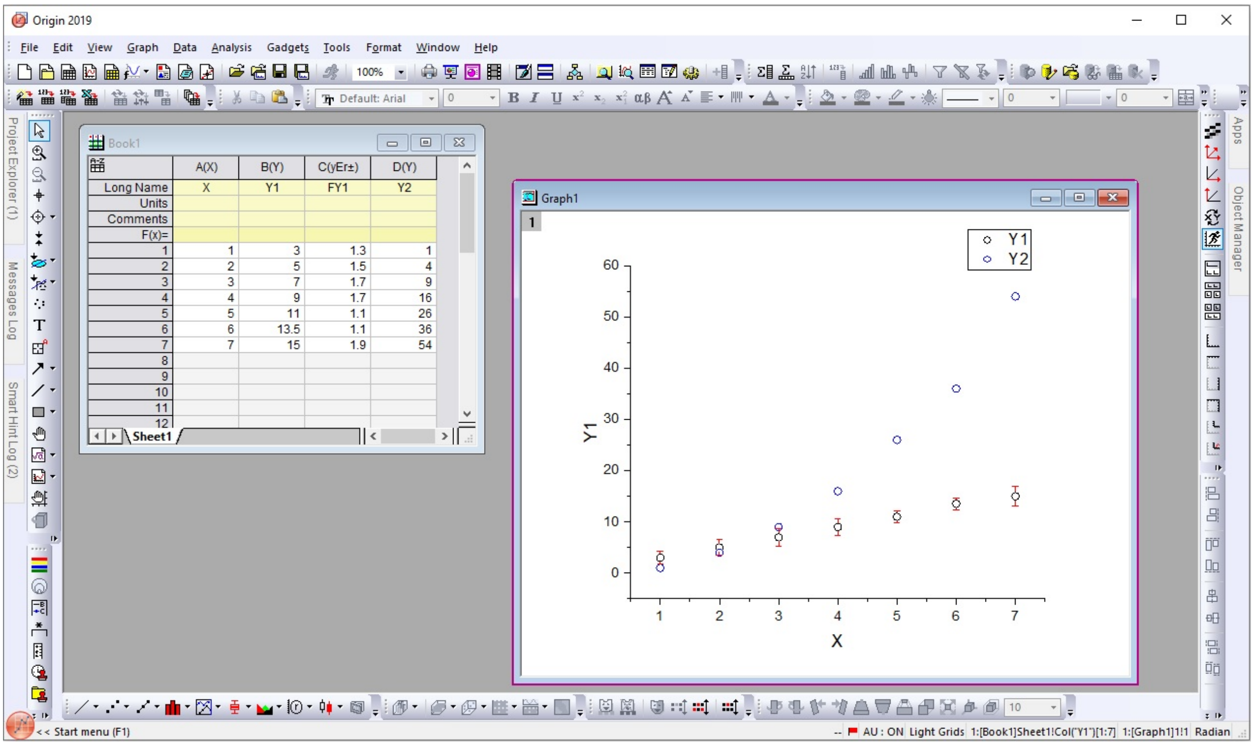


Fig. 4: `Origin` screen after drawing a diagram (*Graph1*). The errors of the Y1-values (black circles) are shown as error bars (red lines).

3.3.2 Performing Calculations on Tabular Data

Calculations with the entered data can be performed as follows: Click on the worksheet to make it the active window, then mark an empty column (denoted E, F or G in the following examples), and subsequently select

→ Column → Set column values.

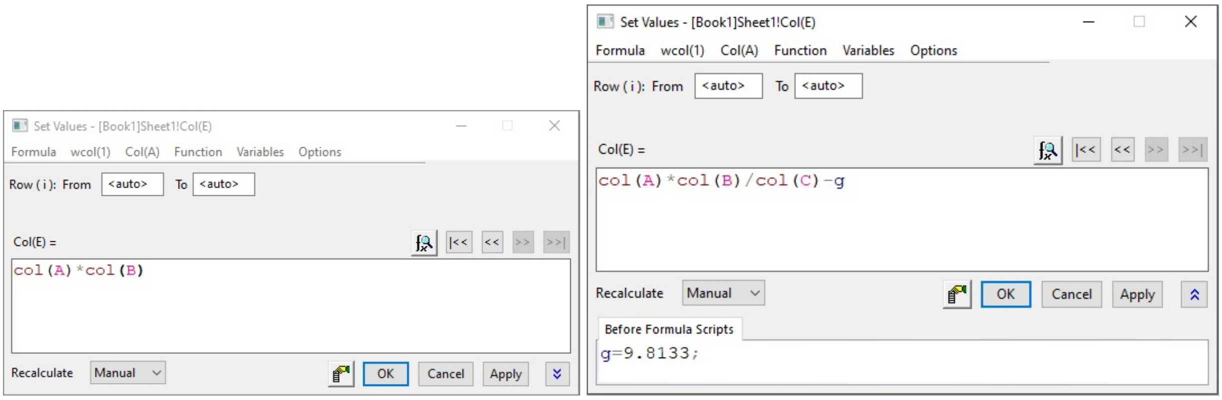


Fig. 5: Text window for entering arithmetic operations (left) and additional script window (right, bottom) for the definition of parameters.

A new window appears in which the required arithmetic operation is entered into the upper text window (Fig. 5 left).

1. Example: Column E should contain the (element-per-element) product of columns A and B. Then, `col(A)*col(B)` is thus typed into the text window (cf. Fig. 5 left) and entered by selecting → OK. It is often necessary to perform calculations for which numerical values of physical quantities, e.g. gravitational acceleration *g*, resistance *R*, capacity *C* etc., are needed in addition to numerical values from different columns. It is helpful in this case to define the numerical values of these quantities once and have them available for all subsequent calculations within the same project. Such definitions can

be performed in a *script window*. For this purpose this window is made visible by clicking on the downward double arrow (Fig. 5 left, bottom right), if the script has not already been opened. Then, the desired definition is inserted into the bottom script window (Fig. 5 right). Every definition is terminated with a semicolon and the enter key. This is shown in the following example:

2. Example: Column F should display the difference between the product of the columns A and B divided by column C, and the earth's gravitational acceleration *g* which has been defined previously in the script window as `g=9.8133`. The expression to be entered into the text window in this case is: `col(A)*col(B)/col(C)-g,` → OK.

3. Example: Column G is to contain the sine of the difference of values in columns B and A: The expression `sin(col(B)-col(A))`must be entered into the text window, followed by selecting → OK.

3.3.3 Linear Regression

A linear regression ("linear fit"), i.e. the calculation of a regression line $y = a + bx$ through data points, which are shown in a diagram, is performed as follows: Click on the diagram window which thereby becomes the active window. Thereafter, click on a data point belonging to the dataset for which the regression line is to be calculated (in this case, a point from the dataset (X,Y1)). Then → Analysis → Fitting → Fit Linear[7]. A window "Linear Fit" will open, in which a variety of parameters can be set. Only the most important ones are mentioned here:

(1) The value in the window "Recalculate" should be set to "Auto". Thereby an automatic recalculation of all parameters of the linear regression is carried out after a change of data.
(2) In the field "Fit Control" → Errors as Weight → No Weighing (for the time being).
(3) For the calculation of the slope *b* and the ordinate intercept *a* of the regression line: Do not check any options in the field "Fit Control".

Hint:
For a calculation of *only* the slope *b* of the regression line (if the ordinate intercept is fixed): Set check mark "Fix intercept", and set the ordinate intercept *a*. For a calculation of *only* the ordinate intercept *a* of the regression line (if the slope of the line is fixed): Set check mark at "Fix Slope", and set the slope value *b*.

(4) In the field "Fitted Curves Plot" under → X-Data Type → Range select "Span to Full Axis Range". The remaining settings can be left unchanged.
(5) After clicking on "OK", the linear fit is carried out. The regression line and a table with the results of the linear fit appear in the diagram window (Fig. 6). After double clicking on the table, it can be adapted to the personal needs as usual.
(6) After carrying out the linear fit, two new worksheets are added automatically to the workbook *Book1*, in which the result of the linear fit (sheet *Fit Linear1*) and a value table of the regression line (sheet *FitLinearCurves1*) are displayed (Fig. 7).
(7) Additionally to the values *a* and / or *b*, the number *N* of data points used for the fit and the correlation coefficient *R* are displayed in the results window. The other parameters are insignificant in the laboratory course for the time being.

Other procedures for calculating regression curves ("Fit") through data-points, e.g. non-linear function-fits, which are available under → Analysis → Fitting, are required later in the introductory laboratory course. They will be introduced in due time.

[7] If a linear fit has already been carried out, you must choose whether old settings shall be taken on or whether a new dialogue window shall be opened.

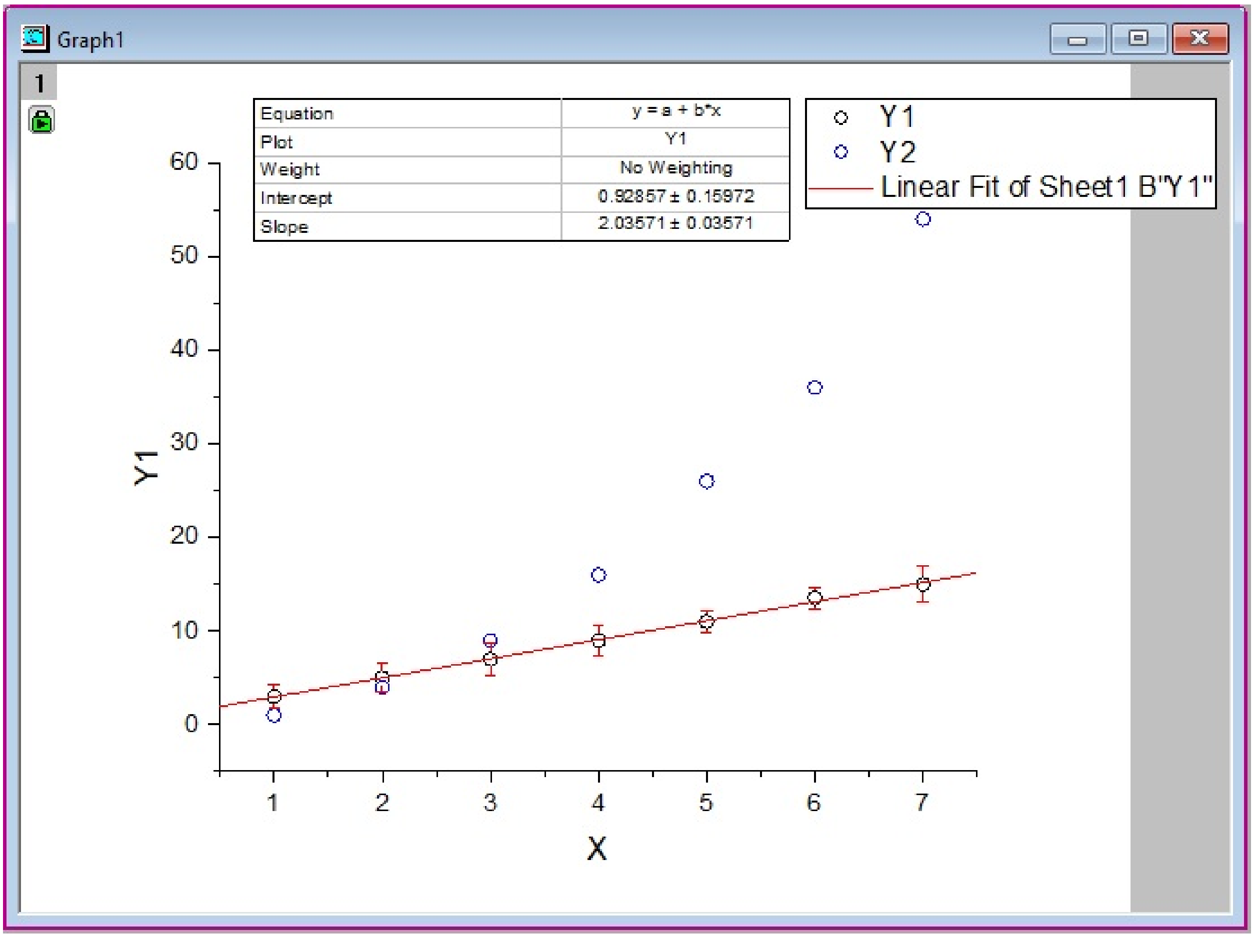
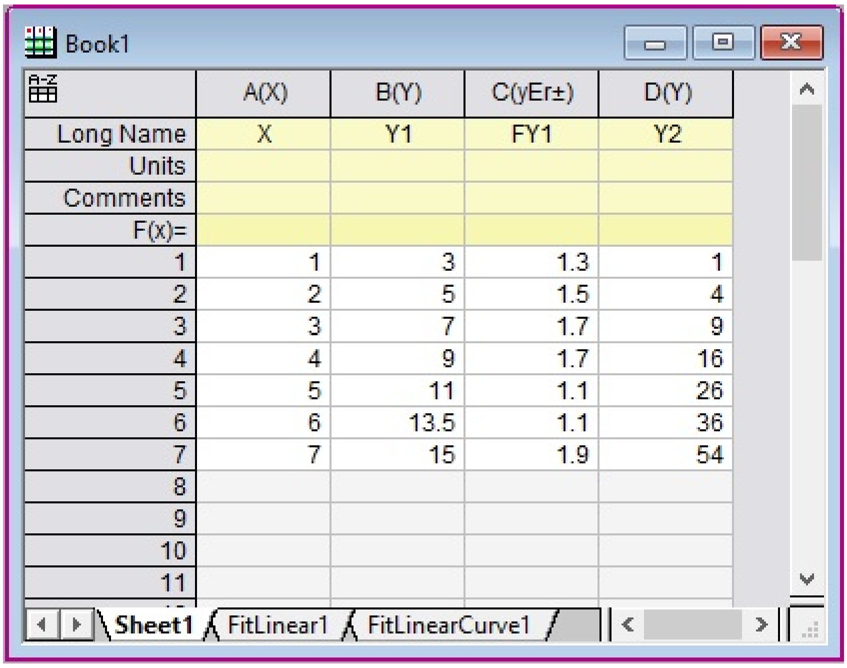


Fig. 6: `Origin` graph window after performing a linear fit on the X/Y1 value pairs.


Book1

	A(X)	B(Y)	C(yEr±)	D(Y)
Long Name	X	Y1	FY1	Y2
Units				
Comments				
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				

Sheet1 FitLinear1 FitLinearCurve1

Fig. 7: `Origin` workbook after carrying out a linear fit.

3.3.4 Linear Regression in Logarithmic Graphs

By taking the logarithm it is possible to linearize certain non-linear relationships. For example, by taking the natural logarithm (base e) of the exponential relation

$$y = a\,\mathrm{e}^{b\,x}$$

it is transformed into

$$\ln y = \ln a + b\,x$$

Thus, plotting y over x in a *semi-logarithmic* graph (the ordinate being logarithmic) a linear graph is obtained: a straight line with slope b and ordinate intercept ln a.

Likewise, the relation

$$y = a\,x^b$$

can be, by taking the logarithm (base 10), transformed into

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

Plotting y over x in a *full logarithmic* graph (ordinate and abscissa logarithmic), a linear representation is again obtained: a straight line with slope b and ordinate intercept log a.

For performing a linear regression on data linearized in this manner, the option “Apparent Fit” has to be selected under → Analysis → Fitting → Linear Fit. Then the fit is carried out through the linearized data as they appear in the diagram.

4 Matlab

The software `Matlab` is used in the laboratory course in order to calculate numerical values of mathematical functions and represent them graphically, to perform calculations with vector quantities (e.g. in the experiment on “*Conservation of momentum and energy*,” and to feed measured data into the computer by means of appropriate hardware and presenting it. This is only a very modest part of the manifold of applications offered by `Matlab`, however, it is sufficient for the tasks to be performed during the introductory laboratory course.[8]

The following remarks are important for solving the task from Chap. 4.1. `Matlab` stands for *matrix laboratory*, which makes it clear that `Matlab` is a tool for calculating with *matrices* (for *matrix algebra*). A matrix consists of $m \times n$ numbers arranged in a two-dimensional scheme with m rows and n columns; it is also called an (m,n) matrix (Fig. 8). In the simplest case $m = n = 1$: The matrix then contains only one element and thus represents a single number (a scalar). In the next simple case the matrix contains only one row or only one column. Then it represents a row or column vector.

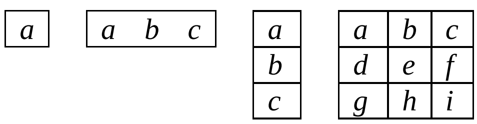


Fig. 8: *From left to right*: schematic representation of scalar, row vector with 3 components, column vector with 3 components, Matrix with 3 × 3 elements.

Usually, in a programming language A and B represent two variables, each representing *one number*, and their product $C = A \times B$ yields *one new number C*. In `Matlab`, however, the variables A and B in general represent *matrices*, the product of which $A \times B$ yields a *new matrix C*: $C = A \times B$. Thus, for example, if two row vectors $A = [1\ \ 2\ \ 3]$ and $B = [4\ \ 5\ \ 6]$ are to be multiplied with each other *element by element* in

[8] A brief introduction to Matlab (“Primer”) is found under: http://www.mathworks.com/access/helpdesk/help/pdf_doc/matlab/getstart.pdf

order to generate the new row vector C=[1×4, 2×5, 3×6], we must *not* write $C = A \times B$ in `Matlab`. This would make no sense according to the laws of matrix algebra, because only matrices can be multiplied where the number of columns of the first matrix is identical with the number of rows of the second matrix. In order to perform a multiplication on an element by element basis we have to write: $C = A\ .\!\times B$ instead. The operator ”×” thus becomes the operator ”.×” in the case of multiplying element by element, resulting in C = [4 10 18] in the example given above. By analogy, there are the operators for dividing (“./”) and raising to a power (“ .^“) element by element. The remainder when dealing with `Matlab` is “learning by doing”, hopefully the solution of the following task will show.

A simple example: Given the x-values 1, 2, 3, 4 the functions

$$y_1 = \sin(x) \qquad \text{and} \qquad y_2 = x\sin(x)$$

are to be computed. The x-values are first input into a row vector x. The corresponding command in `Matlab`-syntax is:

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

With the aid of

```
y1 = sin(x)
```

the row vector `y1` is computed from the row vector `x`, it contains four values of the function since there are four x-values:

```
y1 = 0.8415    0.9093    0.1411   -0.7568
```

In order to compute the function y_2 one writes:

`y2 = x .* y1`   or directly   `y2 = x .* sin(x)`

The dot before the multiplication symbol (*) performs an *element by element* multiplication of the row vectors `x` and `y1` (or `x` and `sin(x)` resp.) and the result is a new row vector `y2` which also contains four elements:

```
y2 = 0.8415    1.8186    0.4234   -3.0272
```

Everything else about the usage of `Matlab` is “learning by doing”, and will become clearer by solving the following exercises.

4.1 Sample Task

Draw a voltage signal U as a function of time t, that represents a damped harmonic oscillation:

(1) $$U(t) = U_0 \sin\left(\omega\, t + \varphi\right) \mathrm{e}^{-\alpha\, t}$$

with the amplitude U_0 = 2 V , the angular frequency $\omega = 2\pi f$, the frequency f = 0,5 Hz, the initial phase $\varphi = \pi/5$, and the damping constant $\alpha = 1/(4T)$ ($T=2\pi/\omega$ being the period of oscillation) in the temporal range between t = 0 s and t = 15 s.

4.2 Hints for the Solution

(1) Start the program by double clicking on the `Matlab` icon (at the moment version 2017b is used).

(2) After start-up, a window according to Fig. 9 is displayed. If the start window looks differently, it can be taken to the form of Fig. 9 Layout → Default.

(3) In the centre, the window contains the ”Command Window”, where `Matlab` commands can be entered directly. At the top right the window “Workspace” containing all used variables is displayed. The window below this displays a history of all commands entered in the “Command Window”,

hence it is called the "Command History". The window on the left labelled "Current Folder" shows the contents of the current working directory which is selected in the path line.

(4) Instead of entering `Matlab`-commands directly in the "Command Window", it is more practicable to type all commands into a text file first, the so-called "`m`-file" (or „`Matlab` Script File", extension "`.m`"). By → New Script the window of the `Matlab Editor` appears, into which the commands can be entered to solve the task. Fig. 10 shows the editor window with these commands and additional comments.

(5) After the input is finished, click on the Symbol ▷ (Save and Run) located in the „Toolbar" of the Editor window. Thereby the `m`-File will be first saved under the name (e.g.).

O:\Mueller_Meier\Matlab_Exercises\Damped_Oscillation.m

Subsequently, the commands in the script-file Damped_Oscillation.m will be executed consecutively. In this example, the respective function is calculated and graphically represented in a separate window "Figure 1" (Fig. 11).

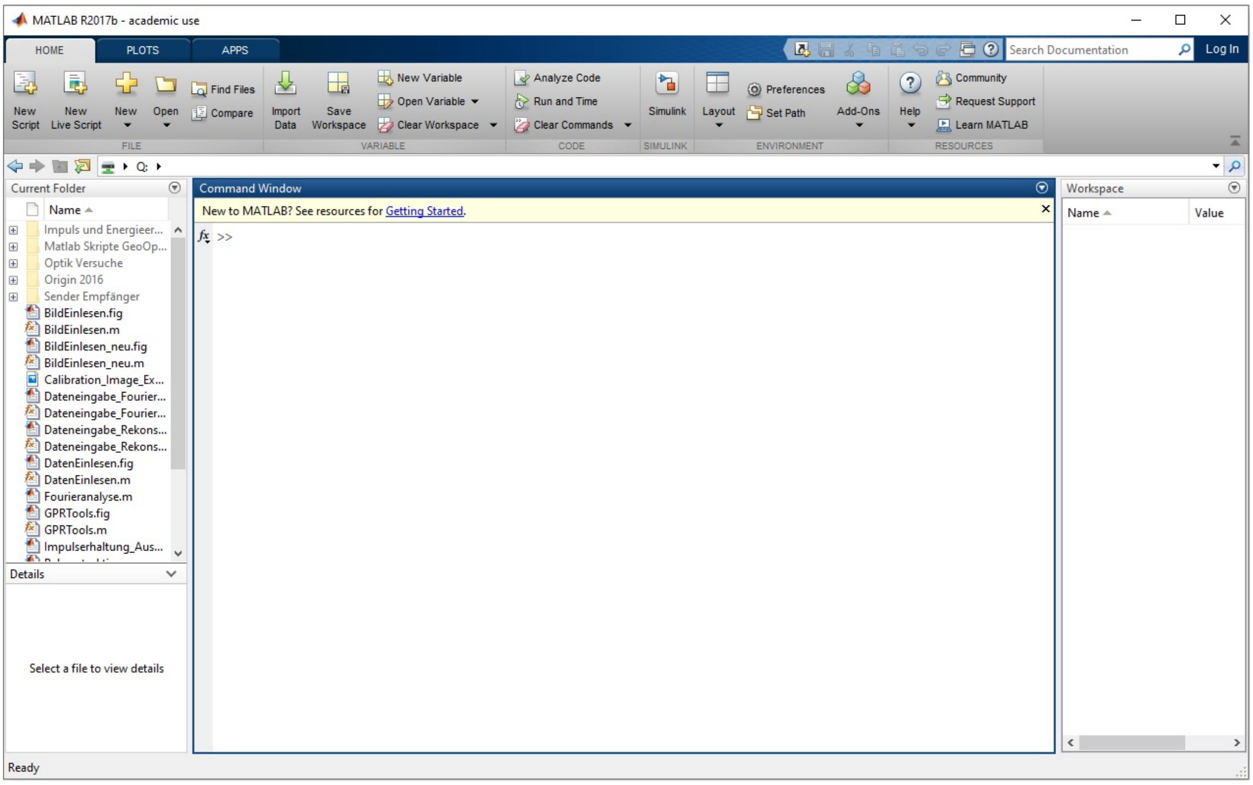


Fig. 9: Screenshot of `Matlab` after start-up (version 2017b).

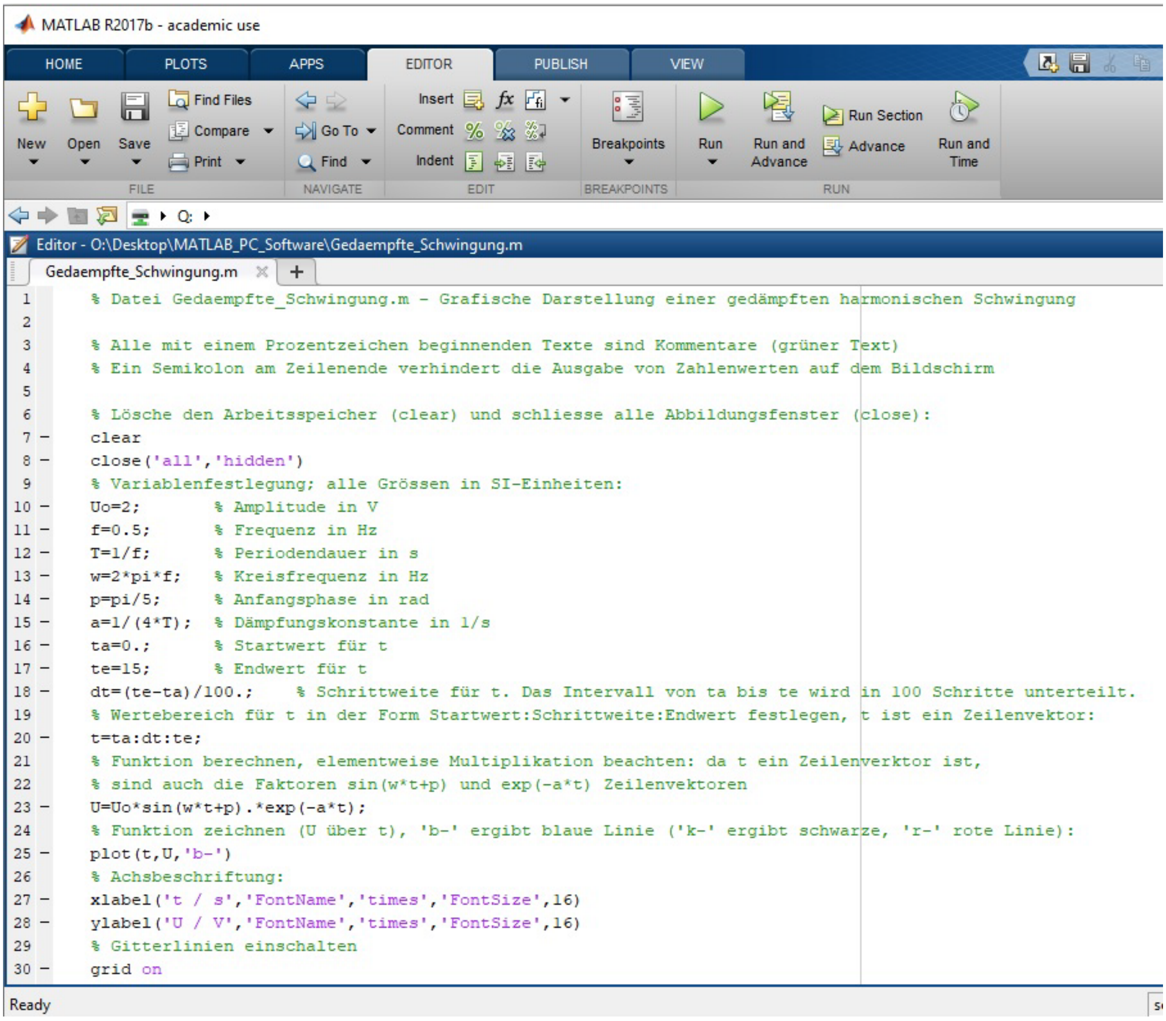


Fig. 10: `Matlab` Editor Window showing commands for calculating and drawing the function given in Eq. (1).

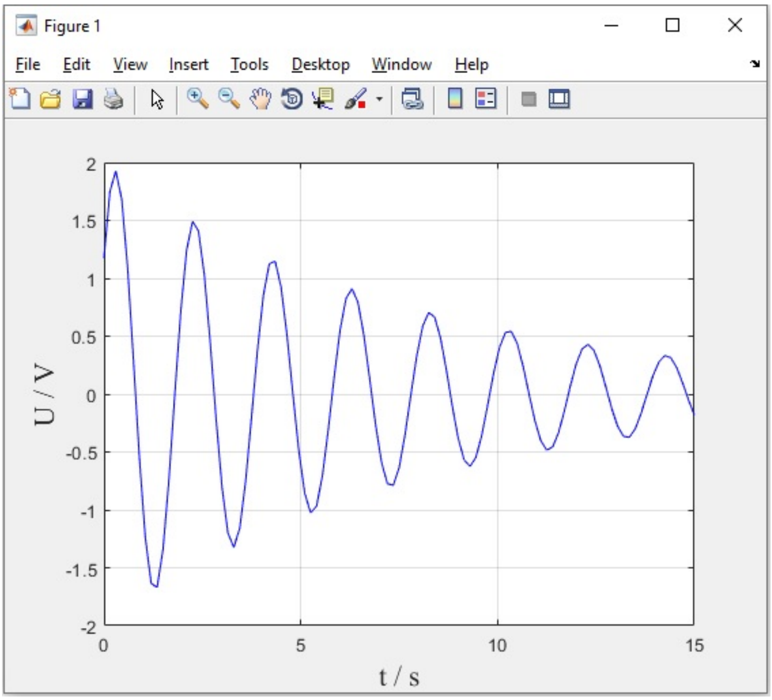


Fig. 11: Graphic representation of the function given by Eq. (1) in the "Figure" window. The representation of the physical quantities *U* and *t* in italic letters is renounced here.