

Tipps zur graphischen Auswertung von Praktikumsergebnissen

geschrieben von: **Christian Barthel**
c.barthel@uni-muenster.de
Daniel Guderian
daniel.guderian@wwu.de

Die graphische Darstellung der aufgenommenen Messdaten ist wichtiger Bestandteil einer gelungenen Auswertung vom Rahmen eines Versuchsprotokolls. Oft werden damit die relevanten Ergebnisse ermittelt oder Sachverhalte qualitativ bestätigt. Um es dem Leser möglichst einfach zu machen, die Aussagen den Diagrammen zu entnehmen, sollen hier einige Hinweise gegeben werden. In den seltensten Fällen ist eine reine Auflistung der Ergebnisse in Tabellenform ausreichend.

Zunächst wird eine kurze Übersicht über die Programme gegeben, die sich für die Darstellung eignen. Dann werden einige Aspekte der graphischen Auswertung besprochen und diese zuletzt an einem Beispiel erörtert.

Programme

Python 2.x

Skripte in Python lassen sich zum einen mit einem beliebigen Texteditor erstellen und über den Terminal ausführen und zum anderen kann man auch eine integrierte Entwicklungsumgebung nutzen. Für das wissenschaftliche Arbeiten mit Python werden einige weitere Bibliotheken benötigt, die separat installiert werden müssen und im folgenden kurz vorgestellt werden.

weitere Bibliotheken

NumPy: Numpy ist eine Bibliothek, die das Nutzen N-dimensionaler Arrays, die Integration von C/C++ und Fortran Code, nützliche Operationen der linearen Algebra, Fourier-Transformation und sehr einfaches Speichern und Laden von Textdateien und Daten in binärem Format ermöglicht. <http://www.numpy.org>

SciPy: SciPy bildet eine Sammlung numerischer Routinen, wie beispielsweise für die numerische Integration und numerische Optimierung. <http://www.scipy.org>

Matplotlib: Matplotlib ist eine Sammlung von Routinen zum Plotten von zwei- und dreidimensionalen Datenstrukturen, wobei die Syntax sehr an jene von Matlab angelehnt ist. <http://www.matplotlib.org>

Ipython: Ipython steht für Interactive Python und ermöglicht das interaktive Arbeiten mit Python. Das heißt Ergebnisse einzelner Zeilen können direkt im Terminal ausgegeben werden, ohne dass ein komplettes Python-Skript geschrieben werden muss. So können zum Beispiel einzelne Funk-

tionen zum Kennenlernen getestet werden, bevor sie in einem Skript Verwendung finden. <http://ipython.org>

Selbsthilfe

Eine Einführung und einen Überblick über Funktionen und Routinen in Python findet man auf www.python.org. Eine sehr gute Anlaufstelle bei der Fehlersuche stellt <http://www.stackoverflow.com> dar. Des Weiteren hilft auch eine Suchmaschine eigener Wahl weiter.

IDE (Integrated Development Environment)

Eine integrierte Entwicklungsumgebung vereinigt einen Texteditor, einen Compiler, einen Debugger, eine IPython-Konsole und weitere Instrumente zum Schreiben eines Codes in einem Programm. Empfohlen werden an dieser Stelle Spyder (<https://pythonhosted.org/spyder>) und für einen größeren Funktionsumfang PyCharm (<https://www.jetbrains.com/pycharm>).

Matlab

Zur Einführung in Matlab bieten folgende Skripte einen guten Einstieg:

- <https://www.mccormick.northwestern.edu/documents/students/undergraduate/introduction-to-matlab.pdf>
- <https://www.maths.dundee.ac.uk/software/MatlabNotes.pdf>
- https://wwwmath.uni-muenster.de/num/Vorlesungen/MATLAB-Kurs_WS08/Script/matlab-einfuehrung.pdf

Des Weiteren besitzt Matlab eine sehr gute und ausführliche, programminterne Hilfefunktion, die immer zu Rate gezogen werden kann.

LaTeX

LaTeX ist eine Sammlung von Software, die auf dem Textsatzsystem TeX basiert. Im Gegensatz zu MS Word und anderen Textverarbeitungsprogrammen folgt LaTeX nicht dem sogenannten WYSIWYG-Prinzip (What-you-see-is-what-you-get), sondern man „programmiert“ das Erscheinungsbild des Dokuments und erhält erst nach dem Kompilieren das eigentliche Dokument. Die Vorteile von LaTeX sind eine sehr einfache Erstellung des Literaturverzeichnis, Inhaltsverzeichnis, Abbildungsverzeichnis und der Verweis auf deren

Einträge sowie das Schreiben von Formeln und die Nutzung von Sonderzeichen jeder Art. Des Weiteren bietet es eine große Flexibilität beim Einbinden von Grafiken. Dem gegenüber steht zu Beginn eine sehr steile Lernkurve, die sich einfacher meistern lässt, wenn man auf eine fertige LaTeX-Vorlage zurück greift und sich von einem erfahrenen Nutzer eine Einführung geben lässt. Einführung in LaTeX:

- http://www.selflinux.org/selflinux/pdf/latex_einfuehrung.pdf
- <https://tobi.oetiker.ch/lshort/lshort.pdf>
- http://www.fernuni-hagen.de/imperia/md/content/zmi_2010/a026_latex_einf.pdf

Installation:

OS X: <https://tug.org/mactex/>, MacTeX umfasst eine LaTeX-Distribution und einen passenden Editor.

Windows: Hier erfolgt die Installation einer LaTeX-Distribution (<http://miktex.org>) separat und ein LaTeX-Editor (Übersicht: http://en.wikipedia.org/wiki/Comparison_of_TeX_editors) kann frei gewählt werden.

Origin

Plotten mit Origin

1. Ausgangspunkt zur Erstellung eines Plots sind die Werte in Abbildung 1. In den Spalten C und D befinden sich die Fehler zu den X und Y-Werten.
2. Durch Markieren einer gesamten Spalte und über Kontextmenü > Setzen Als > X Fehlerbalken / Y Fehlerbalken kann man die Werte einer Spalte als Fehler deklarieren (c. f. Abbildung 2)
3. Menüleiste > Zeichnen > Symbol > XY Fehlerbalken erzeugt einen Plot mit Fehlerbalken (c. f. Abbildung 3 und 4)
4. Ein Doppelklick auf die Fehlerbalken öffnet ein weiteres Fenster. Die linke Seite gibt einen Überblick über die genutzten Daten und in welcher Spalte diese sich befinden. Spalte C wird markiert und auf der rechten Seite wird die Orientierung der Fehlerbalken in x-Richtung festgelegt, für Spalte D wird dieses Häkchen entfernt (c. f. Abbildung 5).

5. Ein Doppelklick auf eine der beiden Achsen ruft ein Fenster hervor, in dem sich u. a. Änderungen am Gitternetz und der Linienbreite vornehmen lassen. Abbildung 6 zeigt den fertigen Plot.
6. Menüleiste > Datei > Grafik exportieren erzeugt eine Bilddatei.

Linearer Fit mit Origin

1. Ausgangslage ist ein Plot mit Fehlerbalken wie in Abbildung 6.
2. Menüleiste > Analyse > Anpassen > Linearer Fit öffnet ein Fenster mit Einstellungsoptionen für den linearen Fit (c. f. Abbildung 7).
3. Nach dem Klicken von „Ok“ wird der lineare Fit durchgeführt und die Fitparameter werden angegeben (c. f. Abbildung 8).

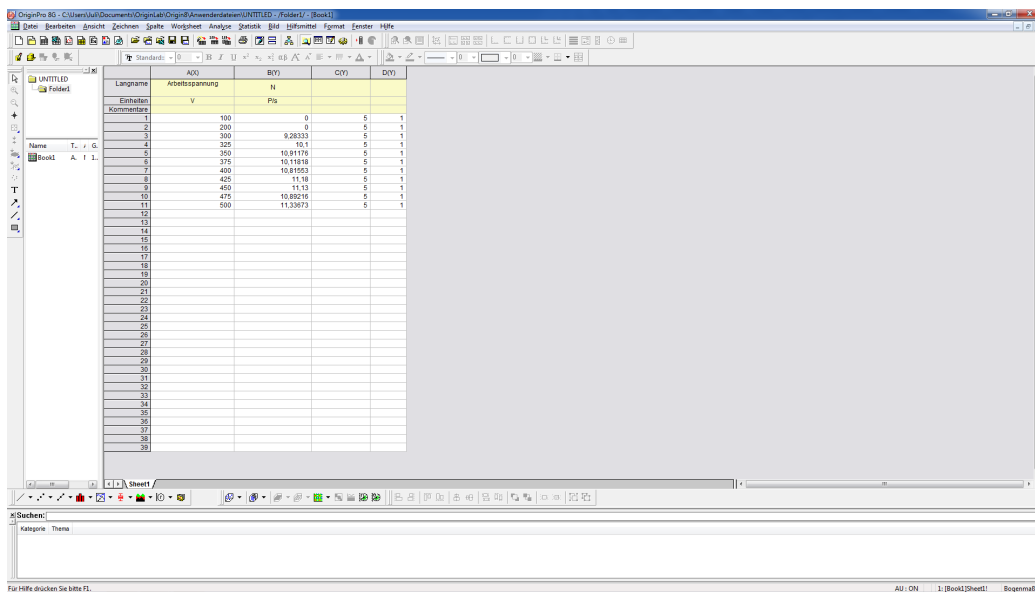


Abbildung 1

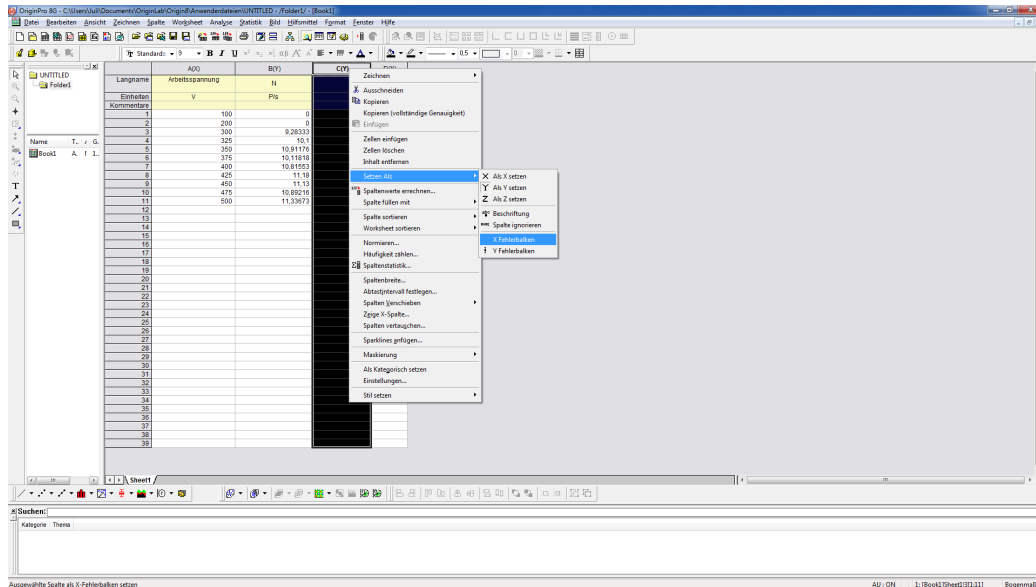


Abbildung 2

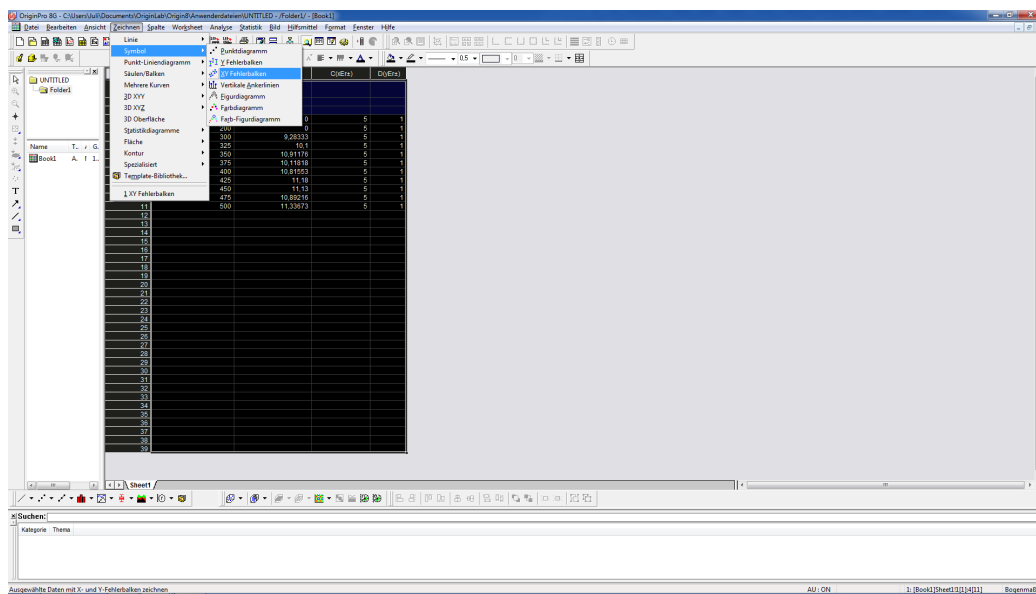


Abbildung 3

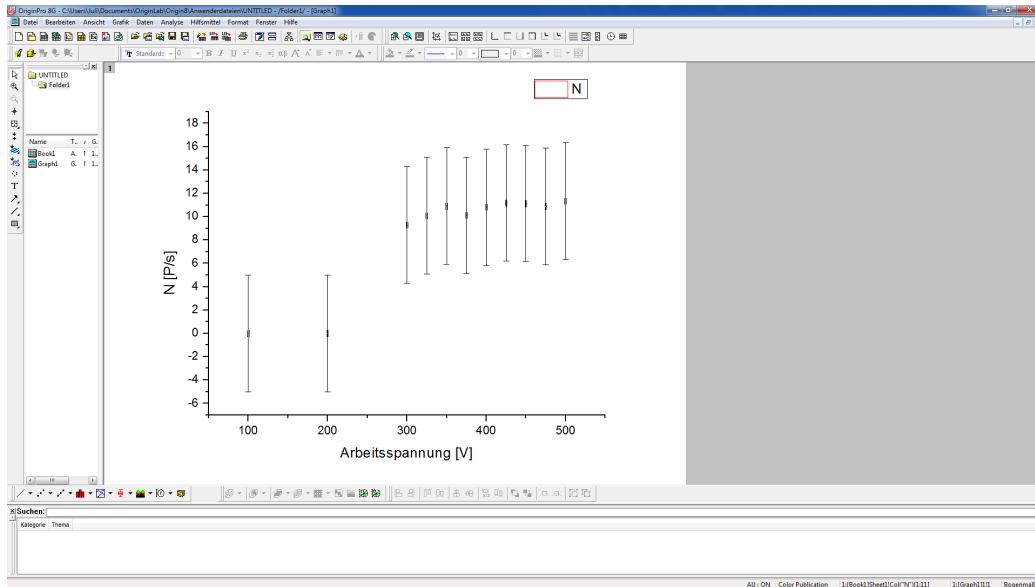


Abbildung 4

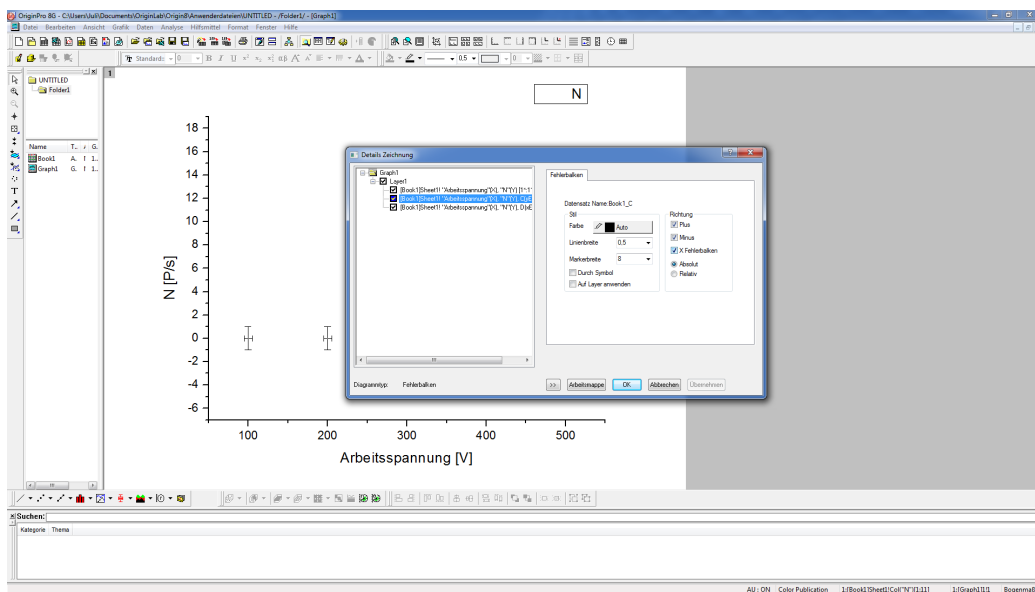


Abbildung 5

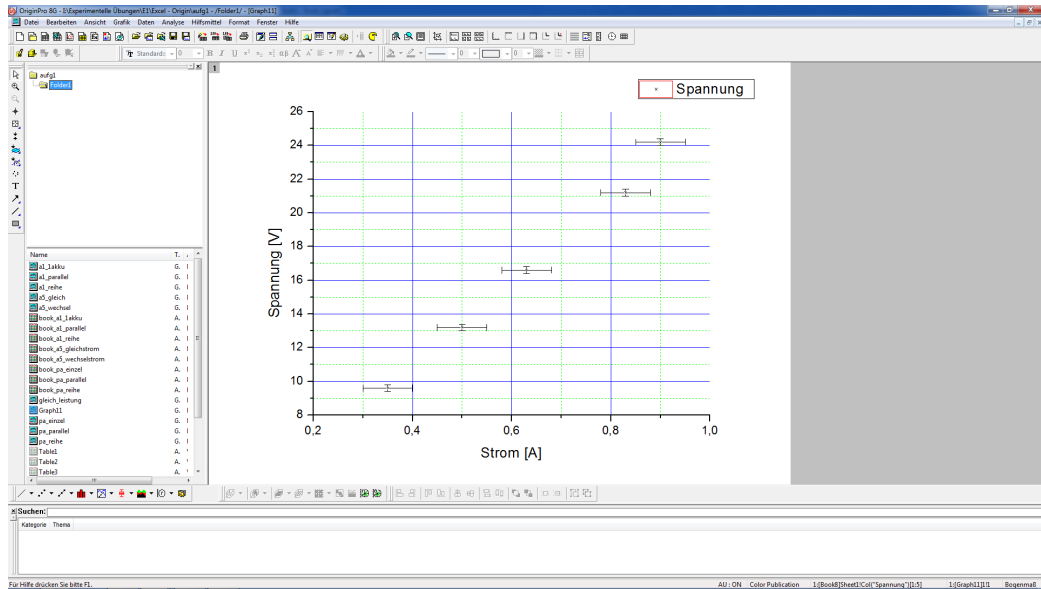


Abbildung 6

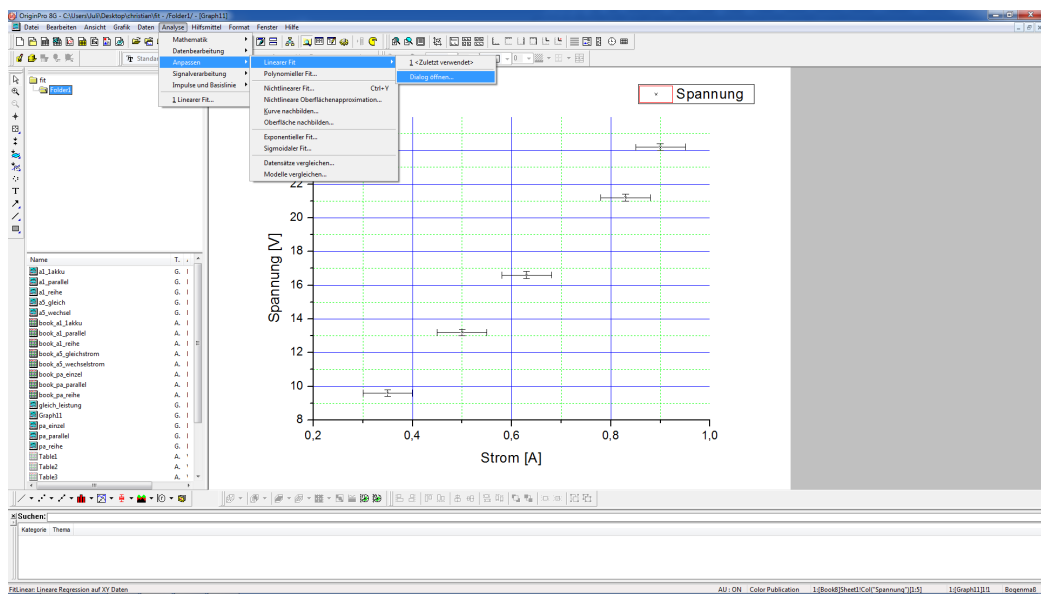


Abbildung 7

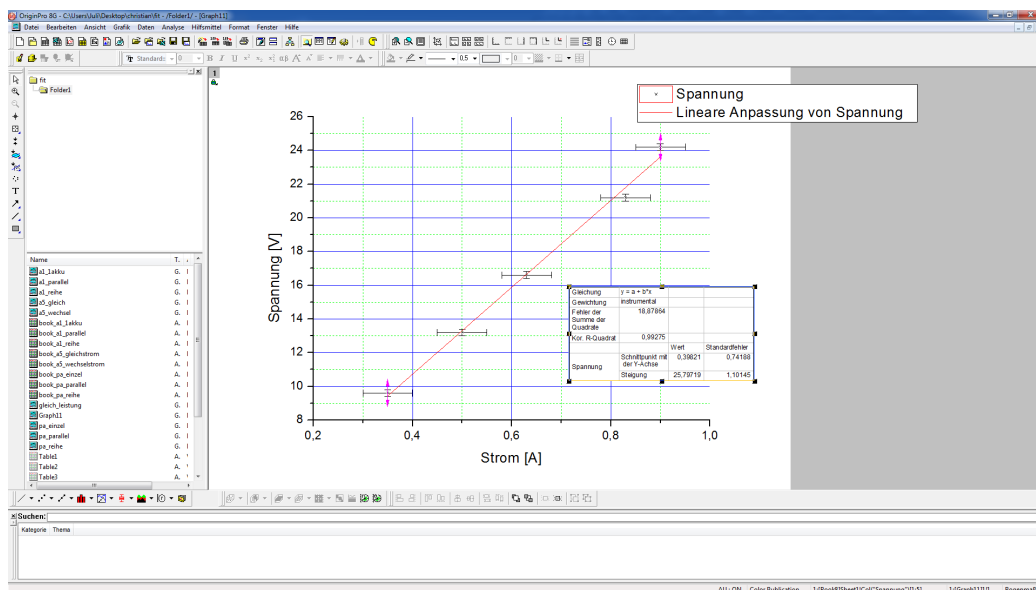


Abbildung 8

Allgemeine Hinweise zur graphischen Auswertung

Ein guter Ausgangspunkt für die Erstellung eines Diagramms ist eine sinnvolle Achsenbeschriftung und -einteilung. Die Namen der Achsen sollten immer beinhalten, was gemessen wurde und in welcher Einheit. In Sonderfällen können auch weiterführende Informationen hinzugefügt werden. Dabei ist darauf zu achten, dass das gleiche Formelzeichen wie im Text benutzt wird.

Bei der Einteilung der Achse sollten ca. 10% unter dem ersten und über dem letzten Messwert liegen. Häufig sind solche Werte Standard in den Programmen. Die Anzahl der Striche sollten sinnvoller Weise nicht zu hoch gewählt werden, aber gerade so, dass die entscheidenden Zahlen mit ausreichender Genauigkeit abgelesen werden können.

Dem gleichen Prinzip, die Übersichtlichkeit zu verbessern, folgen auch die Anforderungen an die Darstellung der Messpunkte. Verschiedene Farben oder Symbole, die entsprechend in der Legende angegeben werden müssen, eignen sich zur Unterscheidung einzelner Messreihen. Sind Fehler vorhanden, sind diese unbedingt mit anzugeben.

Zusammen mit der Bildunterschrift sollten Achsenbeschriftung und Legende den gesamten Sachverhalt, unabhängig vom Text, wiedergeben können. Das bedeutet oft, dass Informationen doppelt genannt werden müssen (im Text

und in der Bildunterschrift).

Zusätzlich zum Eintragen der Messergebnisse ist häufig ein Fit an die Daten vonnöten, um Schlüsse ziehen zu können (so auch im Beispiel unten). Dazu haben die Programme eigene Routinen, die das Berechnen der Parameter übernehmen. Das reduzierte Chi-Quadrat („red. χ^2 “) ist die wichtigste Größe zur Beurteilung der Qualität eines Fits. Es berücksichtigt die einzelnen Fehler der Fitparameter und ist bei guten Fits klein. Falls mit den Werten für die Parameter weiter gerechnet werden soll, sollten diese direkt in der Legende oder in der Bildunterschrift inklusive ihres Fehlers angegeben werden. Pfeile, Linien oder Text im Plot tragen auch zum leichteren Verständnis bei. So dürfen auch schon Ergebnisse (wie zum Beispiel die Identifizierung eines Peaks mit einem Element) eingetragen werden, obwohl sie im Fließtext an der Stelle des Verweises auf die Grafik noch nicht genannt worden sind. Generell muss jedes eingebundene Element, ob Bild, Tabelle oder Diagramm, an einer Stelle im Text erwähnt werden, da sonst - ohne es zu beachten - darüber hinweg gelesen wird.

Beispiel mit Gnuplot

Im Folgenden sollen die genannten Punkte an einem konkreten Beispiel verdeutlicht werden. Als Programm wird Gnuplot verwendet. Der zugrundeliegende Sachverhalt sei die Bestimmung einer Spannung U_0 aus dem Messen der Stromstärke I in Abhängigkeit der angelegten Spannung U . Das bedeutet, der y -Achsenabschnitt, muss ermittelt werden, falls U gegen I aufgetragen wird.

In der Datei „Data.txt“ sind die Messergebnisse zu finden. Mit der Datei „Anweisung für Plot“ kann der in Abbildung 9 gezeigte Plot erstellt werden. Die einzelnen Befehle dienen größtenteils dazu, die Information leichter aus der Grafik entnehmen zu können.

Als Erweiterungen zum Terminal (hier: „aqua“, Standard für Mac, „png-cairo“, „svg“ und weitere für Windows) sind hier „enhanced“ und „dashed“ eingefügt, was Hoch- und Tiefstellen ermöglicht und gestrichelte Linien bietet. Mit dem Befehl „test“ in Gnuplot werden die möglichen Linienarten und weitere Informationen zum verwendeten Terminal angezeigt.

Bei den Achsen wurden zusätzliche Unterteilungen („mxtics“) eingefügt, um die Zahlen leichter ablesen zu können. Die Beschriftungen können mit „offset x,y“ verschoben werden, falls nötig. Die Ausschnitte der Achsen werden meist von Gnuplot passend gewählt, wenn keine Vorgabe gemacht wird. Für diesen Fall ist aber der Wert der Spannung an der Stelle $I = 0$ wichtig, also wird er mit abgebildet.

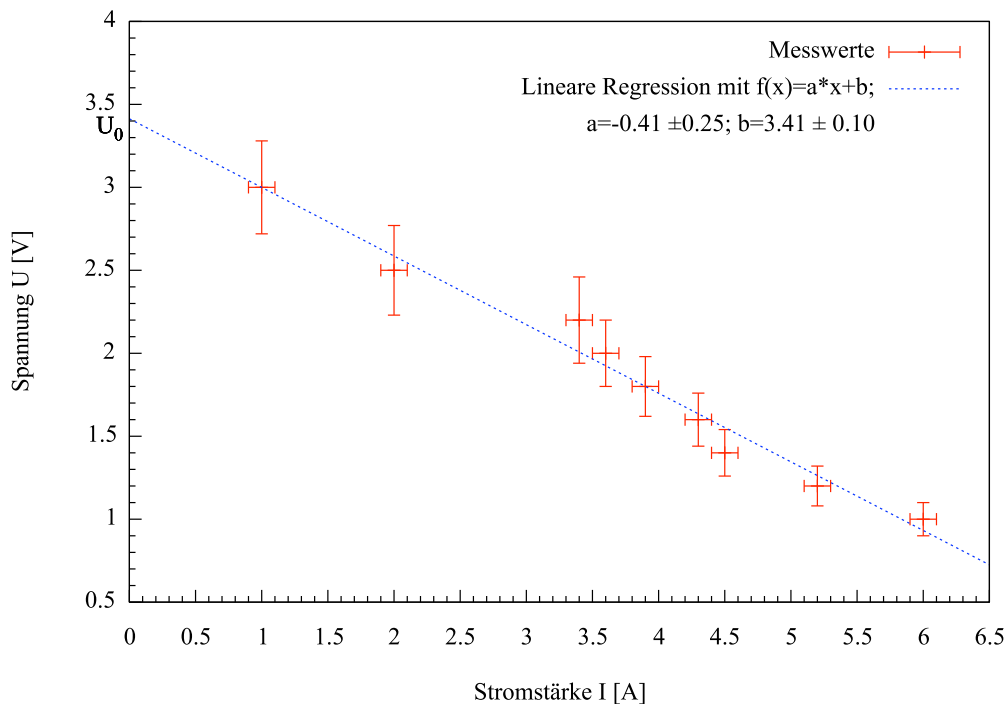


Abbildung 9: Spannung U in Abhängigkeit des gemessenen Stroms I mit linearer Regression. (Bei mehreren Messreihen würden hier noch weitere Informationen über die variierten Größen, wie Widerstand, angegeben werden.)

Anschließend wird die Fitfunktion deklariert und der Fit ausgeführt. Diese Ergebnisse werden in der Konsole angezeigt. Falls die Funktion nicht bekannt ist, die dem physikalischen Modell entsprechen soll, ist darauf zu achten, dass das reduzierte Chi-Quadrat (über „Final set of parameters“) minimal ist. Das Modell (die Funktion) mit dem geringsten Wert eignet sich dann am besten zur Beschreibung der Daten.

Die Legende wurde in die freie Ecke gesetzt und die Fitparameter inklusive der Unsicherheiten direkt mit angegeben. Der Parameter b entspricht dem gesuchten U_0 . Den einzelnen Plotbefehlen können also Namen mitgegeben werden, die direkt die vom Fit bestimmten Variablen enthalten. Die Unsicherheit darauf muss leider, falls benötigt, aus der Konsole ausgelesen werden und manuell eingefügt werden. Das „label“ U_0 wurde an die passende Stelle im Gitter gesetzt.

Beim eigentlichen Plotten werden die Spalten der Quelldatei angegeben, die gegeneinander aufgetragen werden sollen. Ebenso die Unsicherheiten der beiden Werte (oder nur des y -Wertes). Weitere Eigenschaften der Darstellung wie Linienart, -breite, -farbe können hier definiert werden.