

## Lineare Fits mit Octave und Matlab

Die Bestimmung einer Ausgleichsgerade  $y = m \cdot x + b$  durch gemessene Datenpunkte  $(x_i, y_i)$  ist eines der in der Praxis am häufigsten benötigten Verfahren. Für dieses Problem haben die meisten Programme und Taschenrechner zuverlässige eingebaute Funktionen. Anders sieht es aus, wenn die Daten mit Unsicherheiten behaftet sind, d.h.  $(x_i, y_i \pm dy_i)$ ,  $(x_i \pm dx_i, y_i)$  oder sogar  $(x_i \pm dx_i, y_i \pm dy_i)$ .

Treten nur Fehler in y-Richtung auf, so ist das Problem analytisch lösbar<sup>1</sup> und die Unsicherheiten der zu bestimmenden Parameter sind im Rahmen der "Standardfehlerrechnung" eindeutig bestimmt<sup>2</sup>. Liegen nur Fehler in x-Richtung vor, so kann man das Problem trivialerweise durch Vertauschen der Variablen erledigen.

Anders sieht es aus, wenn x und y fehlerbehaftet sind. In diesem Fall gibt es leider keine allgemeingültige analytische Lösung, und man muss numerische Verfahren zur Minimierung von Funktionen verwenden, was nur von wenigen Programmen in geeigneter Weise unterstützt wird. Dies ist besonders ärgerlich, wenn die Fehler von Messpunkt zu Messpunkt variieren, da hier die zu bestimmenden Parameter empfindlich von den einzelnen Fehlern abhängen können<sup>3</sup>.

Für den "Hausgebrauch" habe ich drei unter Matlab und Octave lauffähige Funktionen zum Download bereitgestellt.

Die einfachste ist `FitLin.m`, die die Ausgleichsgerade für Datenpunkte  $(x_i, y_i)$  ohne Fehler berechnet.

Die in `FitLinDy.m` definierte Funktion bestimmt die analytische Lösung für Messdaten, die nur in einer Variablen fehlerbehaftet sind.

Die Funktion in `FitLinDxDy.m` bestimmt die Ausgleichsgerade für den Fall, dass beide Variablen fehlerbehaftet sind. Diese Funktion geht zurück auf die Routine `wtls_line.m` von M. Krystek und M. Anton<sup>4</sup>, die noch ein paar zusätzliche Informationen liefern kann. Ich habe lediglich den Funktionsaufruf für unsere Zwecke etwas vereinfacht.

Damit man den Einfluss der verschiedenen Fehler deutlich sieht, empfiehlt es sich, einen besonders gemeinen Datensatz zu verwenden. Der folgende geht auf Pearson zurück:

---

<sup>1</sup>vgl. Vorlesungsmitschrift oder z.B.: W.R. Leo, *Techniques for Nuclear and Particle Physics Experiments*, §4.7 (Springer 1994)

<sup>2</sup>Da viele Programme die analytische Lösung zu Gunsten eines allgemeineren numerischen Verfahrens ignorieren, ist es in der Praxis oft nicht so *eindeutig*.

<sup>3</sup>Ist dagegen beispielsweise der x-Fehler konstant, so erhält man zumindest vernünftige Schätzwerte für die Parameter (allerdings mit unrealistischen Unsicherheiten), wenn man ihn einfach ignoriert.

<sup>4</sup>M. Krystek and M. Anton, *Meas. Sci. Tech.* 18 (2007), pp3438-3442

```
% x    y    dx    dy
0.0 5.9 0.03162 1.00000
0.9 5.4 0.03162 0.74536
1.8 4.4 0.04472 0.50000
2.6 4.6 0.03536 0.35355
3.3 3.5 0.07071 0.22361
4.4 3.7 0.11180 0.22361
5.2 2.8 0.12910 0.11952
6.1 2.8 0.22361 0.11952
6.5 2.4 0.74536 0.10000
7.4 1.5 1.00000 0.04472
```

Wenn die Prozeduren im aktuellen Arbeitsverzeichnis stehen, kann man sie in Matlab oder Octave einfach aufrufen. Die folgenden Befehle demonstrieren den Umgang mit den Prozeduren:

```
load Pearson
x = Pearson(:,1); y = Pearson(:,2);
dx= Pearson(:,3); dy= Pearson(:,4);

[m1,b1]=FitLin(x,y)
-> y = -0.539577 *x + 5.761185
    m1 = -0.53958
    b1 = 5.7612

[m2,dm2,b2,dm2]=FitLinDy(x,y,dy)
-> slope      m = -0.610813 +- 0.030087
    intercept b = 6.100109 +- 0.204663
    m2 = -0.61081
    dm2 = 0.030087
    b2 = 6.1001
    dm2 = 0.20466

[m3,dm3,b3,dm3]=FitLinDxDy(x,y,dx,dy)
-> m3 = -0.48053
    dm3 = 0.057572
    b3 = 5.4799
    dm3 = 0.29237
```

Die Ergebnisse der verschiedenen Ansätze kann man in der folgenden Graphik vergleichen. Im Allgemeinfall hängen also nicht nur die Unsicherheiten der erhaltenen Parameter von den Messfehlern ab, sondern die Parameter selbst sind von der korrekten Berücksichtigung der Messfehler abhängig. Besonders kritisch wird es, wenn eine Extrapolation der Ausgleichsgerade auf Werte ausserhalb des Messbereichs notwendig ist.

