

Plotten mit Octave und MATLAB

Die Daten mögen in Spaltenvektoren U und I vorliegen. Falls es sich um Daten mit Messunsicherheiten handelt, kommen eventuell noch die Vektoren DU und DI hinzu.

Im Normalfall werden diese Daten in einer 4-spaltigen Datei (z.B. `UI.dat`) vorliegen. Zur einfacheren Handhabung kann man hieraus die einzelnen Vektoren extrahieren:

```
load UI.dat
```

```
U = UI(:,1);  I = UI(:,2);  DU = UI(:,3);  DI = UI(:,4);
```

1. Einfache Datenpunkte

Einen ersten Überblick bekommt man mit `plot(U,I)`

wobei die erhaltene Fieberkurve bei Physikern auf wenig Gegenliebe stoßen wird. Eine erste Verbesserung erreicht man durch Spezifikation eines sogenannten Formatstrings. Mit

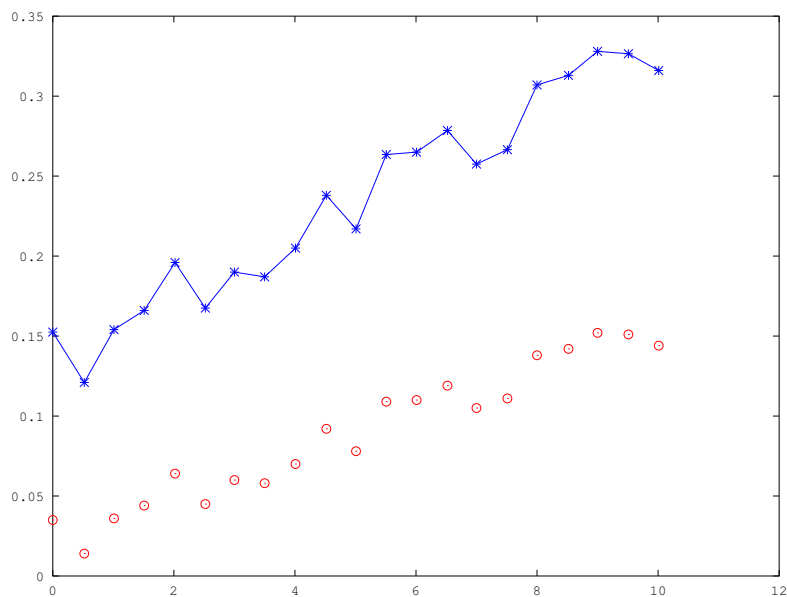
```
plot(U,I,'or')
```

erhält man rote Kringel, mit `'+b'` blaue Kreuze ...

Will man mehrere Datensätze darstellen, so wiederholt man die Elemente im `plot`-Befehl:

```
plot(U,I,'or', U,1.5*I+0.1,'*b-')
```

Die Fieberkurve im blauen Datensatz wurde mit dem `'-'` im Format erreicht. Näheres zu den möglichen Formaten kann man mit `help plot` erfahren.



2. Datenpunkte mit Fehlerbalken

Handelt es sich um Daten mit Unsicherheiten in y-Richtung (hier DI), so wird der Befehl

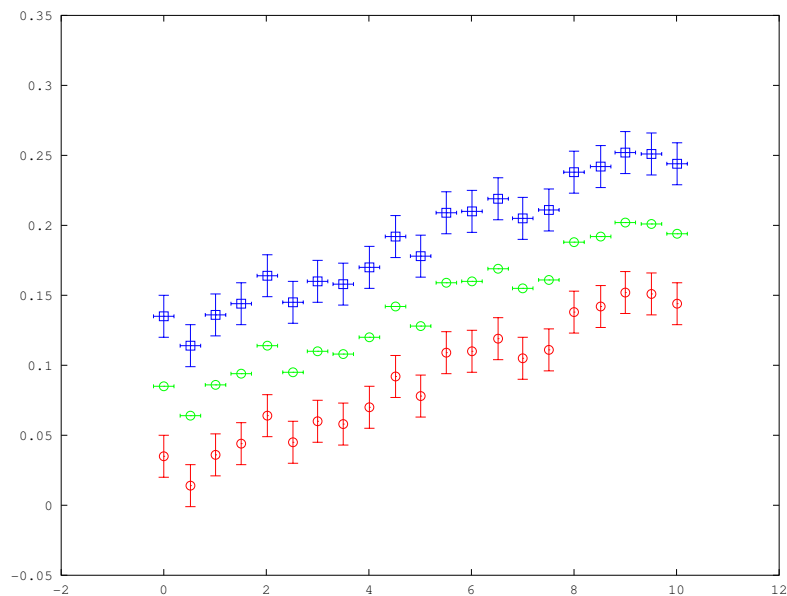
```
errorbar(U,I,DI,'or')
```

verwendet.

Leider scheint Matlab, wie die meisten kommerziellen Programme, Probleme im Umgang mit Messfehlern und Fehlerbalken zu haben, so dass der Rest des Abschnittes sich nur auf Octave bezieht.

In Octave kann der Typ des Fehlerbalkens spezifiziert werden, wobei mit '~' Fehlerbalken in y-Richtung, mit '>' Fehlerbalken in x-Richtung und mit '~>' Fehlerbalken in beiden Richtungen dargestellt werden. Das Beispiel zeigt alle Varianten in einem Plot:

```
errorbar(U,I,DI,'~or',U,I+0.05,DU,'>og',U,I+0.1,DU,DI,'~>sb')
```



3. Minimale Beschriftung

Eine vernünftige Grafik erfordert ein Minimum an Beschriftung. Neben einer Überschrift müssen zumindest die Achsen beschriftet werden. Dies erreicht man mit

```
title('Strom-Spannungskennlinie BX3')
xlabel('U (V)')
ylabel('I (A)')
```

Für veröffentlichungsreife Darstellungen gibt es noch zahlreiche Optionen bei den genannten Befehlen, sowie etliche, die mit der Funktion `set` gesetzt werden können. Im Prinzip ist fast alles möglich, was auch in `gnuplot` möglich ist, wobei es jedoch häufig effizienter ist, die Daten zu exportieren und mit `gnuplot` oder einem anderen Grafikprogramm weiter zu arbeiten.

Oft kann man die Grafik aber auch durch entsprechende Optionen bei der Ausgabe hinreichend “aufpeppen” (s.u.).

4. Exportieren der Grafik

Die wichtigste Frage ist sicherlich, wie man die mühsam errechnete Grafik auf das Papier bekommt. Dies kann man man mit dem `print` Befehl erreichen, der die gerade dargestellte Grafik in verschiedenen Formaten exportieren kann. Neben den (encapsulated) PostScript in Farbe (`eps`) und Schwarz-Weiß (`eps`) sind hier auch `ps`, `jpg`, `png`, `emf`, `pdf`, `svg` und zahlreiche andere Formate möglich. Die obigen Grafiken wurden zum Beispiel mit

```
print UI1.eps -depsc
```

erzeugt und mittels `\includegraphics` in \LaTeX eingebunden.

Die Originalgröße der Darstellung kann man mit der Option `-Ssize,ysize` einstellen (für `ps`, `eps`, `pdf` in Points, für `jpg`, `emf`, `svg` in Pixeln) einstellen. Die Schrift und Schriftgröße kann z.B. mit `-F:12` oder `-Ftimes:12` eingestellt werden. Mit

```
print UI1n.eps -depsc2 "-S950,600" -F:12
```

erhält man so das Resultat auf der nächsten Seite, dessen Qualität für die meisten Zwecke ausreichen sollte.

5. logarithmische Skalierung und ...

Es gibt zahlreiche Varianten der obigen Plotbefehle, deren Dokumentation man mittels der Befehle `doc` und `help` sowie aus den einschlägigen Anleitungen erhalten kann. Zu den wichtigeren gehören Plots, bei denen eine oder beide Achsen des Koordinatensystems logarithmisch geteilt sind. Die zugehörigen Befehle sind

```
semilogx, semilogy, loglog
```

und die entsprechenden Befehle mit Fehlerbalken:

```
semilogxerr, semilogyerr, loglogerr
```

Häufig benötigt wird auch die Funktion `hist` für die Darstellung von Histogrammen, wogegen Funktionen wie `pie`, `bar`, `polar` und andere seltener benötigt werden.

Für 3d-Darstellungen laden `mesh`, `plot3`, `surf`, `contour` und andere zum Experimentieren ein.

