

FH Schmalkalden
Fak. Elektrotechnik

Einführung in pst-plot

Inhaltsverzeichnis

1. Einleitung	4
1.1. Warum pst-plot?	4
1.2. Zusätzliche Literatur	4
2. Benutzung von pst-plot	5
2.1. Nutzung von pst-plot mit LaTeX	5
2.2. Nutzung von pst-plot mit pdfLaTeX	6
2.3. Iterationszyklen verkürzen	7
3. Diagramme	8
3.1. Diagrammarten	8
4. Realisierungsbeispiele für die Diagrammarten	8
4.1. Problemstellung	8
4.2. Qualitative Darstellung	9
4.3. Quantitative Darstellung	11
4.4. Arbeitsdiagramm	16
5. Selbst definierte Funktionen	20
6. Mehrere Kurven in einem Diagramm	22
7. Daten plotten	29
7.1. Daten	29
7.2. Erster Plot	30
7.3. Skalierung	32
7.4. Messwerte als Punkte plotten	36
7.5. Interpolationskurve erzeugen	38
8. Parametrische Plots	40
A. Pfeile an Linienenden	42

1. Einleitung

1.1. Warum pst-plot?

GnuPlot ist ein Programm zum Anzeigen und Drucken mathematischer Zusammenhänge. Es ist gut geeignet, um diese „auf die Schnelle“ zu visualisieren, um Lösungsansätze zu finden oder Anschauungsmaterialien zu erstellen.

Im Bereich der technischen Dokumentationen sollten Diagramme aber entsprechend DIN 461 angefertigt werden. Dies gilt auch für „professionelle Präsentationen“ sowie Qualifikationsnachweise im Rahmen einer Ausbildung (z. B. Beleg-, Diplom- oder Bachelorarbeiten).

Mit dem Paket pst-plot, das Bestandteil des PSTricks-Bundles ist, können DIN-gerechte Diagramme für die Einbindung in \LaTeX -Dokumente erstellt werden.

Die hier vorliegende Anleitung soll Nutzern den Einstieg in pst-plot erleichtern. Eine vollständige Behandlung des pst-plot-Paketes ist nicht das Ziel dieser Anleitung, hierfür sollte die u. a. Literatur herangezogen werden.

1.2. Zusätzliche Literatur

Zusätzlich zu dieser Anleitung sollten Sie folgende Literatur in Reichweite haben:

- DIN 461: Graphische Darstellung in Koordinatensystemen
Diese Norm legt fest, wie Diagramme zu zeichnen und zu beschriften sind.
- [ASI99]
Hier finden Sie im Kapitel 8.1 eine Aufstellung der PostScript-Operatoren. Für die Arbeit mit pst-plot werden insbesondere die Operatoren zur Stack-Manipulation und die mathematischen Operatoren benötigt.
- [IK95]
Dieses Buch kann als Lehrbuch für PostScript verwendet werden. Es ist meiner Meinung nach einfacher verständlich als die vorhergehend genannte PostScript-Referenz. Von Interesse sind auch hier besonders die Kapitel zu Stack-Manipulation und Operatoren.
- [TVZ06]
Diese Dokumentation enthält ausführliche Beschreibungen der hier benutzten und vorgestellten PSTricks-Befehle.

Für die Einarbeitung in PSTricks und pst-plot fand ich folgende Online-Ressourcen besonders hilfreich:

- <http://tug.org/PSTricks/>
enthält viel Material rund um PSTricks.
- http://sarovar.org/project/showfiles.php?group_id=12&release_id=104
PDF-Präsentationen der indischen \TeX user group.

2. Benutzung von pst-plot

2.1. Nutzung von pst-plot mit LaTeX

Um PSTricks mit L^AT_EX zu nutzen, werden folgende Packages in der Dokument-Präambel eingebunden:

```
1 \usepackage{ifthen}
2 \usepackage{calc}
3 \usepackage{pst-plot}
```

Ein einfaches Bild wird erzeugt mit

```
1 \begin{figure}
2   {\centering
3   \begin{pspicture}(-0.2,-0.2)(4.2,2,2)
4   \pspolygon(0,0)(4,0)(4,2)(0,2)
5   \end{pspicture}
6   \caption{Test mit PSTricks}
7   \label{fig:test-pstricks-1}
8   }
9 \end{figure}
```

Mit

```
1 latex test && dvips test
```

wird die Druckdatei erstellt.



Abbildung 1: Test mit PSTricks

2.2. Nutzung von pst-plot mit pdfLaTeX

Um das PSTricks-Bundle mit pdfL^AT_EX zu nutzen, wird das Paket pst-pdf benötigt. Dieses muss nach den anderen PSTricks-Paketen eingebunden werden.

```
1 \usepackage{ifthen}
2 \usepackage{calc}
3 \usepackage{pst-plot}
4 \usepackage{pst-pdf}
```

Alle PostScript-spezifischen Teile – insbesondere die *pspicture*-Umgebungen – werden in *postscript*-Umgebungen eingefügt.

```
1 \begin{figure}
2   {\centering
3   \begin{postscript}
4     \begin{pspicture}(-0.2,-0.2)(4.2,2.2)
5       \pspolygon(0,0)(4,0)(4,2)(0,2)
6     \end{pspicture}
7   \end{postscript}
8   \caption{Test mit PSTricks}
9   \label{fig:test-pstricks-1}
10  }
11 \end{figure}
```

Eine Datei test.tex wird nun mit den Kommandos

```
1 latex test
2 dvips -o test-pics.ps test.dvi
3 ps2pdf test-pics.ps
4 pdflatex test
5 pdflatex test
```

in das PDF-Format übersetzt. Unter Windows muss Zeile 3 ersetzt werden durch

```
1 call ps2pdf test-pics.ps
```

wenn die Kommandos durch eine *.bat-Datei aufgerufen werden.

2.3. Iterationszyklen verkürzen

Meist entsteht ein PSTricks-Bild iterativ, d. h. die Schritte

- Eingabe bzw. Änderung des PSTricks-Codes
- Übersetzung des Dokumentes
- Korrekturlesen und Überprüfen der Bilder

werden mehrfach durchlaufen.

Für Dokumente mit hoher Seitenzahl bzw. mehreren Bildern ist dieser Prozess zeitaufwendig. Die für das Bearbeiten von Bildern benötigte Zeit kann verkürzt werden, indem beim Erstellen von PSTricks-Bildern nicht das gesamte zu erstellende Dokument übersetzt wird sondern ein separates Dokument bild.tex, welches nur das Bild enthält. Die Datei bild.pdf kann dann mit den Kommandos

```
1 pdflatex bild
2 latex bild
3 dvips -o bild-pics.ps bild.dvi
4 ps2pdf bild-pics.ps
5 pdflatex bild
6 pdflatex bild
```

erstellt werden. Wird diese Kommandofolge unter Windows als *.bat-Datei gespeichert, muss Zeile 4 lauten:

```
1 call ps2pdf bild-pics.ps
```

3. Diagramme

3.1. Diagrammarten

DIN 461 unterscheidet zwischen folgenden Arten von Diagrammen:

- Qualitative Darstellung
Der charakteristische Verlauf wird als Übersichtsdiagramm dargestellt, ohne dass Rückschlüsse auf Zahlenwerte möglich sind.
- Quantitative Darstellung
Die Skalen dieser Diagramme sind mit einer Skalenteilung versehen, so dass Zahlenwerte abgelesen werden können.
- Arbeitsdiagramm
Hierbei handelt es sich um quantitative Darstellungen, aus denen Zahlenwerte für weitere Berechnungen mit ausreichender Stellenzahl abgelesen werden sollen.

4. Realisierungsbeispiele für die Diagrammarten

4.1. Problemstellung

Im Beispiel soll der Gesamtwiderstand R_{ges} zweier parallel geschalteter Widerstände R_1 und R_2 dargestellt werden. Dabei ist $R_1 = 10\Omega$, R_2 ist variabel mit $0 < R_2 \leq 10\Omega$.

Die Diagramme werden aufgelöst mit $\frac{1\Omega}{1\text{cm}}$, die Diagrammfläche wird also 10 cm breit und 5 cm hoch. In der entstehenden Abbildung muss dann noch weiterer Platz, z. B. für Beschriftungen und Pfeilspitzen vorgesehen werden.

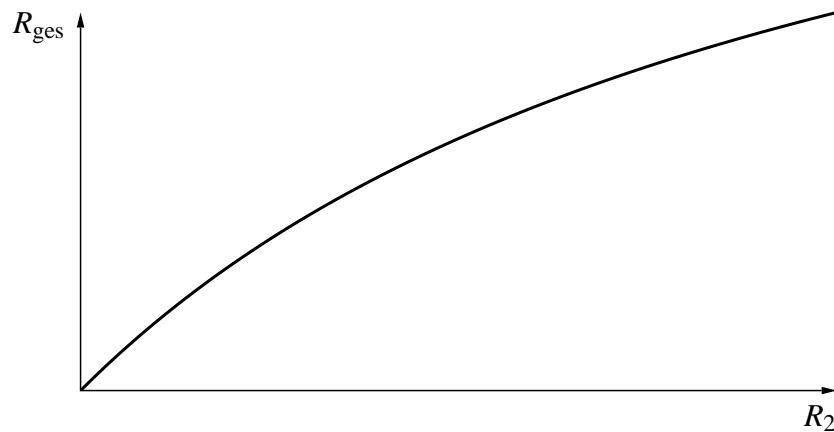


Abbildung 2: Qualitative Darstellung

4.2. Qualitative Darstellung

In der qualitativen Darstellung wird der charakteristische Kurvenverlauf dargestellt. Beide Achsen sind linear geteilt.

Aus der Abbildung 2 und dem Quelltext 1 auf der nächsten Seite kann folgendes entnommen werden:

- Die PSTricks-Anweisungen, die das Bild erzeugen, stehen in einer *pspicture*-Umgebung, diese wiederum in einer *postscript*-Umgebung. Die Argumente beim Öffnen der *pspicture*-Umgebung geben den linken unteren und den rechten oberen Bildpunkt in PSTricks-Einheiten an. Die Länge dieser Einheit wird in Zeile 4 auf 1 cm gesetzt (1 cm ist auch der Standardwert, deshalb wäre die Anweisung hier eigentlich nicht nötig).
- Die *psline*-Anweisung in Zeile 12 erzeugt die beiden Achsen, es wird zunächst eine Linie von (0,5) nach (0,0) gezogen, daran schließt sich ein weiterer Linienzug nach (10,0) an. Das Argument „arrows=<->“ sorgt für Pfeilspitzen an beiden Enden des Linienzuges. Die Form der Pfeilspitze wird über die Angaben „arrowsize“, „arrowlength“ und „arrowinset“ in der *psset*-Anweisung eingestellt, siehe auch Anhang A auf Seite 42.
- Die *psplot*-Anweisung in Zeile 14 besorgt das Plotten. Die Argumente „0.01“ und „10.0“ geben den Start- und Endwert für x an. Das dritte Argument enthält PostScript-Code, um zu einem x -Wert den zugehörigen y -Wert zu berechnen. Der spezielle PostScript-Ausdruck „x“ liefert dabei den x -Wert, für den jeweils die Berechnung erfolgen soll. Das optionale Argument „plotstyle=curve“ legt fest, dass die berechneten Punkte durch eine Kurve miteinander verbunden werden sollen.

Listing 1: Qualitative Darstellung

```

1 \begin{ figure }
2 {\centering
3 \begin{ postscript }
4 \psset{ unit=1.0cm}
5 \begin{ pspicture }(-2, -1.2)(10.2,5.2)
6 \psset{%
7 linewidth=0.2mm,%
8 arrowsize=1.0mm 0.0,%
9 arrowlength=2.0,%
10 arrowinset=0.0%
11 }
12 \psline[arrows=<->](0,5)(0,0)(10,0)
13 \psset{ linewidth=0.4mm}
14 \psplot[plotstyle=curve]{0.01}{10.0}{x 10 mul x 10 add div}
15 \rput[tr]{0}(10, -0.2){\(\mathbb{R}_2\)}
16 \rput[tr]{0}(-0.2,5){\(\mathbb{R}_{\text{ges}}\)}
17 \end{ pspicture }
18 \end{ postscript }
19 \caption{ Qualitative Darstellung }
20 \label{ fig:fig0002 }
21 }
22 \end{ figure }

```

- Die Liniendicke wird jeweils mit *psset*-Anweisungen eingestellt, bevor die entsprechenden Linien gezogen werden (Zeilen 6 und 8). Als Richtwert gilt, dass der Kurvenverlauf ca. doppelt so dick gezeichnet wird wie die Achsen.
- Die *rput*-Anweisungen in den Zeilen 15 und 16 erzeugen die Achsenbeschriftungen.
Die Beschriftungen befinden sich unter bzw. neben der Pfeilspitze und schließen rechtsbündig (Beschriftung der *x*-Achse) bzw. oben bündig (*y*-Achse) mit der Pfeilspitze ab.
- Bei einer qualitativen Darstellung werden immer lineare Skalenteilungen verwendet.

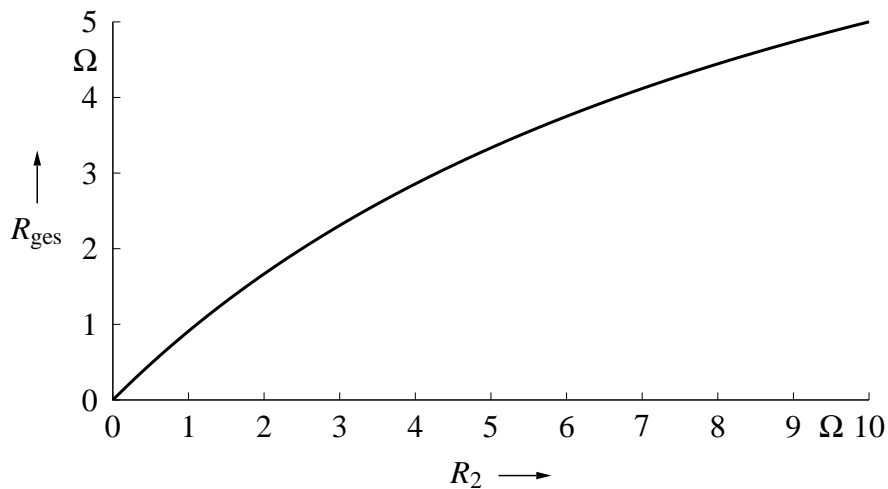


Abbildung 3: Quantitative Darstellung

4.3. Quantitative Darstellung

- In der quantitativen Darstellung sind an beiden Achsen Skalenteilungen angebracht, die beschriftet sind.
- Parallel zur x - und y -Achse verlaufen Pfeile, die angeben, in welcher Richtung die dargestellte Größe wächst.
- Am Fuß dieser Pfeile steht die Größe, die von der jeweiligen Achse dargestellt wird.
- Die zugehörige Einheit wird zwischen die vorletzte und die letzte Achsenbeschriftung geschrieben.

- Um die Achsenteilungen nicht alle einzeln als Linie zu zeichnen, kommen while-Schleifen zum Einsatz. Für diese wird eine Variable in Form eines Zählers benötigt.

Dieser Zähler wird in Zeile 1 mit der *newcounter*-Anweisung angelegt. Der Name muss so gewählt werden, dass er nicht mit anderen Bezeichner kollidiert.

Die erste while-Schleife verläuft von Zeile 9 bis Zeile 14. In Zeile 9 wird der Zähler mit *setcounter* auf den Startwert gesetzt.

Das erste Argument der *whiledo*-Anweisung ist die zu prüfende Bedingung. Das zweite Argument ist eine Folge von \LaTeX -Befehlen.

Mit der *value*-Anweisung wird der Wert des Zählers als numerischer Wert für Berechnungen und Tests ermittelt.

Im Schleifenkörper wird mit *psline* eine kurze Linie gezeichnet. Mit *rput* wird

das zugehörige Label gedruckt. Dabei wird im Label-Text nicht die *value*-Anweisung verwendet sondern „\theVariablename“, da hier der Variableninhalt als Text verwendet werden soll. Anschließend wird mit *addcounter* der Zähler erhöht.

- Für die Liniendicken von Skalenteilungen, Achsen und Kurven gilt:

$$\text{Gitternetz bzw. Skalenteilung} : \text{Achsen} : \text{Kurven} = 1 : 2 : 4$$

Listing 2: Qualitative Darstellung, Einheit und Größe in Bruchdarstellung

```

1 \newcounter{myvaraaaa}
2 \begin{figure}
3 {\centering
4 \begin{postscript}
5 \psset{unit=1.0cm}
6 \begin{pspicture}(-1.8,-1.8)(11,6)
7 \psset{linewidth=0.1mm}
8 % x-axis labels and tics
9 \setcounter{myvaraaaa}{0}
10 \whiledo{\value{myvaraaaa}<11}{%
11 \psline(\value{myvaraaaa},0.0)(\value{myvaraaaa},0.1)%
12 \rput[t]{0}(\value{myvaraaaa},-0.2){\themyvaraaaa}%
13 \addtocounter{myvaraaaa}{1}%
14 }
15 % y-axis labels and tics
16 \setcounter{myvaraaaa}{0}
17 \whiledo{\value{myvaraaaa}<6}{%
18 \psline(0,\value{myvaraaaa})(0.1,\value{myvaraaaa})%
19 \rput[r]{0}(-0.2,\value{myvaraaaa}){\themyvaraaaa}%
20 \addtocounter{myvaraaaa}{1}%
21 }
22 % arrows
23 \psset{%
24 linewidth=0.2mm,%
25 arrowsize=1.0mm 0.0,%
26 arrowlength=2.0,%
27 arrowinset=0.0%
28 }
29 \psline(0,5)(0,0)(10,0)
30 \psline[arrows=->](5.1,-1)(5.8,-1)
31 \psline[arrows=->](-1,2.6)(-1,3.3)
32 \rput[r]{0}(4.9,-1){\(\mathbb{R}_2\)}
33 \rput[t]{0}(9.5,-0.2){\(\Omega\)}
34 \rput[t]{0}(-1,2.4){\(\mathbb{R}_{\text{ges}}\)}
35 \rput[r]{0}(-0.2,4.5){\(\Omega\)}
36 \psset{linewidth=0.4mm}
37 \psplot[plotstyle=curve]{0.01}{10.0}{x 10 mul x 10 add div}
38 \end{pspicture}
39 \end{postscript}
40 \caption{Quantitative Darstellung}
41 \label{fig:fig0003}
42 }
43 \end{figure}

```

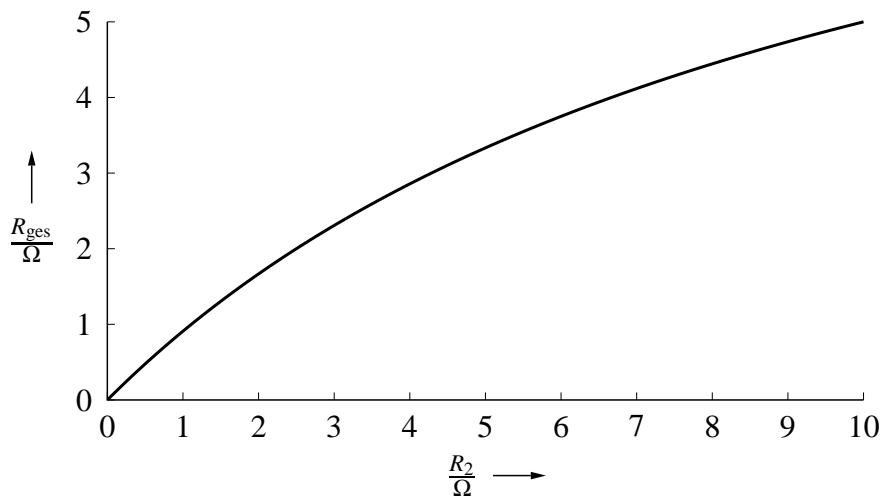


Abbildung 4: Quantitative Darstellung

Ist zwischen der vorletzten und der letzten Beschriftung nicht ausreichend Platz zum Setzen der Einheit, wird der parallel zur Achse verlaufende Pfeil mit einem Bruch Größe / Einheit beschriftet.

Alternativ kann auch die vorletzte Skalenbeschriftung – und weitere davor liegende Beschriftungen, falls nötig – weggelassen werden, um Platz für die Einheitenangabe zu schaffen. Dabei darf aber auf keinen Fall die Beschriftung „0“ weggelassen werden.

Listing 3: Qualitative Darstellung

```

1 \newcounter{myvaraaaab}
2 \begin{figure}
3 {\centering
4 \begin{postscript}
5 \psset{unit=1.0cm}
6 \begin{pspicture}(-1.8,-1.8)(11,6)
7 \psset{linewidth=0.1mm}
8 % x-axis labels and tics
9 \setcounter{myvaraaaab}{0}
10 \whiledo{\value{myvaraaaab}<11}{%
11 \psline(\value{myvaraaaab},0.0)(\value{myvaraaaab},0.1)%
12 \rput[t]{0}(\value{myvaraaaab},-0.2){\themyvaraaaab}%
13 \addtocounter{myvaraaaab}{1}%
14 }
15 % y-axis labels and tics
16 \setcounter{myvaraaaab}{0}
17 \whiledo{\value{myvaraaaab}<6}{%
18 \psline(0,\value{myvaraaaab})(0.1,\value{myvaraaaab})%
19 \rput[r]{0}(-0.2,\value{myvaraaaab}){\themyvaraaaab}%
20 \addtocounter{myvaraaaab}{1}%
21 }
22 % arrows
23 \psset{%
24 linewidth=0.2mm,%
25 arrowsize=1.0mm 0.0,%
26 arrowlength=2.0,%
27 arrowinset=0.0%
28 }
29 \psline(0,5)(0,0)(10,0)
30 \psline[arrows=->](5.1,-1)(5.8,-1)
31 \psline[arrows=->](-1,2.6)(-1,3.3)
32 \rput[r]{0}(4.9,-1){\(\frac{R_2}{\Omega}\)}
33 \rput[t]{0}(-1,2.4){\(\frac{R_{\text{ges}}}{\Omega}\)}
34 \psset{linewidth=0.4mm}
35 \psplot[plotstyle=curve]{0.01}{10.0}{x 10 mul x 10 add div}
36 \end{pspicture}
37 \end{postscript}
38 \caption{Quantitative Darstellung}
39 \label{fig:fig0004}
40 }
41 \end{figure}

```

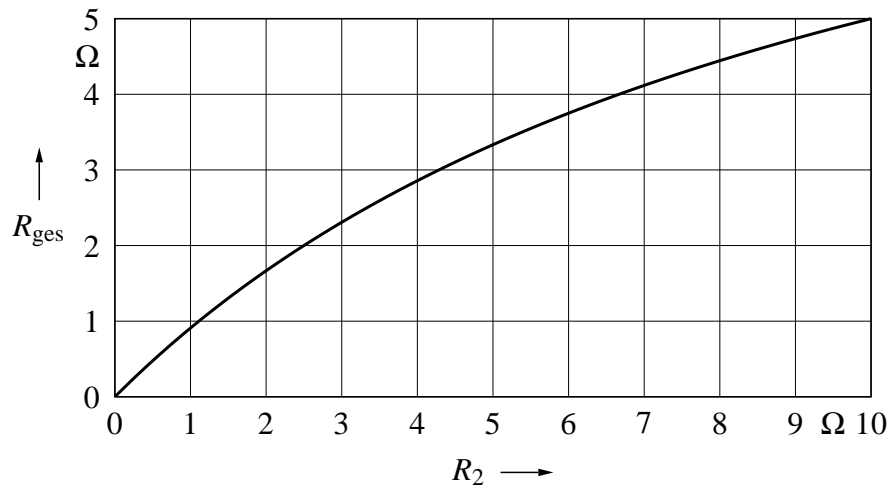


Abbildung 5: Arbeitsdiagramm

4.4. Arbeitsdiagramm

- Das Paket `pst-plot` verfügt zwar über einen Befehl `psgrid` zum Zeichnen eines Gitters. Allerdings werden dabei die Skalenbeschriftungen nicht normgerecht zentriert unter bzw. neben die Skalenstriche gesetzt sondern linksbündig bzw. unten bündig. Deshalb wird auf die Nutzung des Befehles verzichtet. Stattdessen werden die Gitternetzlinien mit `while`-Schleifen gezeichnet wie in den vorangegangenen Beispielen die Skalenteilstriche.
- Die Kurve soll mit möglichst geringer Dicke gezeichnet werden, um eine hohe Ablesegenauigkeit zu erzielen.

Listing 4: Arbeitsdiagramm

```

1 \newcounter{myvaraaaad}
2 \begin{figure}
3   {\centering
4   \begin{postscript}
5     \psset{unit=1.0cm}
6     \begin{pspicture}(-3,-3)(11,6)
7       \psset{linewidth=0.1mm}
8       % vertical grid lines and x-axis labels
9       \setcounter{myvaraaaad}{0}
10      \whiledo{\value{myvaraaaad}<11}{%
11        \psline(\value{myvaraaaad},0)(\value{myvaraaaad},5)%
12        \rput[t]{0}(\value{myvaraaaad},-0.2){\themyvaraaaad}%
13        \addtocounter{myvaraaaad}{1}%
14      }
15      % horizontal grid lines and y-axis labels
16      \setcounter{myvaraaaad}{0}
17      \whiledo{\value{myvaraaaad}<6}{%
18        \psline(0,\value{myvaraaaad})(10,\value{myvaraaaad})%
19        \rput[r]{0}(-0.2,\value{myvaraaaad}){\themyvaraaaad}%
20        \addtocounter{myvaraaaad}{1}%
21      }
22      % grid border
23      \psset{%
24        linewidth=0.2mm,%
25        arrowsize=1.0mm 0.0,%
26        arrowlength=2.0,%
27        arrowinset=0.0%
28      }
29      \pspolygon(0,0)(10,0)(10,5)(0,5)
30      % arrows
31      \psline[arrows=->](5.1,-1)(5.8,-1)
32      \psline[arrows=->](-1,2.6)(-1,3.3)
33      \rput[r]{0}(4.9,-1){\(\text{R}_2\)}
34      \rput[t]{0}(9.5,-0.2){\(\text{\Omega}\)}
35      \rput[t]{0}(-1,2.4){\(\text{R}_{\text{ges}}\)}
36      \rput[r]{0}(-0.2,4.5){\(\text{\Omega}\)}
37      \psset{linewidth=0.4mm}
38      \psplot[plotstyle=curve]{0.01}{10.0}{x 10 mul x 10 add div}
39    \end{pspicture}
40  \end{postscript}
41  \caption{Arbeitsdiagramm}
42  \label{fig:0006}
43 }
44 \end{figure}

```

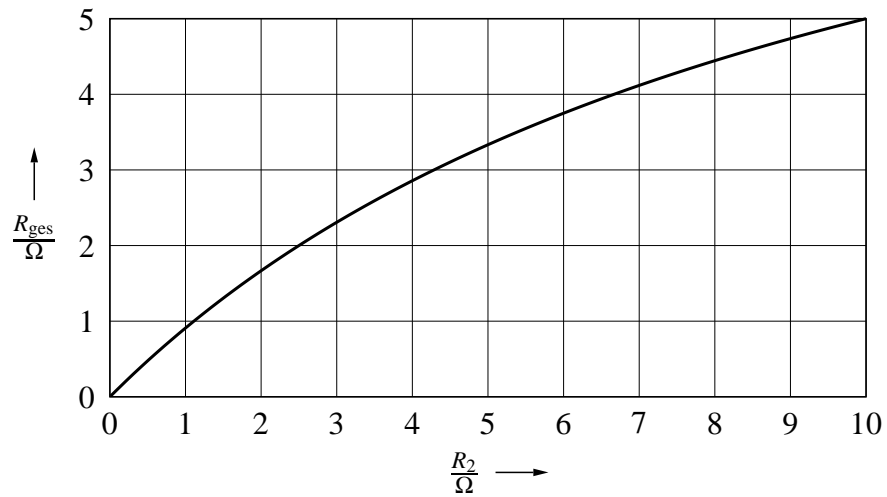


Abbildung 6: Arbeitsdiagramm

- Reicht der Platz zwischen vorletzter und letzter Skalenbeschriftung nicht aus, um die Einheit dort zu setzen, kann die Beschriftung am Fuß des Pfeiles als Bruch Größe / Einheit erfolgen.
Die graphische Darstellung bezieht sich dann auf die Zahlenwerte.

Listing 5: Quantitative Darstellung, Darstellung von Größe und Einheit als Bruch

```

1 \newcounter{myvaraaaae}
2 \begin{figure}
3 {\centering
4 \begin{pscript}
5 \psset{unit=1.0cm}
6 \begin{pspicture}(-3,-3)(11,6)
7 \psset{linewidth=0.1mm}
8 % vertical grid lines and x-axis labels
9 \setcounter{myvaraaaae}{0}
10 \whiledo{\value{myvaraaaae}<11}{%
11 \psline(\value{myvaraaaae},0)(\value{myvaraaaae},5)%
12 \rput[t]{0}(\value{myvaraaaae},-0.2){\themyvaraaaae}%
13 \addtocounter{myvaraaaae}{1}%
14 }
15 % horizontal grid lines and y-axis labels
16 \setcounter{myvaraaaae}{0}
17 \whiledo{\value{myvaraaaae}<6}{%
18 \psline(0,\value{myvaraaaae})(10,\value{myvaraaaae})%
19 \rput[r]{0}(-0.2,\value{myvaraaaae}){\themyvaraaaae}%
20 \addtocounter{myvaraaaae}{1}%
21 }
22 % grid border
23 \psset{%
24 linewidth=0.2mm,%
25 arrowsize=1.0mm 0.0,%
26 arrowlength=2.0,%
27 arrowinset=0.0%
28 }
29 \pspolygon(0,0)(10,0)(10,5)(0,5)
30 % arrows
31 \psline[arrows=->](5.1,-1)(5.8,-1)
32 \psline[arrows=->](-1,2.6)(-1,3.3)
33 \rput[r]{0}(4.9,-1){\(\frac{R_2}{\Omega}\)}
34 \rput[t]{0}(-1,2.4){\(\frac{R_{\text{ges}}}{\Omega}\)}
35 \psset{linewidth=0.4mm}
36 \psplot[plotstyle=curve]{0.01}{10.0}{x 10 mul x 10 add div}
37 \end{pspicture}
38 \end{pscript}
39 \caption{Arbeitsdiagramm}
40 \label{fig:0007}
41 }
42 \end{figure}

```

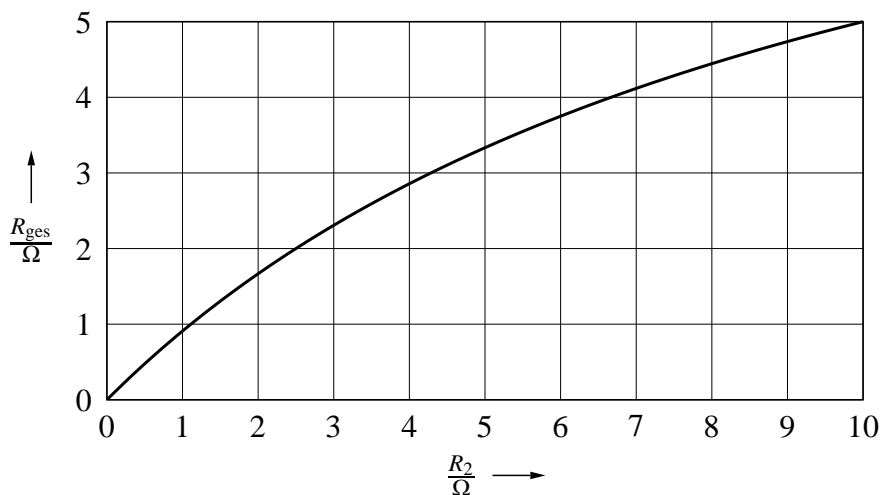


Abbildung 7: Arbeitsdiagramm

5. Selbst definierte Funktionen

Eigene PostScript-Funktionen und -Konstanten können in einer `\special`-Anweisung definiert werden.

Dass es sich um PostScript-Specials handelt, wird durch das „ps:“ am Beginn des Specials kenntlich gemacht. Der Rest des Specials besteht aus PostScript-Code, der von dvips unverändert in die Ausgabe übernommen wird (Zeilen 7 bis 19).

Wichtig ist, dass dieser Code keine Leerzeilen enthält, da diese von L^AT_EX zu „\par“-Anweisungen umgewandelt werden, die dann mit in die PostScript-Ausgabe übernommen würden, wo sie natürlich stören.

Werden Leerzeilen benötigt, um den Code übersichtlicher zu gestalten, muss die Leerzeile wenigstens ein Prozentzeichen am Anfang enthalten. Dieses leitet sowohl für PostScript als auch für L^AT_EX einen Kommentar ein.

Listing 6: Selbst definierte PostScript-Funktion

```

1 \newcounter{myvaraaaaf}
2 \begin{figure}
3 {\centering
4 \begin{pscript}
5 \psset{unit=1.0cm}
6 \begin{pspicture}(-3,-3)(11,6)
7 \special{ps:
8 % Konstante, Wert für R_1
9 /Wert1 10 def
10 %
11 % Funktion zur Widerstandsberechnung
12 /ParallelWiderstand {
13   dup Wert1 mul exch Wert1 add div
14 } def
15 %
16 }
17 \psset{linewidth=0.1mm}
18 \setcounter{myvaraaaaf}{0}
19 \whiledo{\value{myvaraaaaf}<11}{%
20 \psline(\value{myvaraaaaf},0)(\value{myvaraaaaf},5)%
21 \rput[t]{0}(\value{myvaraaaaf},-0.2){\themyvaraaaaf}%
22 \addtocounter{myvaraaaaf}{1}%
23 }
24 \setcounter{myvaraaaaf}{0}
25 \whiledo{\value{myvaraaaaf}<6}{%
26 \psline(0,\value{myvaraaaaf})(10,\value{myvaraaaaf})%
27 \rput[r]{0}(-0.2,\value{myvaraaaaf}){\themyvaraaaaf}%
28 \addtocounter{myvaraaaaf}{1}%
29 }
30 \psset{%
31 linewidth=0.2mm,arrowsize=1.0mm 0.0,arrowlength=2.0,%
32 arrowinset=0.0%
33 }
34 \pspolygon(0,0)(10,0)(10,5)(0,5)
35 \psline[arrows=->](5.1,-1)(5.8,-1)
36 \psline[arrows=->](-1,2.6)(-1,3.3)
37 \rput[r]{0}(4.9,-1){\(\frac{R_2}{\Omega}\)}
38 \rput[t]{0}(-1,2.4){\(\frac{R_{\text{ges}}}{\Omega}\)}
39 \psset{linewidth=0.4mm}
40 \psplot[plotstyle=curve]{0.01}{10.0}{x ParallelWiderstand}
41 \end{pspicture}
42 \end{pscript}
43 \caption{Arbeitsdiagramm}
44 \label{fig:0008}
45 }
46 \end{figure}

```

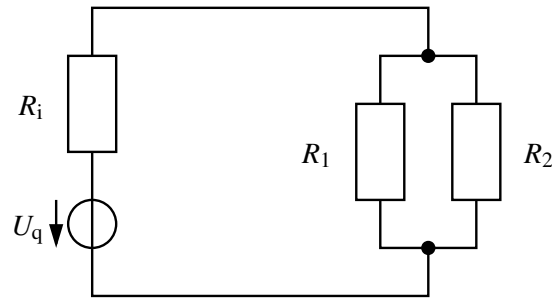


Abbildung 8: Parallelwiderstand versorgt von Spannungsquelle

6. Mehrere Kurven in einem Diagramm

Werden zwei Kurven in einem Diagramm dargestellt, werden häufig zwei Skalen für die y-Achse benötigt.

Im Beispiel wird die bereits bekannte Parallelschaltung zweier Widerstände an einer Spannungsquelle $U_q = 1\text{ V}$ mit einem Innenwiderstand $R_i = 2\Omega$ betrieben. Im Diagramm sollen sowohl der bisher schon dargestellte Gesamtwiderstand der Parallelschaltung und die am Parallelwiderstand umgesetzte Verlustleistung eingezeichnet werden.

$$R_{\text{ges}} = \frac{R_1 R_2}{R_1 + R_2}$$

$$I = \frac{U_q}{R_i + R_{\text{ges}}}$$

$$U_{\text{ges}} = \frac{U_q R_{\text{ges}}}{R_i + R_{\text{ges}}}$$

$$P = \frac{U_q^2 R_{\text{ges}}}{(R_i + R_{\text{ges}})^2}$$

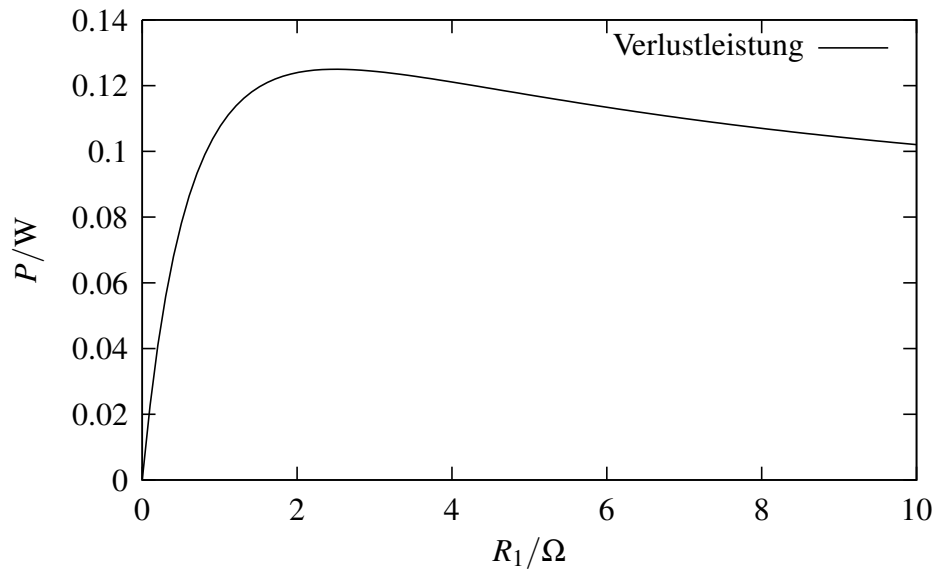


Abbildung 9: Darstellung der Verlustleistung mit GnuPlot

Mit GnuPlot wird eine erste Übersichtsdarstellung erzeugt, siehe Abb. 9.

Listing 7: Darstellung der Verlustleistung mit GnuPlot

```

1 set terminal mp c latex psnfss
2 set output "f0002.mp"
3 set xlabel "\\(R_1/\\Omega\\)"
4 set ylabel "\\(P/\\text{W}\\)"
5 set key left top right
6 show key
7 set data style lines
8 par(x,y)=((x>0.0)?((y>0.0)?((x*y)/(x+y)):0.0):0.0)
9 verlust(uq,ri,ra,rb)=((uq**2)*par(ra,rb))/((ri+par(ra,rb))**2)
10 plot [0:10] verlust(1.0,2.0,x,10.0) title "Verlustleistung"

```

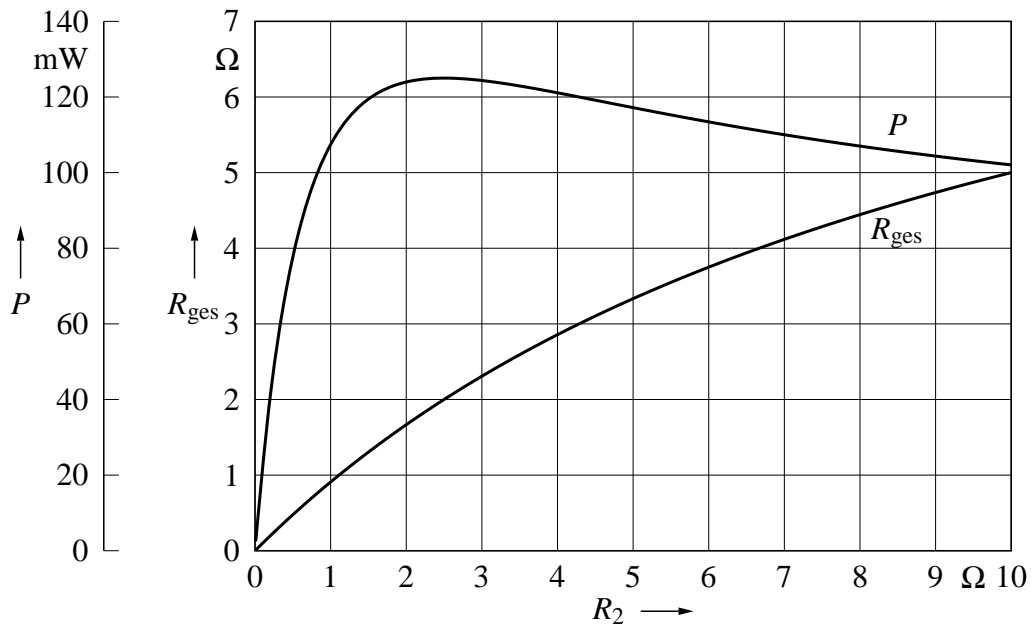


Abbildung 10: Zwei Kurven in einem Diagramm, Beispiel 1

Zu sehen ist, dass der Wertebereich sich bis ca. 0,125 W erstreckt, als Skalenendwert kann 0,14 W verwendet werden. Hierfür werden wir im fertigen Diagramm 7 cm vorsehen.

Listing 8: Zwei Größen in einem Diagramm, Beispiel 1

```

1 \newcounter{myvaraaaag}
2 \begin{figure}
3 {\centering
4 \begin{pscript}\psset{unit=1.0cm}
5 \begin{pspicture}(-3.5,-1.5)(10.2,7.2)
6 \special{ps:
7 /Uq 1.0 def /Ri 2.0 def /Rl 10.0 def
8 /Rges { dup Rl mul exch Rl add div } def
9 /P { Uq dup mul 1 index mul exch Ri add dup mul div } def
10 /ytocm { 50.0 mul } def
11 }
12 \psset{linewidth=0.1mm}\setcounter{myvaraaaag}{0}
13 \whiledo{\value{myvaraaaag}<11}{%
14 \psline(\value{myvaraaaag},0)(\value{myvaraaaag},7)%
15 \rput[t]{0}(\value{myvaraaaag},-0.2){\themyvaraaaag}%
16 \addtocounter{myvaraaaag}{1}%
17 }
18 \setcounter{myvaraaaag}{0}\whiledo{\value{myvaraaaag}<8}{%
19 \psline(0,\value{myvaraaaag})(10,\value{myvaraaaag})%
20 \rput[r]{0}(-0.2,\value{myvaraaaag}){\themyvaraaaag}%
21 \psline(-2.0,\value{myvaraaaag})(-1.8,\value{myvaraaaag})%
22 \addtocounter{myvaraaaag}{1}%
23 }
24 \rput[r]{0}(-2.2,0){\(\Omega\)}\rput[r]{0}(-2.2,1){\(\frac{1}{2}\Omega\)}
25 \rput[r]{0}(-2.2,2){\(\frac{1}{3}\Omega\)}\rput[r]{0}(-2.2,3){\(\frac{2}{3}\Omega\)}
26 \rput[r]{0}(-2.2,4){\(\frac{1}{4}\Omega\)}\rput[r]{0}(-2.2,5){\(\frac{3}{4}\Omega\)}
27 \rput[r]{0}(-2.2,6){\(\frac{1}{5}\Omega\)}\rput[r]{0}(-2.2,7){\(\frac{4}{5}\Omega\)}
28 \psset{linewidth=0.2mm,arrowsize=1.0mm 0.0,arrowlength=2.0,%
29 arrowinset=0.0}
30 \pspolygon(0,0)(10,0)(10,7)(0,7)\psline(-2.0,0)(-2.0,7)
31 \psline[arrows=->](5.1,-0.8)(5.8,-0.8)
32 \psline[arrows=->](-0.8,3.6)(-0.8,4.3)
33 \psline[arrows=->](-3.1,3.6)(-3.1,4.3)
34 \rput[r]{0}(4.9,-0.8){\(\frac{1}{2}R\)}\rput[t]{0}(9.5,-0.2){\(\frac{1}{2}\Omega\)}
35 \rput[t]{0}(-0.8,3.4){\(\frac{1}{2}R_{ges}\)}
36 \rput[r]{0}(-0.2,6.5){\(\frac{1}{2}\Omega\)}
37 \rput[t]{0}(-3.1,3.4){\(\frac{1}{2}P\)}\rput[r]{0}(-2.2,6.5){\(\frac{1}{2}mW\)}
38 \psset{linewidth=0.4mm}
39 \psplot[plotstyle=curve]{0.01}{10.0}{x Rges}
40 \psplot[plotstyle=curve]{0.01}{10.0}{x Rges P ytocm}
41 \rput[t]{0}(8.5,4.4){\(\frac{1}{2}R_{ges}\)}
42 \rput[b]{0}(8.5,5.5){\(\frac{1}{2}P\)}
43 \end{pspicture}\end{pscript}
44 \caption{Zwei Kurven in einem Diagramm, Beispiel 1}
45 \label{fig:0009}
46 }
47 \end{figure}

```

Im PostScript-Special werden jetzt weitere Konstanten definiert (Uq und Ri) sowie weitere Funktionen (P und yto cm). Die bisher verwendete Funktion ParallelWiderstand wurde umbenannt zu Rges, damit das Listing auf die Druckseite passt.

Die bisherigen Diagramme verwendeten jeweils $\frac{1\Omega}{1\text{cm}}$. Unsere zweite y-Achse soll eine Auflösung von $\frac{20\text{mW}}{1\text{cm}}$ haben. Bei der Berechnung von Leistungswerten zum Plotten (Zeile 40) wird folgendermaßen vorgegangen:

- Zunächst wird der x -Wert (Zahlenwert von R_2) auf den Stack gelegt.
- Die Funktion „Rges“ berechnet daraus den Gesamtwiderstand R_{ges} der Parallelschaltung.
- Die Funktion „P“ berechnet die an diesem Widerstand umgesetzte Verlustleistung.
- Die Funktion „yto cm“ rechnet dies in Zentimeter um.
Wenn Zahlenwerte nicht direkt als Zentimeterangaben genutzt werden können, gibt es zwei Möglichkeiten:
 - Mit dem `\psset`-Befehl wird die Einheitslänge (`xunit` bzw. `yunit`) entsprechend geändert.
 - Die Koordinaten werden in Zentimeter umgerechnet, dies kann manuell erfolgen oder wie in unserem Beispiel durch eine entsprechende PostScript-Funktion.

Um die zweite Skale zu erzeugen, werden die Skalenstriche in der `whiledo`-Schleife mit gezeichnet, die auch die Gitternetzlinien zeichnet (Zeile 21). Der senkrechte Strich wird in Zeile 30 gezeichnet. Die Skalenbeschriftung wird in den Zeilen 24 bis 27 erzeugt. Der Pfeil für die zweite Skalenteilung und dessen Beschriftung werden in den Zeilen 33 und 37 erzeugt.

Da das Bild sich jetzt weiter nach links erstreckt, müssen die Koordinaten des linken unteren Punktes beim Öffnen der `pspicture`-Umgebung angepasst werden.

Um die Kurven unterscheiden zu können, wird eine Beschriftung angebracht. Dies erfolgt in den Zeilen 41 und 42.

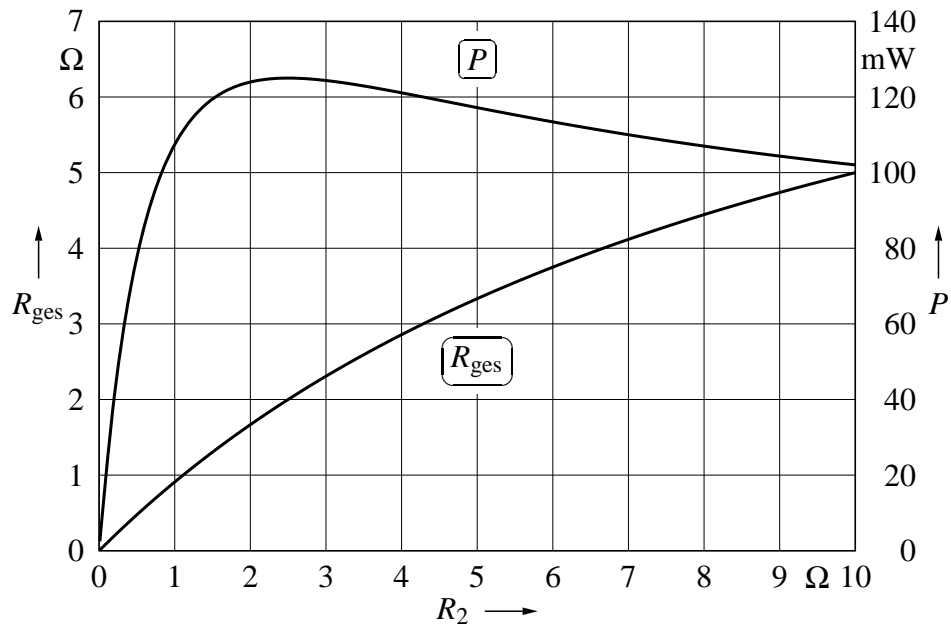


Abbildung 11: Zwei Kurven in einem Diagramm, Beispiel 2

Bei Verwendung mehrerer Skalen sind diese möglichst übersichtlich anzuordnen. Im Beispiel werden 2 Skalen mit gleichen Abständen zwischen den Teilstrichen benutzt. Eine Skale wurde links, die andere rechts platziert. Für die Boxen um die Kurvenbeschriftungen wird das \LaTeX -Paket „fancybox“ verwendet, diese wird mit

```
1 \usepackage{fancybox}
```

eingebunden.

Listing 9: Zwei Größen in einem Diagramm, Beispiel 2

```

1 \newcounter{myvaraaaah}\newcounter{myvaraaaai}
2 \begin{figure}
3 {\centering
4 \begin{postscript}\psset{unit=1.0cm}
5 \begin{pspicture}(-1.5,-1.5)(12.2,7.2)
6 \special{ps:
7 /Uq 1.0 def /Ri 2.0 def /Rl 10.0 def
8 /Rges { dup Rl mul exch Ri add div } def
9 /P { Uq dup mul 1 index mul exch Ri add dup mul div } def
10 /ytocm { 50.0 mul } def
11 }
12 \psset{linewidth=0.1mm}\setcounter{myvaraaaah}{0}
13 \whiledo{\value{myvaraaaah}<11}{%
14 \psline(\value{myvaraaaah},0)(\value{myvaraaaah},7)%
15 \rput[t]{0}(\value{myvaraaaah},-0.2){\themyvaraaaah}%
16 \addtocounter{myvaraaaah}{1}%
17 }
18 \setcounter{myvaraaaah}{0}\setcounter{myvaraaaai}{0}%
19 \whiledo{\value{myvaraaaah}<8}{%
20 \psline(0,\value{myvaraaaah})(10,\value{myvaraaaah})%
21 \rput[r]{0}(-0.2,\value{myvaraaaah}){\themyvaraaaah}%
22 \rput[r]{0}(10.8,\value{myvaraaaah}){\themyvaraaaai}%
23 \addtocounter{myvaraaaah}{1}\addtocounter{myvaraaaai}{20}%
24 }
25 \psset{linewidth=0.2mm,arrowsize=1.0mm 0.0,arrowlength=2.0,%
26 arrowinset=0.0}
27 \pspolygon(0,0)(10,0)(10,7)(0,7)
28 \psline[arrows=->](5.1,-0.8)(5.8,-0.8)
29 \psline[arrows=->](-0.8,3.6)(-0.8,4.3)
30 \psline[arrows=->](11.1,3.6)(11.1,4.3)
31 \rput[r]{0}(4.9,-0.8){\(\text{R}_2\)}\rput[t]{0}(9.5,-0.2){\(\Omega\)}
32 \rput[t]{0}(-0.8,3.4){\(\text{R}_\{\text{ges}\)\}}
33 \rput[r]{0}(-0.2,6.5){\(\Omega\)}
34 \rput[t]{0}(11.1,3.4){\(\text{P}\)}\rput[r]{0}(10.8,6.5){\(\text{mW}\)}
35 \psset{linewidth=0.4mm}
36 \psplot[plotstyle=curve]{0.01}{10.0}{x Rges}
37 \psplot[plotstyle=curve]{0.01}{10.0}{x Rges P yto cm}
38 % \rput[t]{0}(8.5,4.4){\(\text{R}_\{\text{ges}\)\}}
39 \rput{0}(5,2.5){\colorbox{white}{\ovalbox{\(\text{R}_\{\text{ges}\)\}}}}
40 % \rput[b]{0}(8.5,5.5){\(\text{P}\)}
41 \rput{0}(5,6.5){\colorbox{white}{\ovalbox{\(\text{P}\)}}}
42 \end{pspicture}\end{postscript}
43 \caption{Zwei Kurven in einem Diagramm, Beispiel 2}
44 \label{fig:0010}
45 }
46 \end{figure}

```

7. Daten plotten

7.1. Daten

Im Beispiel sollen Messdaten geplottet werden, die aus einer U - I -Messung an einem Widerstand stammen:

Tabelle 1: Beispiel-Messwerte

Spannung in V	Strom in A
1.08	0.000227
2.12	0.000455
3.12	0.000664
4.13	0.000907
5.19	0.001121
6.52	0.001337
7.27	0.001544
8.09	0.001857
9.78	0.002104

Diese Daten werden in eine Datei mw.dat geschrieben.

```
1 1.08,0.000227
2 2.12,0.000455
3 3.12,0.000664
4 4.13,0.000907
5 5.19,0.001121
6 6.52,0.001337
7 7.27,0.001544
8 8.09,0.001857
9 9.78,0.002104
```

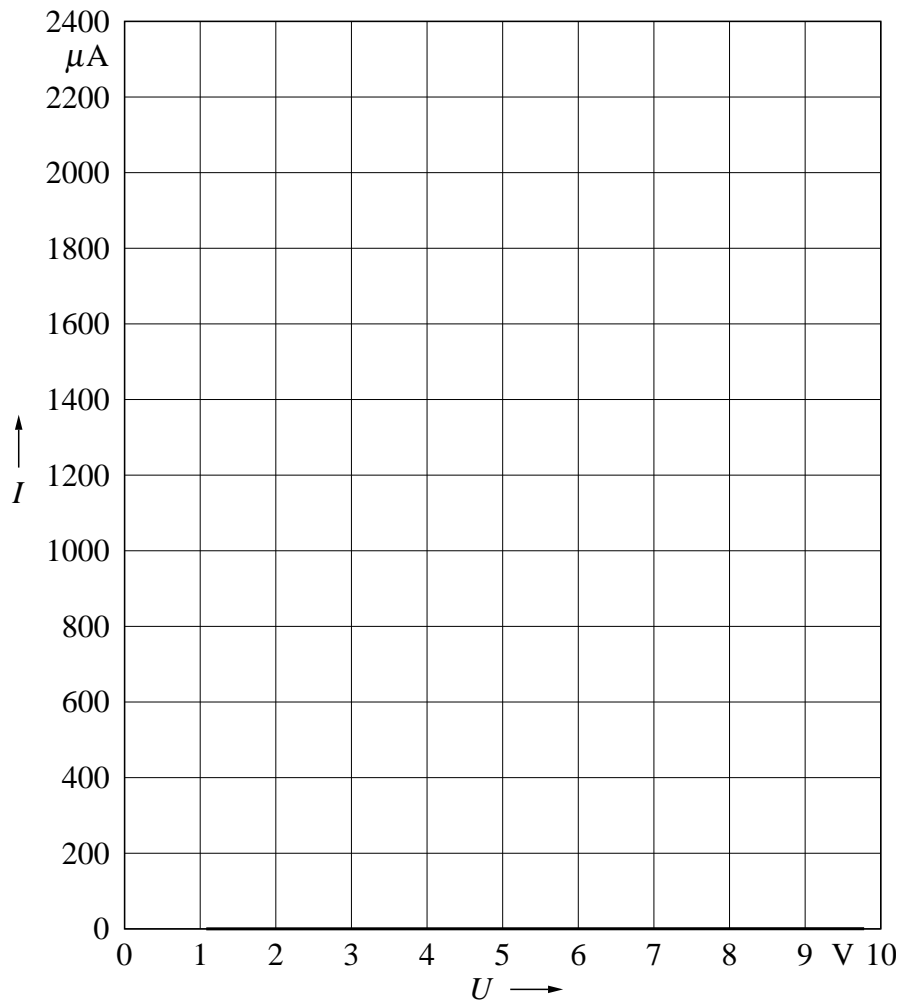


Abbildung 12: Messdaten aus U - I -Messung

7.2. Erster Plot

Werden die Daten entsprechend Listing 10 auf der nächsten Seite geplottet, ergibt sich wegen der kleinen y -Werte eine Linie, die auf der Nulllinie des Diagrammes liegt.

Listing 10: Daten plotten, erster Versuch

```

1 \newcounter{myvaraaaaj}\newcounter{myvaraaaak}%
2 \begin{figure}%
3 {\centering%
4 \begin{postscript}
5 \begin{pspicture}(-1.8,-1.5)(10.2,12.2)
6 \psset{linewidth=0.1mm}
7 \setcounter{myvaraaaaj}{0}\whiledo{\value{myvaraaaaj}<11}{
8 \psline(\value{myvaraaaaj},0)(\value{myvaraaaaj},12)
9 \rput[t]{0}(\value{myvaraaaaj},-0.2){\themyvaraaaaj}
10 \addtocounter{myvaraaaaj}{1}%
11 }
12 \setcounter{myvaraaaaj}{0}\setcounter{myvaraaaak}{0}
13 \whiledo{\value{myvaraaaaj}<13}{
14 \psline(0,\value{myvaraaaaj})(10,\value{myvaraaaaj})
15 \rput[r]{0}(-0.2,\value{myvaraaaaj}){\themyvaraaaak}
16 \addtocounter{myvaraaaaj}{1}%
17 \addtocounter{myvaraaaak}{200}%
18 }
19 \psset{%
20 linewidth=0.2mm,%
21 arrowsize=1.0mm 0.0,%
22 arrowlength=2.0,%
23 arrowinset=0.0%
24 }
25 \pspolygon(0,0)(10,0)(10,12)(0,12)
26 \psline[arrows=->](5.1,-0.8)(5.8,-0.8)
27 \psline[arrows=->](-1.4,6.1)(-1.4,6.8)
28 \rput[r]{0}(4.9,-0.8){\text{U}}\rput[t]{0}(9.5,-0.2){\text{V}}
29 \rput[t]{0}(-1.4,5.9){\text{I}}\rput[r]{0}(-0.2,11.5){\mu\text{A}}
30 \psset{linewidth=0.4mm}
31 \fileplot{mw.dat}
32 \end{pspicture}
33 \end{postscript}%
34 \caption{Messdaten aus \text{U} - \text{I} - Messung}\label{fig:fig0011}%
35 }%
36 \end{figure}%

```

7.3. Skalierung

Listing 11: Daten plotten, zweiter Versuch

```
1 \newcounter{myvaraaaal}\newcounter{myvaraaaam}%
2 \begin{figure}%
3 {\centering%
4 \begin{postscript}
5 \begin{pspicture}(-1.8,-1.5)(10.2,12.2)
6 \psset{linewidth=0.1mm}
7 \setcounter{myvaraaaal}{0}\whiledo{\value{myvaraaaal}<11}{
8 \psline(\value{myvaraaaal},0)(\value{myvaraaaal},12)
9 \rput[t]{0}(\value{myvaraaaal},-0.2){\themyvaraaaal}
10 \addtocounter{myvaraaaal}{1}%
11 }
12 \setcounter{myvaraaaal}{0}\setcounter{myvaraaaam}{0}
13 \whiledo{\value{myvaraaaal}<13}{
14 \psline(0,\value{myvaraaaal})(10,\value{myvaraaaal})
15 \rput[r]{0}(-0.2,\value{myvaraaaal}){\themyvaraaaam}
16 \addtocounter{myvaraaaal}{1}%
17 \addtocounter{myvaraaaam}{200}%
18 }
19 \psset{%
20 linewidth=0.2mm,%
21 arrowsize=1.0mm 0.0,%
22 arrowlength=2.0,%
23 arrowinset=0.0%
24 }
25 \pspolygon(0,0)(10,0)(10,12)(0,12)
26 \psline[arrows=->](5.1,-0.8)(5.8,-0.8)
27 \psline[arrows=->](-1.4,6.1)(-1.4,6.8)
28 \rput[r]{0}(4.9,-0.8){\text{(U)}}\rput[t]{0}(9.5,-0.2){\text{(V)}}
29 \rput[t]{0}(-1.4,5.9){\text{(I)}}\rput[r]{0}(-0.2,11.5){\text{(\mu A)}}
30 \psset{linewidth=0.4mm}
31 \psset{yunit=5000.0cm}
32 \fileplot{mw.dat}
33 \psset{yunit=1.0cm}
34 \end{pspicture}
35 \end{postscript}%
36 \caption{Messdaten aus \text{(U)}-\text{(I)}-Messung}\label{fig:fig0012}%
37 }%
38 \end{figure}%
```

Versuche, mit

```
1 \psset{yunit=5000.0cm}
```

das Koordinatensystem zu skalieren schlagen fehl mit der Fehlermeldung, dass eine zu große Dimension angegeben wurde.

Listing 12: Konvertierungs-Script

```
1 my $newx; my $newy;  
2 while(<>) {  
3   $line = $_; chomp $line;  
4   if($line =~ /^\\s*(\\S+)\\s*,\\s*(\\S+)\\s*$/o) {  
5     $newx = $1; $newy = 5000.0 * $2;  
6     print "$newx,$newy\\n";  
7   }  
8 }
```

Mit einem Perl-Script entsprechend Listing 12 wird mit

```
1 perl co-mw.pl < mw.dat > mw-cor.dat
```

eine neue Messwertedatei „mw-cor.dat“ angelegt. Das Perl-Script multipliziert jeden y-Wert aus der Eingabe mit 5000 und gibt das neue Wertepaar aus. Die Ausgabedatei „mw-cor.dat“ enthält dann die Koordinaten in cm und kann direkt benutzt werden (siehe Listing 13).

Zum Plotten von Daten wird der *fileplot*-Befehl verwendet, siehe Zeile 26. Standardmäßig verbindet dieser Befehl die angegebenen Punkte durch Linienzüge.

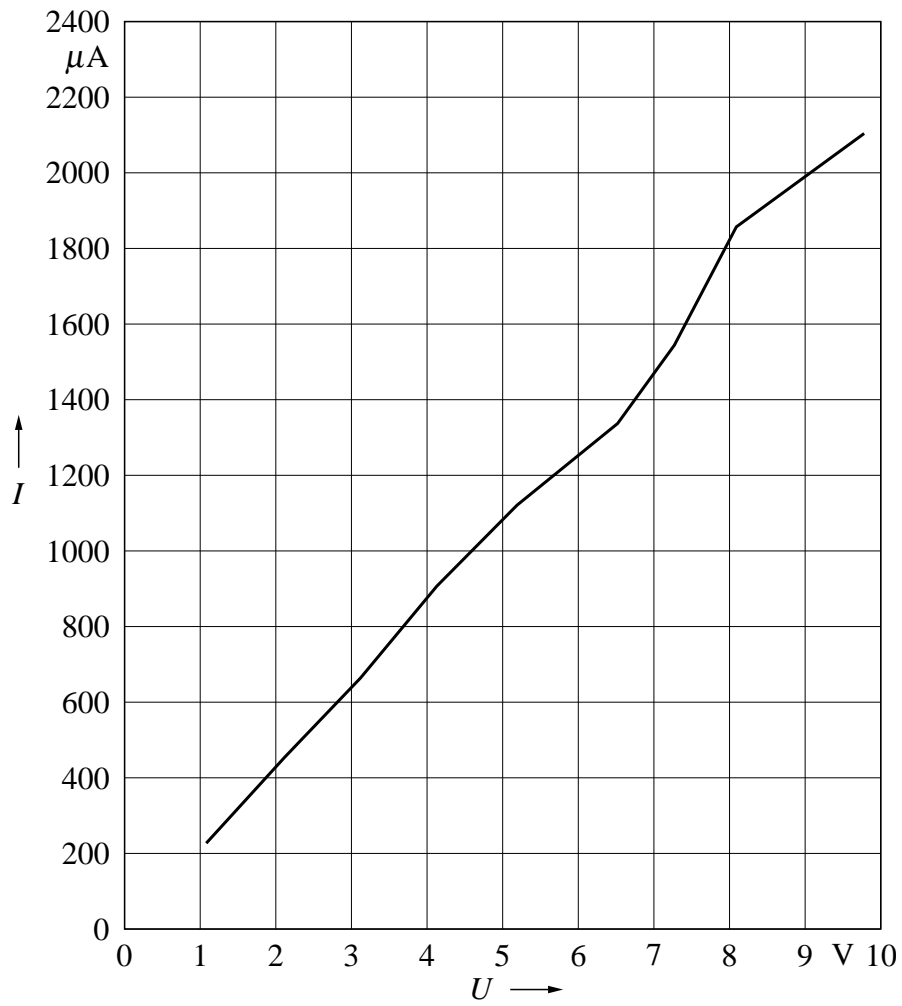


Abbildung 13: Messdaten aus U - I -Messung

Listing 13: Daten plotten, dritter Versuch

```

1 \newcounter{myvaraaaaan}\newcounter{myvaraaaao}%
2 \begin{figure}%
3 {\centering%
4 \begin{postscript}
5 \begin{pspicture}(-1.8,-1.5)(10.2,12.2)
6 \psset{linewidth=0.1mm}
7 \setcounter{myvaraaaaan}{0}\whiledo{\value{myvaraaaaan}<11}{
8 \psline(\value{myvaraaaaan},0)(\value{myvaraaaaan},12)
9 \rput[t]{0}(\value{myvaraaaaan},-0.2){\themyvaraaaaan}
10 \addtocounter{myvaraaaaan}{1}%
11 }
12 \setcounter{myvaraaaaan}{0}\setcounter{myvaraaaao}{0}
13 \whiledo{\value{myvaraaaaan}<13}{
14 \psline(0,\value{myvaraaaaan})(10,\value{myvaraaaaan})
15 \rput[r]{0}(-0.2,\value{myvaraaaaan}){\themyvaraaaao}
16 \addtocounter{myvaraaaaan}{1}%
17 \addtocounter{myvaraaaao}{200}%
18 }
19 \psset{%
20 linewidth=0.2mm,%
21 arrowsize=1.0mm 0.0,%
22 arrowlength=2.0,%
23 arrowinset=0.0%
24 }
25 \pspolygon(0,0)(10,0)(10,12)(0,12)
26 \psline[arrows=->](5.1,-0.8)(5.8,-0.8)
27 \psline[arrows=->](-1.4,6.1)(-1.4,6.8)
28 \rput[r]{0}(4.9,-0.8){\text{U}}\rput[t]{0}(9.5,-0.2){\text{V}}
29 \rput[t]{0}(-1.4,5.9){\text{I}}\rput[r]{0}(-0.2,11.5){\mu\text{A}}
30 \psset{linewidth=0.4mm}
31 \fileplot{mw-cor.dat}
32 \end{pspicture}
33 \end{postscript}%
34 \caption{Messdaten aus \text{U} - \text{I} - Messung}\label{fig:fig0013}%
35 }%
36 \end{figure}%

```

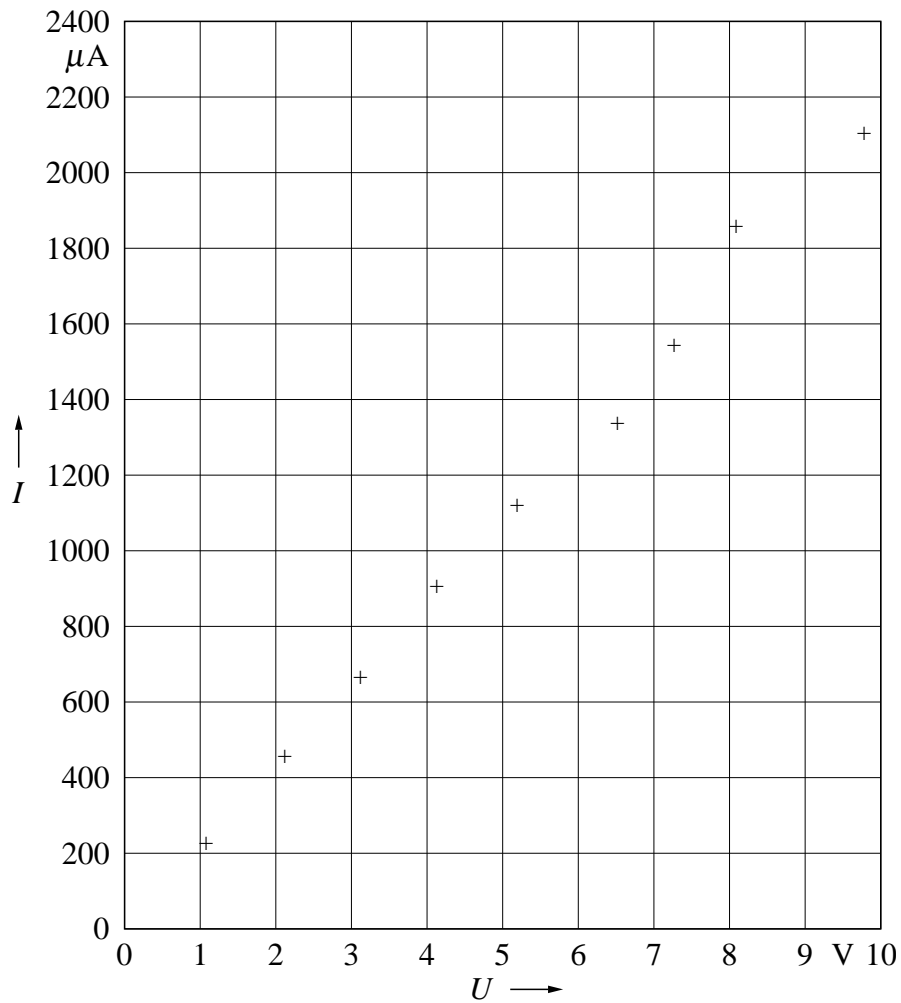


Abbildung 14: Messdaten aus U - I -Messung

7.4. Messwerte als Punkte plotten

Sollen keine Linienzüge gezeichnet werden sondern die Messpunkte, kann dies über den Parameter *plotstyle* eingestellt werden. Mit dem optionalen Parameter „dotstyle=+“ wird die Form der Messpunkte vom standardmäßig verwendeten Kreis auf ein Kreuz geändert.

Listing 14: Daten plotten, Messpunkte einzeichnen

```

1 \newcounter{myvaraaaap}\newcounter{myvaraaaq}%
2 \begin{figure}%
3 {\centering%
4 \begin{postscript}
5 \begin{pspicture}(-1.8,-1.5)(10.2,12.2)
6 \psset{linewidth=0.1mm}
7 \setcounter{myvaraaaap}{0}\whiledo{\value{myvaraaaap}<11}{
8 \psline(\value{myvaraaaap},0)(\value{myvaraaaap},12)
9 \rput[t]{0}(\value{myvaraaaap},-0.2){\themyvaraaaap}
10 \addtocounter{myvaraaaap}{1}%
11 }
12 \setcounter{myvaraaaap}{0}\setcounter{myvaraaaq}{0}
13 \whiledo{\value{myvaraaaap}<13}{
14 \psline(0,\value{myvaraaaap})(10,\value{myvaraaaap})
15 \rput[r]{0}(-0.2,\value{myvaraaaap}){\themyvaraaaq}
16 \addtocounter{myvaraaaap}{1}%
17 \addtocounter{myvaraaaq}{200}%
18 }
19 \psset{%
20 linewidth=0.2mm,%
21 arrowsize=1.0mm 0.0,%
22 arrowlength=2.0,%
23 arrowinset=0.0%
24 }
25 \pspolygon(0,0)(10,0)(10,12)(0,12)
26 \psline[arrows=->](5.1,-0.8)(5.8,-0.8)
27 \psline[arrows=->](-1.4,6.1)(-1.4,6.8)
28 \rput[r]{0}(4.9,-0.8){\text{U}}\rput[t]{0}(9.5,-0.2){\text{V}}
29 \rput[t]{0}(-1.4,5.9){\text{I}}\rput[r]{0}(-0.2,11.5){\mu\text{A}}
30 \psset{linewidth=0.4mm}
31 \fileplot[plotstyle=dots,dotstyle=+]{mw-cor.dat}
32 \end{pspicture}
33 \end{postscript}%
34 \caption{Messdaten aus \text{U} - \text{I} - Messung}\label{fig:fig0014}%
35 }%
36 \end{figure}%

```

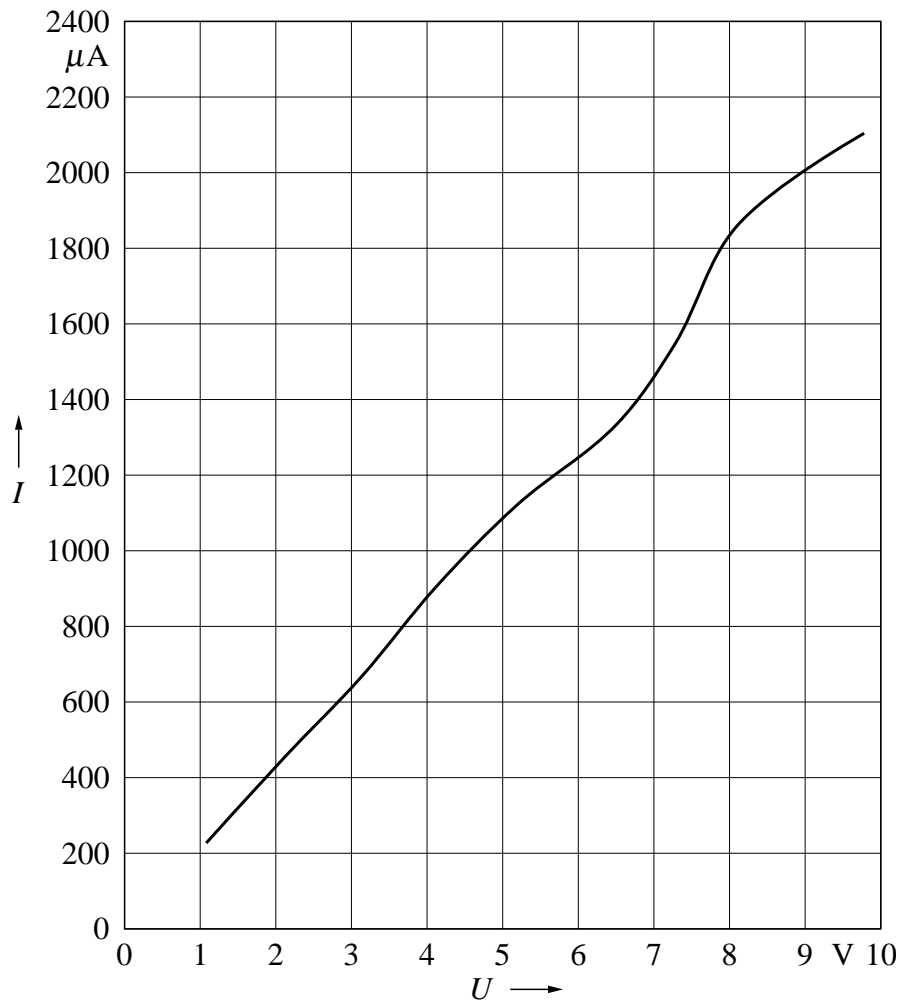


Abbildung 15: Messdaten aus U - I -Messung

7.5. Interpolationskurve erzeugen

Sollen Messdaten mehrfach verwendet werden, können sie mit `\readdata` eingelesen werden. Der erste Parameter ist ein \LaTeX -Identifier (der Name unter dem die Daten gespeichert werden), der zweite Parameter ist der Dateiname der einzulesenden Datei. Das Plotten kann dann mit `\dataplot` erfolgen.

Mit dem Parameter „`plotstyle=curve`“ kann eine Kurve in die Messpunkte hineininterpoliert werden.

Listing 15: Daten plotten, interpolierte Kurve

```

1 \newcounter{myvaraaaar}\newcounter{myvaraaaas}%
2 \begin{figure}%
3 {\centering%
4 \begin{pspicture}
5 \begin{pspicture}(-1.8,-1.5)(10.2,12.2)
6 \psset{linewidth=0.1mm}
7 \setcounter{myvaraaaar}{0}\whiledo{\value{myvaraaaar}<11}{
8 \psline(\value{myvaraaaar},0)(\value{myvaraaaar},12)
9 \rput[t]{0}(\value{myvaraaaar},-0.2){\themyvaraaaar}
10 \addtocounter{myvaraaaar}{1}%
11 }
12 \setcounter{myvaraaaar}{0}\setcounter{myvaraaaas}{0}
13 \whiledo{\value{myvaraaaar}<13}{
14 \psline(0,\value{myvaraaaar})(10,\value{myvaraaaar})
15 \rput[r]{0}(-0.2,\value{myvaraaaar}){\themyvaraaaas}
16 \addtocounter{myvaraaaar}{1}%
17 \addtocounter{myvaraaaas}{200}%
18 }
19 \psset{%
20 linewidth=0.2mm,%
21 arrowsize=1.0mm 0.0,%
22 arrowlength=2.0,%
23 arrowinset=0.0%
24 }
25 \pspolygon(0,0)(10,0)(10,12)(0,12)
26 \psline[arrows=->](5.1,-0.8)(5.8,-0.8)
27 \psline[arrows=->](-1.4,6.1)(-1.4,6.8)
28 \rput[r]{0}(4.9,-0.8){\text{U}}\rput[t]{0}(9.5,-0.2){\text{V}}
29 \rput[t]{0}(-1.4,5.9){\text{I}}\rput[r]{0}(-0.2,11.5){\mu\text{A}}
30 \psset{linewidth=0.4mm}
31 \readdata{\messwerte}{mw-cor.dat}
32 \dataplot[plotstyle=curve]{\messwerte}
33 \end{pspicture}
34 \end{pspicture}%
35 \caption{Messdaten aus \text{U}-\text{I}-Messung}\label{fig:fig0015}%
36 }%
37 \end{figure}%

```

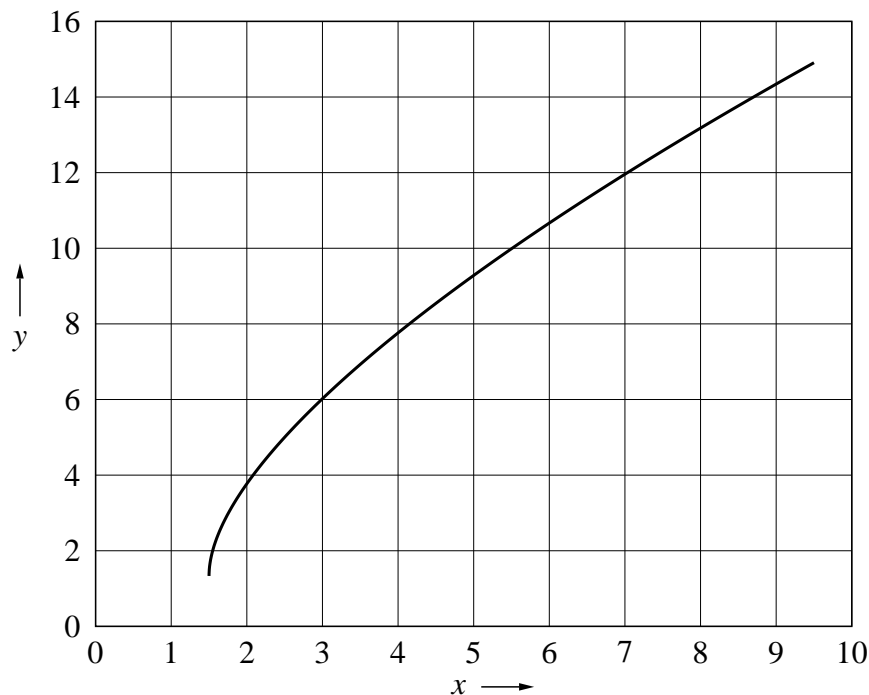


Abbildung 16: Parametrischer Plot

8. Parametrische Plots

Für parametrische Plots durchläuft ein Parameter t ein vorgegebenes Intervall. Für jeden t -Wert kann eindeutig ein Punkt $P(x(t), y(t))$ berechnet werden.

Die Anweisung `\parametricplot` (siehe Zeile 31) dient zum parametrischen Plotten. Die ersten beiden erforderlichen Parameter sind der Startwert und der Endwert für t . Der dritte Parameter enthält PostScript-Code, der den x - und den y -Wert berechnet und auf den Stack legt. Die PostScript-Anweisung „ t “ legt den jeweils aktuellen t -Wert auf den Stack.

Im Beispiel wird die Kurve

$$\tilde{k}: \quad x = \frac{1}{2}t^2 - t + 2 \quad y = \frac{4}{3}\sqrt{t^3} \quad 1 \leq t \leq 5$$

gezeichnet.

Für die Berechnung der x - und y -Werte werden die PostScript-Funktionen „`xfct`“ und „`yfct`“ verwendet. Der y -Wert wird dann mit der Funktion „`ytocm`“ skaliert.

Listing 16: Parametrischer Plot

```

1 \newcounter{myvaraaaat}\newcounter{myvaraaaau}%
2 \begin{figure}%
3 {\centering%
4 \begin{postscript}
5 \begin{pspicture}(-1.8,-1.5)(10.2,8.2)
6 \special{ps:
7 /xfct { dup dup mul 0.5 mul exch sub 2 add } def
8 /yfct { dup dup mul mul sqrt 4 mul 3 div } def
9 /ytocm { 0.5 mul } def
10 }
11 \psset{linewidth=0.1mm}
12 \setcounter{myvaraaaat}{0}\whiledo{\value{myvaraaaat}<11}{
13 \psline(\value{myvaraaaat},0)(\value{myvaraaaat},8)
14 \rput[t]{0}(\value{myvaraaaat},-0.2){\themyvaraaaat}
15 \addtocounter{myvaraaaat}{1}%
16 }
17 \setcounter{myvaraaaat}{0}\setcounter{myvaraaaau}{0}
18 \whiledo{\value{myvaraaaat}<9}{
19 \psline(0,\value{myvaraaaat})(10,\value{myvaraaaat})
20 \rput[r]{0}(-0.2,\value{myvaraaaat}){\themyvaraaaau}
21 \addtocounter{myvaraaaat}{1}%
22 \addtocounter{myvaraaaau}{2}%
23 }
24 \psset{%
25 linewidth=0.2mm,%
26 arrowsize=1.0mm 0.0,%
27 arrowlength=2.0,%
28 arrowinset=0.0%
29 }
30 \pspolygon(0,0)(10,0)(10,8)(0,8)
31 \psline[arrows=->](5.1,-0.8)(5.8,-0.8)
32 \psline[arrows=->](-1,4.1)(-1,4.8)
33 \rput[r]{0}(4.9,-0.8){\!(x\)}
34 \rput[t]{0}(-1,3.9){\!(y\)}
35 \psset{linewidth=0.4mm}
36 \parametricplot[plotstyle=curve]{1.0}{5.0}{t xfct t yfct yto cm}
37 \end{pspicture}
38 \end{postscript}%
39 \caption{Parametrischer Plot}\label{fig:fig0016}%
40 }%
41 \end{figure}%

```

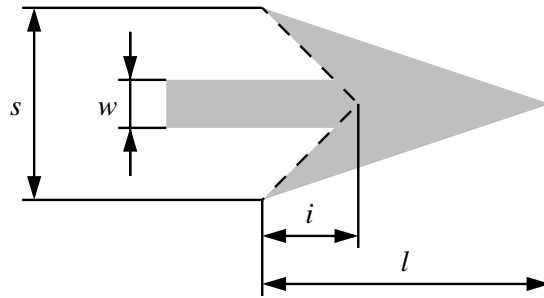


Abbildung 17: Form von Pfeilen

A. Pfeile an Linienenden

Die Form von Pfeilen wird über folgende PSTricks-Parameter eingestellt:

- **arrowsize=dim num**
Dabei ist *dim* eine Längenangabe mit Einheit, *num* eine Zahlenangabe ohne Einheit.
- **arrowlength=num**
- **arrowinset=num**

Ausgehend von der Linienweite w werden die Abmessungen für die Pfeilform folgendermaßen berechnet:

Pfeilbreite	$s = \text{arrowsize.dim} + \text{arrowsize.num} \cdot w$
Pfeillänge	$l = \text{arrowlength} \cdot s$
Einrückung	$i = \text{arrowinset} \cdot l$

Literatur

- [ASI99] Adobe Systems Incorporated. *PostScript language reference third edition*. Adobe Systems Incorporated, 1999. 4
- [IK95] Ingo Klöckl. *PostScript Einstieg - Workshop - Referenz*. Hanser Fachbuchverlag, 1995. 4
- [TVZ06] Timothy Van Zandt. *PSTricks: PostScript macros for Generic TeX. User's Guide*. 1999-2006. 4