Die TEXnische Komödie

DANTE
Deutschsprachige
Anwendervereinigung TEX e.V.

Uwe Siart: *Ansprechende technische Illustration mit METAPOST*, Die TEXnische Komödie 1/2002, S. 26–39.

Ansprechende technische Illustration mit METAPOST

Uwe Siart

METAPOST ist eine grafische Beschreibungssprache zur Erstellung qualitativ hochstehender Vektorzeichnungen. Obwohl der Compiler in praktisch allen modernen TEX-Distributionen enthalten ist, findet METAPOST noch relativ wenig Beachtung. METAPOST ist besonders geeignet zur Erstellung von Strichzeichnungen wie sie in Naturwissenschaft und Technik häufig auftreten. Dabei erzeugt es besonders schlanke Bilddateien, deren Format praktisch identisch zu EPS ist. Allerdings werden Ressourcen, die das TEX-Dokument ohnehin einbindet, einfach weggelassen. Die angeführten Beispiele stammen – wie auch der Autor – aus dem ingenieurwissenschaftlich-technischen Bereich. Sie sollen einige Fähigkeiten von METAPOST aufzeigen und den Leser ermutigen, seine eigenen Anwendungsgebiete für METAPOST zu suchen.

Einführung

Das Erstellen von naturwissenschaftlichen oder technischen Strichzeichnungen ist durch Eingabe mit einem Zeigegerät zwar möglich, erfordert aber meistens sehr viel Geduld. Trotz beachtlichen Zeitaufwands ist das grafische Ergebnis nicht selten von nur mäßiger Qualität. Es treten Positionierungsaufgaben auf, die mit Mauseingabe nur durch Augenmaß gelöst werden können. Änderungen oder Zuweisungen, die auf diese Weise gemacht wurden, sind meist nicht

reproduzierbar. Und benötigt man eine Serie, deren Bilder sich eigentlich nur in einem Parameter unterscheiden, so bedeutet dies nicht selten die mehrfache Zeichenarbeit. Schließlich sind die erzeugten EPS-Dateien häufig durch ungenutzte PostScript-Ressourcen aufgebläht, was bei der PostScript-Datei eines Buches mit mehreren hundert Abbildungen durchaus ins Gewicht fällt. Den Benutzern von pdf(LA)TEX wird angenehm auffallen, dass sie die Bilddateien von METAPOST *nicht* in das PDF-Format umwandeln müssen. Sie können von pdf(LA)TEX direkt konvertiert werden, sodass hier ein Arbeitsschritt entfallen kann.

Die Arbeitsweise von METAPOST ist identisch mit derjenigen von TEX. Das Programm liest eine Eingabedatei mit Anweisungen, die das Bild beschreiben, und legt nach erfolgreichem Lauf eine oder mehrere Bilddateien im Arbeitsverzeichnis ab. Der Name, unter dem METAPOST aufgerufen wird, hängt von der jeweiligen TEX-Distribution ab. In MiKTEX lautet er beispielsweise

```
mp dateiname.mp
```

wobei die Endung `.mp` beim Aufruf weggelassen werden kann. Neben `mp` lautet das Kommando auch häufig `mpost`. In den meisten Fällen dürfte eine dieser beiden Varianten zutreffen. Innerhalb der Eingabedatei befinden sich die Befehle für ein Bild zwischen den Anweisungen

```
beginfig(1);
   ...
endfig;
```

Dieses Anweisungspaar zusammen mit einem Zeichenbefehl ist daher auch der Mindestinhalt einer METAPOST-Datei, um ein Bild zu erzeugen. Die Eingabedatei darf aber auch die Anweisungen für mehr als ein Bild enthalten. Die umschließenden Anweisungen `beginfig(n)` und `endfig` dürfen deshalb auch mehrfach auftreten, eben einmal je Bild. Nach erfolgreichem Durchlauf hat METAPOST Dateien mit den Bezeichnungen `dateiname.1`, `dateiname.2`, usw. erzeugt, wobei die Nummer im Argument von `beginfig` die Endung der zugehörigen Ausgabedatei bestimmt. Dieses sind die Bilddateien, die direkt in LATEX-Dokumente eingebunden werden können. Beim Erzeugen von Beschriftungen ist in METAPOST jedes beliebige TEX-Konstrukt zulässig. Wenn METAPOST zum Setzen von Beschriftungen anstatt TEX lieber LATEX aufrufen soll, so kann dieses entweder mit dem Aufruf

```
mp --tex=latex dateiname
```

oder durch Setzen der Umgebungsvariable `TEX` auf den Wert `latex` erreicht werden. An den Beginn der METAPOST-Eingabedatei ist dann ein gewöhnlicher LATEX-Vorspann in der Form

```
verbatimtex
\documentclass{...}
 eventuell weitere Pakete
\begin{document}
etex
```

zu setzen. Das sonst notwendige `\end{document}` entfällt, weil dieses von METAPOST automatisch erzeugt wird. Abhängig von der jeweiligen TEX-Distribution kann die Anweisung zum Aufruf von LATEX auch in den Vorspann der METAPOST-Datei geschrieben werden. Hierzu sollte sofort nach `verbatimtex` das Kommando `%&latex` eingefügt werden. Welche dieser Methoden funktioniert und welche nicht, kann man – falls nicht in der Dokumentation des eigenen TEX-Paketes angegeben – durch Probieren herausfinden.

Im Vorspann müssen wie in einem LATEX-Dokument alle Pakete geladen werden, auf deren Funktionen innerhalb der METAPOST-Quelldatei zurückgegriffen wird. Um Konflikte beim Laden von Schriften oder unpassende Schriftgrößen zu vermeiden, ist es sinnvoll, wenn die METAPOST-Datei denselben Vorspann aufweist, wie das Dokument, in das die Bilder eingefügt werden.

Ausgewählte Grundfunktionen

Punkte, Geraden, Polygone, Kurven

METAPOST kennt mehrere Variablentypen. Die für diese Kurzeinführung wichtigsten sind Koordinatenpaare (`pair`), Zahlen oder Längenangaben (`numeric`) und Pfade (`path`). Soll eine solche Variable verwendet werden, so muss sie vorher durch

`pair` ⟨*bezeichnung1*⟩, ⟨*bezeichnung2*⟩, ... ;
`numeric` ⟨*bezeichnung1*⟩, ⟨*bezeichnung2*⟩, ... ;
`path` ⟨*bezeichnung1*⟩, ⟨*bezeichnung2*⟩, ... ;

deklariert werden. Wird dabei der Name in der Form `name[]` deklariert, so können später numerische Indizes verwendet werden. Es sind dann alle Variablennamen der Form `name1`, `name2`, ... zulässig. Die Namen `x`, `y` und `z` sind vordefiniert und dürfen ohne vorherige Deklaration mit numerischen Indizes verwendet werden. Dabei ist `z` vom Typ `pair`, `x` und `y` vom Typ `numeric`. Wenn der Punkt `z1` bekannt ist, dann sind auch die Variablen `x1` und `y1` belegt, und zwar mit den Koordinaten des Punktes `z1`. Punkte können z. B. durch

`z1 = (`⟨*dim1*⟩`,`⟨*dim2*⟩`);`

festgelegt werden. Die Angabe von

`x1 =` ⟨*dim1*⟩`;  y1 =`⟨*dim2*⟩`;`

ist äquivalent.

Ausgehend von Punkten können Pfade definiert werden, die als Polygone oder geglättete Kurven die Punkte verbinden. Die Form von geglätteten Kurven kann in METAPOST auf vielfältige Weise beeinflusst werden. In den meisten Fällen wird METAPOST aber ohne weitere Parameter eine gut geformte Kurve erzeugen. Gehen wir davon aus, dass `pair`-Variablen mit den Namen `z[]` bereits belegt sind (entweder durch direkte Eingabe oder aus vorangegangenen Berechnungen), dann kann ein Polygonzug durch die Anweisung

⟨*Pfad*⟩ `= z0--z1--z2--z3;`

und eine geglättete Kurve durch die Anweisung

⟨*Pfad*⟩ `= z0..z1..z2..z3;`

definiert werden. Die Anweisungen `--` und `..` zum Verbinden zweier Punkte durch ein Geradenstück oder durch eine Kurve dürfen innerhalb einer solchen Pfaddefinition auch gemischt auftreten. In diesem Zusammenhang gibt es noch die Anweisung `---`, die bewirkt, dass beim Zusammentreffen mit einer `..`-Anweisung ein gerader und ein gebogener Pfadabschnitt glatt (also ohne Knick) ineinander übergehen.

Um den Verlauf einer von METAPOST erzeugten Kurve steuern zu können, gibt es zusätzliche Anweisungen, die man am besten [3] entnimmt, weil ihre Beschreibung hier zu weit führen würde. So gibt es etwa die Möglichkeit, Bézier-Punkte explizit anzugeben, die Krümmung und die Steigung der Kurve beim Erreichen oder Verlassen eines Stützpunktes festzulegen oder einen Spannungsparameter zu setzen.

Stifte

Zum Zeichnen von Pfaden mit den Befehlen `draw`, `drawarrow` usw. verwendet METAPOST Stifte, deren Spitzenform in Variablen vom Typ `pen` abgelegt werden kann. Durch die Anweisung `pickup` wird ein bestimmter Stift aktiviert und bis zu einer erneuten `pickup`-Anweisung verwendet. Ein Stift mit kreisrunder Spitze und Durchmesser 1 ist unter dem Namen `pencircle` vordefiniert. Durch Skalierung kann mit der Anweisung

`pickup pencircle scaled` ⟨*dim*⟩`;`

jede beliebige Strichstärke eingestellt werden. In LaTeX wird als Basisstrichstärke 0,4 pt verwendet. Es ist daher empfehlenswert, dieses in Zeichnungen beizubehalten. Für besondere Effekte ist die Definition eigener Stiftformen möglich. Das Vorgehen wird in [3] ausführlich beschrieben.

Durch den Zusatz `withpen` kann für einzelne Zeichenanweisungen ein individueller Stift verwendet werden. Wurde beispielsweise durch die Anweisung

```
pickup pencircle scaled 0.4pt;
```

eine globale Strichstärke gewählt, so kann etwa durch

```
draw ⟨Pfad⟩ withpen pencircle scaled 0.85pt;
```

ein einzelner Pfad mit größerer Strichstärke gezeichnet werden.

Richtungsangaben

Die Richtung, unter der ein Pfad einen Stützpunkt erreicht oder verlässt, kann durch Vor- oder Nachstellen einer `pair`-Variablen in geschweiften Klammern festgelegt werden. Die Richtung der Verbindung vom Koordinatenursprung zum Punkt ⟨*pair*⟩ entspricht dann der Tangentialrichtung des Pfades beim Eintreffen oder beim Verlassen des jeweiligen Stützpunktes. Besonders nützlich ist hierbei die `dir` ⟨*Winkel*⟩-Anweisung. Sie erzeugt ein `pair` mit den Koordinaten `(cos` ⟨*Winkel*⟩, `sin` ⟨*Winkel*⟩`)`. Die Anwendung innerhalb einer Pfaddefinition geschieht dann beispielsweise in der Form

```
⟨Pfad⟩ = z0{dir 30}..{dir -50}z1;
```

wodurch erzwungen wird, dass ⟨*Pfad*⟩ den Punkt `z0` unter 30° verlässt und im Punkt `z1` unter −50° ankommt. Zusätzlich sind die Angaben `{up}`, `{down}`, `{left}` und `{right}` vordefiniert, welche die gleiche Wirkung haben wie `{dir 90}`, `{dir -90}`, `{dir 180}` und `{dir 0}`.

Pfeile und Linientypen

Besonders angenehm fällt METAPOST durch seine Fähigkeit auf, schöne Pfeile auch an stark gekrümmte Kurven zu zeichnen. Die Pfeilspitzen sind nicht einfach Dreiecke, die auf der Linie in tangentialer Richtung orientiert sind ohne deren Verlauf zu beachten. METAPOST passt Pfeilspitzen in ihrer Form der lokalen Krümmung der Kurve an. Ohne diese Anpassung würden die Pfeilspitzen bereits bei mäßiger Krümmung von der Kurve „abreißen“, anstatt ihr zu folgen. Selbst modernste Grafikpakete weisen diese Unzulänglichkeit immer noch auf. Der Vergleich von Abb. 1(a) und (b) zeigt diese Fähigkeit von METAPOST.

Zum Zeichnen von Pfeilen dienen die Anweisungen

```
drawarrow     ⟨Pfad⟩;
drawdblarrow  ⟨Pfad⟩;
```

(a) METAPOST (b) modernes Grafikprogramm

Abbildung 1: Vergleich von Pfeilen verschiedener Zeichenprogramme

wobei im ersten Fall ⟨*Pfad*⟩ eine Spitze an seinem Ende erhält. Die zweite Anweisung zeichnet ⟨*Pfad*⟩ mit einer Pfeilspitze am Anfang und am Ende. Pfade in METAPOST haben einen Anfang und ein Ende und damit eine Richtung, die bei der Pfaddefinition festgelegt wird. Durch die Anweisung `reverse` wird allgemein die Richtung eines Pfades umgedreht. Die Auswahl des Pfadendes, welches die Pfeilspitze erhalten soll, kann deshalb durch Hinzufügen oder Weglassen der `reverse`-Anweisung erfolgen.

Verschiedene Linientypen werden innerhalb eines `draw`-Befehls durch die Zusatzanweisung `dashed` mit nachfolgender Angabe eines Strichmusters ausgewählt. Als vordefinierte Muster stehen bereits `evenly` und `withdots` zur Verfügung. Eine einfache Anweisung, einen Pfad gestrichelt zu zeichnen, wäre etwa

`draw` ⟨*Pfad*⟩ `dashed evenly;`

Durch Transformationen (s. u.) können die vordefinierten Muster auch einfach verändert werden. Die Strichlänge von `evenly` kann durch Skalieren verändert werden, sodass

`draw` ⟨*Pfad*⟩ `dashed evenly scaled 0.5;`

den Pfad mit einem engeren Strichmuster zeichnet. Für umfangreichere Aufgaben gibt es zudem den Befehl `dashpattern`, mit dem eigene Muster definiert werden können.

Beschriftungen

Mit Hilfe der Anweisung

`label.`⟨*Suffix*⟩ `(btex` ⟨*Anweisung*⟩ `etex,`⟨*Pair*⟩`);`

können Punkte mit Beschriftungen versehen werden. Als Ort für die Beschriftung stehen dabei die acht Himmelsrichtungen O, NO, N, NW, W, SW, S und SO rund um den zu beschriftenden Punkt zur Verfügung. Die Auswahl der Platzierung geschieht dabei über ⟨*Suffix*⟩, welches als mögliche Werte

`rt`, `urt`, `top`, `ulft`, `lft`, `llft`, `bot` und `lrt`

annehmen kann. Dabei steht `rt` für ‚right', `lft` für ‚left', `ulft` für ‚upper left' usw. Als ⟨*Anweisung*⟩ ist praktisch jede beliebige (La)TEX-Anweisung zulässig, wenn sie nur durch die im Kopf geladenen Pakete abgedeckt ist.

Schnittpunkte

Schnittpunkte zwischen Pfaden können mit den Befehlen

⟨*Paar*⟩ = ⟨*Pfad1*⟩ `intersectiontimes` ⟨*Pfad2*⟩;
⟨*Pfad*⟩ = ⟨*Pfad1*⟩ `cutafter` ⟨*Pfad2*⟩;
⟨*Pfad*⟩ = ⟨*Pfad1*⟩ `cutbefore` ⟨*Pfad2*⟩;

bestimmt werden. Dabei erhält man entweder die zwei Längenparameter, bei denen sich der Schnittpunkt auf den beiden Pfaden befindet oder denjenigen Teil von ⟨*Pfad1*⟩, der sich vor oder nach dem Schnittpunkt mit ⟨*Pfad2*⟩ befindet. Schwierigkeiten ergeben sich nur, wenn es mehrere Schnittpunkte gibt. METAPOST wählt dann einen aus, der möglicherweise nicht der gewünschte ist. In dieser Situation kann man zunächst versuchen, einen der beteiligten Pfade durch `reverse` umzukehren. Liefert METAPOST auch dann nicht das gewünschte Ergebnis, so müssen den obenstehenden Makros Teilpfade übergeben werden, die sich sicher in nur einem Punkt schneiden. Mit etwas Übung versteht man sehr schnell wie METAPOST in solchen Fällen „denkt" und man wird die Pfade von Anfang an so anlegen, dass die gewünschten Ergebnisse vorliegen.

Transformationen

Besonders mächtig und nützlich ist die Möglichkeit, durch elementare Transformationen neue Punkte oder Pfade zu berechnen. Transformationen können auf Punkte, Pfade oder sogar ganze Bilder angewendet werden. Die verfügbaren Transformationen sind

⟨*Objekt*⟩ `rotated` ⟨*Numeric*⟩
⟨*Objekt*⟩ `scaled` ⟨*Numeric*⟩
⟨*Objekt*⟩ `shifted` ⟨*Pair*⟩
⟨*Objekt*⟩ `slanted` ⟨*Numeric*⟩
⟨*Objekt*⟩ `xscaled` ⟨*Numeric*⟩
⟨*Objekt*⟩ `yscaled` ⟨*Numeric*⟩
⟨*Objekt*⟩ `zscaled` ⟨*Pair*⟩
⟨*Objekt*⟩ `rotatedaround` (⟨*Pair*⟩,⟨*Numeric*⟩)
⟨*Objekt*⟩ `reflectedabout` (⟨*Pair*⟩,⟨*Pair*⟩)

Die Namen der Transformationen sind weitestgehend selbsterklärend, detailliertere Informationen sind in [3] zu finden. Es gibt einerseits die Möglichkeit, ein Objekt mittels `scaled` in beiden kartesischen Koordinatenrichtungen um den gleichen Faktor zu skalieren, oder andererseits mit `xscaled` und `yscaled` verschiedene Faktoren auszuwählen. Eine Ellipse mit den Halbachsen $a = 40\,\text{mm}$ und $b = 20\,\text{mm}$, die außerdem um 60° gegen den Uhrzeigersinn geneigt ist, würde man also durch den Befehl

⟨*Pfad*⟩ `= fullcircle xscaled 80mm yscaled 40mm rotated 60;`

erhalten. Bei der Verwendung mehrerer Transformationen innerhalb einer Anweisung ist entweder auf die richtige Reihenfolge oder auf richtige Klammerung zu achten. So liefert

⟨*Objekt*⟩ `shifted` ⟨*Pair*⟩ `rotated` ⟨*Numeric*⟩`;`

ein anderes Resultat, als

⟨*Objekt*⟩ `rotated` ⟨*Numeric*⟩ `shifted` ⟨*Pair*⟩`;`

weil im ersten Fall zuerst ⟨*Pair*⟩ gedreht wird und dann ⟨*Objekt*⟩ um das hieraus resultierende Ergebnis verschoben wird.

Einbindung in LaTeX-Dokumente

Das Einbinden von METAPOST-Bildern in LaTeX-Dokumente geschieht wie bei allen anderen Bildformaten durch das `graphicx`-Paket. Nachdem das Paket mit

`\usepackage[`⟨*Treiber*⟩`]{graphicx}`

geladen wurde, können METAPOST-Bilder einfach durch

`\includegraphics{`⟨*Dateiname*⟩`}`

eingebunden werden. Dabei ist lediglich zu beachten, dass der verwendete DVI-Treiber in der Lage ist, EPS-Grafiken zu verarbeiten. An dieser Stelle sei auch darauf hingewiesen, dass die METAPOST-Bilddateien zwar EPS-Format aufweisen, aber dennoch keine selbstständigen Grafikdateien in dem Sinne sind, dass sie in beliebiger Weise weiterverarbeitet oder mit einem PostScript-Viewer betrachtet werden können. Sie sind sozusagen darauf angewiesen, in ein LaTeX-Dokument eingebunden und vom DVI-Treiber mit Fonts versorgt zu werden. Hinsichtlich einer kleinen Datenmenge ist dieses Vorgehen sehr sinnvoll. Es muss nicht jede Bilddatei von neuem Fonts mitbringen, die vom Rest des Dokuments ohnehin angefordert werden. Bei der Entwicklung von METAPOST-Zeichnungen erfordert dieses aber einen kleinen Kunstgriff, will man nicht zum Betrachten jedes Mal das gesamte Dokument übersetzen. Es ist hierzu ratsam, einfach ein kleines LaTeX-Dokument bereitzuhalten, welches

nur die zur Einbindung einer Grafikdatei notwendigen Anweisungen enthält, und dieses zum Betrachten von einzelnen METAPOST-Bildern zu verwenden.

Für die Anwender von pdf(LA)TEX ist besonders angenehm, dass die METAPOST-Bilder zu diesem Zweck nicht erst in das PDF-Format konvertiert werden müssen, sondern von pdf(LA)TEX direkt eingelesen und umgewandelt werden. Hierzu ist außer der Option `pdftex` für das `graphicx`-Paket lediglich der Befehl

```
\DeclareGraphicsRule{*}{mps}{*}{}
```

im Vorspann notwendig.

Beispiele

Die folgenden Beispiele sollen zeigen, auf welche Weise mit Hilfe von METAPOST ansprechende Zeichnungen erstellt werden können. Die notwendigen Anweisungen sind fast vollständig angegeben, sodass der Leser die Beispiele leicht nachvollziehen kann. Es soll dadurch auch deutlich werden, dass oft nur wenige Programmzeilen notwendig sind, um eine vollständige Zeichnung zu erzeugen.

Graph eines Fachwerks

Als erstes Beispiel soll der in Abb. 2 gezeigte Graph eines Fachwerks mit METAPOST gezeichnet werden. Festgelegt sind dabei die Knotenpunkte `z0`, `z1` und `z2`. Die Knoten `z3` und `z4` sollen exakt in der Mitte zwischen den Verbindungen `z0-z1` und `z0-z2` liegen. Außerdem sollen die Stäbe `z3-z5` und `z4-z6` auf den Verbindungen `z0-z1` und `z0-z2` jeweils senkrecht stehen.

Nach beliebiger Zuweisung der Einheitslänge `u` werden mit

```
z0  = origin;
z1  = (-4u,-2u);
z2  = ( 4u,-2u);
```

zunächst die Eckpunkte festgelegt. Die Anweisungen

```
z3  = 0.5[z0,z1];
z4  = 0.5[z0,z2];
```

berechnen die Punkte `z3` und `z4`, sodass sie in der Mitte zwischen `z0` und `z1` bzw. `z2` liegen. Aus den Bedingungen

```
z5                        = whatever[z1,z2];
z6                        = whatever[z1,z2];
(z1-z0) dotprod (z5-z3) = 0;
```

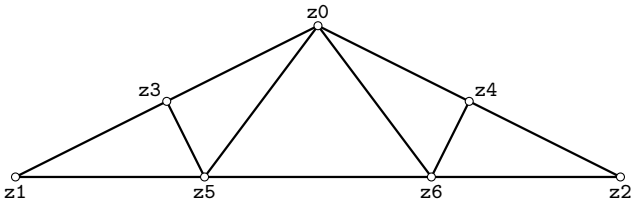


Abbildung 2: Ein Fachwerk aus elf Stäben und sieben Knoten

```
(z2-z0) dotprod (z6-z4) = 0;
```

kann METAPOST schließlich die Punkte `z5` und `z6` berechnen.

Die Stäbe werden durch die Anweisungen

```
pickup pencircle scaled 0.85pt;
draw z0--z1--z2--cycle;
draw z3--z5--z0;
draw z4--z6--z0;
```

mit einem etwas dickeren Stift gezeichnet und die Markierung der Knoten durch kleine Kreise geschieht durch

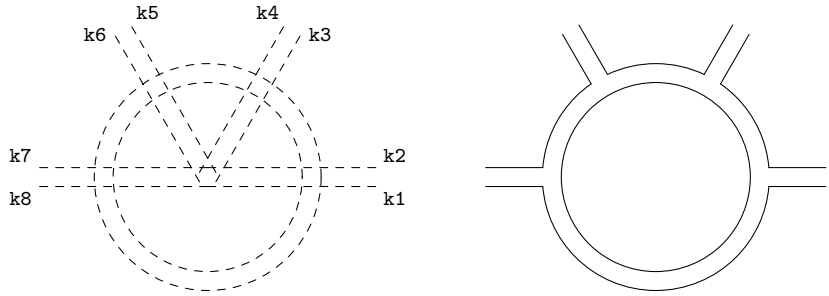
```
pickup pencircle scaled 0.4pt;
for a = 0 upto 6:
   unfill fullcircle scaled 3pt shifted z[a];
   draw   fullcircle scaled 3pt shifted z[a];
endfor
```

Ringhybridkoppler

Die Aufgabe, einen Ringhybridkoppler zu zeichnen, zeigt auch sehr schön die Überlegenheit von METAPOST gegenüber einer Grafikeingabe durch ein Zeigegerät. Ein Ringhybridkoppler ist ein Bauelement aus der Hochfrequenztechnik. Dabei handelt es sich um eine kreisförmige Leitung, von der unter den Winkeln 0°, 60°, 120° und 180° weitere Leitungen abzweigen. Die Verlängerungen der Leitungskanten gehen dabei nicht durch den Mittelpunkt der Ringleitung und die Abzweigleitungen sollen die gleiche Breite haben wie die Ringleitung (Abb. 3).

Von der Außenkante der Ringleitung sind nur die Bögen zwischen den Kanten der abzweigenden Leitungen zu zeichnen. Es ist daher zweckmäßig, durch

```
path    k[], innenkreis, aussenkreis;
numeric a;
```



(a) Hilfspfade (b) fertige Zeichnung

Abbildung 3: Zeichnung eines Ringhybridkopplers

```
innenkreis  = fullcircle scaled 10u;
aussenkreis = fullcircle scaled 12u;
k1          = (0,-0.5u)--(9u,-0.5u);
k2          = (0, 0.5u)--(9u, 0.5u);
k3          = k1 rotated  60;
k4          = k2 rotated  60;
k5          = k1 rotated 120;
k6          = k2 rotated 120;
k7          = k1 rotated 180;
k8          = k2 rotated 180;
```

die Pfade in Abb. 3a zu definieren. Durch Anwendung von `cutbefore` und `cutafter` werden dann die Teilbögen gezeichnet, sodass durch

```
pickup pencircle scaled 0.4pt;

draw innenkreis;
draw aussenkreis cutbefore k2 cutafter k3;
draw aussenkreis cutbefore k4 cutafter k5;
draw aussenkreis cutbefore k6 cutafter k7;
draw aussenkreis cutbefore k8 cutafter k1;

for a = 1 upto 8:
   draw k[a] cutbefore aussenkreis;
endfor
```

die Zeichnung wie in Abb. 3b fertiggestellt wird.

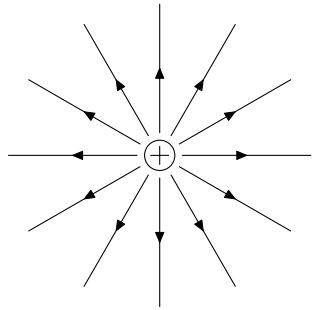


Abbildung 4: Elektrostatisches Feld einer Punktladung

Elektrostatisches Feld einer Punktladung

Als letztes Beispiel soll hier die Zeichnung des elektrostatischen Feldes einer positiven Punktladung dienen. Das Ergebnis ist in Abb. 4 gezeigt. Zunächst wird mit

```
pickup pencircle scaled 0.4pt;

draw fullcircle scaled 0.4u;
draw (-0.12u,0)--(0.12u,0);
draw (0,-0.12u)--(0,0.12u);
```

die Punktladung mit einem +-Zeichen gezeichnet. Die radial verlaufenden Feldlinien mit einer Pfeilspitze in ihrer Mitte entstehen durch die Anweisungen

```
path feldlinie;
numeric a;

feldlinie = (0.3u,0)--(2u,0);

for a = 0 upto 11:
   drawarrow subpath(0,0.5) of feldlinie rotated (a*30);
   draw      subpath(0.5,1) of feldlinie rotated (a*30);
endfor
```

die nach dem vorher Gesagten keine weiteren Erklärungen benötigen und leicht nachvollzogen werden können.

Schlusswort

Auch hier gilt die häufige Schlussbemerkung, dass in diesem Rahmen die ganzen Möglichkeiten und Funktionen von METAPOST nicht aufgezeigt werden können. Die Absicht des Autors war auch nicht die Erstellung einer Referenzkarte, sondern vielmehr den Appetit zu wecken und die Hemmschwelle zur Benutzung einer Beschreibungssprache für die Erstellung von Abbildungen abzubauen. Außerdem gibt es in METAPOST stets mehrere Wege, die zur selben Zeichnung führen. Hier ist sehr viel Spielraum für einen eigenen Programmierstil. Wenn die ersten Schritte einmal gewagt sind, wird man METAPOST aus seiner Grafikumgebung bald nicht mehr wegdenken können.

Literatur

[1] M. Goossens, S. Rahtz und F. Mittelbach: *The LaTeX Graphics Companion*; Addison-Wesley Publishing Company; Reading, Mass.; 1997.

[2] Hans Hagen: *Metafun*; Preliminary Version October 27, 2000; `http://www.pragma-ade.com`.

[3] John D. Hobby: *A User's Manual for MetaPost*; Bell Laboratories; `http://cm.bell-labs.com/cm/cs/cstr/162.ps.gz`, siehe auch `doc/mpman.ps` in `CTAN/graphics/metapost/metapost.tar.gz`.

[4] A. Hoenig: *TeX unbound*; Oxford University Press; Oxford; 1998.

[5] Donald E. Knuth: *The METAFONT Book*; Bd. Volume C of Computers and Typesetting; Addison-Wesley Publishing Company; Reading, Mass.; 1986.

[6] H. Kopka: *LaTeX*; Bd. 2. Ergänzungen; Addison-Wesley; Bonn; 1997.