Einführung in METAPOST

Uwe Siart

3. Bayerischer TEX-Stammtisch

23. Juli 2005

Übersicht

Beschreibung des Programms

Einige wichtige Befehle

Code-Beispiele

Vergleich METAPOST–PSTricks

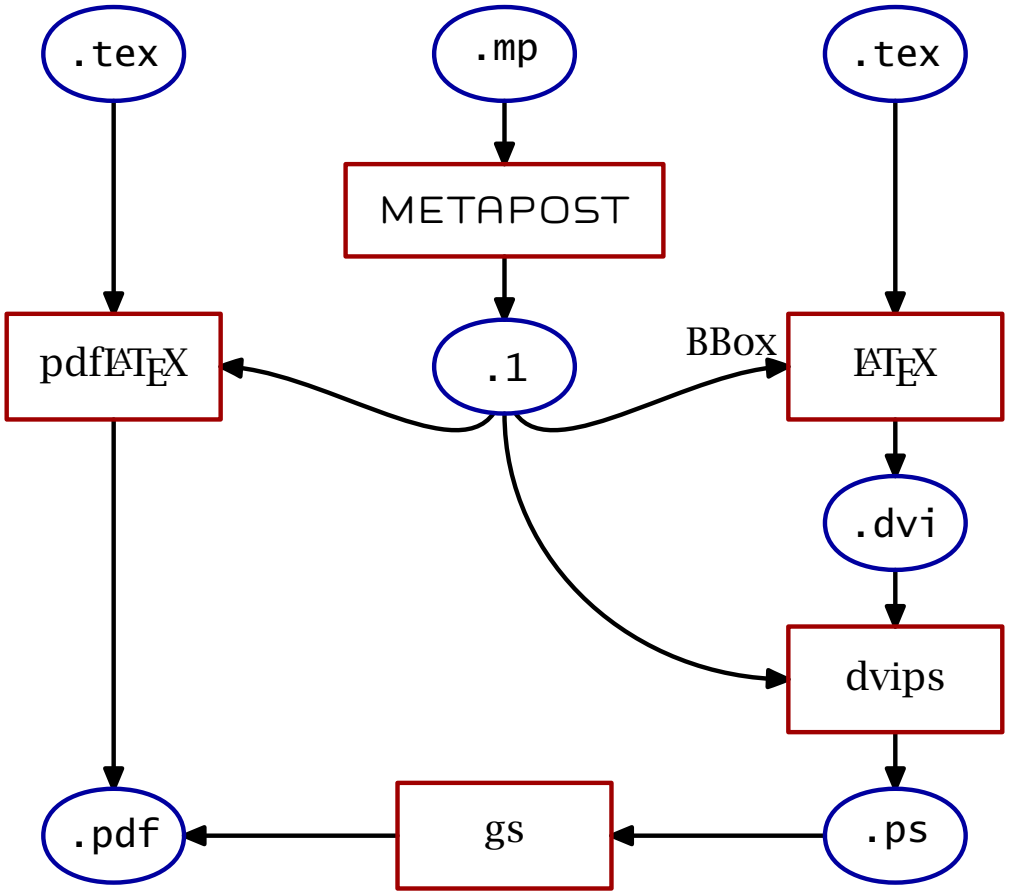
Was ist METAPOST?

- METAPOST ist eine **Modifikation von** METAFONT zur Erzeugung von Encapsulated PostScript (EPS) anstelle von Pixelfonts.
- Grafikelemente werden in einem kartesischen Koordinatensystem beschrieben und mit verschiedenen Stiften, Linientypen und Farben gezeichnet.
- Dabei stehen **programmiertechnische Funktionen und Ablaufstrukturen** zur Verfügung:
  - Wiederholungsschleifen
  - Lösen von Gleichungssystemen
  - mathematische Funktionen und vektoranalytische Transformationen
  - Berechnung von Schnittpunkten, Teilpfaden und Bogenlängen

Vorteile von METAPOST?

- reproduzierbare und mathematisch **präzise definierte Grafiken**, da keine Eingaben über Zeigegerät
- **Skalierung von Bildern** durch Angabe der Einheitslänge, daher kein Einfluss auf Strichstärken und Schriftgrößen.
- **schlanke vektorielle Bilddateien**, weil keine Fonts oder sonstige unnötige Ressourcen eingebunden werden
- Textelemente werden von **TEX/LATEX** gesetzt, daher voller Zugriff auf die gewohnten Fähigkeiten beim Text- und Mathematiksatz
- **ästhetisch hochstehende Fähigkeiten** bei gekrümmten Kurven, Formen und Pfeilspitzen

Einordnung in den LaTeX-Lauf



Grundgerüst einer METAPOST-Eingabedatei

- Falls Beschriftungen anstatt mit TEX mit LATEX gesetzt werden sollen, muss ein LATEX-Kopf in den Vorspann. Der METAPOST-Aufruf lautet dann

  ```
  mpost --tex=latex dateiname
  ```

- Verwendete Pakete müssen im Vorspann geladen werden, das `\end{document}` entfällt, weil es von METAPOST beim LATEX-Aufruf selbst erzeugt wird.
- Eine METAPOST-Datei kann beliebig viele `beginfig`-`endfig`-Paare beinhalten. So entstehen mehrere Bilder in einem Lauf. Die Nummer im Argument von `beginfig` bildet die Endung der zugehörigen Bilddatei.

Grundgerüst einer METAPOST-Eingabedatei

```
% --[LaTeX-Vorspann (optional)]---------------
verbatimtex
\documentclass[10pt,a4paper]{article}
... evtl. weitere Pakete
\begin{document}
etex
% --[Minimalinhalt]--------------------------
beginfig(1);
  Anweisungen
endfig;
end
```

Um Schriftmischungen zu vermeiden, ist es sinnvoll, wenn die METAPOST-Datei denselben Vorspann hat, wie das Dokument, in das die Bilder eingebunden werden.

Variablentypen und deren Deklaration

```
pair      ⟨BezeichnungA⟩, ⟨BezeichnungB⟩, ...;
numeric   ⟨BezeichnungA⟩, ⟨BezeichnungB⟩, ...;
path      ⟨BezeichnungA⟩, ⟨BezeichnungB⟩, ...;
```

- Deklaration der Form

  ```
  variable   name[]
  ```

  erlaubt numerische Indizierung (`name1`, `name2`, ...).
- Die Variablen `z[]` vom Typ `pair` sind vordefiniert. Dabei sind `x1` und `y1` die Koordinaten von `z1`.
- Bei `numeric`-Variablen sind Längeneinheiten möglich (bp (default), pt, in, cm, mm, ...), `numeric`-Variablen können aber auch die Bedeutung eines Skalars oder eines Winkels in Grad haben.

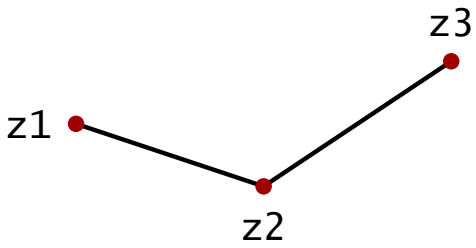
Transformationen

Drehung um (0,0)	⟨*Objekt*⟩	`rotated`	⟨*Numeric*⟩
Skalierung	⟨*Objekt*⟩	`scaled`	⟨*Numeric*⟩
Translation	⟨*Objekt*⟩	`shifted`	⟨*Pair*⟩
Neigung	⟨*Objekt*⟩	`slanted`	⟨*Numeric*⟩
Dehnung in x-Richtung	⟨*Objekt*⟩	`xscaled`	⟨*Numeric*⟩
Dehnung in y-Richtung	⟨*Objekt*⟩	`yscaled`	⟨*Numeric*⟩
komplexe Multiplikation	⟨*Objekt*⟩	`zscaled`	⟨*Pair*⟩
Drehung um belieb. Punkt	⟨*Objekt*⟩	`rotatedaround`	(⟨*Pair*⟩,⟨*Numeric*⟩)
Spiegelung an Gerade	⟨*Objekt*⟩	`reflectedabout`	(⟨*Pair*⟩,⟨*Pair*⟩)

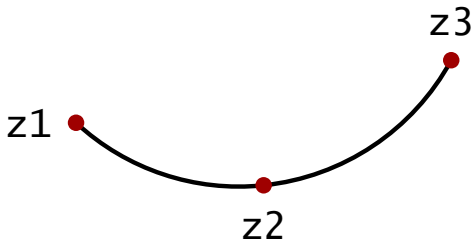
Einige vordefinierte Variablen (Pfade und Bilder)

`fullcircle`	mittelpunktszentrierter Vollkreis mit Durchmesser 1 (bp)
`halfcircle`	mittelpunktszentrierter oberer Halbkreis mit Durchmesser 1 (bp)
`quartercircle`	mittelpunktszentrierter Viertelkreis im 1. Quadranten mit Durchmesser 1 (bp)
`unitsquare`	Einheitsquadrat mit linker unterer Ecke auf (0,0)
`pencircle`	runder Stift mit Durchmesser 1 (bp)
`evenly`	Strichmuster
`withdots`	Punktmuster
`origin`	der Punkt (0,0)

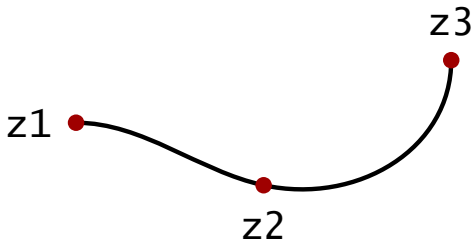
Pfade und Richtungsangaben



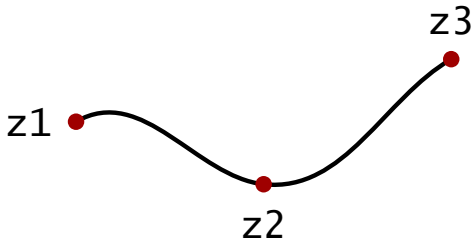

```
draw z1--z2--z3
```




```
draw z1..z2..z3
```



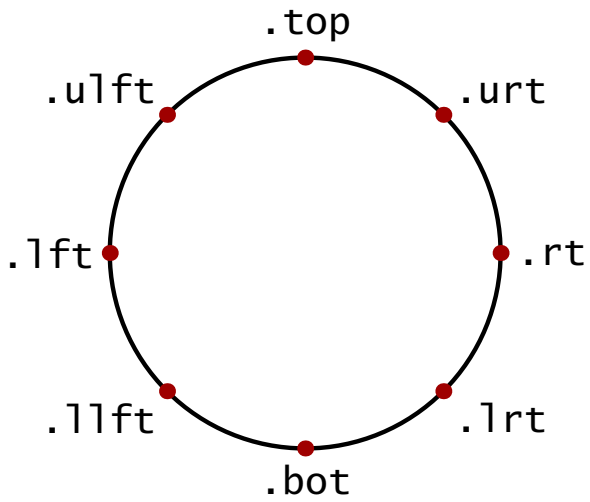

```
draw z1{right}..z2..{up}z3
```




```
draw z1{dir 30}..z2..{dir 30}z3
```

Linientypen

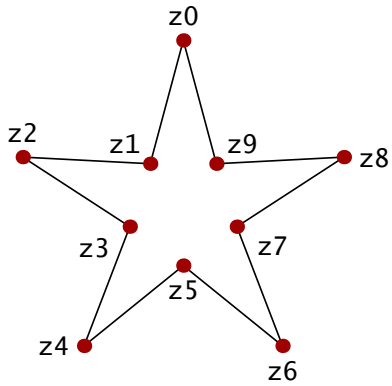
Beschriftungen




```
ring = fullcircle scaled 5u;
```

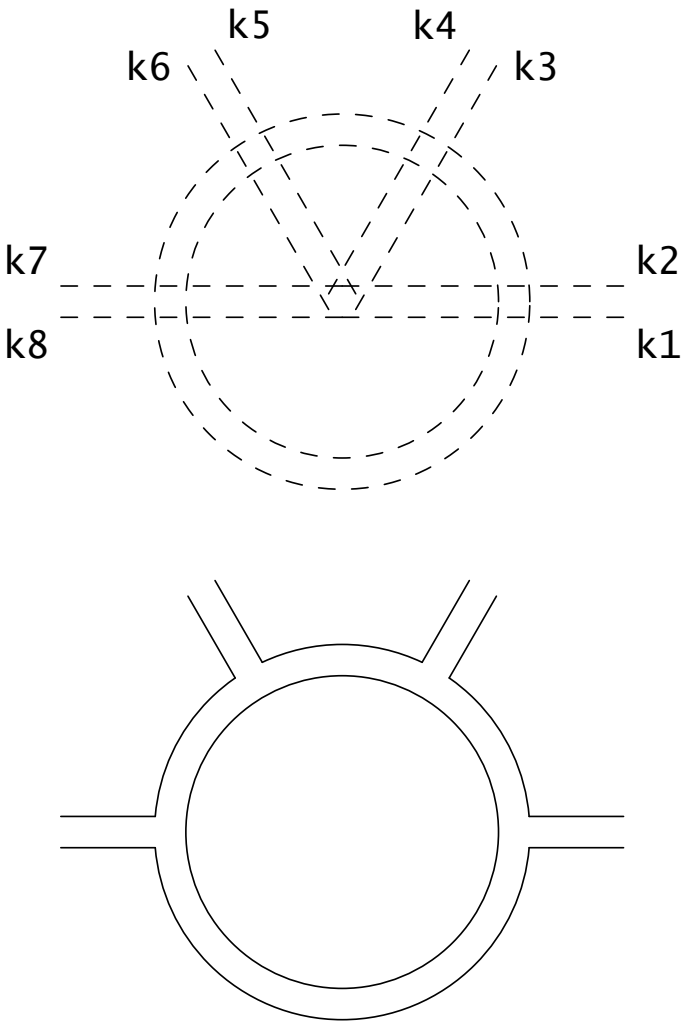
```
label.rt   (btex \texttt{.rt}   etex,point 0 of ring);
label.urt  (btex \texttt{.urt}  etex,point 1 of ring);
label.top  (btex \texttt{.top}  etex,point 2 of ring);
label.ulft (btex \texttt{.ulft} etex,point 3 of ring);
label.lft  (btex \texttt{.lft}  etex,point 4 of ring);
label.llft (btex \texttt{.llft} etex,point 5 of ring);
label.bot  (btex \texttt{.bot}  etex,point 6 of ring);
label.lrt  (btex \texttt{.lrt}  etex,point 7 of ring);
```

Beispiel – Europaflagge (nachempfunden)



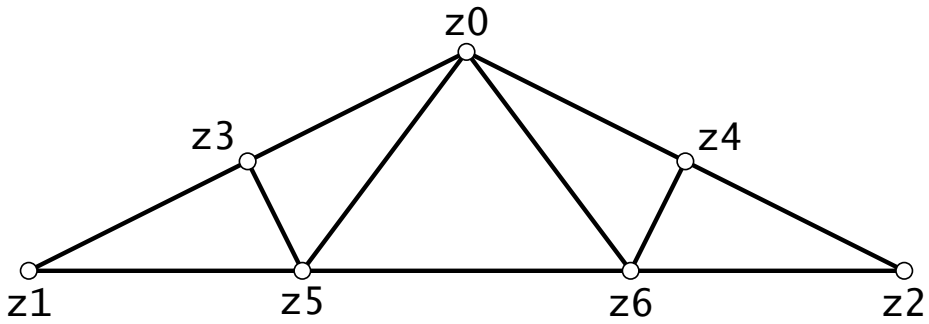
```
numeric a;
path    stern, flagge;
flagge = ( 5u,-4u)--( 5u, 4u)--
         (-5u, 4u)--(-5u,-4u)--cycle;

for a = 0 step 2 until 8:
  z[a] = 0.6u * (dir (a*36 + 90));
endfor
for a = 1 step 2 until 9:
  z[a] = 0.2u * (dir (a*36 + 90));
endfor

stern =
  for a = 0 upto 9:
    z[a]--
  endfor
cycle;

fill flagge withcolor blue;
for a = 0 upto 11:
  fill stern shifted ((2.5u,0)
  rotated (a*360/12)) withcolor red+green;
endfor
```

Beispiel – Ringhybridkoppler




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

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;

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
```

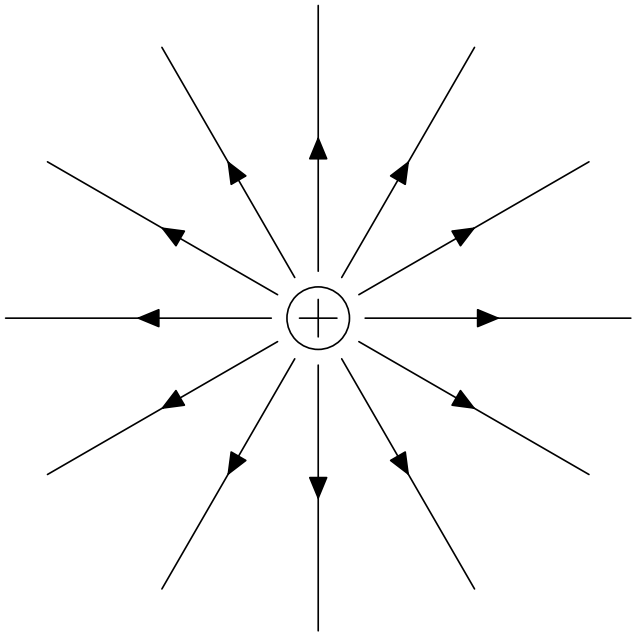
Beispiel – Graph eines Fachwerks




```
numeric a;

z0  = origin;
z1  = (-4u,-2u);
z2  =  (4u,-2u);
z3  = 0.5[z0,z1];
z4  = 0.5[z0,z2];

z5  = whatever[z1,z2];
z6  = whatever[z1,z2];

(z1-z0) dotprod (z5-z3) = 0;
(z2-z0) dotprod (z6-z4) = 0;
```

```
pickup pencircle scaled 0.85pt;

draw z0--z1--z2--cycle;
draw z3--z5--z0--z6--z4;

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
```

Beispiel – Feld einer Punktladung



```
path    feldlinie;
numeric a;

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

pickup pencircle scaled 0.4pt;

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

```
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
```

Vorteile von PSTricks

- Knotenverbindungen automatisch ziehen
- Schraffuren und Kreuzschraffuren, Füllungen
- Plotachsen erzeugen und Funktionen berechnen und plotten
- Krümmungsradien in Knickstellen
- gestrichelte Linien beginnen und enden mit Strich

Vorteile von METAPOST

- Schnittpunkte beliebiger Pfade berechnen
- Zugriff auf Pfadabschnitte (parametrisch und über Pfadlänge)
- angepasste Pfeilspitzen an gekrümmten Linienenden
- Pfade durch Stützpunkte, Steigungen und Spannungen definieren
- analytische Geometrie (2D, Mediation, affine Transformation)
- Lösen linearer Gleichungssysteme
- Zugriff auf Randpunkte von Boxen (boxes-Makro)