

# Dokumentation zum MetaPost-Paket `diagram.mp`

## 1 Allgemeines

Dieses Paket ermöglicht das Erstellen von Diagrammen einschließlich Achsenbeschriftungen. Zusätzlich kann man Punkte (z. B. Meßwerte) eintragen und eine Regressionsgerade (Ausgleichsgerade) berechnen lassen. Unter Verwendung der Projektionsfunktionen `pxy` und `pxz` können alle Koordinaten bzgl. der Diagrammskalierung angegeben werden.

## 2 Erzeugen von Diagrammen

Diagramme werden mit Konstruktion

```
beginndiagram(num)(dimx,dimy);    ...    enddiagram;
```

erzeugt. Jedes Diagramm ist ein eigenes MetaPost Bild mit der Nummer *num*. (Ohne die Befehle `beginfig` und `endfig`!) Die Längenparameter *dimx* und *dimy* geben die Größe des Diagramms an. Innerhalb dieser Umgebung sollte gleich am Anfang ein Aufruf des Befehls `xyscale` erfolgen! (Die Liniendicke darf natürlich noch vorher festgelegt werden.)

## 3 Befehle

- `xyscale(xmin,xmax,nx)(ymin,ymax,ny);`  
Skalierung des Diagramms. Legt die Wertebereiche  $x_{\min} \leq x \leq x_{\max}$  und  $y_{\min} \leq y \leq y_{\max}$  fest und zeichnet ein Raster mit *n<sub>x</sub>* mal *n<sub>y</sub>* Kästchen. (Gibt man Null an, dann kein Raster).
- `crsign(text,size);`  
Legt ein Label *text* mit *size* skaliert in die rechte untere Ecke (als „Copyright“ o. ä.)
- `xtick(num,text);`     `ytick(num,text);`  
Zeichnet bei *num* einen kleinen Strich an die Achse und schreibt *text* darunter bzw. daneben. Diese Befehle werden normalerweise nicht direkt aufgerufen, stattdessen verwendet man folgende:
- `xticks(start,ende,n, m,text);`     `yticks(start,ende,n, m,text);`  
Beschriftet die Achsen von *start* bis *ende* mit *n* + 1 Werten, die auf *m* Nachkommastellen gerundet werden. *text* wird ans Ende der Achse geschrieben (z. B. das zugehörige Formelzeichen).
- `xlabel(shift,text);`     `ylabel(shift,text);`  
Schreibt *text* unter bzw. links neben das Diagramm (ausführliche Achsenbezeichnung). Der Parameter *shift* dient zur Feinpositionierung (Abstandskorrektur).
- `plot(x,y);`  
Zeichnet den Punkt (*x*, *y*) ins Diagramm ein, wobei das Erscheinungsbild durch `_plotsymbol` festgelegt ist. Die Koordinaten werden in die Regressions-Statistik aufgenommen.
- `evaluate;`  
Bestimmt die Parameter `_a`, `_b` und `_r` der linearen Regression unter Verwendung aller bisher mit `plot` eingegebenen Punkte. Die Ergebnisse stehen dann auch in der `.log`-Datei.
- `regline`  
Führt `evaluate` aus und gibt die Regressionsgerade (als `path`) zurück.
- `pxy(objekt)`  
Das *objekt* (ein Ausdruck vom Typ `pair`, `path` oder `picture`) wird in MetaPost Koordinaten umgerechnet. Will man also den Punkt (*x*, *y*) ansprechen, wobei *x* und *y* der Achsenskalierung entsprechen sollen, so muß man `pxy((x,y))` schreiben.

## 4 Eine weitere Achse

Es besteht die Möglichkeit, eine vertikale Achse (z-Achse) einzurichten, falls z. B. verschiedene Größen in einem Diagramm dargestellt werden sollen.

Dazu gibt es die zusätzlichen Befehle `zscale`, `ztick`, `zticks`, `zplot` und `pxz`.

## 5 Interne Variablen

Alle interne Variablen fangen mit `_` an. Normalerweise muß man sie nicht direkt ansprechen, manchmal ist es aber nützlich, die Werte abzufragen oder zu ändern.

<code>_x1, _x2, _dimx</code>	Wertebereich der $x$ -Achse ( $_{x1} \leq x \leq _{x2}$ ) und Breite des Diagramms. Analoge Variablen gibt es für $y$ und $z$ (außer <code>_dimz</code> ).
<code>_n, _sumx, _sumy, _sumxq, _sumyq, _sumxy</code>	Statistische Datenregister: $n, \sum x, \sum y, \sum x^2, \sum y^2, \sum xy$
<code>_a, _b, _r</code>	Regressionsergebnisse (Absolutwert, Anstieg, Korrelationskoeffizient)
<code>_ticklength</code>	Länge der Markierungsstriche (Voreinstellung 4pt)
<code>_plotsymbol</code>	Markierungssymbol als Makro vom Typ <code>picture</code> . Voreinstellung: <code>def _plotsymbol=thelabel(btex \$\circ\$ etex,(0,0)) enddef;</code>

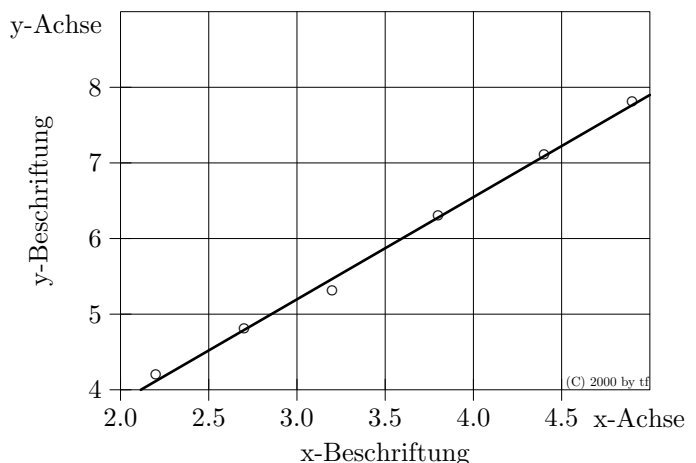
## 6 Neue Funktionalität (ab 08/2003)

Die neue Version ist (hoffentlich) voll abwärtskompatibel. Es haben sich folgende Änderungen/erweiterungen ergeben:

- `pxy` und `pxz` brauchen um das Argument keine Klammern mehr. Das Argument von `plot` ist jetzt `pair`.
- Mit `reg_on`; und `reg_off`; kann man die Regressionsdatenspeicherung (bei `plot`) ein- und ausschalten. Den (bisher undokumentierten) Befehl `regreset`; braucht man so nur noch bei mehreren Regressionen in einem Diagramm.
- Mit `plotsymbol(...)` kann man einfacher das Plotsymbol ändernd (`picture`- oder `btex..etex`-Ausdruck)
- Mit `gridstyle(...)`; kann man das Aussehen der inneren Linien verändern. Es gibt dafür auch das Makro `dotted(dist,width—)`, welches Punkte der Größe `width` im Abstand `dist` zeichnet. Der Rand des Diagramms wird mit der aktuellen Strichstärke gezeichnet (vor `xyscale` festzulegen)
- `xtick` und `yytick` arbeiten wie `xtick` und `ytick`, platzieren die Labels jedoch auf der gegenüberliegenden Seite.  
Alle `...tick`-Befehle haben jetzt ein optionales Suffix `.i` oder `.e`, das angibt, ob ein Tick nur nach innen oder außen zeigt: Beispiel `xticks.i(5,8,6,1,"")`;  
`_ticksep` steuert den Abstand der Beschriftung an Ticks.
- Mit `_bgcolor:=...` kann man die Hintergrundfarbe des Diagrammes einstellen
- `legend(...)` erzeugt eine Legende (als `picture`), welche mit `label` (und ggf. `boxed(...)` aus `tftools.mp`) eingezeichnet wird. Einträge mit `titem(label,xshift)` bzw. `litem(label,linestyle)`.  
Interne Variablen: `_llength` (Linienlänge), `_lysep` (Zeilenabstand), `_ypos` (akt. y-Pos.).

## 7 Beispiele

```
input diagram;
begindialog(1)(7cm,5cm);
pickup pencircle scaled 0.2pt;
xyscale(2,5,6)(4,9,5);
crsign(btex (C) 2000 by tf etex,0.5);
xticks(2,4.5,5,1,"x-Achse");
xlabel(-1mm,"x-Beschriftung");
yticks(4,8,4,0,"y-Achse");
ylabel(-3mm,"y-Beschriftung");
plot(2.2,4.2);
plot(2.7,4.8);
plot(3.2,5.3);
plot(3.8,6.3);
plot(4.4,7.1);
plot(4.9,7.8);
pickup pencircle scaled 1pt;
draw pxy(regline);
enddiagram;
```



```

beginndiagram(2)(6cm,5cm);
gridstyle(dotted(0.25pt,0.3pt));
pickup pencircle scaled 0.7pt;
_ticksep:=2pt;
_bgcolor:=0.9[blue,white];
xyscale(0.50,0.56,6)(0.4,1,6);
xticks.i(0.5,0.55,5,2,btex  $\lambda$  etex);
xlabel(-1mm,btex Wellenl\`ange in  $\mu\text{m}$  etex);
yticks.i(0.4,0.9,5,1,btex  $T$  etex);
ylabel(-3mm,btex Transmission ( $\circ$ ) etex);
zscale(0,0.3,0);
zticks.i(0,0.25,5,2,btex  $\varepsilon$  etex);
zlabel(5mm,btex Extinktion ( $\bullet$ ) etex);
pair T[],E[];
T1:=(0.503,0.91);
T2:=(0.515,0.75);
T3:=(0.529,0.49);
T4:=(0.535,0.46);
T5:=(0.542,0.61);
T6:=(0.557,0.82);
E1:=(0.505,0.041);
E2:=(0.519,0.122);
E3:=(0.535,0.263);
E4:=(0.548,0.171);
E5:=(0.558,0.103);
pickup pencircle scaled 0.8pt;
drawoptions(withcolor blue);
for i:=1 upto 6: plot T[i]; endfor;
draw pxy T1 for i:=2 upto 6: ...pxy T[i] endfor;
drawoptions(withcolor red);
plotsymbol(btex  $\bullet$  etex);
for i:=1 upto 5: zplot E[i]; endfor;
draw pxz E1 for i:=2 upto 5: ...pxz E[i] endfor;
drawoptions();
pickup pencircle scaled 0.5pt;
draw pxy((0.534,0.4)--(0.534,1)) dashed evenly;
xxtick.e(0.534,"0.534");
newlabel.lt(boxed(legend(
  titem(btex Legende etex,0pt);
  litem(btex  $T$  etex,withcolor blue);
  litem(btex  $\varepsilon$  etex,withcolor red);
),2pt,0.5pt),pxy(_x2,_y1)+_ticklength*dir(135));
enddiagram;

```

