

Kapitel 9

Funktionen

9.1 Arithmetisch-logische Operationen	89
9.2 Elementare Funktionen	89
9.3 Ganzzahlige und Rundungsoperationen	90
9.4 Wandeloperationen	90

9.1 Arithmetisch-logische Operationen

MP kennt die vier Grundrechenarten und das Potenzieren, wobei Multiplizieren und Potenzieren die gleiche Bindungskraft besitzen, sodass es auf die Reihenfolge der Operationen ankommt. Die Auswertung erfolgt stets von links nach rechts. Vorrang vor diesen Operationen haben das unäre Plus und Minus, das spezielle Konstrukt zur Berechnung von Zwischenwerten und die Voranstellung eines Zahlenwerts vor eine Variable. Auch die pythagoräischen Operationen sind spezielle Konstrukte von MF und MP. Die Operationen mit der niedrigsten Bindungskraft sind die logischen Operationen, `not`, `and`, `or` mit den Vergleichstests, die `true` oder `false` ergeben können. Die Einzelheiten sind in Tabelle 9.1 zusammengestellt.

9.2 Elementare Funktionen

Die elementaren mathematischen Funktionen umfassen die Wurzel, die trigonometrischen Funktionen sowie die Exponential- und Logarithmusfunktion. Die Arcus- und

hyperbolischen Funktionen und deren Umkehrfunktionen berechnen sich daraus nach bekannten Formeln [8]. Tabelle 9.2 enthält die entsprechenden Definitionen.

9.3 Ganzzahlige und Rundungsoperationen

Zum echten Runden und zum Auf- und Abrunden auf die nächstgelegene ganze Zahl sowie zur Berechnung der Modulfunktion und der ganzzahligen Division gibt es die entsprechenden Operationen in Tabelle 9.3. Die Modulo-Funktion $r = \text{mod}(i, n)$ berechnet den Rest r der ganzzahligen Division i durch n . Ist der Rest gleich Null, ist die Zahl i ein Vielfaches von n .

Anwendung: In einer Schleife soll nach jeweils n Schritten eine bestimmte Anweisung ausgeführt werden.

```
for i=1...:
  if i mod n = 0: <Anweisung ausfuehren> fi
endfor
```

Weitere Operationen betreffen die Bestimmung des Maximums und Minimums einer Zahlenfolge sowie die Berechnung des absoluten Wertes einer Variablen. Die spezielle Operation `format` erfordert das Paket `graph.mp` und formatiert den Zahlenwert einer Variablen anhand eines Formatstrings als Fest- oder Gleitpunktzahl mit vorgeschriebener Stellenzahl.

9.4 Wandeloperationen

Es besteht oft die Notwendigkeit zur Wandlung einer Variablen vom Typ `numeric` in einen String und umgekehrt sowie eines Strings und einer Zahl in den Variablentyp `picture`. Dazu dienen die Operationen in Tabelle 9.4.

Tabelle 9.1: Arithmetisch-logische Operationen. Die Operationen sind in in Gruppen fallender Rangfolge (Bindungskraft) angegeben.

Symbol	Mathematik	Bezeichnung	MP-Beispiele
$+ \ -$	$+a, -a$	unäres Plus/Minus	$-5, -a$
$c[a, b]$	$a + c(b - a)$	Zwischenwerte (MP)	$0.5[2, 7] \rightarrow 4.5$
$* \ /$	$a \ b, a/b$	Multiplikation, Division	$2*3 \rightarrow 6, a*b, 1/2$
$**$	a^b	Potenz	$2**3 \rightarrow 8$ $4*3**2 \rightarrow 144, 4**3*2 \rightarrow 128$
$++$	$\sqrt{a^2 + b^2}$	Pythagoräische Operationen	$4++3 \rightarrow 5$
$+-+$	$\sqrt{a^2 - b^2}$		$5+-3 \rightarrow 4$
$+ \ -$	$a + b, a - b$	Addition, Subtraktion	$2+3 \rightarrow 5, 2-3 \rightarrow -1$
not	$\neg a$	logische Verknüpfungen	$\text{not}(2<3) \rightarrow \text{false}$
and	$a \wedge b$		$(2<3) \text{ and } (2=3) \rightarrow \text{false}$
or	$a \vee b$		$(2<3) \text{ or } (2=3) \rightarrow \text{true}$
$<$	$a < b$	Vergleichsoperatoren, kleiner	$(2<3) \rightarrow \text{true}$
$<=$	$a \leq b$	kleiner gleich	$(a<=0)$
$=$	$a = b$	gleich	$(2=3) \rightarrow \text{false}$
$>=$	$a \geq b$	größer gleich	$(5>=0) \rightarrow \text{true}$
$>$	$a > b$	größer	$(a>0)$
$<>$	$a \neq b$	ungleich	$(2<>3) \rightarrow \text{true}$

Tabelle 9.2: Elementare Funktionen.

Funktion	Mathematik	Bezeichnung	MP-Beispiele
$\text{sqrt}(x)$	$y = \sqrt{x}$	Quadratwurzel	$\text{sqrt } 2 \rightarrow 1.414$
$\text{sin}(\alpha)$	$y = \sin(x), \alpha = \frac{180}{\pi}x$	Sinus	$\text{sin} \ 30 \rightarrow 0.5$
$\text{cosd}(\alpha)$	$y = \cos(x)$	Cosinus	$\text{cosd} \ 30 \rightarrow 0.866$
$\text{mexp}(x')$	$y = e^x = \text{mexp}(256x),$ $x' = 256x$	Exponentialfunktion	$\text{mexp}(256) \rightarrow e = 2.718$
$\text{mlog}(x)$	$y = \ln(x) = 1/256 \text{mlog}(x)$	nat. Logarithmus	$1/256 \text{mlog}(2.718) \rightarrow 1$
	$y = \lg(x) = \text{mlog}(x)/\text{mlog}(10)$	dek. Logarithmus	$\text{mlog}(2)/\text{mlog}(10) \rightarrow 0.301$

Tabelle 9.3: Ganzzahlige und Rundungsoperationen.

Funktion	Mathematik	Bezeichnung	MP-Beispiele
$i \text{ div } n$	$i \div n$	ganzzahlige Division	$7 \text{ div } 2 \rightarrow 3$
$i \text{ mod } n$	$\text{mod}(i, n)$	Modulo	$7 \text{ mod } 2 \rightarrow 1$
$\text{odd } n$		ist n eine ungerade Zahl?	$\text{odd } 17 \rightarrow \text{true}$
$\max(x_1, \dots, x_n)$	$\max_i \{x_i\}$	Maximum	$\max(3, -2, 8) \rightarrow 8$
$\min(x_1, \dots, x_n)$	$\min_i \{x_i\}$	Minimum	$\min(3, -2, 8) \rightarrow -2$
$\text{abs}(x)$	$ a $	Betrag einer Zahl	$\text{abs}(-5) \rightarrow 5$
$\text{abs}(z)$	$ z = \sqrt{x^2 + y^2}$	Betrag eines Punktes	$\text{abs}(1, 1) \rightarrow 1.4159$
$\text{length}(p)$		Länge	von Pfad, String, Bild
$\text{round}(x)$		Runden auf ganze Zahl	$\text{round}(3.14159) \rightarrow 3$
$\text{ceiling}(x)$	$\lceil x \rceil$	Aufrunden auf ganze Zahl	$\text{ceiling}(3.14) \rightarrow 4$, $\text{ceiling}(-3.14) \rightarrow -3$
$\text{floor}(x)$	$\lfloor x \rfloor$	Abrunden auf ganze Zahl	$\text{floor}(3.14) \rightarrow 3$, $\text{floor}(-3.14) \rightarrow -4$
$\text{format}(\text{"\%nf"}, x)$		Rundet x auf n Dezimalen	$\text{format}(\text{"\%2f"}, 3.14159)$ $\rightarrow 3.14$ als picture

Tabelle 9.4: Wandeloperationen.

Funktion	Erläuterung	MP-Beispiele
$\text{decimal}(x)$	Wandelt eine Zahl in einen String	$\text{decimal}(3.14)$ $\rightarrow \text{"3.14"}$
$\text{ASCII } s$	Liest das erste Zeichen des Strings s und gibt dessen Positionsnummer n in der ASCII-Font-Tabelle aus	$\text{ASCII } \text{"A"} \rightarrow 65$
byte	wie ASCII	
$\text{char}(n)$	Gibt das ASCII-Zeichen als String aus, das diese Ordnungsnummer hat	$\text{char } 91 \rightarrow \text{"["}$
$\text{oct } "x"$	Wandelt die Oktalzahl x in eine Dezimalzahl	$\text{oct } \text{"133"} \rightarrow 91$
$\text{hex } "x"$	Wandelt die Hexzahl x in eine Dezimalzahl	$\text{hex } \text{"5B"} \rightarrow 91$
$\text{scantokens}(s)$	Wandelt den String s wie der Parser in eine Tokenliste. Im Fall eines 'numeric token' erhält man den Zahlenwert.	$\text{scantokens}(\text{"3.14"})$ $\rightarrow 3.14$
$\text{str } v$	v ist ein Suffix und kann entweder ein 'tag' oder ein Subscript sein. Falls 'tag' liefert $\text{str } v$ den String " v ", falls Suffix erhält man mit $\text{str } [v]$ den Zahlenwert als String, z. B. "3.14". Zwischen mehrere 'tag' wird je ein Punkt eingefügt, z. B. $\text{str } a \ b \ c \rightarrow \text{"a.b.c"}$.	
thelabel	Wandelt den String-Text in ein Picture	$\text{lab:=thelabel}(\text{"Text"}, \text{origin})$ $\rightarrow \text{lab}$ ist ein Picture
format	Formatiert eine Zahl und wandelt sie in ein Picture	
$\text{btex} \dots \text{etex}$	formatiert einen $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Text und wandelt das Ergebnis in ein Picture.	

Kapitel 10

Beschriftung

10.1 String-Label	94
10.2 T _E X/L ^A T _E X-Label	96
10.3 Dynamische Beschriftung	97
10.4 Platzierung von L ^A T _E X-Text	103

Zum Beschriften einer Zeichnung gibt es in MP den Label-Befehl. Der Einfachheit halber wiederholen wir die Syntax des Label-Befehls aus Abschnitt 3.4 und ergänzen sie durch Transformationen und Optionen.

```
label< label suffix >(< label expr >, < pair expr >)< transforms >< options >;  
< label suffix > → < empty > | top | bot | left | right | ulft | urt | llft | lrt  
< label expr > → < string expr > | < picture expr >< transforms >  
< transforms > → < empty > | < alle in Tabelle 7.2 genannten Transformer >  
< options > → < empty > | withcolor< farbe >
```

Der < label suffix > gibt an, wie der Text relativ zur Position < pair expr > gesetzt werden soll. Ohne Suffix wird der Text bezüglich dieses Punktes zentriert gesetzt. Dem label-Befehl können Transformationen und Optionen angefügt werden. Falls der Label-Text ein Picture ist, kann auch dieser transformiert werden, z. B. durch Drehung um 90° als Beschriftung der y -Achse eines Diagramms.

10.1 String-Label

Wir beginnen mit dem einfachsten Fall und verwenden als Label-Text einen String, der aus einer Folge von ASCII-Zeichen besteht und durch Doppelapostrophe begrenzt wird. Der String muss auf einer Zeile stehen, er darf keine Zeilenumbrüche und keinen Doppelapostroph enthalten. Ein innerer Doppelapostroph muss durch den String `ditto` der Länge 1 ersetzt werden, der als Zeichen einen Doppelapostroph enthält. Mehrere Teilstrings können mit `&` verkettet werden.

Faehre" Sylt" steht auf Kai-5 bereit

Bild 10.1: String-Label aus ASCII-Zeichen formatiert mit default-Font `cmr10`.

Beispiel: Ein String als Label-Text.

```
beginfig(1)
  k:=5;
  label("Faehre "&ditto&"Sylt"&ditto&" steht auf Kai "&
    decimal k&" bereit",origin);
endfig;
```

In einen Label-String können MP-Variable mit dem Operator `decimal` eingefügt werden, der den Zahlenwert in einen String wandelt.

Strings werden von MP mit dem Font `defaultfont` gesetzt, der auf Computer Modern, `cmr10`, voreingestellt ist. Dies ist nicht ganz unproblematisch, da dieser Font im Gegensatz zum ASCII-Code kein Leerzeichen hat und die ASCII-Zeichen `" < > \ _ { | }` fehlen und mit anderen Glyphen besetzt sind. Der ASCII-Text in Strings wird von MP als PostScript-Text interpretiert, gedruckt werden aber die den Codeplätzen entsprechenden Zeichen des eingestellten Fonts `cmr10`. Das Leerzeichen auf Position 32 ist in `cmr10` mit dem Glyph `(-)` besetzt, wie in Bild 10.1 zu sehen. ASCII und PostScript sind für die ASCII-Zeichen identisch. Abhilfe: Man wähle einen PostScript-Font, etwa Helvetica, dann stimmen die Zeichen des ASCII-Codes und des PostScript-Fonts überein. Man setzt den Font, z. B. Helvetica, und die Größe der gewünschten Schrift, z. B. 8 pt skaliert um den Faktor 0.92 mit

```
defaultfont:="phvr8r";
defaultscale:=0.92*8pt/fontsize defaultfont;
```

Im Vorgriff auf Kapitel 13 setzen wir hier noch zwei Variable aus dem Graph-Paket, das wir sehr häufig verwenden werden, die für die Formatierung der automatisch von MP erzeugten Bezifferung von Diagrammen zuständig sind

```
Fmfont_:=defaultfont;
Fmscale_:=defaultscale;
```

Dann werden auch diese Zeichen in der gleichen Schrift und Größe gesetzt. Das MP-Paket `format.mp`, das von dem Paket `graph.mp` automatisch geladen wird, enthält den Befehl `format` zur Formatierung von Zahlen in Festpunkt- oder Gleitpunktdarstellung mit vorgegebener Stellenzahl. Zum Beispiel formatiert

```
format("%2f",x)
```

den Zahlenwert der Variable x als Festpunktzahl mit 2 Nachkommastellen. `format` erzeugt ein Ergebnis vom Typ `picture`.

Die Positionierung mit dem Label-Suffix setzt den Text in einem kleinen Abstand vom angegebenen Ort, der durch den Parameter `labeloffset` (default 3 bp) bestimmt ist. Der Abstand kann durch Wertzuweisung geändert werden.

$$\begin{array}{ccc}
 & \text{ulft} & \text{top} & \text{urt} \\
 (\text{rechtsbündig}) & \text{lft} & \times & \text{rt} & (\text{linksbündig}) \\
 & \text{llft} & \text{bot} & \text{lrt}
 \end{array}$$

Bild 10.2: Ausrichtung eines Labels.

Beispiel: Bild 10.2 zeigt die Positionierungsmöglichkeiten. Zur Verdeutlichung wurde der Parameter `labeloffset` auf den eher ungewöhnlichen Wert von 5 mm gesetzt.

```
beginfig(1)
  interim labeloffset:=5mm;
  label(btex $\times$ etex,origin);
  label.rt(btex rt (linksb\undig) etex,origin);
  label.lft(btex (rechtsb\undig) lft etex,origin);
  label.top(btex top etex,origin);
  label.bot(btex bot etex,origin);
  label.urt(btex urt etex,origin);
  label.ulft(btex ulft etex,origin);
  label.lrt(btex lrt etex,origin);
  label.llft(btex llft etex,origin);
endfig;
```

Der Befehl

```
dotlabel
```

hat die gleiche Syntax wie der Label-Befehl, setzt aber zusätzlich noch einen Punkt an die angegebene Position. Der Durchmesser des Punktes ist mit `dotlabeldiam` auf default 3 bp voreingestellt.

Auch der Befehl

```
thelabel
```

hat die gleiche Syntax, erzeugt aber ein Ergebnis vom Typ `picture`, das als Bild mit `draw` gezeichnet und weiterverarbeitet werden kann. Insbesondere kann man mit `bbox` dessen Boundingbox ermitteln.

Fassen wir zusammen. Strings als Label-Text werden stets in der mit `defaultfont` eingestellten Schrift gesetzt. Die Werte von MP-Variablen können mit `decimal` in den String eingefügt werden. Der String besteht aus einer Folge von ASCII-Zeichen, die in der mit `defaultfont` eingestellten Schrift (default `cmr10`) formatiert werden.

10.2 T_EX/L^AT_EX-Label

Unser Ziel ist es natürlich, die Label-Texte mit T_EX und L^AT_EX zu setzen und alle damit verbundenen Formatierungsmöglichkeiten zu nutzen. Dazu schreibt man in der Label-Anweisung den Label-Text in eine btex-etex-Umgebung, die einen Ausdruck vom Typ picture erzeugt.

```
label(btex <labeltext in TeX oder LaTeX> etex,<pair expr>);
```

Man beachte das Leerzeichen nach btex und vor etex. Außerdem muss der gesamte btex-etex-Ausdruck auf eine Zeile ohne Zeilenumbrüche geschrieben werden. Wenn der Label-Text nur T_EX (kein L^AT_EX!) enthält, sind keine weiteren Maßnahmen erforderlich. Der T_EX-Code wird in Computer Modern formatiert. Andere Schriften können zu Beginn des Quellfiles innerhalb von verbatimex ... etex als T_EX-Präambel nach T_EX-Art definiert werden.

Beispiel: T_EX-Text in Helvetica.

```
verbatimex
\font\myfont=phvr at 8pt
etex;
beginfig(1)
label(btex \myfont Durchmesser etex,origin);
endfig;
```

Die Angaben innerhalb von verbatimex ... etex betreffen nur die Formatierung des T_EX-Codes in btex-etex-Umgebungen, nicht jedoch String-Label-Texte in den anderen Label-Kommandos.

Label-Texte in L^AT_EX werden auch innerhalb der btex-etex-Umgebung im Label-Befehl gesetzt, aber nun sind eine L^AT_EX-Präambel und eine L^AT_EX-Postambel vor end zwingend erforderlich, die MP mitteilen, wie der L^AT_EX-Code formatiert werden soll (Class-File, Class-Options, Font, Fontsize, Packages,...)

L^AT_EX-Präambel:

```
verbatimex
\documentclass{article}
\usepackage[T1]{fontenc}
\usepackage[scaled=0.92]{helvet}
\begin{document}
\sffamily
\footnotesize
etex;
```

L^AT_EX-Postambel:

```
verbatimex
\end{document}
etex;
```


Des weiteren muss man dem MP-Interpreter mitteilen, dass nun L^AT_EX zur Anwendung kommen soll. Dafür gibt es drei Möglichkeiten.

1. Man setzt die Umgebungsvariable TEX auf der Kommandozeile mit

```
export TEX=latex
```

2. Man ruft mpost mit der Option -tex auf

```
mpost -tex=latex <mpfile>
```

3. Man fügt in die L^AT_EX-Präambel unmittelbar nach verbatimtex die Direktive

```
%&latex
```

ein, ohne die Umgebungsvariable TEX zu setzen. Sollte TEX=tex sein, gibt es eine Fehlermeldung. Abhilfe: Setzen Sie TEX auf undefiniert gemäß

```
export TEX=
```

Enthält das MP-Quellfile keine btex-etex-Umgebungen oder nur solche mit reinem T_EX-Code (kein L^AT_EX!), ist keine L^AT_EX-Präambel/Postambel erforderlich, die Umgebungsvariable ist irrelevant, darf aber nicht auf TEX=latex stehen (Widerspruch!) und beim Aufruf von MP braucht die Option -tex=. . . nicht angegeben zu werden, da MP tex als default-Wert verwendet.

Als Nachteil bleibt allerdings, dass innerhalb der btex-etex-Umgebung keine MP-Variablen eingefügt werden können, um deren Zahlenwerte zu setzen. Das geht nur bei String als Label-Text. btex-etex ist aber ein Picture. Der Labeltext kann jedoch nur ein String **oder** ein Picture sein, keine Mischung von beiden. Wie man dieses Problem lösen kann, folgt im nächsten Abschnitt.

10.3 Dynamische Beschriftung

Der Begriff dynamische Beschriftung rührt daher, dass in Schleifen (for-Kontrollstrukturen) der L^AT_EX-Code in einer btex-etex-Umgebung nur beim ersten Schleifendurchlauf ausgewertet und formatiert wird. Die Beschriftung kann also nicht pro Durchlauf modifiziert werden. Letztlich geht es darum, den Zahlenwert einer MP-Variablen in einen T_EX- oder L^AT_EX-Text einzufügen.

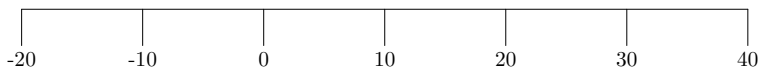


Bild 10.3: Dynamische Beschriftung als String, in den MP-Variable mit decimal eingefügt werden.

Beispiel: Wir zeichnen ein Thermometer in Grad Celsius, wie in Bild 10.3 gezeigt.

```
beginfig(1)
```

```

ut:=1mm;
a:=2ut; d:=6ut;
draw (-20a,0)--(40a,0);
for i=-2 upto 4: % *10 Grad Celsius
  draw ((0,0)--(0,-d)) shifted (i*10a,0);
  label.bot(decimal(10i),(i*10a,-d));
endfor
endfig;

```

Soweit so gut, nun sollte aber an den Zahlen noch ein 'hochgestelltes Gradzeichen' und 'C' für Celsius in \TeX/L\TeX formatiert erscheinen

```
btex \,$^\circ$C etex
```

Leider ist es aber nicht möglich, dieses Picture und den String in den Label-Befehl gemeinsam einzufügen.

```
label.bot(btex decimal(10i)\,$^\circ$C etex,(i*10a,-d));
```

wäre unser Wunsch, ist aber nicht zulässig und führt zu einer Fehlermeldung!

10.3.1 Das Paket TEX.mp

MP bietet für diesen Fall eine Lösung mit dem MP-Paket TEX.mp an, die aber extrem langsam und umständlich ist und ursprünglich auf \TeX -Text (kein \LaTeX !) beschränkt war. Ab Version 0.9 wurde es auf \LaTeX erweitert durch zwei Umgebungsvariable TEXPRE und TEXPOST, in die man die \LaTeX -Prä- bzw. Postambel eintragen muss. Die Präambel im Quellfile ist dabei irrelevant. Im Quellfile muss zuerst das MP-Paket TEX.mp mit input geladen werden, dann folgt die Besetzung der Umgebungsvariablen TEXPRE und TEXPOST mit den gleichen Angaben wie in den verbatimtex-etex-Umgebungen für die \LaTeX -Prä- und Postambel, jedoch ohne die Klammerung mit verbatimtex und etex und alles auf eine Zeile, Zeilenumbrüche allenfalls nach den &-Zeichen. Damit die Direktive auf einer gesonderten Zeile steht, wird char(10) für das ASCII-Steuerzeichen NL (new line) eingefügt.

Im Label-Befehl wird der Label-Text ersetzt durch den Operator TEX(<string>), dessen Argument ein String ist, der aus einer Verkettung von \LaTeX -Code und String-Ausdrücken besteht.

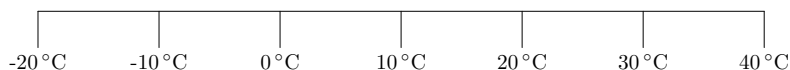


Bild 10.4: Dynamische Beschriftung mit \TeX -Text und dem Paket TEX.mp.

Beispiel: Thermometer in Grad Celsius mit \TeX und TEX.mp (Bild 10.4).

```

input TEX;
beginfig(1)
  ut:=1mm; a:=2ut; d:=6ut;
  draw (-20a,0)--(40a,0);

```

```

for i=-2 upto 4: % 10*Grad Celsius
  draw ((0,0)--(0,-d)) shifted (i*10a,0);
  label.bot(TEX("&decimal(10i)&"$,^{\circ}$C"),(i*10a,-d));
endfor
endfig;

```

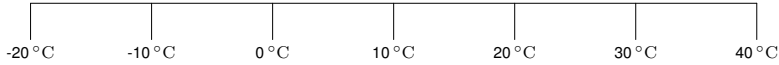


Bild 10.5: Dynamische Beschriftung mit L^AT_EX-Text und dem Paket TEX.mp unter Verwendung der Umgebungsvariablen TEXPRE und TEXPOST.

Beispiel: Thermometer in Grad Celsius mit L^AT_EX und TEX.mp (Bild 10.4).

```

input TEX;
TEXPRE("%&latex"&char(10)&"\documentclass{article}"&
"%\usepackage[T1]{fontenc}\usepackage[scaled=0.92]{helvet}"&
"%\begin{document}\sffamily\footnotesize");
TEXPOST("\end{document}");
beginfig(1)
  ut:=1mm; a:=2ut; d:=6ut;
  draw (-20a,0)--(40a,0);
  for i=-2 upto 4: % 10*Grad Celsius
    draw ((0,0)--(0,-d)) shifted (i*10a,0);
    label.bot(TEX("&decimal(10i)&"$,^{\circ}\mathrm{C}$"),
      (i*10a,-d));
  endfor
endfig;

```

10.3.2 Lösung nach Alan Hoenig

Eine wesentlich einfachere und schnellere Lösung für das Problem der dynamischen Beschriftung stammt von Alan Hoenig [22]. Man schreibt die eigentlich unzulässige Label-Anweisung mit expliziten Zahlenwerten als String in ein externes File und liest dieses nach Abschluss mit 'End-of-File' nach der Schleife mit input wieder ein.

```

beginfig(1)
  string s[];
  s1:="verbatimtex \documentclass{article}\usepackage[T1]{fontenc}"&
    "\usepackage[scaled=0.92]{helvet}\begin{document}"&
    "\sffamily\footnotesize etex;";
  write s1 to "myfile.mp";
  ut:=1mm; a:=2ut; d:=6ut;
  draw (-20a,0)--(40a,0);
  for i=-2 upto 4: % 10*Grad Celsius
    draw ((0,0)--(0,-d)) shifted (i*10a,0);
    s2:="label.bot(btex "&decimal(10i)&
      "\,\\(^{\circ})C etex,("&decimal(i*10a)&","-d));";

```

```

        write s2 to "myfile.mp";
    endfor
    write EOF to "myfile.mp";
    input "myfile.mp";
endfig;

```

Der String EOF markiert das Dateiende und schließt die Datei. Da der input-Befehl das externe File sofort beim Einlesen mit $\text{T}_{\text{E}}\text{X}/\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ unabhängig von der Präambel im Quellfile bearbeitet, muss man dem externen File die $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Präambel voranstellen. Ansonsten behält das Quellfile seine Standardform und die Methode funktioniert mit $\text{T}_{\text{E}}\text{X}$ und $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ gleichermaßen. Die als String in das externe File geschriebene $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Präambel ist gleich der entsprechenden Präambel im Quellfile, muss aber als String auf einer Zeile stehen, wobei Zeilenwechsel nur unmittelbar nach einem &-Zeichen möglich sind.

Für mehrfache Anwendung dieser Methode kann man stets das gleiche externe File verwenden.

Das Ergebnis ist natürlich identisch wie in Bild 10.5.

10.3.3 Das Paket latexmp

Die Beschriftung mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Text und insbesondere die dynamische Beschriftung werden auch von dem Paket latexmp von Jens-Uwe Morawski [32] unterstützt, das auf CTAN und in $\text{T}_{\text{E}}\text{X}$ Live enthalten ist. Der $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ -Text wird als String übergeben, der mit Strings für MP-Variable verkettet werden kann. Das Verfahren ist sehr flexibel, einfach und schnell und implementiert im wesentlichen den von Alan Hoenig vorgeschlagenen Weg.

```

i = 5
i = 4
i = 3
i = 2
i = 1

```

Bild 10.6: Dynamische Beschriftung mit dem Paket latexmp.

Beispiel: Man erzeuge eine Liste von Werten $i = 1, 2, \dots, 5$ wie in Bild 10.6.

```

input latexmp;
beginfig(1)
  for i=1 upto 5:
    label(texttext("$i="&decimal i&"$"),(0,10*i));
  endfor
endfig;
end

```

texttext("string") ist ein Befehl aus dem latexmp-Paket zur Übergabe des Label-Textes als String. latexmp sammelt alle Strings in einem externen temporären File mit Prefix ltx- und Extension tmp. Um die Label-Texte mit $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$ zu formatieren, muss man das temporäre File wieder einlesen. Dies erfolgt automatisch in einer zweiten MP-

Bearbeitung. Zur Anzeige verwenden wir wieder wie üblich ein \LaTeX -Dummy-File. Der Workflow sieht dann wie folgt aus

```
mpost <mpfile>
mpost <mpfile>
latex <dummy file>
dvips <dummy file>
gv <dummy file>.ps
```

Wesentlich ist dabei, dass der mpost-Befehl **zweimal** ausgeführt werden muss. Das Ergebnis zeigt Bild 10.6.

Die latexmp-Präambel Ohne weitere Angaben, wie in unserem ersten Beispiel, wird als Voreinstellung die article-Klasse ohne Klassenoptionen verwendet. Man kann aber auch mit dem Befehl `setupLaTeXMP("<key=value-Liste>")` des latexmp-Pakets die benötigte Klasse, Optionen, Pakete, Kommandos und Einstellungen einzeln angeben, oder die Präambel in einem File, z. B. `latexmp_preamble.tex` zusammenfassen und als Ganzes übergeben gemäß

```
setupLaTeXMP(preamblefile="latexmp_preamble");
```

mit dem folgenden Inhalt für das File `latexmp_preamble.tex`

```
\documentclass[12pt]{report}
\usepackage[T1]{fontenc}
\usepackage[scaled=0.92]{helvet}
```

Alternativ lauten die Einzelangaben dazu

```
setupLaTeXMP("class="report",options="12pt",
fontencoding="T1",
packages="helvet[scaled=0.92]");
```

Pakete, die 'TeX specials' benutzen wie `graphicx`, `color`, ... erfordern eine spezielle Behandlung wie im folgenden gezeigt wird.

Als weitere Schlüsselwörter stehen zur Verfügung (die angegebenen Werte sind beispielhaft):

```
inputencoding="latin1"
language="ngerman"
preamble="\newcommand{\IT}[1]{\textit{#1}}"
mode=normal
```

(`mode=rerun` wird nicht empfohlen, weil die Pakete `MetaFun`, `MetaObj` und insbesondere `graph` nicht im `rerun`-Mode verwendet werden können und Gleichungen in MP wegen Inkonsistenz zu Fehlermeldungen führen.)

String-Label Damit man die String-Label direkt eingeben und bearbeiten kann ohne das Interface `textext()`, setzt man das Schlüsselwort

```
textextlabel=enable
```

Diese Einstellung gilt dann für das ganze Quellfile.

Farben Da das color-Paket nicht direkt verwendet werden kann, gibt es das Schlüsselwort `multicolor`. Mit

```
multicolor=enable
```

stehen die Befehle `\color` und `\textcolor` des color-Pakets für Label-Texte innerhalb von `texttext()` zur Verfügung. Unterstützt werden die Farbmodelle `rgb`, `cmk`, `gray`, `Hsv` und `named`, wobei alle Farben nach `rgb` konvertiert werden, weil MP nur das `rgb`-Farbmodell unterstützt.

The quick brown fox jumps over the lazy dog.

Bild 10.7: Verwendung von Farben mit dem Paket `latexmp`.

Beispiel: Farben mit `latexmp`.

```
input latexmp;
beginfig(1)
  setupLaTeXMP(texttextlabel=enable,multicolor=enable);
  color Qcolor, Lcolor;
  Qcolor:=(0.8,0.4,0.7);
  Lcolor:=(0.2,0.9,0.2);
  label("\color{blue}The \textcolor{Qcolor}{quick}" &
    " \textcolor[rgb]{0.6,0.4,0}{brown fox}" &
    " jumps over \textcolor[Hsv]{30,0.5,0.9}{the}" &
    " \textcolor{Lcolor}{lazy} dog.",origin);
endfig;
```

Graph Das folgende Beispiel zeigt die Anwendung von `latexmp` zusammen mit dem `Graph`-Paket.

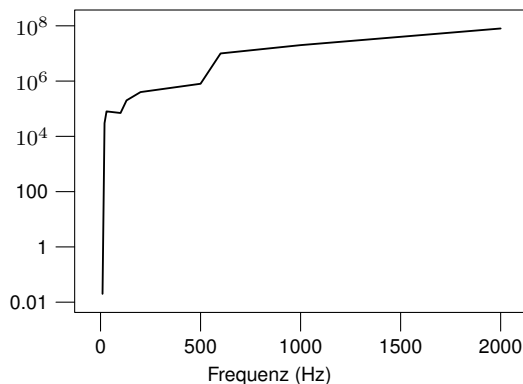


Bild 10.8: Die Verwendung von `graph` mit dem Paket `latexmp`.

Beispiel: Man zeichne unter Verwendung des Pakets `graph.mp` in Kapitel 13 das Diagramm in Bild 10.8.

```

input latexmp;
input graph;
setupLaTeXMP(mode=normal,
packages="helvet[scaled=0.92]",
preamble="\AtBeginDocument{\sffamily\footnotesize}");
init_numbers(texttext("$-$"),texttext("$1$"),texttext("${\times 10}$"),
texttext("${}^-$"),texttext("${}^2$"));
Fe_base:=texttext("$10$");
defaultfont:="phvr8r";
defaultscale:=0.92*8pt/fontsize defaultfont;
Fmfont_:=defaultfont;
Fmscale_:=defaultscale;
beginfig(1)
  draw begingraph(6cm,4cm)
    setcoords(linear,log);
    glabel.bot(texttext("Frequenz (Hz)"),OUT);
    gdraw "lamp.dat" withpen pencircle scaled 0.75bp;
  endgraph;
endfig;

```

Man beachte die zusätzlichen Zeilen `init_numbers`, `Fe_base` und den mit `texttext` modifizierten Befehl `glabel`.

10.4 Platzierung von L^AT_EX-Text

Die Label-Befehle zur Beschriftung wird man im allgemeinen erst nach dem Zeichnen der Grafik angeben, um Linien und Farbflächen im Bereich der Beschriftung mit dem Befehl `unfill` zu entfernen und einen entsprechenden Freiraum zu schaffen, in den die Schrift platziert werden soll.



Bild 10.9: Text in Freiraum platzieren.

Beispiel: Man schaffe Freiraum um die Beschriftung und platziere sie an der vorgegebenen Stelle z_0 zentriert, wie in Bild 10.9 gezeigt.

```

beginfig(1)
  interim labeloffset:=3bp;
  interim bboxmargin:=2bp;
  interim truecorners:=0;
  path p[]; picture lab[];
  z0=origin;

```

```

p1:=unitsquare scaled 20mm shifted (z0-(10mm,10mm));
fill p1 withcolor 0.6;
lab1:=thelabel(btex {\Huge A} etex,z0);
unfill bbox lab1;
draw lab1;
endfig;

```

Bei der Platzierung des L^AT_EX-Textes einer btex-etex-Umgebung ist zu beachten, dass MP für den Text die gleiche Box wählt wie L^AT_EX. Die Nulllinie ist die Basislinie der Schrift, der Nullpunkt liegt am Beginn der Schrift. Die Höhe wird durch die Großbuchstaben, die Tiefe durch die Unterlängen und die Breite durch die Textlänge bestimmt, sodass Boxhöhe = Höhe + Tiefe ist.

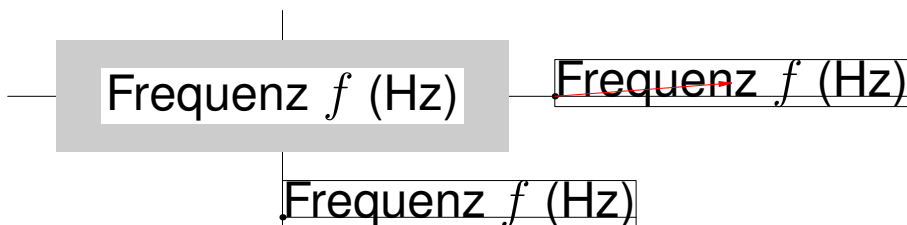


Bild 10.10: Wie L^AT_EX einen Text in der umschließenden Box anordnet.

In Bild 10.10 markiert das Fadenkreuz den Nullpunkt der Zeichnung. Rechts und darunter ist der gleiche Text mit seiner L^AT_EX-Box angegeben, deren Nullpunkt durch einen dicken Punkt gekennzeichnet ist. Ohne die darstellungsbedingte horizontale und vertikale Verschiebung stimmen die Nullpunkte der Zeichnung und der L^AT_EX-Box überein. Der Vektor `center btex <LaTeX-Text> etex` zeigt also von dem dicken Punkt (Nullpunkt der L^AT_EX-Box) zur Mitte der Boundingbox und ist in der Abbildung durch einen Pfeil dargestellt. Um den Mittelpunkt des Textes mit dem Nullpunkt der Zeichnung (origin) zur Deckung zu bringen, muss man das Picture `lab1` um `(-center lab1)` nach links unten verschieben. Der MP-Code zu obigem Bild lautet dann

```

beginfig(1)
  path p[]; picture lab[];
  z0=origin;
  lab1:=btex {\Huge Frequenz $f$ (Hz)} etex;
  fill bbox lab1 shifted (-center lab1) xscaled 1.25 yscaled 2
    shifted z0 withcolor 0.8;
  unfill bbox lab1 shifted (z0-center lab1);
  draw lab1 shifted (z0-center lab1);
endfig;

```

Einfacher ist natürlich die Verwendung des `thelabel`-Befehls, wie im Beispiel zu Bild 10.9 gezeigt, der für die nötige Zentrierung sorgt.

Die Boundingbox ist um einen kleinen Abstand `bbboxmargin` größer als das Bild. Der Wert ist auf default 2 bp voreingestellt und kann mit


```
bboxmargin:=0;
```

auf Wunsch auch unterdrückt werden.

Außerdem kann L^AT_EX-Text auch Freiraum aufweisen, der durch unsichtbare Konstrukte wie `\strut`, `\rule[]{0pt}{}`, etc. entsteht. Um die Boundingbox strikt auf den sichtbaren Text zu beschränken, kann man die interne Variable

```
truecorners:=1;
```

setzen. Der default-Wert ist 0.

Mit `thelabel` kann man auch von einem String-Text die Boundingbox ermitteln und mit `unfill` entsprechenden Freiraum schaffen, während der Befehl `bbox` auf einen String nicht anwendbar ist.

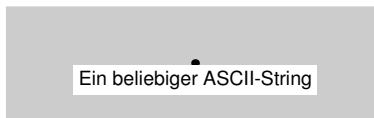


Bild 10.11: Setzen eines beliebigen ASCII-Strings (PostScript-Text) als Picture mit dem `thelabel`-Befehl.

Beispiel: PostScript-Text als String mit `thelabel` setzen, wie in Bild 10.11 gezeigt.

```
beginfig(1)
  picture lab[];
  fill unitsquare xscaled 5cm yscaled 1.5cm shifted (-2.5cm,-0.75cm)
    withcolor 0.8;
  draw origin FETT;
  lab4:=thelabel.bot("Ein beliebiger ASCII-String",origin);
  unfill bbox lab4;
  draw lab4;
endfig;
```

Beispiel: Man platziere das Picture `lab1` (grau) in die linke obere Ecke des Boxpfads `p1` (Bild 10.12).

```
1 beginfig(1)
2   picture lab[]; path p[];
3   ut:=1mm;
4   a:=50ut; b:=50ut; z0=(30ut,-20ut);
5   lab1:=btex \begin{tabular}{l}{\bfseries Kriterien:}\\$n$ gerade/
        ungerade\\Fall a, b, c\\Technologie\end{tabular} etex;
6   p1:=unitsquare xscaled a yscaled b shifted z0;
7   draw p1;
8   draw lab1 withcolor 0.6;
9   draw bbox lab1 withcolor0.6;
10  label.lrt(lab1,point 3 of p1);
11 endfig;
```

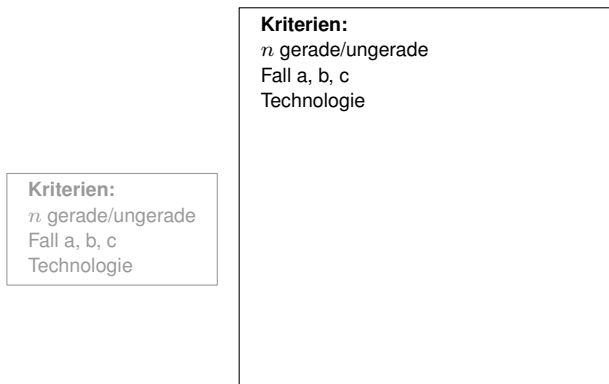


Bild 10.12: Einen Label in eine Box einfügen.

Bitte beachten Sie, dass die 5. und 6. Zeile auf **einer** Zeile ohne Zeilenumbruch stehen müssen.

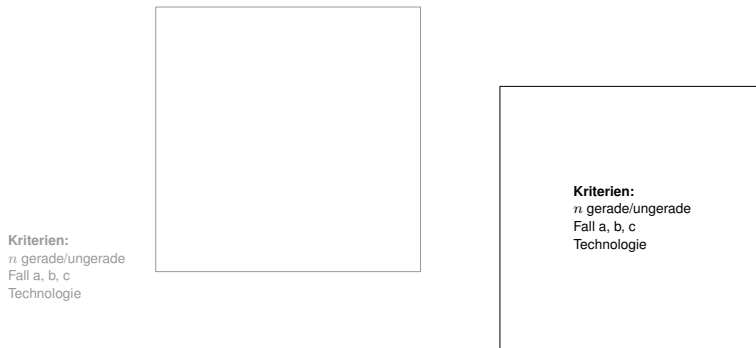


Bild 10.13: Einen Label zentrisch in eine Box einfügen und an einem vorgegebenen Ort platzieren.

Beispiel: Allgemeiner Fall (Bild 10.13). Gegeben sei eine Box als Pfad p_1 und ein Label als Picture lab_1 in beliebiger Lage (grau). Wir wählen zwei Punkte c_1 und c_2 auf dem Rand oder im Zentrum der Box bzw. der Boundingbox von lab_1 und fordern, dass diese Punkte auf z_1 bzw. z_2 zu liegen kommen. Dann muss man die Box um $shift_1 = (z_1 - c_1)$ und das Picture um $shift_2 = (z_2 - c_2)$ verschieben.

```

1 beginfig(1)
2   picture lab[]; path p[];
3   ut:=1mm;
4   a:=50ut; b:=50ut; z1=(120ut,10ut); z2=z1;
5   lab1:=btex \begin{tabular}{l}{\bfseries Kriterien:}\\$n$ gerade/
      ungerade\\Fall a, b, c\\Technologie\end{tabular} etex;
6   p1:=unitsquare xscaled a yscaled b shifted(30ut,0);
7   draw p1 shifted(z1-center p1);

```

```

8  draw lab1 shifted(z2-center lab1);
9  draw p1 withcolor 0.6;
10 draw lab1 withcolor 0.6;
11 endfig;

```

Auch hier muss die 5. und 6. Zeile auf einer Zeile stehen.

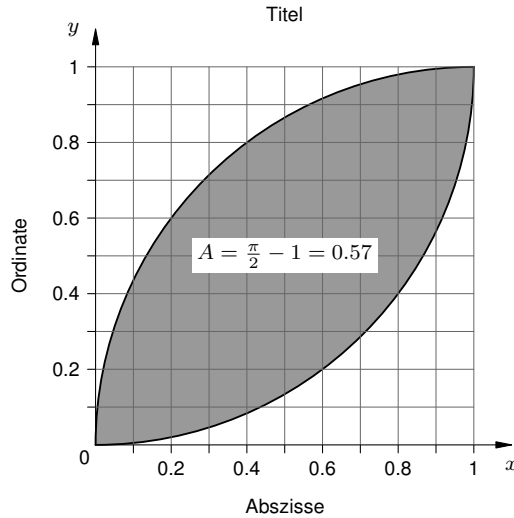


Bild 10.14: Koordinatensystem mit Diagramm und dynamischer Beschriftung mit Formel und MP-Variable nach Hoenig.

Beispiel: Man zeichne ein Koordinatensystem mit Kurve und dynamischer Beschriftung der Achsen nach Hoenig, wie in Bild 10.14 gezeigt.

```

beginfig(1)
  path p[]; picture lab[]; string s;
  ut:=1mm;
  a:=50ut; d:=3bp; Dx:=d+5mm; Dy:=d+7mm; pi:=3.14159;
  %
  p3:=quartercircle scaled 2a rotated 90 shifted (a,0);    % Kurve
  p4:=quartercircle scaled 2a rotated (-90) shifted (0,a);
  p5:=buildcycle(p3,p4);
  fill p5 withcolor 0.6;
  draw p3 fett;
  draw p4 fett;
  %
  p0:=origin -- (a,0);                                     % Achsen
  p1:=origin -- (1.1a,0);
  p2:=origin -- (d,0);                                     % Tick
  drawarrow p1;                                             % x-Achse
  drawarrow p1 rotated 90;                                  % y-Achse
  for i=1 upto 10:
    draw p0 shifted (0,i*0.1a) mager withcolor 0.4;      % Gitter

```

```

draw p0 rotated 90 shifted (i*0.1a,0) mager withcolor 0.4;
draw p2 rotated (-90) shifted (i*0.1a,0);           % Ticks
draw p2 rotated 90 shifted (i*0.1a,0) rotated 90;
if i mod 2 =0:
    label.bot(format("%2f",0.1i),(i*0.1a,-d));      % Bezifferung
    label.lft(format("%2f",0.1i),(-d,i*0.1a));
fi
endfor
label.llft("0",origin);                             % Beschriftung
label.bot(btex  $x$  etex,(1.1a,-d));
label.lft(btex  $y$  etex,(-d,1.1a));
label.bot(btex Abszisse etex,(a/2,-Dx));
label.lft(btex Ordinate etex rotated 90,(-Dy,a/2));
label.top(btex Titel etex,(a/2,a+5mm));             % Titel
                                                    % Kurve beschriften
A:=pi/2-1;
s:="verbatimtex\documentclass{article}\usepackage[T1]{fontenc}"&
    "\usepackage[scaled=0.92]{helvet}\begin{document}"&
    "\sffamily\footnotesize etex;";
write s to "myfile.mp";
s:="lab1:=thelabel(btex  $A=\frac{\pi}{2}-1=$ &
    decimal(round(100A)/100)&"$ etex,center p5);";
write s to "myfile.mp";
write EOF to "myfile.mp";
input "myfile.mp";
unfill bbox lab1;
draw lab1;
endfig;

```

Wir greifen das Thema Beschriftung noch einmal auf in Kapitel 13 bei der Behandlung des MP-Graph-Pakets.