Corrections for slanted stems in METAFONT and METAPOST

Linus Romer

Abstract

Slanting an outline font may change the width and angles of stems. The following article presents some formulae to correct these effects and provides corresponding METAFONT and METAPOST macros.

1 Slanting

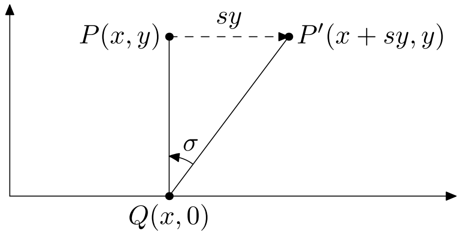
Slanted typefaces are quite common; they are usually called "oblique". E.g., the *URW Gothic L Book* face is slanted forward by an angle of ≈ 10.5° resulting in the *URW Gothic L Book Oblique* face:

Witz *Witz*

Indeed, even the italic faces in *Computer Modern* are designed unslanted, to then be slanted forward by an angle of $\arctan(0.25) \approx 14°$:

Witz *Witz*

In this article, we will assume that slanting means horizontal *shearing*, which is the correct expression in mathematics. The following picture allows us to describe slanting mathematically:



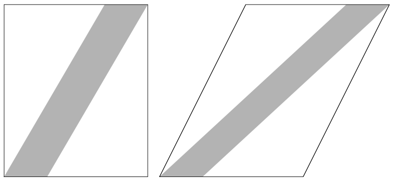
Every coordinate vector $\binom{x}{y}$ is mapped to $\binom{x+sy}{y}$, where s is the slanting amount:

$$\begin{pmatrix} x \\ y \end{pmatrix} \mapsto \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x+sy \\ y \end{pmatrix}$$

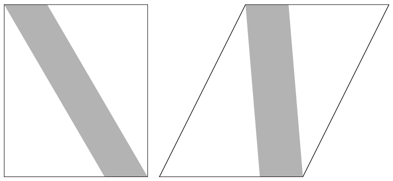
The directed angle $\sigma = \arctan(s)$ denotes the slanting angle. Note that s and σ are negative if and only if the slanting is backward.

2 Width correction for slanted stems

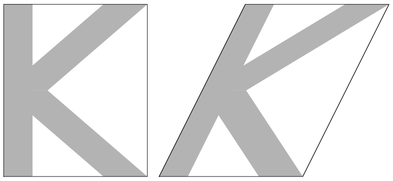
Slanting forward makes forward leaning outline stems slimmer:



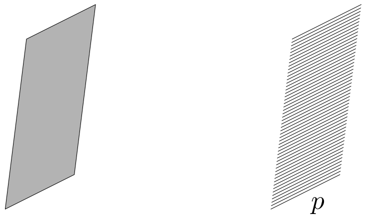
Conversely, slanting forward can make backward leaning stems fatter:



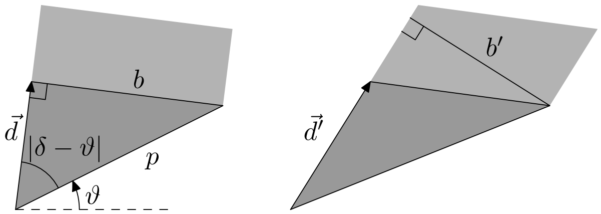
This affects slanted glyphs. E.g., a "K" with even width stems will have stems of different widths after slanting:



Outline stems can be imagined to be drawn by a razor pen of penwidth p.



We want to express the pen width p (before slanting) in terms of the future stem width b' (after slanting), the drawing direction $\vec{d}$ and the pen angle ϑ.



In the figures above, $\vec{d'}$ denotes the slanted vector $\vec{d}$. The angle $\delta = \text{angle}(\vec{d})$ is the directed angle between $\binom{1}{0}$ and $\vec{d}$. The directed pen angle is denoted as ϑ. The distances b and b' are the heights of the nonslanted marked triangle and the slanted marked triangle, respectively. As we know, slanting is an area-preserving transformation. Hence, $|\vec{d}| \cdot b = |\vec{d'}| \cdot b'$ and thus

$$b = b' \cdot \frac{|\vec{d'}|}{|\vec{d}|} = b' \cdot \frac{\left| \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \cdot \vec{d} \right|}{|\vec{d}|} = b' \cdot \left| \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \cdot \frac{\vec{d}}{|\vec{d}|} \right|$$

By applying the definition of the sine function on the non-slanted triangle $\sin|\delta - \vartheta| = \frac{b}{p}$, we obtain the solution:

$$p = b' \cdot \left| \frac{\begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \cdot \frac{\vec{d}}{|\vec{d}|}}{\sin\left(\text{angle}(\vec{d}) - \vartheta\right)} \right| \tag{1}$$

This formula has been published for the special case $|\text{angle}(\vec{d}) - \vartheta| = 90°$ as "slant correction formula" by Jackowski, Nowacki, and Strzelczyk, 2000.

Here is equation (1) as a macro in METAFONT and METAPOST:

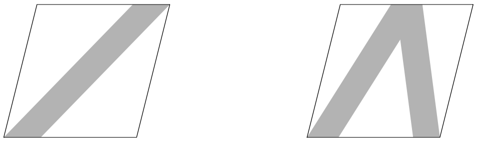
```
def penwidth(expr b,d,theta,s) =
 b*abs(length((d/length(d)) slanted s)
 / sind(angle(d)-theta))
enddef;
```

3 Fitting given boxes

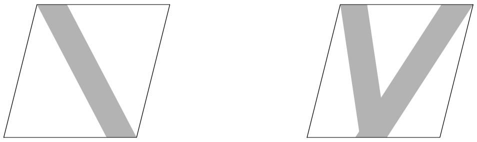
In the following subsections, we will fit diagonal stems in some way into a given rectangular box, such that the diagonal will have the required width b *after* slanting. This is equivalent to the condition requiring that diagonal stems of a required width b fit a given slanted rectangular box (which is a parallelogram).

The following figures visualize the different meanings of "fitting" a stem of width = 10 pt into a slanted box of height = width = 50 pt.

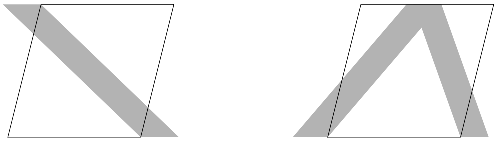
Inscribing (first) diagonal leans forward:



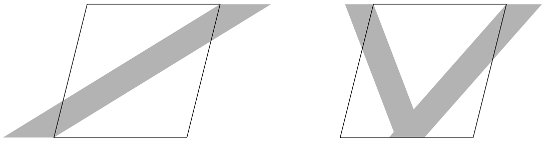
Inscribing (first) diagonal leans backward:



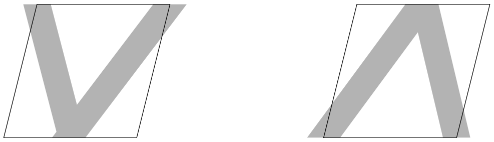
Overlapping (last) diagonal leans backward:



Overlapping (last) diagonal leans forward:



"Half inscribing" chained diagonals:

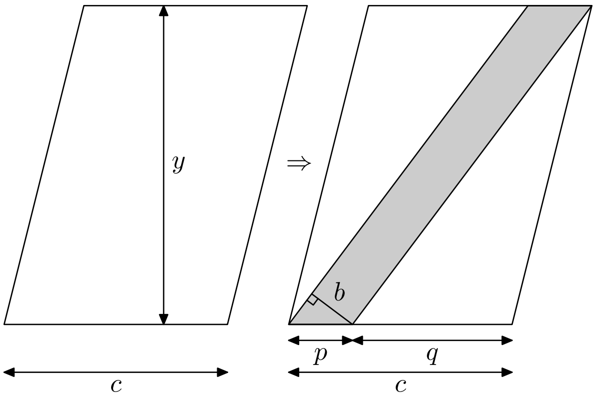


The source code for the preceding figures is given in subsection 3.4.

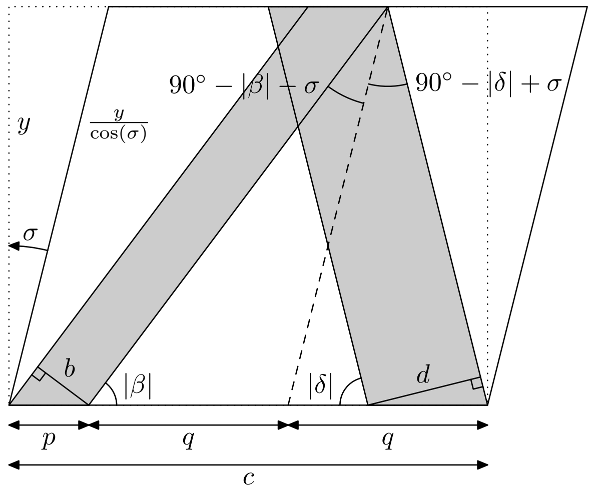
3.1 Inscribing diagonals

3.1.1 First diagonal forward

First, we are trying to *inscribe* a forward diagonal stem of width b into a rectangular box of width c and height y which was slanted by a slanting amount s. We will try to find the penwidth p as depicted below. Jackowski, Nowacki, and Strzelczyk (2000) have already algorithmically solved this problem by defining the binary operator `/\`. Here, we will find an exact solution.



We can generalize the situation by introducing the variable a, which may be toggled between 1 and 2 and which stands for the number of inscribed (chained) diagonals. The situation for $a = 2$ is shown below:



The small rectangular triangle at the left foot leads to the relation

$$p = \frac{b}{\sin|\beta|}.$$

The sine theorem for the left large triangle yields (remember that $\cos(\sigma) > 0$)

$$\frac{q}{\sin\left(90° - (|\beta| + \sigma)\right)} = \frac{y/\cos(\sigma)}{\sin|\beta|}.$$

Hence, we obtain

$$\begin{aligned} q &= \frac{y \sin\bigl(90^\circ - (|\beta| + \sigma)\bigr)}{\cos(\sigma)\sin|\beta|} \\ &= \frac{y\cos(|\beta| + \sigma)}{\cos(\sigma)\sin|\beta|} \\ &= y \cdot \frac{\cos|\beta|\cos(\sigma) - \sin|\beta|\sin(\sigma)}{\cos(\sigma)\sin|\beta|} \\ &= y \cdot \bigl(\cot|\beta| - \tan(\sigma)\bigr) \qquad (2) \\ &= y \cdot \left(\sqrt{1/\sin^2|\beta| - 1} - s\right) \\ &= y \cdot \left(\sqrt{p^2/(b^2-1)} - s\right). \end{aligned}$$

Looking at the total width c, we get

$$c = p + aq = p + ay\left(\sqrt{p^2/(b^2-1)} - s\right).$$

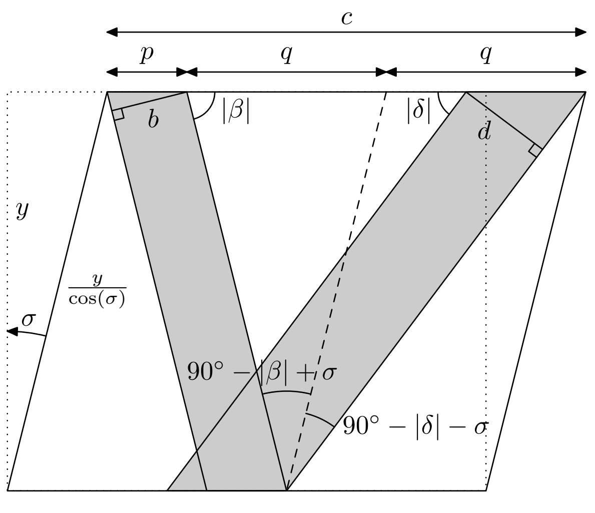
As $p^2/b^2 \geq 1$, this is an ordinary quadratic equation with the solutions

$$p = b \cdot \frac{-b(c + asy) \pm ay\sqrt{(c + asy)^2 + a^2y^2 - b^2}}{a^2y^2 - b^2}.$$

Because $-b(c + asy) \leq 0$, we are left with the only non-negative solution:

$$p = b \cdot \frac{-b(c + asy) + ay\sqrt{(c + asy)^2 + a^2y^2 - b^2}}{a^2y^2 - b^2} \qquad (3)$$

3.1.2 First diagonal backward



Again

$$p = \frac{b}{\sin|\beta|}.$$

The sine theorem for the left large triangle yields

$$\frac{q}{\sin\bigl(90^\circ - (|\beta| - \sigma)\bigr)} = \frac{y/\cos(\sigma)}{\sin|\beta|}.$$

Analogously as in subsection 3.1.1, we obtain

$$q = y \cdot \left(\sqrt{p^2/(b^2-1)} + s\right).$$

The equation

$$c = p + aq = p + ay \cdot \left(\sqrt{p^2/(b^2-1)} + s\right)$$

then leads to the solution

$$p = b \cdot \frac{-b(c - asy) + ay\sqrt{(c - asy)^2 + a^2y^2 - b^2}}{a^2y^2 - b^2}. \qquad (4)$$

3.1.3 Generalization

After introducing the boolean variable e:

$$e = \begin{cases} -1 & \text{if first stem is leaning backward} \\ 1 & \text{otherwise} \end{cases}$$

we can generalize equations (3) and (4) to

$$p = b \cdot \frac{-b(c + aesy) + ay\sqrt{(c + aesy)^2 + a^2y^2 - b^2}}{a^2y^2 - b^2}$$

Replacing the b's and c's by $b_b = \frac{b}{y}$ and $c_c = \frac{c}{y}$ (avoiding arithmetic overflow) yields

$$p = b \cdot \frac{-b_b(c_c + aes) + a\sqrt{(c_c + aes)^2 + a^2 - b_b^2}}{a^2 - b_b^2} \qquad (5)$$

If we calculate the diagonal ratio p/b with equation (5) for the slanting amount $s = 0$, we get the *Computer Modern* macro diag_ratio (Knuth, 2002):

$$\text{diag\_ratio}(a, b, y, c) = b \cdot \frac{-b_b c_c + a\sqrt{c_c + a^2 - b_b^2}}{a^2 - b_b^2}$$

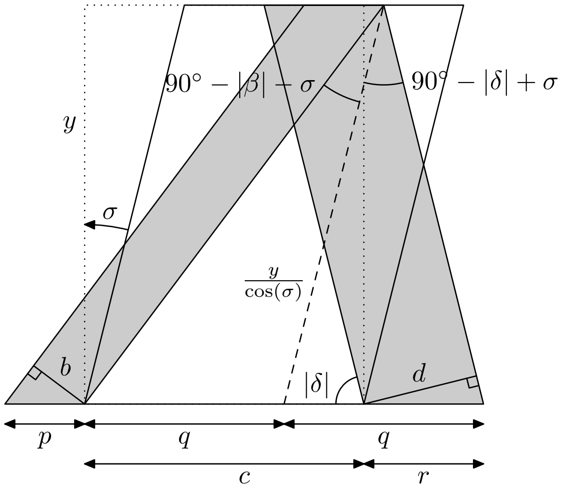
Thus, *Computer Modern* letters like A, V, Λ are computed without slanting corrections. Because of the unequal stem widths, this is not visible for the serif faces. However, the sans faces have uneven stem widths after slanting:

Equation (5) as a METAFONT/METAPOST macro:

```
vardef penwidthin(expr a,e,b,c,y,s) =
 numeric bb,cc; bb=b/y; cc=c/y;
 b*(-bb*(cc+a*e*s)+a*sqrt((cc+a*e*s)**2
 +a*a-bb*bb))/(a*a-bb*bb) enddef;
```

3.2 Overlapping diagonals

3.2.1 Last diagonal backward



The situation is almost the same as in subsection 3.1.1, the only difference being that the diagonals start outside the box and that we are looking for r:

$$r = \frac{d}{\sin|\delta|}$$

The sine theorem for the right large triangle yields

$$\frac{q}{\sin\bigl(90° - (|\delta| - \sigma)\bigr)} = \frac{y/\cos(\sigma)}{\sin|\delta|}.$$

Analogous to subsection 3.1.1, we obtain

$$\begin{aligned} q &= y \cdot \bigl(\cot|\delta| + \tan(\sigma)\bigr) \qquad (6)\\ &= y \cdot \left(\sqrt{p^2/(b^2-1)} + s\right). \end{aligned}$$

Again, we write the total width as

$$c = a \cdot q - r = ay \cdot \left(\sqrt{r^2/(d^2-1)} + s\right) - r.$$

As long as $ay > d$, the only nonnegative solution is

$$r = d \cdot \frac{d(c - asy) + ay\sqrt{(c-asy)^2 + a^2y^2 - d^2}}{a^2y^2 - d^2}. \qquad (7)$$

3.2.2 Last diagonal forward

Analogous to the difference between subsection 3.1.2 and 3.1.1, this subsection differs from subsection 3.2.2 in changing $(90° - |\delta| + \sigma)$ to $(90° - |\delta| - \sigma)$. Hence, it suffices to substitute s by $-s$ in equation (7):

$$r = d \cdot \frac{d(c + asy) + ay\sqrt{(c+asy)^2 + a^2y^2 - d^2}}{a^2y^2 - d^2} \qquad (8)$$

3.2.3 Generalization

As in subsection 3.1.3, we are able to generalize the equations (7) and (8) after introducing

$$e = \begin{cases} -1 & \text{if last stem is leaning backward} \\ 1 & \text{otherwise} \end{cases}$$

to

$$r = d \cdot \frac{d(c + aesy) + ay\sqrt{(c+aesy)^2 + a^2y^2 - d^2}}{a^2y^2 - d^2}$$

Replacing the d's and c's by $d_d = \frac{d}{y}$ and $c_c = \frac{c}{y}$ (avoiding arithmetic overflow) yields

$$r = d_d \cdot \frac{d_d(c_c + aes) + a\sqrt{(c_c+aes)^2 + a^2 - d_d^2}}{a^2 - d_d^2}. \qquad (9)$$

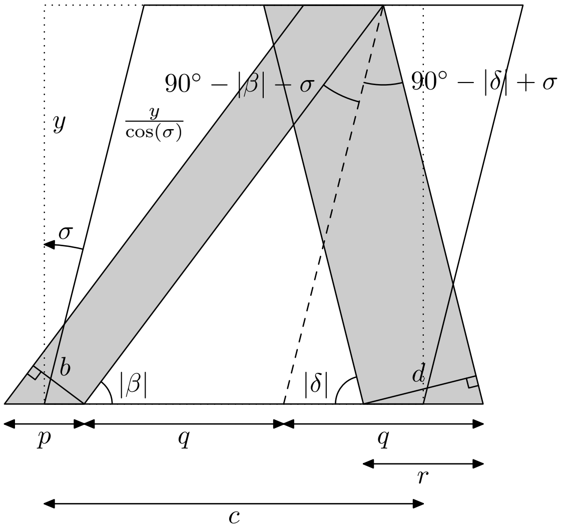
Equation (9) as a METAFONT/METAPOST macro:

```
vardef penwidthover(expr a,e,d,c,y,s) =
 numeric dd,cc; dd=d/y; cc=c/y;
 d*(dd*(cc+a*e*s)+a*sqrt((cc+a*e*s)**2
 +a*a-dd*dd))/(a*a-dd*dd) enddef;
```

Of course, equations (5) and (9) could again be easily united to one single equation.

3.3 Half inscribing diagonals

We are looking at two chained diagonals which fit the given box such that each penwidth is half inside the box. In contrast to the preceding problems, we do not have to consider the case with only one diagonal ($a = 1$) as this case is already covered with equation (1).



If we combine the equations (2) and (6) and include the boolean variable e from subsection 3.1.3, we obtain

$$\begin{aligned} y \cdot \bigl(\cot|\beta| - es\bigr) &= y \cdot \bigl(\cot|\delta| + es\bigr) \\ \Longrightarrow \quad \cot|\delta| &= \cot|\beta| - 2es. \end{aligned}$$

Looking at the total width c, we obtain

$$\begin{aligned} & c = 0.5p + 2q - 0.5r \\ \Longrightarrow\quad & 2c = p + 4q - r \\ \Longrightarrow\quad & 2c = b\sqrt{1+\cot^2|\beta|} + 4y\big(\cot|\beta| - es\big) \\ & \qquad - d\sqrt{1+\big(\cot|\beta| - 2es\big)^2}. \end{aligned}$$

This quartic equation can be solved exactly for $\cot|\beta|$, but the exact solution is long and tedious. For applications, this equation is best solved numerically, e.g., by the bisection method. In the end, we will find both widths p and r by

$$\begin{aligned} p &= b\sqrt{1+\cot^2|\beta|} \\ r &= d\sqrt{1+\big(\cot|\beta| - 2es\big)^2}. \end{aligned}$$

The implementation in METAPOST with the bisection method returns the pair (p, r):

```
vardef poswidthhalf(expr e,b,d,c,y,s) =
 numeric bb,cc,dd,ta,tb,t; % t=cot(beta)
 bb=b/y; dd=d/y; cc=c/y;
 ta=-100; tb=100; % boundaries
 forever:
  exitif abs(ta-tb)<=eps;
  t:=.5[ta,tb];
  if bb*(1++t)+4*(t-e*s)-dd*(1++(t-2*e*s))
   -2*cc>0: tb else: ta fi:=t;
 endfor
 (b*(1++t),d*(1++(t-2*e*s))) enddef;
```

3.4 Source codes for the box fitting figures

In the following, all METAPOST sources of the ten figures at the beginning of section 3 are given as building blocks in a compact form. The most important figures for font design are probably figures 0–3.

```
beginfig(0);
 w:=50pt; h:=50pt; s:=.25; z1l=(0,0); z2r=(w,h);
 z1r-z1l=z2r-z2l=(penwidthin(1,1,10pt,w,h,s),0);
 penstroke z1e--z2e slanted s; endfig;
```

```
beginfig(1);
 w:=50pt; h:=50pt; s:=.25; z1l=(0,0); z4r=(w,0);
 z3r=z2r; y2r=h; x4r-x2r=x2r-x1r;
 z1r-z1l=z2r-z2l=(penwidthin(2,1,10pt,w,h,s),0);
 penstroke z1e--z2e slanted s;
 z3r-z3l=z4r-z4l =(penwidth(10pt,z3r-z4r,0,s),0);
 penstroke (z3e--z4e) slanted s; endfig;
```

```
beginfig(2);
 w:=50pt; h:=50pt; s:=.25; z1l=(0,h); z2r=(w,0);
 z1r-z1l=z2r-z2l=(penwidthin(1,-1,10pt,w,h,s),0);
 penstroke (z1e--z2e) slanted s; endfig;
```

```
beginfig(3);
 w:=50pt; h:=50pt; s:=.25; z1l=(0,h); z4r=(w,h);
 z3r=z2r; y2r=0; x4r-x2r=x2r-x1r;
 z1r-z1l=z2r-z2l=(penwidthin(2,-1,10pt,w,h,s),0);
 penstroke (z1e--z2e) slanted s;
 z3r-z3l=z4r-z4l=(penwidth(10pt,z3r-z4r,0,s),0);
 penstroke (z3e--z4e) slanted s; endfig;
```

```
beginfig(4);
 w:=50pt; h:=50pt; s:=.25; z1r=(0,h); z2l=(w,0);
 z1r-z1l=z2r-z2l=(penwidthover(1,1,10pt,w,h,s),0);
 penstroke (z1e--z2e) slanted s; endfig;
```

```
beginfig(5);
 w:=50pt; h:=50pt; s:=.25; z1r=(0,0); z4l=(w,0);
 z3r=z2r; y2r=h; x4r-x2r=x2r-x1r;
 z3r-z3l=z4r-z4l=(penwidthover(2,1,10pt,w,h,s),0);
 penstroke (z3e--z4e) slanted s;
 z1r-z1l=z2r-z2l=(penwidth(10pt,z2r-z1r,0,s),0);
 penstroke (z1e--z2e) slanted s; endfig;
```

```
beginfig(6);
 w:=50pt; h:=50pt; s:=.25; z1r=(0,0); z2l=(w,h);
 z1r-z1l=z2r-z2l
  =(penwidthover(1,-1,10pt,w,h,s),0);
 penstroke (z1e--z2e) slanted s; endfig;
```

```
beginfig(7);
 w:=50pt; h:=50pt; s:=.25; z1r=(0,h); z4l=(w,h);
 z3r=z2r; y2r=0; x4r-x2r=x2r-x1r;
 z3r-z3l=z4r-z4l=(penwidthover(2,-1,10pt,w,h,s),0);
 penstroke (z3e--z4e) slanted s;
 z1r-z1l=z2r-z2l=(penwidth(10pt,z2r-z1r,0,s),0);
 penstroke (z1e--z2e) slanted s; endfig;
```

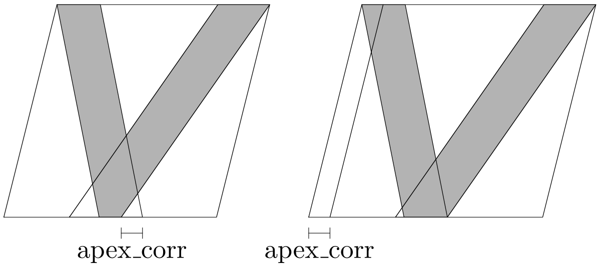
```
beginfig(8); w:=50pt; h:=50pt; s:=.25;
 .5[z1l,z1r]=(0,h); .5[z4l,z4r]=(w,h); z3r=z2r;
 y2l=y2r=y3l=0; y1l=y4l=h; x4r-x2r=x2r-x1r;
 (x1r-x1l,x4r-x4l)=(x2r-x2l,x3r-x3l)
  =poswidthhalf(-1,10pt,10pt,w,h,s);
 penstroke (z1e--z2e) slanted s;
 penstroke (z3e--z4e) slanted s; endfig;
```

```
beginfig(9); w:=50pt; h:=50pt; s:=.25;
 .5[z1l,z1r]=(0,0); .5[z4l,z4r]=(w,0); z3r=z2r;
 y2l=y2r=y3l=h; y1l=y4l=0; x4r-x2r=x2r-x1r;
 (x1r-x1l,x4r-x4l)=(x2r-x2l,x3r-x3l)
  =poswidthhalf(1,10pt,10pt,w,h,s);
 penstroke (z1e--z2e) slanted s;
 penstroke (z3e--z4e) slanted s; endfig;
```

Note: If you use `mfplain.mp` or METAFONT, you do not need to write `slanted s` every time, as this can easily be solved globally.

4 Apex correction

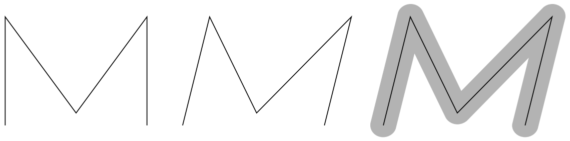
Of course, you generally do not want to join two chained diagonals directly, but you want them to overlap by an amount apex_corr as depicted below on the left:



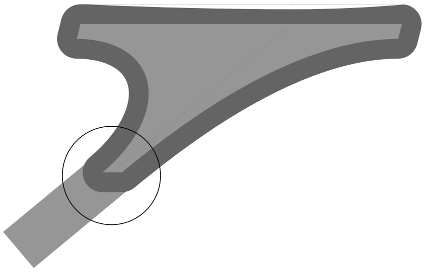
This does not require any new calculation formulae. For calculations, one just needs to add the apex_corr to the box width. In the METAFONT sources of *Computer Modern* (Knuth, 2014), the same trick is applied.

5 Drawn outline borders

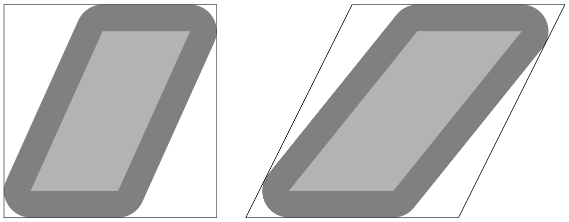
Drawn lines do normally not need slanting corrections, because the paths are slanted first and drawn in the end:



In the italic faces of *Computer Modern*, serifs are not only filled but also stroked. However, the joining stems are sometimes filled only. This creates bumps in letters like *A*, *K* and *X* (Jackowski, Ludwichowski, and Strzelczyk, 2009).



How can we avoid this effect? The formulae presented in section 3 are still valid for an additional circular border pen of width p_b if we use clever substitutions.

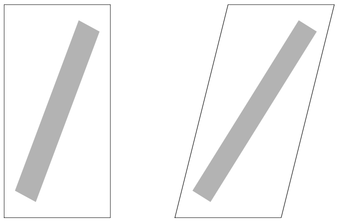


The METAPOST code for the upper right figure is indicated below:

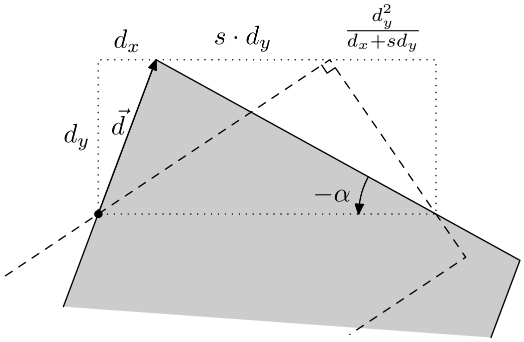
```
beginfig(10); w:=80pt; h:=80pt; s:=.5; pb:=20pt;
pickup pencircle scaled pb slanted s;
lft x1l=0; bot y1l=0; rt x2r=w; top y2r=h;
z1r-z1l=z2r-z2l
 =(penwidthin(1,1,50pt-pb,w-pb,h-pb,s),0);
pickup pencircle scaled pb;
filldraw z1e--z2e slanted s; endfig;
```

6 Orthogonally cut slanted stems

For this section, the pen angle of the stem shall be exactly orthogonal to the stem direction $\vec{d}$ *after* slanting.



We want to find the pen angle α *before* slanting.



In the preceding figure we find

$$\cot(-\alpha) = \left(sd_y + \frac{d_y^2}{d_x + sd_y}\right) : d_y = s + \frac{d_y}{d_x + sd_y},$$

as long as $d_x + sd_y \neq 0$. If $d_x + sd_y = 0$, the stem is vertical after slanting and we have the trivial solution $\alpha = 0$. Therefore,

$$\alpha = \begin{cases} \text{angle}\begin{pmatrix} s + \frac{d_y}{d_x+sd_y} \\ -1 \end{pmatrix} & \text{if } d_x + sd_y \neq 0 \\ 0 & \text{otherwise.} \end{cases} \tag{10}$$

Equation (10) as a METAFONT/METAPOST macro:

```
def angleortho(expr d,s) =
 if xpart(d)+s*ypart(d)>0:
 angle(s+ypart(d)/(xpart(d)+s*ypart(d)),-1)
 else: 0 fi enddef;
```

References

Jackowski, Bogusław, J. B. Ludwichowski, and P. Strzelczyk. “Math fonts: notes from the trenches”. ntg.nl/EuroTeX/2009/slides/jacko-slides.pdf, 2009.

Jackowski, Bogusław, J. Nowacki, and P. Strzelczyk. “Antykwa Półtawskiego: a parameterized outline font”. *MAPS* **25**, 86–102, 2000. ntg.nl/maps/25/13.pdf.

Knuth, Donald E. “The base file for Computer Modern”. ctan.org/tex-archive/fonts/cm/mf/cmbase.mf, 2002.

Knuth, Donald E. “Computer Modern Roman upper case”. ctan.org/tex-archive/fonts/cm/mf/romanu.mf, 2014.

◇ Linus Romer
Oberseestrasse 7
Schmerikon, 8716
Switzerland
linus.romer (at) gmx dot ch