

The Carolingian Minuscul^e fonts^{*}

Peter Wilson
Catholic University of America
Now at `peter.r.wilson@boeing.com`

2002/12/30

Abstract

The `carolmin` and `allcmin` packages provide a set of Metafont Carolingian Minuscul^e book-hands as used for manuscripts in the 8th to the 12th century. This is one in a series of manuscript fonts.

Contents

1	Introduction	2
1.1	Manuscript book-hands	2
1.2	The Carolingian Minuscul ^e script	4
2	The <code>carolmin</code> and <code>allcmin</code> packages	4
2.1	The <code>carolmin</code> package	4
2.2	The <code>allcmin</code> package	4
3	The Metafont code	5
3.1	The parameter file	5
3.2	The driver file	9
3.3	The code for normal letters	28
3.4	The code for capital letters	45
3.5	The punctuation file	62
3.6	The ligatures and dashes file	64
3.7	The digits file	70
4	The font definition files	75
5	The package code	75
5.1	The <code>carolmin</code> package	75
5.2	The <code>allcmin</code> package	76

^{*}This file has version number v1.0, last revised 2002/12/30.

List of Tables

1	The main manuscript book-hands	3
2	The x heights of the CMR and Carolingian Minuscule fonts	77

1 Introduction

The `carolmin` and `allcmin` packages provide a Metafont [Knu92] version of the Carolingian Minuscule manuscript book-hand that was in use between about the eighth and twelfth centuries AD. It is part of a project to provide fonts covering the major manuscript hands between the first century AD and the invention of printing. The principal resources used in this project are listed in the Bibliography.

This manual is typeset according to the conventions of the L^AT_EX DOCSTRIP utility which enables the automatic extraction of the L^AT_EX macro source files [GMS94]. The `docmfp` package is used for documenting the Metafont portions of the distribution [Wil99].

Section 2 describes the usage of the package. Commented code for the fonts is in Sections 3 and 4 and source code for the package is in Section 5.

1.1 Manuscript book-hands

Before the invention of printing all books were written by hand. The book-hands used by the scribes and copyists for the manuscripts changed as time went on. Table 1 lists some of the common book-hands which were used between the 1st and 15th centuries. The later book-hands formed the basis of the fonts used by the early printers, which in turn form the basis of the printing fonts in use today.

The manuscript book-hands were written with a broad nibbed reed or quill pen. Among the distinguishing characteristics of a hand, apart from the actual shape of the letters, are the angle of the pen (which controls the variation between thick and thin strokes) and the height of a letter compared to the width of the nib. The lower the ratio of the letter height to nib width, the more condensed is the script. The scripts also varied in their typical height.

Table 1 gives an ‘average’ x-height for each script, which I obtained by measuring a sample of photographs of manuscripts written in the various hands. About a dozen examples of each book-hand were measured. This figure should not be taken too seriously.

There was not a sharp division between the use of one hand and another. Many manuscripts exhibit a variety of hands in the same document. For example, the scribe writing in an Uncial hand may have used Roman Rustic letters for capitals. Usually the same pen was used for the two different scripts.

Generally speaking, as a hand got older it became more embellished, and therefore took longer to write. As this happened a new hand would appear that was faster, and which would eventually make the earlier one obsolete.

Table 1: The main manuscript book-hands

Name	Century	x-height (mm)	Height (nib widths)	Pen angle	Normalised height
Roman Rustic	1–6	5.7	4–6	45	1.90
Uncial	3–6	4.1	4–5	30	1.37
Half Uncial	3–9	3.2	3–6	20–30	1.07
Artificial Uncial	6–10	4.2	3–6	10	1.40
Insular majuscule	6–9	4.2	5	0–20	1.40
Insular minuscule	6 onward	4.1	5–6	45–70	1.37
Carolingian minuscule	8–12	3.0	3–5	45	1.00
Early Gothic	11–12	3.8	4–6	20–45	1.27
Gothic Textura	13–15	3.9	3–5	30–45	1.30
Gothic Prescius	13 onward	3.3	4–5	45	1.10
Rotunda	13–15	3.2	4–6	30	1.07
Humanist minuscule	14 onward	3.0	4–5	30–40	1.00

Many of the book-hands were single-cased; that is, they did not have an upper- and lower-case as we do nowadays in Western scripts. On the other hand, a script might be majuscule or minuscule. A *majuscule* script is one, like our upper-case, where the letters are drawn between two lines and are of a uniform height with no ascenders or descenders. A *minuscule* script, like our lower-case, is drawn between four lines and has ascenders and descenders.

During the period under consideration arabic numerals were effectively not used. At the beginning they were unknown and even though some knew of them towards the end, the glyphs used for them are not recognisable — to me they look somewhat like cryllic letters — and each locality and time had its own highly individualistic rendering. The general rule when using one of these book-hands is to write all numbers using roman numerals.

The Roman alphabet consisted of 23 capital letters — the J, U and W were absent. The book-hands initially used both a ‘u’ and a ‘v’ interchangeably but by the 10th century the practice had become to use the ‘v’ before a vowel and the ‘u’ otherwise. The letter corresponding to the W sound appeared in England around the 7th century in the form of the runic *wen* character and by about the 11th century the ‘w’ character was generally used. The ‘J’ is the newest letter of all, not appearing until about the mid 16th century.

In the first century punctuation was virtually unknown, and typically would not even be any additional space between individual words, never mind denoting ends of sentences. Sometimes a dot at mid-height would be used as a word separator, or to mark off the end of a paragraph. Effectively a text was a continuous stream of letters. By the time that printing was invented, though, all of our modern punctuation marks were being used.

Among all these manuscript hands, the Carolingian minuscule is the most important as our modern fonts are based on its letter shapes, and it is also at this point in time where the division occurred between the black letter scripts as

used even today in Germany, and the lighter fonts used elsewhere. The Rotunda and Humanist minuscule hands were developed in Italy and were essentially a rediscovery of the Carolingian minuscule. Gutenberg took the Gothic scripts as the model for his types. Later printers, such as Nicholas Jenson of Venice, took the Humanist scripts as their models.

1.2 The Carolingian Minuscule script

Charlemagne (742–814), the Holy Roman Emperor, was naturally anxious to consolidate his empire. Although he was probably illiterate he employed scribes and when he came to the throne there was a wide diversity of bookhands in use. For efficiency he needed to have a more uniform style of writing so that reading mistakes were less likely to happen. To this end he employed the Anglo-Saxon monk Alcuin to be his ‘Minister for Education’. Alcuin produced an elegant, simple and easy-to-write hand that is now called Carolingian Minuscule.

Letters for capitals were taken from contemporary and earlier bookhands, thus helping to preserve them. All the modern punctuation marks were in use and the letter ‘v’ appears for the first time. As well as the ampersand, the most common ligatures were ‘ct’ and ‘st’; also, a special ‘half r’ was used after an ‘o’.

2 The carolmin and allcmin packages

The Carolingian Minuscule font family is called `cmin`. The font is supplied in both OT1 and T1 encodings.

2.1 The carolmin package

This is intended for the occasions when some short pieces of text have to be written in Carolingian Minuscule while the majority of the document is in another font. The normal `baselineskip`s are used.

<code>\cminfamily</code>	The <code>\cminfamily</code> declaration starts typesetting with the Carolingian Minuscule fonts. Use of the Carolingian Minuscule font will continue until either there is another <code>\...family</code> declaration or the current group (e.g., environment) is closed.
<code>\textcmin</code>	The command <code>\textcmin{<text>}</code> will typeset <code><text></code> using the Carolingian Minuscule fonts.

2.2 The allcmin package

This package is for when the entire document will be typeset with the Carolingian Minuscule font. The `baselineskip`s are set to those appropriate to the book-hand.

This is a minimalist package. Apart from declaring Carolingian Minuscule to be the default font and setting the `baselineskip`s appropriately, it makes no other alterations. Vertical spacing before and after section titles and before and after lists, etc., will be too small as the L^AT_EX design assumes a font comparable in size to normal printing fonts, and the book-hand is much taller. To capture more of

the flavour of the time, all numbers should be written using roman numerals. The `romannum` package [Wil99b] can be used so that \LaTeX will typeset the numbers that it generates, like sectioning or caption numbers, using roman numerals instead of arabic digits.

<code>\cmrfamily</code> <code>\textcmr</code> <code>\cmssfamly</code> <code>\textcmss</code> <code>\cmttfamily</code> <code>\textcmtt</code>	<p>The ...family declarations start typesetting with the Computer Modern Roman (<code>\cmrfamily</code>), the Computer Modern Sans (<code>\cmssfamly</code>), and the Computer Modern Typewriter (<code>\cmttfamily</code>) font families. The <code>\textcm.{<text>}</code> commands will typeset <i><text></i> in the corresponding Computer Modern font.</p> <p>The <code>allcmin</code> package automatically loads the <code>carolmin</code> package, so its font commands are available if necessary.</p>
---	---

3 The Metafont code

As previously noted, this work is part of a larger project to provide fonts covering the main manuscript book-hands. As such, one of the aims is to produce a coordinated set of fonts, especially as multiple hands might be used in a single document.

Noting that the hands tend to be somewhat larger than the typical 10pt size (where the x-height is approximately 1.5mm) used for modern books, I have also designed the fonts at a larger than normal size, then applied some non-linear factors when reducing them down to a 10pt size. Further, I have used the height of the Carolingian minuscule as a normalising factor when deciding on the absolute height of any particular script. The x-height of the Carolingian font is made equal to the x-height of the Computer Modern Roman (CMR) font.

Modern fonts are effectively drawn. That is, the outline of the letter is drawn carefully and the center is filled with ink. This is shown to good effect in the Metafont code for the Computer Modern fonts [Knu87]. In contrast, the manuscript letters were inked by single pen strokes in a calligraphic manner. I have tried to repeat this calligraphic style in the Metafont code.

As much as possible I have tried to use parameter values from the Computer Modern Roman (CMR) fonts in order to reduce possible infelicities if the CM and manuscript fonts are used together. However, few of the CMR parameters are applicable to the calligraphic style.

3.1 The parameter file

We deal with the parameter files first, and start by announcing what they are for. The Carolingian Minuscule font comes in three sizes and also as a normal and a bold font.

```

1 <*base7 | base10 | base17 | base7b | base10b | base17b>
2 <base7>%%% CMIN7.MF    Carolingian Minuscule at 7 point design size.
3 <base10>%%% CMIN10.MF  Carolingian Minuscule at 10 point design size.
4 <base17>%%% CMIN17.MF  Carolingian Minuscule at 17 point design size.
5 <base7b>%%% CMINB7.MF   Carolingian Minuscule Bold at 7 point design size.
6 <base10b>%%% CMINB10.MF Carolingian Minuscule Bold at 10 point design size.
```

```

7 <base17b>%%% CMINB17.MF Carolingian Minuscule Bold at 17 point design size.
8 %
Parameters from CMR are used as much as possible. We also make sure that
cmbase is loaded as well as plain Metafont.
9 if unknown cmbase: input cmbase fi
10
11 <base7>font_identifier="CMIN"; font_size 7pt#;
12 <base10>font_identifier="CMIN"; font_size 10pt#;
13 <base17>font_identifier="CMIN"; font_size 17.28pt#;
14 <base7b>font_identifier="CMINB"; font_size 7pt#;
15 <base10b>font_identifier="CMINB"; font_size 10pt#;
16 <base17b>font_identifier="CMINB"; font_size 17.28pt#;
17

jutstretch      The CMR scaling for lowercase serifs wrt 17pt size.
18 <base7 | base7b> jutstretch:=1.19;
19 <base10 | base10b> jutstretch:=1.152;
20 <base17 | base17b> jutstretch:=1.0;

stemstretch     The CMR scaling for lowercase stem widths wrt 17pt size.
21 <base7 | base7b> stemstretch:=1.50;
22 <base10 | base10b> stemstretch:=1.31;
23 <base17 | base17b> stemstretch:=1.0;

caprat          The scaling for 'capitals' wrt 'lowercase'. This is pretty much a guess (the
ratio for CMR10 is 1.59).
24 caprat:=1.5;          % ratio of capital height to minuscule height

cap_jutstretch  The CMR scaling for uppercase serifs wrt 17pt size.
25 <base7 | base7b> cap_jutstretch:=1.3;
26 <base10 | base10b> cap_jutstretch:=1.2;
27 <base17 | base17b> cap_jutstretch:=1.0;

cap_stemstretch The CMR scaling for uppercase stem width wrt 17pt size.
28 <base7 | base7b> cap_stemstretch:=1.45;
29 <base10 | base10b> cap_stemstretch:=1.31;
30 <base17 | base17b> cap_stemstretch:=1.0;

bfudge          Letter width scaling for bold font wrt normal font.
31 <base7 | base10 | base17> bfudge:=1.0;
32 <base7b | base10b | base17b> bfudge:=1.2;

szfudge         Width scaling wrt 17pt letter width.
33 <base7 | base7b> szfudge:=1.18;
34 <base10 | base10b> szfudge:=1.0;
35 <base17 | base17b> szfudge:=1.0;

hstretch        Horizontal stretching factor wrt 17pt size letter width.
36 hstretch:=szfudge*bfudge;

carol_height     The x-height of the Carolingian minuscule font.
37 <base7 | base7b> carol_height#:=108.5/36pt#;
38 <base10 | base10b> carol_height#:=155/36pt#;
39 <base17 | base17b> carol_height#:=268/36pt#;

```

vstretch The height of this font wrt the Carolingian font.

40 **vstretch**:=1.00;
41

x_height The x-height of lower case letters. Scaled from the Carolingian font.

42 **x_height**#:vstretch*carol_height#;

u The unit width. The CMR values are used.

43 **⟨base7⟩ u**#:15.5/36pt#;
44 **⟨base10⟩ u**#:20/36pt#;
45 **⟨base17⟩ u**#:32.5/36pt#;
46 **⟨base7b⟩ u**#:17.9/36pt#;
47 **⟨base10b⟩ u**#:23/36pt#;
48 **⟨base17b⟩ u**#:37/36pt#;

width_adj The majority of these parameters and values are constant. They are for fine
serif_fit adjustments of characters. The CM values are used.

cap_serif_fit 49 **width_adj**#:0pt#; % width adjustment for certain characters
letter_fit 50 **serif_fit**#:0pt#; % extra sidebar near lowercase serifs

51 **⟨base7 | base7b⟩ cap_serif_fit**#:3.5/36pt#; % extra sidebar near uppercase serifs
52 **⟨base10 | base10b⟩ cap_serif_fit**#:5/36pt#; % extra sidebar near uppercase serifs
53 **⟨base17 | base17b⟩ cap_serif_fit**#:8/36pt#; % extra sidebar near uppercase serifs
54 **⟨base7 | base7b⟩ letter_fit**#:0pt#; % extra space added to all sidebars
55 **⟨base10 | base10b⟩ letter_fit**#:0pt#; % extra space added to all sidebars
56 **⟨base17 | base17b⟩ letter_fit**#:−0.1pt#; % extra space added to all sidebars

sb A length used for adjusting the side bearings of characters.

57 **⟨base7 | base7b⟩ sb**#:1/2u#; % unit for side bearings
58 **⟨base10 | base10b⟩ sb**#:1/2u#; % unit for side bearings
59 **⟨base17 | base17b⟩ sb**#:1/2u#; % unit for side bearings
60

cap_height The height of capital letters.

61 **⟨base7 | base7b⟩ cap_height**#:172.2/36pt#; % height of caps
62 **⟨base10 | base10b⟩ cap_height**#:246/36pt#; % height of caps
63 **⟨base17 | base17b⟩ cap_height**#:425/36pt#; % height of caps

thickfudge The reciprocal of the font height in nib widths. Normal font height is 5 nib
widths and the bold font height is 4 nib widths.

64 **⟨base7 | base10 | base17⟩ thickfudge**:=1/5;
65 **⟨base7b | base10b | base17b⟩ thickfudge**:=1/4;

thinfudge The nib sharpness.

66 **thinfudge**:=1/6;

thick The maximum nib width. That is, the width of the thickest line that can be
penned.

67 **thick**:=stemstretch*thickfudge*x_height#; % max pen breadth

thin The nib thinness. That is, the width of the thinnest line that can be penned.

68 **thin**:=thinfudge*thick#;

pangle The pen angle (in degrees).

69 **pangle**:=45; % pen angle

ascratio	The height of lower case ascenders wrt <code>x_height</code> .
70	<code>ascratio:=6/4;</code>
asc_height	The height of lower case ascenders.
71	<code>asc_height#:=ascratio*x_height#;</code>
descratio	The depth of lower case descenders wrt <code>x_height</code> .
72	<code>descratio:=3/4;</code>
desc_depth	The depth of lower case descenders.
73	<code>desc_depth#:=descratio*x_height#;</code>
fig_height	The height of numerals. Make these midway between normal and capital letters.
74	<code>fig_height#:=(0.5[1,caprat])*x_height#;</code>
fig_width	The width of numerals. All numerals are the same width (roughly, an 'o').
75	<code>fig_width#:=hstretch*(x_height#);</code>
body_height	The height of the tallest character.
76	<code>body_height#:=caprat*asc_height#;</code>
body_depth	The depth of the lowest character.
77	<code>body_depth#:=caprat*desc_depth#;</code>
half_height	Half the normal letter height.
78	<code>half_height#:=1/2x_height#; % half the height of x height letters</code>
79	
side	We might be using a square for the character design.
80	<code>side#:=x_height#;</code>
rule_thickness	Thickness of rules (in math symbols). The CMR values are used.
81	<code>⟨base7 base7b⟩ rule_thickness#:=.34pt#;</code>
82	<code>⟨base10 base10b⟩ rule_thickness#:=.4pt#;</code>
83	<code>⟨base17 base17b⟩ rule_thickness#:=.6pt#;</code>
84	
jutfudge	Controls the protrusion of serifs.
85	<code>%%jutfudge:= 3/4;</code>
86	<code>jutfudge:= 1/2;</code>
jut	The lowercase serif protrusion.
87	<code>jut#:=jutstretch*jutfudge*thick#;</code>
srad	<code>srad</code> with $0 < \text{srad} < 1$ controls the serif radius.
88	<code>srad:=0.8; % was 0.5</code>
trijut	The lowercase triangular serif protrusion.
89	<code>trijut#:=0.75jutstretch*thick#;</code>
clubjutfudge	The fractional protrusion of a club serif.
90	<code>clubjutfudge:=1/2;</code>
clubjut	The lowercase club serif protrusion.
91	<code>clubjut#:=jutstretch*clubjutfudge*thick#;</code>


```

o      Amount of overshoot for curves (as in an 'o'). Use CMR values.
92 <base7 | base7b> o#:=5/36pt#;
93 <base10 | base10b> o#:=8/36pt#;
94 <base17 | base17b> o#:=10/36pt#;
95
slant   slant is the amount the font slopes to the right.
        5 degrees = 9/100; 10 degrees = 11/50.
96 slant:=0;      % tilt ratio  $(\Delta x / \Delta y)$ 
97 %%%%slant:= sind 10/cosd 10;  %% 10 degree slant
98
monospace We are generating a variable-width font.
99 monospace:=false;  % should all characters have the same width?
100
dot_size The size of (punctuation) dots.
101 dot_size#:=5thin#;  % size of dots
comma_width The horizontal width of a comma.
102 comma_width#:=5/4dot_size#;
103
        Finally, call the driver file for the font.
104 generate cmintitle      %% switch to the driver file
105
        The end of this code section.
106 </base7 | base10 | base17 | base7b | base10b | base17b>

```

3.2 The driver file

The driver file sets up the font parameters and also contains the macros used in the definition of the characters.

```

107 <*mfd>
108 % This is CMINTITLE.MF for the Carolingian Minusculæ font.
109 % This makes a short font.
110 % This file also contains special macros universally needed in the
111 % font but either not in the cmbase.mf file or modifications of these.
112 %
makebox This is a new version of the cmbase.mf makebox macro. Essentially, as we are
        only generating a single case font, we don't need all the horizontals for the CMR
        characters.
113 % A new version of cmbase.mf makebox macro
114 def makebox(text rule) =
115   for y=0,half_height,cap_height,body_height,x_height,-desc_depth,-body_depth:
116     rule((l,y)t_,(r,y)t_); endfor % horizontals
117   for x=l,r: rule((x,-body_depth)t_,(x,body_height)t_); endfor % verticals
118   for x=u*(1+floor(1/u)) step u until r-1:

```

```

119     rule((x,-body_depth)t_,(x,body_height)t_); endfor          % more verticals
120   if charic<>0:
121     rule((r+charic*pt,h.o_),(r+charic*pt,.5h.o_)); fi % italic correction
122 enddef;
123

```

proofpcbb The macro `proofpcbb(S, xoff, yoff)` draws crosses at the corners of the square, and at the bottom and top midpoints, where the point `zS` is at position `(xoff,yoff)` in the square.

```

124
125 % A macro to delineate a letter's bounding square in proof mode.
126 % Point z$ is at (xoff,yoff) in the bounding square.
127 def proofpcbb(suffix $)(expr xoff, yoff) =
128 if proofing > 0:
129   x$bl = x$ - xoff; y$bl = y$ - yoff;          % Bottom Left Hand Corner
130   z$br = (x$bl+side, y$bl);                    % BRHC
131   z$tr = (x$bl+side, y$bl+side);                % TRHC
132   z$t1 = (x$bl, y$bl+side);                     % TLHC
133   z$bc = 1/2[z$bl,z$br]; z$tc = 1/2[z$t1,z$tr]; % horizontal mid points
134   pair hc, vc;
135   hc = (2u,0); vc = (0,2u);
136   pickup pensquare scaled 2;
137   for s = z$bl, z$br, z$tr, z$t1, z$bc, z$tc:
138     draw s-hc--s+hc;
139     draw s-vc--s+vc;
140   endfor
141 fi
142 enddef;
143

```

carolmin_font_setup This macro contains the set up for the Carolingian Minusculer fonts.

```

144
145 def carolmin_font_setup= % contains special stuff for carolmin fonts
146   define_pixels(side,half_height,carol_height,comma_width);
147   define_pixels(u,dot_size,letter_size,cap_height,width_adj,serif_fit,
148     cap_serif_fit,jut,clubjut,cap_jut,tribut);
149   define_pixels(fig_width);
150   define_whole_pixels(letter_fit,sb);
151   define_whole_vertical_pixels(body_height,asc_height,cap_height,x_height,
152     desc_depth,body_depth,fig_height);
153   define_corrected_pixels(o);
154   define_blacker_pixels(thick,thin);
155   let adjust_fit = normal_adjust_fit;

```

stylus The pen for drawing the thinnest lines is `stylus`.

```

156   pickup pencircle scaled thin;
157   stylus:=savepen;

```

thickstylus The pen for drawing lines a bit thicker than the thinnest is `thickstylus`.

```

158   pickup pencircle scaled (1.5thin);
159   thickstylus:=savepen;

```

```

quill    The normal calligraphic pen is quill.
160  pickup pencircle xscaled thick yscaled thin rotated pangle;
161  quill:=savepen;
162

threequarterquill    A pen with 3/4 of the width as quill (but with the same thinness).
163  pickup pencircle xscaled 3/4thick yscaled thin rotated pangle;
164  threequarterquill := savepen;
165

halfquill    A pen with 1/2 of the width as quill (but with the same thinness).
166  pickup pencircle xscaled 1/2thick yscaled thin rotated pangle;
167  halfquill := savepen;
168

quarterquill    A pen with 1/4 of the width as quill (but with the same thinness).
169  pickup pencircle xscaled 1/4thick yscaled thin rotated pangle;
170  quarterquill := savepen;
171

172  numeric pcshiftx#; % x value of LHS of box wrt origin of char box
173  define_pixels(pcshiftx);
174  numeric sqxside#;    % x size of the square
175  numeric sqyside#;    % y size of the square
176  define_pixels(sqxside,sqyside);
177
178 enddef; % end of carolmin_font_setup
179

ductus    A boolean to control drawing the ductus (TRUE) or the full letter (FALSE).
180 boolean ductus;
181 ductus:=false;

qhor    The cosine and sine of the pen angle.
qvert  182 numeric qhor, qvert;
183 qhor:=cosd(pangle); qvert:=sind(pangle);

pdir    The vector corresponding to the pen angle.
184 pair pdir; pdir=dir(pangle);

ppdir    The vector that is perpendicular (anticlockwise) to the pen angle.
185 pair ppdir; ppdir=dir(pangle+90);

draw_serif    A macro to draw a serif. draw_serif(1,2,f) uses the current pen to draw a serif
of extent jut at z1 to the left of the line from z2 to z1. The radius is controlled
by f, where  $0 < f < 1$ . The 'stem' from z1 to z2 starts at the point z1a.
186
187 def draw_serif(suffix $, $$)(expr fr)=
188   numeric beta, ajut, bjut;
   fr and beta control the serif radius.
189   beta=1-fr;
190   ajut=fr*jut; bjut=beta*jut;

```

The points $\$c$, $\$b$ and $\$a$ define the serif. For a top (bottom) serif the leftmost (rightmost) point is $\$c$ and $\$a$ is on the stem. $\$a$ is moved from $\$$ towards $\$$ by the distance $ajut$, and the other points are calculated from this. If the stem is not vertical, then we have to rotate $\$b$ to match.

```

191 pair vec[]; vec1=unitvector(z$$-z$);
192 vec2=ajut*vec1;
193 z$a=z$ shifted vec2;
194 z$b=z$a rotatedaround (z$,-90);
195 boolean top_serif;
196 if y$>y$$:
197     top_serif := true;
198     z$c=z$b shifted (-bjut*mdir);
199 else:
200     top_serif:=false;
201     z$c=z$b shifted (bjut*mdir);
202 fi
    Draw the parts, and finish off.
203 if top_serif:
204     draw z$c--z$b{mdir}..{z$$-z$}z$a;    % the serif
205 else:
206     draw z$c--z$b{-mdir}..{z$$-z$}z$a;
207 fi
208 labels($a,$b,$c);
209 enddef; % end of draw_serif
210
```

draw_roundserif The `draw_serif` routine can leave a bump at the end of the stem. The `draw_roundserif` routine, which takes the same arguments, is meant to eliminate the bump.

```

211
212 def draw_roundserif(suffix $, $$)(expr fr)=
213     draw_serif($, $$, fr);
214     undraw z$--z$a;
215 enddef; % end of draw_roundserif
216
```

draw_flatserif A macro to draw a serif. `draw_serif(1,2,f)` uses the current pen to draw a serif of extent jut at $z1$ to the left of the line from $z2$ to $z1$. The radius is controlled by f , where $0 < f < 1$. The ‘stem’ from $z1$ to $z2$ starts at the point $z1a$. This produces a serif like `draw_serif` except that one side of the serif is flat.

```

217
218 def draw_flatserif(suffix $, $$)(expr fr)=
219     numeric beta, ajut, bjut;
    fr and beta control the serif radius.
220     beta=1-fr;
221     ajut=fr*jut; bjut=beta*jut;
```

The points $\$c$, $\$b$ and $\$a$ define the serif. For a top (bottom) serif the leftmost (rightmost) point is $\$c$ and $\$a$ is on the stem. $\$a$ is moved from $\$$ towards $\$$ by the distance $ajut$, and the other points are calculated from this. If the stem is not vertical, then we have to rotate $\$b$ to match.

```

222 pair vec[]; vec1=unitvector(z$$-z$);
223 vec2=ajut*vec1;
224 z$a=z$ shifted vec2;
225 %% z$b=z$a rotated around (z$,-90);
226 boolean top_serif;
227 if y$>y$$:
228   top_serif := true;
229   z$c=z$ shifted (-jut*pdir);
230 else:
231   top_serif:=false;
232   z$c=z$ shifted (jut*pdir);
233 fi
234 z$b = fr[z$,z$c];
  Draw the parts, and finish off.
235 draw z$a--z$;
236 if top_serif:
237   draw z$c--z$b{pdir}..{z$$-z$}z$a;    % the serif
238 else:
239   draw z$c--z$b{-pdir}..{z$$-z$}z$a;
240 fi
241 labels($a,$b,$c);
242 enddef; % end of draw_flatserif
243

```

draw_clubserif A macro to draw a clubbed serif. `draw_clubserif(1,2,f)` uses the current pen to draw a clubbed serif of extent `clubjut` at z_1 to the left of the line from z_2 to z_1 . The serif matches z_1 like the flatserif does and ends at the point $f[z_1, z_2]$.

```

244
245 def draw_clubserif(suffix $, $$)(expr fr)=
246   numeric beta, ajut, bjut;
  fr controls the length of the serif.
247 %% beta=1-fr;
248 ajut=clubjut; bjut=beta*jut;

```

The points $\$c$, $\$b$ and $\$a$ define the serif. For a top (bottom) serif the leftmost (rightmost) point is $\$c$ and $\$a$ is on the stem, fr between $\$$ and $\$$.

```

249 pair vec[]; vec1=unitvector(z$$-z$);
250 %% vec2=ajut*vec1;
251 %% z$a=z$ shifted vec2;
252 boolean top_serif;
253 if y$>y$$:
254   top_serif := true;
255   z$c=z$ shifted (-clubjut*pdir);
256 else:

```

```

257     top_serif:=false;
258     z$c=z$ shifted (clubjut*mdir);
259   fi
260   z$a = fr[z$,z$];
  Draw the parts, and finish off.
261   draw z$c..z$a{vec1};
262   labels($a,$b,$c);
263 enddef; % end of draw_clubserif
264

```

draw_triserif A macro to draw a triangular top left serif. **draw_triserif(1,tj)** draws a serif at **z1** with its leftmost x value at **x1-tj**, where it is assumed that the top point on the stem is at **z1** and the stem was drawn with **quill**.

```

265
266 def draw_triserif(suffix $(expr tj)=
267   pen oldpen; oldpen:=currentpen;
268   pickup quill;
269   x$ts1'=lft x$; y$ts1'=bot y$;
270   pickup stylus;
271   lft x$ts1=x$ts1'; bot y$ts1=y$ts1';
272   x$ts2=x$ts1-tj; z$ts2=whatever[z$,z$ts1];
273   x$ts3=x$ts1; y$ts3=y$ts2-tj;
274   pickup stylus;
275   %% filldraw z$ts1--z$ts2--z$ts3--cycle;
276   filldraw z$--z$ts2--z$ts3--cycle;
277   pickup oldpen;
278   labels($ts1,$ts2,$ts3);
279 enddef; % end of draw_triserif
280

```

draw_istem A macro to draw the shape of an **i**. **draw_istem(1,2,f)** draws a stem whose top is at **z1** and bottom at **z2**. It draws with the current pen and the serifs are of extent **jut**. There is a top left and bottom right serif. The serif radius is controlled by **f**, with $0 < f < 1$.

```

281
282 def draw_istem(suffix $, $$)(expr fr)=
283   draw_serif($, $$, fr); % top serif
284   draw_serif($$, $, fr); % bottom serif
285   draw z$a--z$$a; % the stem
286 enddef; % end of draw_istem
287

```

draw_tstem A macro to draw the shape of the stem of a **t**. **draw_tstem(1,2,f)** draws a stem whose top is at **z1** and bottom at **z2** with a right serif at **z2**. The radius is controlled by **f**, with $0 < f < 1$. It draws with the current pen and the serif is of extent **jut**. NOTE: for the CMIN font, just the stem is drawn — no serif!

```

288
289 def draw_tstem(suffix $, $$)(expr fr)=

```

```

290 %% draw_serif($$, $, fr); % the serif
291 %% draw z$--z$$a; % the stem
292 draw z$--z$$;
293 enddef; % end of draw_tstem
294

draw_tlserif A macro to draw a top left serif. draw_tlserif(1,2,f) draws a serif on top of
a stem whose absolute top is at z1 and bottom at z2. It draws with the current
pen and the serif is of extent jut. It also draws the stem to the point z2. The
serif radius is controlled by f, with  $0 < f < 1$ .

295
296 def draw_tlserif(suffix $, $$)(expr fr)=
297 draw_serif($, $$, fr); % the serif
298 draw z$a--z$$; % the stem
299 enddef; % end of draw_tlserif
300

draw_flick A macro to draw a flick (like an eyebrow). draw_flick(1,rise,fall,fjut)
draws a stroke starting at z1 rising up by rise and falling down by fall wrt z1.
The overall stroke distance is fjut to the right of z1. It is drawn with the current
pen.

301
302 def draw_flick(suffix $)(expr rise, fall, fjut)=
The points z$a and z$b, together with z$ make up three points on the ductus.
303 rt x$b = x1+fjut; top y$b=y$-fall;
304 x$a=0.5[x$,x$b]; top y$a=y$+rise;
Draw the part, and finish off.
305 draw z$..z$a..z$b;
306 labels($a,$b);
307 enddef; % end of draw_flick
308

draw_wave A macro to draw a (horizontal) stroke with a flick at each end (down at the left
and up at the right). draw_wave(1,2,f) uses the current pen to draw from z1wl
to z1wr, where x1wl=x1, y1wl=y1-f, and x1wr=x2, y1wr=y2+f, with z1 assumed
to be at the left of z2.

309
310 def draw_wave(suffix $, $$)(expr f)=
311 x$wl=x$; y$wl=y$-f;
312 x$wr=x$$; y$wr=y$$+f;
313 z$wc=0.5[z$wl,z$wr];
314 draw z$wl{pdir}..z$wc{right}..{pdir}z$wr;
315 labels($wl,$wc,$wr);
316 enddef; % end of draw_wave

draw_tail A macro to draw a tail at the bottom of a (vertical) stem. draw_tail(1,fjut)
draws a tail at the bottom of the stem which ends at z1. The depth is controlled
by the desc_depth value and the jut by fjut.

```

```

317
318 def draw_tail(suffix $(expr fj)=
319   pen oldpen; oldpen:=currentpen;
320   pickup quill;
321   x$t1' = lft x$; y$t1=bot y$;
322   pickup stylus;
323   lft x$t1=x$t1';
324   x$tb=x$t1-fj; bot y$tb=y$t1-1/2desc_depth;
325   filldraw z${down}..{-pdir}z$tb & z$tb{pdir}..{up}z$t1--cycle;
326   pickup oldpen;
327   labels($tb,$t1);
328 enddef; % end of draw_tail

```

draw_triangle A macro to draw a downward pointing equilateral triangle. **draw_triangle(1,sz)** draws a triangle of side length sz with z1 in the middle of the base.

```

329
330 def draw_triangle(suffix $(expr sz)=
331   pen oldpen; oldpen:=currentpen;
332   pickup stylus;
333   x$t1=x$-sz/2; x$t3=x$+sz/2;
334   y$t1=y$t3=y$;
335   x$t2=x$; y$t2=y$-0.9sz; % approx sqrt(3)/2
336   filldraw z$t1--z$t2--z$t3--cycle;
337   pickup oldpen;
338   labels($t1,$t2,$t3);
339 enddef; % end of draw_triangle

```

draw_utriangle A macro to draw an upward pointing equilateral triangle. **draw_utriangle(1,sz)** draws a triangle of side length sz with z1 in the middle of the base.

```

340
341 def draw_utriangle(suffix $(expr sz)=
342   pen oldpen; oldpen:=currentpen;
343   pickup stylus;
344   x$t1=x$-sz/2; x$t3=x$+sz/2;
345   y$t1=y$t3=y$;
346   x$t2=x$; y$t2=y$+0.9sz; % approx sqrt(3)/2
347   filldraw z$t1--z$t2--z$t3--cycle;
348   pickup oldpen;
349   labels($t1,$t2,$t3);
350 enddef; % end of draw_utriangle

```

draw_rtriangle A macro to draw a downward pointing right angled triangle. **draw_rtriangle(1,sz)** draws a triangle of side length sz with z1 at the right (where the right angle is).

```

351
352 def draw_rtriangle(suffix $(expr sz)=
353   pen oldpen; oldpen:=currentpen;
354   pickup stylus;
355   x$t1=x$-sz; rt x$t3=x$;
356   y$t1=y$t3=y$;

```



```

357 x$t2=x$t3; y$t2=y$-sz;
358 filldraw z$t1--z$t2--z$t3--cycle;
359 pickup oldpen;
360 labels($t1,$t2,$t3);
361 enddef; % end of draw_rtriangle

draw_rutriangle A macro to draw an upward pointing right angled triangle. draw_rutriangle(1,sz)
draws a triangle of side length sz with z1 at the right (where the right angle is).

362
363 def draw_rutriangle(suffix $(, $))(expr sz)=
364   pen oldpen; oldpen:=currentpen;
365   pickup stylus;
366   x$t1=x$-sz; rt x$t3=x$;
367   y$t1=y$t3=y$;
368   x$t2=x$t3; y$t2=y$+sz;
369   filldraw z$t1--z$t2--z$t3--cycle;
370   pickup oldpen;
371   labels($t1,$t2,$t3);
372 enddef; % end of draw_rutriangle

draw_fork A macro to draw a forked line. draw_fork(1,2,sz) draws a graduated line from
z1 to z2 with a forked end of size sz.

373
374 def draw_fork(suffix $(, $$(, $))(expr sz)=
375   pen oldpen; oldpen:=currentpen;
376   pickup quill;
377   x$f1'=lft x$; y$f1'=bot y$;
378   x$f3'=rt x$; y$f3'=top y$;
379   pickup stylus;
380   lft x$f1=x$f1'; bot y$f1=y$f1';
381   rt x$f3=x$f3'; y$f3-y$=2*(y$-y$f1);
382   rt x$f2=x$-sz; bot y$f2=y$;
383   rt x$f4= rt x$f5=x$;
384   bot y$f5=y$-sz; top y$f4=y$+1/2sz;
385   filldraw z$f1--z$f2--z$f3--cycle; % the line
386   draw z$f2{right}..z$f4; % upper fork
387   draw z$f2{right}..z$f5; % lower fork
388   pickup oldpen;
389   labels($f1,$f2,$f3,$f4,$f5);
390 enddef; % end of draw_fork

draw_urswish A macro to draw a line up and to the right with an under curl at the end.
draw_urswish(1,2) draws from z1 towards z2, which is the max upper right
of a bounding box, where the curl is.

391
392 def draw_urswish(suffix $(, $$(, $))(expr sz)=
393   pen oldpen; oldpen:=currentpen;
394   pickup quill;
395   rt x$$s1=x$-thick; top y$$s1=y$;

```

```

396   rt x$$$2=x$$;
397 %%   x$$$3=x$$-thick;
398   lft x$$$3=x$$$1;
399   bot y$$$3=y$$-thick; y$$$2=0.5[y$$$1,y$$$3];
400 %%   draw z$---z$$$1{right}..z$$$2{down}..{-pdir}z$$$3;
401   draw z$..z$$$1{pdir}..z$$$2{down};
402   labels($$s1,$$s2,$$s3);
403 enddef; % end of draw_urswish

draw_bowl_stem Calculate where a bowl should meet a stem so that the pen is moving in the pen
direction. draw_bowl_stem(1,2) calculates the point z1i, where z1 is on the bowl and
there is a vertical line at x2.

404
405 def draw_bowl_stem(suffix $, $$)=
406   x$'=0.5[x$,x$$$]; y$'=y$; % the mid-point
407   z$''=z$' shifted (thick*pdir);
408   x$i=x$$$; z$i=whatever[z$',z$''];
409   labels($', $i);
410 enddef; % end of draw_bowl_stem

draw_clubbase Draw a clubbed bottom of a vertical stem for the CMIN font. draw_clubbase(1, 2, st, left, right)
draws from st[z2,z1] to z2. At z2 the pen is shifted left fraction of the pen
width and right fraction of the pen width, both in the direction of the pen angle.
Pen draws smoothly the path z1, left, right, z1.

411
412 % draw clubbed bottom of vertical stem for CMIN font.
413 def draw_clubbase(suffix $, $$(expr st, lft, rt)=
414   z$$l = z$$ shifted (lft*thick*(-pdir));
415   z$$r = z$$ shifted (rt*thick*(pdir));
416   z$$t = st[z$$,z$];
417   draw z$$t{z$$-z$}..z$$l--z$$r..{z$-z$$}z$$t;
418   labels($$z,$$l,$$r);
419 enddef; % end of draw_clubbase
420

draw_pdot A macro to draw a period dot. draw_pdot(1,sz) uses stylus to draw a dot
centered at z1 of overall size sz.

421
422 def draw_pdot(suffix $)(expr sz)=
  Draw a square rotated 45 degrees.
423   z$$s = z$ shifted (0.5sz*left);
424   z$$t = z$ shifted (0.5sz*up);
425   z$$u = z$$s shifted (sz*right);
426   z$$v = z$$t shifted (sz*down);
427   pickup stylus;
428   filldraw z$$s--z$$t--z$$u--z$$v--cycle;
429   labels($s,$t,$u,$v);
430 enddef; % end of draw_pdot
431

```

draw_pcomma A macro to draw a period comma. `draw_pcomma(1,t,w)` draws a comma placed at `z1` (the top), line width is `t` and bottom of line is `w` horizontally from `z1`. The pen is `stylus`.

```

432
433 def draw_pcomma(suffix $)(expr sz, ar)=
    Draw a line at z$ of width sz sloping back and down for a total width of ar and
    depth of 2ar.
434   z$s = z$ shifted (0.5sz*left);
435   z$t = z$s shifted (sz*right);
436   x$=x$s-ar; y$v=y$s-2ar;
437   z$u = z$v shifted (sz*right);
438   z$sv = (x$s, 1/3[y$s,y$v]);
439   z$tu = (x$t, y$sv);
440   z$uv = 1/3[z$u,z$v];
441   z$vu = 1/3[z$v,z$u];
442   pickup stylus;
443 %% filldraw z$s--z$t--z$u--z$v--cycle;
444   filldraw z$s--z$t--z$tu{down}..{left}z$uv--z$vu{right}..{up}z$sv--cycle;
445   labels($s,$t,$u,$v,$sv,$tu,$uv,$vu);
446 enddef; % end of draw_pcomma
447

```

draw_plq A macro to draw a period left quote. `draw_plq(1,t,w)` draws a left quote placed at `z1` (the top), line width is `t` and bottom of line is `w` horizontally from `z1`.

```

448
449 def draw_plq(suffix $)(expr sz, ar)=
    Draw a line of at z$ of width sz sloping forward and down for a total width of ar
    and depth of 2ar.
450   z$s = z$ shifted (0.5sz*left);
451   z$t = z$s shifted (sz*right);
452   x$=x$s+ar; y$v=y$s-2ar;
453   z$u = z$v shifted (sz*right);
454   z$sv = (x$s, 1/3[y$s,y$v]);
455   z$tu = (x$t, y$sv);
456   z$uv = 1/3[z$u,z$v];
457   z$vu = 1/3[z$v,z$u];
458   pickup stylus;
459 %% filldraw z$s--z$t--z$u--z$v--cycle;
460   filldraw z$s--z$t--z$tu{down}..{right}z$uv--z$vu{left}..{up}z$sv--cycle;
461   labels($s,$t,$u,$v);
462 enddef; % end of draw_plq
463

```

offsetbowl A macro to calculate the outer and inner paths (`obouter` and `obinner`) for a non-centered letter ‘o’. `offsetbowl($, sx, sy, fx, fy)` calculates the paths for an ‘o’ enclosed in a rectangle size `sx` by `sy` with its bottom lefthand corner located at `z$`, where the center of the o is at `(fx*sx, fy*sy)` in the bounding rectangle.

Two construction paths, namely `obfirst` and `obsecond` are also available. They should be identical.

All paths have points listed in a clockwise direction, starting with `z$nh` or `z$minl` as appropriate.

`offsetbowl($, sx, sy, 1/2, 1/2)` gives the same results as `obowl($, sx, sy)`.

```

464
465 path obouter, obinner, obfirst, obsecond;
466 def offsetbowl(suffix $(expr sx, sy, fx, fy)=
467   numeric inx, iny, cx, cy;
468   pair pt[];
469   path pth[];
470   inx := 1/2thick*cosd(pangle); iny := 1/2thick*sind(pangle);
471   cx := x$+fx*sx; cy := y$+fy*sy;
472   z$obc = (cx,cy);

```

Calculate points on the centre path of the pen. These are offset by `inx` from the rectangles vertical sides and by `iny` from the horizontal sides. If the pen is parallel to a side, offset the point by half the pen's dimension. First do the West, East, North and South points.

```

473   if pangle = 90:
474     z$wt=(x$+1/2thin, cy);
475     z$et=(x$+sx-1/2thin, cy);
476   else:
477     z$wt=(x$+inx, cy);
478     z$et=(x$+sx-inx, cy);
479     penpos$wt(thick,pangle);
480     penpos$et(thick,pangle);
481   fi
482   if pangle = 0:
483     z$nh=(cx, y$+sy-1/2thin);
484     z$sh=(cx, y$+1/2thin);
485   else:
486     z$nh=(cx, y$+sy-iny);
487     z$sh=(cx, y$+iny);
488     penpos$nh(thick,pangle);
489     penpos$sh(thick,pangle);
490   fi
491   obfirst := z$nh{right}..z$et{down}..z$sh{left}..z$wt{up}..cycle;

```

Now calculate the points on the path `obfirst` where the pen will draw the min width lines (i.e., the points where the pen will be moving parallel to `pangle`).

```

492   if pangle = 90:
493     z$minl = z$wt; z$maxr = z$nh; z$minr = z$et; z$maxl = z$sh;
494   else:
495     if pangle = 0:
496       z$minl = z$nh; z$maxr = z$et; z$minr = z$sh; z$maxl = z$wt;
497     else:
498       z$minl = directionpoint dir(pangle) of obfirst;
499       z$minr = directionpoint -dir(pangle) of obfirst;

```

The max points are roughly where the inner and outer diverge the most. These are simple for 0 and 90 degree pens and have been calculated above. It's not so simple for intermediate angles. For a reasonable looking result, we pick the max points to be halfway between the min points. This involves calculating some angles, paths and intersections.

```

500     pt1 := z$minr - z$minl;           % direction
501     pt2 := (- ypart pt1, xpart pt1); % perpendicular
502     pt3 := 1/2[z$minl, z$minr];
503     pt4 := pt3 shifted pt2;
504     pt5 := pt3 shifted -pt2;
505     pth1 := pt3--pt4;
506     pth2 := pt3--pt5;
507     z$maxr = pth1 intersectionpoint obfirst;
508     z$maxl = pth2 intersectionpoint obfirst;
509     \fi
510     \fi

```

Place the pen at each of these points to get the outer and inner points.

```

511     penpos$minl(thin,pangle+90);
512     penpos$maxr(thick,pangle);
513     penpos$minr(thin,pangle+90);
514     penpos$maxl(thick,pangle);

```

Finally, the outer and inner paths.

```

515     obsecond := z$minl..z$maxr..z$minr..z$maxl..cycle;
516     z$minlo = z$minl.r; z$maxro = z$maxr.r;
517     z$minro = z$minr.l; z$maxlo = z$maxl.l;
518     obouter := z$minlo{dir(pangle)}...z$maxro...z$minro{-dir(pangle)}...z$maxlo...cycle;
519     z$minli = z$minl.l; z$maxri = z$maxr.l;
520     z$minri = z$minr.r; z$maxli = z$maxl.r;
521     obinner := z$minli...z$maxri...z$minri...z$maxli...cycle;
522     labels($obc, $nh, $et, $sh, $wt);
523     labels($minl, $maxr, $minr, $maxl);
524     labels($minlo, $maxro, $minro, $maxlo);
525     labels($minli, $maxri, $minri, $maxli);
526 enddef;
527

```

obowl A macro to calculate the outer and inner paths (**obouter** and **obinner**) for the letter 'o'. **obowl**(\$, **sx**, **sy**) calculates the paths for an 'o' centered and enclosed in a rectangle size **sx** by **sy** with its bottom lefthand corner located at **z\$**.

```

528
529 def obowl(suffix $(expr sx, sy)=
530     offsetbowl($, sx, sy, 1/2, 1/2);
531 enddef;
532

```

scalarprod **v scalarprod w** calculates the scalar product between the two vectors **v** and **w**.

```

533
534 primarydef v scalarprod w =

```

```

535 (xpart v * ypart w - ypart v * xpart w)
536 enddef;
537

```

tangentpoint `tangentpoint($, path)` calculates the point `z$tan` on the path so that `z$tan--z$` is tangent to the path. It uses a binary search to find a point on the path such that the path direction is (anti-) parallel to the direction from that point to the given point.

```

538
539 def tangentpoint(suffix $(expr pth)=
540   pair vecll, veclr, veclm;    % "tangent" line directions
541   numeric pl, pr, pm;         % path parameter values
542   pair vecpl, vecpr, vecpm;    % path directions
543   pair ptl, ptr, ptm;         % path points
544   numeric sl, sr, sm;         % scalar products of path/line directions
545   numeric stpsize;            % size of current binary step
546   numeric tantol;            % tolerance
547   boolean wtest;              % a boolean
548
549   tantol := eps;

```

Initialise using the left right and middle points on the path.

```

550   pl:=0; pr:=length pth; pm:=0.5(pl+pr); stpsize:=pr-pl;
551   ptl := point pl of pth;
552   ptm := point pm of pth;
553   ptr := point pr of pth;

```

Now the various vectors, and make sure they are unit vectors to have a consistent measure scale when we take the scalar products.

```

554   vecll := unitvector (z$-ptl);
555   veclm := unitvector (z$-ptm);
556   veclr := unitvector (z$-ptr);
557   vecpl := unitvector (direction pl of pth);
558   vecpm := unitvector (direction pm of pth);
559   vecpr := unitvector (direction pr of pth);
560   sl := vecll scalarprod vecpl;
561   sm := veclm scalarprod vecpm;
562   sr := veclr scalarprod vecpr;

```

We stop if any of the points is the tangent point, or if the step size is too small, or if the signs of the left and right scalar products are the same.

```

563   wtest := (stpsize < tantol) or (sl*sr > 0) or
564           (abs sl < tantol) or (abs sm < tantol) or (abs sr < tantol);
565   if wtest:
566     if (stpsize < tantol):
567       z$tan=(0,0);
568       errmessage("No tangent; path too short");
569     else:
570       if (sl*sr > 0):
571         z$tan=(0,0);

```

```

572         errmessage("No tangent; end products have the same sign");
573     else:
574         if (abs sl < tantol):
575             z$tan=ptl;
576         else:
577             if (abs sm < tantol):
578                 z$tan=ptm;
579             else:
580                 z$tan=ptr;
581             fi
582         fi
583     fi
584 fi
585 else:

```

There should be a tangent and we iterate to find it, using bisection. If there is one to be found it will be at the current middle point. The tangent point will be somewhere between the two points that lead to opposite signed scalar products.

```

586     forever:
587         if (sl*sm < 0): % opposite signs, look between pl & pm
588             pr:=pm; pm:=0.5(pr+pl);
589             vecpr := vecpm; veclr := veclm; sr:=sm;
590         else:
591             pl:=pm; pm:=0.5(pr+pl);
592             vecpl:=vecpm; vecll:=veclm; sl:=sm;
593         fi
594         stpsize:=pr-pl;
595         ptm := point pm of pth;
596         veclm := unitvector (z$-ptm);
597         vecpm := unitvector (direction pm of pth);
598         sm := veclm scalarprod vecpm;

```

We stop the iteration if the step size is too small or the middle point is the tangent point. In either case, the middle point is the best guess that can be made.

```

599         wtest := (stpsize < tantol) or (abs sm < tantol);
600         if wtest:
601             if (abs sm < tantol):
602                 z$tan = ptm;
603             else:
604                 z$tan = ptm;
605         %%% errmessage("Warning: tangent approximated");
606         fi
607     fi
608     exitif wtest;
609 endfor
610 fi
611 enddef;
612

```

Set the mode and font setup.

```

613
614 font_coding_scheme:="ASCII minuscules and punctuation";
615 mode_setup;
616 carolmin_font_setup;
617
    We generate letters, together with some punctuation and alphabets.

618
619 input cminl;          % lower case
620 input cminu;          % capitals
621 input cminpunct;      % punctuation
622 input cminlig;        % the several dashes
623 input cmindig;        % arabic digits
624
    Set up the spacings, stretches and shrinks.

625
626 font_slant slant; font_x_height x_height#;
627 if monospace:
628     font_normal_space vstretch*9u#; % no stretching or shrinking
629     font_quad vstretch*18u#;
630     font_extra_space vstretch*9u#;
631 else:
632     font_normal_space vstretch*6u#+2letter_fit#;
633     font_normal_stretch vstretch*3u#; font_normal_shrink vstretch*2u#;
634     font_quad vstretch*18u#+4letter_fit#;
635     font_extra_space vstretch*2u#;
636
    Finally, do the ligtables, but I don't think they are needed (yet?).

637
638 fi
639
640 bye.
641
642 k#:=-.5u#; kk#:= -1.5u#; kkk#:= -2u#; % three degrees of kerning
643 kks#:= -4u#; kxx#:= -1.5u#; kkj#:= -2u#; kkr#:= -5u#; kkr#:= -2u#; kkr#:= -8u#;
644 ligtable "s":
645     "a" kern kks#, "A" kern kks#,
646     "c" kern kks#, "C" kern kks#,
647     "d" kern kks#, "D" kern kks#,
648     "e" kern kks#, "E" kern kks#,
649     "f" kern kks#, "F" kern kks#,
650     "g" kern kks#, "G" kern kks#,
651     "i" kern kks#, "I" kern kks#,
652     "j" kern kks#, "J" kern kks#,
653     "k" kern kks#, "K" kern kks#,
654     "m" kern kks#, "M" kern kks#,
655     "n" kern kks#, "N" kern kks#,
656     "o" kern kks#, "O" kern kks#,

```



```

657         "p" kern kks#, "P" kern kks#,
658         "q" kern kks#, "Q" kern kks#,
659         "r" kern kks#, "R" kern kks#,
660         "t" kern kks#, "T" kern kks#,
661         "u" kern kks#, "U" kern kks#,
662         "v" kern kks#, "V" kern kks#,
663         "w" kern kks#, "W" kern kks#,
664         "x" kern kks#, "X" kern kks#,
665         "y" kern kks#, "Y" kern kks#,
666         "z" kern kks#, "Z" kern kks#;
667
668     ligtable "S":
669         "a" kern kks#, "A" kern kks#,
670         "c" kern kks#, "C" kern kks#,
671         "d" kern kks#, "D" kern kks#,
672         "e" kern kks#, "E" kern kks#,
673         "f" kern kks#, "F" kern kks#,
674         "g" kern kks#, "G" kern kks#,
675         "i" kern kks#, "I" kern kks#,
676         "j" kern kks#, "J" kern kks#,
677         "k" kern kks#, "K" kern kks#,
678         "m" kern kks#, "M" kern kks#,
679         "n" kern kks#, "N" kern kks#,
680         "o" kern kks#, "O" kern kks#,
681         "p" kern kks#, "P" kern kks#,
682         "q" kern kks#, "Q" kern kks#,
683         "r" kern kks#, "R" kern kks#,
684         "t" kern kks#, "T" kern kks#,
685         "u" kern kks#, "U" kern kks#,
686         "v" kern kks#, "V" kern kks#,
687         "w" kern kks#, "W" kern kks#,
688         "x" kern kks#, "X" kern kks#,
689         "y" kern kks#, "Y" kern kks#,
690         "z" kern kks#, "Z" kern kks#;
691
692     ligtable "x":
693         "a" kern kks#, "A" kern kks#,
694         "b" kern kks#, "B" kern kks#,
695         "c" kern kks#, "C" kern kks#,
696         "d" kern kks#, "D" kern kks#,
697         "e" kern kks#, "E" kern kks#,
698         "g" kern kks#, "G" kern kks#,
699         "j" kern kks#, "J" kern kks#,
700         "l" kern kks#, "L" kern kks#,
701         "o" kern kks#, "O" kern kks#,
702         "q" kern kks#, "Q" kern kks#,
703         "t" kern kks#, "T" kern kks#,
704         "u" kern kks#, "U" kern kks#,
705         "v" kern kks#, "V" kern kks#,
706         "w" kern kks#, "W" kern kks#,

```

```

707         "y" kern kxx#, "Y" kern kxx#;
708
709     ligtable "X":
710         "a" kern kxx#, "A" kern kxx#,
711         "b" kern kxx#, "B" kern kxx#,
712         "c" kern kxx#, "C" kern kxx#,
713         "d" kern kxx#, "D" kern kxx#,
714         "e" kern kxx#, "E" kern kxx#,
715         "g" kern kxx#, "G" kern kxx#,
716         "j" kern kxx#, "J" kern kxx#,
717         "l" kern kxx#, "L" kern kxx#,
718         "o" kern kxx#, "O" kern kxx#,
719         "q" kern kxx#, "Q" kern kxx#,
720         "t" kern kxx#, "T" kern kxx#,
721         "u" kern kxx#, "U" kern kxx#,
722         "v" kern kxx#, "V" kern kxx#,
723         "w" kern kxx#, "W" kern kxx#,
724         "y" kern kxx#, "Y" kern kxx#;
725
726     ligtable "a": "A":
727         "b": "B":
728         "c": "C":
729         "d": "D":
730         "e": "E":
731         "h": "H":
732         "i": "I":
733         "k": "K":
734         "l": "L":
735         "m": "M":
736         "n": "N":
737         "o": "O":
738         "p": "P":
739         "t": "T":
740         "u": "U":
741         "v": "V":
742         "w": "W":
743         "z": "Z":
744         "j" kern kkj#, "J" kern kkj#;
745
746     ligtable "r":
747         "a" kern kkr#, "A" kern kkr#,
748         "b" kern kkr#, "B" kern kkr#,
749         "c" kern kkr#, "C" kern kkr#,
750         "d" kern kkr#, "D" kern kkr#,
751         "e" kern kkr#, "E" kern kkr#,
752         "f" kern kkr#, "F" kern kkr#,
753         "g" kern kkr#, "G" kern kkr#,
754         "h" kern kkr#, "H" kern kkr#,
755         "i" kern kkr#, "I" kern kkr#,
756         "j" kern kkj#, "J" kern kkj#,

```

```

757 "k" kern kkr#, "K" kern kkr#,
758 "l" kern kkr#, "L" kern kkr#,
759 "m" kern kkr#, "M" kern kkr#,
760 "n" kern kkr#, "N" kern kkr#,
761 "o" kern kkr#, "O" kern kkr#,
762 "p" kern kkr#, "P" kern kkr#,
763 "q" kern kkr#, "Q" kern kkr#,
764 "r" kern kkr#, "R" kern kkr#,
765 "s" kern kkr#, "S" kern kkr#,
766 "t" kern kkr#, "T" kern kkr#,
767 "u" kern kkr#, "U" kern kkr#,
768 "v" kern kkr#, "V" kern kkr#,
769 "w" kern kkr#, "W" kern kkr#,
770 "x" kern kkr#, "X" kern kkr#,
771 "y" kern kkr#, "Y" kern kkr#,
772 "z" kern kkr#, "Z" kern kkr#;
773
774 ligtable "R":
775 "a" kern kkr#, "A" kern kkr#,
776 "b" kern kkr#, "B" kern kkr#,
777 "c" kern kkr#, "C" kern kkr#,
778 "d" kern kkr#, "D" kern kkr#,
779 "e" kern kkr#, "E" kern kkr#,
780 "f" kern kkr#, "F" kern kkr#,
781 "g" kern kkr#, "G" kern kkr#,
782 "h" kern kkr#, "H" kern kkr#,
783 "i" kern kkr#, "I" kern kkr#,
784 "j" kern kkr#, "J" kern kkr#,
785 "k" kern kkr#, "K" kern kkr#,
786 "l" kern kkr#, "L" kern kkr#,
787 "m" kern kkr#, "M" kern kkr#,
788 "n" kern kkr#, "N" kern kkr#,
789 "o" kern kkr#, "O" kern kkr#,
790 "p" kern kkr#, "P" kern kkr#,
791 "q" kern kkr#, "Q" kern kkr#,
792 "r" kern kkr#, "R" kern kkr#,
793 "s" kern kkr#, "S" kern kkr#,
794 "t" kern kkr#, "T" kern kkr#,
795 "u" kern kkr#, "U" kern kkr#,
796 "v" kern kkr#, "V" kern kkr#,
797 "w" kern kkr#, "W" kern kkr#,
798 "x" kern kkr#, "X" kern kkr#,
799 "y" kern kkr#, "Y" kern kkr#,
800 "z" kern kkr#, "Z" kern kkr#;
801
802
803 fi
804
805 bye.
806

```

807 $\langle /mfd \rangle$

3.3 The code for normal letters

The following code is for the lower case characters.

```
808  $\langle *min \rangle$ 
809 % CMINL.MF Program file for Carolingian Minusculer lower case
810 %
811
```

a The letter 'a'.

```
812 cmchar "CMIN letter a";
813 beginchar("a", 2sb#+hstretch*(3/4x_height#+jut#),
814           x_height#, 0);
815 adjust_fit(0,0);
816 pcshiftx := sb;
817 numeric n[];
818 path pth[];
819 pair pr[];
820 pickup quill;
821 n1 := w - 2pcshiftx -jut; % effective letter width
822 % "stem"
823 x1 = pcshiftx + 6/8n1; top y1=h;
824 rt x2 = pcshiftx + n1; bot y2=0;
825 z16 = 1/6[z2,z1];
826 rt x6 = x16; y6=y16;
827 lft x4 = pcshiftx; y4 = 1/2h;
828 % stem path
829 pth1 := z1{down}..z2;
830 pr1 := direction 1 of pth1;
831 z20 = z2 shifted -10pr1;
832
833 z11 = point 3/13 of pth1; z15 = point 10/13 of pth1;
834 lft x13 = pcshiftx; y13=1/4h;
835 %%ductus:=true;
836 if ductus:
837   pickup stylus;
838 else:
839   % draw stem
840   draw pth1;
841   draw_roundserif(1,2,srad);
842   draw_roundserif(2,20,srad);
843   % draw bowl
844   %% draw z1{-pdir}..z4{down}..z6{pdir};
845   draw z11{-pdir}..z13{down}..z15{pdir};
846 fi
847 ductus:=false;
848 penlabels(1,2,3,4,5,6,7,8,11,12,13,14,15,100);
849 endchar; % end "a"
```

850

b The letter 'b'.

```
851 cmchar "CMIN letter b";
852 beginchar("b", 2sb#+hstretch*(3/4x_height#+clubjut#),
853         asc_height#, 0);
854 adjust_fit(0,0);
855 numeric n[];
856 path pth[];
857 pcshiftx := sb;
858 pickup quill;
859 n1:=w-2pcshiftx; % actual letter width
860 n2:=h/ascratio; % top of bowl
861 %%ductus:=true;
862 lft x1 = pcshiftx+clubjut; top y1=h;
863 x3 = x1; y3=1/2n2;
864 rt x5 = w-pcshiftx; y5=y3;
865 x4 = 1/2[x3,x5]; bot y4=0;
866 x6=x4; top y6=n2;
867 pth1 := z1..z3{down}..z4{right}.. z5{up};
868 z98=(0,3/4n2); z99=(w,y98);
869 pth2 := z98--z99;
870 z7 = pth1 intersectionpoint pth2;
871 if ductus:
872     pickup stylus;
873 fi
874 draw z1..z3{down}..z4{right}..z5{up}..z6{left}..z7{-pdir};
875 draw_clubserif(1,3,0.75);
876 ductus:=false;
877 penlabels(1,2,3,4,5,6,7,8,9,100);
878 endchar; % end "b"
879
```

c The letter 'c'.

```
880 cmchar "CMIN letter c";
881 beginchar("c", 2sb#+hstretch*(5/8x_height#),
882         x_height#, 0);
883 adjust_fit(0,0);
884 numeric n[];
885 pcshiftx := sb;
886 pickup quill;
887 n1:=w-2pcshiftx; % actual letter width
888 n2:=6/5w-2pcshiftx; % width of full bowl
889 x1=pcshiftx + 1/2n1; top y1 = h;
890 lft x2 = pcshiftx; y2 = 1/2h;
891 x3 = x1; bot y3 = 0;
892 rt x4 = w-pcshiftx; y4 = 1/4h;
893 x24 = x4; y24 = 3/4h;
894 %%ductus:=true;
895 if ductus:
```

```

896 pickup stylus;
897 fi
898 draw z24..z1{left}..z2{down}..z3{right}..z4{pdir};
899 ductus:=false;
900 penlabels(1,2,3,4,5,6,14,16,17,18,100);
901 endchar; % end "c"
902

```

d The letter 'd'.

```

903 cmchar "CMIN letter d";
904 beginchar("d", 2sb#+hstretch*(5/8x_height#+jut#),
905           asc_height#, 0);
906 adjust_fit(0,0);
907 numeric n[];
908 path pth[];
909 pcshiftx := sb;
910 pickup quill;
911 n1 = h/ascratio; % x-height
912 rt x1=w-pcshiftx-jut; top y1=h;
913 x2 = x1; bot y2=0;
914 x3 = x1; y3 = 1/4n1;
915 lft x5 = pcshiftx; y5=1/2n1;
916 x4 = 1/2[x3,x5]; bot y4 = 0;
917 x6=x4; top y6 = n1;
918 x7=x3; y7=3/4n1;
919 if ductus:
920 pickup stylus;
921 fi
922 draw_tstem(1,2,srad);
923 %%draw_flatserif(1,2,srad);
924 draw_clubserif(1,2,0.25);
925 draw_roundserif(2,1,srad);
926 draw z3..z4{left}..z5{up}..z6{right}..z7;
927 ductus:=false;
928 penlabels(0,1,2,3,4,5,6,7,8,100);
929 endchar; % end "d"
930

```

e The letter 'e'.

```

931 cmchar "CMIN letter e";
932 beginchar("e", sb#+hstretch*(8/8x_height#),
933           x_height#, 0);
934 adjust_fit(0,0);
935 pcshiftx := sb;
936 numeric n[];
937 path pth[];
938 pickup quill;
939 n1 := w-pcshiftx; % actual letter width
940 n2 := 6/8n1; % width of bowl
941 x1 = pcshiftx + 5/8n2; top y1=h;

```

```

942 lft x2 = pcshiftx; y2 = 1/2h;
943 x3 = pcshiftx + 1/2n2; bot y3 = 0;
944 rt x4 = pcshiftx + n2; y4 = 1/4h;
945 %%z98=(0,3/8h); z99=(w,y98);
946 z98=(0,2/8h); z99=(w,y98);
947 pth1 := z98--z99;
948 pth2 := z1{left}..z2{down}..z3{right};
949 z7 = pth1 intersectionpoint pth2;
950 rt x9 = w-pcshiftx; y9 = 3/4h;
951 z8=1/2[z7,z9];
952
953 z70 = z7 shifted 10pdir;
954 z17 = whatever[z7,z70];
955 %%x17 = w-pcshiftx;
956 top y17 = h;
957
958 rt x27=w-pcshiftx; top y27=h;
959 %%ductus:=true;
960 if ductus:
961   pickup stylus;
962 fi
963 draw z7{pdir}..z1{left}..z2{down}..z3{right}..z4{pdir};
964 %%draw z7{pdir}..z8{right}..z9{pdir};
965 %%draw z7--z17;
966 draw z7--z27;
967 ductus:=false;
968 penlabels(1,2,3,4,5,6,7,8,9,17,70,100);
969 endchar; % end "e"
970

```

f The letter 'f'.

```

971 cmchar "CMIN letter f";
972 beginchar("f", 2sb#+hstretch*(3/4x_height#+jut#),
973           (asc_height#), 0);
974 adjust_fit(0,0);
975 numeric n[];
976 pcshiftx := sb;
977 pickup quill;
978 n1 := h/ascratio; % x-height
979 %% stem
980 lft x1=lft x2=pcshiftx+jut; top y1=n1; bot y2=0;
981 %% flick
982 x4=x2; y4 = 1/2n1;
983 rt x6=w-pcshiftx; y6 = 2/3h;
984 x5 = 2/3[x4,x6]; top y5=h;
985 %% bar
986 z14=1/2[z4,z1];
987 rt x16 = w -pcshiftx; top y16 = 1/2[y14,y6];
988 %%ductus:=true;
989 if ductus:

```

```

990 pickup stylus;
991 fi
992 draw_tstem(1,2,srad);          % stem
993 draw_flatserif(1,2,srad);
994 draw_roundserif(2,1,srad);
995 draw z4{up}...z5{right}...z6;   % top
996 draw z14{right}..z16{pdir};     % bar
997 ductus:=false;
998 ductus:=false;
999 penlabels(1,2,3,4,5,6,7,14,15,16,100);
1000 endchar; % end "f"
1001

```

g The letter 'g'.

```

1002 cmchar "CMIN letter g";
1003 beginchar("g", 2sb#+hstretch*(5/4x_height#),
1004           x_height#, desc_depth#);
1005 adjust_fit(0,0);
1006 path pth[];
1007 pair vec[];
1008 numeric n[];
1009 pcshiftx := sb;
1010 pickup quill;
1011 n1 := 4/5w - 2pcshiftx; % width of top o
1012 %% top 0
1013 x1 = x3 = pcshiftx + 1/2n1; top y1=h; bot y3=1/8h;
1014 lft x4 = pcshiftx; rt x2 = pcshiftx+n1; y2 = y4 = 1/2[y1,y3];
1015
1016 %% top flick
1017 lft x11 = rt x1; y11=y1;
1018 rt x13 = w-pcshiftx; y13 = h;
1019
1020 %% bottom
1021 z31=z3;
1022 x33 = w-pcshiftx; y33 = -1/2d;
1023 x35=x31; bot y35 = -d;
1024 rt x36 = lft x4; y36 = 1/2[y35,y33];
1025
1026 %%ductus:=true;
1027 if ductus:
1028 pickup stylus;
1029 fi
1030 draw z1..z2..z3..z4..cycle; % top 0
1031 draw z11{right}..z13{pdir}; % flick
1032 draw z31{right}..z33{down}..z35{left}..z36;
1033 ductus:=false;
1034 penlabels(1,2,3,4,5,6,7,8,9,11,12,13,31,32,33,34,35,36,100);
1035 endchar; % end "g"
1036

```


h The letter 'h'.

```

1037 cmchar "CMIN letter h";
1038 beginchar("h", 2sb#+hstretch*(6/8x_height#+clubjut#),
1039           asc_height#, 0);
1040 adjust_fit(0,0);
1041 pcshiftx := sb;
1042 pickup quill;
1043 numeric n[];
1044 %% stem
1045 lft x1=lft x2=pcshiftx+jut;
1046 top y1=h; bot y2=0;
1047 %% bowl
1048 z3=1/2[z2,z1];
1049 rt x5 = w-pcshiftx; y5=y3;
1050 x4=1/2[x3,x5]; top y4 = h/ascratio;
1051 %%lft x6=x4; bot y6=0;
1052 x6=1/2[x4,x5];
1053 bot y6=0;
1054 %%ductus:=true;
1055 if ductus:
1056   pickup stylus;
1057 fi
1058 draw_tstem(1,2,srad); % stem
1059 %%draw_flatserif(1,2,srad);
1060 draw_clubserif(1,2,0.5);
1061 draw_roundserif(2,1,srad);
1062 draw z3{pdir}..z4..z5..z6{-pdir};
1063
1064 ductus:=false;
1065 penlabels(1,2,3,4,5,6,100);
1066 endchar; % end "h"
1067

```

i The letter 'i'.

```

1068 cmchar "CMIN letter i";
1069 beginchar("i", 2sb#+hstretch*(thick#+2jut#),
1070           x_height#, 0);
1071 adjust_fit(0,0);
1072 pcshiftx := sb;
1073 pickup quill;
1074 lft x1=lft x2=pcshiftx+jut;
1075 top y1=h; bot y2=0;
1076 %%ductus:=true;
1077 draw_tstem(1,2,srad);
1078 draw_flatserif(1,2,srad);
1079 draw_roundserif(2,1,srad);
1080 ductus:=false;
1081 penlabels(1,2,100);
1082 endchar; % end "i"
1083

```

j The letter 'j'. This letter was not in the script, it being a later invention.

```

1084 cmchar "CMIN letter j";
1085 beginchar("j", sb#+hstretch*(3/8x_height#),
1086           x_height#, desc_depth#);
1087 adjust_fit(0,0);
1088 pcshiftx := sb;
1089 numeric n[];
1090 pickup quill;
1091 %% stem
1092 rt x1 = w-pcshiftx; top y1 = h;
1093 x2=x1; y2=-1/2d;
1094 %% bottom hook
1095 lft x3 = 0; bot y3 = -d;
1096 %%ductus:=true;
1097 if ductus:
1098   pickup stylus;
1099 fi
1100 draw z1--z2; % stem
1101 draw_flatserif(1,2,srad);
1102 draw z2{down}..z3{-pdir}; % hook
1103 ductus:=false;
1104 penlabels(1,2,3,100);
1105 endchar; % end "j"
1106

```

k The letter 'k'.

```

1107 cmchar "CMIN letter k";
1108 beginchar("k", 2sb#+hstretch*(7/8x_height#+clubjut#),
1109           asc_height#, 0);
1110 adjust_fit(0,0);
1111 pcshiftx := sb;
1112 path pth[];
1113 numeric n[];
1114 pickup quill;
1115 %% stem
1116 n1 = h/ascratio; % x-height
1117 lft x1=lft x2=pcshiftx+clubjut;
1118 y1=h; bot y2=0;
1119 %% top arm
1120 %%z3=(x1,1/2n1);
1121 x3 = x1; top y3 = 1/2n1;
1122 rt x5=5/8[x1,w-pcshiftx]; top y5=n1;
1123 x4=1/2[x3,x5]; top y4=n1;
1124 pth1 := z3{pdir}..z5{up};
1125 %% lower arm
1126 pth2 := (rt x1,0)--(rt x1,h);
1127 z13 = pth1 intersectionpoint pth2;
1128 pickup threequarterquill;
1129 lft x14 = x13; bot y14 = y13;
1130 rt x7=w-pcshiftx; bot y7=0;

```

```

1131 pickup quill;
1132 %%ductus:=true;
1133 if ductus:
1134   pickup stylus;
1135 fi
1136 draw_tstem(1,2,srad);           % stem
1137 draw_clubserif(1,2,0.5);
1138 draw_roundserif(2,1,srad);
1139 draw pth1;                     % upper arm
1140 pickup threequarterquill;
1141 %%draw z14{down}..z7{-ppdir};   % lower arm
1142 %%draw z14{down}..z7;           % lower arm
1143 draw z14{-ppdir}..z7{right};    % lower arm
1144 pickup quill;
1145 ductus:=false;
1146 penlabels(1,2,3,4,5,6,7,8,9,13,14,100);
1147 endchar; % end "k"
1148

1 The letter 'l'.

1149 cmchar "CMIN letter l";
1150 beginchar("l", 2sb#+hstretch*(thick#+clubjut#+jut#),
1151           asc_height#, 0);
1152 adjust_fit(0,0);
1153 pcshiftx := sb;
1154 pickup quill;
1155 lft x1=pcshiftx+clubjut; top y1=h;
1156 x2=x1; bot y2=0;
1157 %%ductus:=true;
1158 if ductus:
1159   pickup stylus;
1160 fi
1161 draw_tstem(1,2,srad);
1162 draw_clubserif(1,2,0.5);
1163 draw_roundserif(2,1,srad);
1164 ductus:=false;
1165 penlabels(1,2,3,4,100);
1166 endchar; % end "l"
1167

m The letter 'm'.

1168 cmchar "CMIN letter m";
1169 beginchar("m", 2sb#+hstretch*(x_height#+2jut#),
1170           x_height#, 0);
1171 adjust_fit(0,0);
1172 pcshiftx := sb;
1173 numeric n[];
1174 path pth[];
1175 pickup quill;
1176 %% left stem

```

```

1177 lft x1=pcshiftx+jut; top y1=h;
1178 x2=x1; bot y2=0;
1179 %% start of bowl
1180 x13=x1; y13=3/4h;
1181 %% bottom of right and middle stems
1182 rt x6 = w -pcshiftx - jut; y6=y2;
1183 z4 = 1/2[z2,z6];
1184 %% top of middle and right stems
1185 x3=x4; y3=y1;
1186 x5=x6; y5=y13;
1187 %% top of bowls
1188 x14=2/3[x13,x3]; top y14 = h;
1189 x33=x4; y33=y13;
1190 x34=2/3[x33,x5]; y34 = y14;
1191 %%ductus:=true;
1192 if ductus:
1193   pickup stylus;
1194 fi
1195 draw_tstem(1,2,srad); % left half
1196 draw_flatserif(1,2,srad);
1197 draw_roundserif(2,1,srad);
1198 draw z13{pdir}..z14{right}..z33{down}--z4;
1199 draw_roundserif(4,3,srad);
1200 draw z33{pdir}..z34{right}..z5{down}--z6;
1201 draw_roundserif(6,5,srad);
1202 ductus:=false;
1203 penlabels(1,2,3,4,5,6,7,8,9,13,14,33,34,100);
1204 endchar; % end "m"
1205

```

n The letter 'n'.

```

1206 cmchar "CMIN letter n";
1207 beginchar("n", 2sb#+hstretch*(5/8x_height#+2jut#),
1208           x_height#, 0);
1209 adjust_fit(0,0);
1210 pcshiftx := sb;
1211 pickup quill;
1212 %% left stem
1213 lft x1=pcshiftx+jut; top y1=h;
1214 x2=x1; bot y2=0;
1215 % start of bowl
1216 x13=x1; y13=3/4h;
1217 %% right stem
1218 rt x4 = w-pcshiftx-jut; y4=y2;
1219 x3 = x4; y3 = y13;
1220 % top of bowl
1221 x14 = 2/3[x13,x3]; top y14 = h;
1222 %%ductus:=true;
1223 if ductus:
1224   pickup stylus;

```

```

1225 fi
1226 draw_tstem(1,2,srad); % left stem
1227 draw_flatserif(1,2,srad);
1228 draw_roundserif(2,1,srad);
1229 draw z13{pdir}..z14{right}..z3{down}--z4;
1230 draw_roundserif(4,3,srad);
1231 ductus:=false;
1232 penlabels(1,2,3,4,5,6,13,14,100);
1233 endchar; % end "n"
1234

```

o The letter 'o'.

```

1235 cmchar "CMIN letter o";
1236 beginchar("o", 2sb#+hstretch*(3/4x_height#),
1237           x_height#, 0);
1238 adjust_fit(0,0);
1239 pcshiftx := sb;
1240 pickup quill;
1241 lft x1=pcshiftx; rt x3=w-pcshiftx; y1=y3=h/2;
1242 x2=x4=w/2; top y2=h+o; bot y4=-o;
1243 draw z1..z2..z3..z4..cycle;
1244 z5=(pcshiftx,-o);
1245 obowl(5, w-2pcshiftx, h+2o);
1246 %%ductus:=true;
1247 if ductus:
1248   pickup stylus;
1249   draw obfirst; draw obsecond; draw obouter; draw obinner;
1250 else:
1251   %% fill obouter; unfill obinner;
1252 fi
1253 ductus:=false;
1254 penlabels(1,2,3,4,5,100);
1255 endchar; % end "o"
1256

```

p The letter 'p'.

```

1257 cmchar "CMIN letter p";
1258 beginchar("p", 2sb#+hstretch*(3/4x_height#+jut#),
1259           x_height#, desc_depth#);
1260 adjust_fit(0,0);
1261 pcshiftx := sb;
1262 numeric n[];
1263 pickup quill;
1264 %% stem
1265 lft x1=lft x2=pcshiftx+jut;
1266 top y1=h; bot y2=-d;
1267 %% bowl % from "d"
1268 z3=3/4[(x1,0),z1];
1269 rt x5 = w-pcshiftx; y5=1/2h;
1270 x4=1/2[x3,x5]; top y4 = h;

```

```

1271 x6=x4; bot y6=0;
1272 z7=1/4[(x1,0),z1];
1273
1274 %%ductus:=true;
1275 if ductus:
1276   pickup stylus;
1277 fi
1278 draw_tstem(1,2,srad); % stem
1279 draw_flatserif(1,2,srad);
1280 draw_roundserif(2,1,srad)
1281 draw z3{pdir}..z4{right}..z5{down}..z6{left}..z7; % bowl (from d)
1282 ductus:=false;
1283 penlabels(1,2,3,4,5,6,7,8,100);
1284 endchar; % end "p"
1285

```

q The letter 'q'.

```

1286 cmchar "CMIN letter q";
1287 beginchar("q", 2sb#+hstretch*(3/4x_height#+jut#),
1288           x_height#, desc_depth#);
1289 adjust_fit(0,0);
1290 pcshiftx := sb;
1291 numeric n[];
1292 pickup quill;
1293 %% right stem
1294 rt x1=w-pcshiftx-jut; top y1=h;
1295 x2=x1; bot y2=-d;
1296 %% bowl
1297 z3=1/4[(x1,0),z1];
1298 lft x5=pcshiftx; y5=1/2h;
1299 x4=1/2[x3,x5]; bot y4=0;
1300 x6=x4; top y6=h;
1301 z7=3/4[(x1,0),z1];
1302 %%ductus:=true;
1303 if ductus:
1304   pickup stylus;
1305 fi
1306 draw_tstem(1,2,srad);
1307 %%draw_flatserif(1,2,srad);
1308 draw_roundserif(2,1,srad);
1309 draw z3{-pdir}..z4{left}..z5{up}..z6{right}..z7;
1310 ductus:=false;
1311 penlabels(0,1,2,3,4,5,6,7,8,100);
1312 endchar; % end "q"
1313

```

r The letter 'r'.

```

1314 cmchar "CMIN letter r";
1315 beginchar("r", 2sb#+hstretch*(7/8x_height#+jut#),
1316           x_height#, 0);

```

```

1317 adjust_fit(0,0);
1318 pcshiftx := sb;
1319 numeric n[];
1320 pickup quill;
1321 %% stem
1322 lft x1=lft x2=pcshiftx+jut;
1323 top y1=h; bot y2=0;
1324 %% bowl
1325 z3=3/4[z2,z1];
1326 rt x5 = w-pcshiftx; y5=y3;
1327 x4=1/2[x3,x5]; top y4 = h;
1328 %%ductus:=true;
1329 if ductus:
1330   pickup stylus;
1331 fi
1332 draw_tstem(1,2,srad); % stem
1333 draw_flatserif(1,2,srad);
1334 draw_roundserif(2,1,srad);
1335 draw z3{pdir}..z4{right}..z5{pdir};
1336 ductus:=false;
1337 penlabels(1,2,3,4,5,6,7,8,9,100);
1338 endchar; % end "r"
1339

```

s The letter 's'. This is a 'long' S.

```

1340 cmchar "CMIN letter s";
1341 beginchar("s", 2sb#+hstretch*(3/4x_height#+clubjut#),
1342           asc_height#, 0);
1343 adjust_fit(0,0);
1344 numeric n[];
1345 pcshiftx := sb;
1346 pickup quill;
1347 n1 := h/asratio; % x-height
1348 %% stem
1349 lft x1=lft x2=pcshiftx+clubjut; top y1=n1; bot y2=0;
1350 %% flick
1351 x4=x2; y4 = 1/2n1;
1352 rt x6=w-pcshiftx; y6 = 2/3h;
1353 x5 = 2/3[x4,x6]; top y5=h;
1354 %%ductus:=true;
1355 if ductus:
1356   pickup stylus;
1357 fi
1358 draw_tstem(1,2,srad); % stem
1359 draw_clubserif(1,2,0.5);
1360 draw_roundserif(2,1,srad);
1361 draw z4{up}...z5{right}...z6;
1362 ductus:=false;
1363 penlabels(1,2,3,4,5,6,7,11,17,20,26,27,28,100);
1364 endchar; % end "s"

```

1365

t The letter 't'.

```
1366 cmchar "CMIN letter t";
1367 beginchar("t", 2sb#+hstretch*(x_height#),
1368           x_height#, 0);
1369 adjust_fit(0,0);
1370 pcshiftx := sb;
1371 numeric n[];
1372 pickup quill;
1373 %% bar
1374 top y3=h; n3 := h - top y3;
1375 lft x1 = pcshiftx; rt x2=w-pcshiftx;
1376 %%top y1 = y3-n3; bot y2=y3+n3;
1377 y1 = y3-n3; y2=y3;
1378 n1 := w; % width of bowl
1379 n2 := h; % height of bowl
1380 x3=1/2[x1,x2];
1381 lft x4 = pcshiftx + 1/8n1; y4 = 1/2h;
1382 x5=x3; bot y5 = 0;
1383 rt x6 = pcshiftx + 3/4n1; top y6 = 1/4h;
1384 %%ductus:=true;
1385 if ductus:
1386   pickup stylus;
1387 fi
1388 draw z1{pdir}..z3..z2{pdir}; % bar
1389 draw z3..z4{down}..z5{right}..z6{pdir};
1390 ductus:=false;
1391 penlabels(1,2,3,4,5,6,11,12,100);
1392 endchar; % end "t"
1393
```

u The letter 'u'.

```
1394 cmchar "CMIN letter u";
1395 beginchar("u", 2sb#+hstretch*(5/8x_height#+2jut#),
1396           x_height#, 0);
1397 adjust_fit(0,0);
1398 pcshiftx := sb;
1399 numeric n[];
1400 pickup quill;
1401 %% left stem
1402 lft x1 = pcshiftx + jut; top y1 = h;
1403 % start of bowl
1404 x13=x1; y13=1/3h;
1405 % right stem
1406 rt x3=w-pcshiftx-jut; y3=y1;
1407 x4=x3; bot y4=0;
1408 z33=1/4[z4,z3]; % start of bowl
1409 % middle of bowl
1410 x14=1/3[x1,x3]; bot y14=0;
```



```

1411 %%ductus:=true;
1412 if ductus:
1413   pickup stylus;
1414 fi
1415 draw z1--z13{down}..z14{right}..z33{pdir};
1416 draw_flatserif(1,13,srad);
1417 draw_tstem(3,4,srad);           % right stem
1418 draw_flatserif(3,4,srad);
1419 draw_roundserif(4,3,srad);
1420 ductus:=false;
1421 penlabels(1,2,3,4,5,6,7,8,9,11,13,14,16,33,100);
1422 endchar; % end "u"
1423

v The letter 'v'.
1424 cmchar "CMIN letter v";
1425 beginchar("v", 2sb#+hstretch*(3/4x_height#+jut#),
1426           x_height#, 0);
1427 adjust_fit(0,0);
1428 pcshiftx := sb;
1429 pickup quill;
1430 %% left stem
1431 lft x1 = pcshiftx + jut; top y1 = h;
1432 % start of bowl
1433 x13=x1; y13=1/3h;
1434 % right stem
1435 rt x3=w-pcshiftx-jut; y3=y1;
1436 x4=x3; bot y4=0;
1437 z33=1/4[z4,z3]; % start of bowl
1438 % middle of bowl
1439 x14=1/3[x1,x3]; bot y14=0;
1440 rt x23=w-pcshiftx; y23=3/4h;
1441 lft x26=x14; top y26=h;
1442 %%ductus:=true;
1443 if ductus:
1444   pickup stylus;
1445 fi
1446 draw z1--z13{down}..z14{right}; % left stem
1447 draw_flatserif(1,13,srad);
1448 draw z14{pdir}..z23{up}..z26;
1449
1450 ductus:=false;
1451 penlabels(1,2,3,4,5,11,13,14,23,26,100);
1452 endchar; % end "v"
1453

w The letter 'w'. This is a modern version as the font did not have a W.
1454 cmchar "CMIN letter w";
1455 beginchar("w", 2sb#+hstretch*(x_height#+2jut#),
1456           x_height#, 0);

```

```

1457 adjust_fit(0,0);
1458 pcshiftx := sb;
1459 pickup quill;
1460 %% left stem
1461 lft x1 = pcshiftx + jut; top y1 = h;
1462 % start of bowl
1463 x13=x1; y13=1/3h;
1464 % right stem
1465 rt x5=w-pcshiftx-jut; y5=y1;
1466 x6=x5; bot y6=0;
1467 z53=1/4[z6,z5]; % start of bowl
1468 % middle stem
1469 z3 = 1/2[z1,z5];
1470 z33=(x3,y13);
1471 % middle of bowl
1472 x14=1/3[x1,x3]; bot y14=0;
1473 % middle of right bowl
1474 x34=1/3[x3,x5]; y34=y14;
1475 %%ductus:=true;
1476 if ductus:
1477   pickup stylus;
1478 fi
1479 draw z1--z13{down}..z14{right}..z33{pdir}; % left
1480 draw_flatserif(1,13,srad);
1481 draw z3--z33{down}..z34{right}..z53{pdir}; % middle
1482 draw_flatserif(3,33,srad);
1483 draw_tstem(5,6,srad); % right stem
1484 draw_flatserif(5,6,srad);
1485 draw_roundserif(6,5,srad);
1486 ductus:=false;
1487 penlabels(1,2,3,4,5,6,7,8,9,11,13,14,33,34,53,16,100);
1488 endchar; % end "w"
1489

```

x The letter 'x'.

```

1490 cmchar "CMIN letter x";
1491 beginchar("x", 2sb#+hstretch*(x_height#),
1492           x_height#, 0);
1493 adjust_fit(0,0);
1494 path pth[];
1495 numeric n[];
1496 pcshiftx := sb;
1497 pickup quill;
1498 n1 := w-2pcshiftx; % total width
1499 n2 := thick; % triangle side
1500 %% diag top left to bottom right
1501 lft x1=pcshiftx; bot y1=2/3h;
1502 x2=pcshiftx+1/7n1; top y2=h;
1503 x5=pcshiftx+11/14n1; bot y5=-1/4d;
1504 rt x6=pcshiftx+n1; top y6=1/6h;

```

```

1505 z25 = 1/3[z2,z5];
1506 pth1 := z2{right}..z5{right};
1507 z251 = point 1/3 of pth1;
1508 z252 = point 2/3 of pth1;
1509 %% diag bottom left to top right
1510 lft x10 = -w + 2pcshiftx; bot y10=-d;
1511 x50=x5; top y50=h;
1512 rt x60 = w -pcshiftx; bot y60=2/3h;
1513 x40=pcshiftx+8/14n1; y40=h;
1514
1515 x150=x2; bot y150 = 0;
1516 lft x160 = pcshiftx; top y160=1/3h;
1517 %%ductus:=true;
1518 if ductus:
1519   pickup stylus;
1520 fi
1521 draw z1..z2{right}..z5{right}..z6;    % top left to bottom right
1522 %%draw z10{1/2pdir}..z25..z50{right}..z60;    % bottom left to top right
1523 draw z251..z50{right}..z60;    % middle to top right
1524 draw z252..z150{left}..z160;    % middle to bottom left
1525 ductus:=false;
1526 penlabels(1,2,3,4,5,6,7,8,10,25,251,252,40,50,150,60,160,100);
1527 endchar; % end "x"
1528

```

y The letter ‘y’. Modern, but with a dot on top.

```

1529 cmchar "CMIN letter y";
1530 beginchar("y", 2sb#+hstretch*(7/8x_height#+jut#),
1531           x_height#, desc_depth#);
1532 adjust_fit(0,0);
1533 pcshiftx := sb;
1534 path pth[];
1535 numeric n[];
1536 pickup quill;
1537 %% left stem top
1538 lft x1 = pcshiftx + jut; top y1 = h;
1539 bot y3 = 0;
1540 %% right stem
1541 rt x5=w-pcshiftx; top y5 = h;
1542 lft x9=-1/2w; bot y9 = -d;
1543 pth1 := z9{right}..z5{up};
1544 pth2 := (0,y3)--(w,y3);
1545 z99 = pth1 intersectionpoint pth2;
1546 x3 = x99;
1547 %%ductus:=true;
1548 if ductus:
1549   pickup stylus;
1550 fi
1551 draw z1{down}..z3;          % left stem
1552 draw_flatserif(1,3,srad);

```

```

1553 draw pth1; % right stem
1554 draw_flatserif(5,3,srad);
1555 ductus:=false;
1556 penlabels(1,2,3,4,5,6,7,8,9,10,100);
1557 endchar; % end "y"
1558

z The letter 'z'.

1559 cmchar "CMIN letter z";
1560 beginchar("z", 2sb#+hstretch*(x_height#),
1561           x_height#, 0);
1562 adjust_fit(0,0);
1563 numeric n[];
1564 path pth[];
1565 pcshiftx := sb;
1566 pickup quill;
1567 %% top bar ends
1568 top y12 = h;
1569 n1 := h - y12;
1570 lft x1 = pcshiftx; rt x2 = w-pcshiftx;
1571 %%y1 = y12-n1; y2=y12+n1;
1572 y1 = y12-n1; y2=y12;
1573 x12 = 1/2[x1,x2];
1574 pth1 := z1{pdir}..z12{right}..z2{pdir};
1575 %% bottom bar ends
1576 bot y34 = 0;
1577 x3=x1; x4=x2;
1578 %%y3=y34-n1; y4=y34+n1;
1579 y3=y34-n1; y4=y34;
1580 x34 = 1/2[x3,x4];
1581 pth2 := z3{pdir}..z34{right}..z4{pdir};
1582 %% diagonal
1583 z5=z3;
1584 pth3 := z5--(3/4w,h+jut);
1585 z6 = pth3 intersectionpoint pth1;
1586 %%ductus:=true;
1587 if ductus:
1588   pickup stylus;
1589 fi
1590 draw pth1; draw pth2; % bars
1591 draw z5--z6;
1592 ductus:=false;
1593 penlabels(1,2,3,4,5,6,10o,10i,20o,20i,100);
1594 endchar; % end "z"
1595

```

The end of this file

```
1596 </min>
```

3.4 The code for capital letters

Carolingian Minuscule ‘capitals’ were drawn with the same pen as for the minuscules, and the letters were merely larger versions of the minuscules.

```

1597 (*maj)
1598 % CMINU.MF Program file for Carolingian Minuscule ‘capitals’
1599 %
1600

```

A The letter ‘A’.

```

1601 cmchar "CMIN letter A";
1602 beginchar("A", 2sb#+caprat*hstretch*(3/4x_height#+jut#),
1603           caprat*x_height#, 0);
1604 adjust_fit(0,0);
1605 pcshiftx := sb;
1606 numeric n[];
1607 path pth[];
1608 pair pr[];
1609 pickup quill;
1610 n1 := w - 2pcshiftx -jut; % effective letter width
1611 % "stem"
1612 x1 = pcshiftx + 6/8n1; top y1=h;
1613 rt x2 = pcshiftx + n1; bot y2=0;
1614 z16 = 1/6[z2,z1];
1615 rt x6 = x16; y6=y16;
1616 lft x4 = pcshiftx; y4 = 1/2h;
1617 % stem path
1618 pth1 := z1{down}..z2;
1619 pr1 := direction 1 of pth1;
1620 z20 = z2 shifted -10pr1;
1621
1622 z11 = point 3/13 of pth1; z15 = point 10/13 of pth1;
1623 lft x13 = pcshiftx; y13=1/4h;
1624 %%ductus:=true;
1625 if ductus:
1626   pickup stylus;
1627 else:
1628   % draw stem
1629   draw pth1;
1630   draw_roundserif(1,2,srad);
1631   draw_roundserif(2,20,srad);
1632   % draw bowl
1633 %% draw z1{-pdir}..z4{down}..z6{pdir};
1634 draw z11{-pdir}..z13{down}..z15{pdir};
1635 fi
1636 ductus:=false;
1637 penlabels(1,2,3,4,5,6,7,8,11,12,13,14,15,100);
1638 endchar; % end "A"
1639

```

B The letter 'B'.

```

1640 cmchar "CMIN letter B";
1641 beginchar("B", 2sb#+caprat*hstretch*(3/4x_height#+clubjut#),
1642           caprat*asc_height#, 0);
1643 adjust_fit(0,0);
1644 numeric n[];
1645 path pth[];
1646 pcshiftx := sb;
1647 pickup quill;
1648 n1:=w-2pcshiftx;    % actual letter width
1649 n2:=h/ascratio;    % top of bowl
1650 %%ductus:=true;
1651 lft x1 = pcshiftx+clubjut; top y1=h;
1652 x3 = x1; y3=1/2n2;
1653 rt x5 = w-pcshiftx; y5=y3;
1654 x4 = 1/2[x3,x5]; bot y4=0;
1655 x6=x4; top y6=n2;
1656 pth1 := z1..z3{down}..z4{right}.. z5{up};
1657 z98=(0,3/4n2); z99=(w,y98);
1658 pth2 := z98--z99;
1659 z7 = pth1 intersectionpoint pth2;
1660 if ductus:
1661   pickup stylus;
1662 fi
1663 draw z1..z3{down}..z4{right}..z5{up}..z6{left}..z7{-pdir};
1664 draw_clubserif(1,3,0.75);
1665 ductus:=false;
1666 penlabels(1,2,3,4,5,6,7,8,9,100);
1667 endchar; % end "B"
1668

```

C The letter 'C'.

```

1669 cmchar "CMIN letter C";
1670 beginchar("C", 2sb#+caprat*hstretch*(5/8x_height#),
1671           caprat*x_height#, 0);
1672 adjust_fit(0,0);
1673 numeric n[];
1674 pcshiftx := sb;
1675 pickup quill;
1676 n1:=w-2pcshiftx;    % actual letter width
1677 n2:=6/5w-2pcshiftx; % width of full bowl
1678 x1=pcshiftx + 1/2n1; top y1 = h;
1679 lft x2 = pcshiftx; y2 = 1/2h;
1680 x3 = x1; bot y3 = 0;
1681 rt x4 = w-pcshiftx; y4 = 1/4h;
1682 x24 = x4; y24 = 3/4h;
1683 %%ductus:=true;
1684 if ductus:
1685   pickup stylus;
1686 fi

```

```

1687 draw z24..z1{left}..z2{down}..z3{right}..z4{pdir};
1688 ductus:=false;
1689 penlabels(1,2,3,4,5,6,14,16,17,18,100);
1690 endchar; % end "C"
1691

```

D The letter 'D'.

```

1692 cmchar "CMIN letter D";
1693 beginchar("D", 2sb#+caprat*hstretch*(5/8x_height#+jut#),
1694           caprat*asc_height#, 0);
1695 adjust_fit(0,0);
1696 numeric n[];
1697 path pth[];
1698 pcshiftx := sb;
1699 pickup quill;
1700 n1 = h/ascratio; % x-height
1701 rt x1=w-pcshiftx-jut; top y1=h;
1702 x2 = x1; bot y2=0;
1703 x3 = x1; y3 = 1/4n1;
1704 lft x5 = pcshiftx; y5=1/2n1;
1705 x4 = 1/2[x3,x5]; bot y4 = 0;
1706 x6=x4; top y6 = n1;
1707 x7=x3; y7=3/4n1;
1708 if ductus:
1709   pickup stylus;
1710 fi
1711 draw_tstem(1,2,srad);
1712 %%draw_flatserif(1,2,srad);
1713 draw_clubserif(1,2,0.25);
1714 draw_roundserif(2,1,srad);
1715 draw z3..z4{left}..z5{up}..z6{right}..z7;
1716 ductus:=false;
1717 penlabels(0,1,2,3,4,5,6,7,8,100);
1718 endchar; % end "D"
1719

```

E The letter 'E'.

```

1720 cmchar "CMIN letter E";
1721 beginchar("E", sb#+caprat*hstretch*(8/8x_height#),
1722           caprat*x_height#, 0);
1723 adjust_fit(0,0);
1724 pcshiftx := sb;
1725 numeric n[];
1726 path pth[];
1727 pickup quill;
1728 n1 := w-pcshiftx; % actual letter width
1729 n2 := 6/8n1; % width of bowl
1730 x1 = pcshiftx + 5/8n2; top y1=h;
1731 lft x2 = pcshiftx; y2 = 1/2h;
1732 x3 = pcshiftx + 1/2n2; bot y3 = 0;

```

```

1733 rt x4 = pcshiftx + n2; y4 = 1/4h;
1734 %%z98=(0,3/8h); z99=(w,y98);
1735 z98=(0,2/8h); z99=(w,y98);
1736 pth1 := z98--z99;
1737 pth2 := z1{left}..z2{down}..z3{right};
1738 z7 = pth1 intersectionpoint pth2;
1739 rt x9 = w-pcshiftx; y9 = 3/4h;
1740 z8=1/2[z7,z9];
1741
1742 z70 = z7 shifted 10pdir;
1743 z17 = whatever[z7,z70];
1744 %%x17 = w-pcshiftx;
1745 top y17 = h;
1746
1747 rt x27=w-pcshiftx; top y27=h;
1748 %%ductus:=true;
1749 if ductus:
1750   pickup stylus;
1751 fi
1752 draw z7{pdir}..z1{left}..z2{down}..z3{right}..z4{pdir};
1753 %%draw z7{pdir}..z8{right}..z9{pdir};
1754 %%draw z7--z17;
1755 draw z7--z27;
1756 ductus:=false;
1757 penlabels(1,2,3,4,5,6,7,8,9,17,70,100);
1758 endchar; % end "E"
1759

```

F The letter 'F'.

```

1760 cmchar "CMIN letter F";
1761 beginchar("F", 2sb#+caprat*hstretch*(3/4x_height#+jut#),
1762           (asc_height#), 0);
1763 adjust_fit(0,0);
1764 numeric n[];
1765 pcshiftx := sb;
1766 pickup quill;
1767 n1 := h/ascratio; % x-height
1768 %% stem
1769 lft x1=lft x2=pcshiftx+jut; top y1=n1; bot y2=0;
1770 %% flick
1771 x4=x2; y4 = 1/2n1;
1772 rt x6=w-pcshiftx; y6 = 2/3h;
1773 x5 = 2/3[x4,x6]; top y5=h;
1774 %% bar
1775 z14=1/2[z4,z1];
1776 rt x16 = w -pcshiftx; top y16 = 1/2[y14,y6];
1777 %%ductus:=true;
1778 if ductus:
1779   pickup stylus;
1780 fi

```



```

1781 draw_tstem(1,2,srad);          % stem
1782 draw_flatserif(1,2,srad);
1783 draw_roundserif(2,1,srad);
1784 draw z4{up}...z5{right}...z6;   % top
1785 draw z14{right}..z16{pdir};     % bar
1786 ductus:=false;
1787 ductus:=false;
1788 penlabels(1,2,3,4,5,6,7,14,15,16,100);
1789 endchar; % end "F"
1790

```

G The letter 'G'.

```

1791 cmchar "CMIN letter G";
1792 beginchar("G", 2sb#+caprat*hstretch*(5/4x_height#),
1793           caprat*x_height#, desc_depth#);
1794 adjust_fit(0,0);
1795 path pth[];
1796 pair vec[];
1797 numeric n[];
1798 pcshiftx := sb;
1799 pickup quill;
1800 n1 := 4/5w - 2pcshiftx; % width of top o
1801 %% top 0
1802 x1 = x3 = pcshiftx + 1/2n1; top y1=h; bot y3=1/8h;
1803 lft x4 = pcshiftx; rt x2 = pcshiftx+n1; y2 = y4 = 1/2[y1,y3];
1804
1805 %% top flick
1806 lft x11 = rt x1; y11=y1;
1807 rt x13 = w-pcshiftx; y13 = h;
1808
1809 %% bottom
1810 z31=z3;
1811 x33 = w-pcshiftx; y33 = -1/2d;
1812 x35=x31; bot y35 = -d;
1813 rt x36 = lft x4; y36 = 1/2[y35,y33];
1814
1815 %%ductus:=true;
1816 if ductus:
1817   pickup stylus;
1818 fi
1819 draw z1..z2..z3..z4..cycle; % top 0
1820 draw z11{right}..z13{pdir}; % flick
1821 draw z31{right}..z33{down}..z35{left}..z36;
1822 ductus:=false;
1823 penlabels(1,2,3,4,5,6,7,8,9,11,12,13,31,32,33,34,35,36,100);
1824 endchar; % end "G"
1825

```

H The letter 'H'.

```

1826 cmchar "CMIN letter H";
1827 beginchar("H", 2sb#+caprat*hstretch*(6/8x_height#+clubjut#),
1828           caprat*asc_height#, 0);
1829 adjust_fit(0,0);
1830 pcshiftx := sb;
1831 pickup quill;
1832 numeric n[];
1833 %% stem
1834 lft x1=lft x2=pcshiftx+jut;
1835 top y1=h; bot y2=0;
1836 %% bowl
1837 z3=1/2[z2,z1];
1838 rt x5 = w-pcshiftx; y5=y3;
1839 x4=1/2[x3,x5]; top y4 = h/ascratio;
1840 %%lft x6=x4; bot y6=0;
1841 x6=1/2[x4,x5];
1842 bot y6=0;
1843 %%ductus:=true;
1844 if ductus:
1845   pickup stylus;
1846 fi
1847 draw_tstem(1,2,srad); % stem
1848 %%draw_flatserif(1,2,srad);
1849 draw_clubserif(1,2,0.5);
1850 draw_roundserif(2,1,srad);
1851 draw z3{pdir}..z4..z5..z6{-pdir};
1852
1853 ductus:=false;
1854 penlabels(1,2,3,4,5,6,100);
1855 endchar; % end "H"
1856

```

I The letter 'I'.

```

1857 cmchar "CMIN letter I";
1858 beginchar("I", 2sb#+caprat*hstretch*(thick#+2jut#),
1859           caprat*x_height#, 0);
1860 adjust_fit(0,0);
1861 pcshiftx := sb;
1862 pickup quill;
1863 lft x1=lft x2=pcshiftx+jut;
1864 top y1=h; bot y2=0;
1865 %%ductus:=true;
1866 draw_tstem(1,2,srad);
1867 draw_flatserif(1,2,srad);
1868 draw_roundserif(2,1,srad);
1869 ductus:=false;
1870 penlabels(1,2,100);
1871 endchar; % end "I"
1872

```

J The letter ‘J’. This letter was not in the script, it being a later invention.

```

1873 cmchar "CMIN letter J";
1874 beginchar("J", sb#+caprat*hstretch*(3/8x_height#),
1875           caprat*x_height#, desc_depth#);
1876 adjust_fit(0,0);
1877 pcshiftx := sb;
1878 numeric n[];
1879 pickup quill;
1880 %% stem
1881 rt x1 = w-pcshiftx; top y1 = h;
1882 x2=x1; y2=-1/2d;
1883 %% bottom hook
1884 lft x3 = 0; bot y3 = -d;
1885 %%ductus:=true;
1886 if ductus:
1887   pickup stylus;
1888 fi
1889 draw z1--z2; % stem
1890 draw_flatserif(1,2,srad);
1891 draw z2{down}..z3{-pdir}; % hook
1892 ductus:=false;
1893 penlabels(1,2,3,100);
1894 endchar; % end "J"
1895

```

K The letter ‘K’.

```

1896 cmchar "CMIN letter K";
1897 beginchar("K", 2sb#+caprat*hstretch*(7/8x_height#+clubjut#),
1898           caprat*asc_height#, 0);
1899 adjust_fit(0,0);
1900 pcshiftx := sb;
1901 path pth[];
1902 numeric n[];
1903 pickup quill;
1904 %% stem
1905 n1 = h/ascratio; % x-height
1906 lft x1=lft x2=pcshiftx+clubjut;
1907 y1=h; bot y2=0;
1908 %% top arm
1909 %%z3=(x1,1/2n1);
1910 x3 = x1; top y3 = 1/2n1;
1911 rt x5=5/8[x1,w-pcshiftx]; top y5=n1;
1912 x4=1/2[x3,x5]; top y4=n1;
1913 pth1 := z3{pdir}..z5{up};
1914 %% lower arm
1915 pth2 := (rt x1,0)--(rt x1,h);
1916 z13 = pth1 intersectionpoint pth2;
1917 pickup threequarterquill;
1918 lft x14 = x13; bot y14 = y13;
1919 rt x7=w-pcshiftx; bot y7=0;

```

```

1920 pickup quill;
1921 %%ductus:=true;
1922 if ductus:
1923   pickup stylus;
1924 fi
1925 draw_tstem(1,2,srad);           % stem
1926 draw_clubserif(1,2,0.5);
1927 draw_roundserif(2,1,srad);
1928 draw pth1;                     % upper arm
1929 pickup threequarterquill;
1930 %%draw z14{down}..z7{-ppdir};   % lower arm
1931 %%draw z14{down}..z7;           % lower arm
1932 draw z14{-ppdir}..z7{right};    % lower arm
1933 pickup quill;
1934 ductus:=false;
1935 penlabels(1,2,3,4,5,6,7,8,9,13,14,100);
1936 endchar; % end "K"
1937

```

L The letter 'L'.

```

1938 cmchar "CMIN letter L";
1939 beginchar("L", 2sb#+caprat*hstretch*(thick#+clubjut#+jut#),
1940           caprat*asc_height#, 0);
1941 adjust_fit(0,0);
1942 pcshiftx := sb;
1943 pickup quill;
1944 lft x1=pcshiftx+clubjut; top y1=h;
1945 x2=x1; bot y2 =0;
1946 %%ductus:=true;
1947 if ductus:
1948   pickup stylus;
1949 fi
1950 draw_tstem(1,2,srad);
1951 draw_clubserif(1,2,0.5);
1952 draw_roundserif(2,1,srad);
1953 ductus:=false;
1954 penlabels(1,2,3,4,100);
1955 endchar; % end "L"
1956

```

M The letter 'M'.

```

1957 cmchar "CMIN letter M";
1958 beginchar("M", 2sb#+caprat*hstretch*(x_height#+2jut#),
1959           caprat*x_height#, 0);
1960 adjust_fit(0,0);
1961 pcshiftx := sb;
1962 numeric n[];
1963 path pth[];
1964 pickup quill;
1965 %% left stem

```

```

1966 lft x1=pcshiftx+jut; top y1=h;
1967 x2=x1; bot y2=0;
1968 %% start of bowl
1969 x13=x1; y13=3/4h;
1970 %% bottom of right and middle stems
1971 rt x6 = w -pcshiftx - jut; y6=y2;
1972 z4 = 1/2[z2,z6];
1973 %% top of middle and right stems
1974 x3=x4; y3=y1;
1975 x5=x6; y5=y13;
1976 %% top of bowls
1977 x14=2/3[x13,x3]; top y14 = h;
1978 x33=x4; y33=y13;
1979 x34=2/3[x33,x5]; y34 = y14;
1980 %%ductus:=true;
1981 if ductus:
1982   pickup stylus;
1983 fi
1984 draw_tstem(1,2,srad); % left half
1985 draw_flatserif(1,2,srad);
1986 draw_roundserif(2,1,srad);
1987 draw z13{pdir}..z14{right}..z33{down}--z4;
1988 draw_roundserif(4,3,srad);
1989 draw z33{pdir}..z34{right}..z5{down}--z6;
1990 draw_roundserif(6,5,srad);
1991 ductus:=false;
1992 penlabels(1,2,3,4,5,6,7,8,9,13,14,33,34,100);
1993 endchar; % end "M"
1994

```

N The letter 'N'.

```

1995 cmchar "CMIN letter N";
1996 beginchar("N", 2sb#+caprat*hstretch*(5/8x_height#+2jut#),
1997           caprat*x_height#, 0);
1998 adjust_fit(0,0);
1999 pcshiftx := sb;
2000 pickup quill;
2001 %% left stem
2002 lft x1=pcshiftx+jut; top y1=h;
2003 x2=x1; bot y2=0;
2004 % start of bowl
2005 x13=x1; y13=3/4h;
2006 %% right stem
2007 rt x4 = w-pcshiftx-jut; y4=y2;
2008 x3 = x4; y3 = y13;
2009 % top of bowl
2010 x14 = 2/3[x13,x3]; top y14 = h;
2011 %%ductus:=true;
2012 if ductus:
2013   pickup stylus;

```

```

2014 fi
2015 draw_tstem(1,2,srad); % left stem
2016 draw_flatserif(1,2,srad);
2017 draw_roundserif(2,1,srad);
2018 draw z13{pdir}..z14{right}..z3{down}--z4;
2019 draw_roundserif(4,3,srad);
2020 ductus:=false;
2021 penlabels(1,2,3,4,5,6,13,14,100);
2022 endchar; % end "N"
2023

O The letter 'O'.

2024 cmchar "CMIN letter O";
2025 beginchar("O", 2sb#+caprat*hstretch*(3/4x_height#),
2026 caprat*x_height#, 0);
2027 adjust_fit(0,0);
2028 pcshiftx := sb;
2029 pickup quill;
2030 lft x1=pcshiftx; rt x3=w-pcshiftx; y1=y3=h/2;
2031 x2=x4=w/2; top y2=h+o; bot y4=-o;
2032 draw z1..z2..z3..z4..cycle;
2033 z5=(pcshiftx,-o);
2034 obowl(5, w-2pcshiftx, h+2o);
2035 %%ductus:=true;
2036 if ductus:
2037 pickup stylus;
2038 draw obfirst; draw obsecond; draw obouter; draw obinner;
2039 else:
2040 %% fill obouter; unfill obinner;
2041 fi
2042 ductus:=false;
2043 penlabels(1,2,3,4,5,100);
2044 endchar; % end "O"
2045

P The letter 'P'.

2046 cmchar "CMIN letter P";
2047 beginchar("P", 2sb#+caprat*hstretch*(3/4x_height#+jut#),
2048 caprat*x_height#, desc_depth#);
2049 adjust_fit(0,0);
2050 pcshiftx := sb;
2051 numeric n[];
2052 pickup quill;
2053 %% stem
2054 lft x1=lft x2=pcshiftx+jut;
2055 top y1=h; bot y2=-d;
2056 %% bowl % from "d"
2057 z3=3/4[(x1,0),z1];
2058 rt x5 = w-pcshiftx; y5=1/2h;
2059 x4=1/2[x3,x5]; top y4 = h;

```

```

2060 x6=x4; bot y6=0;
2061 z7=1/4[(x1,0),z1];
2062
2063 %%ductus:=true;
2064 if ductus:
2065   pickup stylus;
2066 fi
2067 draw_tstem(1,2,srad); % stem
2068 draw_flatserif(1,2,srad);
2069 draw_roundserif(2,1,srad)
2070 draw z3{pdir}..z4{right}..z5{down}..z6{left}..z7; % bowl (from d)
2071 ductus:=false;
2072 penlabels(1,2,3,4,5,6,7,8,100);
2073 endchar; % end "P"
2074

```

Q The letter 'Q'.

```

2075 cmchar "CMIN letter Q";
2076 beginchar("Q", 2sb#+caprat*hstretch*(3/4x_height#+jut#),
2077           caprat*x_height#, desc_depth#);
2078 adjust_fit(0,0);
2079 pcshiftx := sb;
2080 numeric n[];
2081 pickup quill;
2082 %% right stem
2083 rt x1=w-pcshiftx-jut; top y1=h;
2084 x2=x1; bot y2=-d;
2085 %% bowl
2086 z3=1/4[(x1,0),z1];
2087 lft x5=pcshiftx; y5=1/2h;
2088 x4=1/2[x3,x5]; bot y4=0;
2089 x6=x4; top y6=h;
2090 z7=3/4[(x1,0),z1];
2091 %%ductus:=true;
2092 if ductus:
2093   pickup stylus;
2094 fi
2095 draw_tstem(1,2,srad);
2096 %%draw_flatserif(1,2,srad);
2097 draw_roundserif(2,1,srad);
2098 draw z3{-pdir}..z4{left}..z5{up}..z6{right}..z7;
2099 ductus:=false;
2100 penlabels(0,1,2,3,4,5,6,7,8,100);
2101 endchar; % end "Q"
2102

```

R The letter 'R'.

```

2103 cmchar "CMIN letter R";
2104 beginchar("R", 2sb#+caprat*hstretch*(7/8x_height#+jut#),
2105           caprat*x_height#, 0);

```

```

2106 adjust_fit(0,0);
2107 pcshiftx := sb;
2108 numeric n[];
2109 pickup quill;
2110 %% stem
2111 lft x1=lft x2=pcshiftx+jut;
2112 top y1=h; bot y2=0;
2113 %% bowl
2114 z3=3/4[z2,z1];
2115 rt x5 = w-pcshiftx; y5=y3;
2116 x4=1/2[x3,x5]; top y4 = h;
2117 %%ductus:=true;
2118 if ductus:
2119   pickup stylus;
2120 fi
2121 draw_tstem(1,2,srad); % stem
2122 draw_flatserif(1,2,srad);
2123 draw_roundserif(2,1,srad);
2124 draw z3{pdir}..z4{right}..z5{pdir};
2125 ductus:=false;
2126 penlabels(1,2,3,4,5,6,7,8,9,100);
2127 endchar; % end "R"
2128

```

S The letter ‘S’. This is a ‘long’ S.

```

2129 cmchar "CMIN letter S";
2130 beginchar("S", 2sb#+caprat*hstretch*(3/4x_height#+clubjut#),
2131           caprat*asc_height#, 0);
2132 adjust_fit(0,0);
2133 numeric n[];
2134 pcshiftx := sb;
2135 pickup quill;
2136 n1 := h/ascratio; % x-height
2137 %% stem
2138 lft x1=lft x2=pcshiftx+clubjut; top y1=n1; bot y2=0;
2139 %% flick
2140 x4=x2; y4 = 1/2n1;
2141 rt x6=w-pcshiftx; y6 = 2/3h;
2142 x5 = 2/3[x4,x6]; top y5=h;
2143 %%ductus:=true;
2144 if ductus:
2145   pickup stylus;
2146 fi
2147 draw_tstem(1,2,srad); % stem
2148 draw_clubserif(1,2,0.5);
2149 draw_roundserif(2,1,srad);
2150 draw z4{up}...z5{right}...z6;
2151 ductus:=false;
2152 penlabels(1,2,3,4,5,6,7,11,17,20,26,27,28,100);
2153 endchar; % end "S"

```


2154

T The letter 'T'.

```
2155 cmchar "CMIN letter T";
2156 beginchar("T", 2sb#+caprat*hstretch*(x_height#),
2157           caprat*x_height#, 0);
2158 adjust_fit(0,0);
2159 pcshiftx := sb;
2160 numeric n[];
2161 pickup quill;
2162 %% bar
2163 top y3=h; n3 := h - top y3;
2164 lft x1 = pcshiftx; rt x2=w-pcshiftx;
2165 %%top y1 = y3-n3; bot y2=y3+n3;
2166 y1 = y3-n3; y2=y3;
2167 n1 := w;      % width of bowl
2168 n2 := h;      % height of bowl
2169 x3=1/2[x1,x2];
2170 lft x4 = pcshiftx + 1/8n1; y4 = 1/2h;
2171 x5=x3; bot y5 = 0;
2172 rt x6 = pcshiftx + 3/4n1; top y6 = 1/4h;
2173 %%ductus:=true;
2174 if ductus:
2175   pickup stylus;
2176 fi
2177 draw z1{pdir}..z3..z2{pdir};          % bar
2178 draw z3..z4{down}..z5{right}..z6{pdir};
2179 ductus:=false;
2180 penlabels(1,2,3,4,5,6,11,12,100);
2181 endchar; % end "T"
2182
```

U The letter 'U'.

```
2183 cmchar "CMIN letter U";
2184 beginchar("U", 2sb#+caprat*hstretch*(5/8x_height#+2jut#),
2185           caprat*x_height#, 0);
2186 adjust_fit(0,0);
2187 pcshiftx := sb;
2188 numeric n[];
2189 pickup quill;
2190 %% left stem
2191 lft x1 = pcshiftx + jut; top y1 = h;
2192 % start of bowl
2193 x13=x1; y13=1/3h;
2194 % right stem
2195 rt x3=w-pcshiftx-jut; y3=y1;
2196 x4=x3; bot y4=0;
2197 z33=1/4[z4,z3]; % start of bowl
2198 % middle of bowl
2199 x14=1/3[x1,x3]; bot y14=0;
```

```

2200 %%ductus:=true;
2201 if ductus:
2202   pickup stylus;
2203 fi
2204 draw z1--z13{down}..z14{right}..z33{pdir};
2205 draw_flatserif(1,13,srad);
2206 draw_tstem(3,4,srad);           % right stem
2207 draw_flatserif(3,4,srad);
2208 draw_roundserif(4,3,srad);
2209 ductus:=false;
2210 penlabels(1,2,3,4,5,6,7,8,9,11,13,14,16,33,100);
2211 endchar; % end "U"
2212

```

v The letter 'V'.

```

2213 cmchar "CMIN letter V";
2214 beginchar("V", 2sb#+caprat*hstretch*(3/4x_height#+jut#),
2215           caprat*x_height#, 0);
2216 adjust_fit(0,0);
2217 pcshiftx := sb;
2218 pickup quill;
2219 %% left stem
2220 lft x1 = pcshiftx + jut; top y1 = h;
2221 % start of bowl
2222 x13=x1; y13=1/3h;
2223 % right stem
2224 rt x3=w-pcshiftx-jut; y3=y1;
2225 x4=x3; bot y4=0;
2226 z33=1/4[z4,z3]; % start of bowl
2227 % middle of bowl
2228 x14=1/3[x1,x3]; bot y14=0;
2229 rt x23=w-pcshiftx; y23=3/4h;
2230 lft x26=x14; top y26=h;
2231 %%ductus:=true;
2232 if ductus:
2233   pickup stylus;
2234 fi
2235 draw z1--z13{down}..z14{right}; % left stem
2236 draw_flatserif(1,13,srad);
2237 draw z14{pdir}..z23{up}..z26;
2238
2239 ductus:=false;
2240 penlabels(1,2,3,4,5,11,13,14,23,26,100);
2241 endchar; % end "V"
2242

```

w The letter 'W'. This is a modern version as the font did not have a W.

```

2243 cmchar "CMIN letter W";
2244 beginchar("W", 2sb#+caprat*hstretch*(x_height#+2jut#),
2245           caprat*x_height#, 0);

```

```

2246 adjust_fit(0,0);
2247 pcshiftx := sb;
2248 pickup quill;
2249 %% left stem
2250 lft x1 = pcshiftx + jut; top y1 = h;
2251 % start of bowl
2252 x13=x1; y13=1/3h;
2253 % right stem
2254 rt x5=w-pcshiftx-jut; y5=y1;
2255 x6=x5; bot y6=0;
2256 z53=1/4[z6,z5]; % start of bowl
2257 % middle stem
2258 z3 = 1/2[z1,z5];
2259 z33=(x3,y13);
2260 % middle of bowl
2261 x14=1/3[x1,x3]; bot y14=0;
2262 % middle of right bowl
2263 x34=1/3[x3,x5]; y34=y14;
2264 %%ductus:=true;
2265 if ductus:
2266   pickup stylus;
2267 fi
2268 draw z1--z13{down}..z14{right}..z33{pdir}; % left
2269 draw_flatserif(1,13,srad);
2270 draw z3--z33{down}..z34{right}..z53{pdir}; % middle
2271 draw_flatserif(3,33,srad);
2272 draw_tstem(5,6,srad); % right stem
2273 draw_flatserif(5,6,srad);
2274 draw_roundserif(6,5,srad);
2275 ductus:=false;
2276 penlabels(1,2,3,4,5,6,7,8,9,11,13,14,33,34,53,16,100);
2277 endchar; % end "W"
2278

```

X The letter 'X'.

```

2279 cmchar "CMIN letter X";
2280 beginchar("X", 2sb#+caprat*hstretch*(x_height#),
2281          caprat*x_height#, 0);
2282 adjust_fit(0,0);
2283 path pth[];
2284 numeric n[];
2285 pcshiftx := sb;
2286 pickup quill;
2287 n1 := w-2pcshiftx; % total width
2288 n2 := thick; % triangle side
2289 %% diag top left to bottom right
2290 lft x1=pcshiftx; bot y1=2/3h;
2291 x2=pcshiftx+1/7n1; top y2=h;
2292 x5=pcshiftx+11/14n1; bot y5=-1/4d;
2293 rt x6=pcshiftx+n1; top y6=1/6h;

```

```

2294 z25 = 1/3[z2,z5];
2295 pth1 := z2{right}..z5{right};
2296 z251 = point 1/3 of pth1;
2297 z252 = point 2/3 of pth1;
2298 %% diag bottom left to top right
2299 lft x10 = -w + 2pcshiftx; bot y10=-d;
2300 x50=x5; top y50=h;
2301 rt x60 = w -pcshiftx; bot y60=2/3h;
2302 x40=pcshiftx+8/14n1; y40=h;
2303
2304 x150=x2; bot y150 = 0;
2305 lft x160 = pcshiftx; top y160=1/3h;
2306 %%ductus:=true;
2307 if ductus:
2308   pickup stylus;
2309 fi
2310 draw z1..z2{right}..z5{right}..z6;    % top left to bottom right
2311 %draw z10{1/2pdir}..z25..z50{right}..z60;    % bottom left to top right
2312 draw z251..z50{right}..z60;    % middle to top right
2313 draw z252..z150{left}..z160;    % middle to bottom left
2314 ductus:=false;
2315 penlabels(1,2,3,4,5,6,7,8,10,25,251,252,40,50,150,60,160,100);
2316 endchar; % end "X"
2317

```

Y The letter ‘Y’. Modern, but with a dot on top.

```

2318 cmchar "CMIN letter Y";
2319 beginchar("Y", 2sb#+caprat*hstretch*(7/8x_height#+jut#),
2320           caprat*x_height#, desc_depth#);
2321 adjust_fit(0,0);
2322 pcshiftx := sb;
2323 path pth[];
2324 numeric n[];
2325 pickup quill;
2326 %% left stem top
2327 lft x1 = pcshiftx + jut; top y1 = h;
2328 bot y3 = 0;
2329 %% right stem
2330 rt x5=w-pcshiftx; top y5 = h;
2331 lft x9=-1/2w; bot y9 = -d;
2332 pth1 := z9{right}..z5{up};
2333 pth2 := (0,y3)--(w,y3);
2334 z99 = pth1 intersectionpoint pth2;
2335 x3 = x99;
2336 %%ductus:=true;
2337 if ductus:
2338   pickup stylus;
2339 fi
2340 draw z1{down}..z3;          % left stem
2341 draw_flatserif(1,3,srad);

```

```

2342 draw pth1; % right stem
2343 draw_flatserif(5,3,srad);
2344 ductus:=false;
2345 penlabels(1,2,3,4,5,6,7,8,9,10,100);
2346 endchar; % end "Y"
2347

```

Z The letter 'Z'.

```

2348 cmchar "CMIN letter Z";
2349 beginchar("Z", 2sb#+caprat*hstretch*(x_height#),
2350           caprat*x_height#, 0);
2351 adjust_fit(0,0);
2352 numeric n[];
2353 path pth[];
2354 pcshiftx := sb;
2355 pickup quill;
2356 %% top bar ends
2357 top y12 = h;
2358 n1 := h - y12;
2359 lft x1 = pcshiftx; rt x2 = w-pcshiftx;
2360 %%y1 = y12-n1; y2=y12+n1;
2361 y1 = y12-n1; y2=y12;
2362 x12 = 1/2[x1,x2];
2363 pth1 := z1{pdir}..z12{right}..z2{pdir};
2364 %% bottom bar ends
2365 bot y34 = 0;
2366 x3=x1; x4=x2;
2367 %%y3=y34-n1; y4=y34+n1;
2368 y3=y34-n1; y4=y34;
2369 x34 = 1/2[x3,x4];
2370 pth2 := z3{pdir}..z34{right}..z4{pdir};
2371 %% diagonal
2372 z5=z3;
2373 pth3 := z5--(3/4w,h+jut);
2374 z6 = pth3 intersectionpoint pth1;
2375 %%ductus:=true;
2376 if ductus:
2377   pickup stylus;
2378 fi
2379 draw pth1; draw pth2; % bars
2380 draw z5--z6;
2381 ductus:=false;
2382 penlabels(1,2,3,4,5,6,10o,10i,20o,20i,100);
2383 endchar; % end "Z"
2384

```

The end of this file

```

2385 </maj>

```

3.5 The punctuation file

The main punctuation marks are defined in this file.

```

2386 ⟨*punct⟩
2387 % IMINPUNCT.MF This file contains punctuation marks
2388
! The '!' character, which had not been invented at this time.
2389
2390 cmchar "Exclamation mark";
2391 beginchar("!", 2sb#+hstretch*(thick#+trijut#),
2392          asc_height#, 0);
2393 adjust_fit(0,0);
2394 pcshiftx := sb;
2395 pickup quill;
2396 x1=w/2; top y1 = h;
2397 z3=(x1, 1/2dot_size);
2398 z2=(x1, y3+3/2dot_size);
2399 draw z1--z2;
2400 draw_pdot(3,dot_size);
2401 penlabels(1,2,3);
2402
2403 endchar; % end of "!"
2404
? The '?' character, which had not been invented at this time.
2405
2406 cmchar "Question mark";
2407 beginchar("?", 2sb#+hstretch*(3/4x_height#),
2408          asc_height#, 0);
2409 adjust_fit(0,0);
2410 pcshiftx := sb;
2411 numeric n[];
2412 path pth[];
2413 pickup quill;
2414 n1 := w-2pcshiftx; % actual character width
2415 n2 := 1/3h; % height of top bowl
2416 n3 := n2; % height of bottom bowl
2417 z5=(1/2w,1/2dot_size);
2418 x4 = x5; y4=y5+3/2dot_size;
2419 lft x1 = pcshiftx; y1 = 3/4h;
2420 x2 = 1/2w; top y2 = h;
2421 rt x3 = w-pcshiftx; y3 = 2/3h;
2422
2423 %%ductus:= true;
2424 if ductus:
2425   pickup stylus;
2426 fi
2427 draw z1..z2{right}..z3{down}..z4{down};

```

```

2428 draw_pdot(5, dot_size);
2429 penlabels(1,2,3,4,5);
2430 endchar; % end of "?"
2431

```

, The ‘,’ character.

```

2432
2433 cmchar "Comma";
2434 beginchar(", ", 2sb#+hstretch*(comma_width#+dot_size#), x_height#, 0);
2435 adjust_fit(0, 0);
2436 pcshiftx:=sb;
2437 x1=w-pcshiftx-(0.5hstretch*dot_size); y1=1/2dot_size;
2438 draw_pcomma(1, dot_size, comma_width);
2439 penlabels(1);
2440 endchar; % end of ", "
2441

```

. The ‘.’ character.

```

2442
2443 cmchar "Period";
2444 beginchar(".", 2sb#+hstretch*(dot_size#), x_height#, 0);
2445 pcshiftx:=sb;
2446 z1=(w/2, 1/2dot_size);
2447 draw_pdot(1, dot_size);
2448 penlabels(1);
2449 endchar; % end of "."
2450

```

: The ‘:’ character.

```

2451
2452 cmchar "Colon";
2453 beginchar(":", 2sb#+hstretch*(dot_size#), x_height#, 0);
2454 pcshiftx:=sb;
2455 x1=x2=w/2; y1=2/3h; y2=1/3h;
2456 draw_pdot(1, dot_size);
2457 draw_pdot(2, dot_size);
2458 penlabels(1, 2);
2459 endchar; % end of ":"
2460

```

; The ‘;’ character.

```

2461
2462 cmchar "Semicolon";
2463 beginchar(";", 2sb#+hstretch*(comma_width#+dot_size#), x_height#, 0);
2464 adjust_fit(0, 0);
2465 pcshiftx:=sb;
2466 x1=x2=w-pcshiftx-(0.5hstretch*dot_size); y1=2/3h; y2=1/3h;
2467 draw_pdot(1, dot_size);
2468 draw_pcomma(2, dot_size, comma_width);

```

```

2469 penlabels(1,2);
2470 endchar; % end of ","
2471
' The " character.
2472
2473 cmchar "Apostrophe";
2474 beginchar("'",2sb#+hstretch*(comma_width#+dot_size#),4/3x_height#,0);
2475 adjust_fit(0,0);
2476 pcshiftx:=sb;
2477 x1=w-pcshiftx-(0.5hstretch*dot_size); y1=h;
2478 draw_pcomma(1,dot_size,comma_width);
2479 penlabels(1);
2480 endchar; % end of "'"
2481
' The " character.
2482
2483 cmchar "Reverse apostrophe";
2484 beginchar(" ",2sb#+hstretch*(comma_width#+dot_size#),4/3x_height#,0);
2485 adjust_fit(0,0);
2486 pcshiftx:=sb;
2487 x1=pcshiftx+(0.5hstretch*dot_size); y1=h;
2488 draw_plq(1,dot_size,comma_width);
2489 penlabels(1);
2490 endchar; % end of " "
2491
The end of this file.
2492 </punct>

```

3.6 The ligatures and dashes file

The ligatured letters, punctuation marks and dashes are defined in this file.

```

2493 <*lig>
2494 % CMINLIG.MF Carolingian ligatured letters, punctuation and dashes
2495
First the ligtables.
2496
2497 ligtable " ": " "=:oct"134";
2498 ligtable " ' ": " ' "=:oct"042"; %% , "?" kern 2u#, "!" kern 2u#;
2499
2500 ligtable "-": "- "=:oct"173";
2501 ligtable oct"173": "- "=:oct"174";
2502
2503 ligtable "c": "t "=:oct"014";
2504 ligtable "s": "t "=:oct"015";
2505

```


ct The 'ct' ligature.

```

2506
2507 cmchar "The ct ligature";
2508 beginchar(oct"014", 2sb#+hstretch*(6/4x_height#),
2509          asc_height#, 0);
2510 adjust_fit(0,0);
2511 pcshiftx := sb;
2512 numeric n[];
2513 path pth[];
2514 pair vc[];
2515 pickup quill;
2516 %%%%%%%%% C
2517
2518 n1:=1/2(w-2pcshiftx);      % actual letter width
2519 n2 := h/ascratio;         % x-height
2520 x1=pcshiftx + 1/2n1; top y1 = n2;
2521 lft x2 = pcshiftx; y2 = 1/2n2;
2522 x3 = x1; bot y3 = 0;
2523 rt x4 = n1; y4 = 1/4n2;
2524 x6 = x4;
2525 %y6 = 3/4n2;
2526 top y6 = y1;
2527 pth1 := z6..z1{left}..z2{down}..z3{right}..z4{pdir};
2528 %%%%%%%%% T
2529
2530 % bar
2531 top y33=n2; n3 := n2 - top y33;
2532 %lft x31 = n1+pcshiftx;
2533 lft x31 = n1;
2534 rt x32=w-pcshiftx;
2535 %top y31 = y33-n3; bot y32=y33+n3;
2536 top y31 = y33-n3; y32=y33;
2537 x33=1/2[x31,x32];
2538 pth2 := z31{pdir}..z33..z32{pdir};          % T bar
2539 n31 := n1;      % width of bowl
2540 %lft x34 = pcshiftx + n1 + 1/8n31;
2541 lft x34 = n1 + 1/8n31;
2542 y34 = 1/2n2;
2543 x35=x33; bot y35 = 0;
2544 rt x36 = pcshiftx + n1 + 3/4n31; top y36 = 1/4n2;
2545 pth3 := z33..z34{down}..z35{right}..z36{pdir}; % T bowl
2546
2547 %%%%%%%%% Ligature
2548
2549 z51=z6;
2550 z55=z33;
2551 x53 = 1/2[x51,x55];
2552 top y53 = h;
2553 vc51 := direction 0 of pth1;

```

```

2554 vc55 := direction 0 of pth3;
2555 pth4 := z51{vc51}..z53{right}..z55{vc55}; % loop
2556 %%ductus:=true;
2557 if ductus:
2558   pickup stylus;
2559 fi
2560 draw pth1;          % C
2561 draw pth2;          % T bar
2562 draw pth3;          % T bowl
2563 draw pth4;          % ligature loop
2564 ductus:=false;
2565 penlabels(1,2,3,4,5,6,14,16,17,18,31,32,33,34,35,36,100);
2566 endchar; % end of "ct"
2567

```

st The 'st' ligature.

```

2568
2569 cmchar "The st ligature";
2570 beginchar(oct"015", 2sb#+hstretch*(9/8x_height#+clubjut#),
2571           asc_height#, 0);
2572 adjust_fit(0,0);
2573 pcshiftx := sb;
2574 numeric n[];
2575 path pth[];
2576 pair vc[];
2577 pickup quill;
2578 %%%%%%%%% S
2579
2580 n1:=3/7(w-2pcshiftx-clubjut); % actual letter width
2581 n2 := h/ascratio; % x-height
2582
2583 %% stem
2584 lft x1=lft x2=pcshiftx+clubjut; top y1=n2; bot y2=0;
2585 %% flick
2586 x4=x2; y4 = 1/2n1;
2587 %%%%%%%%% T
2588
2589 %% bar
2590 top y33=n2; n3 := n2 - top y33;
2591 %lft x31 = n1+pcshiftx;
2592 lft x31 = n1;
2593 rt x32=w-pcshiftx;
2594 %top y31 = y33-n3; bot y32=y33+n3;
2595 top y31 = y33-n3; y32=y33;
2596 x33=1/2[x31,x32];
2597 pth2 := z31{pdir}..z33..z32{pdir}; % T bar
2598 n31 := n1; % width of bowl
2599 %lft x34 = pcshiftx + n1 + 1/8n31;
2600 lft x34 = n1 + 1/8n31;
2601 y34 = 1/2n2;

```

```

2602 x35=x33; bot y35 = 0;
2603 rt x36 = pcshiftx + n1 + 3/4n31; top y36 = 1/4n2;
2604 pth3 := z33..z34{down}..z35{right}..z36{pdir}; % T bowl
2605
2606 %%%%%%%%%%% Ligature
2607
2608 z51=z4;
2609 z55=z33;
2610 x53 = 5/8[x51,x55];
2611 top y53 = h;
2612 vc51 := up;
2613 vc55 := direction 0 of pth3;
2614 pth4 := z51{vc51}...z53{right}..z55{vc55}; % loop
2615 %ductus:=true;
2616 if ductus:
2617   pickup stylus;
2618 fi
2619 draw z1--z2; % S stem
2620 draw_clubserif(1,2,0.5);
2621 draw_roundserif(2,1,srad);
2622 draw pth2; % T bar
2623 draw pth3; % T bowl
2624 draw pth4; % ligature loop
2625 ductus:=false;
2626 penlabels(1,2,3,4,5,6,14,16,17,18,31,32,33,34,35,36,100);
2627 endchar; % end of "st"
2628

```

’, The ‘’ ligature.

```

2629
2630 cmchar "Closing quotes";
2631 beginchar(oct"042",2sb#+hstretch*(comma_width#+3dot_size#),4/3x_height#,0);
2632 adjust_fit(0,0);
2633 pcshiftx:=sb;
2634 x1=w-pcshiftx-(0.5hstretch*dot_size); y1=y2=h;
2635 x2=x1-(2hstretch*dot_size);
2636 draw_pcomma(1,dot_size,comma_width); % right
2637 draw_pcomma(2,dot_size,comma_width); % left
2638 penlabels(1,2);
2639 endchar; % end of "''"
2640

```

‘‘ The ‘‘ ligature.

```

2641
2642 cmchar "Opening quotes";
2643 beginchar(oct"134",2sb#+hstretch*(comma_width#+3dot_size#),4/3x_height#,0);
2644 adjust_fit(0,0);
2645 pcshiftx:=sb;
2646 x1=pcshiftx+(0.5hstretch*dot_size); y1=y2=h;
2647 x2=x1+(2hstretch*dot_size);

```

```

2648 draw_plq(1,dot_size,comma_width); % left
2649 draw_plq(2,dot_size,comma_width); % right
2650 penlabels(1,2);
2651 endchar; % end of "‘"
2652

```

- The ‘-’ (hyphen) character.

```

2653
2654 cmchar "Hyphen";
2655 beginchar("-",sb#+hstretch*(0.8x_height#),x_height#,0);
2656 adjust_fit(0,0);
2657 pcshiftx:=sb/2;
2658 numeric alpha; alpha:=w/8;
2659 %%pickup quill;
2660 %%lft x1 = pcshiftx; rt x2=w-pcshiftx; y1=y2=h/2;
2661 %%x11=lft x1+alpha/2; y11=top y1-0.5thin;
2662 %%x12=rt x2; y12=bot y1+0.5thin;
2663 %%draw_triangle(11,alpha);
2664 %%draw_rutriangle(12,alpha);
2665 %%pickup thickstylus;
2666 pickup halfquill;
2667 lft x21=pcshiftx+jut; rt x22=w-pcshiftx-jut; y21=y22=1/2h;
2668 draw z21--z22;
2669 draw_roundserif(21,22,srad); draw_flatserif(22,21,srad);
2670 penlabels(1,2);
2671 endchar; % end of "-"
2672

```

-- The ‘--’ (en-dash) ligature.

```

2673
2674 cmchar "En dash";
2675 beginchar(oct"173",sb#+hstretch*(x_height#),x_height#,0);
2676 adjust_fit(0,0);
2677 pcshiftx:=sb/2;
2678 numeric alpha; alpha:=w/8;
2679 pickup quill;
2680 %%lft x1 = pcshiftx; rt x2=w-pcshiftx; y1=y2=h/2;
2681 %%x11=lft x1+alpha/2; y11=top y1-0.5thin;
2682 %%x12=rt x2; y12=bot y1+0.5thin;
2683 %%draw_triangle(11,alpha);
2684 %%draw_rutriangle(12,alpha);
2685 %%pickup thickstylus;
2686 pickup halfquill;
2687 lft x21=pcshiftx+jut; rt x22=w-pcshiftx-jut; y21=y22=1/2h;
2688 draw z21--z22;
2689 draw_roundserif(21,22,srad); draw_flatserif(22,21,srad);
2690 penlabels(1,2);
2691 endchar; % end of "--"
2692

```

--- The ‘—’ (em dash) ligature. The em dash is twice the width of the en dash.

```

2693
2694 cmchar "Em dash";
2695 beginchar(oct"174",sb#+hstretch*(2x_height#),x_height#,0);
2696 adjust_fit(0,0);
2697 pcshiftx:=sb/2;
2698 numeric alpha; alpha:=w/8;
2699 pickup quill;
2700 %%lft x1 = pcshiftx; rt x2=w-pcshiftx; y1=y2=h/2;
2701 %%x11=lft x1+alpha/2; y11=top y1-0.5thin;
2702 %%x12=rt x2; y12=bot y1+0.5thin;
2703 %%draw_triangle(11,alpha);
2704 %%draw_rutriangle(12,alpha);
2705 %%pickup thickstylus;
2706 pickup halfquill;
2707 lft x21=pcshiftx+jut; rt x22=w-pcshiftx-jut; y21=y22=1/2h;
2708 draw z21--z22;
2709 draw_roundserif(21,22,srad); draw_flatserif(22,21,srad);
2710 penlabels(1,2);
2711 endchar; % end of "---"
2712

```

& The ampersand (which is really an ‘et’ ligature).

```

2713
2714 cmchar "CMIN ampersand";
2715 beginchar("&", 2sb#+hstretch*(5/4x_height#+2jut#), asc_height#, 0);
2716 adjust_fit(0,0);
2717 numeric n[];
2718 path pth[];
2719 pcshiftx:=sb;
2720 pickup quill;
2721 %% alpha looking part
2722 n1 := w-2pcshiftx-jut; % actual letter width
2723 n2 := h/ascratio; % x-height
2724 n3 := 16/16n2; % height of alpha
2725 x1 = w-pcshiftx; bot y1 = 0;
2726 x3 = pcshiftx + 12/16n1; y3 = 1/2n3; % 10/16 too small
2727 lft x5 = pcshiftx; y5=y3;
2728 x4 = x6 = 1/2[x5,x3];
2729 top y4 = n3; bot y6 = 0;
2730 rt x8 = w-pcshiftx-jut; top y8 = h;
2731 x9 = x8-2jut; x10 = x8+2jut;
2732 y9 = y10 = y8;
2733 pth1 := z1{left}..z3..z4{left}..z5{down}..z6{right}..z3..z8{up};
2734 %% loop
2735 z21 = z5;
2736 z25 = point 1.5 of pth1;
2737 x23 = 1/2[x21,x25];
2738 y23 = 6/8[y4,y8];
2739 pth2 := z21{up}..z23{right}..z25{down};

```

```

2740 %%ductus:=true;
2741 if ductus:
2742   pickup stylus;
2743 fi
2744 draw pth1; draw z9--z10;           % alpha part
2745 draw pth2;                         % loop
2746 penlabels(1,2,3,4,5,6,7,8,9,10,16,17,18,19,21,22,23,24,25);
2747 endchar; % end of "&"
2748

```

The end of this file.

```

2749 </lig>

```

3.7 The digits file

These are arabic digits in the style of the book-hand. They are a modern invention.

```

2750 <*dig>
2751 % CMINDIG.MF Program file for Carolingian Minuscule digits
2752 %
2753

```

1 The digit '1'.

```

2754 cmchar "Carolingian digit 1";
2755 beginchar("1", 2u#+fig_width#, fig_height#, 0);
2756 adjust_fit(0,0);
2757 pcshiftx := u;
2758 y100 = h/2; x100 = pcshiftx+y100; % square center
2759 pickup quill;
2760 x1=x2=w/2;
2761 y1=h; bot y2=0;
2762 %%ductus:=true;
2763 draw_tstem(1,2,srad);
2764 draw_flatserif(1,2,srad);
2765 draw_roundserif(2,1,srad);
2766 ductus:=false;
2767 proofpcbb(100, y100, y100);
2768 penlabels(1,2,100);
2769 endchar; % end "1"
2770

```

2 The digit '2'.

```

2771 cmchar "Carolingian digit 2";
2772 beginchar("2", 2u#+fig_width#, fig_height#, 0);
2773 adjust_fit(0,0);
2774 pcshiftx := u;
2775 pickup quill;
2776 % stem
2777 lft x1=pcshiftx+o; top y1=h-2thin;
2778 x2=1/2w; top y2= h+o;

```

```

2779 rt x3=w-pcshiftx-o; y3=3/4h;
2780 % foot
2781 lft x5=pcshiftx; bot y5=0;
2782 rt x6=w-pcshiftx; y6=y5;
2783 %%ductus:=true;
2784 if ductus:
2785   pickup stylus;
2786 fi
2787 draw_wave(5,6,thin); % foot
2788 draw z1..z2{right}..z3{down}..{-pdir}z5wl; % curve
2789 ductus:=false;
2790 penlabels(1,2,3,4,5,6,100);
2791 endchar; % end "2"
2792

```

3 The digit '3'.

```

2793 cmchar "Carolingian digit 3";
2794 beginchar("3", 2u#+fig_width#, fig_height#, 0);
2795 adjust_fit(0,0);
2796 pcshiftx := u;
2797 pickup quill;
2798 % x values
2799 lft x3=pcshiftx; x10=x3;
2800 rt x5=w-pcshiftx; x8=x5-3thin;
2801 x4=0.5[x3,x5];
2802 x9=0.5[x10,x8];
2803 %lft x6=pcshiftx+3thin; x7=x6;
2804 x6=1/2[x3,x9]; x7=x6;
2805 % y values
2806 bot y3=thin; top y10=h-2thin;
2807 y6=0.5[y3,y10]; y7=y6;
2808 bot y4=-thin; top y9=h+thin;
2809 y5=0.5[y3,y6];
2810 y8=11/12[y7,y10];
2811 %%ductus:=true;
2812 if ductus:
2813   pickup stylus;
2814 fi
2815 %%draw z3..z4{right}..z5{up}..{-pdir}z6; % bottom bowl
2816 %%draw z7{pdir}..z8{up}..z9{left}..z10; % top bowl
2817 draw z3..z4{right}..z5{up}..{-left}z6; % bottom bowl
2818 draw z7{right}..z8{up}..z9{left}..z10; % top bowl
2819 ductus:=false;
2820 penlabels(1,2,3,4,5,6,7,8,9,10,100);
2821 endchar; % end "3"
2822

```

4 The digit '4'.

```

2823 cmchar "Carolingian digit 4";
2824 beginchar("4", 2u#+fig_width#, fig_height#, 0);

```

```

2825 adjust_fit(0,0);
2826 pcshiftx := u;
2827 pickup quill;
2828 % the stem
2829 x1=x2=w-pcshiftx-jut;
2830 y1=h; bot y2=0;
2831 % the crossbar
2832 lft x3=pcshiftx; y3=0.33h;
2833 rt x4=w-pcshiftx; y4=y3;
2834 %%ductus:=true;
2835 if ductus:
2836   pickup stylus;
2837 fi
2838 draw_tstem(1,2,srad);           % main stem
2839 draw_wave(3,4,thin);          % crossbar
2840 draw z3wl--z1;                 % diagonal
2841 ductus:=false;
2842 penlabels(1,2,3,4,100);
2843 endchar; % end "4"
2844

```

5 The digit '5'.

```

2845 cmchar "Carolingian digit 5";
2846 beginchar("5", 2u#+fig_width#, fig_height#, 0);
2847 adjust_fit(0,0);
2848 pcshiftx := u;
2849 pickup quill;
2850 % for the top
2851 rt x1=w-pcshiftx; top y1=h;
2852 lft x2=pcshiftx; y2=y1;
2853 % the S curve
2854 x3=x2; y3=0.5h+thin;
2855 x4=0.5[x1,x3]; y4=0.5[y3,y5];
2856 rt x5=w-pcshiftx; y5=0.3h;
2857 x6=w/2; bot y6=0;
2858 lft x7=pcshiftx; bot y7=2thin;
2859 %%ductus:=true;
2860 if ductus:
2861   pickup stylus;
2862 fi
2863 draw_wave(2,1,thin);           % top squiggle
2864 draw z3{pdir}..z5{down}..z6{left}..z7; % curve
2865 draw z2wl--z3;                 % vertical
2866 ductus:=false;
2867 penlabels(1,2,3,4,5,6,7,100);
2868 endchar; % end "5"
2869

```

6 The digit '6'.


```

2870 cmchar "Carolingian digit 6";
2871 beginchar("6", 2u#+fig_width#, fig_height#, 0);
2872 adjust_fit(0,0);
2873 pcshiftx := u;
2874 pickup quill;
2875 rt x1=w-pcshiftx; top y1=h-2thin;
2876 x2=1/2w; top y2= h;
2877 lft x3=pcshiftx; y3=1/2h;
2878 x4=x2; bot y4=0;
2879 rt x5=w-pcshiftx; y5=1/4h;
2880 lft x6=rt x3; bot y6=y3;
2881 %%ductus:=true;
2882 if ductus:
2883   pickup stylus;
2884 fi
2885 draw z1..z2{left}..z3{down}..z4{right}..z5{up}..{-pdir}z6;
2886 ductus:=false;
2887 penlabels(1,2,3,4,5,6,100);
2888 endchar; % end "6"
2889

```

7 The digit '7'.

```

2890 cmchar "Carolingian digit 7";
2891 beginchar("7", 2u#+fig_width#, fig_height#, 0);
2892 adjust_fit(0,0);
2893 pcshiftx := u;
2894 pickup quill;
2895 % top bar
2896 lft x1=pcshiftx;
2897 rt x2=w-pcshiftx;
2898 top y1=top y2=h;
2899 % bowed leg
2900 x4=w/2; bot y4=0;
2901 x3=0.5[x2,x4]-3thin; y3=0.5[y2,y4];
2902 %%ductus:=true;
2903 if ductus:
2904   pickup stylus;
2905 fi
2906 draw_wave(1,2,thin);           % top bar
2907 draw z1wr..z3..z4;
2908 ductus:=false;
2909 penlabels(1,2,3,4,5,6,100);
2910 endchar; % end "7"
2911

```

8 The digit '8'.

```

2912 cmchar "Carolingian digit 8";
2913 beginchar("8", 2u#+fig_width#, fig_height#, 0);
2914 adjust_fit(0,0);
2915 pcshiftx := u;

```

```

2916 numeric alpha; alpha:=1;
2917 pickup quill;
2918 lft x1=alpha*pcshiftx+o; rt x3=w-alpha*pcshiftx-o; y1=y3=3/4h;
2919 x2=x4=w/2; top y2=h+o; bot y4=-o;
2920 x5=x2; y5=h/2+2o;
2921 lft x11=pcshiftx; rt x13=w-pcshiftx; y11=y13=1/4h;
2922 %%ductus:=true;
2923 if ductus:
2924   pickup stylus;
2925 fi
2926 draw z1..z2..z3..z5..z11..z4..z13..z5..cycle;
2927 ductus:=false;
2928 penlabels(1,2,3,4,5,11,13,100);
2929 endchar; % end "8"
2930

```

9 The digit '9'.

```

2931 cmchar "Carolingian digit 9";
2932 beginchar("9", 2u#+fig_width#, fig_height#, 0);
2933 adjust_fit(0,0);
2934 pcshiftx := u;
2935 pickup quill;
2936 lft x1=pcshiftx; bot y1=2thin;
2937 x2=1/2w; bot y2= 0;
2938 rt x3=w-pcshiftx; y3=1/2h;
2939 x4=x2; top y4=h;
2940 lft x5=pcshiftx; y5=3/4h;
2941 rt x6=lft x3; top y6=y3;
2942 %%ductus:=true;
2943 if ductus:
2944   pickup stylus;
2945 fi
2946 draw z1..z2{right}..z3{up}..z4{left}..z5{down}..{pdir}z6;
2947 ductus:=false;
2948 penlabels(1,2,3,4,5,6,100);
2949 endchar; % end "9"
2950

```

0 The digit '0'.

```

2951 cmchar "Carolingian digit 0";
2952 beginchar("0", 2u#+fig_width#, fig_height#, 0);
2953 adjust_fit(0,0);
2954 pcshiftx := u;
2955 y100 = h/2; x100 = pcshiftx+y100; % square center
2956 pickup quill;
2957 lft x1=pcshiftx; rt x3=w-pcshiftx; y1=y3=h/2;
2958 x2=x4=w/2; top y2=h+o; bot y4=-o;
2959 %%ductus:=true;
2960 if ductus:
2961   pickup stylus;

```

```

2962 fi
2963 draw z1..z2..z3..z4..cycle;
2964 ductus:=false;
2965 proofpcbb(100, y100, y100);
2966 penlabels(1,2,3,4,100);
2967 endchar; % end "0"
2968

```

The end of this file.

```
2969 </dig>
```

4 The font definition files

The font comes in normal and bold weights only.

```

2970 <*fdotl>
2971 \DeclareFontFamily{OT1}{cmin}{
2972   \DeclareFontShape{OT1}{cmin}{m}{n}{ <-8.5> cmin7 <8.5-15> cmin10 <15-> cmin17 }{}
2973   \DeclareFontShape{OT1}{cmin}{m}{sl}{ <-> sub * cmin/m/n }{}
2974   \DeclareFontShape{OT1}{cmin}{m}{it}{ <-> sub * cmin/m/n }{}
2975   \DeclareFontShape{OT1}{cmin}{m}{sc}{ <-> sub * cmin/m/n }{}
2976   \DeclareFontShape{OT1}{cmin}{m}{u}{ <-> sub * cmin/m/n }{}
2977   \DeclareFontShape{OT1}{cmin}{bx}{n}{ <-8.5> cminb7 <8.5-15> cminb10 <15-> cminb17 }{}
2978   \DeclareFontShape{OT1}{cmin}{bx}{it}{ <-> sub * cmin/bx/n }{}
2979   \DeclareFontShape{OT1}{cmin}{bx}{sl}{ <-> sub * cmin/bx/n }{}
2980   \DeclareFontShape{OT1}{cmin}{b}{n}{ <-> sub * cmin/bx/n }{}
2981 </fdotl>

2982 <*fdtl>
2983 \DeclareFontFamily{T1}{cmin}{
2984   \DeclareFontShape{T1}{cmin}{m}{n}{ <-8.5> cmin7 <8.5-15> cmin10 <15-> cmin17 }{}
2985   \DeclareFontShape{T1}{cmin}{m}{sl}{ <-> sub * cmin/m/n }{}
2986   \DeclareFontShape{T1}{cmin}{m}{it}{ <-> sub * cmin/m/n }{}
2987   \DeclareFontShape{T1}{cmin}{m}{sc}{ <-> sub * cmin/m/n }{}
2988   \DeclareFontShape{T1}{cmin}{m}{u}{ <-> sub * cmin/m/n }{}
2989   \DeclareFontShape{T1}{cmin}{bx}{n}{ <-8.5> cminb7 <8.5-15> cminb10 <15-> cminb17 }{}
2990   \DeclareFontShape{T1}{cmin}{bx}{it}{ <-> sub * cmin/bx/n }{}
2991   \DeclareFontShape{T1}{cmin}{bx}{sl}{ <-> sub * cmin/bx/n }{}
2992   \DeclareFontShape{T1}{cmin}{b}{n}{ <-> sub * cmin/bx/n }{}
2993 </fdtl>

```

5 The package code

5.1 The carolmin package

The carolmin package is for typesetting short pieces of text in the Carolingian Minuscul fonts.

Announce the name and version of the package, which requires L^AT_EX 2_ε.

```

2994 <*usc>
2995 \NeedsTeXFormat{LaTeX2e}
2996 \ProvidesPackage{carolmin}[2002/12/30 v1.0 package for Carolingian Minuscule fonts]

\Tienc  A macro for testing the value of \encodingdefault.
2997 \providecommand{\Tienc}{T1}

\cminfamily  Selects the Carolingian Minuscule font family in the OT1 or T1 encoding.
2998 \ifx\Tienc\encodingdefault
2999   \newcommand{\cminfamily}{\usefont{T1}{cmin}{m}{n}}
3000 \else
3001   \newcommand{\cminfamily}{\usefont{OT1}{cmin}{m}{n}}
3002 \fi

\textcmin  Text command for the Carolingian Minuscule font family.
3003 \DeclareTextFontCommand{\textcmin}{\cminfamily}

The end of this package.
3004 </usc>

```

5.2 The allcmin package

The allcmin package is intended for use when the entire document will be typeset in the book-hand. This is a minimal package. Because the book-hand should not contain arabic digits a fuller package would redefine anything numbered in L^AT_EX to use roman instead of arabic numbering. Also, because of the large `\baselineskip` many other aspects of L^AT_EX to do with vertical positioning should also be redefined.

Announce the name and version of the package, which requires L^AT_EX 2_ε. It also uses the carolmin package.

```

3005 <*uscall>
3006 \NeedsTeXFormat{LaTeX2e}
3007 \ProvidesPackage{allcmin}[2002/12/30 v1.0 package for all Carolingian Minuscule fonts]
3008 \RequirePackage{carolmin}

\Tienc  A macro for testing the value of \encodingdefault.
3009 \providecommand{\Tienc}{T1}

Redefine the default fonts to be Carolingian Minuscule, which has only one
family member.

\rmdefault
\sfdefault 3010 \renewcommand{\rmdefault}{cmin}
\ttdefault 3011 \renewcommand{\sfdefault}{cmin}
3012 \renewcommand{\ttdefault}{cmin}

```

Table 2: The x heights of the CMR and Carolingian Minuscule fonts

Design Size	CMR X Height	Carolingian Minuscule X Height
5	2.14	3.00
6	2.58	3.61
7	3.00	4.20
8	3.44	4.81
9	3.86	5.40
10	4.31	6.03
11	4.74	6.64
12	5.17	7.24
14	6.03	8.44
17	7.44	10.41
20	8.75	12.25
25	10.94	15.32

The ‘leading’ in the book-hands is a function of the **x-height**, which is normally much greater than for the CMR fonts. I have borrowed code from Frank Jensen’s **beton** package [Jen95] to do this. The **x-height** (in points) for the CMR fonts is given in Table 2. Note that the CMR design sizes are 5–10, 12, and 17.28 points. The other values given are scaled from these figures.

`\cmin@baselineskip@table` A table of the normal font sizes and the corresponding baselineskip. The distance between baselines for Carolingian Minuscule is over twice the **x-height**.

```

3013 \newcommand{\cmin@baselineskip@table}{%
3014   <\@vpt>7.5%
3015   <\@vipt>9.0%
3016   <\@viipt>10.5%
3017   <\@viiipt>12.0%
3018   <\@ixpt>13.5%
3019   <\@xpt>15.0%
3020   <\@xipt>16.5%
3021   <\@xiipt>18.0%
3022   <\@xivpt>21.0%
3023   <\@xvipt>26.0%
3024   <\@xxpt>30.5%
3025   <\@xxvpt>38.5}
```

`\cmin@new@setfontsize` This is a macro that replaces the `\@setfontsize` macro which is called by the font size changing commands.

```

3026 \newcommand{\cmin@new@setfontsize}[3]{%
3027   \edef\@tempa{\def\noexpand\@tempb###1<#2}%
3028   \@tempa>##2<##3\@nil{\def\cmin@baselineskip@value{##2}}}%
3029   \edef\@tempa{\noexpand\@tempb\cmin@baselineskip@table<#2}%
3030   \@tempa><\@nil
3031   \ifx\cmin@baselineskip@value\@empty
```

```

3032     \def\cmin@baselineskip@value{#3}%
3033     \fi
3034     \cmin@old@setfontsize{#1}{#2}\cmin@baselineskip@value}

```

We had better give an author a means of using The Computer Modern fonts if necessary.

`\cmrfamily` These macros select the Computer Modern Roman, Sans, and Typewriter font families in either the T1 or OT1 encodings.

```

\cmttfamily 3035 \ifx\Tienc\encodingdefault
3036     \providecommand{\cmrfamily}{\usefont{T1}{cmr}{m}{n}}
3037     \providecommand{\cmssfamily}{\usefont{T1}{cmss}{m}{n}}
3038     \providecommand{\cmttfamily}{\usefont{T1}{cmtt}{m}{n}}
3039 \else
3040     \providecommand{\cmrfamily}{\usefont{OT1}{cmr}{m}{n}}
3041     \providecommand{\cmssfamily}{\usefont{OT1}{cmss}{m}{n}}
3042     \providecommand{\cmttfamily}{\usefont{OT1}{cmtt}{m}{n}}
3043 \fi

```

`\textcmr` Text command for the Computer Modern Roman font family.

```
3044 \DeclareTextFontCommand{\textcmr}{\cmrfamily}
```

`\textcmss` Text command for the Computer Modern Sans font family.

```
3045 \DeclareTextFontCommand{\textcmss}{\cmssfamily}
```

`\textcmtt` Text command for the Computer Modern Typewriter font family.

```
3046 \DeclareTextFontCommand{\textcmtt}{\cmttfamily}
```

At the start of the document, change the `\@setfontsize` macro and call the normal font to implement the change.

```

3047 \AtBeginDocument{%
3048     \let\cmin@old@setfontsize=\@setfontsize
3049     \let\@setfontsize=\cmin@new@setfontsize}
3050 \AtBeginDocument{\normalsize}

```

The end of this package.

```
3051 \>/uscall)
```

Bibliography

- [And69] Donald M. Anderson. *The Art of Written Forms: The Theory and Practice of Calligraphy*. Holt, Rinehart and Winston, 1969.
- [Bol95] Giulia Bologna. *Illuminated Manuscripts: The Book before Gutenberg*. Crescent Books, 1995.
- [Day95] Lewis F. Day. *Alphabets Old & New*. (3rd edition originally published by B. T. Batsford, 1910) Senate, 1995.

- [Dro80] Marc Drogin. *Medieval Calligraphy: Its History and Technique*. Allenheld, Osmun & Co., 1980.
- [Dru95] Johanna Drucker. *The Alphabetic Labyrinth*. Thames & Hudson, 1995.
- [Fir93] Richard A. Firmage. *The Alphabet Abecedarium*. David R Goodine, 1993.
- [Fli98] Daniel Flipo. *The LETTRINE package*. (Available from CTAN in `macros/latex/contrib/supported`). 1998.
- [Har95] David Harris. *The Art of Calligraphy*. DK Publishing, 1995.
- [Jen95] Frank Jensen. *The BETON package*. (Available from CTAN in `macros/latex/contrib/supported`). 1995.
- [Joh71] Edward Johnston (ed. Heather Child). *Formal Penship and Other Papers*. Penthalic, 1971.
- [Knu87] Donald E. Knuth. *Computer Modern Typefaces*. Addison-Wesley, 1987.
- [Knu92] Donald E. Knuth. *The METAFONTbook*. Addison-Wesley, 1992.
- [GMS94] Michel Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTeX Companion*. Addison-Wesley Publishing Company, 1994.
- [Tho75] Alan G. Thomas. *Great Books and Book Collectors*. Weidenfield and Nicolson, 1975.
- [Wil99] Peter R. Wilson. *The DOCMFP Package*. (Available from CTAN in `macros/latex/contrib/supported`). 1999.
- [Wil99b] Peter R. Wilson. *The ROMANNUM Package*. (Available from CTAN in `macros/latex/contrib/supported`). 1999.

Index

Numbers written in *italic* refer to the page where the corresponding entry is described; numbers underlined refer to the code line of the definition; numbers in *roman* refer to the code lines where the entry is used.

Symbols		Numbers
<code>&</code> (routine)	<code>:</code> (routine) <u>2451</u>	<code>0</code> (routine) <u>2951</u>
<code>'</code> (routine) <u>2472</u>	<code>;</code> (routine) <u>2461</u>	<code>1</code> (routine) <u>2754</u>
<code>''</code> (routine) <u>2629</u>	<code>=</code> (routine) <u>2389</u>	<code>2</code> (routine) <u>2771</u>
<code>,</code> (routine) <u>2432</u>	<code>?</code> (routine) <u>2405</u>	<code>3</code> (routine) <u>2793</u>
<code>-</code> (routine) <u>2653</u>	<code>\@setfontsize</code> 3048, 3049	<code>4</code> (routine) <u>2823</u>
<code>--</code> (routine) <u>2673</u>	<code>\@tempa</code> 3027–3030	<code>5</code> (routine) <u>2845</u>
<code>---</code> (routine) <u>2693</u>	<code>\@tempb</code> 3027, 3029	<code>6</code> (routine) <u>2870</u>
<code>.</code> (routine) <u>2442</u>	<code>'</code> (routine) <u>2482</u>	<code>7</code> (routine) <u>2890</u>
	<code>“</code> (routine) <u>2641</u>	

8 (routine)	2912	comma_width (variable)	9	E	
9 (routine)	2931	ct (routine)	2506	E (routine)	1720
A		D		e (routine)	931
A (routine)	1601	D (routine)	1692	\edef	3027, 3029
a (routine)	812	d (routine)	903	\encodingdefault	2998, 3035
asc_height (variable)	8	\DeclareFontFamily	2971, 2983	F	
ascratio (variable)	8	\DeclareFontShape	2972–2980, 2984–2992	F (routine)	1760
\AtBeginDocument	3047, 3050	\DeclareTextFontCommand	3003, 3044–3046	f (routine)	971
B		\Delta	96	fig_height (variable)	8
B (routine)	1640	desc_depth (variable)	8	fig_width (variable)	8
b (routine)	851	descratio (variable)	8	G	
bfudge (variable)	6	dot_size (variable)	9	G (routine)	1791
body_depth (variable)	8	draw_clubbase (routine)	411	g (routine)	1002
body_height (variable)	8	draw_clubserif (routine)	244	H	
bowl_stem (routine)	404	draw_flatserif (routine)	217	H (routine)	1826
C		draw_flick (routine)	301	h (routine)	1037
C (routine)	1669	draw_fork (routine)	373	half_height (variable)	8
c (routine)	880	draw_istem (routine)	281	halfquill (variable)	11
cap_height (variable)	7	draw_pcomma (routine)	432	hstretch (variable)	6
cap_jutstretch (variable)	6	draw_pdot (routine)	421	I	
cap_serif_fit (variable)	7	draw_plq (routine)	448	I (routine)	1857
cap_stemstretch (variable)	6	draw_roundserif (routine)	211	i (routine)	1068
caprat (variable)	6	draw_rtriangle (routine)	351	J	
carol_height (variable)	6	draw_rutriangle (routine)	362	J (routine)	1873
carolmin_font_setup (routine)	144	draw_serif (routine)	186	j (routine)	1084
clubjut (variable)	8	draw_tail (routine)	317	jut (variable)	8
clubjutfudge (variable)	8	draw_triangle (routine)	329	jutfudge (variable)	8
\cmin@baselineskip@table	3013, 3029	draw_triserial (routine)	295	jutstretch (variable)	6
\cmin@baselineskip@value	3028, 3031, 3032, 3034	draw_utriangle (routine)	340	K	
\cmin@new@setfontsize	3026, 3049	ductus (variable)	11	K (routine)	1896
\cmin@old@setfontsize	3034, 3048			k (routine)	1107
\cminfamily 4, 2998, 3003				L	
\cmrfamily 5, 3035, 3044				L (routine)	1938
\cmssfamily 5, 3035, 3045				l (routine)	1149
\cmttfamily 5, 3035, 3046				letter_fit (variable)	7
				M	
				M (routine)	1957
				m (routine)	1168
				makebox (routine)	113
				monospace (variable)	9

N			
N (routine)	1995		
n (routine)	1206		
\noexpand	3027, 3029		
\normalsize	3050		
O			
O (routine)	2024		
o (routine)	1235		
o (variable)	9		
obowl (routine)	528		
offsetbowl (routine)	464		
P			
P (routine)	2046		
p (routine)	1257		
pangle (variable)	7		
pdir (variable)	11		
ppdir (variable)	11		
proofpcbb (routine)	124		
\providecommand	2997,		
	3009, 3036–		
	3038, 3040–3042		
\ProvidesPackage	..		
	2996, 3007		
Q			
Q (routine)	2075		
q (routine)	1286		
qhor (variable)	11		
quarterquill (vari-			
able)	11		
quill (variable)	11		
qvert (variable)	11		
R			
R (routine)	2103		
r (routine)	1314		
\RequirePackage	.. 3008		
\rmdefault	3010		
routines:			
&	2713		
'	2472		
''	2629		
,	2432		
-	2653		
--	2673		
---	2693		
.	2442		
:	2451		
;	2461		
=	2389		
?	2405		
'	2482		
''	2641		
0	2951		
1	2754		
2	2771		
3	2793		
4	2823		
5	2845		
6	2870		
7	2890		
8	2912		
9	2931		
A	1601		
a	812		
B	1640		
b	851		
bowl_stem	404		
C	1669		
c	880		
carolmin_font_setup	144		
ct	2506		
D	1692		
d	903		
draw_clubbase	411		
draw_clubserif	244		
draw_flatserif	217		
draw_flick	301		
draw_fork	373		
draw_istem	281		
draw_pcomma	432		
draw_pdot	421		
draw_plq	448		
draw_roundserif	211		
draw_rtriangle	351		
draw_rutriangle	362		
draw_serif	186		
draw_tail	317		
draw_tlserif	295		
draw_triangle	329		
draw_triserif	265		
draw_tstem	288		
draw_urswish	391		
draw_utriangle	340		
draw_wave	309		
E	1720		
e	931		
F	1760		
f	971		
G	1791		
g	1002		
H	1826		
h	1037		
I	1857		
i	1068		
J	1873		
j	1084		
K	1896		
k	1107		
L	1938		
l	1149		
M	1957		
m	1168		
makebox	113		
N	1995		
n	1206		
O	2024		
o	1235		
obowl	528		
offsetbowl	464		
P	2046		
p	1257		
proofpcbb	124		
Q	2075		
q	1286		
R	2103		
r	1314		
S	2129		
s	1340		
scalarprod	533		
st	2568		
T	2155		
t	1366		
tangentpoint	538		
U	2183		
u	1394		
V	2213		
v	1424		
W	2243		
w	1454		
X	2279		
x	1490		
Y	2318		
y	1529		
Z	2348		
z	1559		

rule_thickness (variable)	8	u (variable)	7	qvert	11		
S							
S (routine)	2129	V (routine)	2213	rule_thickness . .	8		
s (routine)	1340	v (routine)	1424	sb	7		
sb (variable)	7	variables:		serif_fit	7		
scalarprod (routine)	533	asc_height	8	side	8		
serif_fit (variable)	7	ascratio	8	slant	9		
\sfdefault	3010	bfudge	6	srad	8		
side (variable)	8	body_depth	8	stemstretch	6		
slant (variable)	9	body_height	8	stylus	10		
srad (variable)	8	cap_height	7	szfudge	6		
st (routine)	2568	cap_jutstretch . .	6	thick	7		
stemstretch (variable)	6	cap_serif_fit . . .	7	thickfudge	7		
stylus (variable)	10	cap_stemstretch .	6	thickstylus	10		
szfudge (variable)	6	caprat	6	thin	7		
T						thinfudge	7
T (routine)	2155	carol_height	6	threequarterquill	11		
t (routine)	1366	clubjut	8	trijut	8		
tangentpoint (routine)	538	clubjutfudge	8	u	7		
\textcmin	4, 3003	comma_width	9	vstretch	7		
\textcmr	5, 3044	desc_depth	8	width_adj	7		
\textcmss	5, 3045	descratio	8	x_height	7		
\textcmtt	5, 3046	dot_size	9	vstretch (variable)	7		
thick (variable)	7	ductus	11	W			
thickfudge (variable)	7	fig_height	8	W (routine)	2243		
thickstylus (variable)	10	fig_width	8	w (routine)	1454		
thin (variable)	7	half_height	8	width_adj (variable)	7		
thinfudge (variable)	7	halfquill	11	X			
threequarterquill (variable)	11	hstretch	6	X (routine)	2279		
\Tienc	2997, 2998, 3009, 3035	jut	8	x (routine)	1490		
trijut (variable)	8	jutfudge	8	x_height (variable)	7		
\ttdefault	3010	jutstretch	6	Y			
U						Y (routine)	2318
U (routine)	2183	letter_fit	7	y (routine)	1529		
u (routine)	1394	monospace	9	Z			
V						Z (routine)	2348
u (variable)	7	o	9	z (routine)	1559		
W							
asc_ratio	8	pangle	7				
bfudge	6	pdir	11				
body_depth	8	ppdir	11				
body_height	8	qhor	11				
cap_height	7	quarterquill	11				
cap_jutstretch	6	quill	11				
cap_serif_fit	7						
cap_stemstretch	6						
caprat	6						
carol_height	6						
clubjut	8						
clubjutfudge	8						
comma_width	9						
desc_depth	8						
descratio	8						
dot_size	9						
ductus	11						
fig_height	8						
fig_width	8						
half_height	8						
halfquill	11						
hstretch	6						
jut	8						
jutfudge	8						
jutstretch	6						
letter_fit	7						
monospace	9						
o	9						
pangle	7						
pdir	11						
ppdir	11						
qhor	11						
quarterquill	11						
quill	11						