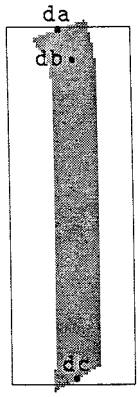
Notice how natural this description is: we can specify the apparent width (which in handwriting can be controlled by pressure) and angle of our pen at each of three critical points on our stroke, and we control the curve drawn by "pulling the curve tighter" via the `tension` parameter. Notice, too, that despite all this control, there is still much flexibility here: the values of the three arguments passed to `downstroke` are determined precisely for each character at run time.

Having thus defined `downstroke`, we can describe an

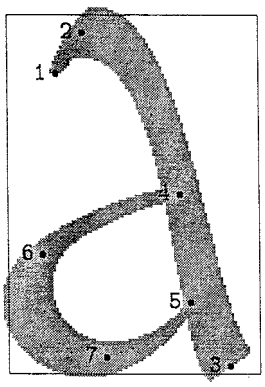


very succinctly indeed:

```
beginchar(73,4.5/60em#,m#,0); "The letter I";
  downstroke(m+verto,left_edge,0-verto);
endchar;
```

Here, by invoking `beginchar`, we assign a suitable ASCII code to the image METAFONT will create, and specify the height, depth, and width of the box in which the character sits. All we need do then is invoke `downstroke` with suitable arguments, and invoke `endchar` to wrap up all loose ends.

Creating a solid working set of subroutines for all the recurring motifs in a font is just the first, albeit rather major, step in the creation of our alphabet. As we have seen, METAFONT takes to this task like a duck to water. It is equally easy to deal with the "odd cases", the letters that are not composed in whole or part of common letter parts. A good example is an




```
beginchar(65,28/60em#,m#,0); "The letter A";
% First, pickup the appropriate pen
  pickup ucpen;
% Describe positioning of pen at
% all key points
  penpos1(uvthinp,180);
  penpos2(1.15uchorzp,90-15);
  penpos3(1.15uchorzp,35);
  penpos4(uvthinp,90+15);
  penpos5(uhthinp,15);
  penpos6(1.15uchorzp,180+15);
  penpos7(3/4uchorzp,270+15);
% Specify the locations of key points
  y4=1/2m;
  y5=1/5m;
  y6=1/3m;
  x6r=left_edge;
  y7r=0;
  x7=1/2[x6r,x5r];
  x1r=1/6w;
  y1r=5/6m;
  y2r=m+verto;
  x2r=1/5[x1r,x3r];
  x3r=right_edge;
  y3l=0-verto;
  z5l=whatever[z3l,z2r];
  z4l=whatever[z3l,z5r];
% Draw through those points
  penstroke z3e{-1,1}..
    tension 2..z2e{-6,-1}..
    tension 4..z1e
  penstroke z4e..
    tension 1.5..z6e..
    z7e..tension 1.15..z5e;
  labels(1,2,3,4,5,6,7);
endchar;
```

This code should give the reader a good feeling for just how easy METAFONT descriptions are. This is probably the most complicated code I had to write for my uncial character set, and yet I believe it is fairly straightforward and easy to follow along in conjunction with the proof.

So, my experiment gave METAFONT good marks for ease and speed of use. The positive reactions I have received thus far on the character set's appearance tend to also rate it favorably on the question of how well it emulates the model. (I welcome the readers' opinions of the appearance of the uncial sample given at the end of this article to add to the evidence here.)

Of course, we cannot expect *any* 300 dpi digitized image to perfectly echo the graceful tapers and curves we are after, but I suspect this METAFONT version could hold its own with another produced by hand for the same resolution. The acid test, though, is to print these fonts on a very high resolution device, and compare that copy to an analogue version.

The final question I hoped to answer regarded the robustness of these METAFONT characters. While I found that I could generate these characters at a good range of point sizes and a good range of resolutions without any complaints from METAFONT, the results on my low resolution printer were unacceptable at small point sizes:

abcdefghijklm
nopqrstuvwxyz
0123456789

Disturbing breaks and fadeouts in the fine lines occurred. While this is bothersome (and contrasts with the results I am used to with METAFONT79), I do not consider it an insurmountable problem. There are certain parameters that can be tweaked (namely `blacker` and `fillin`) that can handle many problems. The overriding impression I get, though, is that METAFONT is a rich enough and powerful enough and precise enough language that with time and care this problem can be dealt with. However, the production of good fonts at typical text sizes for low resolution (300 dpi and under) devices is one task that METAFONT needs to be able to do well, and is one that I intend to focus on in the early part of 1987.

As always, I welcome my readers comments on this article or on any aspect of font design.

G.K.M. Tobin
31 December 1986

Knowledge
is good, method is good
but one thing beyond all others
is necessary: and that is to have
a head, not a pumpkin,
on your shoulders and
brains, not pudding,
in your head.
— a.e. housman

abcdefghijklm
nopqrstuvwxyz
0123456789

This document printed at 300 dpi with experimental Metafoundry fonts.