

With a suitable setting of the parameters, we can even imitate a typewriter with its fixed width letters, as shown in Figure 13e. There is also a provision to slant the letters as in Figure 13f; here the pen position is varied, but the actual shape of the pen is not being slanted, so circular dots remain circles.

Finally, Figure 13i illustrates the variations you can get by giving weirder settings to the parameters. **In other words, there is a unique tangent at every point of the curve.** *This chapter and the next are for readers who share my fascination with original source documents. The draft report I wrote during the first third of May was entitled TEXDR.AFT*

Several years ago when I began to look at the problem of designing suitable alphabets for use with modern printing equipment, I found that 25 of the letters were comparatively easy to deal with. The other letter was S. For three days and nights I had a terrible time trying to understand how a proper S could really be defined. The solution I finally came up with turned out to involve some interesting mathematics, and I believe that students of calculus and analytic geometry may enjoy looking into the question as I did. The purpose of this paper is to explain what I now consider to be the “right” mathematics underlying printed S’s, and also to give an example of the METAFONT language I have recently been developing.* (A complete description of METAFONT, which is a computer system and language intended to aid in the design of letter shapes, appears in [3, part 3].)

1 First part

Before getting into a technical discussion, I should probably mention why I started worrying about such things in the first place. The central reason is that today’s printing technology is essentially based on discrete mathematics and computer science, not on the properties of metals or of movable type. The task of making a plate for a printed page is now essentially that of constructing a gigantic matrix of 0s and 1s, where the 0s specify white space and the 1s specify ink. I wanted the second edition of one of my books to look like the first edition, although the first edition had been typeset with the old hot-lead technology; and when I realized that this problem could be solved by using appropriate techniques of discrete mathematics and computer science, I couldn’t resist trying to find my own solution.

1.1 Sub-sect

By studying this example we can get some idea of the problems involved in specifying a proper S shape. However, I was actually seeking the solution to a more general problem than the one Torniello faced: Instead of specifying only one particular S, I needed many different variations, including bold face S’s that are much darker than the normal text. I discussed this recently with Alan Perlis, who pointed out that a central issue arising whenever we try to automate something properly is what he calls “the art of making constant things variable.”

In the case of letter design, we don’t merely want to take a particular drawing and come up with some mathematics to describe it; we really want to find the principles underlying the drawing, so that we can generate infinitely many drawings (including the given one) as a function of appropriate parameters.

My goal was to create entire alphabets that would depend on a dozen or two parameters in such a way that all the letters would vary in a compatible manner as the parameters would change.

2 Second part

After looking at these Renaissance constructions and a lot of modern S shapes, I came to the conclusion that the main stroke of the general S curve I sought would be analogous to the curve in Figure 6; each boundary curve was to be an ellipse followed by a straight line followed by another ellipse. This led me to pose the following problem: What ellipse has its topmost point at (x_t, y_t) and its leftmost point at (x_l, y_l) for some y_l , and is tangent to the straight line of slope m that passes through (x_c, y_c) , given the values of x_t , y_t , x_l , m , x_c , and y_c ? (The ellipse in question is supposed to have the coordinate axes as its major and minor axes; in other words, it should have left-right symmetry. See Figure 7 on the next page.) The reason for my posing this problem should be fairly clear from our previous discussions: We know a point that is supposed to be the top of the S curve, and we also know how far the curve should extend to the left; furthermore we have a straight line in mind that will form the middle link of the stroke.

The problem stated in the preceding paragraph is interesting to me for several reasons. In the first place, it has a nice answer (as we will see). In the second place, the answer does in fact lead to satisfactory S curves. In the third place, the answer isn't completely trivial; during a period of two years or so I came across this problem four different times and each time I was unable to find my notes about how to solve it, so I spent several hours deriving and rederiving the formulas whenever I needed them. Finally I decided to write this paper so that I wouldn't have to derive the answer again.

EXAMPLE