

---

## About the DK versions of Lucida

Charles Bigelow

Donald Knuth’s informative and amusing letter, “A footnote about ‘Oh, oh, zero!’” [4], tells of his efforts in the late 1960s to reduce confusion between capital letter ‘O’ and zero in the typography of computing journals and books, when he recommended to the ACM (Association for Computing Machinery) and his publisher Addison-Wesley a new, squarish shape of capital ‘O’ to distinguish it from oval zero in fonts that simulated typewriter (fixed-width) text. After much correspondence, ACM did not follow his suggestion but Addison-Wesley did, commissioning a special squarish ‘O’ for the first volume of *The Art of Computer Programming*.

A dozen years later, Don designed a squarish capital ‘O’ for Computer Modern Typewriter, which he made with his newly developed Metafont system. His beloved squarish ‘O’ is now standard in his own books and in publications of others who use his Computer Modern typefaces.

He doesn’t explicitly state in his “Footnote” (maybe it seemed too obvious to mention) that in effect he was proposing a new parameter of typographic distinction: “squarishness” versus “roundishness” of curves (my words) within a typeface.

### Structure of a typeface design

The graphical structure of a typeface design is built from a complex set of distinctive features, some of which apply within a face, and others of which apply between faces. Within a face, several features like round, straight, and diagonal, ascender or descender, dot or no dot distinguish individual letters from each other. Groups of letters are distinguished by a few features, including height, width, pointyness (which distinguishes punctuation (hence the name) from letters proper) and weirdness (for lack of a better word) for characters like @ # \$ % & that look somehow weirder than letters, are usually logographs symbolizing a word rather than a single sound, and for which people can’t seem to agree on names or purported historical derivations. For instance, commercial “at” @ (plausibly derived from a contracted ligature of Latin “ad” meaning “at, to, toward” but poorly attested before the 16th century) is called “at” by most English speakers, but its name varies wonderfully from language to language, with meanings of ear, strudel, elephant trunk, monkey tail, and so on; number sign # (probably an abbreviation of Latin “numerus” meaning “number”) is variously called “hash”, “octothorpe”, “pound”, “sharp” and “number” in English. I expect that readers will enjoy noting

other names and suggesting alternative derivations.

When we read, we recognize all these kinds of distinctions almost unconsciously because they are part of our passive “vocabulary” of typographical signifiers. They help us understand the structuring of text. In type design, these distinctions are elements of a designer’s active graphical vocabulary, the conceptual toolkit used to shape the look of information.

Graphical distinctions within and between capitals and lower-case, roman and italic typefaces have been devised, refined, and standardized since the early 15th century, when humanist scholar and scribe Poggio Bracciolini developed the humanist handwriting that eventually became canonical roman type. Thanks to Poggio, nearly all Latin text typefaces contain two alphabets: upper-case (capitals) and lower-case (small letters), and thanks to Poggio’s friend, Niccolò Niccoli, who wrote a cursive variant of Poggio’s hand, roman has been distinguished from italic.

Over time, the capitals gained semantic import; marking a significant distinction. A “Bill” is a person, but a “bill” is an invoice, a bird beak, or a cap brim. I can attest that something similar applies to “Chuck” and “chuck”. The distinction between upper- and lower-case is therefore graphemic; the graphical difference signifies a difference in meaning. Interestingly, the patterns of capital usage differ between orthographies. In German orthography, initial capitals mark nouns and are correspondingly frequent, whereas in modern French, capitals are rather rare except at the beginnings of sentences and personal names. In English orthography, capital usage was fairly frequent in earlier centuries but has diminished since the 19th century.

Size. Capital letters are comparatively bigger, as denoted by their Latin-derived name “majuscule” (bigish) and lower-case letters smaller, in Latin “minuscule” (smallish). The Latin majuscule/minuscule distinction is rare among typographic writing systems, occurring first in Latin type, and later in Greek, Cyrillic, and Armenian types influenced by the Latin model but deriving their capital versus lower-case forms from different roots and by different means. The “case” distinction is not found in most other writing systems, including Hebrew, Arabic, Chinese, Korean, Japanese, or the several related systems used in India, such as Devanagari for Hindi and Sanskrit.

Width. The width of characters is another distinction. In proportionally spaced typefaces, capital ‘O’ is wider than zero, which is usually adequate for differentiation, but the width distinction is almost neutralized in monospaced (fixed-width) fonts, and that is where confusion between ‘O’ and zero most commonly occurs.

Marks. The dot over the minuscule letter ‘i’ appeared in the late middle ages as a light stroke over the letter ‘i’ in blackletter “textura” script, to mark the letter as separate from other so-called “minim” letters. The dot, along with later accents and diacritics, like those proposed in the 16th century by French typographer Geoffrey Tory, has been continued in most Euro-Latin typography, and there are occasional extensions when adapting Latin letters to other languages or clarifying distinctions, as in fonts for linguistics. Hence it seemed natural to add marks to distinguish capital letter ‘O’ from zero when confusion between the two cropped up in the early days of computing.

In my article “Oh, oh, zero!” [2], which Don Knuth “Footnoted”, I gave examples of various proposals from computer journals and typography journals for distinguishing ‘O’ and zero. These included the addition (or rarely, subtraction) of slashes, dots, loops, bars, and other twiddles to ‘O’ or to zero. Proposals from scientists, technologists, engineers, or mathematicians usually added distinguishing marks to the ‘O’, keeping their zero pristine in its symbolic mathematical emptiness. Proposals from humanists, artists, scholars and designers added distinguishing marks to the zero, keeping their capital letter ‘O’ by historical precedent and charismatic authority, citing the classical Roman inscriptions, or the famous story of the early Renaissance painter Giotto, as told by Giorgio Vasari. When an emissary from the Pope asked Giotto for a drawing to show his skill:

Giotto ... took a sheet of paper, and with a brush dipped in red, fixing his arm firmly against his side to make a compass of it, with a turn of his hand he made an ‘O’ [tondo] so perfect in curve and contour that it was a marvel to see it.

Giorgio Vasari, *Lives of the most excellent painters, sculptors, and architects*

(Here I translate Vasari’s Italian word “tondo” as ‘O’ in English, but it can also be translated with justification as ‘circle’. Modern Italians do both when referring to Vasari’s story, sometimes saying ‘cerchio’ (circle) or ‘O’ instead of “tondo”, which incidentally also means “roman” when referring to a typeface.)

The skill of drawing a marvelous ‘O’ was not lost with Giotto. The late Fr. Edward Catich, who revived the painting and carving of Imperial Roman Inscriptional capitals, could paint a wonderfully round capital ‘O’ with two strokes of the brush, in keeping with his analysis of how the Trajan capitals were painted and carved back in 113 A.D. When Catich would demonstrate this to the admiring gasps of onlookers, he would grin and say, “Perfetto come la ‘O’ di Giotto.” Kris Holmes, who studied with

Catich briefly and with Catich’s pupil, Fr. Bob Paladino, can recount his other feats of brush-writing mastery, including writing a perfect Roman capital ‘R’ behind his back and upside down.

Contrast. A distinction called “contrast”, or modulation of thick and thin strokes, was used in “old-style” types from the 15th to the late 18th century, and in modern revivals, in which the figure zero was cut approximately at lower-case height but as an annulus, an unmodulated ring without thick-thin variation, distinguishing it from the lower-case ‘o’, which retained the varying line thickness of the humanist pen written letter. Contrast is also a major feature of the typeface genre called “modern”.

In his 1960s suggestion to Addison-Wesley, Don proposed a new way to distinguish the curve of the ‘O’ from that of the zero: “squarishness” versus “roundishness” (my words). Although, as he relates, Addison-Wesley followed his suggestion, it wasn’t a generally practical or enduring solution. In Metafont some ten years later, Don implemented his solution as a parameter he called “superness” (from Piet Hein’s term “superellipse”) [5]. Superness “controls the amount by which the curve differs from a true ellipse.” Thus, Don gave squarishness versus roundishness in typeface design a precise algebraic expression: the exponent of the general equation describing the ellipse.

The classic ellipse of the conic curves analyzed by Greek mathematicians is algebraically described by an equation of degree 2. In the early 19th century, the French mathematician Gabriel Lamé generalized the ellipse by varying the exponent. In particular, an exponent above 2.0 makes the corners of the ellipse smoothly bulge or inflate beyond the classical elliptical oval. In 1959, Danish polymath Piet Hein empirically determined that an ellipse of degree 2.5 was, for him, the most agreeable compromise between the ellipse and the rectangle, and he used the shape to design a traffic oval. Hein called the ellipse of degree 2.5 a “superellipse”. I believe that, to date, Metafont is the only type design tool that provides control over this parameter.

In “Oh, oh, zero!” [2], I also showed samples of recent digital fonts that distinguish zero and ‘O’ by various means, but I neglected to show a sample of Computer Modern Typewriter (shown here in fig. 1), although Karl Berry used CMTT in his “Production Notes” on the article [1], which shows instances of the squarish ‘O’. Knuth refers to this in his “Footnote”.

In this note, I wish to correct my oversight and also to relate my subsequent efforts to respond to Don’s appeal for a squarish ‘O’ for Lucida Console, a typeface that Kris Holmes and I designed in 1993, based on our Lucida Sans Typewriter of 1986.

### Squarish ‘O’s in Lucida

At the end of his *TUGboat* “Footnote”, Don wrote:

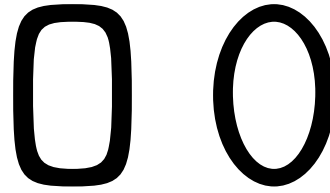
Alas, however, Chuck’s essay demonstrates that I’m still standing alone in this respect: None of the nine monospaced typefaces in his Fig. 9 have anything like an Oh that I would want to use. (Nowhere did I see a really satisfactory Oh in Chuck’s discussion until I came to Karl Berry’s production notes at the end, and Karl’s reference to `ZeroFont0T.otf`.) I herewith submit a humble request to have squarish O and Q available as alternates in the next edition of Lucida Console.

I could not ignore Don’s appeal, nor the challenge inherent in it, so I set about crafting a squarish ‘O’ for Lucida Console, but I began with Lucida Grande Mono, the most general case of three nearly identical designs: Lucida Sans Typewriter, which was designed to look like Lucida Sans but monospaced, Lucida Console, a version of Lucida Sans Typewriter with shortened capitals for the console/terminal window(s) of an operating system, and Lucida Grande Mono, which has the taller capitals of the original design, the larger character set of Lucida Console, and various small adjustments based on three decades of experience with the original design.

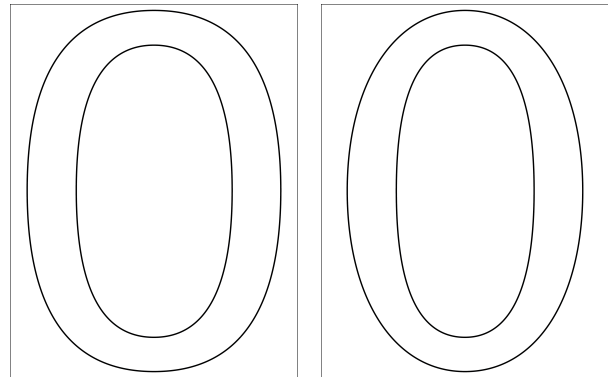
Of course, the task of making a simple squarish ‘O’ turned out to take much longer than I estimated. I figured it would take a few weeks, with interruptions and hiatuses, but it took several months. The practical difference between estimate and reality conformed to a Knuthian heuristic for estimating the length of time a project will take: make your best estimate in some time unit, add one, and jump to the next higher level of time measure.

Now that the fonts are at last done, Karl has asked me to write some words about the design process. I wish I could describe it as a clear, precise, and logical process but I must confess that, to borrow and invert a phrase used by Leslie Valiant for evolutionary “ecorithms” (contrasted with “algorithms”), I used a “probably approximately incorrect” method.

I started with the capital ‘O’ character from Lucida Grande Mono as the most general case, as described above. This is the same glyph as the original ‘O’ from Lucida Sans Typewriter (fig. 2 shows the outline of the ‘O’ and zero glyphs). It had been hand-drawn in 1986, digitized with Peter Karow’s Ikarus software as contour points on cubic Hermite splines. Those were converted to conic splines at the Imagen Corporation, using a contour representation developed by Vaughan Pratt at Stanford, and first converted to bitmap fonts and released in 1986 as Lucida Sans Typewriter. The Ikarus splines were later con-



**Figure 1:** Computer Modern Typewriter (10pt design size, enlarged): capital Oh (left) and zero (right).

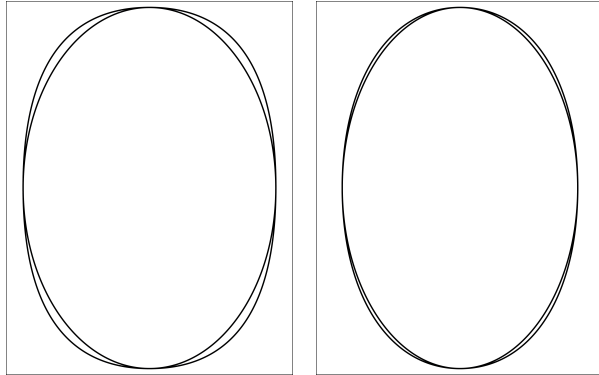


**Figure 2:** Hand-drawn capital Oh (left) and zero (right) in Lucida Grande Mono (originally Lucida Sans Typewriter).

verted to Sun Microsystems’ F3 general conic format developed by Jacobo Valdes and Eduardo Martinez (also based on Pratt’s conics). In 1990, we converted the Ikarus font to TrueType format for Microsoft, and some years later, to Bézier format for Adobe.

The original Lucida Sans Typewriter capital ‘O’ is elliptical-ish, somewhat more squarish than a true ellipse. The shape was drawn visually; its slightly greater width and squarishness helped distinguish it from the zero and gave it a little more interior area within the Procrustean confinement of the fixed-width typewriter cell. The zero is closer to a true ellipse (fig. 3).

At the time of the original design, we debated whether to put a slash in the zero to make a clear distinction between zero and ‘O’. We eventually did not add the slash because of various conflicts, the foremost of which was possible confusion with the Norwegian ‘Ø’ letter (which may have inspired the null set symbol with slash, according to mathematician André Weil). Also, some computer scientists and engineers — supposedly the main users of such a font — disdained an altered zero. Moreover, one of our goals for Lucida Sans Typewriter was to closely resemble proportionally spaced Lucida Sans, which of course did not have slashed zero. And, traditional “typewriter” fonts also did not have slashed zero.



**Figure 3:** Outer contours of above hand-drawn capital Oh (left) and zero (right), compared to true ellipses (inner contours) with the same major and minor axes.

To gain a sense of what Don wanted in a squarish ‘O’ shape, I examined the ‘O’ in his Computer Modern Typewriter. My task would have been much simpler if I had just imported his ‘O’s Bézier contours from a PostScript font (originally made by collective effort of Y&Y, Blue Sky Research, and Projective Solutions), scaled it to the proportions of Lucida Grande Mono, plunked it (along with accented variants) into the proper Unicode slots, and been done with it. But, some four decades ago, when Kris Holmes and I began working on type designs together, she said it is a mistake to look at other typefaces when designing a new one, because inevitably you can’t get the other designs out of your head, and that prevents you from creating something new from your own concepts. Also, it’s more interesting, and you learn more by starting afresh than by copying.

So, having looked at the squarish ‘O’ in Computer Modern Typefaces, I closed the book and put it back on the shelf. I then proceeded to craft a shape that looked squarish to my eye by starting from scratch. Designing a squarish ‘O’ for Lucida wasn’t simply a problem of making a squarish ‘O’ but of making a squarish ‘O’ that visually harmonized with everything else in the typeface, although there wouldn’t be any other letters like it except ‘Q’, or, in an extended character set, Greek Omicron and Theta, Cyrillic ‘O’, and the various accented ‘O’s.

In 1992 with Y&Y, developers of a PC-based version of T<sub>E</sub>X, we developed Lucida fonts in PostScript Type 1 format for use with T<sub>E</sub>X. For character contours, we tried to minimize the number of Bézier points, to keep font file sizes small, and the conversion from Ikarus format to PostScript Bézier put on-curve spline points at the  $x$ - and  $y$ - extremes of curves, in accordance with how we hand-digitized the

outlines. Having spline points at extremes facilitated the later hinting of the PostScript outlines by Y&Y. The capital ‘O’, then, comprised four Bézier curve segments with four on-curve spline knots where segments joined, and eight associated off-curve control points, or handles.

As using Bézier curve-based software for designing fonts directly on computers became available, another reason to minimize curve segments became evident—it was difficult for designers to smooth contours composed of many Bézier segments with spline knots and handles that had to be controlled by the unintuitive process of moving curve “handles” (control points) around in the plane and sometimes adding on-curve points. Cubic Bézier curves are tremendously flexible; in addition to seemingly well-behaved curves that look like (but are not exactly) classical conic sections, Bézier curves can make a vast number of loops, cusps, inflections, and other surprising shapes that are never needed in type design but which can confound the visual artist trying to control them. So, to get a handle on the curves (sorry), designers tend to be minimalists, using as few Bézier curves as possible if the curves must be manipulated by hand and eye.

A problem with the minimalist approach is that Bézier curves cannot exactly match true circles and ellipses. Arbitrarily closer approximations can be achieved by splitting Bézier arcs into shorter segments, but more segments involve more curve joins and handles, which thus increase the probability of making awkward joins, cusps, bulges, or dips, thereby complicating the task of the designer working with the curves. The minimization of Bézier segments may be a reason that many of the typefaces developed directly on computer screens seem to me to have subliminal similarity of curves. When I look at the shapes of many recent fonts, I have a sense of *déjà vu* all over again, as the late philosopher Yogi Berra is said to have said.

At any rate, because we had already reduced the contours of Lucida Sans Typewriter to a near-minimal number of Bézier curves when working with Y&Y, I could start with that version and manipulate the Bézier handles until the ‘O’ shape looked visibly squarish. This went through several iterations because I had to test the new ‘O’ shape in sample settings with the rest of the typeface. When the ‘O’ looked outstandingly squarish, it didn’t jibe with the rest of the characters in the font, but when it was rather roundish, it didn’t quite look different enough from the oval zero. As usual in type design, it wasn’t enough to achieve a “just noticeable difference” between the two characters. What was needed was a

“definitely noticeable difference”.

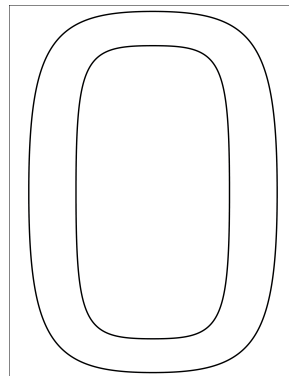
I then experimented with making the interior counter somewhat more squarish than the exterior contour. This gave the shape an illusory shoulder-padded look reminiscent of TV soap opera actress costumes of the 1980s.

When I had an ‘O’ that seemed adequately squarish, that wasn’t the end, because getting the isolated shape to look right was not enough; it also had to play well with others. Because the squarish ‘O’ was no longer in keeping with the ovality of most of the other letters, I had to re-fit its sidebearings—the spaces on each side. In a fixed-width font, side spacing usually involves compromises because letters like capital ‘T’ have lots of air around them, while ‘M’ and ‘W’ are packed in tight. The oval bowls of the original ‘O’ had been adjusted to seem roughly equal in spacing next to letters with straight sides like ‘H’ or with round sides, like ‘C’ or ‘D’.

Compared to the oval bowls of the original ‘O’, the sides of the squarish ‘O’ were almost upright, just slightly bowed, so they seemed closer to other characters, and hence their side spacing had to be increased to restore approximately equal letter spacing, but in the fixed-width cell, the letter also had to be narrowed to make room for the increased spacing. The squarish form also necessitated adjusting the weight of the character, because the straighter sides carried more weight, since they didn’t as rapidly thin down to the thinner horizontal arches. To make the squarish ‘O’ seem the same weight as the original ‘O’ and to match the visual gray tone as the other capitals, I had to re-weight it, slightly shaving down the sides and arches. In a fixed-width font with a fairly strong stem weight like Lucida, this is another challenge.

If a fixed-width font is light in overall weight, it is easier to adjust visual spacing and weight, but Lucida Sans Typewriter (and hence Lucida Grande Mono and Lucida Console) has a relatively sturdy weight that makes it work well on back-lighted screens (which tend to erode the visual weight of a font), and also to work well in programming environments that use color to denote aspects of code. But, darker weight is harder to equalize because there is less white space available.

To get quantitative correlation of visual intuition, I rasterized the original ‘O’ and used photo software to count the percentage of black pixels within the cell, and did the same with the new squarish ‘O’. I then adjusted the outlines of the squarish ‘O’ sides and arches until the percentage of black pixels to total pixels was approximately the same as that of the original ‘O’, while keeping the visual look of the verticals in keeping with other capitals.



**Figure 4:** Hand-made (with Bézier splines) squarish Oh from Lucida Grande Mono (abandoned).

All that having been done, I made sixteen variants of the new, squarish ‘O’ and its siblings for the WGL character set of Lucida Grande Mono (which includes Unicode blocks Basic Latin, Latin-1, Latin Extended-A, part of Latin Extended-B, basic Greek, and basic Cyrillic). The new characters included ‘O’ versions with diacritics, OE digraphs, ‘Q’, Greek Omicron and Omicron-tonos, and Cyrillic ‘O’.

I didn’t make the Greek capital Theta squarish because, although traditionally similar in shape to Omicron, which is the ancestor of Latin ‘O’, Theta isn’t directly related to ‘O’, is a consonant not a vowel, isn’t confusable with an unadorned zero, and Greek mathematicians like Apollonius of Perga developed mathematics for traditional conics, not super-ellipses. Figuring I needed cultural advice on this, I asked the opinion of a Greek mathematician who uses Lucida math fonts, Antonis Tsolomitis, and he said that he didn’t think the Theta needed to match the squarish ‘O’.

After all those new ‘O’s and related characters were installed in the regular weight font, I made bold versions following the same process. The bolder weight made harmonization and fitting of the squarish ‘O’ with the rest of the capitals an even more elaborate process of refinement, because as weight increases in a fixed-width cell, the character has to be narrowed more in order to have adequate side spacing. Next, I obliqued both regular and bold versions to the same angle (11.3 degrees) and added them to the italic styles.

After the four Lucida Grande Mono faces were done, I then scaled down the squarish ‘O’s in the *y* dimension to make all those characters all over again for the four Console versions. All in all, the two font families, Lucida Grande Mono and Lucida Console, had 128 new squarish ‘O’-like characters (fig. 4).

(By this time, I wished Don had “thought different” (to convert an advertising phrase from Apple from imperative to past tense) and opposed the common mathematician-scientist preference for modifying the ‘O’, and had instead requested a modified zero. My task would have required only 8 new characters instead of 128, saving four binary orders of magnitude. And, if other scientists later asked for variant ‘O’s instead of zero, I could have resisted by citing Knuth’s great authority.)

After testing the final fonts for any last-minute problems, I generated OpenType fonts and sent them to Karl Berry at TUG, who enlisted Michael Sharpe to add OpenType tables to allow switching between the default Lucida Grande Mono and Lucida Console versions, with slashed zero and oval ‘O’, and the new “DK” version with open zero and squarish ‘O’.

I wish I could say that was the end of the process. But, it was not. It was only the end of the first, and ultimately discarded, phase.

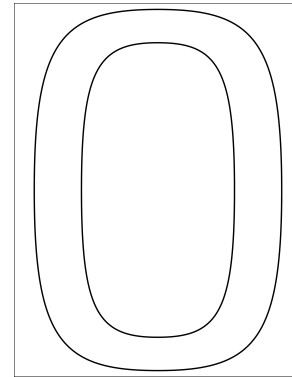
### Superelliptical-pi ‘O’s in Lucida Grande Mono

After Karl and Michael received the fonts and made a test OpenType version, I began to have second thoughts. I should say, “2nd order” thoughts.

I began to think that making a squarish ‘O’ by visual intuition wasn’t the optimal way to make a squarish ‘O’, at least not for Don Knuth. I figured I needed a more mathematical approach, although I am not a mathematician.

Hence, I investigated the superellipse numerically as well as visually. In Lamé’s algebraic generalization of the ellipse, the exponent of Don’s Computer Modern Typewriter squarish ‘O’ is 4, which Don presumably found clearly distinguishable from more or less elliptical zero, but by visual inspection, it was too squarish to harmonize with the fitting, weight, and look of the rest of Lucida Grande Mono.

Piet Hein’s superellipse has an exponent of 2.5, instead of the classical 2.0, and has been popular. Hein empirically settled on 2.5 as the exponent that gave the most satisfying compromise between the rectangle and the ellipse. That may be, but he was assessing the aesthetics of the shape in isolation, for instance, in his famous traffic round-about. From a long-ago infatuation with the superellipse, I have a small metal “super egg”, a superellipsoid of revolution of a superellipse. It has the charming ability to stand stably on one end, unlike a hen’s egg (barring Columbus’s demonstration). Also, I have a porcelain superellipse dish, suitable for baking a pie. So, I was personally acquainted with the superellipse in household objects.



**Figure 5:** Lucida Grande Mono DK Oh superelliptical-pi, outer and inner contours.

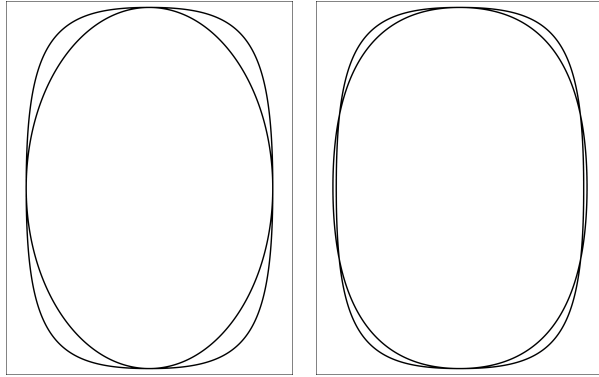
For a typeface design, the squarish ‘O’ can’t be a mere compromise between ellipse and rectangle. It has to distinguish ‘O’ from zero unambiguously. Not by a mere “just-noticeable” difference, nor a “just-preferable” difference, but an “obviously visible” difference.

So, I figured I needed to find a superellipse with an exponent somewhere between Hein’s 2.5 (not squarish enough) and Knuth’s 4 (too squarish for Lucida). So next I tried the so-called “natural superellipse” with an exponent of the natural logarithm base  $e$  (approximately 2.718). It was more squarish but still pleasing, and mathematically,  $e$  is transcendental and irrational, which, based on such cool names, I thought should be important qualities, although I can’t explain exactly why. Alas,  $e$  was still not squarish enough to be unambiguous.

Nevertheless, I didn’t give up, taking heart from Mark Twain’s observation in *A Tramp Abroad*: “A round man cannot be expected to fit in a square hole right away. He must have time to modify his shape.”

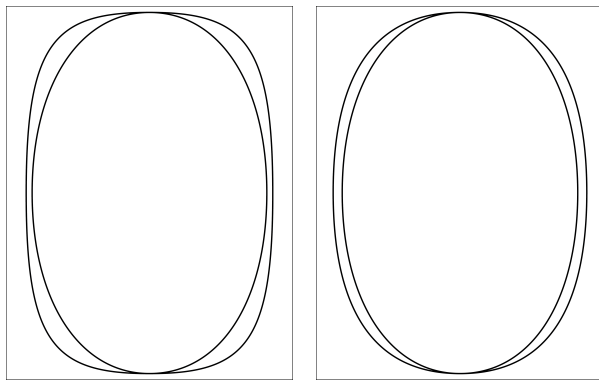
So I tried degree 3.0. Better, but still not quite squarish enough. So then I tried an exponent of 3.5, but that made the superelliptical ‘O’ look slightly too squarish, because the near-vertical sides and near-horizontal top and bottom didn’t rasterize at low resolutions with enough curvature to please me. At low-res, nearly but not quite straight curves often rasterize with infelicitous bit patterns. But, I was converging on something between 3.0 and 3.5. Perhaps 3.25 should have been next, but having tried  $e$ , I thought another transcendental irrational number might offer mathematical elegance, even if I couldn’t explain it. So I chose  $\pi$  (see fig. 5), the best-known transcendental number, which had been discovered by the Greeks in ancient times, and which is intimately linked to the circle.

With an OpenType font em only 1000 units in



**Figure 6:** Left: Lucida Grande Mono DK Oh superelliptical-pi outer contour, compared to true ellipse with same major and minor axes.

Right: Lucida Grande Mono DK superelliptical-pi capital Oh outer contour, with standard Lucida Grande Mono hand-drawn Oh outer contour intersecting. The superelliptical-pi contour is narrower, for more harmonious letter spacing and weighting.



**Figure 7:** Left: outer contour of superelliptical-pi capital Oh from Lucida Grande Mono DK (outer), with outer contour of zero (inner) for comparison.

Right: hand-drawn capital Oh contour with zero inside. Zero is narrower and has lesser shoulders.

height, in which the actual ‘O’ height was 759 units, and a fixed width cell only 603 units in width per character, in which the ‘O’ was only 519 units wide, a long decimal expansion of  $\pi$  was not needed, so I settled for an approximate exponent of 3.142. And even that was more precise than I was able to achieve in practice.

I used a superellipse calculator to plot a bitmap approximation, brought that into a font development tool as a background image, and hand-fit Bézier splines along the curves. As I worked on adjusting the shape, I found that, with the minimum number of four Bézier segments, I could not exactly model the plotted superellipse of exponent  $\pi$ . I wondered if

Bézier approximations to superellipses had the same sort of slight error as approximations to classic circles and ellipses, so I asked Berthold Horn, with whom we had worked so well on the original PostScript Type 1 versions of Lucida. He affirmed the error problem and sent me helpful plots showing the small differences. He suggested dividing the Bézier curves into smaller segments for greater accuracy, but, as I explain above, I resisted that suggestion because of the user-interface problem of messing around with the off-curve handles on on-curve spline joins of multiple curves. This is why I describe my process as “probably approximately incorrect” (fig. 6, left).

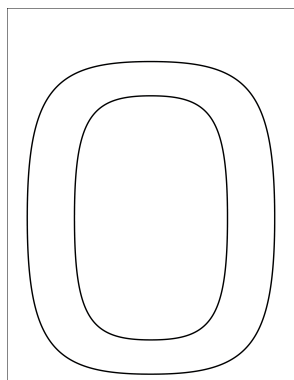
Nevertheless, considering the usual resolutions at which the letter would be rendered in text sizes on computer screens and printout, and the limitations of human visual acuity, the approximation seemed essentially as good as precision. For Lucida Grande Mono in 10 point text at 300 dots per inch on a laser printer, the ‘O’ height would be roughly 30 bilevel pixels tall, around 34 grayscale pixels on an iPhone Retina screen, around 42 grayscale pixels on the iPhone 6s Plus, and around 60 pixels on a 600 dpi printer. At those sizes and resolutions, tiny errors in approximating  $\pi$  seem negligible.

Figure 6 (right) compares the outer contour of the superelliptical-pi capital Oh for the DK font to the outer contour of the standard Lucida Grande Mono Oh. Figure 7 compares, on the left, the outer contours of the superelliptical-pi Oh with the zero from the same font, and, on the right, the hand-drawn original Oh with the zero.

### Superelliptical-pi ‘O’s in Lucida Console

The superelliptical ‘O’ may prove to be only a quaint curiosity in Lucida Grande Mono, but in Lucida Console, it resolves an important problem.

Lucida Console has shortened capitals to adapt it to the graphical shortcomings of a terminal window in Windows NT. It has functioned well for that purpose for more than 20 years, and has a distinctive look that people have adopted for general purposes, not just terminal windows. However, current programming styles like CamelCase and PascalCase (also called BumpyCaps, mixedCase, etc.) use compounded words or phrases in which the separate parts are signified by capitals. When the capital letters are shortened, as in Lucida Console, some pairs are harder to distinguish, especially capital ‘O’ and lower-case ‘o’. The  $\pi$  superellipse in the DK version of Lucida Console (fig. 8) solves the particular problem of distinguishing capital ‘O’ from zero, taller and more oval than the shortened ‘O’, and also distinguishes capital ‘O’ from lower-case ‘o’, which



**Figure 8:** Lucida Console DK Oh, superelliptical- $\pi$  contours (derived from the Grande Mono DK Oh).

is shorter and more circular.

After choosing  $\pi$  for the new exponent of the superelliptical ‘O’, I threw away the previous “shoulder pad” version and rebuilt all 128 characters but also added Theta, feeling that I should not deprive our Greek friends of the innovation (after all, they invented the forms of our capital alphabet), so there were 144 new characters to make. A gross of ‘O’s!

In summary, this is an extension to monospaced font families, beyond Computer Modern Typewriter, of Don Knuth’s innovation of superellipticity as a distinctive feature within a typeface design. A new way to solve the venerable and perennial ‘O’ versus zero problem.

I feel it would have been more accurate to have rendered the letters with the greater precision of Metafont than with the font tools I used, which involved hand-fitting Bézier splines. Contemporary WYSIWYG font design tools offer more user-friendly interfaces (although far from truly congenial) but less precision than Metafont, although there is no real need for such a trade-off. If desire for superelliptical figures gained currency, we might hope that in the future, font tool developers would integrate a superellipse function into their software, such as FontLab, Glyphs or RoboFont, some of which already offer an ellipse-drawing function. Knuth’s source code is publicly available, and published in *Metafont: The Program*. Probably, most font designers will not need a precise superellipse to solve the special problem of ‘O’ versus zero, but adjustable superellipticity could be useful in creating typefaces within the genre of superelliptical styles, as described further below.

Knuth’s innovative parameterization of elliptical versus superelliptical within a typeface to distinguish ‘O’ and zero is not, however, the first instance of squarish shapes in text typefaces. Type designers have shown a feeling for such forms, even without the aid of algebra.

### Historical designs with squarish shapes

The late Hermann Zapf designed Melior, released in 1952 as a news text face, with superelliptical forms. Zapf employed several techniques to craft Melior [6] — his elegant calligraphic shaping of letters with a slight superelliptical trait (harder to achieve by hand than in traditional calligraphy), his preternatural skill in drawing letters precisely at small sizes, and his ingenuity in combining traits of transitional and modern designs. Although initially intended for newspaper text, Melior achieved wider usage in magazines as well as in advertising typography and display, where its distinctive look was at once both modern and classical, appealing to typographers in search of a new look.

Aldo Novarese and Alessandro Butti’s Microgramma titling face of 1952 (some sources say 1951) also has a distinctive “squarish” look, although some letter shapes appear to owe more to the modernist concept that informed Marcel Breuer’s bent tubing furniture than to the superellipse per se. As a display face, Microgramma lacked lower-case, but Novarese’s Eurostile of 1962 extended the design concept to include lower-case, and provided a lower-case for Microgramma in its later releases. Microgramma and Eurostile were, and still are, popular in titling and logos supposed to evoke the future. This has often puzzled me because in nearly every movie I have ever seen about the future, nobody is reading, whether classical or superelliptical fonts. Everyone is too busy blasting with ray-guns or vaulting into hyper-space to relax and read a good book.

Of all designers, Zapf has explored superelliptical forms most extensively (fig. 9). After Melior, he created Hunt Roman, a private press face for the Hunt Botanical Library at Carnegie Mellon University. Developed in association with Jack Stauffacher, then the book designer for the library, Hunt shows subtle traces of Melior-like superelliptical forms. As a private press face, it was produced only in metal. Comenius is a later, commercial phototype relative of Hunt, which also shows hints of superellipticity. Two of Zapf’s type families for ITC are further explorations of the concept. Zapf Book has formal, Walbaum-like “Modern” high-contrast seriffed forms, while Zapf International gives a flowing, informal, hand-lettered look to active but slightly superelliptical shapes.

Zapf’s experiments with superelliptical designs continued with three of the earliest original digital typefaces that he designed for Dr.-Ing. Rudolf Hell’s “Digiset” digital typesetters in the 1970s and early 1980s. Marconi, intended for newspaper headlines



Melior  
**OHamburg**

Zapf Book  
**OHamburg**

Marconi  
**OHamburg**

Edison  
**OHamburg**

**Figure 9:** Some of Hermann Zapf’s “squarish” designs, all typeset at design size 36 pt.

and subheads, is a severely formal display family that combines Bodoni-like contrast, traditionally popular in newspaper headings, with superelliptical shapes. Its text companion, Edison [3], designed to save on newsprint costs while maintaining open contours that won’t clog or fill with news ink on high-speed presses, has a stunningly big x-height for a serifed design, nearly 53% of the body. At 8 point, Edison looks as big as Times Roman at 12 point. Although nearly 40 years old, it looks surprisingly new even today.

Zapf’s last typeface for Hell was Aurelia, released in 1983. Aurelia is a fascinating application of subtle superellipticity to the Venetian Humanist genre, essentially the first successful typographic roman type. Thus, Zapf combined some of the newest concepts in digital type design with some of the oldest in metal type. Inspired by the admirable humanist typeface in books printed by Nicolas Jenson in Venice in the 1470s, Zapf rendered Jenson’s definitive roman with a distinctively calligraphic touch first seen in Palatino, and with a hint of the superellipse first seen in Melior.

### Availability

The Lucida DK fonts are included in the complete Lucida OpenType font set, available through the T<sub>E</sub>X Users Group: <http://tug.org/lucida>. They are also available on their own to TUG members. If there is demand, they could be made available through B&H’s web site as well: <http://lucidafonts.com> (which provides many Lucida variants of all kinds not available elsewhere).

To conclude, here is one last example, showing the four variants each of the original Lucida Sans

Typewriter with the new Lucida Grande Mono DK and Lucida Console DK, all typeset uniformly at a nominal size of (approximately) 8 pt.

ABOQ xyz 012 LucidaSansTypewriterOT  
 ABOQ xyz 012 *LucidaSansTypewriterOT-Oblique*  
**ABOQ xyz 012 LucidaSansTypewriterOT-Bold**  
**ABOQ xyz 012 *LucidaSansTypewriterOT-BoldOblique***  
 ABOQ xyz 012 LucidaGrandeMonoDK  
 ABOQ xyz 012 *LucidaGrandeMonoDK-Italic*  
**ABOQ xyz 012 LucidaGrandeMonoDK-Bold**  
**ABOQ xyz 012 *LucidaGrandeMonoDK-BoldItalic***  
 ABOQ xyz 012 LucidaConsoleDK  
 ABOQ xyz 012 *LucidaConsoleDK-Italic*  
**ABOQ xyz 012 LucidaConsoleDK-Bold**  
**ABOQ xyz 012 *LucidaConsoleDK-BoldItalic***

### References

- [1] Berry, Karl. Production notes. *TUGboat* 34:2, pp. 181–181, 2013. <http://tug.org/TUGboat/tb34-2/tb107prod.pdf>.
- [2] Bigelow, Charles. Oh, oh, zero! *TUGboat* 34:2, pp. 168–181, 2013. <http://tug.org/TUGboat/tb34-2/tb107bigelow-zero.pdf>.
- [3] FontShop. Edison (LT). <https://www.fontshop.com/families/edison-lt#info>.
- [4] Knuth, Donald E. A footnote about ‘Oh, oh, zero!’, *TUGboat* 35:3, pp. 232–234, 2014. <http://tug.org/TUGboat/tb34-2/tb111knut-zero.pdf>.
- [5] Knuth, Donald E. *The METAFONTbook*, volume C of *Computers and Typesetting*. Addison-Wesley, Reading, MA, USA, 1986. (Notes on superellipticity on pp. 126ff.)
- [6] Zapf, Hermann. *Alphabet Stories: A Chronicle of Technical Developments*. RIT Cary Graphic Arts Press, Rochester, 2007. ISBN 978-1-933360-29-4. (Notes on superellipticity on pp. 30–31, 114–115.) <http://ritpress.rit.edu/publications/books/alphabet-stories.html>.

◇ Charles Bigelow  
<http://lucidafonts.com>



Superelliptical Apple pie, baked and photographed by Kris Holmes (co-designer of Lucida). Superellipse baking dish and super egg designed by Piet Hein. (In addition to apples, the pie filling contains a bit of quince and a hint of poncirus.)