**Bridges**
**Mathematical Connections**
**in Art, Music, and Science**

# The Endless Wave

Bob Brill
7 Lois Court
Ann Arbor, MI 48103, USA

bobbrill@migate.net
http://users.migate.net/~bobbrill

**Abstract**

This paper presents a selection of that portion of my algorithmic artwork that has been inspired by the sine function and especially by sine waves. I have produced a very large body of work based on a wide variety of algorithms. To keep this paper to a reasonable length I will focus on the one topic and offer but a few samples of sine-inspired imagery.

## 1. Embellished Lissajous Figures

Over the years the sine wave has proved to be an inexhaustable source of inspiration. It began for me with an exploration of Lissajous figures, which constitute a family of curves well-known to scientists and engineers who study the properties of waves. They are well-behaved, being continuous at every point, forming elegant, sweeping curves that reveal their derivation from the sine function, closing seamlessly upon themselves, and fitting neatly into any desired rectangle [1, 2, 3].

They can be generated by the following C program. One supplies values for the parameters (capitalized variables below). The closure function returns true when the generation of the curve is complete.

```
for (t = 0.0; !closure(); t += Tstep) {
  x = Xamplitude * sin(Xfrequency * t + Xphase);
  y = Yamplitude * sin(Yfrequency * t + Yphase);
  plot(x,y);
}
```

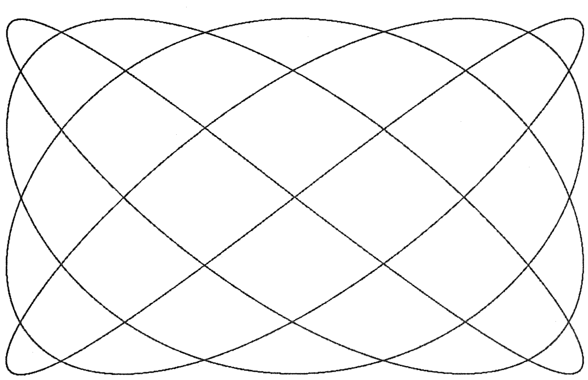


**Figure 1**

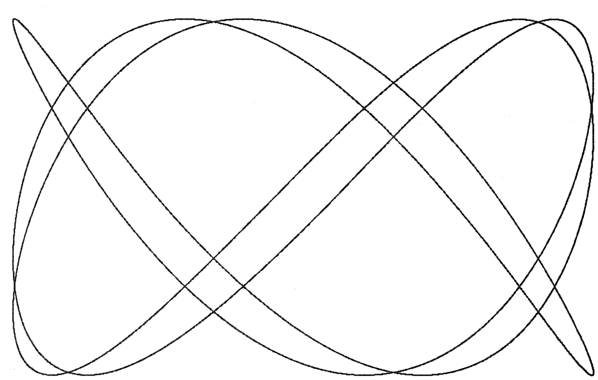


**Figure 2**

Instead of plotting closely spaced points of the curve as in Figures 1 and 2, the embellishment algorithm increases the distance between the calculated points and draws straight lines or curves centered on those points.

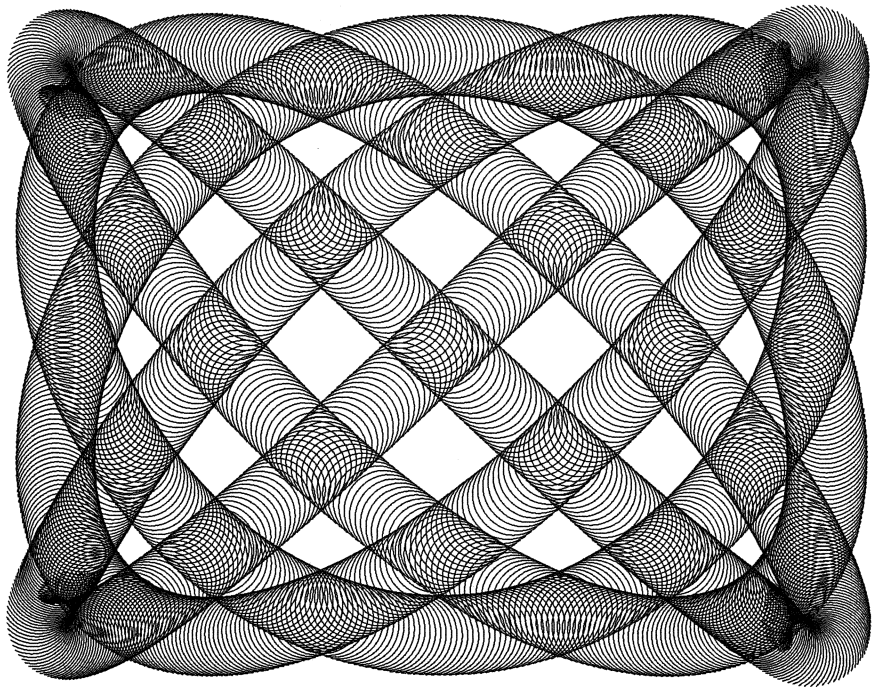
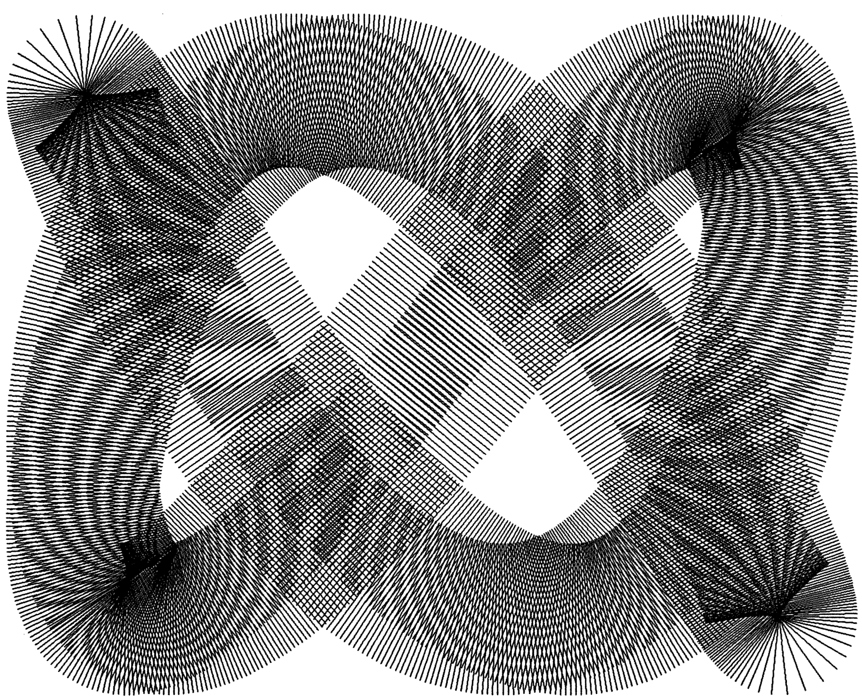


**Figure 3:** Figure 1 embellished with semi-circles.

**Figure 4:** Figure 2 embellished with straight lines.



## 2. Beyond Lissajous

After creating numerous variations in the embellishment, I began to explore ways of modifying the Lissajous equations themselves to produce new families of sine-related curves. One of the most interesting of these I call Lissajous Plus. After initializing **y** at 0, the equations are modified as follows:

```
x = Xamplitude * sin(Xfrequency * t + Xphase) - y * sin(F * t);
y = Yamplitude * sin(Yfrequency * t + Yphase) + x * sin(F * t);
```

where **F** may be any number, generally a small integer. See Figures 5 and 6.

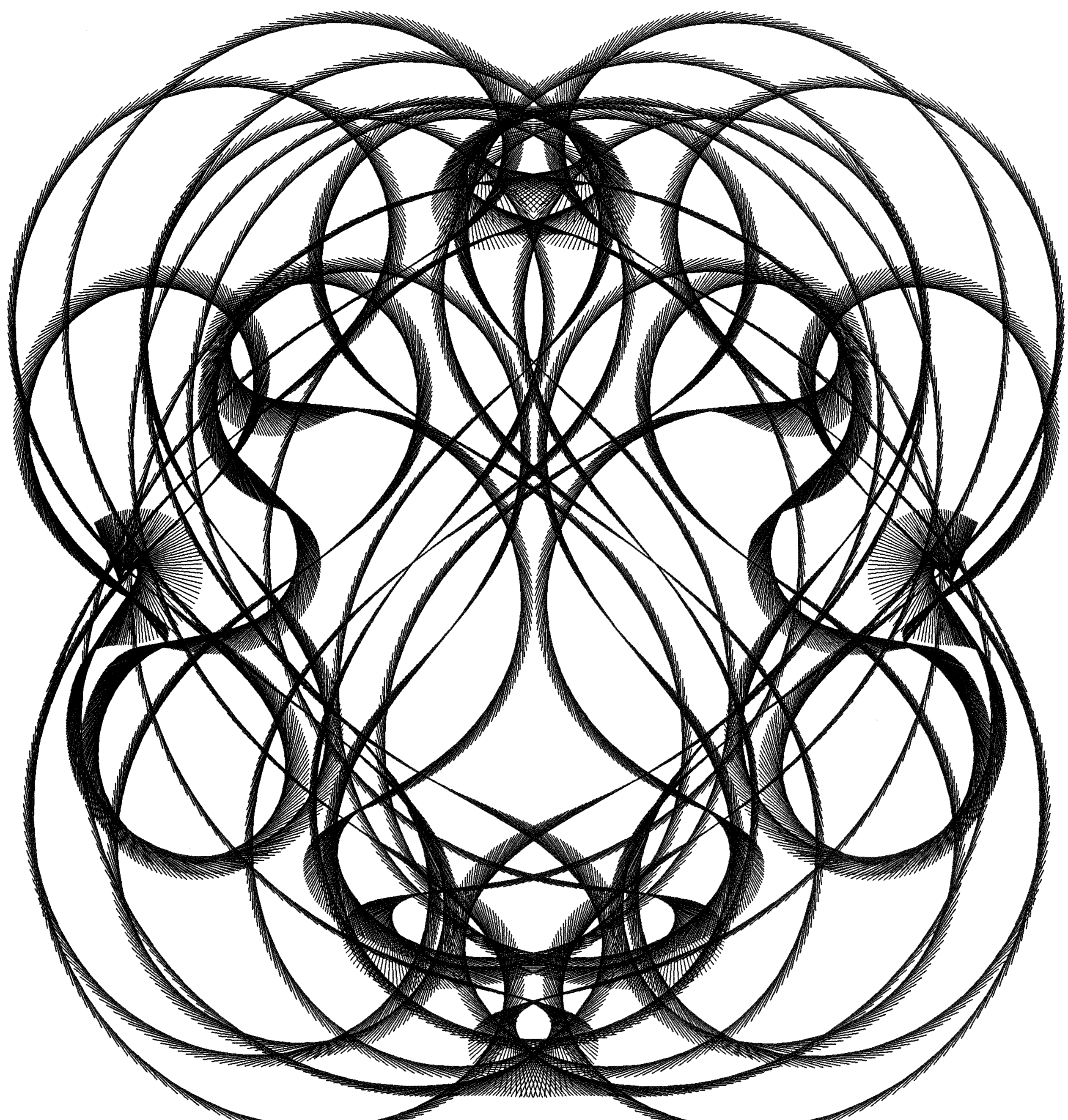


**Figure 5**

**Figure 6:** A Lissajous Plus figure embellished with sine waves.

I also discovered another open-ended class of sine-related curves by using compound sine waves (two sine waves added together) to specify the x and y coordinates. I call these Compound Lissajous Figures.

```
x = Xamplitude1 * sin(Xfrequency1 * t + Xphase1) +
    Xamplitude2 * sin(Xfrequency2 * t + Xphase2);
y = Yamplitude1 * sin(Yfrequency1 * t + Yphase1) +
    Yamplitude2 * sin(Yfrequency2 * t + Yphase2);
```

And combining Compound Lissajous Figures with Lissajous Plus:

```
x = Xamplitude1 * sin(Xfrequency1 * t + Xphase1) +
    Xamplitude2 * sin(Xfrequency2 * t + Xphase2) - y * sin(F * t);
y = Yamplitude1 * sin(Yfrequency1 * t + Yphase1) +
    Yamplitude2 * sin(Yfrequency2 * t + Yphase2) + x * sin(F * t);
```

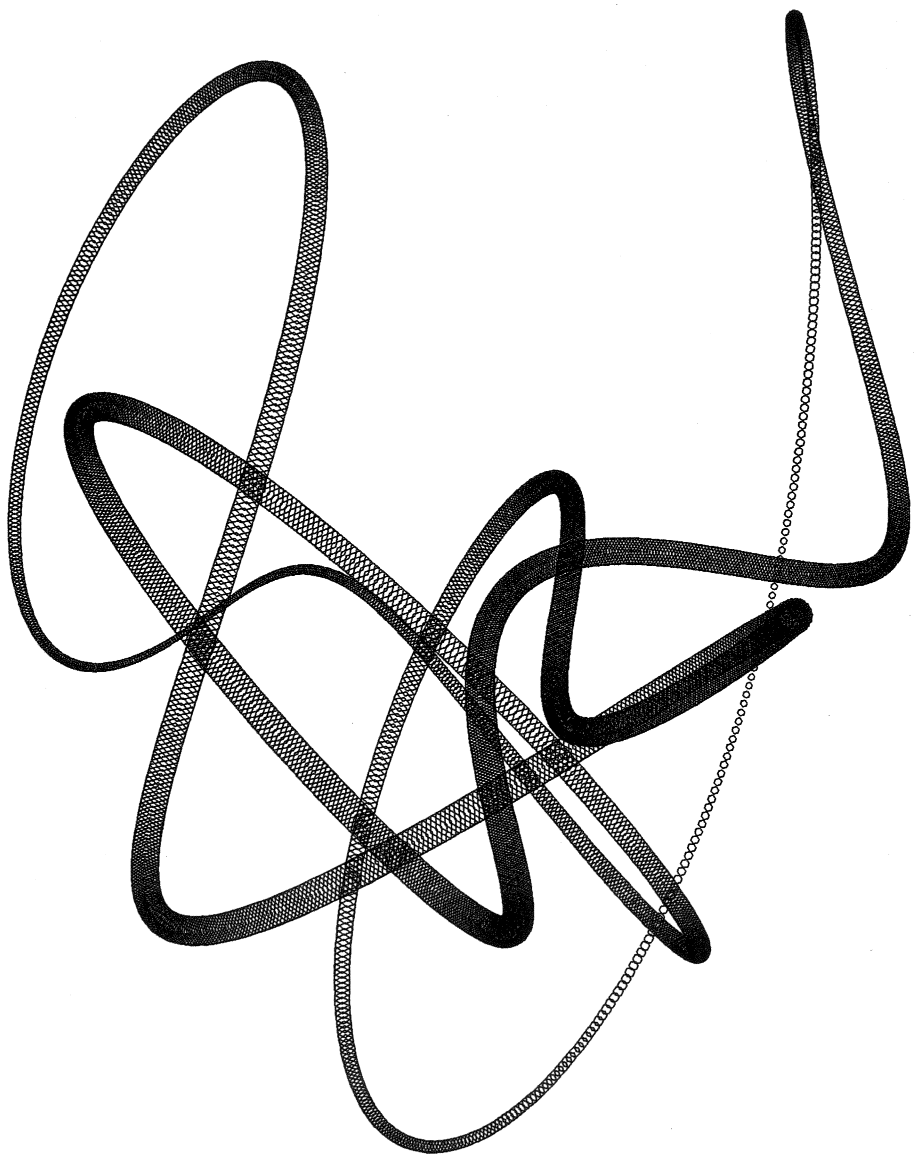


**Figure 7:** Compound Lissajous Plus

## 3. Concentric Deformed Circles

I found an appealing algorithm attributed to Norton Starr in [4]. It uses the sine function as well as the cosine function, which is just a phase-shifted sine function. The algorithm generates a set of concentric deformed circles. As each circle is drawn the radius varies and the path traversed periodically reverses direction, so that the circle turns into an undulating shape that sometimes contains loops. I added three new parameters to this algorithm, greatly expanding the range of possibilities as shown in Figures 8 through 11.

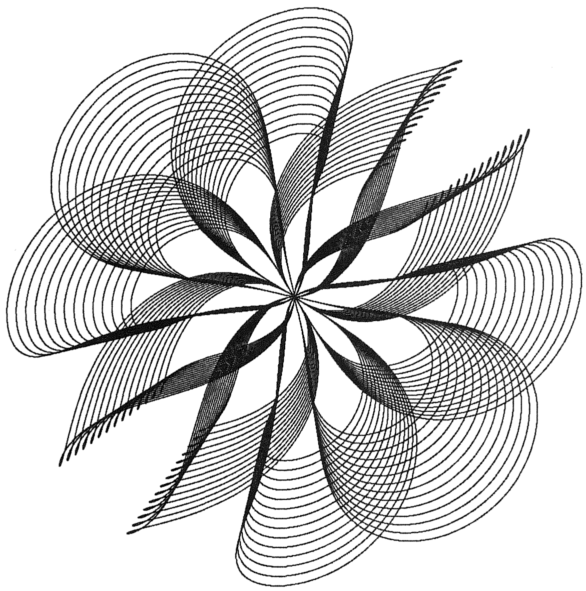


**Figure 8**

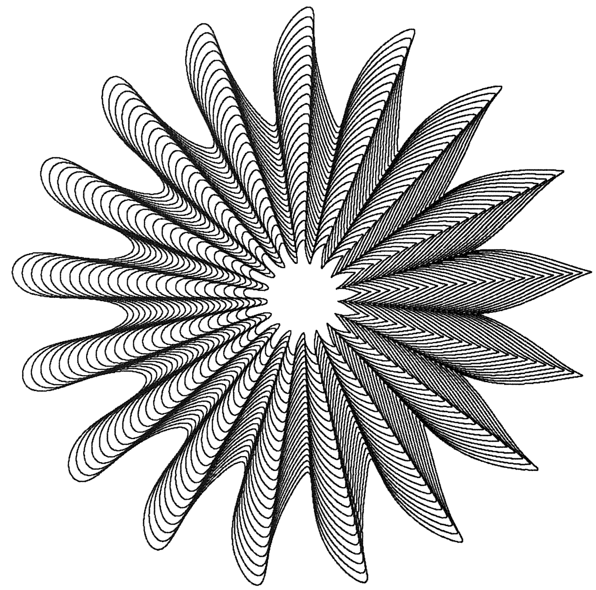


**Figure 9**

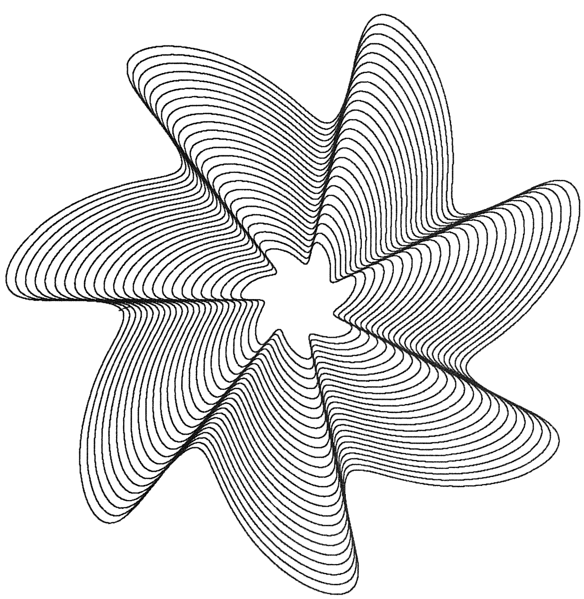


**Figure 10**

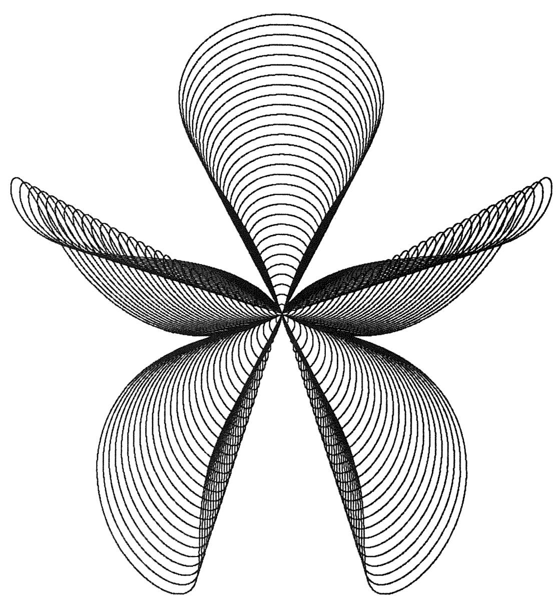


**Figure 11**

## 4. Waves

Figure 12 shows compound sine waves again, but this time stacked one behind the other, giving rise to wavy landscapes that could be water waves or rolling hills. An additional algorithm replaces the original colors with a gradient of gray shades.

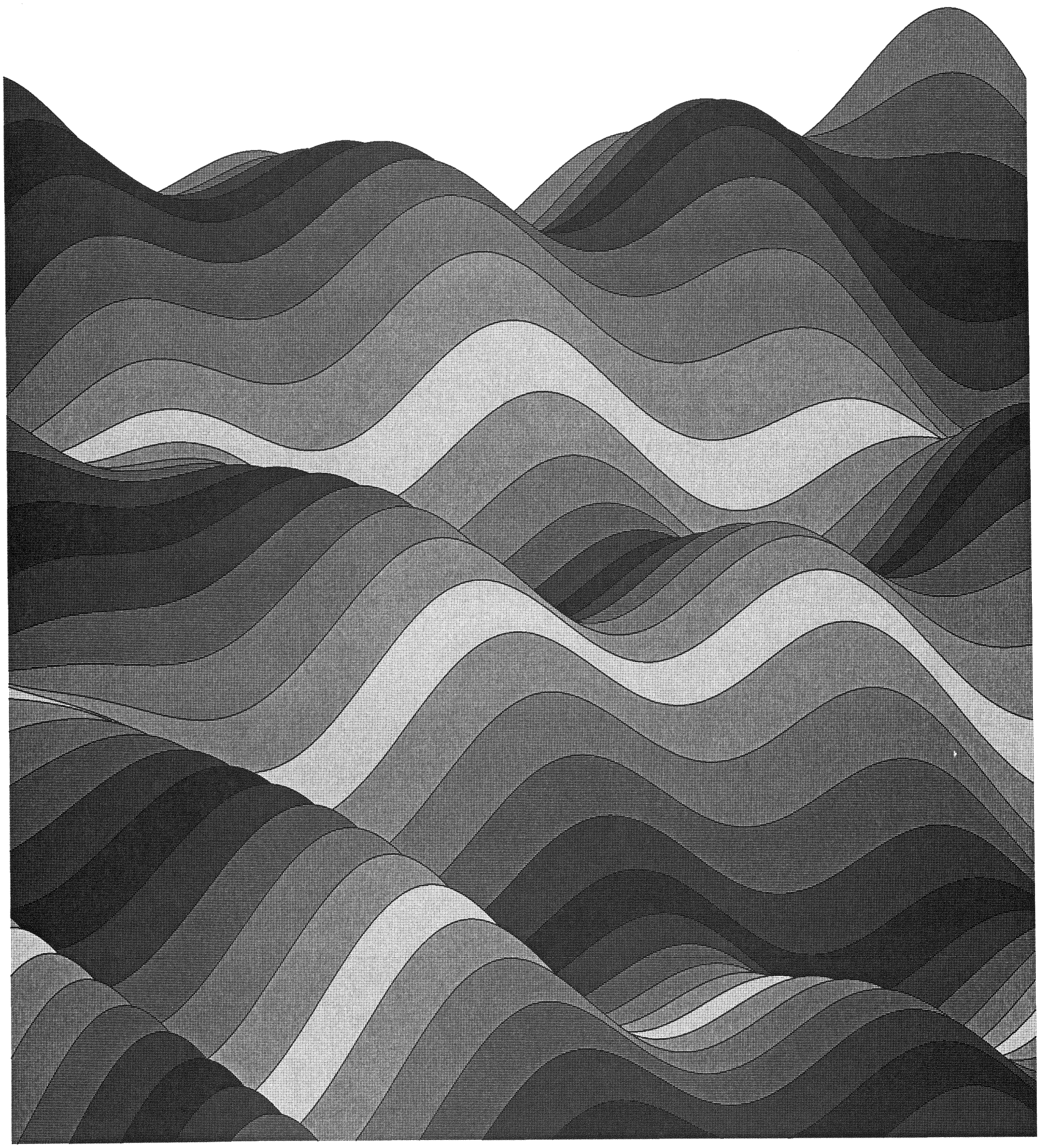


**Figure 12**

## 5. Lissajous Pursuit

One last look at Lissajous figures. Here I combined two themes I had long explored independently, Lissajous figures and pursuit curves. The embellishment here is a pursuit curve [5,6,7,8], symbolized by a hound chasing a hare. The hare is constrained to hop along the path of a Lissajous figure. The hound, starting at the center of the figure, draws a bead on the hare, a straight line is drawn between hound and hare, and the hound advances a certain distance along this path while the hare hops to its next position. The hound draws another bead on the hare and both creatures advance again. The algorithm terminates when the hound catches the hare. The parameters that determine when this happens, and ultimately what the image looks like, are the distances advanced by each creature at each iteration of the process. The image consists entirely of straight lines drawn between hound and hare. The wildly meandering curve, which is the path of the hound, is an emergent property of the algorithm, rising into visibility because the straight lines drawn are all tangent to this curve.

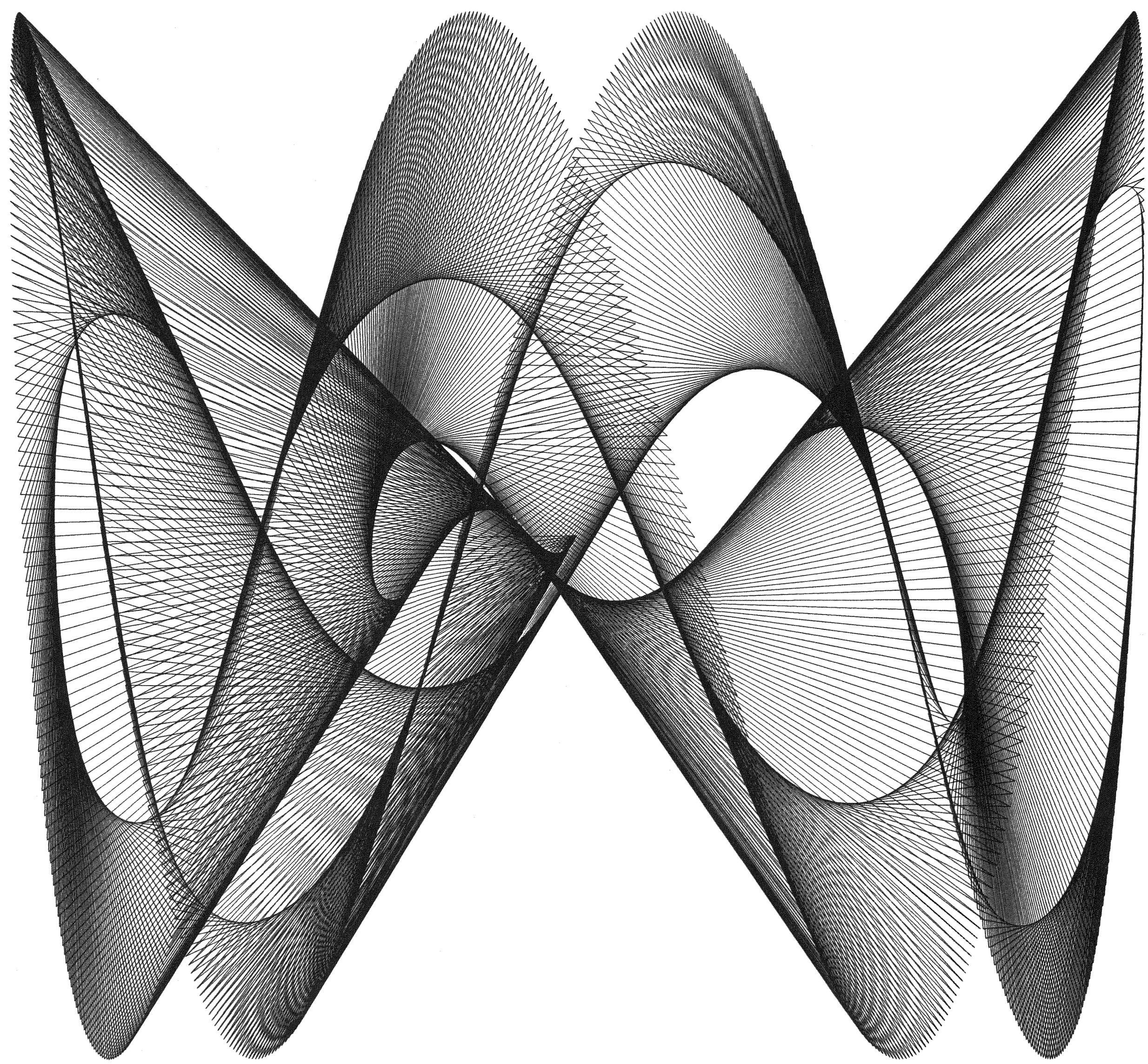


**Figure 13**

## 6. Passing a Wave Through an Image

When I saw Douglas Peden's art [9,10] I was struck by the beauty of his paintings. His Gridfield Geometry intrigued me. Peden does not use a computer to calculate his wavy grid lines. All is worked out by hand. I started writing a program to implement Peden's method, but I soon diverged onto a path of my own. Rather than building a wavy grid and laying colors into the cells as Peden does, I saw that I could easily take an already existing image and pass a sine wave through it in either the x or y directions or both, making the whole image go wavy. Figure 14 shows Figure 13 after passing a wave through it in both the x and y directions.

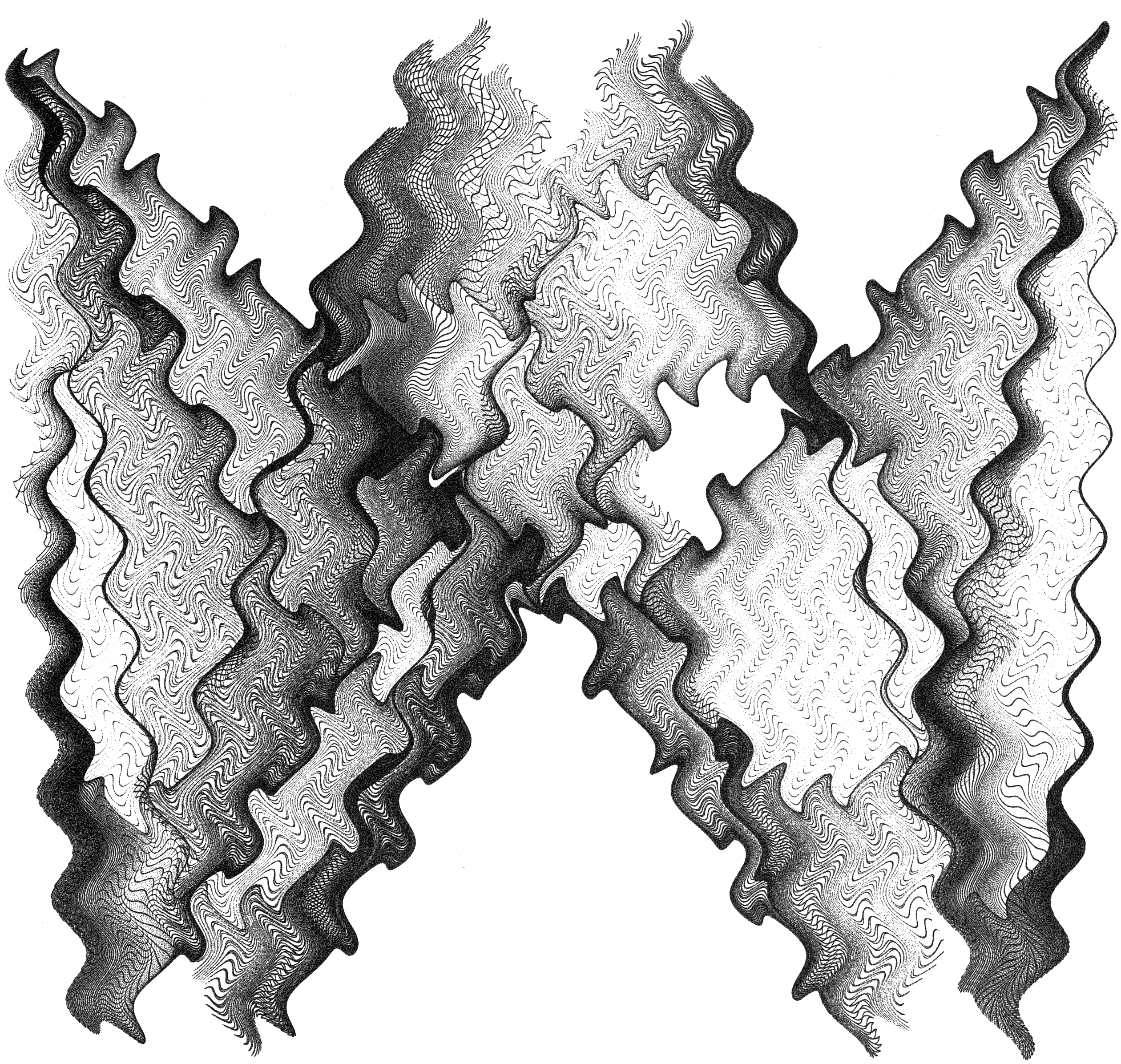


**Figure 14**

## 7. Finding Imagery in the Real Plane

My interest in algorithmic art began with a 1986 article by A. K. Dewdney in Scientific American [11]. It contained an algorithm that I enjoyed exploring and later expanded. The algorithm establishes a one-to-one correspondence between the pixels of an image rectangle and an equal number of evenly spaced sample points from a rectangular region of the real plane. The x and y coordinates of each sample point are input to a mathematical expression in x and y, which when evaluated, returns a color value. The corresponding pixel is painted that color. I've tried many different expressions, most of them not involving the sine function. Figure 15 shows an image in which the sine function did play a role. The original colors have been replaced by shades of gray.

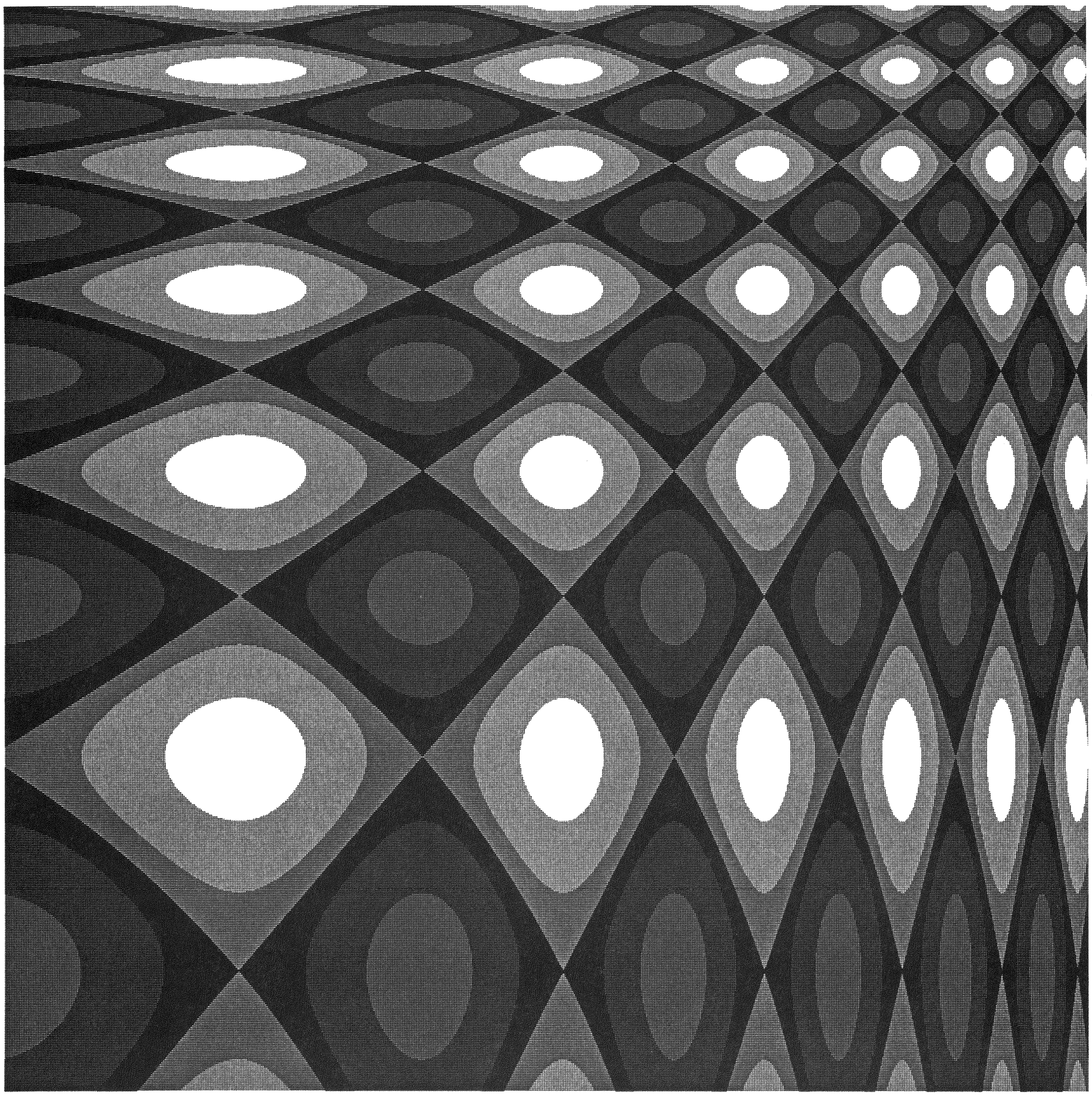


**Figure 15:** $e^{y}\left(\sin\left(\frac{1}{x}\right)\right) + e^{x}\left(\cos\left(\frac{1}{y}\right)\right)$

## 8. Finding Imagery in the Complex Plane

The same game can be played on the complex plane simply by ensuring that all calculations follow the rules for complex arithmetic on numbers of the form **x + yi**. In Figure 16 note that **tan(z) = sin(z) / cos(z)**. The color value is derived from the real term of the evaluated expression.

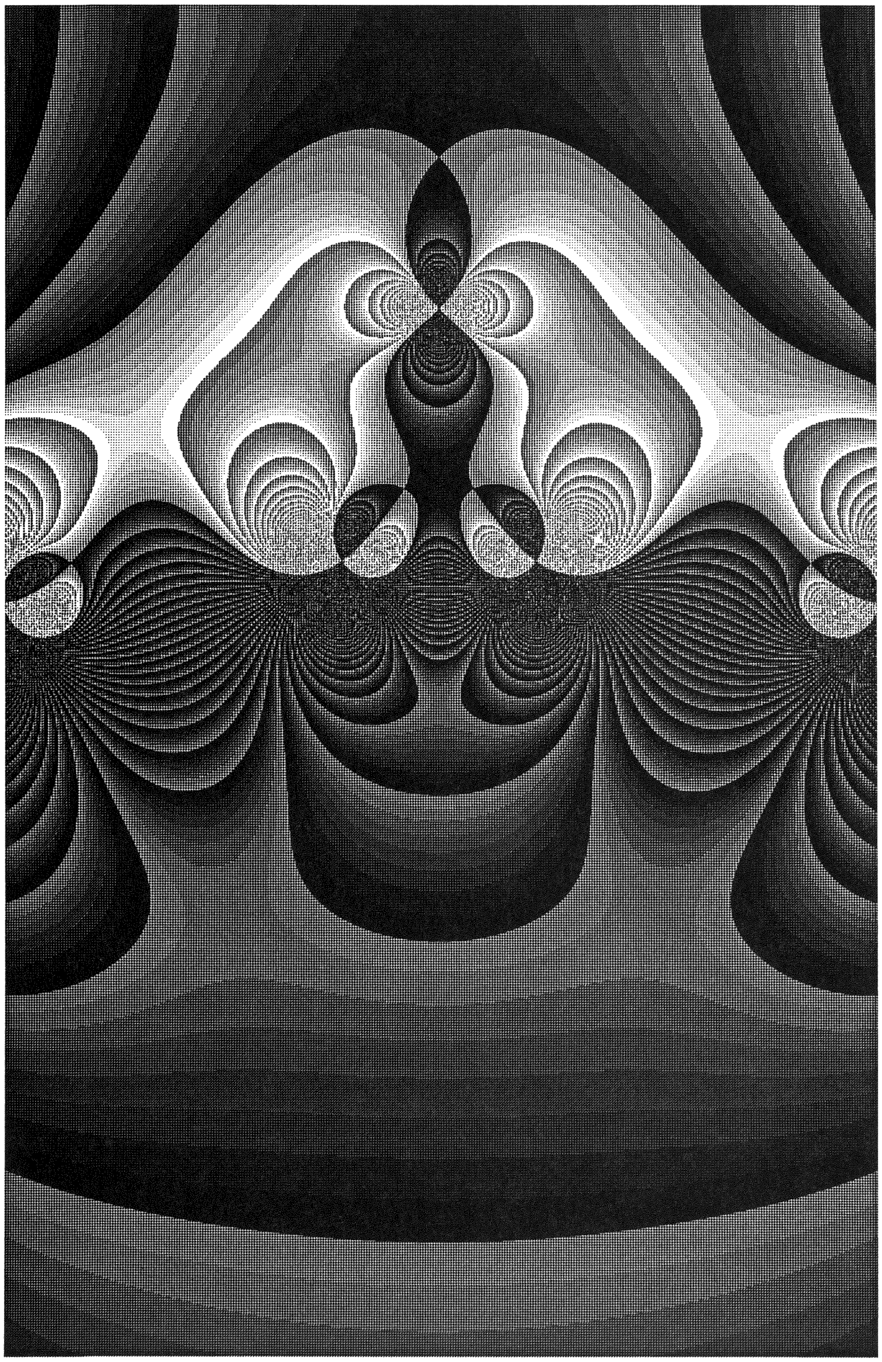


**Figure 16:** $\left(\frac{z}{\tan z}\right)^2$

## 9. Concluding Remarks

The Endless Wave goes on and on. I could fill a book with sine-inspired imagery already lying in my files. And while I was writing this paper several new ideas came to me which I have not yet had time to explore. I invite the interested reader to join the fun of exploring this limitless domain. For greater detail on the algorithms mentioned in this paper see [3]. This is the companion book for the software I use for creating art. It contains a whole chapter on Lissajous figures, showing many variations. Visit my website at *http://users.migate.net/~bobbrill* if you wish to order the book. (The software comes free with the book.) All the illustrations for this paper were made with this software. For any algorithms discussed here, but not covered in the book, email me at *bobbrill@migate.net*.

## References

[1] Brill, Bob, "Embellished Lissajous Figures" in *The Pattern Book*, edited by Clifford Pickover, World Scientific, 1995, pp. 183 - 193.

[2] Peterson, Ivars, *Fragments of Infinity: A Kaleidoscope of Math and Art*, John Wiley & Sons, 2001. My work is discussed and illustrated in this book. Embellished Lissajous Figures (pp. 4 - 5), Poincaré Recurrrence (pp. 94 - 96, 99 - 102, plate 9), Pursuit Curves (p. 103).

[3] Brill, Bob, *Algorithmic Imagery Using E: A Tutorial and Reference Manual for E-Run and the E Language*, Zenagraf, 2001.

[4] Wagon, Stan, *Mathematica in Action*, W. H. Freeman, 1991, pp. 137 - 140.

[5] Peterson, Ivars, "Art of Pursuit" in Science News online edition, MathTrek column for week of July 21, 2000, dealing with my Pursuit Curve images.
http://www.sciencenews.org/20010721/mathtrek.asp

[6] Sharp, John, "In Pursuit of Pursuit Curves" in *ISAMA Conference Proceedings*, 1999, pp. 429 - 436.

[7] Boyd, Rutherford, "Mathematical Ideas in Design" in *Scripta Mathematica*, Vol. 14, No. 2, June, 1948, pp. 128 - 135.

[8] Gardner, Martin, "Mathematical Games" in *Scientific American*, July, 1965.

[9] Peden, Douglas D., "Bridges of Mathematics, Art, and Physics" in *Bridges Conference Proceedings*, 1998, pp. 73 - 82.

[10] Peden, Douglas D., "The Influence and Use of Music and Mathematics in My Art" in *Bridges Conference Proceedings*, 2001, pp. 223 - 231.

[11] Dewdney, A. K., "Computer Recreations (Wallpaper for the Mind)" in *Scientific American*, Sep. 1986, pp. 14 - 23.