

Voderberg Deconstructed & Triangle Substitution Tiling

Cye H. Waldman

[cye@att.net](mailto:cye@att.net)

Copyright 2014

**Abstract**

The Voderberg is one of an infinite number of tiles that can be created by deforming an isosceles triangle. Analytical expressions are derived for the segment lengths and internal angles of a generalized Voderberg tile. This generalized tile has all the familiar properties of the Voderberg tile, including spiral tiling and the ability of a pair of tiles to completely enclose a third tile or even a pair of tiles. These tiles can be readily tessellated in a radial pattern and in spiral patterns using Goldberg's method. We present an algorithm for converting lengths and angles into coordinates in the complex plane and a general program with examples for triangle substitution tiling. The advent of deformable tiles has permitted the creation of interesting and unique animations.

**Introduction**

Wikipedia says it most succinctly, "The first spiral monohedral tiling was discovered by Heinz Voderberg in 1936, with the Voderberg tiling having a unit tile that is a nonconvex enneagon." It is arguably the most popular and reproduced of all spiral tilings. We resisted Voderberg tiling for a long time for a couple of reasons. First of all, they are extremely common. Many have tried the tilings. Thus, our first concern was whether we had anything new to contribute. Then too, while many want to show the Voderberg spiral, very few actually discuss how to do it. We found two graphical methods and one iterative numerical solution.

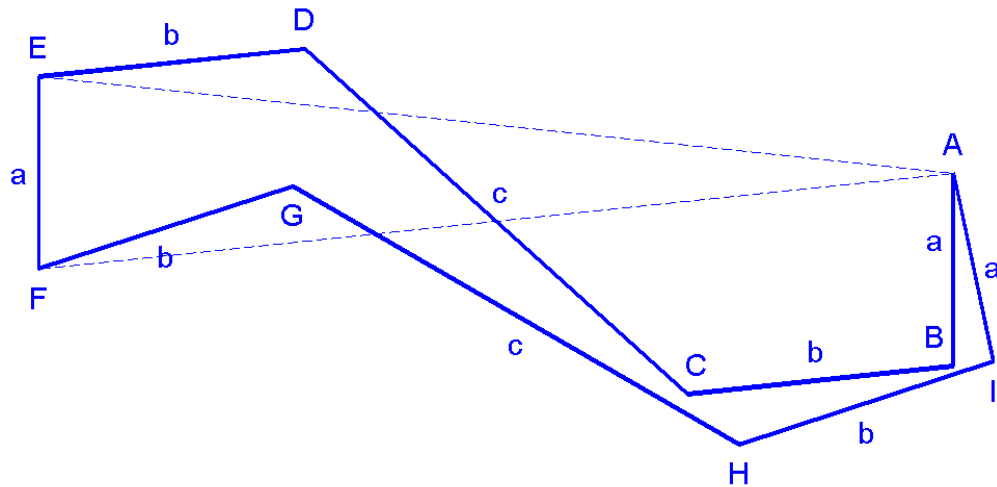
In this paper we discuss the equations for a generalized Voderberg tile that is parameterized in terms of a single angle; all segment lengths and internal angles can be expressed in terms of it. Then we demonstrate how to turn those lengths and angles into a tile in the complex plane. Next we address the question of creating radial and spiral tilings by the method of triangle substitution tiling. A brief look at other spiral tilings puts the present work in perspective.

Finally, we present a complete Matlab program for triangle substitution tiling with several built-in tiles. This program can be readily modified by the user in order to modify or add tiles.

**Technical Discussion**

VODERBERG TILE

Figure 1 shows a generic Voderberg tile and its related isosceles triangle with a  $12^\circ$  vertex. It is a bent enneagon and here we have adopted the notation of Dutch (1999).



**Figure 1: A generic Voderberg tile.**

In order to assure tessellation, the following conditions must be met: (1)  $ABCDE$  is congruent with  $AIHGF$ , i.e., the latter being the former rotated through the vertex angle, (2) there are only three unique lengths,  $a$ ,  $b$ , and  $c$ , and (3) the vertex angle must be  $12^\circ$ .

We found the following descriptions of Voderberg tile development: Trott (2006) gives an iterative procedure for the specific case where  $a = b$ . This procedure is not only very limiting, but it is unnecessarily complicated. Meissner (2009, in German), shows a graphical method for constructing a Voderberg tile, but gives no information as to segment lengths and internal angles. Roskes (2008) does much the same thing, but goes much further in that she shows how to actually create radials and spirals, albeit graphically. Again, there is no specific calculation of the lengths and angles. Moreover, these graphical methods are somewhat random, such as, draw a line between here and there. In order to create tilings mathematically, we need precise definitions of the tiles in order to assure tessellation and reproducibility.

After much gnashing of teeth and pulling of hair, and, yes, trigonometry and the law of sines, we determined that the entire solution to the Voderberg tile could be parameterized in terms of a single parameter, which we chose to be the large obtuse angle at  $D$ , hereafter referred to as  $\beta$ , as well as the vertex angle at  $A$ , hereafter,  $\alpha$ , which is known. It was assumed here, without any loss of generality, that the base of the isosceles triangle is unity, i.e.,  $a = 1$ .

Table 1 and Eqs. (1) show the results of the analysis. The table shows the angle and segment lengths (counterclockwise) for each node in Figure 1.

Node	Angle	Length
A	$\alpha = 12^\circ$	1
B	$\frac{3\pi - \alpha}{2}$	$x$
C	$2\pi - \beta$	$L$
D	$\beta$	$x$
E	$\frac{\pi + \alpha}{2}$	1
F	$\frac{\pi - 3\alpha}{2}$	$x$
G	$2\pi - \beta$	$L$
H	$\beta$	$x$
I	$\frac{\pi + \alpha}{2}$	1

**Table 1: Properties of the generic Voderberg tile.**

$$L = 2 \sin\left(\frac{\pi - \alpha}{2}\right) / \cos\left(\beta - \frac{\pi}{2}\right) \quad (1)$$

$$x = \frac{\csc(\alpha/2) - L \cos(\pi - \beta)}{2} - \sin(\alpha/2)$$

We found that  $111 \lesssim \beta \lesssim 153^\circ$  will assure that the two legs do not cross each other. As a parity check, you can see that the sum of the internal angles is  $7\pi$ , consistent with an enneagon. Moreover, we validated that for any such values of  $\beta$  we can completely enclose one or two tiles with only two tiles. This is exemplified in the associated animation. Likewise, we showed that all these tiles will tessellate both radially and spirally, again as seen in the animations. In the following sections we will describe exactly how we generated these tessellations. But first we must develop a proper tile for use in the algorithms.

## DEVELOPING TILES FOR THE COMPLEX PLANE

The above solution for the Voderberg tile is not very useful for tiling in the complex plane. We need to transform it into a single complex vector that gives the coordinates of each node. Then we can readily translate and rotate the tile where we need it in the plane.

The method we developed is based on the concept of fractal linkages, as given by Gazalé (1999). He begins with the concept of an old-fashioned surveyor's chain composed of articulated links of unit length. If the links are laid out to conform to some shape, then each link can be thought of as a unit vector given by  $e^{i\varphi}$ , where  $\varphi$  is the polar angle. Thus, the nodes of the chain,  $z_n$  are given by the cumulative sum  $\sum_{k=1}^n e^{i\varphi_k}$ . This works quite nicely if you know the external angles of your tile, as you might if developing the tile on square or triangle graph paper, for example. More generally, the internal angles,  $\theta_n$  of the tile are known and the sides,  $s_n$  may not be of unit length.

You can readily determine that the external angles are given by the cumulative sum  $\varphi_n = \sum_{k=1}^n \pi - \theta_k$ .

Variable lengths can be accommodated by repeating the unit vectors, or more simply by multiplying the link vectors by the appropriate lengths. Thus, given

$$S = [s_1, s_2, s_3, \dots]$$

$$\Theta = [\theta_1, \theta_2, \theta_3, \dots]$$

there follows

$$\varphi_n = \sum_{k=1}^n \pi - \theta_k$$

$$z_n = \sum_{k=1}^n s_k e^{i\varphi_k}$$

The Matlab code is given simply by

```
S=[s1;s2;s3;...];
THETA=[theta1;theta2;theta3;...];
PHI=cumsum(pi-THETA);
z=cumsum(S.*exp(i*PHI));
z=[0;z]; % close the tile
```

TRIANGLE SUBSTITUTION TILING

Triangle substitution tiling has been described by Gardner (2001), and in greater detail by Grünbaum and Shephard (1986). Both attribute the idea to Goldberg (1955), who pointed out that if we take a radial tiling of isosceles triangles and slice it in half, the two half-planes can be shifted by one or more tiles to make a spiral form. The general idea is shown in Figure 2. Moreover, the triangles can be distorted in infinite ways by replacing the legs with congruent lines, such as those in Figure 1. As Gardner pointed out, Voderberg created his spiral “in a complicated procedure. Goldberg’s method of obtaining it makes it almost trivial.”

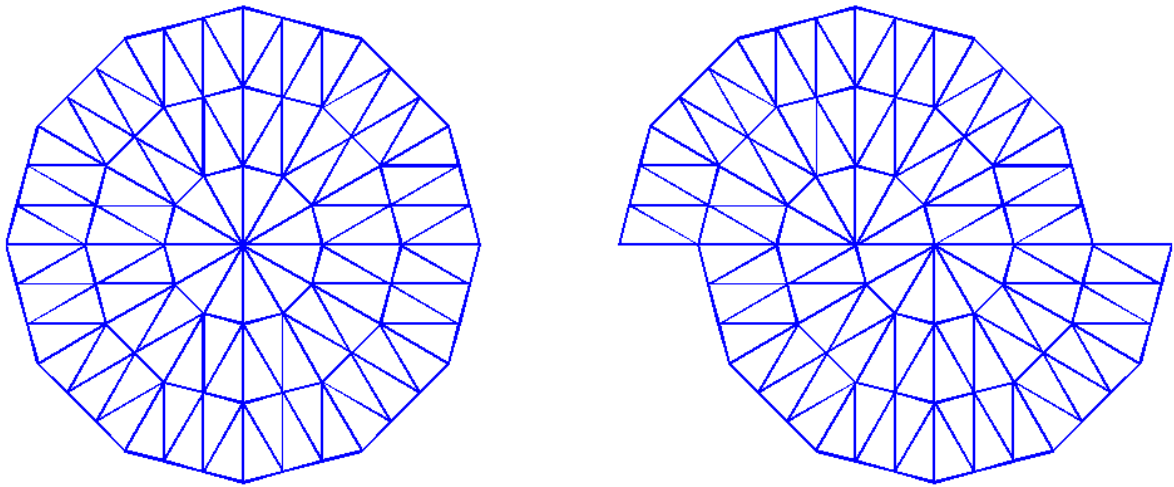
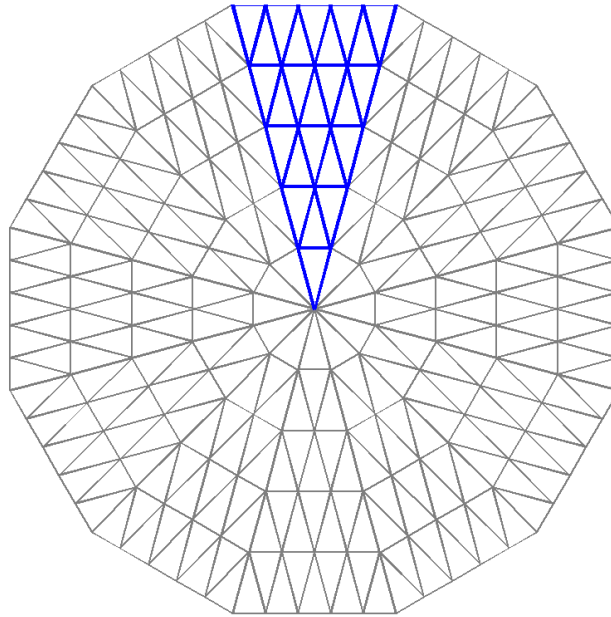


Figure 2: Creating spiral from radial tilings.

We developed a simple and (almost) foolproof method for creating these tilings that we call V-A tiling. Here, V represents a tile standing on its head, while A is the tile resting on its base. Radial tiling can be constructed by corona or by sector. The latter is simpler and will be adopted here. Referring to Figure 3, we see that the core consists of a single V-tile for each sector. Within the sector, the first corona consists of three tiles, in the order V-A-V, the second corona has five tiles, V-A-V-A-V, and so on. In all cases, we assume that the base of the triangle (or its modification) is unity. Assuming that we start with the V-tile, the A-tile is usually given by  $A = -V + \frac{1}{2} + i \cdot h$ , where  $h$  is the height of the tile, i.e.,  $h = \frac{1}{2} \tan(\alpha/2)$ . Here, we have defined A such that it is tessellated with V. You can see that the  $n^{\text{th}}$  corona consists of  $n$  A-tiles and  $n+1$  V-tiles. Thus, for the  $n^{\text{th}}$  corona,

$$z_n = \left[ [V A] [V A]_{+1} [V A]_{+2} \dots [V A]_{+n-1} V_{+n} \right]_{-n + \frac{1}{2} + i \cdot n \cdot h} \quad (2)$$

And this applies to substitution tiles, such as the Voderberg, as well.



**Figure 3: V-A tiling.**

To set up for spiral tiling, it is convenient to rotate the sector  $Z$  counterclockwise so that it rests on the negative  $x$ -axis. Assuming the origin is at the vertex of the sector, we can say

$$Z = Ze^{i\pi/2}e^{-i\alpha/2} \quad (3)$$

The spirals are then created quite simply as follows:

$$Z_{spiral} = [Z_{top} \quad Z_{bot} + g\Delta x] \quad (4)$$

where  $Z_{top}$  and  $Z_{bot}$  are the sectors in the top and bottom halves of the plane, respectively,  $g$  is the number of the Goldberg shift, and  $\Delta x$  is the leg length of the reference isosceles triangle, to wit

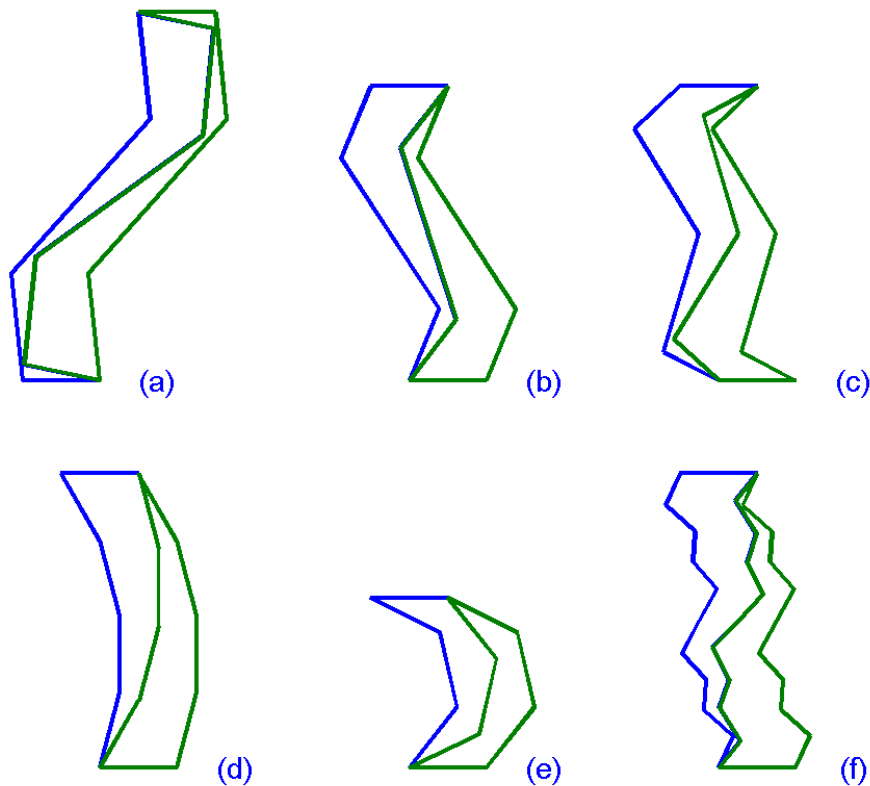
$$\Delta x = 1/2 \sin(\alpha/2) \quad (5)$$

This is amazingly simple, but there are a few caveats that must be noted. First, the vertex angle of the reference triangles must be *mod* 360 and the number of sectors must be even. Second, while it's true that any distorted triangle will tessellate radially, only those whose legs are antisymmetric with respect to their centers, will spiral as advertised. When they are symmetric, special rules apply; these will be discussed in the next section.

Let us just remark in passing that you could work with an odd number of sectors on an *ad hoc* basis. We've done stranger things, including vertex angles that were not *mod* 360.

## SOME SAMPLE TILES

Figure 4 shows the V-A configuration of the six tiles (in addition to the isosceles triangle) that are included with the enclosed program. It's no coincidence that the first five of these tiles are the same as those in Grünbaum and Shephard (1986); it was merely expedient. We have, however, reimagined the top three tiles, i.e., (a,b,c) as deformable tiles that can be varied over a range of shapes. Tiles (d) and (e) are based on specific polygons and are fixed in shape. Tile (f) is a random icosihenagon (21-gon) that we developed for this program. It will be used as the basis for describing the process of creating new tiles below.



**Figure 4: Built-in tiles for the enclosed triangle substitution tiling program.**

The term *deformable tiling* was suggested by Grünbaum (2014) and is being adopted here.

Table 2 shows the properties of these tiles. The ‘Name’ is simply the way we reference them, ‘Description’ is just that, and ‘Figure’ refers to its position in Figure 4. Now, the columns ‘Deformable or free parameter’ and ‘Range’ refer to those tiles that allow the user to create tessellations with variations of the same basic tile shape. However, only the Voderberg, Lightning, and Sigma tiles permit a continuous variation in the tile shapes that allow us to animate tilings. These parameters are used in the program input.

Name	Description	Figure	Deformable or free parameter	Range	Asymmetry
<b>Triangle</b>	Isosceles triangle		$\alpha$ , vertex angle	$\text{mod } 360$ $2\pi/\alpha$ even	TRUE
<b>Voderberg</b>	Bent enneagon	(a)	$\beta$ , obtuse angle	$\beta \sim 111-153^\circ$	TRUE
<b>Lightning</b>	Bent heptagon	(b)	$\ell$ , a length	$\ell \lesssim 1.75$	TRUE
<b>Sigma</b>	Bent enneagon	(c)	$\ell$ , a length	$\ell \sim 0.72-2.65$	FALSE
<b>Bent Wedge</b>	Another enneagon	(d)	NA	FIXED	FALSE
<b>Tent</b>	Reflexed heptagon	(e)	NA	FIXED	FALSE
<b>Random</b>	A random icosihenagon	(f)	$\alpha$ , vertex angle	$\text{mod } 360$ $2\pi/\alpha$ even	TRUE (can be user-modified to FALSE)

**Table 2: Properties of the built-in tiles.**

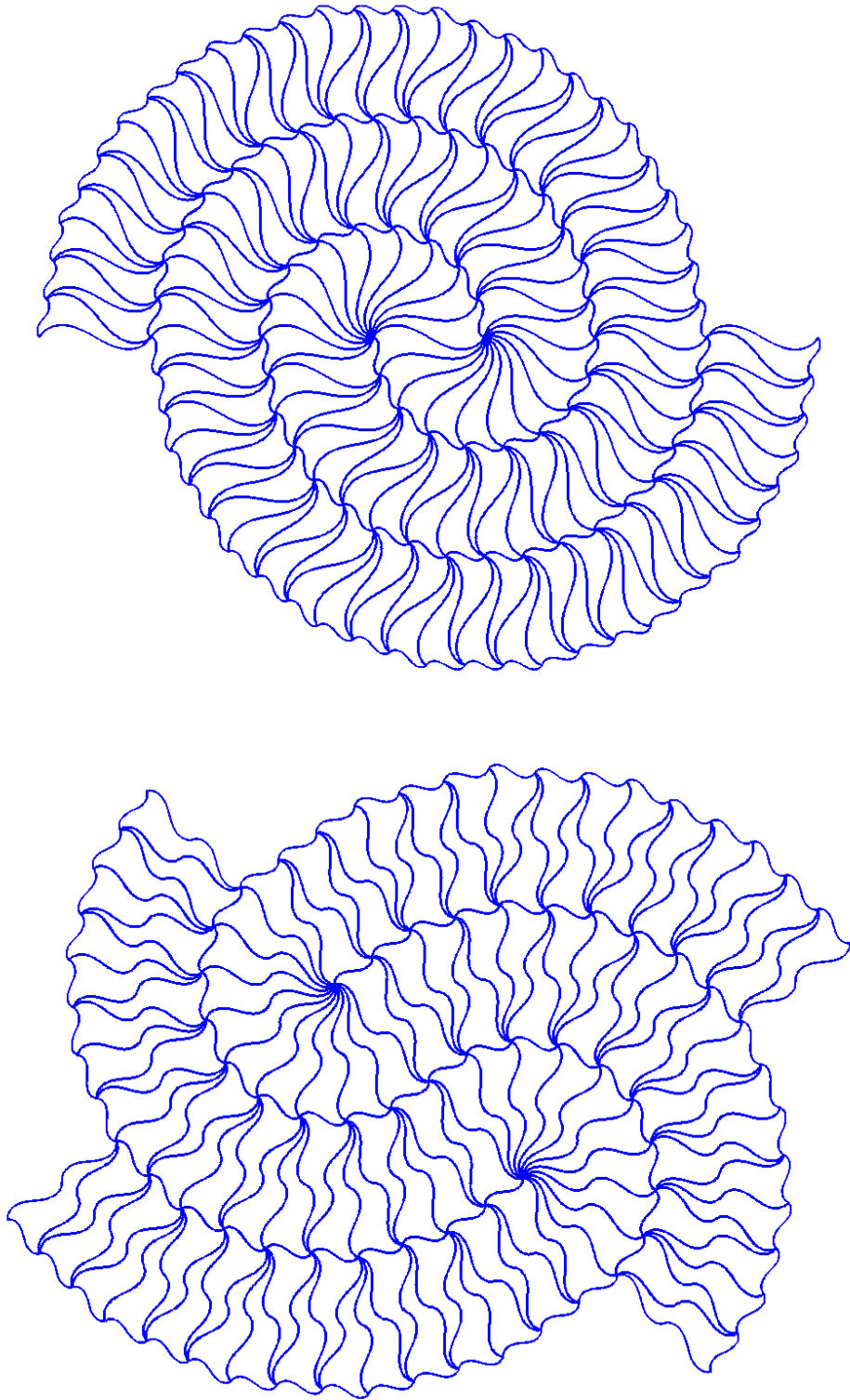
The reason we differentiate between antisymmetric and symmetric tiles is that they tessellate differently, and this will impact the Goldberg shift. In plain English, antisymmetric tiles tessellate with themselves and symmetric tiles tessellate with their mirror (conjugate) selves.

The upshot of this is that in order to tessellate spirals of symmetric tiles, some modifications are required. Specifically, (1) the radial coronas must alternate between the specified tile and its conjugate and (2) for an even number Goldberg shift the procedure is the same as for antisymmetric tiles, as shown in Eq. (4), however, for an odd number shift,  $Z_{bot}$  must be replaced by  $-Z_{bot}^*$ , that is, the negative conjugate of the entire bottom half plane.

To our knowledge, this property has not been noted previously in the literature. However, it is consistent with the double spiral Sigma tiling (Goldberg shift equal to 1) seen in Grünbaum and Shephard (1986) [p. 514, Fig. 9.5.4 (b)].

Before leaving triangle substitution tiling, we point out that the tiles need not be polygons, nor must the base of the triangle be flat. Figure 5 shows two tiles that we developed based on continuous Cornu spiral segments. Notice that all sides of the tile are antisymmetric. This precludes the possibility of radial tiling with alternate coronas of conjugate tiles. Finally, let's not forget that all V-A tiles will tile periodically as well.





**Figure 5: Triangle substitution tiling with curved tiles (Cornu spirals).**

## A RECIPE FOR CREATING NEW TILES

In this section we'll discuss a simple method for creating new tiles. As an example, we'll develop the random icosihenagon shown above. First we'll present the general steps involved and then the specific example. Thus,

1. Select a vertex angle,  $\alpha$  that is *mod* 360 and  $2\pi/\alpha$  is even (for an even number of sectors)
2. Calculate the height and leg length of the 'parent' isosceles triangle, say  $h$  and  $s$
3. Create a random vertical line between  $z=0$  and  $z=i\cdot s/2$
4. Concatenate this with its negative or conjugate; re-order the vector so that it's bottom to top; set the bottom to  $z=0$  (negative is antisymmetric, conjugate is symmetric)
5. Create the full tile by rotating this line by  $\pm\alpha/2$ ; once again concatenate and order the vector counterclockwise; eliminate any duplicate points from the concatenations
6. Create the V- and A-tiles; typically,  $A=-V+1/2+i\cdot h$  or  $A=V^*+1/2+i\cdot h$  for antisymmetric and symmetric tiles, respectively (the asterisk denotes conjugate, as usual)

And here is the Matlab code for the random icosihenagon with a  $15^\circ$  vertex in an antisymmetric or symmetric form. Notice that here, as throughout the enclosed code, we use the Hebrew alphabet for  $\alpha$  and  $\beta$ . This is simply to avoid confusion with built-in Matlab functions of the same name, as can occur occasionally.

```
antisym=1;
alef=15*pi/180;
h=1/(2*tan(alef/2));
s=1/(2*sin(alef/2));
% let's make a 5-legged line,
y=linspace(0,s/2,6)';
x=s/3*(-.5+1*rand(4,1));
% get the correct endpoints
x=[0;x;0];
z1=complex(x,y);
if antisym
    z=[flipud(-z1);z1(2:end)];
else
    z=[flipud(conj(z1));z1(2:end)];    % for symmetric case
end
z=z-z(1);
% make the tile
z=[z*exp(-i*alef/2);flipud(z*exp(i*alef/2))];
V=z;
if antisym
    A=-V+1/2+i*h;
else
    A=conj(V)+1/2+i*h;    % for symmetric case
end
```

By the way, this code snippet is not bullet proof. You will encounter cases where the two legs of the tile cross each other. The culprit is in the definition of  $x$ ; this can be remedied with a more conservative (smaller) value, but at the cost of losing really good zig-zag tiles. Your choice.

## ABOUT THE PROGRAM

A complete Matlab program, `TriangleSubstitutionTiling4NCB.m` is available in the Appendix. The input and output variables are described here. All calculations are carried out in function `RadialSpiralTiling4NCB` by the radial sector method described above. The program plots the results of the specified tiling. It can be easily modified to alter or add to the built-in tiles.

The program is run from the command line as follows:

```
Z=TriangleSubstitutionTiling4NCB(tile,param,coronas,shift);
```

The output, `Z` is a complex matrix of tiles. The columns are the tiles and the rows are their complex coordinates. The input parameters are described in Table 3.

Parameter	Description	Remarks
<b>tile</b>	Built-in tiles: 'triangle', 'Voderberg' 'lightning' 'sigma', 'bentwedge' 'tent', 'random'	A string parameter; case insensitive  Refer to Table 2
<b>param</b>	This is the parameter for the deformable tiles and the vertex angle for the triangle and random tiles	All angles are input in degrees For fixed tiles use anything  Refer to Table 2
<b>coronas</b>	The number of annuli about the central core tiles	
<b>shift</b>	The number of tiles to shift in order to create spirals by Goldberg's method	This number may be positive or negative; normally $ \mathbf{shift}  \leq \mathbf{coronas}$  Refer to Table 2

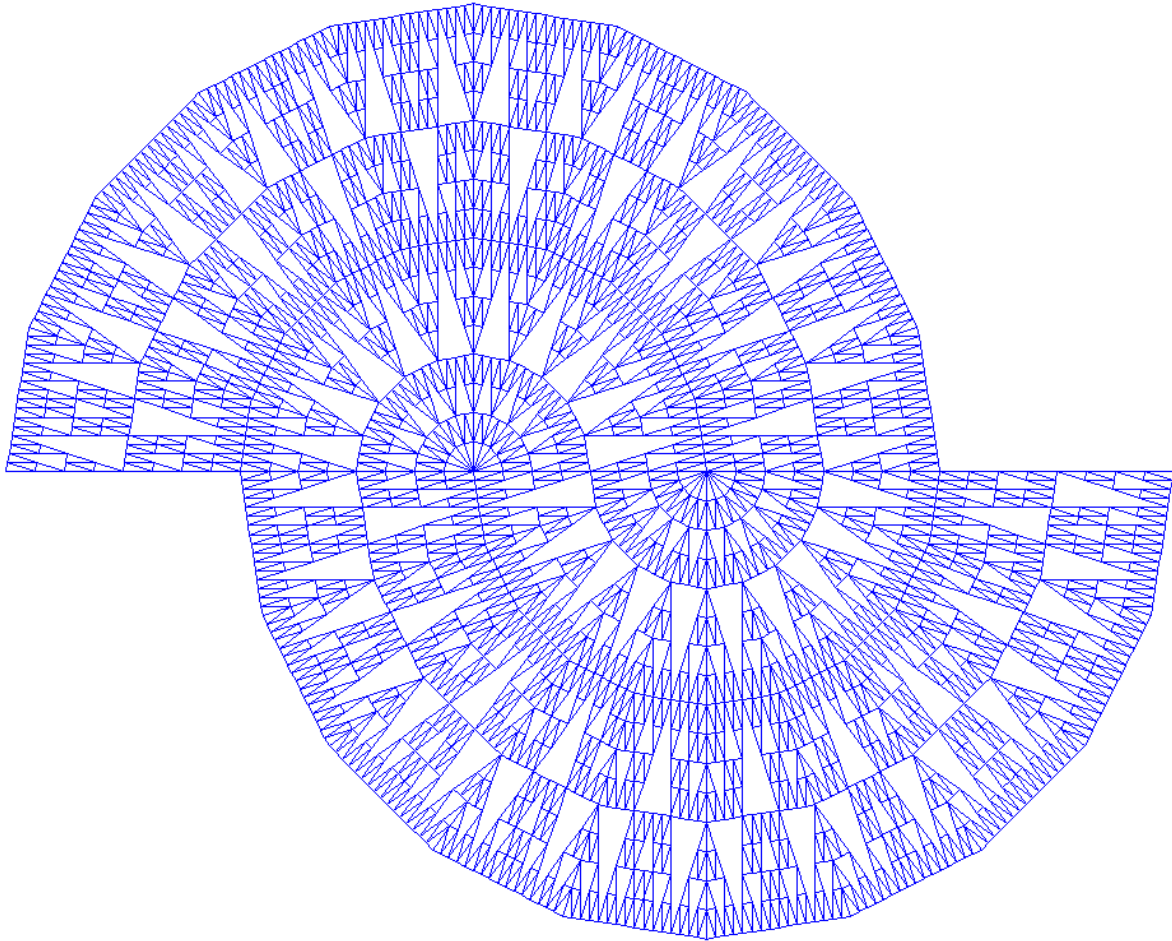
**Table 3: Program input parameters.**

For example, the following code will produce the classic Voderberg spiral in 0.15 seconds:

```
tile='voderberg';
param=132;
coronas=3;
shift=1;
Z=TriangleSubstitutionTiling4NCB(tile,param,coronas,shift);
```

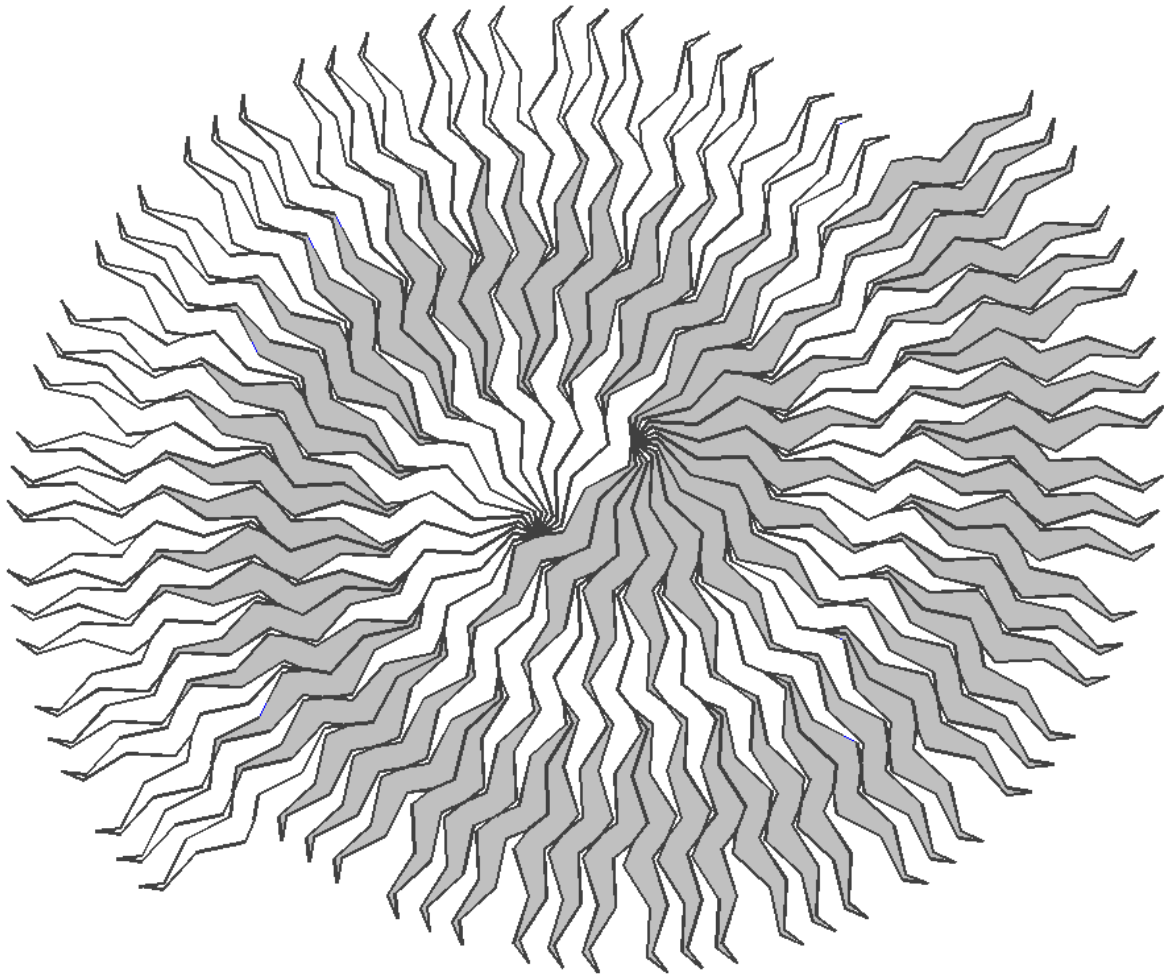
## MORE FUN WITH SPIRALS

There are many aspects to spiral tiling, so you have to wonder why it is that every time you go looking for them you find the same old thing. Here's another example of triangle substitution tiling. This this one is composed of triangles... Sierpinski triangles. Figure 6 shows a simple spiral. Actually, many fractals lend them themselves to various types of tiling.



**Figure 6: Sierpinski triangle applied to spiral.**

The most difficult of all the spirals are the *ad hoc* cases for which there are no guidelines. The fun, in spite of the frustration, is bringing order out of chaos. Such was the case in the development of the spiral shown in Figure 7. Here, we created a tile composed of two Sigma tiles, concatenated at the base in the antisymmetric configuration. However, this is not a triangle substitution tile. Can you see why? This original tiling was somewhat difficult, but it is a unique, deformable tile (see the associated animation), and it was worth the effort.



**Figure 7: The double-Sigma double spiral.**

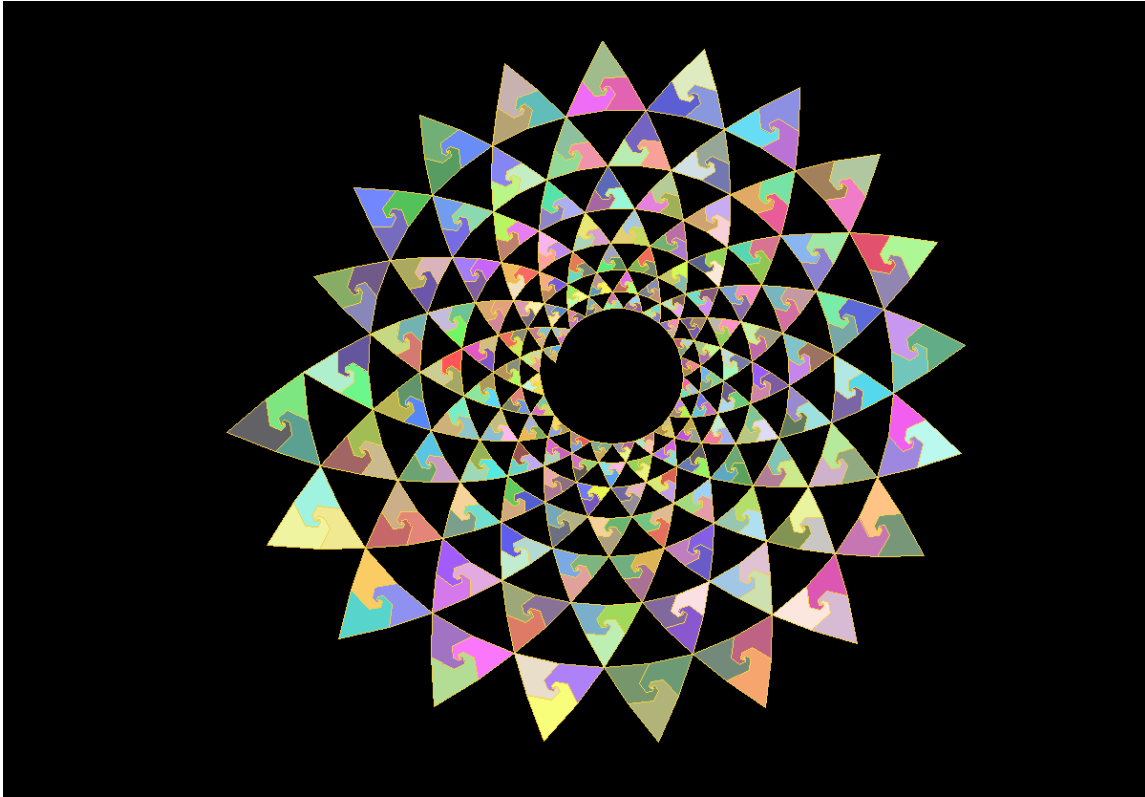
This double-Sigma will tile radially, and subsequent efforts led to something akin to the Goldberg shift. Ultimately, we were able to create spirals with half and full integer shifts. In the final analysis, it was a lot simpler than triangle substitution tiling. The same algorithms can reproduce the double-Lightning and Lord & Wilson tilings in Grünbaum and Shephard (1986), p. 516, and for spirals of any order. This may lead to a new tiling paradigm: rhombus substitution tiling. From this perspective, the double-Sigma tiling doesn't seem as special as originally thought. Order from chaos.

Then there are the faux spirals, optical illusions with a *trompe l'œil* effect. Figure 8 shows two examples. The top is our variation of the well-known "Fraser spiral" and that on the bottom is the double-Sigma tile shown above. In fact, these are purely logarithmic radial tilings.



Figure 8: *Trompe l'œil* spirals.

Now, any tile that can cover the plane periodically can be expressed as an aperiodic logarithmic radial tiling as in Figure 8. An almost identical algorithm allows the creation of logarithmic *spiral* tilings. Figure 9 shows an example using the twinkle tile; see Gazalé (1999) for that one. This is a great tile; it's a rep-tile whose gnomon is an equilateral triangle.



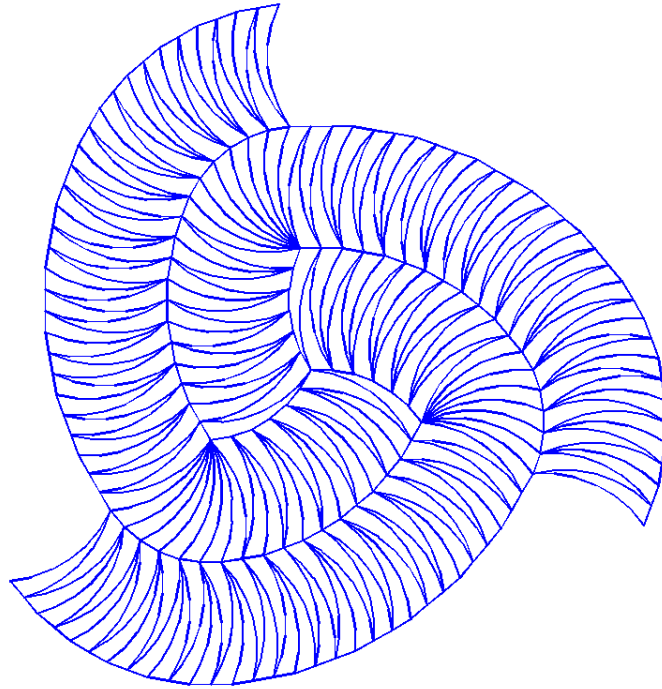
**Figure 9: A logarithmic spiral of twinkle triangles tiles.**

The Bent Wedge is also known to tile in one- and three-armed spirals [Grünbaum and Shephard (1986), pp. 22 and 514]. But it is also a member of a class of *versatiles* [see, for example, Grünbaum and Shephard (1979) and Gailiunas (2000)]. Any of these tiles can be applied to triangle substitution tiling. Our observation is that these tiles all behave identically in tiling; the same program can tile any of them with specification of a single parameter. Figure 10 shows a three-armed spiral of a tridecagon versatile.

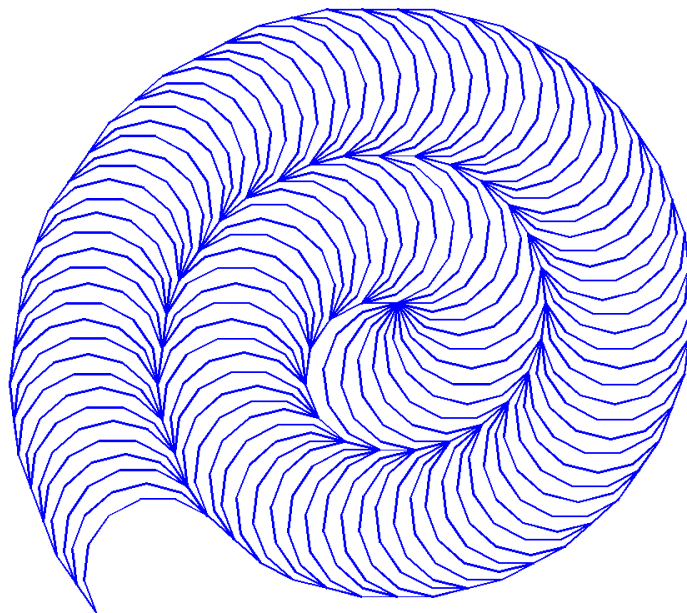
The Tent is also known to tile in a one-armed spiral [Grünbaum and Shephard (1986), p. 515]. Similarly, the Tent tile is a member of a set of reflexed (odd) polygons that all tile similarly. Figure 11 shows a one-armed spiral for a reflexed tridecagon.

The point to be made about these families of tiles is that the code that produced these figures could produce similar ones for any and all of the set members, and to any number of turns. Thus, the algorithms could truly cover the plane if you had the computer resources to do it.





**Figure 10: Three-armed spiral of a tridecagon versatile.**



**Figure 11: One-armed spiral of a reflexed tridecagon.**



In 2000, Paul Gailiunas developed a new system of spirals based on regular polygons. These spirals are noteworthy on several counts. In the first place, they all spiral out from a common center point; second, the spirals zig-zag as they wind out; and third, perhaps most significant, is that you can create a spiral of any number of arms  $\geq 3$ . To put this in perspective, you may recall from Grünbaum and Shephard (1986), that "...at present, we know of no prototile which admits a monohedral  $r$ -armed spiral tiling for any *odd* value of  $r \geq 5$ ." With Paul's tiles, this is now possible. We observed from Paul's figures that all these radial tilings obeyed exactly the same set of rules and we constructed a simple program that can create any or all of them to any number of coronas. Figure 12 shows a few examples. (These are Paul's spirals; we just programmed here.)

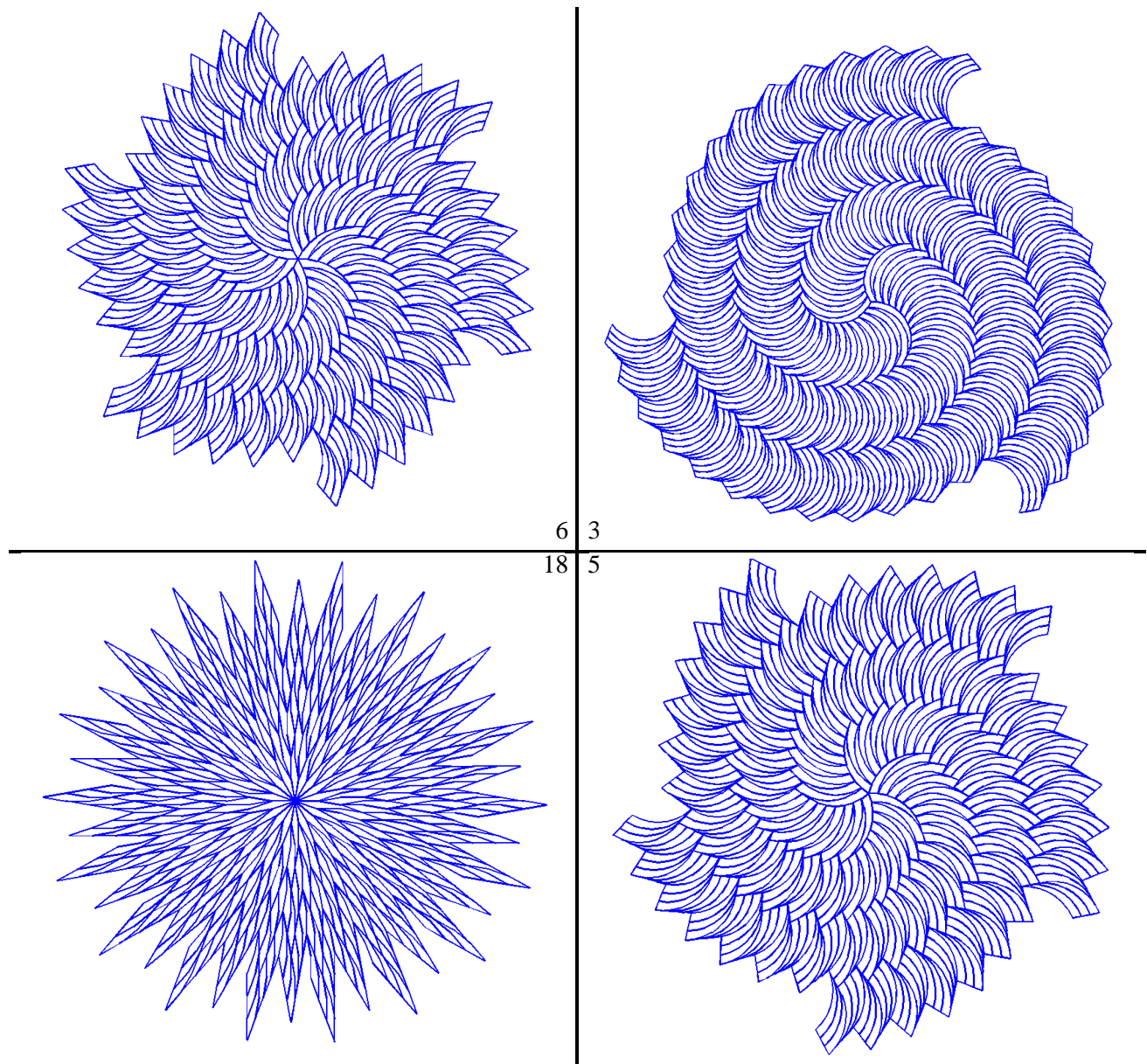


Figure 12: Some example of spirals with Paul Gailiunas's tiles.

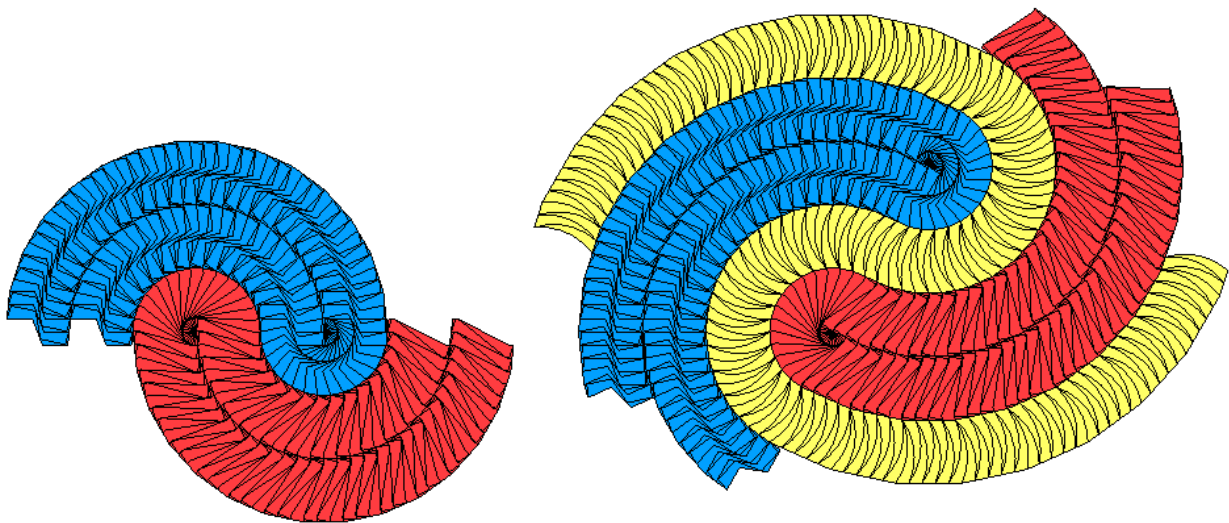
## Summary

We have developed the equations for a generalized Voderberg tile and shown that this deformable tile satisfies all the requirements of Voderberg's original tile. We demonstrated how to describe tiles in the complex plane, develop new tiles for radial and spiral tiling, and build such tilings with triangle substitution tiling and the Goldberg shift. A complete program for radial and spiral tiling with the Voderberg and several other tiles is included. We concluded with a pastiche of other spiral in order to encourage further efforts in the field.

Earlier we mentioned the idea of bringing order out of chaos in tiling. In the present effort we have sought to develop a model for any triangle substitution tiling. In all of our work, we seek the mathematical framework to simplify the process to where we can think beyond the mechanics to the creative. Along the way, we come to better understand the underlying principles and the ability to handle the *ad hoc* cases with aplomb. Sometimes.

Understanding V-A tiling has shown us how these tiles behave similarly; it is what allowed us to easily create the faux Voderberg triple spiral seen in the animations.

It also suggests that we can mix-and-match tiles in spiral tiling or in radial coronas. Figure 13 shows some possibilities to explore. Each colored zone is composed of different tiles of the same vertex angle, in this case, two Voderberg tiles and a versatile hendecagon. The options increase with the number of shifts. This works absolutely because the different tiles only touch at the base.



**Figure 13: Mix-and-match tiling possibilities.**

It seems to us that there are still a lot of unexplored possibilities for spiral tilings.

## Appendix: Triangle Substitution Tiling Program

```

function Z=TriangleSubstitutionTiling4NCB(tile,param,coronas,shift)
% this program was created for the National Curve Bank to accompany the
% submission "Voderberg Deconstructed and Triangle Substitution Tiling,"
% see http://curvebank.calstatela.edu/waldman9/waldman9.htm for details
%-----
% HERE IS A SHORT PROGRAM FOR RANDOM TESTING
% function TestTriangleSubstitutionTiling4NCB
% % title says it all
% close all
% mod360=[12 15 18 30 36 45 60]';
%
tiles={'triangle','Voderberg','LiGhTnInG','SIGMA','BentWedge','TEnt','rAnD0m'};
%
% for k=1:length(tiles)
%     params=[mod360(floor(1+7*rand)),111+(153-111)*rand,1.75*rand,...
%             0.72+(2.65-0.72)*rand,1,1,mod360(floor(1+3*rand))];
%     coronas=floor(1+6*rand);
%     shift=-3+floor(6*rand);
%     TriangleSubstitutionTiling4NCB(tiles{k},params(k),coronas,shift);
%     title([tiles{k} ': (P,C,S) = (' num2str(params(k)) ','...
%           num2str(coronas) ',' num2str(shift) ')'])
% end
% return
%-----
% Copyright 2014, Cy H. Waldman
% Let me know how this works out for you at cye@att.net

tile=lower(tile);
switch tile
    case 'triangle'
        alef=param*pi/180;
        x=1;
        h=1/(2*sin(alef/2));
        S=[x;h;h];
        phi=[0;pi/2+alef/2;3*pi/2-alef/2];
        z=cumsum(S.*exp(i*phi));
        z=[0;z];
        V=conj(z)-1/2+i*max(abs(imag(z)));
        A=z;
        sectors=2*pi/alef;
        if mod(sectors,2)>0
            error('Need an even integer number of sectors for spirals')
        end
        antisym=1;

    case 'voderberg'
        alef=12*pi/180;
        b=param;
        beth=b*pi/180;
        L=2*sin((pi-alef)/2)/cos(beth-pi/2);
        x=(csc(alef/2)/2-L*cos(pi-beth))/2-sin(alef/2);
        S=[1;x;L;x;1;x;L;x;1];
        theta=[alef;(3*pi-alef)/2;(2*pi-beth);beth;(pi+alef)/2;
              (pi-3*alef)/2;(2*pi-beth);beth;(pi+alef)/2];

```

```

phi=cumsum(pi-theta);
z1=cumsum(S.*exp(i*phi));
z1=[0;z1];
z3=-z1+1/2+i*max(imag(z1));
V=z1;
A=z3;
sectors=2*pi/alef;
antisym=1;

case 'lightning'
alef=pi/12;
sectors=2*pi/alef;
% h=1/(2*tan(alef/2));
s=1/(2*sin(alef/2));
l=param;
theta=pi/3;
z=l*exp(i*theta);
z=[0;z];
z=[z;z-z(2)+i*s];
z=[z;flipud(z*exp(i*alef))];
V=z*exp(-i*alef/2);
A=(-z+i*s)*exp(-i*alef/2);
antisym=1;

case 'sigma'
alef=pi/12;
sectors=2*pi/alef;
s=1/(2*sin(alef/2));
lratio=param;
f=@(x) cos(x)./cos(pi/2+x)+lratio;
theta=fzero(f,.5);
phi=acos(-cos(theta)/2);
l1=s/2*sin(phi-pi/2)/sin(pi-phi+theta);
l2=2*l1;
z=[0;l1*exp(i*theta);l1*exp(i*theta)+l2*exp(i*phi)];
z=[z;flipud(conj(z(1:end-1))+i*s)];
z=z*exp(-i*alef/2);
z=[z;flipud(z*exp(i*alef))];
V=z;
A=conj(z)+i*s*exp(-i*alef/2);
V=-conj(V);
A=-conj(A)+1;
antisym=0;

case 'bentwedge'
alef=pi/12;
theta=[7;11;11;11;1;13;13;13;4]*alef;
phi=cumsum(pi-theta); % 12 is 180 deg when d=12
z1=cumsum(exp(i*phi));
z1=[0;z1];
z0=z1(5);
z1=z1-z0;
V=conj(z1);
A=z1+1/2+i*max(abs(imag(z1)));
sectors=24;
antisym=0;

```

```

case 'tent'
    d=14;
    alef=2*pi/d;
    theta=[7;5;5;5;1;9;9]*alef;
    phi=cumsum(pi-theta);
    z=cumsum(exp(i*phi));
    z=[0;z];
    V=conj(z)-1/2+i*max(abs(imag(z)));
    A=z;
    sectors=14;
    antisym=0;

case 'random'
    antisym=1;
    a=param;
    alef=a*pi/180;
    h=1/(2*tan(alef/2));
    s=1/(2*sin(alef/2));
    % let's make a 5-legged line,
    y=linspace(0,s/2,6)';
    x=s/3*(-.5+1*rand(4,1));
    % get the correct endpoints
    x=[0;x;0];
    z1=complex(x,y);
    if antisym
        z=[flipud(-z1);z1(2:end)];
    else
        z=[flipud(conj(z1));z1(2:end)]; % for symmetric case
    end
    z=z-z(1);
    % make the tile
    z=[z*exp(-i*alef/2);flipud(z*exp(i*alef/2))];
    V=z;
    if antisym
        A=-V+1/2+i*h;
    else
        A=conj(V)+1/2+i*h; % for symmetric case
    end
    sectors=2*pi/alef;
    if mod(sectors,2)>0
        error('Need an even integer number of sectors for spirals')
    end
otherwise
    error('Please select from the available tiles')
end

Z=RadialSpiralTiling(V,A,sectors,coronas,shift,antisym);
figure;plot(Z,'b');axis equal;axis off;set(gcf,'Color','w')
v=axis;axis(1.01*v)

return

```

```

function Z=RadialSpiralTiling(V,A,sectors,coronas,shift,antisym)
% this function produces a normal radial tiling of unit equilateral
% triangle V, with origin at the center, in sectors and coronas
% Copyright 2014, Cye H. Waldman

% setup
alef=2*pi/sectors;
fin=V;
duo=[V A];           % the V-A pair
dely=1/(2*tan(alef/2)); % corona height

z=V;
for r=1:coronas
    this=[];
    % calculate the rows
    for k=1:r
        this=[this duo-r/2+k-1+i*r*dely];
    end
    last=fin+r/2+i*r*dely;
    next=[this last];
    if ~antisym && mod(r,2)==1; next=-conj(next); end
    z=[z next];
end
% rotate to lie on negative x-axis
z=z*exp(i*pi/2)*exp(-i*pi/sectors);

% if shift<=coronas
Z=z;
for k=1:sectors-1
    Z=[Z z*exp(-i*k*alef)];
end
% and finally the spiral if shift~=0
if abs(shift)>0
    Ztop=Z(:,1:end/2);
    Zbot=Z(:,end/2+1:end);
    if ~antisym && mod(shift,2)>0; Zbot=-conj(Zbot); end
    delx=1/(2*sin(alef/2));
    Z=[Ztop Zbot+shift*delx];
end

return

```

Please notify us at [cye@att.net](mailto:cye@att.net) if there are any problems. And, by all means, let us know of any new tiles or tilings you develop for the program.

**References**

Gardner, M. (2001). *The Colossal Book of Mathematics*, Norton. (The chapter including triangle substitution tiling originally appeared in a 1977 *Scientific American* column.)

Gailiunas, P. (2000). "Spiral Tilings" *Bridges 2000 Conference Proceedings*, pp.133-140. Also available at <http://archive.bridgesmathart.org/2000/bridges2000-133.pdf>.

Gazalé, M.J. (1999). *Gnomon: From Pharaohs to Fractals*, Princeton. I was very excited to discover this book and have learned a lot from it; highly recommended.

Goldberg, M. (1955). "Central Tessellations," *Scripta Mathematica*, **21**, pp 253-260.

Grünbaum, B. (2014) . Private communication.

Grünbaum, B. and Shephard G.C. (1979) "Spiral Tilings and Versatiles," *Mathematics Teaching*, No.88, Sept.1979, pp.50-1.

Grünbaum, B. and Shephard, G.C. (1986). *Tilings and Patterns*, Freeman. This book is a must-have for anyone who is even moderately interested in tiling. At almost 20 years old, it is still the bible of tiling. Get the complete hardback edition, not the paperback 'Introduction' edition.

Dutch, S. (1999). "Some Special Radial and Spiral Tilings," <http://www.uwgb.edu/dutchs/symmetry/radspir1.htm>. Dutch's extensive Web site has many interesting radial, spiral, and logarithmic spiral tilings; explore the many links therein.

Meisser, K. (2009). <https://app.box.com/shared/5p80jtlmaf>.

Roskes , B. and Choate, J. (2009). *Advanced Spiral Tiling: Hirschhorn, Bent Wedge, and Voderberg Tiles in Google SketchUp 7*, (GeomeTricks Aperiodic Patterns, Book 2), 3DVinci. See the complete set of GeomeTricks printed and e-books at <http://www.3dvinci.net/ccp0-catshow/GM.html>.

These books are highly recommended, even if you don't use the SketchUp program, for the wealth of ideas on and examples of tiling within them. Nicely organized and profusely illustrated with step-by-step instructions.

Trott, M. (2005). *The Mathematica GuideBook for Numerics*, Springer.

Wikipedia, "Tessellation," <http://en.wikipedia.org/wiki/Tessellation>.