

Parquet Tiling
Cye H. Waldman
cye@att.net
Copyright 2015

Abstract

Parquet, or parallelogram substitution, tiling is a new paradigm for radial and spiral tiling. It is inspired by, but decidedly different than rhombus substitution tiling. We describe the tiles that are suitable for this tiling and how to create them. We demonstrate how to produce radial and spiral tilings with the Goldberg shift.

Introduction

Most radial tilings are based on isosceles triangles and triangle substitution tiling [see, for example Grünbaum and Shephard (1986), Dutch (1999), and Waldman (2014)]. What these tilings have in common is that the prototiles are odd-numbered n -gons with two congruent legs that join at one end, thus the ‘triangle’ substitution. Waldman (2014) has recently formalized the procedure and made a Matlab code freely available.

We have extended this concept in Waldman (2015a) to rhombus substitution tiling. ‘Rhombus’ tiles consist of two ‘triangle’ tiles concatenated at the base. These tiles have the property that when aligned horizontally they touch only at the waist. Thus, when tiled periodically, rows and columns are both staggered.

Now we wish to consider tiles that are fundamentally different. Specifically, tiles that are concatenated even n -gons. These tiles will stack both horizontally and vertically, rather than stagger. The question is whether they can be persuaded to tile radially and spirally.

The tiles created by Gailiunas (2000) are the parent tiles for the concatenation. We review their properties and show that parqueting allows us to create a radial sector that can be rotated to fulfill a radial tiling. From there we can easily coax the spirals out with the Goldberg (1955) shift.

A gallery of images demonstrates the broad class of tilings that accrue.

Finally, we present a complete Matlab program for Parquet tiling. This program can be readily modified by an enterprising user who wishes to explore further.

There are two animations associated with this paper [Waldman (2015b), this paper]. The first shows the buildup of a sector, then radial and spiral tilings. The second shows how a single parent tile can lead to three distinct tilings.

Technical Discussion

GAILIUNAS AND PARQUETING TILES

The present work is predicated on the Gailiunas tiles. It is highly recommended to read the paper by Gailiunas (2000), which is freely available online. Nevertheless, a brief review is given here.

These tiles arise from overlapping n -gons as shown in Figure 1. Tiles start at the lower left and are created by cutting along any of the lines indicated. We can describe these tiles simply as a function of n and K , where n is the order of the regular polygon and K is one-half the order of the tile, that is, half the number of line segments.

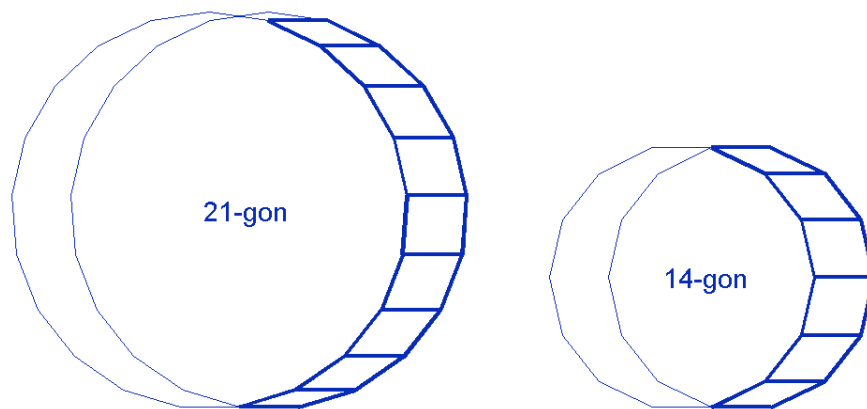


Figure 1: Creating Gailiunas tiles from overlapping regular n -gons.

Figure 2 shows the complete set of Gailiunas tiles for $n=23$. We chose an odd n -gon to underscore the fact that the maximum value of $K = \lfloor n/2 \rfloor$ (floor). The vertex and base angles are designated as α and β , respectively, as shown in the figure. These angles are given by

$$\begin{aligned}\alpha &= \frac{2\pi}{n} = \pi - \theta \\ \beta &= (K-1)\alpha\end{aligned}\tag{1}$$

where θ is the internal angle of the regular n -gon.

Gailiunas phenomenal vertex spiral tiling, which will spiral with any number of arms, is based on the angle β as the vertex. As such, β must be *mod* 360, or alternatively $n/(K-1)$ must be an integer. So, for example, for $n=6$ to 36, there are 52 possible spirals of various arms.

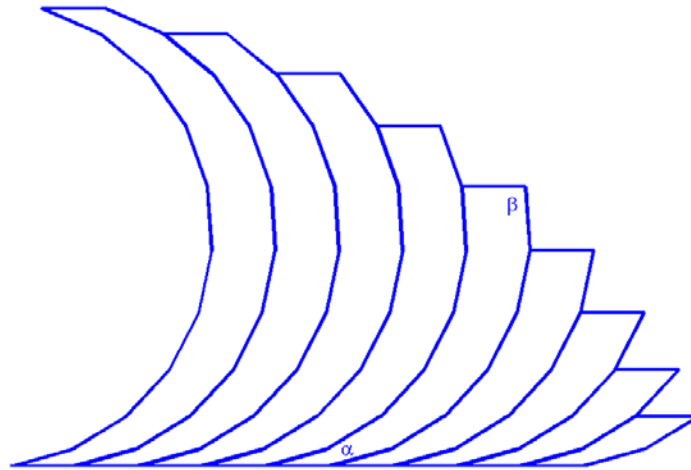


Figure 2: Example Gailiunas tiles for $n = 23$, $K = 3:11$ (right to left).

The present work, however, is based on α as the vertex angle. All these tiles will tile radially because all α are *mod* 360. There are 256 possible tilings between $n = 6$ to 36, of which 136 are of even n that can be spiraled with the Goldberg shift. But we get ahead of ourselves.

The tiles used for Parquet tiling are Gailiunas tiles that are concatenated at the top in an antisymmetric configuration, e.g., β joins with $\pi - \beta$. Figure 3 shows some examples based on the same tiles as in Figure 2. And just like the Gailiunas tiles, these are completely specified by the two parameters n and K . The only difference is that there are now $4K-2$ equilateral sides to these polygons.

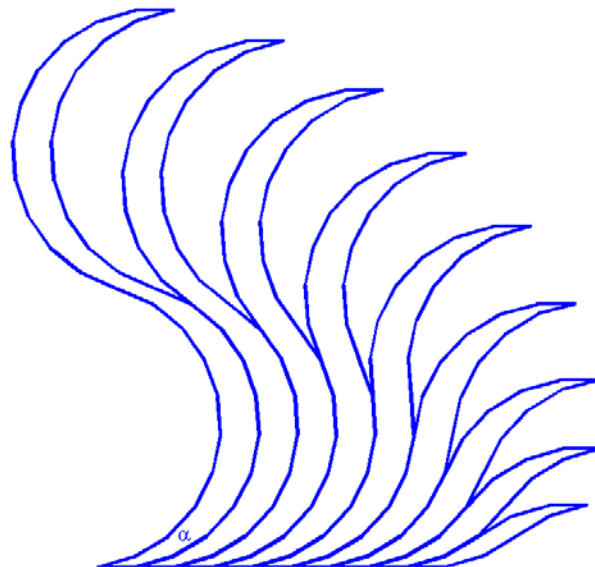


Figure 3: Parquetting tiles for $n = 23$, $K = 3:11$ (right to left).

PARQUET TILING

Figure 4 demonstrates the concept of parquet tiling. Tiles are laid out with each ‘corona’ having one more tile than the last and at right angles to it. The tiles are numbered to show the ordering and colored to indicate the different coronas. The process can continue indefinitely. Believe it or not, we now have a radial sector that can be rotated to fill a radial tiling. Now, this tile is described by $(n, K) = (4, 2)$ and is not very interesting, but it makes the point. Actually, tiles with $K < 3$ are degenerate and not of interest. In practice, we restrict ourselves to $K \geq 3$ and $n \geq 6$.

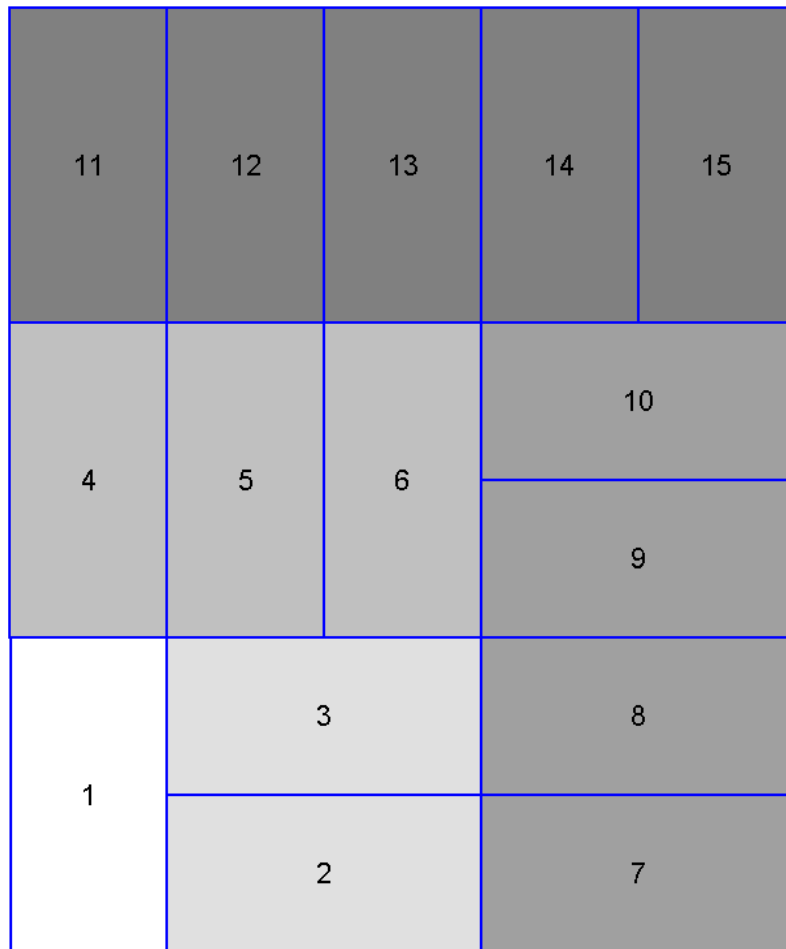


Figure 4: Parquet tiling layout; shades of grey indicate coronas.

Although Figure 4 clearly shows the parqueting property, it can become more difficult to discern as n and K increase. Table 1 shows some additional sectors for other values of n and K . Bear in mind that $K \leq \lfloor n/2 \rfloor$, hence the vacant cells. In addition to successive coronas being oblique to each other, they are composed of conjugate tiles.

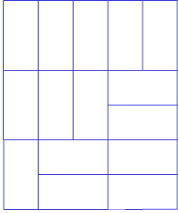
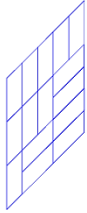




Tile $n \rightarrow$ $K \downarrow$	4	8	16
2			
4			
8			

Table 1: Some examples of Parquet tiling sectors as a function of n and K .

It almost goes without saying that these sectors will tessellate upon rotation for any values of n and $K \leq \lfloor n/2 \rfloor$. Moreover, they will spiral for all even n , in accordance with the Goldberg shift. In this paper we take the shift, Δx to be the distance between peaks and troughs of the sector edges seen in the table. The spirals are then created quite simply as follows:

$$Z_{spiral} = [Z_{top} \quad Z_{bot} + g\Delta x] \tag{2}$$

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.

This is amazingly simple, but there is one caveat that must be noted. When the absolute value of the shift, g is an odd number, then we must replace Z_{bot} by $-Z_{bot}^*$, i.e., the negative conjugate.

A Matlab program is provided for calculating these tilings; it is described in the next section.

ABOUT THE PROGRAM

A complete Matlab program, `ParquetTiling4NCB.m` is available in Appendix A. The input and output variables are described here. The program optionally plots the results of the specified tiling. It can be easily modified at the user's whim.

The program is run from the command line as follows:

```
W=ParquetTiling4NCB(n,K,coronas,shift,dual,trim,plotting);
```

The output, `W` 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 2.

Parameter	Description	Remarks
n	The order of the regular n -gon that is the grandparent of the parquet tile	$n \geq 6$
K	The second tile parameter; it is $\frac{1}{4}$ the number of sides of the tile	$3 \leq K \leq \lfloor n/2 \rfloor$
coronas	The number of annuli about the central core tiles	This should be ≥ 1 .
shift	The number of tiles to shift in order to create spirals by Goldberg's method	Positive or negative integers; normally $ \mathbf{shift} \leq \mathbf{coronas}$
dual	Indicates whether to create Parquet (= 1) or Gailiunas (<i>not</i> 1) tiling	See below
trim	Indicates whether to trim (=1) or not (<i>not</i> 1) to trim the fins (spikes)	This is ignored when dual=0 . See below
plotting	Indicates whether to plot (=1) or not (<i>not</i> 1) to plot the results	

Table 2: Program input parameters.

Figure 5 (a) shows a typical spiral, in this case, a Parquet tile with $(n, K) = (14, 6)$, three coronas, and a shift of minus one. Now, an interesting thing about the program is that it doesn't care if the Parquet tile is replaced by its parent tiles. In this case, you recover the same tiling, but with all the tiles split in two, as seen in Figure 5 (b). The striking fins on the periphery are the residual tiles from the 'parquet.' They can be optionally omitted in the program to fully reproduce a more typical looking spiral of Gailiunas tiles, as seen in Figure 5 (c). These tilings are not same as those in Gailiunas (2000). So, in a word, a single pair (n, K) can produce three distinctive tilings.

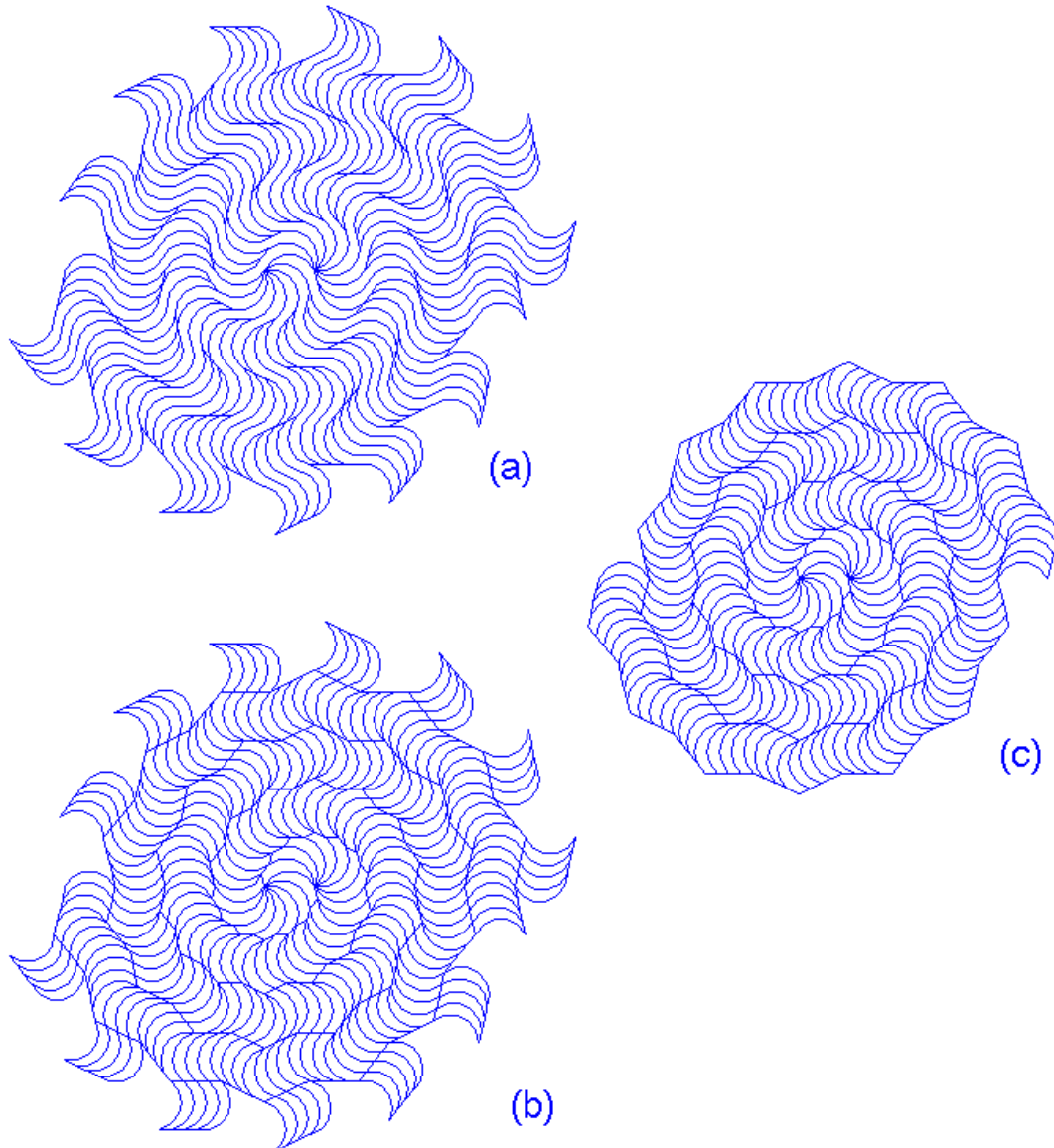


Figure 5: Examples of Parquet tiling program output: (a) 'parquet' configuration; (b) 'Gailiunas' configuration (includes residual spikes); (c) 'Gailiunas' configuration with optional removal of residual tiles).

The following code will produce Figure 5a in 0.08 seconds:

```
n=14;K=6;coronas=3;shift=-1;dual=0;trim=0;plotting=1;
W=ParquetTilings4NCB(n,K,coronas,shift,dual,trim,plotting);
```

The program has no limits on the order of the regular polygon, n or the number of coronas. As a practical matter, the plots will become too dense to see anything beyond a certain limit. That's not to say that you couldn't zoom in and find some interesting things, such as that seen in Figure 6.

$(n,K,c,s,d,t) = [120,60,3,-1,0,0]$

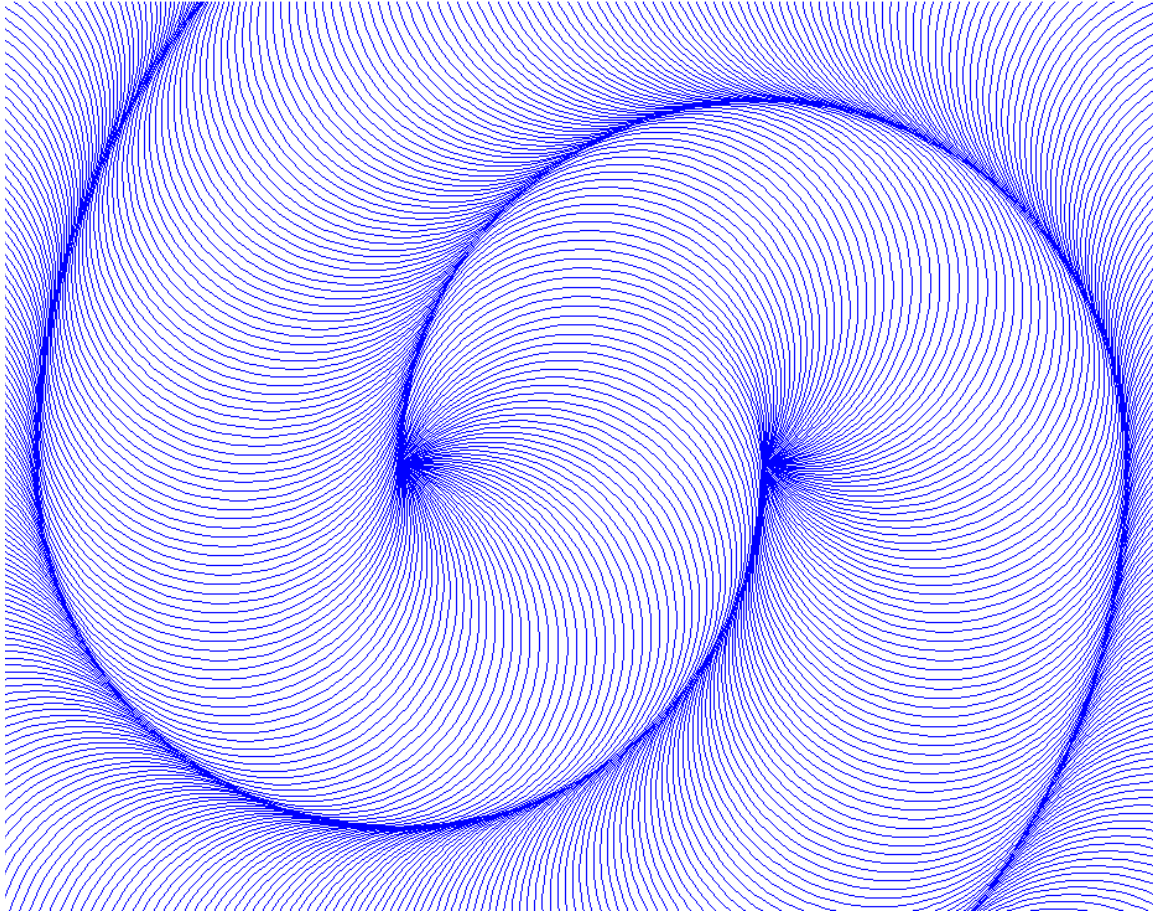


Figure 6: Example of extreme tiling: $n = 120$, $K = 60$.

The quest for additional tiles that lend themselves to Parquet tiling will be left as an exercise for the reader. We have provided the tools and the rules.

A gallery of randomly created images is shown below. The program for generating these, as well as testing the program, is embedded in the header section of the main program. The title in each figure gives the input parameters: $n, K, coronas, shift, dual, trim$. The gallery has been sorted into increasing values of n and K to demonstrate the transition from simplicity to complexity.

BONUS: GAILIUNAS TILING

Gailiunas tiling has been mentioned several times herein. We are providing the code for generating these, if for no other reason than to demonstrate that Parquet tiling is different and unique. The tiles are, of course, described by the same two parameters, n and K , albeit with the restriction that $n/(K-1)$ is an integer. In fact, this is the number of arms, N of the spiral.

We can reliably create spirals of N arms with tiles of n and K defined as follows:

$$\begin{aligned} n &= N(k-1) & k &= 1, 2, 3, \dots \\ K &= k + 2 \end{aligned} \tag{3}$$

Figure 7 shown an example tiling with 5 arms and $(n, K) = (35, 8)$. This figure demonstrates the distinguishing feature of Gailiunas tiling. They all emanate from the center. In the algorithm, we develop a single arm and rotate it through the appropriate number of arms.

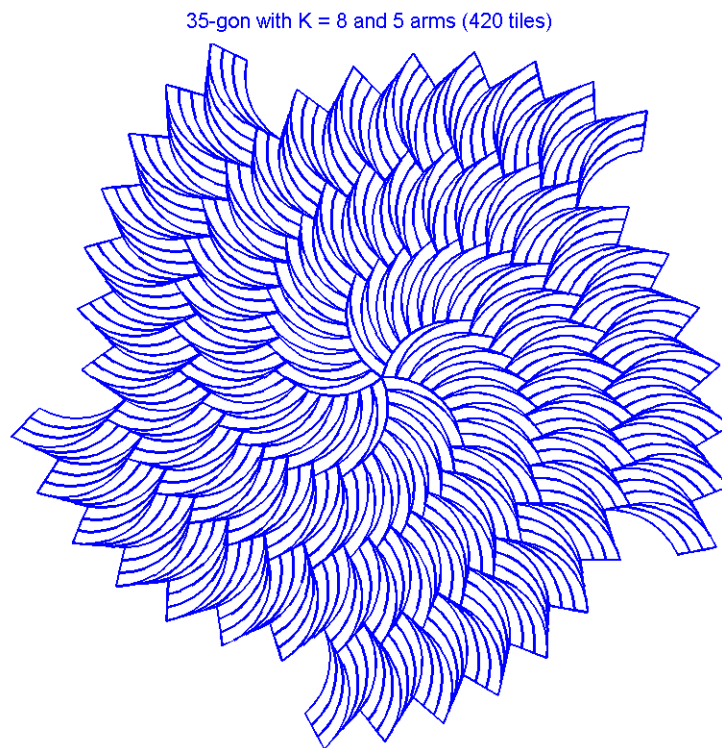


Figure 7: Example Gailiunas tiling with 5 arms: $n = 35$, $K = 8$.

The program `GailiunasTiling4NCB.m` found in Appendix B will produce the full set 52 tilings for $n = (6:36)$ — in less than 15 seconds. Actually, the calculation takes less than $1/10^{\text{th}}$ of a second; it's the figures that consume the bulk of the time. Of course, the user may change the maximum value of n , as well as the `level` (or the radius of the spiral). Both parameters are found near the top of the program. The default level, as in Figure 7, is `level=3`.

Summary

We developed a new paradigm for radial and spiral tilings because the prototiles we were interested in did not fit the old paradigms. Previous work was concerned with rhombus substitution tiles, whose radial sections consisted of staggered row of parallel tiles. The prototiles for the present effort are decidedly different. First, they are not rhombus substitutions, but more like parallelogram substitutions. Second, sectors are built up from successive coronas that are oblique to each other. Thus the new paradigm was dubbed ‘Parquet’ tiling. The prototiles are concatenated Gailiunas tiles and tessellate perfectly in this configuration. Basically, the entire tiling algorithm depends on the two parameters that fully define the tile itself. None of the tiles requires any special attention within the program. The remaining parameters are essentially just window dressing: How many coronas do want? Spirals? How many? Etc.

An interesting feature of the program is that you can substitute a pair of the parent Gailiunas tiles for the Parquet tile and get two optional tilings. Thus we may have inadvertently expanded the scope of Gailiunas tiling.

It was about two weeks from when we tried rhombic tiling of the Gailiunas tile, to miserable failure, and finally to where we are today.

References

Dutch, S. (1999). “Radial and Spiral Tilings,”

<https://www.uwgb.edu/dutchs/symmetry/rad-spir.htm> .

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

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

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.

Waldman, C.H. (2014). “Voderberg Deconstructed & Triangle Substitution Tiling,”

<http://curvebank.calstatela.edu/waldman9/waldman9.htm>.

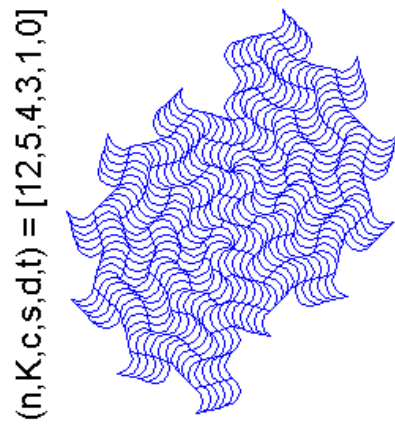
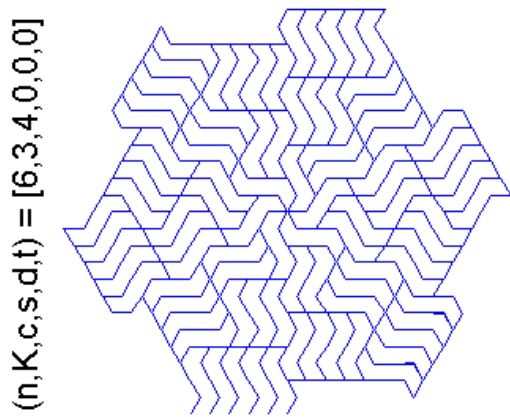
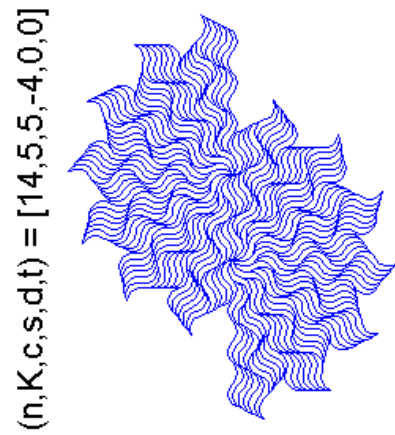
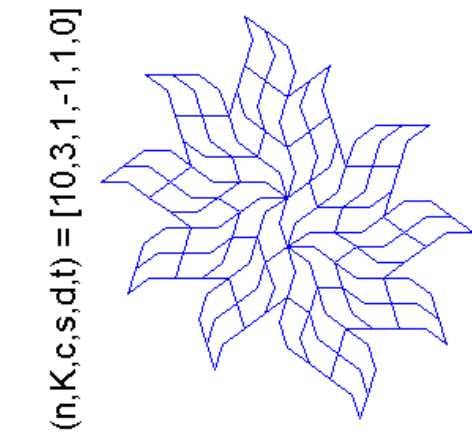
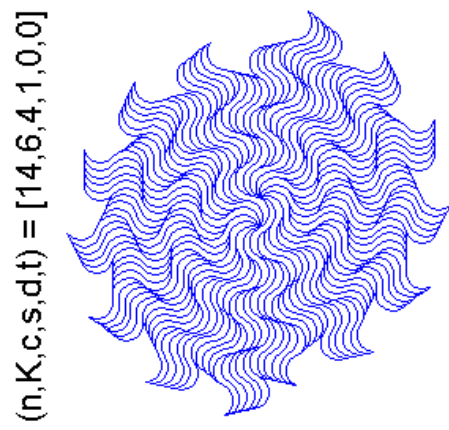
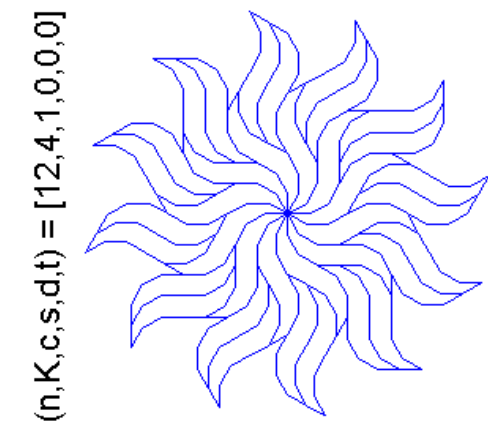
Waldman, C.H. (2015a). “Rhombus Substitution Tiling,”

<http://curvebank.calstatela.edu/waldman14/waldman14.htm>.

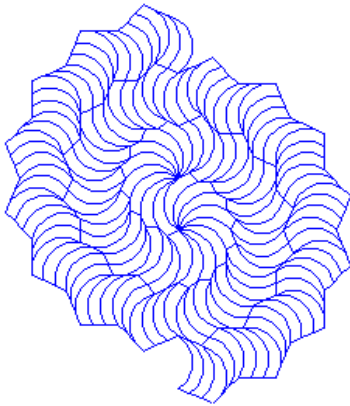
Waldman, C.H. (2015b). “Parquet Tiling,”

<http://curvebank.calstatela.edu/waldman15/waldman15.htm>.

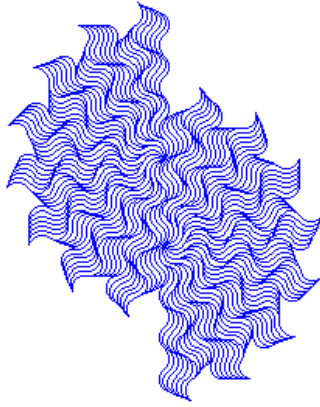
Gallery of random Parquet tilings



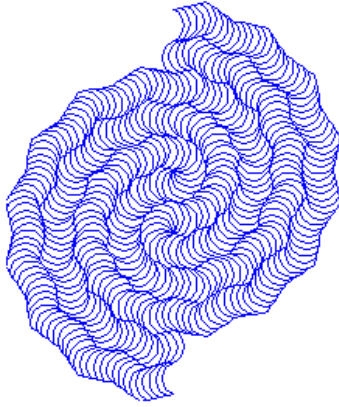
$(n,K,c,s,d,t) = [16,6,2,-1,1,1]$



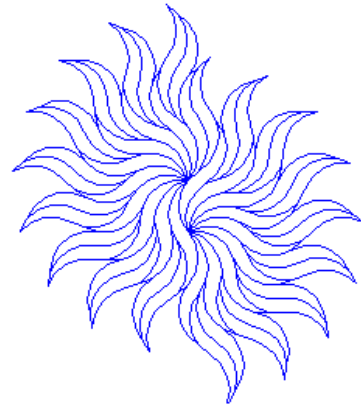
$(n,K,c,s,d,t) = [16,6,5,-4,0,0]$



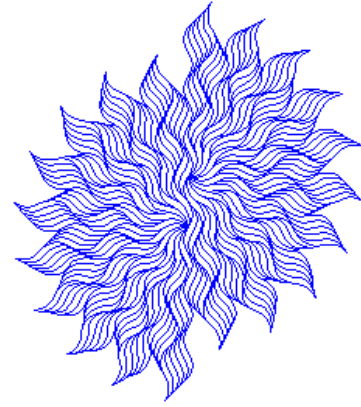
$(n,K,c,s,d,t) = [16,7,4,2,1,1]$



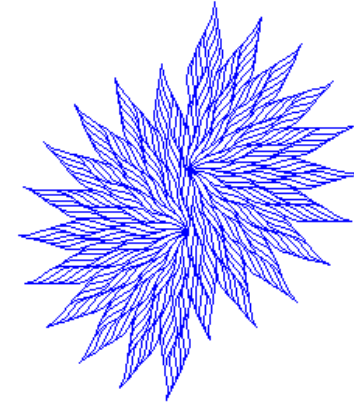
$(n,K,c,s,d,t) = [18,5,1,-1,0,0]$



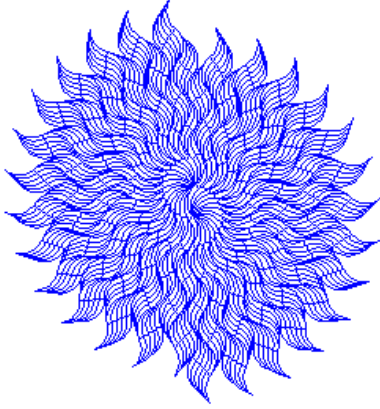
$(n,K,c,s,d,t) = [20,5,4,2,0,0]$



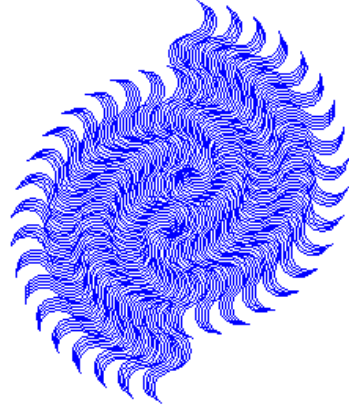
$(n,K,c,s,d,t) = [22,3,4,3,0,0]$



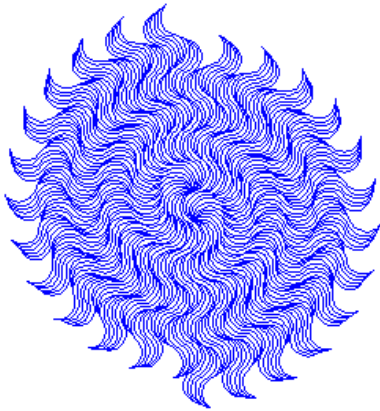
$(n, K, c, s, d, t) = [28, 7, 4, -1, 1, 0]$



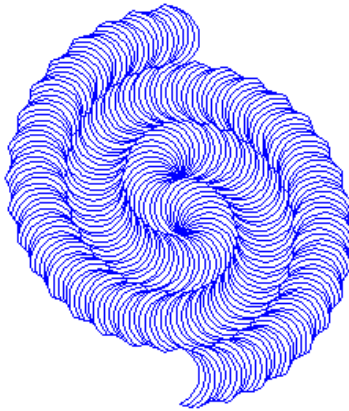
$(n, K, c, s, d, t) = [36, 14, 3, 2, 0, 0]$



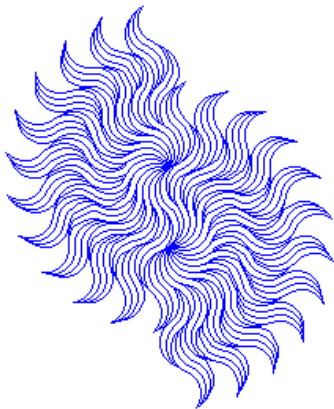
$(n, K, c, s, d, t) = [26, 10, 4, -1, 0, 0]$



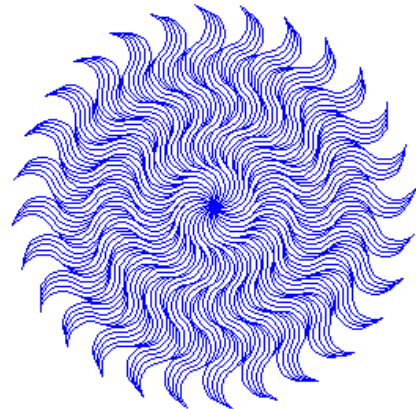
$(n, K, c, s, d, t) = [34, 14, 2, -1, 1, 1]$



$(n, K, c, s, d, t) = [24, 8, 2, -2, 0, 0]$



$(n, K, c, s, d, t) = [28, 10, 3, 0, 0, 0]$



Appendix A: Parquet Tiling Program

```

function W=ParquetTilings4NCB(n,K,coronas,shift,dual,trim,plotting)
% this program was created for the National Curve Bank to accompany the
% submission "New Gailiunas Tiling,"
% see http://curvebank.calstatela.edu/waldman12/waldman12.htm for details
%-----
% HERE IS A SHORT PROGRAM FOR RANDOM TESTING
% % function TestParquetTilings4NCB
% close all
%
%
% kmax=6;
% nmax=4;
% for n=1:nmax
%     figure
%     for k=1:kmax
%         % n=floor(6+31*rand);
%         n=2*floor(3+16*rand);
%         K=3+floor((n/2-2)*rand);
%         coronas=1+floor(5*rand);
%         shift=-coronas+floor(2*coronas*rand);
%         dual=round(rand);
%         trim=round(rand);
%         plotting=0;
%         % n=14;K=6;coronas=3;shift=-1;dual=1;trim=1;
%         W=ParquetTilings4NCB(n,K,coronas,shift,dual,trim,plotting);
%         subplot(2,3,k)
%         plot(W,'b','LineWidth',0.5);axis equal
%         title(['(n,K,c,s,d,t) = ['...
%             num2str(n) ',' num2str(K) ',' num2str(coronas) ',' ...
%             num2str(shift) ',' num2str(dual) ',' num2str(trim)
%         ]'],'FontSize',10)
%         axis off
%         axis tight
%         set(gcf,'Color','w')
%     end
% end
%
% return
%-----
% Copyright 2014, Cy H. Waldman
% Let me know how this works out for you at cye@att.net

iseven=inline('rem(number,2) == 0;', 'number');
isodd=inline('rem(number,2) == 1;', 'number');

% calculate the tile
z0=Gailiunas(n,K);
h=max(imag(z0));

zm=-z0+1+2*real(z0(K+2))+i*2*h;
Z1=[z0(1:K);zm(K+2:end);zm(2:K);z0(K+2:end)];
theta=angle(Z1(2*K));
Z1=Z1*exp(i*(pi/2-theta));
Z0=Z1*exp(-i*(pi/2-theta));

```

```

z0=Z0;
zsave=z0;
z1=[Z0(1:K+1);Z0(3*K:end)];
zu=[Z0(2*K:3*K);Z0(K+1:2*K)];
if dual
    z0=[z1 zu];
end

zc=-conj(z0);
theta=angle(zc(K+1)-zc(K));

zc=-conj(z0);
theta=angle(zc(K+1)-zc(K));
% precalculate zn, the thiles for each corona
prev=z0;
for c=1:coronas;
    next=[z0 prev+1];
    prev=next;
    if isodd(c)
        assignHere(['z' num2str(c)],-conj(next)*exp(-i*theta));
    else
        assignHere(['z' num2str(c)],next)
    end
end

% calculate the tiles in a sector up to the specified corona
Z0=z0;
if coronas==1 && dual && trim
    z1=z1(:,1:2:end);
end
Z1=[Z0 z1+Z0(K)];

% the Z's
last=1;
that=Z1;
for c=2:coronas
    this=that;
    next=eval(['z' num2str(c)]);
    if c==coronas && dual && trim
        next=next(:,1:2:end);
    end
    last=last+c-2;
    if dual
        that=[this next+this(2,2*last)];
    else
        that=[this next+this(2*K+1,last)];
    end
end
Z=that;

% now rotate it so that the left wall is vertical
beth=angle(zsave(2*K+1));
Z=Z*exp(i*(pi/2-beth));
% and then level with the negative x-axis
Z=Z*exp(i*pi/2);
W=Z;

```

```

for k=1:n-1
    W=[W Z*exp(-i*k*2*pi/n)];
end

if abs(shift)>0 && iseven(n)
    Wtop=W(:,1:end/2);
    if iseven(shift)
        Wbot=(W(:,end/2+1:end));
    else
        Wbot=-conj(W(:,end/2+1:end));
    end
    delx=abs(z0(K));
    W=[Wtop Wbot+delx*shift];
end

if plotting
    figure;plot(W,'b','LineWidth',0.5);axis equal
    title(['(n,K,c,s,d,t) = ['...
        num2str(n) ',' num2str(K) ',' num2str(coronas) ',' ...
        num2str(shift) ',' num2str(dual) ',' num2str(trim) ']], 'FontSize',12)
    axis off
    set(gcf,'Color','w')
end

return

function z=Gailiunas(n,K)
% classic Gailiunas tile

if K>floor(n/2)
    warning('K was too large; n was djusted to 2*K');
    n=2*K;
end

theta=(n-2)*pi/n*ones(n,1);
phi=cumsum(pi-theta);
z=cumsum(exp(i*phi));
zn=[0;z];
zs=zn-1;

% calculate the tile
z1=zn(1:K);
z2=flipud(z1-1);

z=[z2(end);z1;z2];
z=z+1;

return

```



```
function assignHere(varargin)
%ASSIGNHERE Creates a variable in the current workspace.
%  ASSIGNHERE is a little trick to get ASSIGNIN to assign a
%  variable in the current workspace.
%
%  Example:  assignHere('test', [1 2 3]);
%  If this were attempted within a function file, it would not be
%  possible to assign in the current workspace, only the caller or
%  base.

%  Doug Hull <hull@mathworks.com>      3/11/2003
%  Copyright 1984-2003 The MathWorks, Inc.
%  This function is not supported by The MathWorks, Inc.
%  It is provided 'as is' without any guarantee of
%  accuracy or functionality.

assignin('caller', varargin{:})

return
```

Appendix B: Gailiunas Tiling Program

```

function GailiunasTiling4NCB
% this program calculates the vertex spirals for the tiles derived from
% overlapping regular n-gons by Paul Gailiunas (2000).
% "Spiral Tilings" Bridges 2000 Conference Proceedings, pp.133-140.
% Available at http://archive.bridgesmathart.org/2000/bridges2000-133.pdf.
%-----
% Copyright 2014, Cy H. Waldman
% Let me know how this works out for you at cye@att.net

index=0;
level=2;
nmax=36;
for n=6:nmax
    for K=3:floor(n/2)
        arms=n/(K-1);
        if mod(arms,1)==0
            theta=(n-2)*pi/n*ones(n,1);
            phi=cumsum(pi-theta);
            z=cumsum(exp(i*phi));
            zn=[0;z];
            zs=zn-1;

            % calculate the tile
            z1=zn(1:K);
            z2=flipud(z1-1);

            z=[z2(end);z1;z2];
            z=z+1;
            % arrange the tile to congorm to Gailiunas convention
            % the origin is at the base angle and tile is rotated
            z=(z-z(K+1))*exp(i*pi);
            z=[z(K+1:end);z(2:K);0];

            % the building blocks
            zc=conj(z);
            alef=angle(zc(K+2)-zc(K+1));
            beth=angle(z(3)-z(2));
            this=(zc-zc(K+1))*exp(-i*(alef-beth))+z(2);
            zzc=[z this];

            % start the spiral with zzc
            Z=[];
            for k=1:K-1
                alef=angle(zzc(2,1)-zkc(1,1));
                beth=angle(z(2*K+1-k)-z(2*K+2-k));
                this=(zkc-zkc(1,1))*exp(-i*(alef-beth))+z(2*K+2-k);
                Z=[Z this];
            end

            % build the 1st repeating tile
            z2=[z z+1];
            zc2=[zc zc+1];
            alef=angle(zc(K+2)-zc(K+1));

```

```

beth=angle(z2(3,end)-z2(2,end));
this=(zc2-zc2(K+1,1))*exp(-i*(alef-beth))+z2(2,end);
Z2=[z2 this];

alef=angle(z(K+2)-z(K+1));
for N=3:level
    tempn=eval(['z' num2str(N-1) ' z+' num2str(N-1) '']);
    tempc=eval(['zc' num2str(N-1) ' zc+' num2str(N-1) '']);
    beth=angle(tempn(3,end)-tempn(2,end));
    this=(tempc-tempc(K+1,1))*exp(-i*(alef-beth))+tempn(2,end);
    that=[tempn this];
    assignHere(['z' num2str(N)],tempn);
    assignHere(['zc' num2str(N)],tempc);
    assignHere(['Z' num2str(N)],that);
    % assignHere is in Appendix A
end

alef=angle(z(end-1)-z(1));
for m=2:level
    temp=eval(['Z' num2str(m)]);
    for k=1:K-1
        beth=angle(Z(K,end)-Z(K+1,end));
        this=(temp-temp(1,1))*exp(-i*(alef-beth))+Z(K+1,end);
        Z=[Z this];
    end
end

W=[];
for k=0:arms-1
    W=[W Z*exp(i*k*2*pi/arms)];
end

figure;plot(W,'b','LineWidth',1)
axis equal;axis off;set(gcf,'Color','w')
title([num2str(n) '-gon with K = ' num2str(K)...
    ' and ' num2str(arms) ' arms (' num2str(length(W))...
    ' tiles)' ],'Color','b')
drawnow
if index<10
    numstr=['00' num2str(index)];
elseif k<100
    numstr=['0' num2str(index)];
else
    numstr=num2str(index);
end
namestr=['Gailiunas' numstr];
% print('-dpng',namestr)
index=index+1;
end
end
end
return

```