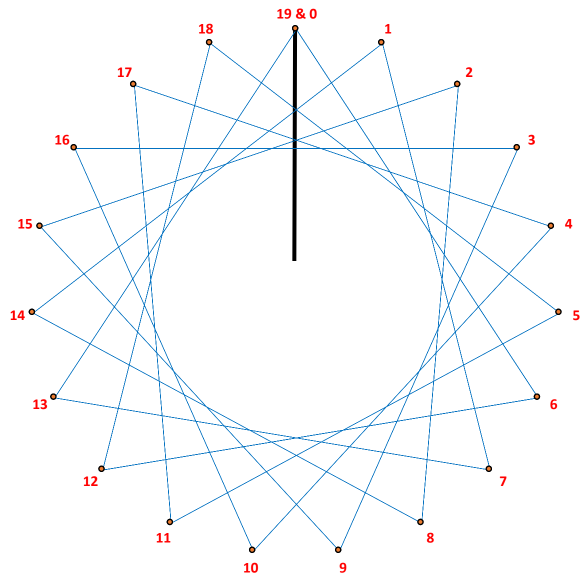
Automating *P* to Produce Images that Avoid the Donut Hole

One can always avoid the donut hole by setting ***P*** = 1 or ***P*** = ***S*** because the resulting image is simply the VF. But beyond this trivial solution, is it always possible to avoid the donut hole?

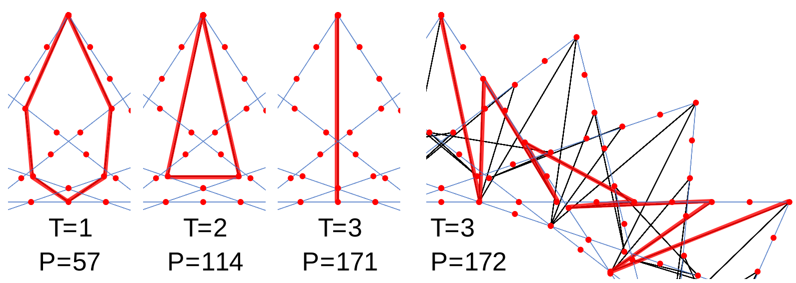
The answer is no if ***J*** = 1 (and ***J*** = ***n***-1) because all non-trivial images extend into the interior of the polygon created by connecting polygonal vertices. But once we consider 1 < ***J*** < ***n***-1 one can typically, but not always, find such images (if VCF = 1). As noted in the Waves of Images explainer, ***P*** values satisfying this condition are either on the first or last line of the VF or on a portion of other lines of the VF that are in the top half of the image. There are ***J***-1 such lines of the VF if ***J*** < ***n***/2 (having endings at polygon vertices **1** to *J*-1) or ***n***-***J***-1 having endings at polygon vertices ***n*-1** to *n*-*J*+1 if ***J*** > ***n***/2). The order in which these lines are added to the VF is the subject of the Patterns in Continuously Drawn Stars explainer.

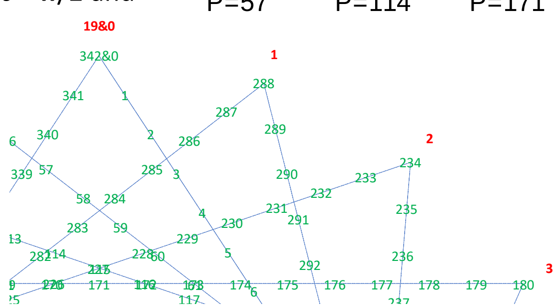
The easiest way to view this is to ask what parts of the VF intersect with the vertical radius from the center of the circle to vertex *n*&0 (radius in black to the right for ***n*** = 19 and ***J*** = 6 or 13).



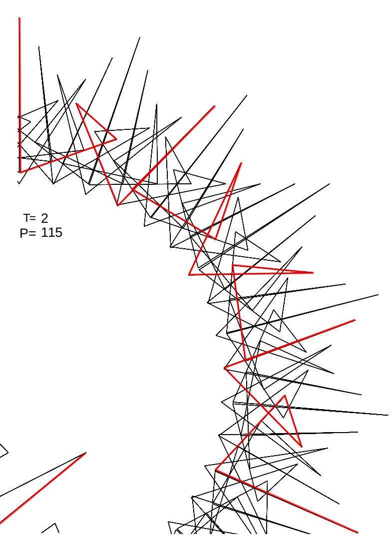
If ***J*** < ***n***/2, the first piece of the VF crossing this radius is the INTEGER(***n***/***J***)+1st line. For example, if ***J*** = 6 (shown), the first crossing occurs on the 4th line (from **18** to **5**) but if ***J*** = 4 (not shown), it occurs on the 5th line (from **16** to **1**). In the first instance, the crossing occurs close to the start of this segment but in the second it is close to the end and in both instances, the fractional part of ***n***/***J*** (1/6th or 3/4th) provides a rough balance between the two endpoints. The closest number to ***S***·***n***/***J*** will be reasonably close to this vertical line and this is the **Suggested *P***. To find subdivision points close to the vertical radius for the other ***J***-2 VF lines (if ***J*** > 2), simply multiply ***S***·***n***/***J*** times 2, 3, …, ***J***-1 (this is the times factor ***T*** in the equation noted in the text-box next to the altered *Excel* file dashboard that controls ***P***).

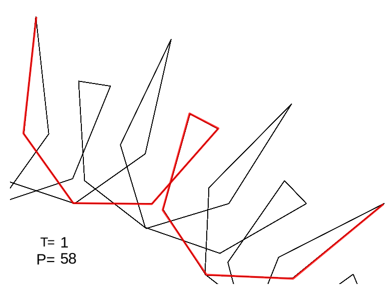
These ***n*** = 19, ***S*** = 18, and ***J*** = 6 images provide the rationale for including the addition factor ***a*** in the **Suggested *P*** equation. The first three images result from ***a*** = 0 for ***T*** = 1 (and 5), ***T*** = 2 (and 4), and ***T*** = 3 which have: 6-lines, SCF = 57; 3 lines, SCF = 114; and 2 lines, SCF = 171. The third is simply a vertical line that goes from the top to the middle of the horizontal VF line from vertex **16** to **3** above and back to the top. (More generally, vertical lines images result from ***a*** = 0 if ***J*** < ***n***/2 and ***S*** are both even, ***n*** is odd, and ***T*** = ***J***/2.)



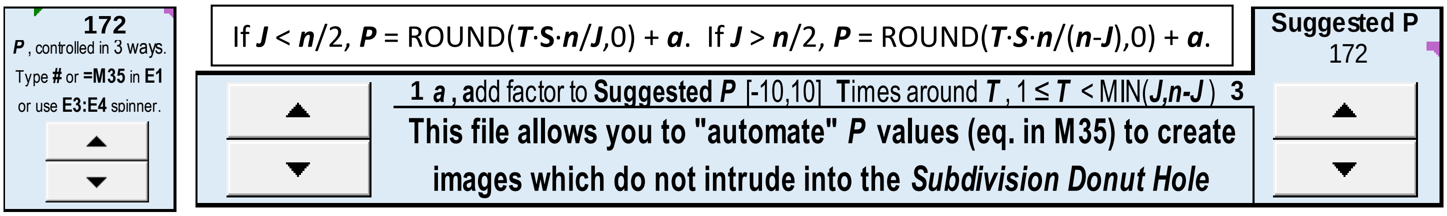


Nudging ***P*** from 171 to 172 (by setting ***a*** = 1 in the dashboard) produces the 4th image with the first **9-line cycle** shown in **red**. The image has 171 lines and SCF = 2. Note the location of the first cycle subdivision endpoints at **172**, **2**, **174**, **4**, **176**, **6**, **178**, **8** and **180** (**2** = 2·172 MOD 342 (342 = 19·18)). These are the last **5 even endpoints** on the 10th VF line (**16** to **3**) and the **first 4 even endpoints** on the 1st VF line (**0** to **6**). Only even endpoints are used because ***nS*** is even and SCF = 2. The bottom images increase ***P*** by 1 from those shown above. ***T*** = 1 has 171 lines, first cycle ends at **3** and SCF = 2. The *extreme needles* ***T*** = 2 has 342 lines, first cycle ends at **6** and SCF = 1.





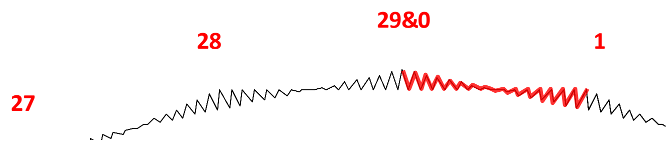
When ***J*** > ***n***/2, we must replace ***J*** with ***n-J*** in order to intersect the vertical radius. To see why, note that when ***J*** = 13 (13 = 19-6), the 2nd line of the VF (2 = INTEGER(19/13)+1) is from **13** to **7** (7 = 13+13 MOD 19) and the VF line between **13** and **7** lies below the center of the circle but the 4th (4 = INTEGER(19/(19-13)+1) line from **1** to **14** intersects the vertical radius roughly 1/6th of the way past **1** (when viewed counterclockwise). The closest number to ***S·n***/(***n-J***) will be reasonably close to this vertical radius and closest numbers to ***T·S·n***/(***n-J***) for 2 ≤ ***T*** ≤ ***n-J***-1 produces other subdivision points that are near the cross-over point on each of the other ***n-J***-2 VF lines that intersect the vertical radius (when ***n-J*** > 2).



The dashboard above shows the modifications necessary to automate ***P***. The left box notes that ***P*** can be controlled by direct numerical entry in E1, by entering an equation in E1, or by using the arrows in E3:E4. ***P*** is automated to **Suggested *P*** if the second method is chosen by typing =E35 in E1 in which case ***P*** is controlled by the arrows controlling ***a*** and ***T***.

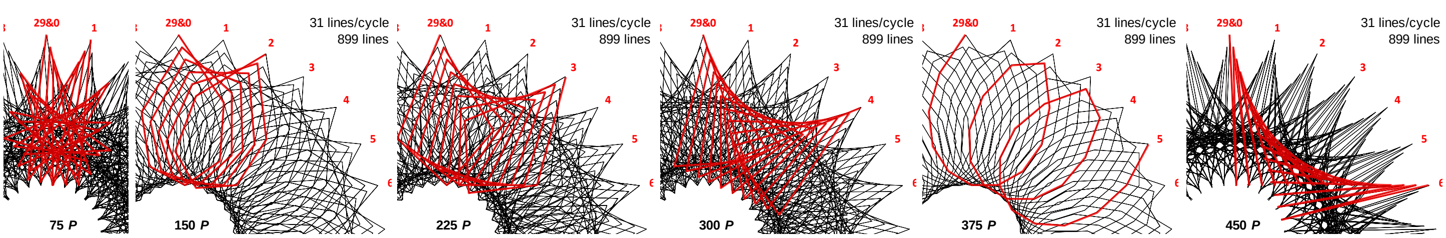
As we see in this example, the **Suggested *P*** can produce an image that is quite thin. Note that ***P*** is just a bit more than half way around (ROUND(29·31/2,0) = ROUND(449.5,0) = 450 because 0.5 rounds up) and the **31-line first cycle** is a "static" wave ending at vertex **1**. Had ***a*** = -1, the first cycle would have have been the mirror image wave ending at **28**. As a quick glance at the first 4 **Subdivision endpoints** makes clear, this is an example of a single-step image of length 2.

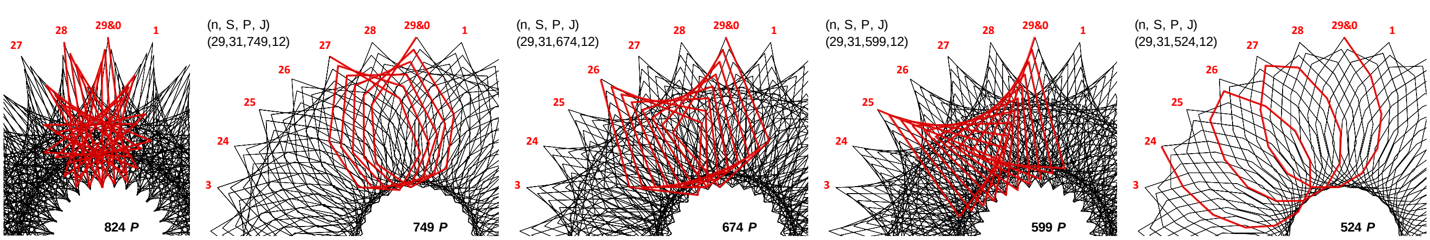
(n, S, P, J) VCF = 1 31 lines/cycle
(29,31,450,2) SCF = 1 899 lines



Subdivision endpoint		
k	***kP***	***E_k*** =MOD(***kP***,***B***)
1	450	450
2	900	1
3	1350	451
4	1800	2

The 11 partial images below show the **first cycle** for ***T*** = 1 to ***T*** = 11 for ***J*** = 12 (and ***a*** = 0) with ***n*** = 29 and ***S*** = 31 as above. The images are organized as follows: the first row is ***T*** = 1 to 6 from left to right, and the second row is ***T*** = 11, 10, 9, 8, and 7 from left to right. Many of these images are like those in polygons and stars in a cycle and ***T*** = 6 is a porcupine star.





Notice the following: there are really only 6 static images. Each of the first 5 in the top row has a counterpart in the bottom row where the only difference is ending vertex of the first cycle. The rule on this is clear: for ***T*** < ***J***/2, replace ***T*** with ***J-T*** and the same static image appears. Also note that ***P*** values from these paired static images are related to one another in a simple way: they sum to 899 = ***n·S***, just as expected from 2.4b. Finally, the end of the first cycle increases from **1** to **6** then increases from **-5** (**24**) to **-1** (**28**) as ***T*** ranges from 1 to 11 = ***J***-1 (in the above example). This vertex by vertex progression of the end of the first cycle vertices does not always hold. For example, change ***n*** to 37 keeping ***S*** = 31 and ***J*** = 12 and the first cycle ends at vertex **5** (for ***T*** = 1 at right). ***T*** = 2-11 cycle endings are: **-2**, **3**, **-4**, **1**, **6**, **-1**, **4**, **-3**, **2**, and **-5**.