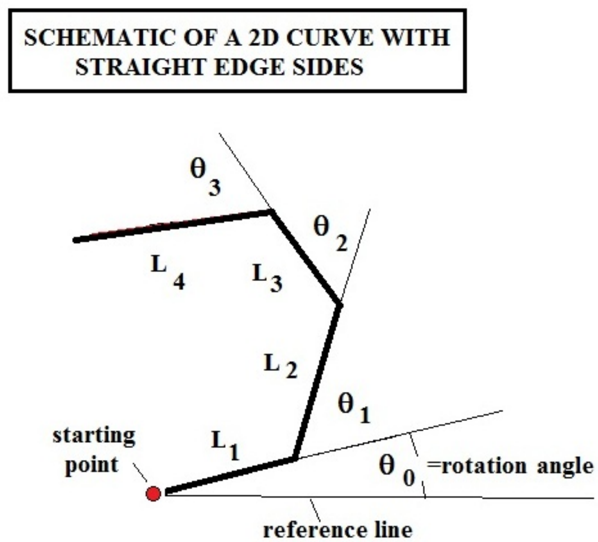
GENERATING 2D CURVES USING GENETIC ALGORITHMS

Several years ago we showed (see http://www2.mae/ufl.edu/~uhk/ Genetic-Codes.pdf) how 2D curves formed by connecting straight lines of variable length L_n can be generated by simply giving the individual line length and the angle θ_n between the nth and n+1 line segments. We want here to expand on these discussions by looking at what happens when the connected straight line side-lengths are stretched by a fixed amount k and the whole resultant 2D curve is rotated by a fixed degree θ_0.

Our starting point is the following schematic for a curve with straight line edges-



Once the angle of the first line is specified, all other lengths are uniquely defined by their length L_n and the angle θ_n they make with respect to the next line segment L_{n+1}. For a curve which becomes closed after n elements have been connected, the sum of the angles of such polygons will always add up to 2π radians. The concatenation of n lines each with its own specified angle forms the sequence-

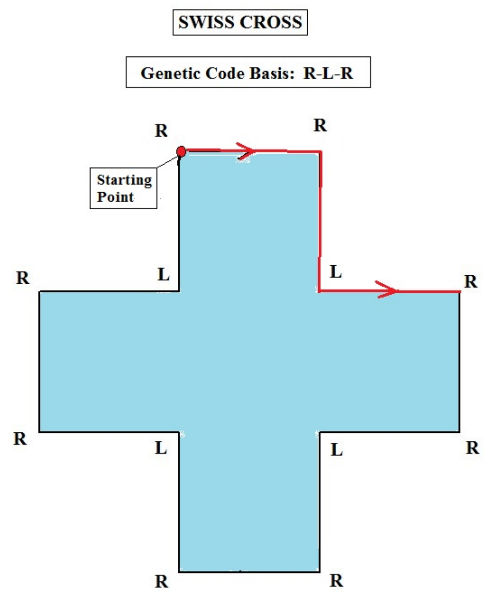
$$[L_1,\theta_1]-[L_2,\theta_2]-[L_3,\theta_3]-....-[L_n,\theta_n]$$

This may be thought of as the genetic code for producing a unique 2D curve. To fix the orientation of the curve one adds the additional element $[0,\theta_0]$ with θ_0 being the angle L_1 makes with respect to the x axis.

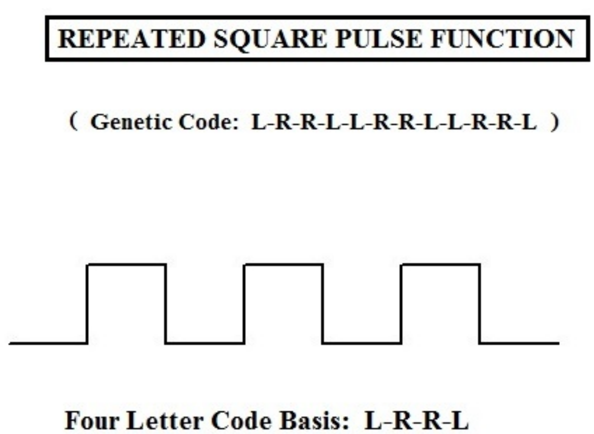
One can think of the above sequence as being equivalent to a genetic code for constructing the presently considered straight edged curves. Many special cases exist. For example, if θ is allowed to only take on the values π/2 ,0, or -π/2 and all line segments are taken to be exactly 1, the code will depend on only a left (L=+π/2), right(R=-π/2) , or no turn(0) at the end of unit length straight lines. This type of code is the simplest to envision. Take for example, the code-

R-L-R-R-L+R-R-L-R-R-L-R

Here we see the code basis is the chunk R-L-R which repeat indefinitely but will form a closed curve after 3 repetitions of the original basis. Here is the resultant closed curve-



We see that the code reproduces a standard Swiss Cross. Many other 2D curves may be produced with this simpler type of code. Another curve which is easily generated is one whose code basis is L-R-R=L. It produces the following repeated square pulse function-



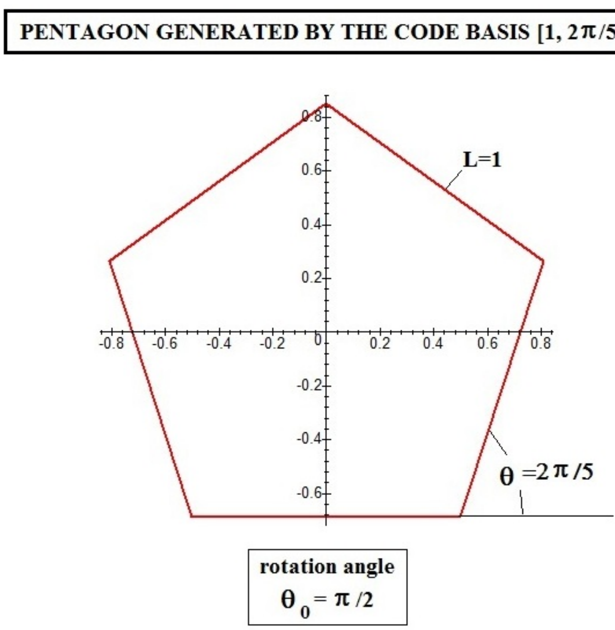
We next look at what will generally produce more complicated curves where θ_n can take on angles other than just 0 or $\pm\pi/2$. For the more general case we need to not only give the element's length but also the not necessarily right angles between neighboring elements. One of the easiest of such codes occurs for regular polygons. In particular, the genetic code-

$$[1,2\pi/5]\text{-}\ [1,2\pi/5]\text{-}\ [1,2\pi/5]\text{-}\ [1,2\pi/5]\text{-}\ [1,2\pi/53]$$

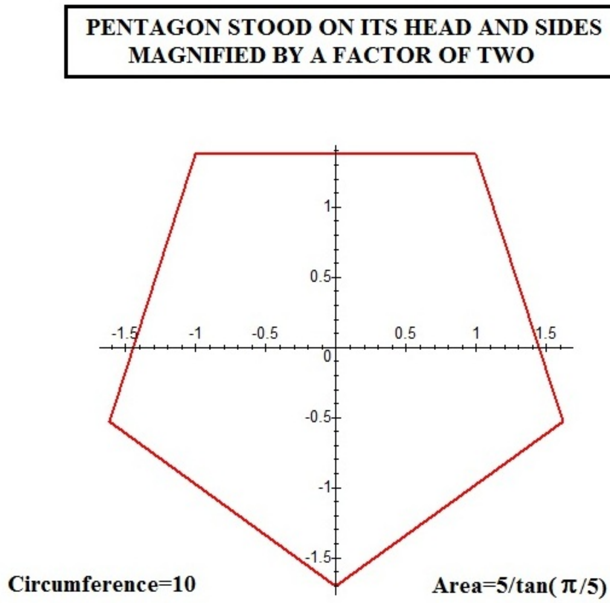
has the single term $[1,2\pi/5]$ as its basis. Hooking together the basis five times produces a closed 2D curve since $5(2\pi/5)=2\pi$. The curve is recognized as a regular pentagon with each of its larger radii equal to $1/2\sin(\pi/5)=0.85065$. The sides have unit length. Here is our MAPLE computer program which produces the pentagon predicted by this code-

```
listplot([seq([1/(2*sin(Pi/5)),Pi/2+2*Pi*(n)/5],n=1..6)],coords=polar,axes=normal,color=red,thickness=2,numpoints=4000,scaling=constrained);
```

On running the program, we find the regular pentagon –



Notice we had to add a starting angle of $\pi/2$ to get the bottom element to lie horizontal. The pentagon has an area of $A=5/(4\tan(\pi/5))=1.720477$. To accomplish a magnification of this pentagon curve one simply needs to multiply L=1 by k. A rotation of φ radians from the present orientation is gotten using a $\theta_0=\pi/2+\varphi$. So a magnification by a factor of two and a π radian rotation produces the upside down pentagon-



It is clear that doubling the side-length of a regular polygon will increase the area by a factor of four.

We next look at a case where the L_ns vary but the θ_ns are held constant. One of many genetic codes which satisfies these conditions is-

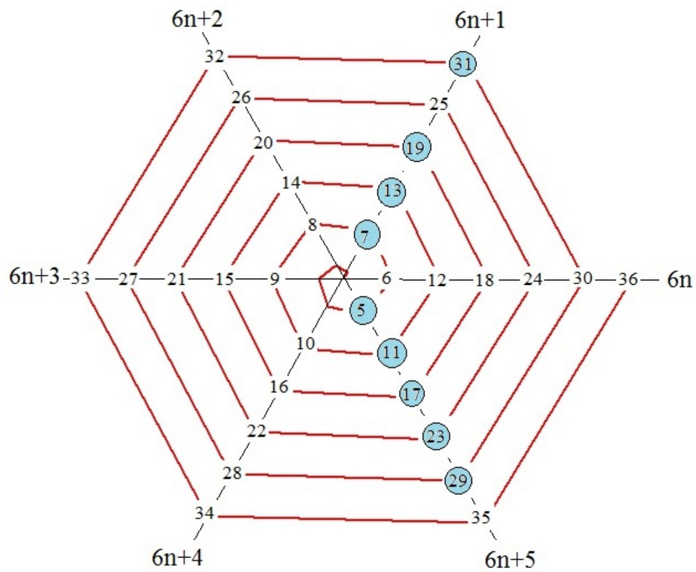
$$[1,\pi/3]-[2,\pi/3]-[3,\pi/3]-[4,\pi/3]-[5,\pi/3]-\ldots$$

This clearly is an open curve whose radial distance from the origin increases by one unit per element. The element length is also increasing by one unit per element. Expressed in polar coordinates we have-

```
listplot([seq([n,Pi*(n)/3],n=1..36)],coords=polar,axes=normal,color=red,thickness=2,numpoints=4000,scaling=constrained);
```

This produces the interesting hexagonal integral spiral which we discovered a few years ago and one which has been used extensively by us in discussing the location of prime numbers. We have added a few radial lines to allow a distinguishing of different Mod(6) values for the integers. The figure looks like this-

HEXAGONAL INTEGER SPIRAL AND THE LOCATION OF THE FIRST FEW Q PRIMES



For the present purpose it is sufficient to point out that the code for this hexagonal spiral follows the relatively simple form-

$$[L_n , \theta]= [n, \pi/3]$$

It has a non-repeating basis of infinite length and is an open curve.

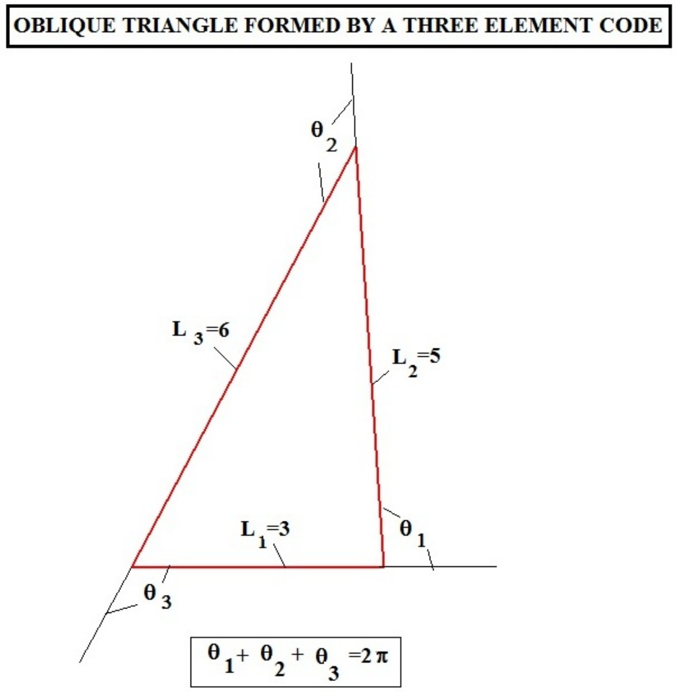
Finally let us look at a code where both L and θ vary. We consider the three element basis given by-

$$[3, \text{arcos}(1/15)]\text{-}[5,\text{arcos}(-13/15)]\text{-}[6,\text{arcos}(-5/9)]\text{-}\ldots$$

What is this? We note that the basis forms a closed oblique triangle since-

$$\text{arccos}(1/15)+\text{arcos}(-13/15)+\text{arcos}(-3/9)=2\pi$$

The closed resultant 2D curve looks like this-



This oblique triangle has a circumference of C=14 and an area (by Heron's Formula) of A=2 sqrt(14).

We have shown through various examples above that one can construct an infinite number of 2D curves consisting of straight line segments connected to their neighbors through specified angles. Relatively simple code basis provide information for constructing intricate curves just like genetic codes are the blueprints for complex organic molecule production.

May 16, 2015