On Transforming Spirographic Output with Trigonometric and Other Functions

T.Gangopadhyay
XLRI
C.H.Area (E), Jamshedpur,
India

ABSTRACT
In spirographs generated by Farris equations there is always a manifold symmetry. In the present paper a study has been made of the effect of diverse trigonometric and other transformations on the output of a spirograph. The effect is to reduce symmetry and produce various flying figures and other recognizable shapes.

General Terms

Wheel, Algorithm, Turbo C++, Program.

Keywords
spirograph, transformation,trigonometric, logarithmic.

1. INTRODUCTION
In earlier papersa study has been made of the effect of the average of two special transformations on standard escape-time fractals (Gangopadhyay[4]) as well as their effect on popcorn fractals (Gangopadhyay[5]). In yet another paper the effect of related trigonometric coefficients on affine transformations in terms of the IFS fractals generated by them(Gangopadhyay[6]) has been studied. In the present paper, trigonometric, logarithmic and other functions are used on a typical spirograph(Fisher[1]). Spirographs such as hypotrochoid and epitrochoid have been widely studied(Garg[3],lawrence[7],Little[8],O'Connor[9]).The version of spirograph used here as a starting point is the one generated by Frank A. Farris in his paper "Wheels on wheels on wheels"(Farris[2]).. Usual spirographs have a manifold symmetry and hence resemble intricate geometric designs.

Fig. 1: A typical spirograph

Figure 1 generates a typical spirograph that uses three wheels.

The application of a suitable transformation reduces the multiple symmetry and makes the resulting image more focussed.The result is often recognizable flying creatures such as iconic figures like batman or crowned faces. This new form of rendering is the distinctive feature of this paper.

2. THE ALGORITHM
The generalised Farris equation for n interacting circles is

$$\mathbf{Z(t)}= \sum_{k=1}^{k=n} a_k e^{2\pi i(n_k t+\theta_k)}, \quad t \in [0,1].$$

In this paper n=3 and θk =0 for all k. In the code in the next section the Farris equation for three circles appear as

```
z=complex(a1,0)*exp(ii*complex(cn1*t,0)+complex(cs1,0))+
complex(a2,0)*exp(ii*complex(cn2*t,0)+complex(cs2,0))+
complex(a3,0)*exp(ii*complex(cn3*t,0)+complex(cs3,0));
```

Instead of plotting this directly, as would be the case for a spirograph, some further transformations are applied on z. These range from trigonometric functions such as sin, cosine, arctan functions as well as log and square root functions. Sometimes the real and imaginary parts of z are switched by the use of flip function.

In section 4, a variety of output is generated by choosing suitable values for n1,n2 and n3. In the final section, the output is further controlled by an additional condition as to which pixels are to be realized.

The next section contains a programming code in Turbo C++ and also the output generated.

3. THE CODE
The code uses a function spiro which is declared first.

```
complex flip(complex c)

{return complex(imag(c),real(c)); }

void spiro()

{floatn1=38, n2=-18,n3=-2;

float a1=1,a2,a3=1,s1=0,s2=0,s3=0,p=3.14;

float cn1,cn2,cn3,cs1,cs2,cs3;

complex ii=complex(0,1),z;

cn1=2*p*n1 ;  cn2=2*p*n2 ;cn3=2*p*n3 ;

cs1=2*p*s1 ; cs2=2*p*s2 ; cs3=2*p*s3 ;

for(a2=1.75;a2>1.5;a2-=.1)

for(float t=0;t<=1;t+=.00001)

{cs3=2*p*s3 ;

z=complex(a1,0)*exp(ii*complex(cn1*t,0)+complex(cs1,0))+
complex(a2,0)*exp(ii*complex(cn2*t,0)+complex(cs2,0))+co
mplex(a3,0)*exp(ii*complex(cn3*t,0)+complex(cs3,0));
```

```
z=flip(sqrt(sin(z))*atan(z));          …….(*)
putpixel(real(z)*20+320,imag(z)*20+240,(int)(norm(atan(z))*
500+1)%256+1 );}
}
void main()
{initgraph(&gdriver, &gmode, "..\\bgi");
spiro();
getch();
closegraph();}
```

The output of the sample code is a batnamesque figure illustrated in Figure 2.

.

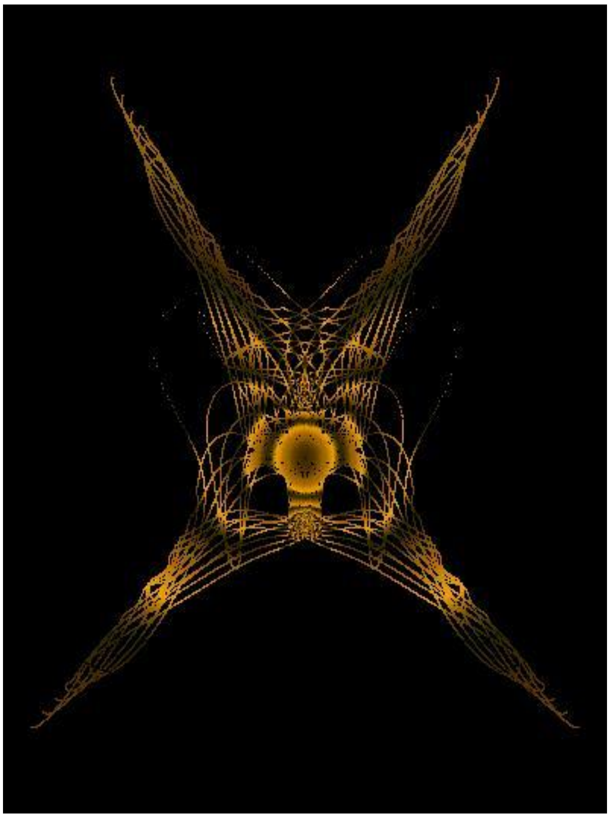

Fig 2 : Output of the sample code- Batman

4. VARIATIONSON THE SAME THEME

By changing the values of n1,n2,n3 in the sample code as well as altering the form of the function in (*) a number of interesting figures may be generated.,

VARIATION 1.

Let n1=3,n2=-19,n3=-13,a1=1.7 and (*) be replaced by z=log(z)*sqrt(cmplx(1,0)-z);z=-flip(z);

Then the resulting output is illustrated in Figure 3.

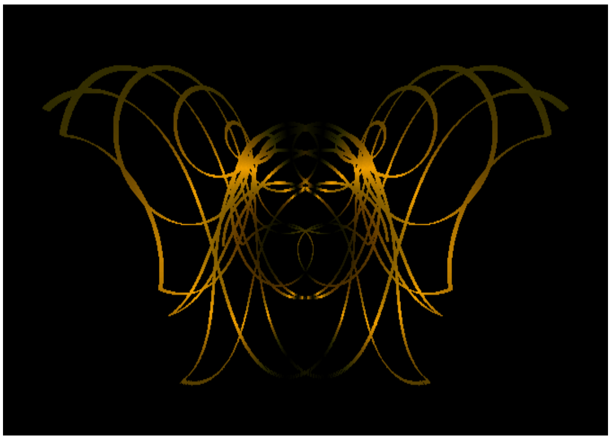

Fig. 3 : Variation 1 - Batgirl

VARIATION 2.

Let n1=45,n2=-35,n3=-3 and (*) be replaced by
z=((flip(log(flip(z))))*sin(flip(sqrt(z))));
Then the resulting output is illustrated in Figure 4.

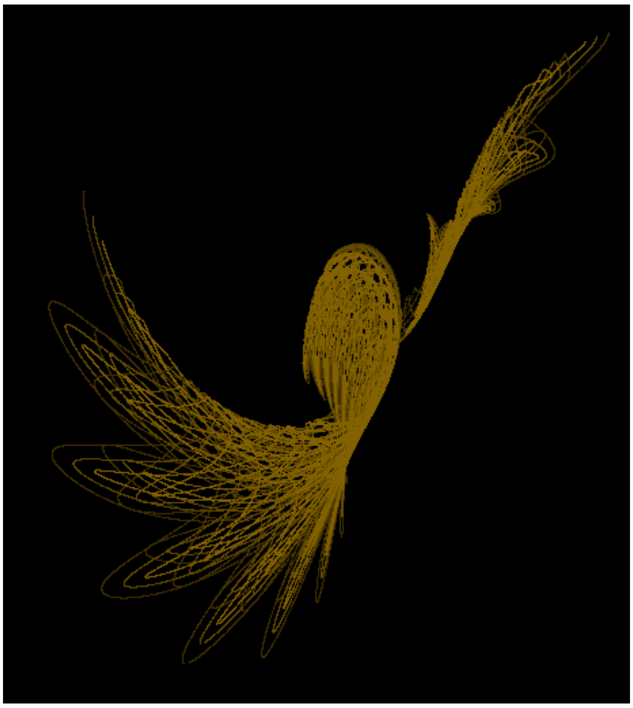

Fig. 4 : Variation 2 – Dancing Bird

VARIATION 3.

Let n1=3,n2=-19,n3=-13,a1=1.7,a2=1 and (*) be replaced by z=atan(sqrt(z))*sqrt(cmplx(1,0)-z);z=-flip(z);Also, let the multiplier of real(z) and imag(z) vary from 80 to 83.

Then the resulting output is illustrated in Figure 5.

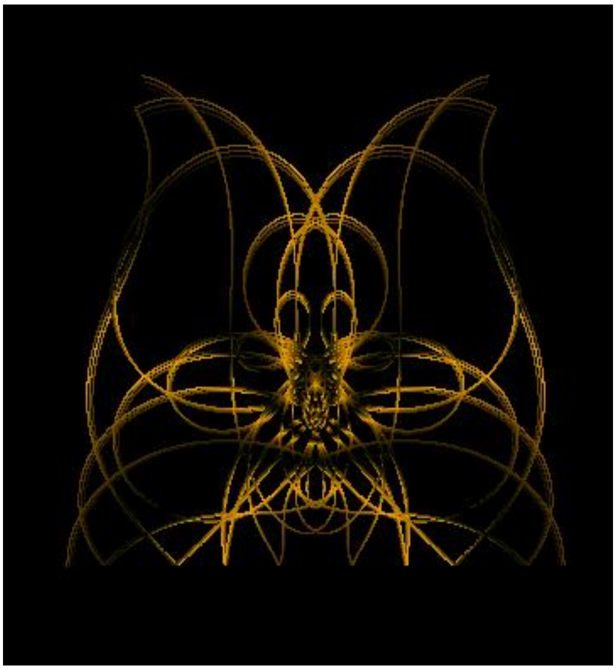

Fig. 5 : Variation 3 – Winged Bat

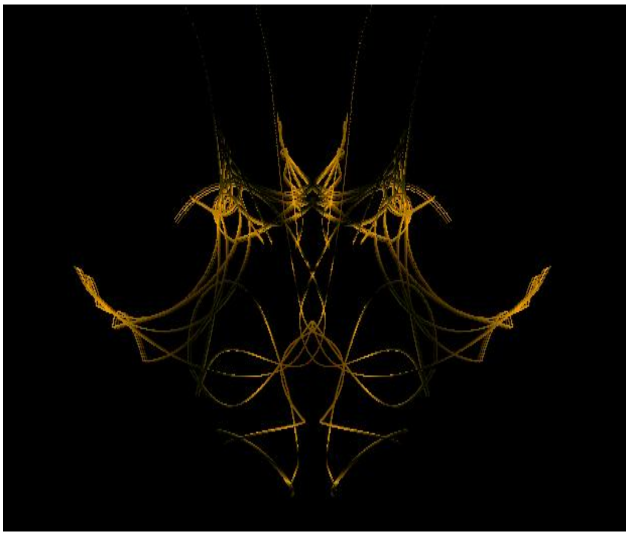

Fig. 6 : Fairy King

Fig. 7 : Anubis

OTHER VARIATIONS

For Figures 5,6,7 and 8 we give the documentation below:

Figure 6:n1=3,n2=-19,n3=13,a1=1.7,a2=1 and (*) is replaced by z=atan(log(z))*log(cmplx(1,0)-z);z=-flip(z);The multiplier is as in variation 3.

Figure 7:n1=3,n2=-9,n3=-13,a1=1.7,a2=1 and (*) is replaced by z=atan(sqrt(z))*sqrt(cmplx(1,0)-z);z=-flip(z);The multiplier is as in variation 3.

Figure 8:n1=38,n2=-18,n3=-2,a1=17 and (*) is replaced by z=((flip(atan(flip(z))))*flip(sqrt(sin(-z))));

Figure 9: n1=14,n2=-6,n3=-2, and (*) is replaced by z=flip((flip(log(flip(z))))*sqrt(flip(log(z))));

Figure 10: n1=56,n2=-56,n3=-4, and (*) is replaced

by z=cmplx(25,0)*(sqrt(atan(flip(z1))))*(sqrt(log(flip(z))));

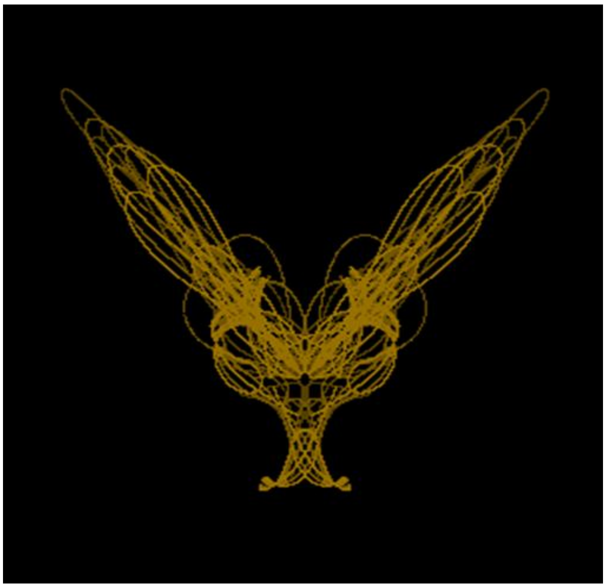

Fig. 8 : Bugs Bunn

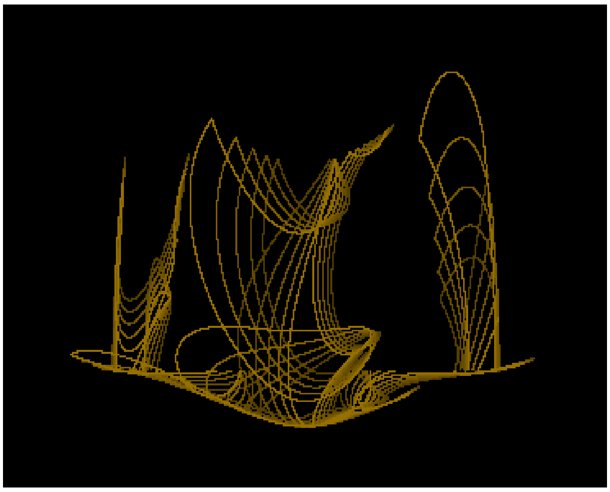

Fig. 9 : Boat

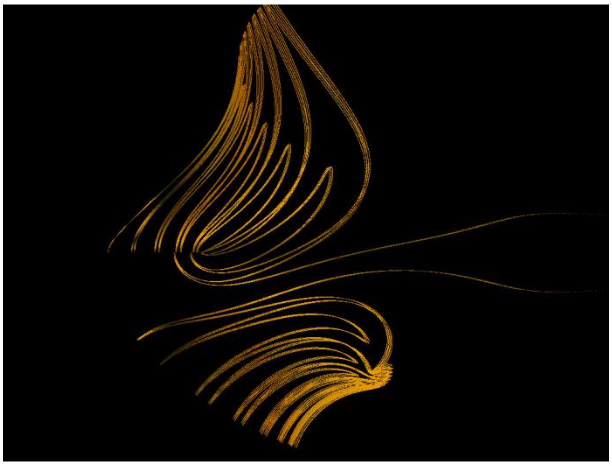

Fig. 10 : Snail

5. Controlling the output by putting an additional condition

An interesting effect can be achieved by putting an additional condition as to which pixels are to be realized. For instance if a1=1.3 and (*) is replaced by

z=cmplx(25,0)*sqrt(atan(z))*sin((z));

and, in addition, the realization of pixels is controlled by the additional condition : norm(atan(z))*1500<3556,

then the output, as generated by random values of n1,n2 and n3 is given in figure 11:

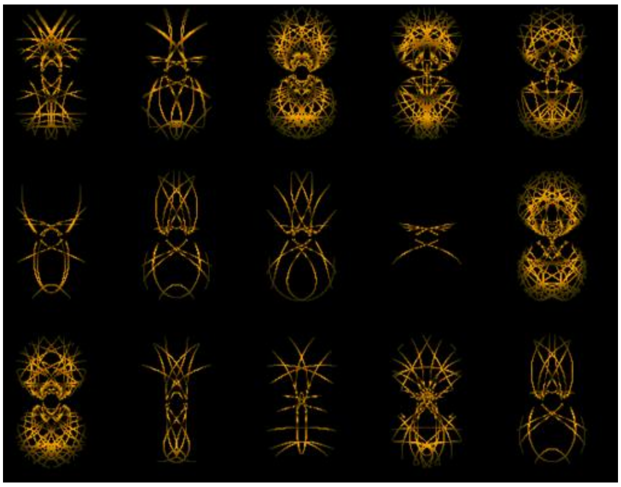

Fig. 11 : Controlling the realization

Figure 12 gives a zoom of the first three images in row one and two of Figure 10.

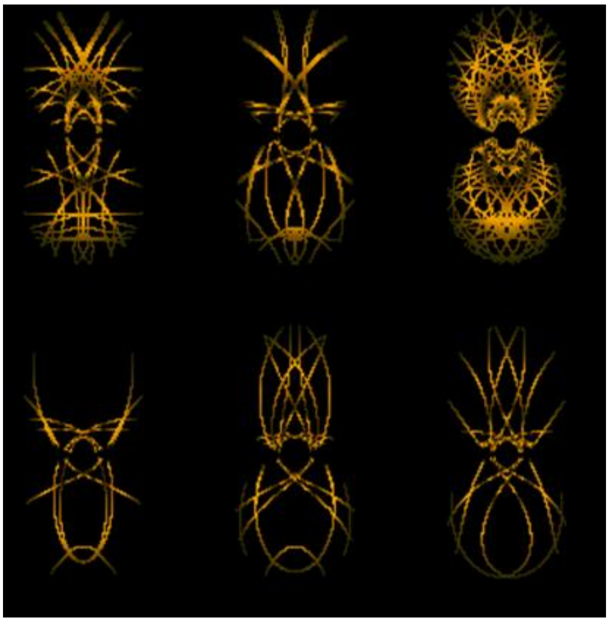

Fig. 12 : Kings and Roaches

Similarly, let n1=56,n2=-56,n3=4 and (*) be replaced by

z=sqrt(atan(flip(z)))*log(log(flip(z1)));z=cmplx(25,0)*z;

and by controlling the realization of pixels on the basis of a condition on the variable t, one may obtain the image produced in Figure 13.

Fig. 13 : Snails

6. CONCLUSION

This paper presents the effect of applying a sequence of transformations on the output of spirographs. Owing to these transformations, a highly symmetric geometric design is modified to yield asymmetric shapes that are recognizable.A similar application could be studied on harmonograph and Llissajous figures. These approacheswould be explored in future work.

7. ACKNOWLEDGMENTS

The author wishes to acknowledge his debt to the referee(s) for their constructive suggestions and encouragement

8. REFERENCES

[1] "Denys Fisher - Times Online Obituary". London: The Times. 2002-10-26. Retrieved 2008-07-26.

[2] F. Farris, Wheels on wheels on wheels – surprising symmetry, Mathematics Magazine 69, 185 (1996).

[3] A. Garg, Spirograph, last accessed 2006-09-30., URL http: //www.wordsmith.org/anu/java/spirograph.html.

[4] Gangopadhyay, T. 2012 On generating skyscapes through escape-time fractals, International journal of Computer Applications 43(2012)17-19

[5] Gangopadhyay, T. 2012 On Transforming Popcorn Fractals With Spherical And Other Functions, International journal of Computer Applications 50(2012)28-32.

[6] Gangopadhyay, T. 2012 IFS Fractals generated by affine transformation with trigonometric coefficients and their transformationsInternational journal of Computer Applications 53(2012)29-32.

[7] J. Dennis Lawrence, A Catalog of Special Plane Curves, Dover Books, 1972 [8]

[8] D. Little, Spirograph (2001), last accessed 2006-09-30., http://www.math.psu.edu/dlittle/java/parametricequations/spirograph/index.html.

[9] O'Connor, John J.; Robertson, Edmund F.,hypotrochoid", MacTutor History of Mathematics archive, University of St Andrews., http://www-history.mcs.st-and.ac.uk/,July,2012.

.