

Some links between turtle geometry and analytic geometry

Neil C. Rowe

Department of Computer Science, Code CS/Rp, Naval Postgraduate School, Monterey , CA 93943,
ncrowe at nps.edu

Abstract

The computer language Logo facilitates the teaching of analytic geometry and calculus from the notion of curvature, through its "turtle geometry" facility. We provide some theoretical basis for finding turtle geometry equivalents of familiar curves in analytic geometry, and vice versa, by some simple methods apparently previously unnoticed. In particular, we study turtle geometry programs where the curvature of a line is a trigonometric function of its orientation.

This paper appeared in *International Journal of Mathematical Education in Science and Technology*, 16, 1 (1985), 101-112. The equations were redone to make much more readable in 2007.

1. Introduction

Recently the computer language Logo has become available for the first time on a wide variety of personal computers (Abelson82, Watt83, Allen83). A distinctive feature of Logo is an emphasis in the graphics package on "turtle geometry", a geometry emphasizing discrete changes to local curvature of curves rather than coordinate systems. Despite its simplicity, turtle geometry is quite powerful, and provides an alternative viewpoint that makes certain mathematical concepts clearer, as discussed in (AbelsonDiSessa81). But an issue not well addressed in (AbelsonDiSessa81) is relationship of curves created by turtle geometry to the familiar ones of analytic geometry. In this paper, thus, we present methods for finding small-step turtle-geometry equivalents (mostly previously unnoticed) for some well-known analytic-geometry curves, and vice versa. This material provides a good exercise for calculus students, while also demonstrating some numerical analysis concepts.

2. Turtle geometry

Turtle geometry draws planar curves using two commands, FORWARD and RIGHT. A cursor called a turtle represents a point and orientation on a graphics screen. FORWARD causes the turtle to move forward in a straight line a distance (approximately in millimeter units) specified as its argument, in the direction the turtle is pointing. RIGHT causes the turtle to turn in place a number of degrees specified by its argument.

Turtles thus have a state consisting of a screen position and an orientation (or heading). The Cartesian coordinates are referred to as XCOR and YCOR, and the heading as HEADING (though we will use the abbreviation H for it in this paper). We assume the turtle starts with XCOR=0, YCOR=0 (the center of the screen), and HEADING=0 (pointing straight up). And a HEADING of 90 is straight to the right, 180 straight down, and 270 straight to the left. Note that HEADING can be thought of as the total amount a turtle has turned right since it started.

Logo programs are written in terms of procedures. A procedure consists of a sequence of commands, on successive lines. The first line contains the word TO followed by the name of the procedure, and then arguments if any. The last line of a procedure is signalled by END.

3. Curvature

We start from the classical definition of curvature in a Cartesian coordinate system, as given in most advanced calculus treatments (e.g. (Bartsch74), p. 305), assuming that clockwise is positive curvature:

$$K(x, y) = -(d^2 y / dx^2) [1 + (dy / dx)^2]^{-3/2}$$

We can put this in a form closer to turtle geometry by noting that $dy/dx = \cot(H)$, H the heading of the turtle (Rowe79), assuming $H=0$ for straight up and $H=90$ degrees for horizontal to the right:

$$K(x, y, H) = -(d^2 y / dx^2) [1 + \cot^2 H]^{3/2} = -(d^2 y / dx^2) \sin^3 H$$

We call this the "hybrid" curvature formula because it includes both Cartesian and heading variables.

Curvature is the rate at which a curve turns clockwise, and thus can be approximated in Logo by the ratio of the amount a turtle turns RIGHT to the amount it goes FORWARD -- provided the turtle takes small enough steps so as to make its path look sufficiently smooth. (Practically speaking, this means something like steps of 1 millimeter and turtle turns of five degrees on most graphics terminals that use Logo.) So if we define the curvature function we want a curve to follow by the procedure K of no arguments, either of these recursive Logo procedures will probably draw what we want:

```
TO CURVE
FORWARD 1
RIGHT K
CURVE
END
```

```
TO CURVE
FORWARD (QUOTIENT 1 K)
RIGHT 1
CURVE
END
```

We shall call them the "angle-control" and "step-control" variants, respectively. Which should we use? Unfortunately, sometimes they do not work the same, and only one gives to the correct result. This is because these are discrete approximations of smooth curves, and anytime the absolute value of K becomes large in the first version, or the absolute value of K approaches 0 in the second, the approximation breaks down. So we will have to watch for that.

In what follows we shall only concern ourselves with the *shape* of a curve, and not its size or orientation. Thus we shall ignore constant multipliers in front of a curvature formula, except sometimes the sign, and we shall throw away any constant multipliers generated in analysis routinely. We also will not concern ourselves with at what screen location we start drawing the shape (except in regrading the square root), since turtle geometry does not use a coordinate system, though the usual convention in Logo is to start at the center of a

graphics screen. But it will make a difference what heading we start with in the procedures we discuss -- different pieces of the curve will appear for different starting headings, and for some headings not defined for the curve (e.g. asymptotes), pathological curves like straight lines will be generated.

Special note: in this paper we assume that headings of greater than 360 degrees or less than 0 are meaningful and significant as indications of "how far the turtle has turned". Most Logo implementations, however, take "heading" as this modulo 360. As a result, some of the Logo procedures discussed below will not execute properly in those implementations (in particular, those where heading (H) appears other than as an argument to a trigonometric function). But it is easy to define a counter variable to use instead of HEADING.

3.1. Some examples: from $y=f(x)$ to $K=g(H)$

We give now some examples of curves initially described as $y = f(x)$ in new terms of their curvature as function of heading, $K = g(H)$. The method is (1) find dy/dx and d^2y/dx^2 in terms of x ; (2) express d^2y/dx^2 in terms of dy/dx only, no x 's; and (3) then substitute $\cot(H)$ for dy/dx , and multiply by $-\sin^3 H$ to get the curvature K in terms of H . This method is not guaranteed to work because step (2) is not possible for some $f(x)$, but it will work for some interesting functions.

3.2. Conic sections

Let us apply the hybrid formula to some familiar curves. First, the general parabola $y = ax^2 + bx + c$.

Since $d^2y/dx^2 = 2a$, a constant, following our policy of ignoring constants we just turn right $\sin^3 H$ at each turn in the "angle-control" procedure, as follows:

```
TO PARABOLA
FORWARD 1
RIGHT (PRODUCT (SIN HEADING) (SIN HEADING) (SIN HEADING))
PARABOLA
END
```

If the turtle is started with a heading between 0 and 180, it will draw a downward-tending parabola; else, an upwards-tending one. One of the first things people learn to do with Logo is to draw a circle by repeating many times a forward-right pair of commands with constant arguments:

```
TO CIRCLE
FORWARD 1
RIGHT 1
CIRCLE
END
```

(This is a special case of the well-known Logo procedure, POLY.) We can prove that in fact

$y = \sqrt{r^2 - x^2}$ is drawn with this procedure. $dy/dx = -x/\sqrt{r^2 - x^2}$, and $d^2y/dx^2 = r^2/(r^2 - x^2)^{3/2}$. Hence $1 + (dy/dx)^2 = r^2/(r^2 - x^2)$ and $(1 + (dy/dx)^2)^{3/2}/r = d^2y/dx^2$. Substituting $\cot(H) = dy/dx$,

$d^2 y/dx^2 = (1 + \cot^2 H)^{3/2} = \csc^3 H$. Hence the curvature is $-\sin^3 H * \csc^3 H = -1$, a constant.

We can generalize this to ellipses and hyperbolas. First, consider the general ellipse $y = -c\sqrt{a^2 - x^2}$,

where c is the ratio of the height to the the width. Then $dy/dx = -cx/\sqrt{a^2 - x^2}$ and

$$1 + (dy/dx)^2 / c^2 = a^2 / (a^2 - x^2),$$

$$c(1 + (dy/dx)^2 / c^2)^{3/2} = d^2 y/dx^2$$

$d^2 y/dx^2 = ca^2 / (a^2 - x^2)^{3/2}$. Hence $= c(1 + (1/c^2)\cot^2 H)^{3/2} = (1/c^2)(c^2 + \cot^2 H)^{3/2}$

Multiplying by $-\sin^3 H$, and ignoring the multiplicative constant, this then leads to a total curvature of $-(c^2 \sin^2 H + \cos^2 H)^{3/2} = -((c^2 - 1)\sin^2 H + 1)^{3/2}$. Since

$\sin^2 H > 0, c^2 - 1 > -1$ for $0 \neq c$, the curvature is always negative for c not = 0, but the degree of curve varies periodically with the heading. Here is the Logo procedure for this general ellipse:

```
TO ELLIPSE :C
FORWARD 1
RIGHT THREEHALVES DIFFERENCE (PRODUCT @!(DIFFERENCE SQUARE :C 1)
@/SQUARE SIN HEADING) 1
ELLIPSE :C
END
TO THREEHALVES :X
SQRT (PRODUCT :X :X :X)
END
```

By similar analysis we can get the curvature of a general hyperbola, $y = c\sqrt{a^2 + x^2}$. It turns out

$$dy/dx = cx/\sqrt{a^2 + x^2}, d^2 y/dx^2 = ca^2 / (a^2 + x^2)^{3/2}, c(1 - (1/c^2)\cot^2 H)^{3/2} = d^2 y/dx^2$$

and the curvature is proportional to $-(c^2 \sin^2 H - \cos^2 H)^{3/2} = -((c^2 + 1)\sin^2 H - 1)^{3/2}$. This has maximum curvature at H=90 degrees, and then gradually turns left more and more slowly, reaching an asymptote when the curvature becomes zero, or

$$(c^2 + 1)\sin^2 H = 1, \text{ or } H = \arcsin \sqrt{1/(1 + c^2)} = \text{arc cot}(c).$$

Note both the ellipse and hyperbola become a parabola when c=0, since $(1 - \sin^2 H)^{3/2} = \cos^3 H$. And when c approaches infinity, both approach $(c^2 \sin^2 H)^{3/2} = c^2 \sin^3 H$, which is also a parabola.

3.3. Inverses

Take $y = e^x = dy/dx = d^2 y/dx^2 = \cot H$, and the curvature needed to draw it is

$$-\sin^3 H \cot H = -\sin^2 H \cos H = -0.5 \sin H \sin 2H.$$

Since the inverse function of $f(x)$ looks just like it but "turned over" and oriented in a different direction, we should expect their curvature formulas to be related. We can verify this for the inverse function of

e^x , $f^{-1}(x) = \ln(x)$. Then $dy/dx = 1/x$, $d^2y/dx^2 = -1/x^2 = -\cot^2 H$, and the curvature is $\sin H \cos^2 H = 0.5 \cos H \sin 2H$.

In general, we can obtain the curvature of the inverse of a function by interchanging all $\sin(H)$ and $\cos(H)$ terms and then reversing the sign of the total expression so that it curves in the opposite direction. This is because $dx/dy = 1 / (dy/dx) = \tan(H)$. So $\tan(H)$ should be substituted for $\cot(H)$ in all occurrences of it, and vice versa. But since

$$1 + (dy/dx)^2 = 1 + \cot^2 H = 1/\sin^2 H$$

$$1 + (dx/dy)^2 = 1 + \tan^2 H = 1/\cos^2 H$$

$\sin(H)$ and $\cos(H)$ must necessarily interchange too.

3.4. Problems with inflection points

Unfortunately, we usually cannot use an expression for the curvature in terms of only the heading when the curvature changes sign somewhere. Often these conditions arise with square roots which can be either

positive or negative. An example is $y = x^3$, $dy/dx = 3x^2$, $d^2y/dx^2 = 6x$. So trying to express

d^2y/dx^2 in terms of dy/dx , we get $\pm\sqrt{12(dy/dx)} = \pm\sqrt{12 \cot H}$, and a curvature of

$\pm \sin^3 H \sqrt{12 \cot H}$. But we cannot choose only one sign to cover the whole range -- the sign must change at the inflection point, the point where $6x = 0$, $x=0$, or the place where $H = 90$. We must therefore add a conditional branch, something like this (assuming we start with a heading which is a small positive number):

```
TO CUBE
FORWARD 1
RIGHT PRODUCT (SQUARE SIN HEADING) (SQRT COT HEADING)
IF HEADING = 90 CUBERIGHTSIDE ELSE CUBE
END
```

```
TO CUBERIGHTSIDE
FORWARD 1
RIGHT MINUS PRODUCT (SQUARE SIN HEADING) (SQRT COT HEADING)
CUBERIGHTSIDE
END
```

where SQRT is the square root function (see the Logo manuals cited for the exact interpretation of this syntax). But since the curve is really composed of small straight lines, with some roundoff error in calculating the heading over a period of time, we may never actually reach a heading of zero. Instead we had better say for procedure CUBE:

```

TO CUBE
FORWARD 1
RIGHT PRODUCT (SQUARE SIN HEADING) (SQRT COT HEADING)
IF HEADING < 89 CUBE ELSE CUBERIGHTSIDE
END

```

with CUBERIGHTSIDE remaining as above.

4. Parametric equations: from $x = f_1(t), y = f_2(t)$ to $K=g(H)$

Some curves are easier to express in terms of parametric equations for x and y than in terms of a equation including both x and y. To find curvature in terms of heading for them there are two steps: (1) express the parametric equations in terms of H entirely (use $dy/dx = \cot(H)$), and (2) compute derivatives with respect to this expression to obtain the curvature. We can rewrite the basic curvature formula in terms of derivatives of H:

$$\begin{aligned}
 K(x, y, H) &= -\sin^3 H * d^2 y / dx^2 = -\sin^3 H * (d / dx)(dy / dx) \\
 &= -\sin^3 H * (d / dx)(\cot H) \\
 &= -\sin^3 H * (\csc^2 H) * dH / dx = -\sin H / (dx / dH)
 \end{aligned}$$

But also

$$dH / dx = dH / dy * dy / dx = dH / dy * \cot H$$

so also

$$K(x, y, H) = -\cos H / (dy / dH).$$

As an example, take the cycloid described by

$$x = a(t - \sin t), y = a(1 - \cos t),$$

$$\cot H = dy / dx = (dy / dt) / (dx / dt) = \sin t / (1 - \cos t) = \cot(t / 2)$$

hence $t/2 = H$, $t = 2H$, and

$$\begin{aligned}
 x &= a(2H - \sin(2H)), y = a(1 - \cos(2H)) \\
 dx/dH &= 2a(1 - \cos(2H)), dy/dH = 2a \sin(2H)
 \end{aligned}$$

So the curvature is, using the simpler expression dy/dH :

$$-\sin(H) / 2a \sin(2H) = -1 / 4a \cos(H)$$

and hence the figure can be drawn by the Logo procedure

```

TO CYCLOID
FORWARD SIN HEADING
RIGHT 1
CYCLOID
END

```

Note we use the "step-control" procedure format because $\sec(H)$ becomes infinite at 180 degrees.

5. Polar coordinates to $K=g(H)$

We can also draw curves given in polar coordinates, $r = f(\theta)$:

$$x = r \cos \theta, y = r \sin \theta$$

$$dx/d\theta = (dr/d\theta) \cos \theta - r \sin \theta, dy/d\theta = (dr/d\theta) \sin \theta + r \cos \theta$$

$$\cot H = dy/dx = (\partial y / \partial \theta) / (\partial x / \partial \theta)$$

$$= \cos \theta - \arctan(dr/d\theta) / r - \sin \theta - \arctan(dr/d\theta) / r$$

$$H = -\theta + \arccot(r / (dr/d\theta)), \cot(H + \theta) = r / (dr/d\theta)$$

The curvature of a curve expressed in polar coordinates can be found by manipulation of the x-y curvature formula, and is given in many books (e.g. (Bartsch74), p. 306):

$$K(r, \theta) = [r^2 + 2(dr/d\theta)^2 - r(d^2r/d\theta^2)] / (r^2 + (dr/d\theta)^2)^{3/2}$$

We substitute into it the relationship just derived, i.e. $(dr/d\theta) / r = \tan(H + \theta)$:

$$K(r, \theta, H) = 1 + 2 \tan^2(H + \theta) - (d^2r/d\theta^2) / r / (r(1 + \tan^2 \dots))$$

$$= 2 \cos(H + \theta) - \cos^3(H + \theta) - \cos^3(H + \theta) * (d^2r/d\theta^2)$$

$$= 2 \cos(H + \theta) / r - \cos^3(H + \theta) * (1 + d^2r/d\theta^2)$$

For particular situations we may be able to express the above only in terms of H. For example, take the logarithmic spiral $r = e^{c\theta}$ for which $\tan(H + \theta) = (dr/d\theta) / r = ce^{c\theta} / e^{c\theta} = c$. Hence:

$$dr/d\theta = ce^{c\theta}, d^2r/d\theta^2 = c^2 e^{c\theta},$$

$$K(H, r) = 2 \cos(\arctan(c)) / r - \cos^2(\arctan(c)) * (1 + c^2 / r)$$

Which just factors into a constant term divided by $r = e^{\theta} = e^{c(\arccot(c) - H)}$. Hence the curvature is just a constant times $e^{-cH} = b^{-H}$, $b = e^c$. So it can be drawn by this procedure (assuming H can increase beyond 360 degrees):

```
TO LOGSPIRAL
FORWARD EXPONENT 1.01 H
RIGHT 1
LOGSPIRAL
END
```

where EXPONENT represent the first argument taken to the power represented by the second argument.

6. Changing-argument recursive-procedure forms

There's a simpler way we can express LOGSPIRAL (given in chapter 2 of (AbelsonDiSessa81)), by giving it an argument:

```
TO LOGSPIRAL :SIDE
FORWARD :SIDE
RIGHT 1
LOGSPIRAL (:SIDE * 1.01)
END
```

where :SIDE is a parameter to the procedure.

Many procedures that draw smooth curves can be written in this form. Letting the curvature on two successive recursive calls be K1 and K2, we try to find a simple expression relating K2 to K1, preferably independent of heading. In the above, $K2 = K1 / 1.01$ which is independent of heading. Note if the curve is smooth enough, we can approximate K2 by a Taylor series about K1, that is

$K2(H) = K1(H) + (\Delta K(H) / \Delta H) \Delta H$ so there is always an "additive" approximation for any Logo step-control procedure:

```
TO CURVE :CURVATURE
FORWARD 1
RIGHT :CURVATURE
CURVE SUM :CURVATURE (DERIVK HEADING)
END
```

where DERIVK is a procedure that computes the derivative of the curvature function. But since this is only an approximation it will (usually) gradually diverge from the true curve, though slower for some functions than others.

7. From $K=g(H)$ to $y=f(x)$

Now we address the inverse problem, which is to determine the Cartesian description of a function given its curvature as a function of heading. Two approaches are possible.

7.1. From $g(H)$ to $y=f(x)$

Let $z = dy/dx$. Then if the curvature is a function of heading, $K(H)=g(H)$:

$$g(H) = -(\sin^3 H)(dz/dx)$$

$$-csc^3(\text{arc cot}(z)) * g(\text{arc cot}(z)) = dz/dx$$

$$dx = dz / -csc^3(\text{arc cot}(z)) * g(\text{arc cot}(z))$$

$$x = -\int dz / g(\text{arc cot}(z))(1+z^2)^{3/2}$$

In particular cases we may be able to solve this, then invert it and integrate with respect to x to get an f(x). For instance, consider the curve drawn by the procedure


```
TO MYSTERY1
FORWARD 1
RIGHT SIN HEADING
MYSTERY1
END
```

which draws a kind of "squashed parabola". Its curvature is $K(H) = \sin(H)$. Hence the equation is

$$x = -\int dz / (1 + z^2) = C - \arctan(z), z = \tan(C - x) = dy / dx$$

Hence $y = = \ln|\cos(C-x)|$

which is thus the mystery curve as a function of x . As another example, consider the similar procedure:

```
TO MYSTERY2
FORWARD 1
RIGHT PRODUCT SIN HEADING SIN HEADING
MYSTERY2
END
```

which draws a less-squashed parabola. Its curvature is $K(H) = \sin^2(H)$, and:

$x = -\int dz / \sqrt{1 + z^2} = C - \arcsin h(z), y = \int \sinh(C - x) dx = \cosh(C - x)$ which is a catenary. This method depends however on three steps, two integrals and a function inversion. If any of these are not possible, the method will not work. But anytime the first integral is possible, one does derive a differential equation in terms of dy/dx and x , and it may be possible to solve this with other methods, or perhaps approximate a solution by infinite series. For instance, the procedure

```
TO MYSTERY3
FORWARD 1
RIGHT EXPONENT (SIN HEADING) 5
MYSTERY3
END
```

leads to the differential equation

$$(dy/dx)^3 / 3 + (dy/dx) = x$$

which has no easy solution.

7.2. From $g(H)$ to $x=f(H)$ and $y=h(H)$

If, however, it is acceptable to determine the curve in terms of the parameter H , then we can often solve other situations too. Recall the formulas from section 4 for curvature in terms of the heading parameter exclusively:

$$K(H) = -\sin(H) / (dx/dH), \text{ or } K(H) = -\cos(H) / (dy/dH)$$

We can rearrange these expressions for $K(H)$ and integrate them:

$$dx = -\sin H dH / K(H), dy = -\cos H dH / k(H)$$

$$x = -\int \sin(H) dH / k(H), y = -\int \cos(H) dH$$

Sometimes we can then find a substitution to eliminate H and get an equation in terms of only x and y , but

not always. As an example, consider curves drawn by the "step-control" form

```
TO AMPOLY :C
FORWARD SIN PRODUCT :C HEADING
RIGHT 1
AMPOLY :C
END
```

where :C denotes a parameter. (AMPOLY stands for "amplitude modulation POLY", since the curvature of a circle is being "modulated" at some frequency.) Here the curvature is $1/\sin(cH)$, and hence

$$\begin{aligned}x &= \int \sin(H) \sin(cH) dH = 0.5 \int [\cos((1-c)H) - \cos((1+c)H)] dH \\ &= (0.5(1-c)) \sin((1-c)H) - (0.5(1+c)) \sin((1+c)H) \\ y &= \int \cos(H) \sin(cH) dH = 0.5 \int [\sin((1+c)H) - \sin((1-c)H)] dH \\ &= (0.5(1-c)) \cos((1-c)H) - (0.5(1+c)) \cos((1+c)H)\end{aligned}$$

These are parametric equations for general hypocycloids and epicycloids ((Bartsch74), p. 312-314), with the substitutions $t/2 = H$ and $(1-c)/2 = b/a$. When $c > 1$, it is a hypocycloid; when $c < 1$, an epicycloid; and when $c = 1$, a cycloid, as we showed in section 4. We can ignore the constants of integration -- they just affect the location of the shape.

As another example, take the Logo procedure ((AbelsonDiSessa81), ch. 1)

```
TO POLYSPI :SIDE
FORWARD :SIDE
RIGHT 1
POLYSPI SUM :SIDE 1
END
```

which can also be written, following the discussion of section 6, and assuming a starting heading of 0, as

```
TO POLYSPI
FORWARD HEADING
RIGHT 1
POLYSPI
END
```

Hence $K(H) = -1/H$. Then from integration by parts:

$$\begin{aligned}x &= \int H \sin(H) dH = \sin(H) - H \cos(H) \\ y &= \int H \cos(H) dH = \cos(H) - H \sin(H)\end{aligned}$$

We cannot find an easy nonparametric form in general, but note as H approaches infinity the second terms predominate, so

$$x \approx -H \cos(H), y \approx -H \sin(H), r = \sqrt{x^2 + y^2}, \theta = H$$

and $r = \theta$ is a polar expression for the curve. Thus as H becomes large the curve approaches the spiral of Archimedes ((Bartsch74), p. 317). We can show in general that

```
TO SPIRAL :N
FORWARD EXPONENT HEADING :N
RIGHT 1
SPIRAL :N
END
```

approaches $r = \theta^N$ H becomes large for $N > 0$.

7.3. Connections to Fourier analysis

The inverse of the curvature K is often called the "radius of curvature" R. Our equations for parameterized x and y can be written as

$$x = -\int R(H) \sin(H) dH, \quad y = -\int R(H) \cos(H) dH$$

and one can see these are just the formulas for the coefficients of $\sin(H)$ and $\cos(H)$ in a Fourier series expansion of $R(H)$. There is another connection to Fourier analysis, however. Note if

$$R(H) = R_1(H) + R_2(H), \text{ then}$$

$$x = -\int R_1(H) \sin(H) dH - \int R_2(H) \sin(H) dH$$

$$y = -\int R_1(H) \cos(H) dH - \int R_2(H) \cos(H) dH$$

and the resulting curve is the "vector addition" of the two original curves, as in DUOPOLY of ch. 3 of (AbelsonDisessa81). Now since we can approximate any curve sufficiently accurately with a sufficient number of terms from its Fourier series, we can approximate an *arbitrary* $R(H)$ as the sum of weighted sines and cosines:

$$R(H) = \sum_i [a_i \cos(iH/c) + b_i \sin(iH/c)]$$

where c is a constant representing how fine accuracy we desire in the approximation. Hence we can express x and y as

$$x = -\sum_i \int [a_i \cos(iH/c) + b_i \sin(iH/c)] \sin(H) dH$$

$$y = -\sum_i \int [a_i \cos(iH/c) + b_i \sin(iH/c)] \cos(H) dH$$

and the result is a weighted vector sum of hypocycloids and epicycloids, in the same way that MULTIPOLY in ch. 3 of (AbelsonDisessa81) is the sum of circles of different sizes and curvatures.

8. Conclusion

We have shown a variety of methods for obtaining small-step turtle geometry programs from equations for analytic-geometry curves, and vice versa. Unfortunately, each method only works part of the time, and no guarantees can be given. See @cite(Rowe79) for further student problems related to this material.

9. References

(AbelsonDiSessa81) Abelson, H. and DiSessa, A., *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*, MIT Press, Cambridge, MA, 1981.

(Abelson82) Abelson, H., *Logo for the Apple II*, Byte/McGraw-Hill, Peterborough NH, 1982.

(Watt83) Watt, D. C., *Learning With Logo*, Byte/McGrawHill, Peterborough NH, 1983.

(Allen83) Allen, J. R., Davis, R. E., and Johnson, J. F., *Thinking About TLC Logo*, Holt, Rinehart, and Winston, New York, 1983;

((Rowe79) Rowe, N. C., "Sine POLOYs – Some Geometric Explorations", *Creative Computing*, November 1979, pp. 92-95.

[Go to paper index](#)