

AD-A138 319

SOME LINKS BETWEEN TURTLE GEOMETRY AND ANALYTIC
GEOMETRY(U) NAVAL POSTGRADUATE SCHOOL MONTEREY CA
N C ROWE JAN 84 NPS52-84-002

1/1

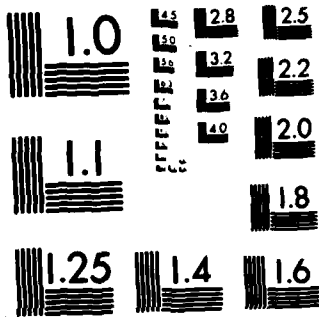
UNCLASSIFIED

F/G 9/2

NL



END
DATE
FILMED
3 84
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

2

NPS52-84-002

NAVAL POSTGRADUATE SCHOOL

Monterey, California



AD A138319

SOME LINKS BETWEEN TURTLE GEOMETRY
AND ANALYTIC GEOMETRY

Neil C. Rowe

January 1984

DTIC
COLLECTED
FEB 29 1984
E

DTIC FILE COPY

Approved for public release; distribution unlimited

Prepared for:

Chief of Naval Research
Arlington, VA 22217

84 02 27 144

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Commodore R. H. Shumaker
Superintendent

D. A. Schrady
Provost

The work reported herein was supported in part by the Foundation Research Program of the Naval Postgraduate School with funds provided by the Chief of Naval Research.

Reproduction of all or part of this report is authorized.

This report was prepared by:

Neil C. Rowe

NEIL C. ROWE
Associate Professor of
Computer Science

Reviewed by:

Released by:

David K. Hsiao
DAVID K. HSIAO, Chairman
Department of Computer Science

Kneale T. Marshall
KNEALE T. MARSHALL
Dean of Information and
Policy Sciences

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|-----------------------|---|
| 1. REPORT NUMBER NPS52-84-002 | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) Some Links Between Turtle Geometry and Analytic Geometry | | 5. TYPE OF REPORT & PERIOD COVERED |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s) Neil C. Rowe | | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, CA 93943 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 61152: RR00-01-10 N001484AF0001 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Chief of Naval Research Arlington, VA 22217 | | 12. REPORT DATE January 1984 |
| | | 13. NUMBER OF PAGES 13 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | |
| 18. SUPPLEMENTARY NOTES <i>The author</i> | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) education: calculus, education, calculus, analytic geometry, turtle geometry, Logo, computer languages, curvature, polygons, conic sections, cycloid, spiral, catenary, active learning, computers in education | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) The computer language Logo facilitates the teaching of analytic geometry and calculus from the notion of curvature, through its "turtle geometry" facility [2]. It provides 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 <i>he studied</i> turtle geometry programs where the curvature of a line is a trigonometric function of its orientation. | | |

DD FORM 1473
1 JAN 73

EDITION OF 1 NOV 68 IS OBSOLETE
S/N 0102-LF-014-6601

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Some links between turtle geometry
and analytic geometry

Neil C. Rowe
Department of Computer Science
Code 52
Naval Postgraduate School
Monterey, CA 93943



| | |
|----------------------|-------------------------------------|
| Accession For | |
| NTIS GRA&I | <input checked="" type="checkbox"/> |
| DTIC TAB | <input type="checkbox"/> |
| Unannounced | <input type="checkbox"/> |
| Justification | |
| By | |
| Distribution/ | |
| Availability Codes | |
| Dist | Avail and/or Special |
| A-1 | |

Table of Contents

| | |
|---|----|
| Abstract | 1 |
| 1. Introduction | 1 |
| 2. Turtle geometry | 1 |
| 3. Curvature | 2 |
| 4. Some examples: from $y = f(x)$ to $K = g(H)$ | 3 |
| 4.1. Conic sections | 3 |
| 4.2. Inverses | 4 |
| 4.3. Problems with inflection points | 5 |
| 5. Parametric equations: from $x = f_1(t)$, $y = f_2(t)$ to $K = g(H)$ | 6 |
| 6. Polar coordinates to $K = g(H)$ | 6 |
| 7. Changing-argument recursive-procedure forms | 7 |
| 8. From $K = g(H)$ to $y = f(x)$ | 8 |
| 8.1. From $g(H)$ to $y = f(x)$ | 8 |
| 8.2. From $g(H)$ to $x = f(H)$ and $y = h(H)$ | 9 |
| 8.3. Connections to Fourier analysis | 11 |
| 9. Conclusion | 11 |

Abstract

The computer language Logo facilitates the teaching of analytic geometry and calculus from the notion of curvature, through its "turtle geometry" facility [2]. 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.

1. Introduction

Recently the computer language Logo has become available for the first time on a wide variety of personal computers [1, 6, 3]. 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 [2].

But an issue not well addressed in [2] 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. [4], p. 305), assuming that clockwise is positive curvature:

$$K(x,y) = -(\frac{d^2y}{dx^2}) / [1 + (\frac{dy}{dx})^2]^{1.5}$$

We can put this in a form closer to turtle geometry by noting that $dy/dx = \cot(H)$, H the heading of the turtle [5]:

$$K(x,y,H) = -(\frac{d^2y}{dx^2}) / [1 + \cot^2 H]^{1.5} = -(\frac{d^2y}{dx^2}) * \sin^3 H$$

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 section 4.3), 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.

4. 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.

4.1. 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 $d^2y/dx^2 = r^2/(r^2-x^2)^{1.5}$. Hence $1 + (dy/dx)^2 = r^2/(r^2-x^2)$, and $(1 + (dy/dx)^2)^{1.5}/r = d^2y/dx^2$. Substituting $\cot(H) = dy/dx$, $d^2y/dx^2 = (1 + \cot^2(H))^{1.5} = \csc^3(H)$. 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 width. Then $dy/dx = -cx/\sqrt{(a^2-x^2)}$ and $d^2y/dx^2 = ca^2/(a^2-x^2)^{1.5}$. Hence $1 + (dy/dx)^2/c^2 = a^2/(a^2-x^2)$, and $c(1 + (dy/dx)^2/c^2)^{1.5} = d^2y/dx^2 = c(1 + (1/c^2)\cot^2(H))^{1.5} = (1/c^2)(c^2 + \cot^2(H))^{1.5}$. Multiplying by $-\sin^3(H)$, and ignoring the multiplicative constant, this then leads to a total curvature of $-(c^2\sin^2(H) + \cos^2(H))^{1.5} = -[(c^2-1)\sin^2(H) + 1]^{1.5}$. Since $\sin^2(H) \geq 0$ and $c^2-1 > -1$ for $c \neq 0$, the curvature is always negative for $c \neq 0$, 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 $dy/dx = cx/\sqrt{(a^2+x^2)}$ and $d^2y/dx^2 = ca^2/(a^2+x^2)^{1.5}$, so $c(1 + (1/c^2)\cot^2(H))^{1.5} = d^2y/dx^2$, and the curvature is proportional to $-(c^2\sin^2(H) - \cos^2(H))^{1.5} = -[(c^2+1)\sin^2(H) - 1]^{1.5}$. 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 $(c^2+1)\sin^2(H) = 1$, or $H = \arcsin[\sqrt{1/(1+c^2)}] = \text{arccot}(c)$.

Note both the ellipse and hyperbola become a parabola when $c=0$, since $(1-\sin^2(H))^{1.5} = \cos^3(H)$. And when c approaches ∞ , both approach $(c^2\sin^2(H))^{1.5} = c^3\sin^3(H)$, which is also a parabola.

4.2. Inverses

Take $f(x) = e^x$; $dy/dx = d^2y/dx^2 = e^x$, so $d^2y/dx^2 = \cot(H)$, and the curvature needed to draw it is $-\sin^3(H) * \cot(H) = -\sin^2(H)\cos(H) = -.5 \sin(H) \sin(2H)$.

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 , $\ln(x)$; $dy/dx = 1/x$ and $d^2y/dx^2 = -1/x^2$, so $d^2y/dx^2 = -\cot^2(H)$, and the curvature is $\sin(H)\cos^2(H) = .5 \cos(H)\sin(2H)$.

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

$$\begin{aligned} 1 + (dy/dx)^2 &= 1 + \cot^2(H) = 1/\sin^2(H) \\ 1 + (dx/dy)^2 &= 1 + \tan^2(H) = 1/\cos^2(H) \end{aligned}$$

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

4.3. 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 $y = x^3$; $dy/dx = 3x^2$, and $d^2y/dx^2 = 6x$. 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 $\mp \sin^3(H) \sqrt{[12\cot(H)]}$.

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 $\text{arccot}[3(0)^2] = H = 90$. 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.

5. 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 :

$$K(H) = -\sin^3(H) * d^2y/dx^2$$

$$\text{But } d^2y/dx^2 = d/dx (dy/dx) = d/dx (\cot(H)) = \csc^2(H) * dH/dx$$

$$\text{Hence } K(H) = -\sin(H) / dx/dH$$

But also

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

$$\text{so also } K(H) = -\cos(H) / dy/dH$$

As an example, take the cycloid described by

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

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

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

$$x = a(2H - \sin(2H)), y = a(1 - \cos(2H))$$

$$dx/dH = 2a(1 - \cos(2H)), dy/dH = 2a \sin(2H)$$

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.

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

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

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

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

$$\begin{aligned}\cot(H) &= dy/dx = (\partial y/\partial \theta)/(\partial x/\partial \theta) = \\ &= [(dr/d\theta)\sin(\theta) + r\cos(\theta)] / [(dr/d\theta)\cos(\theta) - r\sin(\theta)] \\ &= \cos[\theta - \text{Arctan}[(dr/d\theta)/r]] / -\sin[\theta - \text{Arctan}[(dr/d\theta)/r]] \\ &= -\cot[\theta - \text{Arccot}[(dr/d\theta)]]\end{aligned}$$

Hence

$$H = -\theta + \text{Arccot}[r/(dr/d\theta)], \text{ or } \cot(H + \theta) = r/(dr/d\theta)$$

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. [4], p. 306):

$$K(\theta) = [r^2 + 2(dr/d\theta)^2 - r(d^2r/d\theta^2)] / [r^2 + (dr/d\theta)^2]^{1.5}$$

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

$$\begin{aligned}K(\theta, H) &= [1 + 2\tan^2(H + \theta) - (d^2r/d\theta^2)/r] / r[1 + \tan^2(H + \theta)]^{1.5} \\ &= [2\cos(H + \theta) \cdot \cos^3(H + \theta) \cdot [\cos^3(H + \theta) * (d^2r/d\theta^2)/r]] / r \\ &= [2\cos(H + \theta)]/r \cdot [\cos^3(H + \theta) * [1 + (d^2r/d\theta^2)/r^2]]\end{aligned}$$

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:

$$\begin{aligned}dr/d\theta &= ce^{c\theta}, d^2r/d\theta^2 = c^2e^{c\theta} \\ K(H, r) &= 2\cos(\arctan(c))/r \cdot \cos^3(\arctan(c)) * [1 + c^2]/r\end{aligned}$$

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

```
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.

7. Changing-argument recursive-procedure forms

There's a simpler way we can express LOGSPIRAL (given in chapter 2 of [2]), 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 K_1 and K_2 , we try to find a simple expression relating K_2 to K_1 , preferably

independent of heading. In the above, $K_2 = K_1/1.01$ which is independent of heading. Note if the curve is smooth enough, we can approximate K_2 by a Taylor series about K_1 , that is

$$K_2(H) = K_1(H) + (\delta K(H)/\delta H)\Delta H$$

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.

8. 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.

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

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

$$\begin{aligned} g(H) &= -\sin^3(H) * dz/dx \\ -\csc^3(\operatorname{arccot}(z)) * g(\operatorname{arccot}(z)) &= dz/dx \\ dx &= dz / [-\csc^3(\operatorname{arccot}(z)) * g(\operatorname{arccot}(z))] \\ x &= -\int dz / [g(\operatorname{arccot}(z))(1+z^2)^{1.5}] \end{aligned}$$

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

$$\begin{aligned} x &= -\int dz/(1+z^2) = C - \operatorname{arctan}(z) \\ \text{So } z &= \tan(C-x) = dy/dx. \\ \text{Hence } y &= \int \tan(C-x)dx = \ln|\cos(C-x)| \end{aligned}$$

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:

$$x = -\int dz / \sqrt{(1+z^2)} = C \cdot \operatorname{arcsinh}(z)$$

$$\text{So } z = \sinh(C \cdot x) = dy/dx.$$

$$\text{Hence } y = \int \sinh(C \cdot x) dx = \cosh(C \cdot x)$$

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.

8.2. 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 5 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:

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

$$x = -\int [\sin(H)dH / K(H)]$$

$$y = -\int [\cos(H)dH / K(H)]$$

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

$$\begin{aligned}x &= \int \sin(H)\sin(cH)dH = .5\int[\cos((1-c)H) - \cos((1+c)H)]dH \\ &= (1/2(1-c))\sin((1-c)H) - (1/2(1+c))\sin((1+c)H) \\ y &= \int \cos(H)\sin(cH)dH = .5\int[\sin((1+c)H) - \sin((1-c)H)]dH \\ &= (1/2(1-c))\cos((1-c)H) - (1/2(1+c))\cos((1+c)H)\end{aligned}$$

These are parametric equations for general hypocycloids and epicycloids ([4], 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 5. We can ignore the constants of integration -- they just affect the location of the shape.

As another example, take the Logo procedure ([2], ch. 1)

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

which can also be written, following the discussion of section 7, 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

$$\begin{aligned}x &\approx -H\cos(H), y \approx -H\sin(H) \\ r &= \sqrt{x^2 + y^2} = H, \theta = \arctan[\tan(H)] = H\end{aligned}$$

and $r = \theta$ is a polar expression for the curve. Thus as H becomes large the curve approaches the spiral of Archimedes ([4], p. 317). We can show in general that

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

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

8.3. 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

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

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 $R(H) = R_1(H) + R_2(H)$, then

$$\begin{aligned}x &= -\int R_1(H)\sin(H)dH - \int R_2(H)\sin(H)dH \\y &= -\int R_1(H)\cos(H)dH - \int R_2(H)\cos(H)dH\end{aligned}$$

and the resulting curve is the "vector addition" of the two original curves, as in DUOPOLY of ch. 3 of [2]. 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

$$\begin{aligned}x &= -\sum_i \int [a_i \cos(iH/c) + b_i \sin(iH/c)] \sin(H) dH \\y &= -\sum_i \int [a_i \cos(iH/c) + b_i \sin(iH/c)] \cos(H) dH\end{aligned}$$

and the result is a weighted vector sum of hypocycloids and epicycloids, in the same way that MULTIPOLY in ch. 3 of [2] is the sum of circles of different sizes and curvatures.

9. 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 [5] for further student problems related to this material.

References

1. Abelson, H.. *Logo for the Apple II*. Byte/McGraw-Hill, Peterborough NH, 1982.
2. Abelson, H. and DiSessa, A.. *Turtle Geometry: The Computer as a Medium for Exploring Mathematics*. MIT Press, Cambridge, MA, 1981.
3. Allen, J. R., Davis, R. E., and Johnson, J. F.. *Thinking About TLC Logo*. Holt, Rinehart and Winston, New York, 1983.
4. Bartsch, H.-J.. *Handbook of Mathematical Formulas*. Academic Press, New York, 1974.
5. Rowe, N. C. "Sine POLYs -- Some Geometric Explorations." *Creative Computing* (November 1979), 92-95.
6. Watt, D. C.. *Learning With Logo*. Byte/McGraw-Hill, Peterborough NH, 1983.

INITIAL DISTRIBUTION LIST

| | |
|--|----|
| Defense Technical Information Center Cameron Station Alexandria, VA 22314 | 2 |
| Dudley Knox Library Code 0142 Naval Postgraduate School Monterey, CA 93943 | 3 |
| Office of Research Administration Code 012A Naval Postgraduate School Monterey, CA 93943 | 1 |
| Chairman, Code 52Hq Department of Computer Science Naval Postgraduate School Monterey, CA 93943 | 40 |
| Associate Professor Neil C. Rowe, Code 52Rp Department of Computer Science Naval Postgraduate School Monterey, CA 93943 | 25 |
| Dr. Robert Grafton Code 433 Office of Naval Research 800 N. Quincy Arlington, VA 22217 | 1 |
| Dr. David W. Mizell Office of Naval Research 1030 East Green Street Pasadena, CA 91106 | 1 |

ED
88