The curves defined by intrinsic curvature in LOGO

Izabella Foltynowicz

Adam Mickiewicz University, Faculty of Chemistry, Theoretical Chemistry Department

Ul. Grunwaldzka 6, PL 60-780, Poznań, Poland

iza@rovib.amu.edu.pl

Abstract

The aim of this article is to show how easily we can draw in Logo plane curves defined by intrinsic (local) curvature. In the centre of my interest is the Cornu spiral [2], also known as the clothoid or Euler spiral. In this main example, the curvature is a linear function of the arc length. Some other examples of curves with other functions of arc length will only be mentioned and will be developed in the future.

The intrinsic equation of the curve is also called the natural equation and is really natural for the turtle graphics. To define and draw a curve using intrinsic curvature (equation) is complementary to defining curves by extrinsic curvature (equation) which is preferable in "school" mathematics. The attempts to draw the Cornu spiral brings us to some calculations in the area of discrete modular arithmetic.

Keywords

Intrinsic curvature, Cornu spiral, Logo

1. Introduction

Lets us concentrate on the simple instruction:

repeat 360 [fd :s rt :phi]

where *s* is the arc length and *phi* is the tangential angle. The curvature is defined by

$$\kappa = \frac{d\phi}{ds} \text{ or rather } \frac{\Delta\phi}{\Delta s}$$

Complementary to Armon [1], I am interested mainly in:

repeat 360 [fd 1 rt :phi]

In this case *phi* is the measure of curvature. The curvature function of *s* can be thought of as specifying how much to turn a turtle at every moment in order to keep it moving along the curve.

2. Circle

How to draw a circle? The answer can be as follows:

repeat 360 [fd 1 rt 1]

as well as:

repeat 360 [fd 1 lt 1]

repeat 360 [bk 1 rt 1]

repeat 360 [bk 1 lt 1]

What is the convenient (natural) measure of the size of the circle drawn in such a way? It is not the radius. It is the length of circumference.

$$2\pi r = 360 \text{ pixels}$$

From this the radius can be calculated:

$$r = 180 / \pi \text{ pixels}$$

How to scale the circle drawn in such a way? If we choose the possibility:

repeat 360 [fd :x rt 1],

we will draw circles whose lengths of circumference are *360 * x* (and radiuses are *180 / π * x*). *1 / x* is a measure of the curvature. Another possibility is to use the following instruction:

repeat 360 / :y [fd 1 rt :y]

Data *y* is a measure of the curvature, $2\pi r = 360 / y$. The third possibility is to draw a circle of a given circumference:

repeat :z [fd 1 rt 360 / :z]

How to get the circle of a given radius? There are two possibilities:

*repeat 360 [fd 3.14 * :r / 180 rt 1]*

*repeat 6.28 * :r [fd 1 rt 180 / (3.14 * :r)]*

$$\kappa = \frac{180}{\pi \cdot r} = \frac{1}{r}[rad]$$

The local curvature at any particular point is defined as the reciprocal of the radius of a circle that approximates the curve at that point.

The curvature of a circle is a constant. Natural (intrinsic) equation of a circle is:

$$\kappa = const$$

where κ is the curvature. What happens when the local curvature is made to vary according to several simple rules as one goes along the curve?

3. Cornu spiral

Let the curvature be a linear function of arc length.

$$\kappa = \kappa(s)$$

To analyze this case, we can try these two simple programs:

```
to cornu1 :n
   repeat :n * 360
   [fd 1 rt repc / :n]
end
```

```
to cornu2 :n
   repeat :n * 2 * 360
   [fd 1 rt repc / :n]
end
```

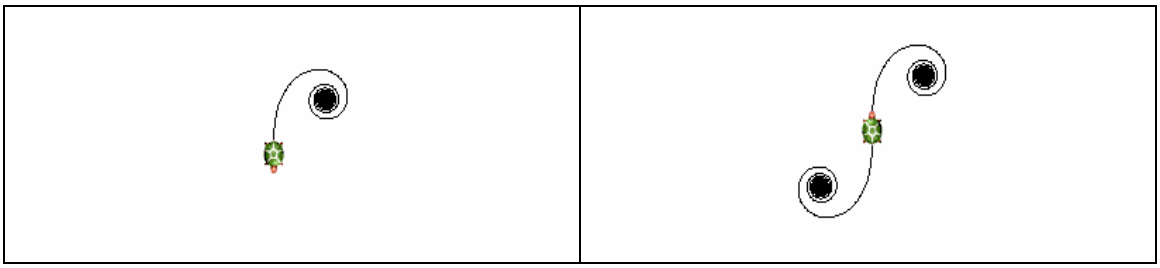
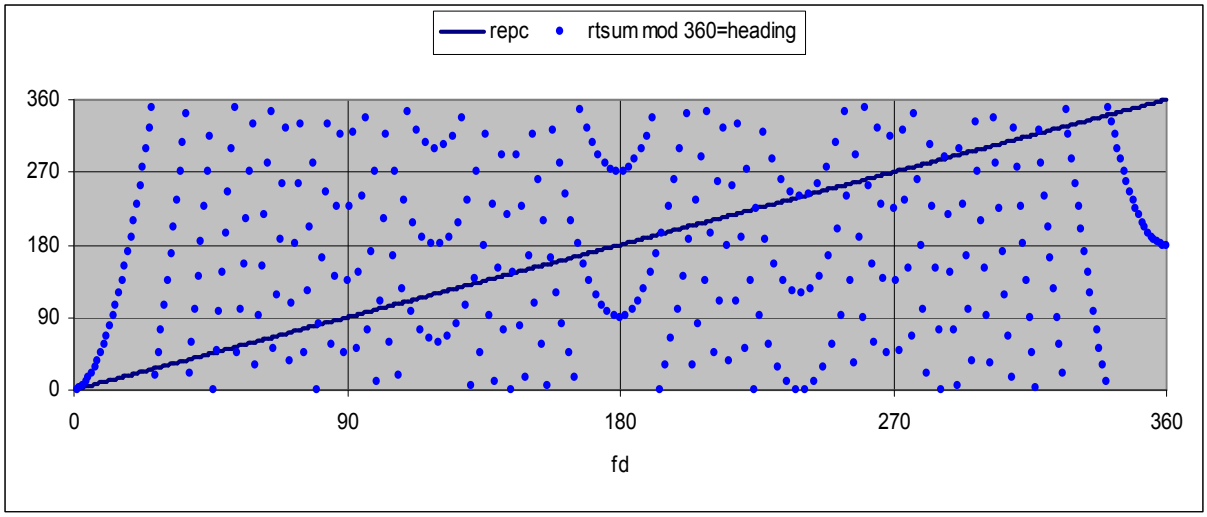
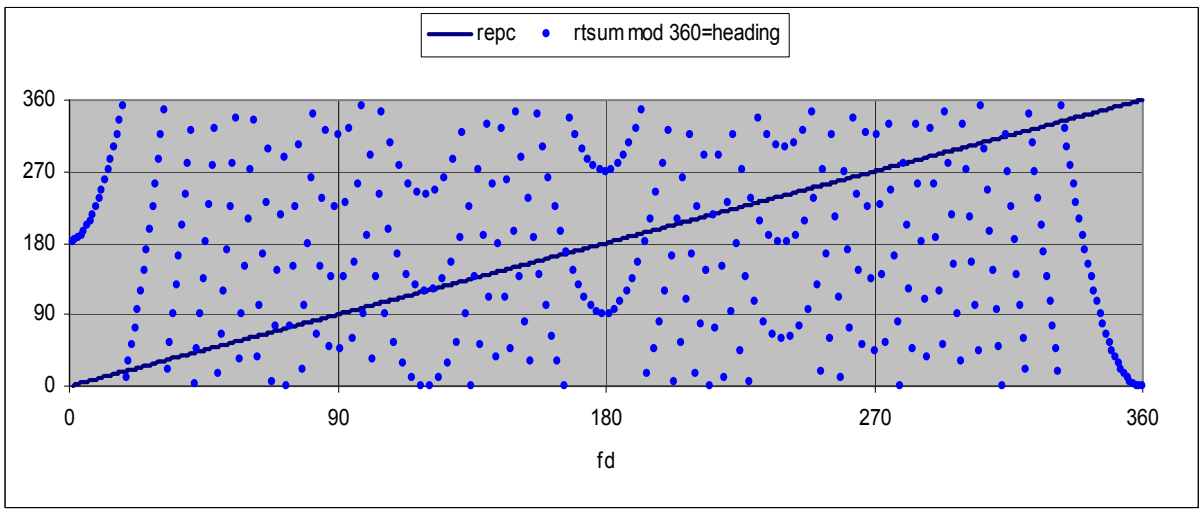


Figure 1. The results of the cornu1 and cornu2 procedures

It is really easier to write the program than to understand how it works. For every step of the turtle we know the angle it rotates (local feature) and we can also calculate its heading after each performed procedure, which is its global feature. First, let us calculate its heading for the case $n = 1$ which is convenient to calculate but not to draw (too big curvature, too small picture). In this case the heading is a sum of the arithmetic series (with the step = 1) *modulo* 360. Below is the result done and viewed graphically in Excel:





When the turtle did 180 steps, its heading is 90 and its next turn is *rt* 181. It means it starts going back the same trajectory (because its curvature is a linear function of arc length). After 360 steps it achieves his home position with the heading = 180 (the beginning of the second plot), so it is correct to fold the plot.

We can modify our calculations for n > 1.

Case $n = 1$:

repeat :x [fd 1 rt repc] *heading1 = ½ x (1+x) mod 360*

Case $n > 1$:

repeat :y [fd 1 rt repc / :n] *heading2 = ½ y/n (1+y) mod 360*

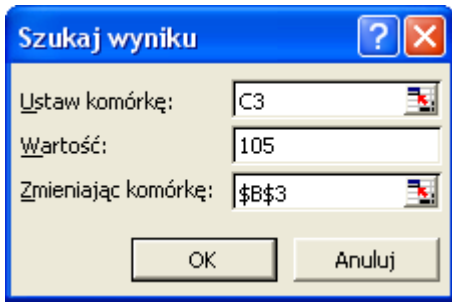
The next two figures show how to calculate in Excel the heading of the turtle for a given n and x and for the same end value of *repc* (so that $y = n\ x$). In the shown example $x = 8$, $n = 100$, $y = 800$. In case $n = 1$ *heading* = 36, in case $n = 100$ *heading* = 324. If the home heading is different than 0, its value ought to be added to the calculated value of the end heading.

	A	B	C
6	x=	8	36
7	n=	100	
8	y=	800	324

	A	B	C
6	x=	8	=MOD(0,5*B6*(1+B6);360)
7	n=	100	
8	y=	=B7*B6	=MOD(0,5*B8/B7*(1+B8);360)

Sometimes I would like to know what is the y for a given x and n that gives the same heading. To do that we can use one of the iterative tools of Excel, namely Tools/Goal Seek:

	A	B	C
1	x=	30	105
2	n=	100	
3	y=	144,4146	105



	A	B	C
1	x=	30	=MOD(0,5*B1*(1+B1);360)
2	n=	100	
3	y=	144,414630041276	=MOD(0,5*B3/B2*(1+B3);360)

We ought to remember that the solution obtained with such an iterative tool depends on the values taken at the beginning of the process. It means also that I can obtain not only one solution. Solutions are very sensitive to small changes of values (modular arithmetic). Additionally, I can choose such values of heading for which *heading mod* 360 = 0.

In Excel I can easily compare the two mentioned cases and find all regular coincidences.

$y = x*n$, n is in the cell L2.

	G	H	I	J
1	x	heading1	heading2	mod(360,h2)
7	6	=MOD(0,5*G7*(1+G7);360)	=MOD(0,5*G7*(1+G7*L2);360)	=MOD(360;I7)
13	12	=MOD(0,5*G13*(1+G13);360)	=MOD(0,5*G13*(1+G13*L2);360)	=MOD(360;I13)
19	18	=MOD(0,5*G19*(1+G19);360)	=MOD(0,5*G19*(1+G19*L2);360)	=MOD(360;I19)
25	24	=MOD(0,5*G25*(1+G25);360)	=MOD(0,5*G25*(1+G25*L2);360)	=MOD(360;I25)
31	30	=MOD(0,5*G31*(1+G31);360)	=MOD(0,5*G31*(1+G31*L2);360)	=MOD(360;I31)
37	36	=MOD(0,5*G37*(1+G37);360)	=MOD(0,5*G37*(1+G37*L2);360)	=MOD(360;I37)
41	40	=MOD(0,5*G41*(1+G41);360)	=MOD(0,5*G41*(1+G41*L2);360)	=MOD(360;I41)
49	48	=MOD(0,5*G49*(1+G49);360)	=MOD(0,5*G49*(1+G49*L2);360)	=MOD(360;I49)
61	60	=MOD(0,5*G61*(1+G61);360)	=MOD(0,5*G61*(1+G61*L2);360)	=MOD(360;I61)

$n = 100$

	G	H	I	J
1	x	heading1	heading2	mod(360,h2)
7	6	21	3	0
13	12	78	6	0
19	18	171	9	0
25	24	300	12	0
31	30	105	15	0
37	36	306	18	0
41	40	100	100	60
49	48	96	24	0
61	60	30	30	0
73	72	108	36	0
81	80	0	0	
83	82	163	1	0
91	90	135	45	0
99	98	171	9	0
101	100	10	10	0
117	116	306	18	0
121	120	60	60	0
141	140	150	150	60
145	144	0	72	0
153	152	108	36	0
156	155	210	7,5	0
171	170	135	45	0
181	180	90	90	0

	G	H	I	J
201	200	300	300	60
225	224	0	72	0
241	240	120	120	0
245	244	10	82	32
261	260	90	90	0
281	280	100	100	60
301	300	150	150	60
321	320	240	240	120
323	322	163	1	0
341	340	10	10	0
345	344	300	12	0
361	360	180	180	0

Having done all these preparations one can use the calculated numbers to try to understand how the programs placed below work.

```
to multispiral1 :x :heading
   repeat 360 / :heading [repeat 100 * :x [fd 1 rt repc / 100]]
end
```

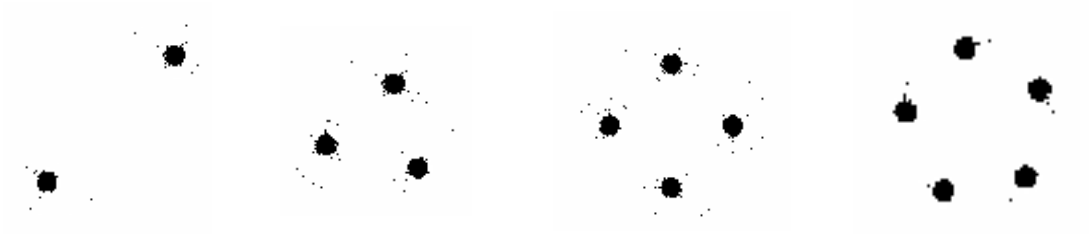


Figure 2. The results of the multispiral1 procedure

```
to multispiral2 :x :z
   repeat :z [repeat 100 * :x [fd 1 rt repc / 100]]
end
```

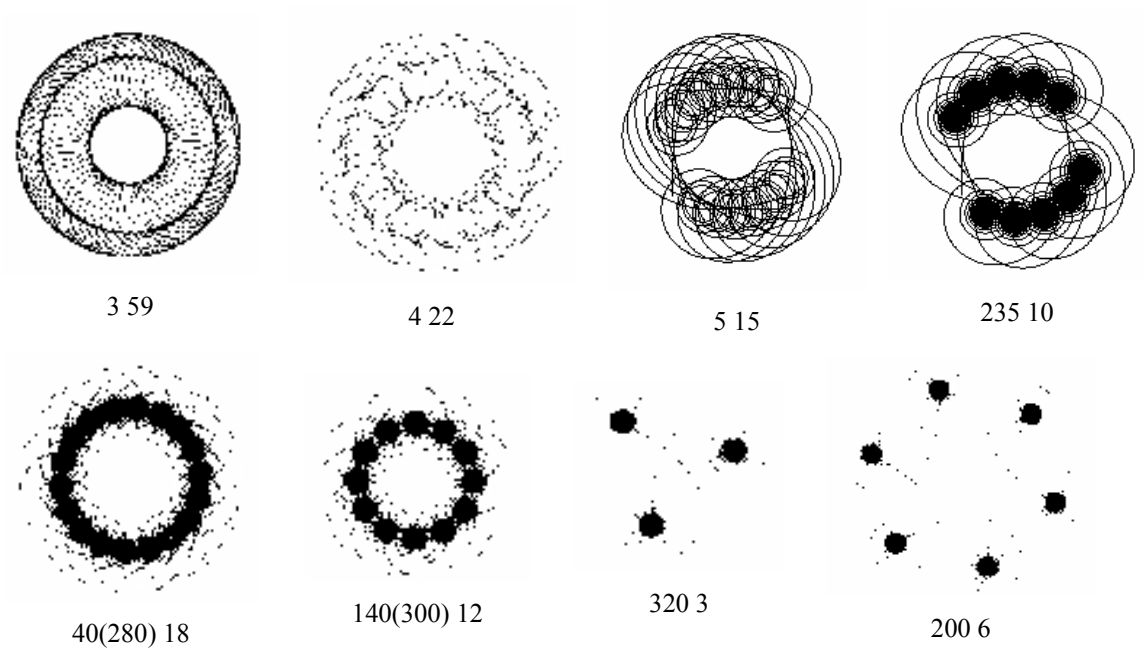


Figure 3. The results of the multispiral2 procedure

```
to multispiral3 :x :z :k
   repeat :z [repeat :k * 90 * :x [fd 1 rt repc / 100]]
end
```

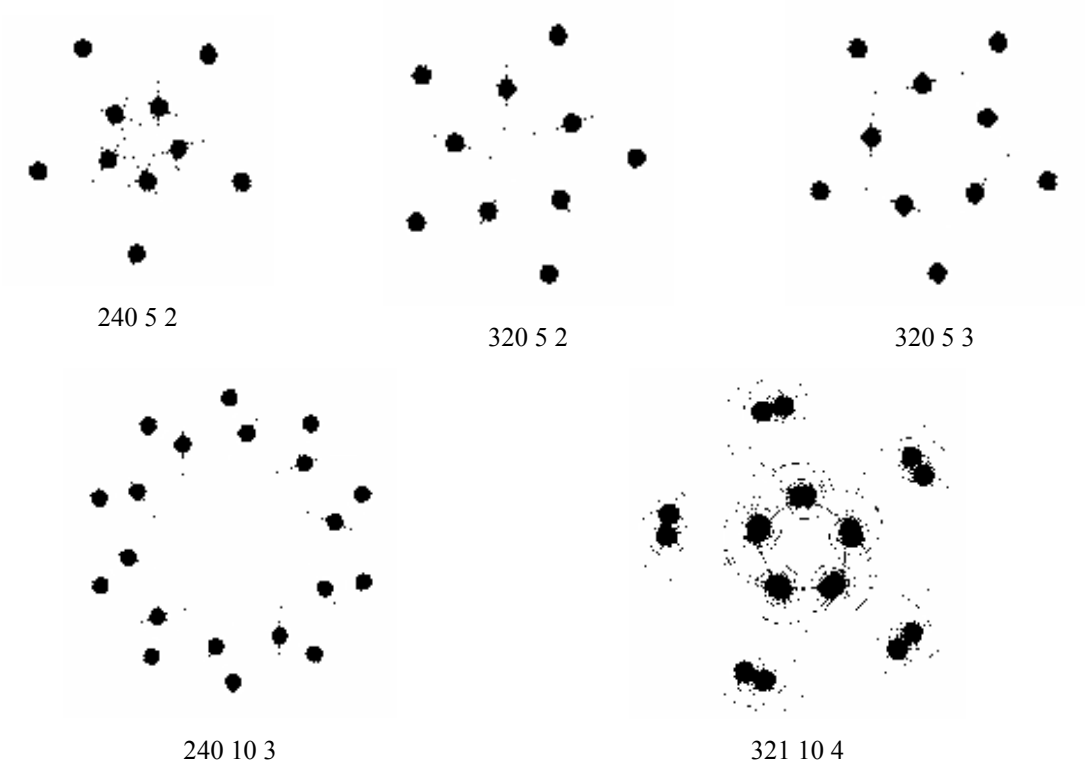
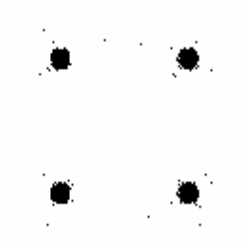


Figure 4. The results of the multispiral3 procedure

```
  repeat 2 * :n * 360 [bk 1 lt repc / :n]
  home
  repeat 2 * :n * 360 [bk 1 rt repc / :n]
  home
```



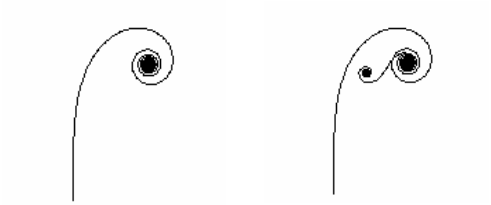
```
to start1
 cs st
 for "i [10 100 20][cornu2 :i rt :i  wait 100]
end
to cornu2 :n
 repeat :n * 2 * 360 [fd 1 rt repc / :n]
end
```

4. Curves for which the local curvature is a quadratic function of arc length

The class of "polynomial spirals" for which the curvature is a polynomial function of arc length can be tried as the further generalization of the Cornu spiral.

$$\kappa = \kappa(s^2)$$

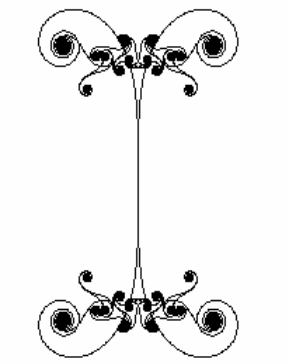
```
  to curve3 :n
     repeat :n * 360
     [fd 1 rt repc * repc / 10000]
  end
```



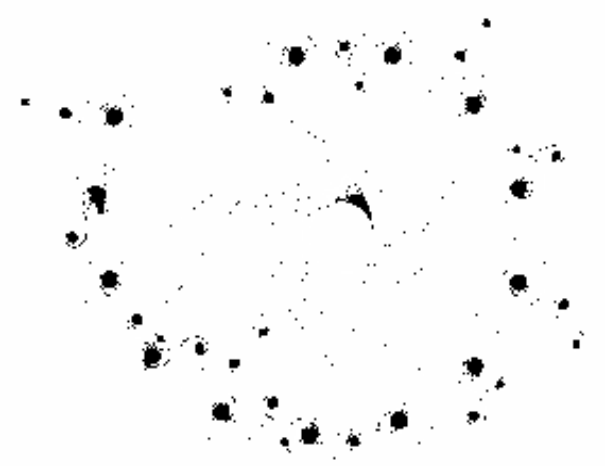
n = 5 n = 6

This is the effect of a nonlinear increase of the curvature. These nonlinear effects are sometimes really beautiful.

```
to curve13 :n
; n=15
repeat :n * 360 [fd 1 rt repc * repc / 10000] home
repeat :n * 360 [bk 1 rt repc * repc / 10000] home
repeat :n * 360 [fd 1 lt repc * repc / 10000] home
repeat :n * 360 [bk 1 lt repc * repc / 10000] home
end
```



```
to curve17 :n
; n= 10
cs st
for "c [4 10 .5]
[repeat 2 * :n * 360
[fd 1 rt (repc * repc - :c *
10000) / 50000 ] home]
end
```



Sometimes the nonlinearity generates such disordered pictures:

```
to curve2 :n
  repeat :n * 360 [fd 5 rt (repc * repc) / 50000 ]
end
```

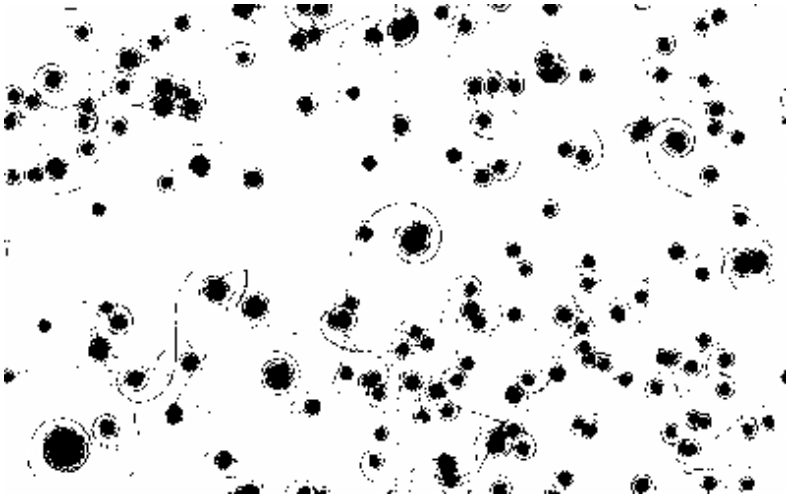


Figure 5. The result of the curve2 150 procedure

More complex quadratic function and how it depends on the value of parameter (c):

$$\kappa = \kappa\,(s^2 - c)$$

```
to curve14 :n
  cs st
  for "c [3 4.5 .5]
  [repeat :n * 360 [fd 1 rt (repc * repc - :c * 10000) / 50000 ] home
  repeat :n * 360 [fd 1 lt (repc * repc - :c * 10000) / 50000 ] home
  repeat :n * 360 [bk 1 lt (repc * repc - :c * 10000) / 50000 ] home
  repeat :n * 360 [bk 1 rt (repc * repc - :c * 10000) / 50000 ] home]
end
```

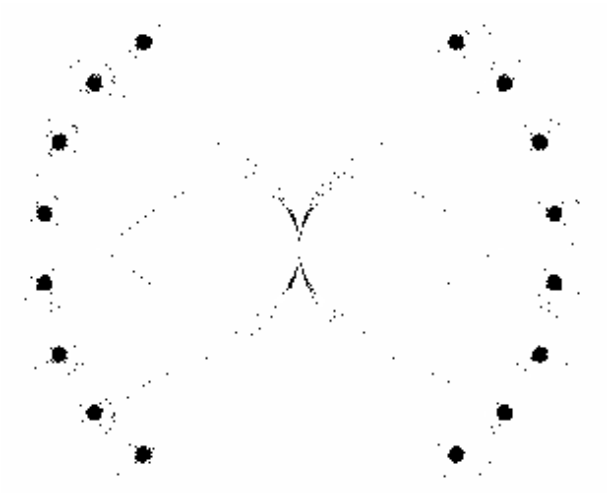
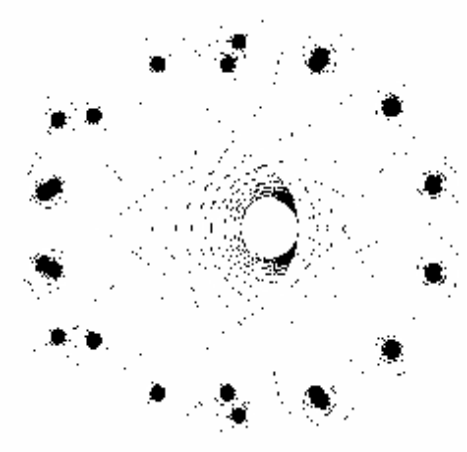


Figure 6. The result of the curve14 10 procedure

```
to curve16 :n
 cs st
 for "c [4 10 .5][repeat :n * 360 [fd 1
rt (repc * repc - :c * 10000) / 50000
]home
 repeat :n * 360 [bk 1 lt (repc * repc
- :c * 10000) / 50000 ] home]
end
```



5. Some other examples of curvatures

Circle involute [3]: $\kappa = \kappa\ (1/sqrt\ s)$

```
to curve5 :n
repeat :n * 360 [fd 1 rt 30 / sqrt repc]
end
to start5
  cs st curve5 30
end
```

Logarithmic or equilateral spiral [4]: $\kappa = \kappa\ (1/s)$

```
to curve4 :n
  repeat :n * 360 [fd 1 rt 1500 / repc]
end
to start4
 cs st curve4 15
end
```

Finally, I would like to show a very simple program which demonstrates the behavior of the curve for which: $\kappa = \kappa(\sin s)$

```
to demo
  cs ht for "i [1 4.5 .01][print :i curve8 :i  wait 100 cs]
end
to curve8 :n
  repeat :n * 360 [fd 1 rt (2 * (sin (repc - 1) / :n))]
end
```

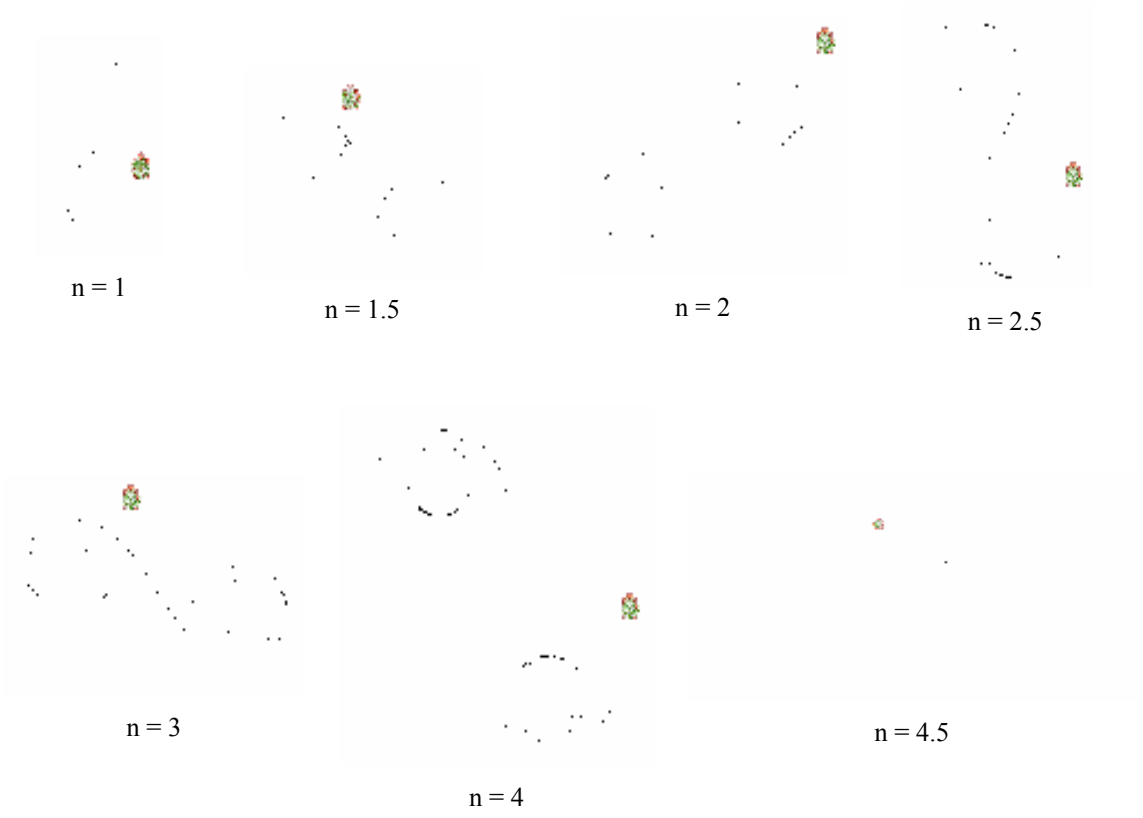


Figure 7. The results of the demo procedure

References

1. Uzi Armon, *An algorithm that translates intrinsic equations of curves into intrinsic procedures of these curves*, http://eurologo.web.elte.hu/lectures/intrins.html
2. http://mathworld.wolfram.com/CornuSpiral.html
3. http://mathworld.wolfram.com/Involute.html
4. http://mathworld.wolfram.com/LogarithmicSpiral.html