Cycloids and limaçons in the turtle graphics

Izabella Foltynowicz, *iza@rovib.amu.edu.pl*
Dept of Chemistry, A. Mickiewicz University, ul. Grunwaldzka 6, 60-780 Poznań, Poland

Abstract

In my understanding the main purpose of education in science is to help our students with deeper understanding of the main ideas of mathematics and physics and teach them to find the proper tools and methods of discovering the rules, and as a result to encourage them to study on their own to continue the so-called lifelong learning process.

The ready demonstration programs prepared by the teacher or found in the internet are applied to check the way the results depend on the data initially introduced. If the student writes a demonstration program himself/ herself/ they get a chance of deeper understanding of the problem. One approach does not exclude the other or rather they are complementary. We should study the problem from different points of view; to show different approaches to the same problem. The purpose of the paper is to give a proper tool rather than to give ready answers.

In this paper two points of view are contrasted and compared: the intrinsic (pen-coordinate system) and the extrinsic one (page coordinate system) and also Cartesian and polar plots of the sine family function. Uniform circular motion is a source of a sine wave, circular motion combined with translation gives regular cycloid and a combination of two circular motions produce Lissajous curves and when one circle rolls upon another circle the cycloids are created. The rotational motion is convenient to be realized in intrinsic way, when it is combined with translation or another rotation it is convenient to relate it to the extrinsic coordinate system but it is not necessary. Intrinsic realization of combined motions slows down the process of drawing. Then we can simulate the parallel processes using many turtles.

The following two articles have inspired me to do all this trials: (Armon, 1997) and (Farcas, 2003). Another reason for writing this work is my experience connected with quantum chemistry course when we are trying to imagine what the orbitals looks like. The source of our difficulties is the fact that the students have seen polar plots of the sine family function never before.

I start with the source definition of the sine function. Then I show the simplest way to create the Lissajous figures (the conventional way of creating them from the system of parametric equation in Excel is shortly presented in Appendix 1). Next I demonstrate how the circular motion generates a Cartesian plot and next the polar plot of the sine family functions (rhodonea, limaçons). (It was like discovering the original simplicity of the problem which we have already learnt to see as complicated.) At the next step the superposition of two waves is graphically presented in both coordinate systems. I also present the procedure for generation of a family of cycloidal curves (in a wide meaning) – hypo- and epi-cycloids are the special cases of this general procedure. Then I propose to investigate the Cartesian and polar plots of cycloidally modulated waves and also cycloidally modulated Lissajous figures. This paper presents only a few curves from a huge variety and number of possible ones that can be generated using the procedures proposed. At the and I mention the alternative way of creating polar plots described in details in Appendix 2 (in the form of a laboratory exercise) and a beautiful way of creating cycloids and other curves defined by intrinsic curvature proposed by Armon (Armon, 1997).

I intend to show that Logo is convenient and fruitful tool for investigation of the Cartesian plots and polar plots of the curves from the families of limaçons, rhodoneas and cycloids created in the intrinsic style.

Keywords

circular motion, harmonic oscillator model, complex harmonic motion, Lissajous curves (figures), polar coordinates, sine function, cycloid, limaçon

Introduction – the definition of the sine function

How to show, how to explain the main ideas of mathematics and physics? How to avoid routine in thinking? One possibly answer is: by creating new routines (!) and using them in a trial and error approach.

The harmonic oscillator is the most powerful and most widely used single model in all of physical science. It is the basis of much of our understanding of the behaviour of molecules and electromagnetic radiation. In the harmonic oscillator model the sine function describes displacement, while the cosine – velocity, as a function of time,

The sine function and every superposition (linear combination) of different sinusoidal waves (Fourier series) are the solutions of the classical wave equation. The sine function is an important component of the solutions of quantum-mechanical wave equation (orbitals). What is the source definition of the sine function? Probably when hearing “the sine function” most people see a Cartesian plot of this function, a sinusoidal wave. But we know, that there is not one and only way to represent this function graphically. The Cartesian plot is one of many possible conventions. The source definition of the sine function is (Weisstein, 2006):

The sine function is the vertical coordinate of the arc endpoint of the unit circle rotating uniformly counter clockwise.

Does this rotation need to be counter clockwise? It depends on the initial position of the rotating point (e.g. (one could say) turtle) or, in other words we can say that the initial position of the arc endpoint (turtle drawing the unit circle) depends on the direction of the circular motion. How – see Fig.1, inspired by (Farkas, 2003)

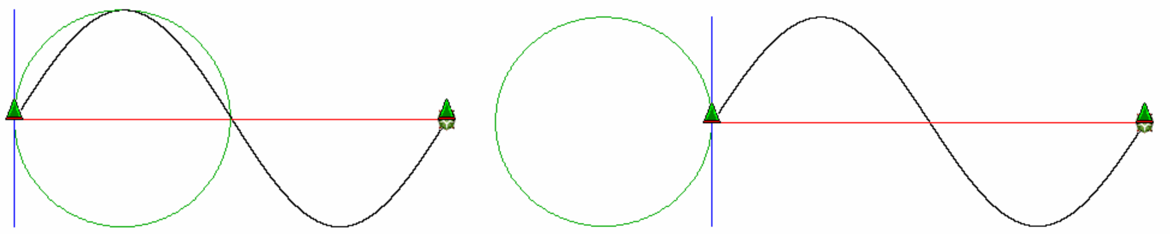


Fig.1 The plot of the sine function as a result of the origin definition: left for clockwise rotation, right – for counter clockwise one (inspired by Farkas, 2003)

The circular motion is assumed to be at a constant speed (angular speed) or alternatively with a constant frequency measured in revolutions per unit of time. It means that the angle or time is the variable. The sine curve is created when the y coordinate of the constant circular motion becomes the y coordinate of the Cartesian plot, whereas the x coordinate originates from a translation along the x axis at a constant speed related (in time) to the speed of the circular motion. The relation is established by the frequency of the circular motion, i.e. by the number of revolutions made during translation in the x direction over a distance of two diameters of the circle.

We need four turtles with the same initial position and different shapes and pen colours to draw a sine curve. The first turtle (t4) draws a circle and transfers his y coordinate to another turtle (t3), the third one (t2) goes along the x axis (i.e. the propagation direction of the wave) with a proper constant step (hx) and the last one (t1) takes the y coordinate from t4 or t3 and x coordinate from t2 and draws a sine curve as a result. Turtles t2 and t3 are redundant for drawing sine curve: they only transfer coordinates between the turtle performing the circular motion (t4) and the turtle performing the sine curve (t1), but they visualize the whole process very well.

It is convenient to assume that the circle is performed by the statement:

repeat 360 [fd :s rt 1]

where :s parameter is the arc length step corresponding to Δs and $\Delta\varphi$ is constant and equal to 1. Therefore 1/:s is a measure of the intrinsic curvature of the circle and also the angular speed of the point moving over a circular orbit. Let the sine function be defined as:

y = a sin (n x + p)

where a is the amplitude, n is the frequency and p is the phase. If n = 1, the distance of two diagonals of the circle must be completed during one revolution of a circle, for n = 2 – during two revolutions of a circle etc. The amplitude a is the radius of the circle. Let one step in translation in x direction be hx. Phase p is the initial position of the turtle drawing a circle (circular motion at a constant speed) measured in degrees. Phase 90 means that y = a cos (n x).

We can choose a, n, and p as an input for the procedure and calculate Δs and hx or we can start with Δs, n and p and calculate a and hx. The sine function is described by three independent variables. The table below shows the mutual relations between the parameters.

$a = \frac{180 \cdot \Delta s}{\pi}$	$\Delta s = \frac{\pi \cdot a}{180}$	$hx = \frac{2 \cdot \Delta s}{n \cdot \pi} = \frac{2 \cdot a}{n \cdot 180}$
90	$\frac{\pi}{2} = 1.57$	$\frac{1}{n}$
$\frac{180}{\pi} = 57.3$	1	$\frac{1}{n} \cdot \frac{2}{\pi} = 0.64$
100	$\frac{\pi \cdot 100}{180} = 1.74$	$\frac{1}{n} \cdot \frac{100}{90} = \frac{1}{n} \cdot 1.11$

Table 1. The relations between Δs, a, n and hx parameters of motion ($\Delta\varphi$ = 1)

The procedure sinecurve1 creates a sine curve with four turtles, procedure sinecurve2 – with two turtles. What will happen if we change rt into lt in these two procedures?

```
to sinecurve1 :a :n :p
 ;y = a sin (n x + p)
 make"s :a * 3.14 / 180 make "hx 2 * :s / :n / 3.14
 ;initial positions for turtles
 cs ask [t1 t2 t3 t4][pu setpos [-300 0] pd]
 ask "t4 [repeat :p [fd :s rt 1]]
 ask "t1 [pu setycor ask "t4 [ycor] pd]

 repeat 360 * :n [ask [t4] [fd :s rt 1] ask [t1 t2] [setxcor xcor + :hx]
                  ask [t1 t3] [setycor ask "t4 [ycor]] wait 10]
end
```

```
? sinecurve1 100 2 90
? ask [t1 t2 t3 t4][stamp]
```

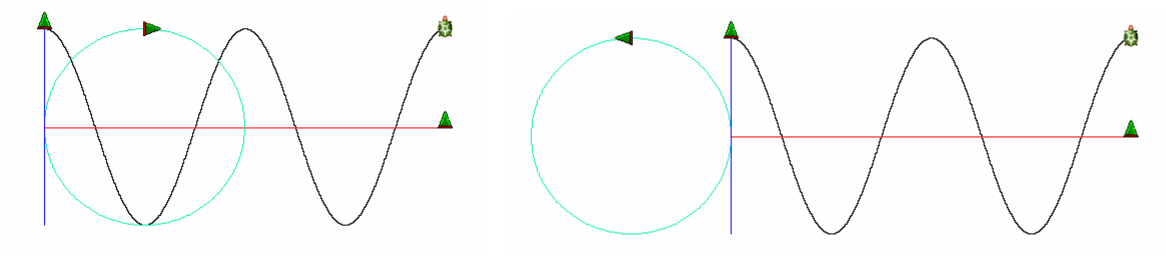


Fig.2 The cosine function with rt (left) and lt (right) created by procedure sinecurve1

```
to sinecurve2 :a :n :p
 ;without turtles t2 and t3
 ;y = a sin (n x + p)
 make"s :a * 3.14 / 180 make "hx 2 * :s / :n / 3.14
 ;initial positions for turtles
 cs ask [t1 t4][pu setpos [-300 0] pd]
 ask "t4 [repeat :p [fd :s rt 1]]
 ask "t1 [pu setycor ask "t4 [ycor] pd]

 repeat 360 * :n [ask [t4] [fd :s rt 1] ask [t1 ] [setxcor xcor + :hx setycor ask "t4 [ycor]]
                         wait 10]
end
```

```
? sinecurve2 100 2 90
? ask [t1 t4][stamp]
```

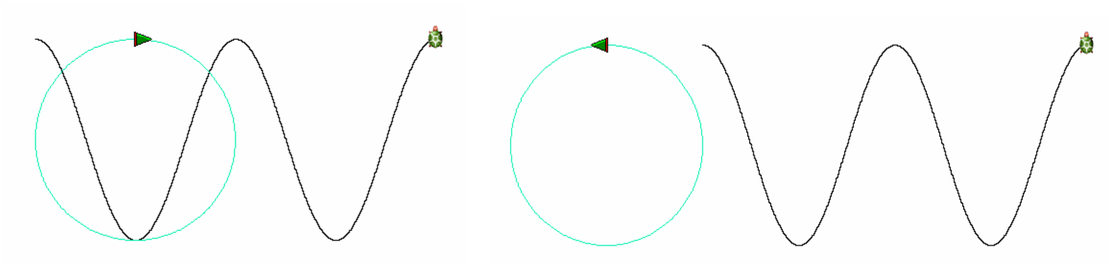


Fig.3 The cosine function with rt (left) and lt (right) created by procedure sinecurve2

Lissajous curves in the turtle graphics.

This family of curves was investigated by Nathaniel Bowditch in 1815, and later in more detail by Jules Antoine Lissajous in 1857.

Jules Antoine Lissajous (1822-1880) was a French mathematician and physicist. He was interested in waves and developed an optical method for studying vibrations. Lissajous curves have applications in physics, astronomy, and other sciences.

Some of these figures, and geometrical methods of constructing them, have been discovered independently, e.g. the Lemniscate of Gerono (eight curve) (Xah Lee, 2004).

The Lissajous figures are created by the system of parametric equations which describes two perpendicular harmonic vibrations:

$$\begin{cases} x = r_1 \sin\left(n_1 t + p\right) \\ y = r_2 \sin\left(n_2 t\right) \end{cases}$$

Appendix 1 shows how to create the Lissajous curves conventionally in Excel.

In the previous section we have checked that we understand really well how the family of sine curves could be generated from the circular motion. Do we need the sinusoid to generate the Lissajous curves, or could only two circular motions create these figures? If the latter was possible, it would be the simplest way to create the Lissajous figures.

The examples of the initial positions of the turtles are shown in Fig.4.

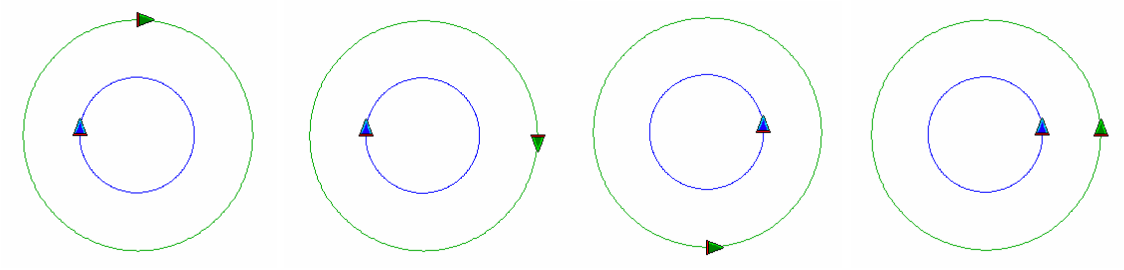


Fig. 4 The initial positions of the turtles for two concentric circles for the phase difference 0 and 90: two left pictures for rt, the next two for lt.

The time variable t, changes from the moment when the process starts to the moment when the process ends (after n_1 revolutions along first circle and n_2 revolutions along the second circle, circular motions are simultaneous). The turtle drawing the Lissajous curve takes the x coordinate from one turtle moving along the green circle and y coordinate from the turtle drawing the second, blue, circle. We can use three turtles or five turtles. In the last case these additional two turtles transfer the coordinates between those drawing the circles and the one drawing the Lissajous figure showing two perpendicular harmonic vibrations. We can also use 3 turtles and make two of them invisible: in that case we will see only one turtle drawing the Lissajous curve. The parameters of the procedure (data) could be the steps for circular motion (Δs_1 and Δs_2) or the radii (amplitudes r_1 and r_2), phase (p) and frequencies (n_1 and n_2). After some trials we will see that the shape of the L figures depends on the ratio of the frequencies of the two perpendicular waves. Dissimilar shapes can be selected by dividing the number of repetitions by the greatest common divisor of the two frequencies or by choosing only these values of frequencies whose greatest common divisor is 1.

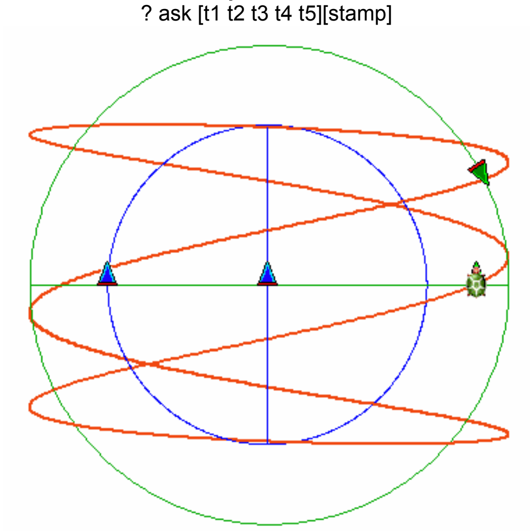
The procedures are given for rt case only. The turtles must be created with proper attributes at the beginning.

```
to turtles
 new "Turtle [penColour "green4]
 ask "t2 [setshape loadImage "|Triangle Turtle.lgf|]
 new "Turtle [penColour "blue]
 ask "t3 [setshape loadImage "|Triangle Turtleb.lgf|]
 new "Turtle [penColour "green4]
 ask "t4 [setshape loadImage "|Triangle Turtle.lgf|]
 new "Turtle [penColour "paleRed penwidth 2]
 ask "t5 [setshape loadImage "|Turtle.lgf|]
 ask "t1 [setshape loadImage "|Triangle Turtleb.lgf| setpencolour "blue]
end

to GCD :a :b
 while [:b <> 0] [let "r mod :a :b let "a :b let "b :r]
 op :a
end

to hop :dx :dy
 pu rt 90 fd :dx lt 90 fd :dy pd
end
```

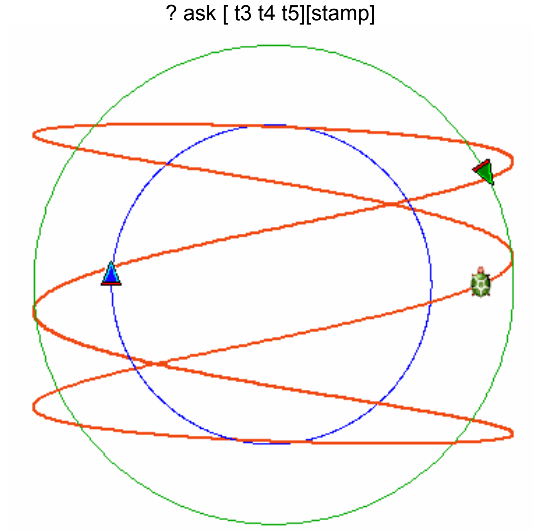
```
to Lissajous :s1 :s2 :p :n1 :n2
 cs
 ;amplitudes
 make "x 180 / 3.14
 make "r1 :s1 * :x make "r2 :s2 * :x
 ;initial position for turtles
 ask "t3 [hop -:r1 0]
 ask "t4 [hop 0 :r2 rt 90 pu repeat :p
            [fd :s2 rt 1] pd ]
 ask "t5 [pu setxcor ask "t4 [xcor]
             setycor ask "t3 [ycor] pd]

 repeat 360 * :n1 * :n2 / GCD :n1 :n2
            [ask [t3] [fd :s1 / :n2 rt 1 / :n2]
             ask [t1 t5] [setycor ask "t3 [ycor]]
             ask [t4] [fd :s2 / :n1 rt 1 / :n1]
             ask [t2 t5] [setxcor ask "t4 [xcor]]
             wait 10]
end
```

```
to Lissajous1 :s1 :s2 :p :n1 :n2
 cs
 ;without t1 and t2
 ;amplitudes
 make "x 180 / 3.14
 make "r1 :s1 * :x make "r2 :s2 * :x
 ;initial position for turtles
 ask "t3 [hop -:r1 0]
 ask "t4 [hop 0 :r2 rt 90 pu repeat :p
             [fd :s2 rt 1] pd ]
 ask "t5 [pu setxcor ask "t4 [xcor]
             setycor ask "t3 [ycor] pd]
repeat 360 * :n1 * :n2 / GCD :n1 :n2
          [ask [t3] [fd :s1 / :n2 rt 1 / :n2]
           ask [t4] [fd :s2 / :n1 rt 1 / :n1]
           ask [t5] [setxcor ask "t4 [xcor]
                     setycor ask "t3 [ycor]]
      wait 10]
end
```

? Lissajous 2 3 60 3 1
? ask [t1 t2 t3 t4 t5][stamp]



? Lissajous1 2 3 60 3 1
? ask [t3 t4 t5][stamp]



In the following demo the statement "/ GCD :n1 :n2" in the Lissajous procedure may be removed.

```
to demoLissajous
 for "s1 [2 3] [for "n1 [1 5][for "n2 (se :n1 5) [for "p [0 360 15] [if GCD :n1 :n2 = 1
 [(print [phase=] :p) (print [amplitude1=] :s1) (print [amplitude2=] 3)
 (print [frequency1=] :n1) (print [frequency2=] :n2)
 Lissajous :s1 3 :p :n1 :n2] wait 100] wait 1000]]]
end
```

The bolded part of the Lissajous procedure could have another different shape – it gives the same result but works quicker and less accurate.

repeat 360 [ask [t3] [dot fd :s1 * :n1 rt :n1] ask [t4] [dot fd :s2 * :n2 rt :n2]
ask [t5] [setxcor ask "t4 [xcor] setycor ask "t3 [ycor] dot]]

It can be improved and saved as a web project, but it is not our main purpose.

We have created a tool for investigation of the L curves and by doing it ourselves we have understood what L figures really are.

Polar coordinates and polar plot of the sine functions

What determines the choice of the coordinate system in a given situation? Why the Cartesian coordinates are not used to identify the position of our town on the globe? For the same reason we do not use the Cartesian coordinates to determine the position of an electron with respect to the atomic nucleus. The reason is symmetry. When the coordinate system suits the symmetry of the object under consideration, the equations describing it become simpler, and the calculations easier. For example: the equation of a circle of radius r centred at the pole is $r^2 = x^2 + y^2$ in the Cartesian coordinates, while in the polar coordinates it is: $r = const$. I suspect that in everyday life we use polar coordinates and their three dimensional counterpart, i.e. spherical coordinates (in geography), more frequently than the Cartesian coordinates. However, we usually do not know how simple mathematical functions look in the polar coordinates. What benefit could come from drawing them in polar coordinates? Could we understand better for example the shapes of the orbitals (one electron wave functions which are the solutions of time independent Schrödinger equation) in chemistry? Let us investigate only one family of sine functions, namely r = sin (n φ). Is there anything the students/pupils can discover in this area? I ask my students to describe in a laboratory report their findings they found particularly interesting. One of the students answered: “Honestly speaking, every subsequent function was a huge discovery, great news and a great riddle for me.” His answer encouraged me to share my didactic experience with you. Appendix 2 gives a description of this laboratory exercise, which after completion becomes the laboratory report.

For the definition of the polar coordinates see for example (Weisstein, 2002).

R is the radial distance from the origin (initial position of the turtle), φ is defined as the counter clockwise angle from the x axis, but in Logo it will be convenient to choose φ as the clockwise angle from the N-S axis (by the way: it is a good exercise to see what is the difference between the different possibilities of choosing the reference axis and the direction of counting the angle).

We offer a job to the fifth turtle (besides four introduced in the Introduction) who for subsequent values of angle φ (from 0 to 360 with the step 1) draws a line (vector) of length equal to the value of the sine function for this particular value of variable, namely equal to sin φ. The value of sin φ is equal to the value of y coordinate of the turtle rounding along the circle (and turtle oscillating along the y axis). Besides of the usual “mathematical convention” we can try also the frequently used convention of drawing the radii (vectors r) corresponding to the negative values of the function as positive and in different colour. Let us call it the “two-colour convention” (unfortunately I use two colours in every convention).

```
to turtles
 repeat 4 [new "Turtle []]
 ask [t2 t3 t4] [setshape loadImage "|Triangle Turtle.lgf|]
 ask "t1 [setPW 2]
 ask "t5 [setPW 2 setHomestate [[200 0]0] setshape loadImage "|Turtle.lgf|]
end
```

Below the procedure sinepolardot which is a good tool to demonstrate the shapes of the sine family functions in the polar coordinates in the “mathematical convention”

```
to sinepolardot :a :n :p
 ;  y = a sin (n x + p)
 let"s :a * 3.14 / 180 let "hx 2 * :s / :n / 3.14
 ; nitial positions for turtles
 cs ask [t1 t2 t3 t4][pu setpos [-300 0]pd]
 ask "t4 [repeat :p [fd :s rt 1]]
 ask "t1 [pu setycor ask "t4 [ycor] pd]
 ask [t1 t5][setPW 2 pu]

 repeat 360 * :n [let "y ask "t4 [ycor]
```

```
                    ask [t4] [fd :s rt 1]
                    ask [t1 t2] [setxcor xcor + :hx]
                    ask [t1 t3] [setycor :y]
                    ask [t1] [ifelse :y < 0 [setPC 4] [setPC 2] dot]
                    ask [t5] [ifelse :y < 0 [setPC 4] [setPC 2] fd :y dot bk :y rt 1 / :n]
                    wait 50]
end
```

? sinepolardot 90 2 0

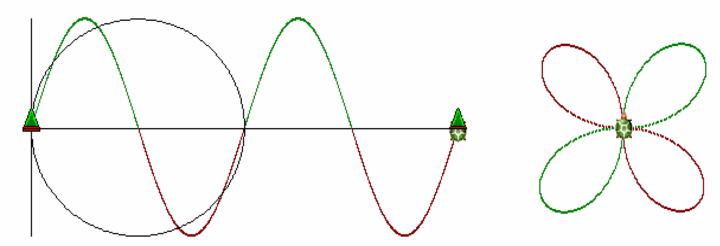


Fig. 5. The Cartesian plot of the function y = sin 2x and the polar plot of the function r = sin 2 fi

On the Fig. 5 is one example of rhodonea (Wassenaar, 2005). As before, we can remove the redundant turtles t2 and t3, as well as hide the motion of turtles t4 and t1 if we would like to see the polar plot only. To obtain the functions from the limaçon (Weistein, 2003) family we must only add the constant value to the ycor in the procedure sinepolardot (Fig. 6 and 7 are the examples).

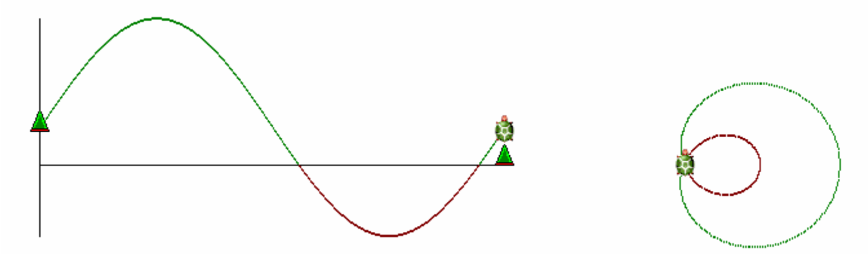


Fig. 6 The Cartesian plot of the function y = 1/3 + sin x and the polar plot of the function r = 1/3 + sin fi

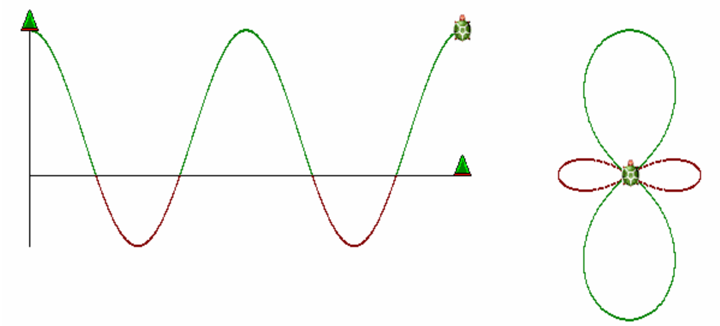


Fig. 7 The Cartesian plot of the function y = 1/3 + cos 2x and the polar plot of the function r = 1/3 + cos 2fi

The example result for fractional n and enlarged number of repetitions is shown on the Fig.8.

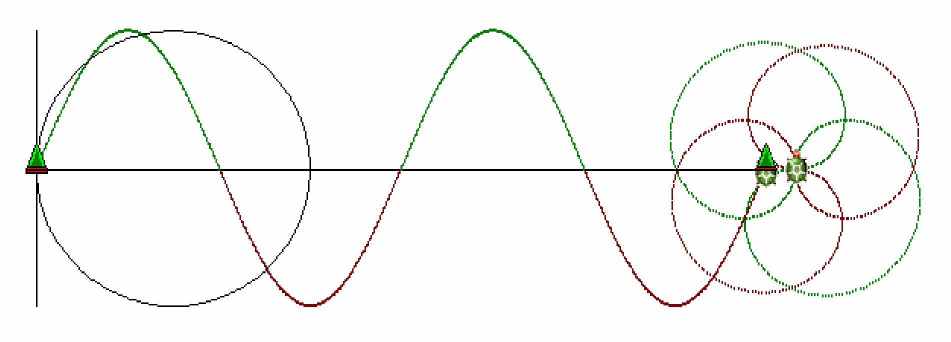


Fig. 8 The Cartesian plot of the function y = sin 2/3x and the polar plot of the function r = sin 2/3fi

Program startij demonstrates the limaçon (Wassenaar, 2004) family functions r = a + sin (i/j * phi) and shows how the shapes depend on the value of a for a given small integer values of i and j. The positive values are in red, negative in blue, the circle of radius a with black.

```
to sinpij :a :i :j
 ; r = a + sin (i/j * phi)
 cs
 repeat 360 * :i * :j / GCD :i :j [let "y 100 * (sin :i / :j * (repc - 1)) fd :a + :y
          ifelse :y < 0 [ setpc 1] [setpc 12 ] setpw 3 dot bk :y setpc 0 setpw 2 dot bk :a rt 1]
end

to startij
 pu ht
 for "i [1 5][ for "j [1 5][if GCD :i :j = 1 [(pr :i :j)
                   for "a [100 0 -5] [sinpij :a :i :j wait 10] wait 1000]
                   for "a [0 -100 -5] [sinpij :a :i :j wait 10] wait 1000]]
end
```

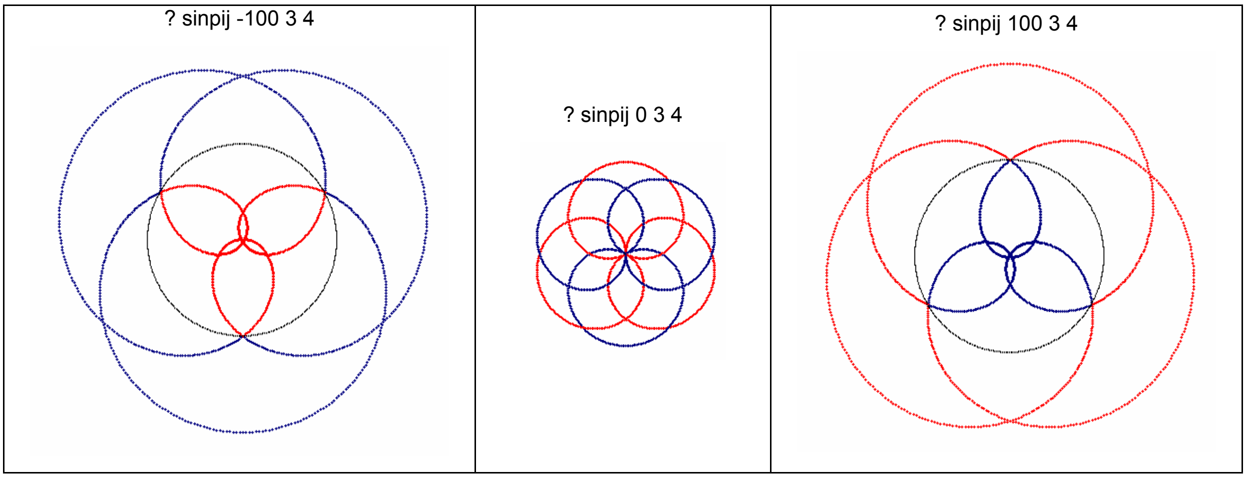


Fig. 9. The polar plots of the functions r = -1+sin3/4 fi, r=sin3/4fi and r=1+sin 3/4fi.

Superposition of waves

Now we are ready to demonstrate the Cartesian and polar superposition (interference, beat) of any two waves, each of them characterized by three parameters: amplitude, frequency and phase. We could offer a job for 12 turtles: t4 and t6 are making circular motions, t3 and t9 demonstrates one-dimensional (y direction) harmonic motions, t2 and t8 are making translation in x direction, t1 and t7 create Cartesian plots, t5 and t10 – polar plots and t11 and t12 – Cartesian and polar sums of two given waves. For clarity we can make some of the plots invisible as well as we can hide some of the turtles to show step by step how the superposition of two waves is created and to compare clearly the components and the sum in the Cartesian and polar coordinates. We can also compare the polar plots in the "mathematic" and the "two-colour" conventions.

It is a certain inconsistency to avoid, but only in the Cartesian coordinates, not in the polar ones, when the amplitudes are not the same: one could have the impression that the wave of a small amplitude propagates with a lower velocity but it is a wrong conclusion (see Fig 10, the Cartesian components and the polar sum). We can easy improve the procedure below by scaling hx value by the values of two amplitudes; hx1 ought to be of the same value as hx2.

```
? sinepolarsumdot 90 45 3 6 0 90

? ask all [stamp]
```

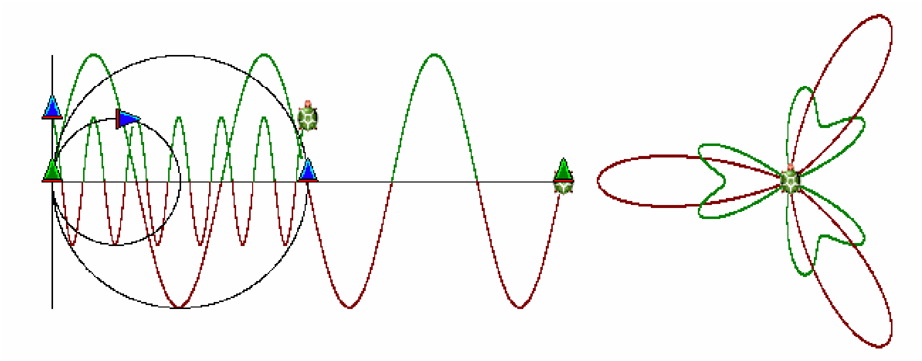


Fig. 10. The Cartesian components and the polar sum of two waves of different amplitudes

Below is one of many possible procedures, sinepolarsumdot:

```
to sinepolarsumdot3 :a1 :a2 :n1 :n2 :p1 :p2
 ; y1 = a1 sin (n1 x + p1), y2 = a2 sin (n2 x + p2)
 let"s1 :a1 * 3.14 / 180 let"s2 :a2 * 3.14 / 180
 let "hx1 2 * :s1 / (:n1 * :n2 * 3.14)
 let "hx2 2 * :s2 / (:n2 * :n1 * 3.14)
 ifelse :hx1 > :hx2 [let "hx :hx1][let "hx :hx2]
 ; nitial positions for turtles
 cs ask [t1 t2 t3 t4 t6 t7 t8 t9 t11][pu setpos [-300 0]pd]
 ask [t11 t12][st pu]
 ask [t4 t6 t8 t9][ht pu] ask [t3 t2][ht]
 ask "t4 [repeat :p1 * :n2 [fd :s1 / :n2 rt 1 / :n2]]
 ask "t6 [repeat :p2 * :n1 [fd :s2 / :n1 rt 1 / :n1]]
 ask "t1 [pu setycor ask "t4 [ycor] pd]
 ask "t7 [pu setycor ask "t6 [ycor] pd]
 ask [t1 t5 t7 t10][ht pu]

 repeat 360 * :n1 * :n2 [let "y1 ask "t4 [ycor]
                         let "y2 ask "t6 [ycor] let "y :y1 + :y2
                         ask [t4] [fd :s1 / :n2 rt 1 / :n2]
                         ask [t6] [fd :s2 / :n1 rt 1 / :n1]
                         ask [t1 t2] [setxcor xcor + :hx1]
                         ask [t1 t3] [setycor :y1]
                         ask [t7 t8] [setxcor xcor + :hx2]
                         ask [t7 t9] [setycor :y2]
                         ask [t11][setxcor xcor + :hx setycor :y]
                         ask [t1] [ifelse :y1 < 0 [setPC 4] [setPC 2] dot]
                         ask [t7] [ifelse :y2 < 0 [setPC 12] [setPC 1] dot]
```

```
                      ask [t5] [ifelse :y1 < 0 [setPC 4] [setPC 2]
                               fd :y1 dot bk :y1 rt 1 / :n1 / :n2]
                      ask [t10] [ifelse :y2 < 0 [setPC 12] [setPC 1]
                               fd :y2 dot bk :y2 rt 1 / :n1 / :n2]
                      ask [t11] [ifelse :y < 0 [setPC 13] [setPC 11] dot]
; mathematical convention
                      ask [t12] [ifelse :y < 0 [setPC 13] [setPC 11]
                               fd :y dot bk :y rt 1 / :n1 / :n2 ]
; two-colour convention
;                      ask [t12] [ifelse :y < 0 [setPC 13 bk :y dot fd :y]
;                                                  [setPC 11 fd :y dot bk :y]
;                               rt 1 / :n1 / :n2 ]
                      ]
end
```

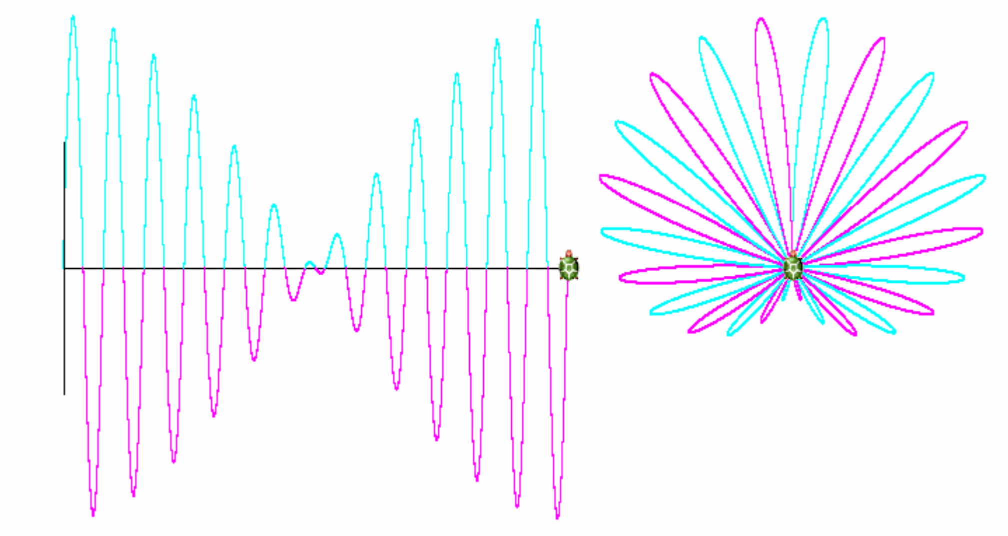


Fig. 11 The Cartesian and the polar (two-colour convention) plots of the superposition of two waves with parameters: 90 90 12 13 0 0 (beat)

Cycloids

A cycloid (Weisstein, 2003, 2004) is a plane curve that is the trajectory of a point lying on a circle that rolls without slipping along a straight line (regular cycloid) or upon another circle. A cycloid is called an epicycloids or hypocycloid, depending on whether the rolling circle has external or internal contact with the fixed circle. When the ratio of the radii of these two circles is rational, then the cycloidal curve is a closed algebraic curve.

In procedure cycx one turtle draws a regular cycloid combining two motions: circular motion (intrinsic equation of a circle, green colour) and translation in x direction (the extrinsic Cartesian coordinate system, blue colour).

```
to cycx :s
 cs pu setxy -100 0 setheading 270
 repeat 360 [dot fd :s rt 1 setxcor xcor + :s  wait 10]
 ;circuference is 360 * s, radius is 180 * s / pi
end
```

The procedure cyc draws a regular cycloid on a wholly intrinsic way and may be simplified to the procedure cyc1:

```
to cyc
 cs setPW 1 st setxy -100 0 setheading 270
 repeat 360 [pd st setPC 12 fd 1 rt 1 wait 20
             pu ht repeat repc [lt 1 bk 1]
             pd st setPC 0 bk 1 wait 20
             pu ht repeat repc [fd 1 rt 1]]
end
```

```
to cyc1 :s
 cs pu setPW 1 ht setxy -100 0 setheading 270
 repeat 360 [dot fd :s rt 1 repeat repc [lt 1] bk :s repeat repc [rt 1]]
end
```

The main procedure for hypo- and –epicycloids could be as follows:

```
to hecycs :f :i :s
;i=-1 epi, i=1 hypo, f = 1, 2, 3, 4, ...
;dfi of fixed circle=const; s (scale) (from 1 to 2)
 cs ht
 repeat 360 [pd setPC 12 fd :s rt :i * :f
        pu repeat repc [lt :i * :f bk :s]
        pd setPC 0 lt 1 bk :s
        pu repeat repc [fd :s rt :i * :f]]
end
```

where parameter f = n_1/n_2 (ratio of frequencies).

Procedure hecycs could have two interesting modifications: hecycsrt in which the direction of the rolling circle is the same as the fixed one and hecycsm in which the centre of the rolling circle moves along the circumference of the fixed one.

```
to hecycsrt :f :i :s
 cs ht
 repeat 360 [pd setPC 12 fd :s rt :i * :f
        pu repeat repc [lt :i * :f bk :s]
        pd setPC 0 fd :s rt 1
        pu repeat repc [fd :s rt :i * :f]]
end
```

```
to hecycsm :f :i :s
 cs ht
 repeat 360 [pd setPC 12 lt 90 fd :s rt :i * :f
        pu repeat repc [lt :i * :f bk :s]
        pd setPC 0 rt 90 lt 1 bk :s lt 90
        pu repeat repc [fd :s rt :i * :f] rt 90]
end
```

We can create the whole family of cycloidal curves in quite a different way. In the procedures hypoepicyc and hypoepicyc1 the parameters s1 and s2 are the steps of arc lengths (these values are directly proportional to the radiuses) of the rolling and fixed circles, parameters n1 and n2 are the frequencies of these two circular motions. In each of these procedures for the same n1 and n2, the ratio of frequencies is the same, but is obtained in a different way and in consequence the first procedure draws slower and more accurate than the second one.

```
to hypoepicyc :s1 :s2 :n1 :n2 :i
 ; hypo i = 1, epi i = -1
 cs pu
 repeat 360 * :n1 * :n2
 [setPC 12 setPW 2 dot fd :s1 / :n2 rt :i / :n2
 repeat repc [lt :i / :n2 bk :s1 / :n2]
 setPC 0 setPW 1 dot lt 1 / :n1 bk :s2 / :n1
 repeat repc [fd :s1 / :n2 rt :i / :n2] ]
end
```

```
to hypoepicyc1 :s1 :s2 :n1 :n2 :i
;faster but with low quality
 cs pu setPW 2
 ; additional with phase difference :p
 ; repeat :p / :n1 [fd :s1 * :n1 rt :i * :n1]
 repeat 360 [setPC 12 dot fd :s1 * :n1 rt :i * :n1
 repeat repc [lt :i * :n1 bk :s1 * :n1]
 setPC 0 dot lt :n2 bk :s2 * :n2
 repeat repc [fd :s1 * :n1 rt :i * :n1]]
end
```

If in the procedure hypoepicyc lt 1 / :n1 change to rt 1 / :n1 and in hypoepicyc1 lt :n2 change to rt :n2 hypocycloid change into epicycloid with the same parameters and vice versa.

The “natural” cycloids created by the procedure hecycs could be obtained by generalized procedure hypoepicyc for the data fulfilled the basic cycloidal rule:

$$\frac{s1}{s2} = \frac{r1}{r2} = \frac{n2}{n1} = \frac{1}{f}$$

For example, the results of hecycs 3 1 1, hecycs 3 1 1.5, hypoepicyc 1 3 3 1 1 and hypoepicyc .5 1.5 3 1 1 differ only with the size.

Below is the version of the hypoepicyc procedure realized with three turtles: t1 draws the cycloid, t3 draws the fixed circle, t2 is invisible and cooperates with t1 and t3 in transferring the Cartesian coordinates and the turtle heading.

```
to turtles
 repeat 2 [new "Turtle []]
 ask [t1 t2 t3] [setHomestate [[-100 0]0]]
end
```

```
to hypoepicyc2 :s1 :s2 :n1 :n2 :i
 cs ask [t1 t2 t3][pu] ask "t2 [ht]
 ask "t1 [ pd setPW 2 setPC 12] ask "t3 [setPC 0 setPW 1]

 repeat 360 * :n1 * :n2
  [ask "t2 [setpos ask "t1 [pos] setheading ask "t1 [heading] repeat repc - 1 [lt :i / :n2 bk :s1 / :n2]]
   ask "t3 [setpos ask "t2 [pos] pd lt 1 / :n1 bk :s2 / :n1 pu]
   ask "t2 [setpos ask "t3 [pos] setheading ask "t3 [heading] repeat repc - 1 [fd :s1 / :n2 rt :i / :n2]]
   ask "t1 [setpos ask "t2 [pos] setheading ask "t2 [heading] fd :s1 / :n2 rt :i / :n2]]
end
```

Let us see only one the most interesting modification: the fixed circle is additionally modified by the cycloid. Compare for example hypoepicyc2 1 3 3 1 1 with hypoepicyc3 1 2 2 1 1 and hypoepicyc3 2 1 1 2 1.

```
to hypoepicyc3 :s1 :s2 :n1 :n2 :i
 cs ask [t1 t2 t3][pu] ask "t2 [ht]
 ask "t1 [ pd setPW 2 setPC 12] ask "t3 [setPC 0 setPW 1]

 repeat 360 * :n1 * :n2
      [ask "t2 [setpos ask "t1 [pos] repeat repc - 1 [lt :i / :n2 bk :s1 / :n2]]
       ask "t3 [setpos ask "t2 [pos] pd lt 1 / :n1 bk :s2 / :n1 pu]
                                              ;or pd fd :s2 / :n1 pu rt 1 / :n1]
       ask "t2 [setpos ask "t3 [pos] repeat repc - 1 [fd :s1 / :n2 rt :i / :n2]]
       ask "t1 [setpos ask "t2 [pos] fd :s1 / :n2 rt :i / :n2]]
end
```

Cartesian and polar plots of the cycloidally modulated waves

We can ask what are the shapes of the Cartesian and the polar functions created by cycloidal motion instead of circular one. Below is one of the many possible procedures; the cycloid is drawn by turtle t4.

```
to sinepolardot3 :s1 :s2 :n1 :n2 :i
 let "hx 2 * :s2 / :n1 / :n2 / 3.14
 ;initial positions for turtles
 home ask [t1 t2 t3 t4][pu setpos [-300 0]pd]
 ;ask "t4 [repeat :p [fd :s rt 1]]
 ask "t1 [pu setycor ask "t4 [ycor] pd]
 ask [t1 t5][setPW 2 pu]
 ask [t4] [ht pu] ask "t5 [ht]

repeat 360 * :n1 * :n2 [ask [t4] [setPC 1 dot lt 90 fd :s1 / :n2 rt :i / :n2
                                     repeat repc [lt :i / :n2 bk :s1 / :n2]
```

```
                    setPC 0 dot rt 90 fd :s2 / :n1 rt 1 / :n1 lt 90
                    repeat repc [fd :s1 / :n2 rt :i / :n2] rt 90 ]
            let "y ask "t4 [ycor]
            ask [t1 t2] [setxcor xcor + :hx]
            ask [t1 t3] [setycor :y]
            ask [t1] [ifelse :y < 0 [setPC 4] [setPC 2] dot]
            ask [t5] [ifelse :y < 0 [setPC 4] [setPC 2] rt 1 / :n1 / :n2 fd :y dot bk :y]]
end
```

? cs sinepolardot3 1 2 3 1 1

? sinepolardot3 1 2 5 1 -1

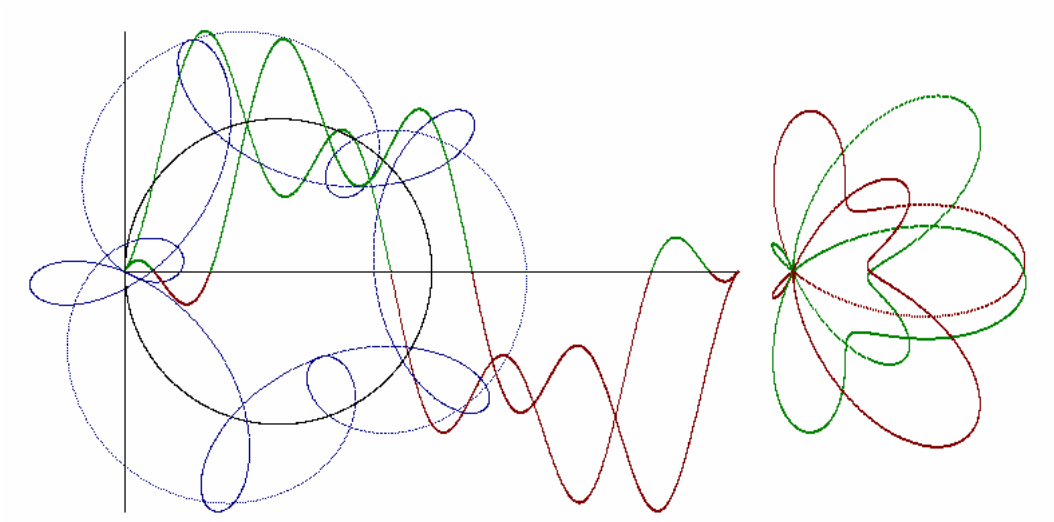


Fig. 12 Two Cartesian and polar functions created by two cycloidal motions (the centre of the rolling circle moves along the circumference of the fixed one).

Cycloidally modulated Lissajous curves

Also Lissajous curves could be modulated by cycloidal motion giving a wide variety of shapes. Below there is the procedure Lissajouscycloid, which is a modification of the procedure Lissajous1 – turtle t3 draws cycloid, turtle t4 usual circle. Turtle t5 takes xcor from t4 and ycor from t3 and draws cycloidally modified Lissajous curve.

```
to Lissajouscycloid :s1 :s2 :s3 :p :n1 :n2 :n3 :i
 cs
;s3, n3: rolling circle (t3), s1, n1: fixed circle (t3), s2, n2, p: usual circle (t4)
 ;hypo i = 1, epi i = -1
 ;amplitudes
 make "r1 180 * :s1 / 3.14 make "r2 180 * :s2 / 3.14
 ;initial position for turtles
 ask "t3 [ht hop -:r1 0 pu] ask "t4 [hop 0 :r2 rt 90 pu repeat :p [fd :s2 rt 1] pd]
 ask "t5 [pu setxcor ask "t4 [xcor] setycor ask "t3 [ycor] pd setPW 2]
 ask [t3 t4] [setPW 1] let "g GCD (GCD :n1 :n3) :n2
 repeat 360 * :n1 * :n2 * :n3  / :g [ask [t3] [dot fd :s3 / :n1 / :n2 rt :i / :n1 / :n2
                              repeat repc [lt :i / :n1 / :n2 bk :s3 / :n1 / :n2]
                              dot fd :s1 / :n2 / :n3 rt 1 / :n2 / :n3
                              repeat repc [fd :s3 / :n1 / :n2 rt :i / :n1 / :n2]]
                         ask [t4] [fd :s2 / :n1 / :n3 rt 1 / :n1 / :n3]
                         ask [t5] [setxcor ask "t4 [xcor] setycor ask "t3 [ycor]]]
end
```

? Lissajouscycloid 2 3 1 60 1 3 3 1 ? Lissajouscycloid 2 3 2 60 1 3 2 1

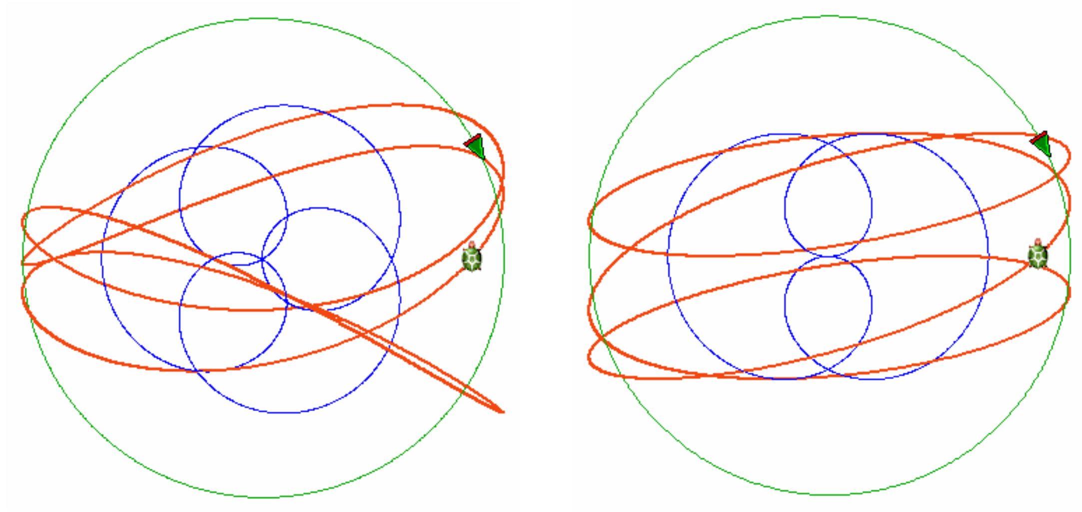


Fig. 13 Curve Lissajous 2 3 60 1 3 *cycloidally modulated..*

Comments

Playing with all these above presented programs we should not forget about two recurrent (or its iterative versions) procedures: the one creating a polar plot of a given function (see Appendix 2) and the general procedure creating a curve of the intrinsic curvature described by a given function of fi variable (Armon, 1997).

```
? cs pu
to polarec :s :fi :kfi :fun :lfi
 if :fi > :lfi [stop]
 let "x :s * run :fun
 ifelse :x < 0 [setPC 4]
                    [setPC 2] fd :x dot bk :x rt :kfi
 polarec :s :fi + :kfi :kfi :fun :lfi
end
```

```
? cs pd
to intr_graph :s :fi :kfi :fun :lfi
 if :fi > :lfi [stop]
 rt 1 let "x :s * run :fun
 ifelse :x < 0 [setPC 4]
                    [setPC 2] fd :x
 intr_graph :s :fi + :kfi :kfi :fun :lfi
end
```

Let us look at the program:

```
to startintr1
 cs ht pd
 for "a [-15 -1.1 .1] [cs intr_graph3 :a 0 1 wait 10]
wait 30
 for "a [-1 1.1 .1] [cs intr_graph1 :a 0 1 wait 30]
wait 30
 for "a [1.1 15 .1] [cs intr_graph2 :a 0 1 wait 10]
end
```

```
to intr_graph1 :a :fi :kfi
 if :fi > 1440 [stop]
 rt 1 let "x :a + (sin 3 / 4 * :fi)
 ifelse :x < 0 [setPC 4]
                    [setPC 2] fd :x
 intr_graph1 :a :fi + :kfi :kfi
end
```

```
to intr_graph2 :a :fi :kfi
 if :fi > 1440 [stop]
 rt 1 let "x (:a + (sin 3 / 4 * :fi)) / :a
 ifelse :x < 0 [setPC 4]
                    [setPC 2] fd :x
 intr_graph2 :a :fi + :kfi :kfi
end
```

```
to intr_graph3 :a :fi :kfi
 if :fi > 1440 [stop]
 rt 1 let "x (:a + (sin 3 / 4 * :fi)) / (-:a)
 ifelse :x < 0 [setPC 4]
                    [setPC 2] fd :x
 intr_graph3 :a :fi + :kfi :kfi
end
```

and compare Fig. 14 with Fig.9.

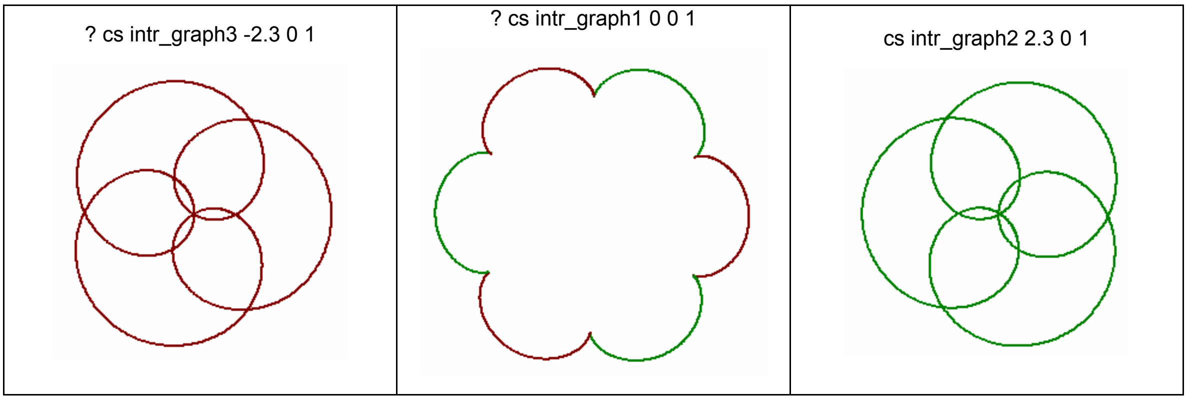


Fig. 14 The plots of the curves of the intrinsic curvature described by the functions: -2.3+sin3/4fi, sin3/4fi and 2.3+sin3/4fi

In my opinion it is worth trying to create different curves in different ways, to compare them, to classify them in a new way, to generalize them, because there are several problems connected with the intrinsic representations of mathematical curves and also connected with the physical meaning of the concept of motion, that have not been solved yet or, at least, that are not well and simply described on the educational level.

References

Uzi Armon, (1997) An algorithm that translates intrinsic equations of curves into intrinsic procedures of these curves, http://eurologo.web.elte.hu/lectures/intrins.html

Károly Farkas, (2003) Logo and native language. Intrinsic procedures of some curves, in Proceedings of Eurologo 2003, Porto, Portugal, August, pp. 69-79.

Weisstein, Eric W.(2006) "Sine." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Sine.html

http://www.hardycalculus.com/calcindex/IE_lissajous.htm

http://www-groups.dcs.st-and.ac.uk/~history/Biographies/Lissajous.html

Weisstein, Eric W. (2002) "Polar Coordinates." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/PolarCoordinates.html

Weisstein, Eric W. (2004) "Epicycloid." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Epicycloid.html

Weisstein, Eric W. (2003) "Hypocycloid." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Hypocycloid.html

Weisstein, Eric W. (2003)"Limaçon." From MathWorld--A Wolfram Web Resource. http://mathworld.wolfram.com/Limacon.html

Jan Wassenaar (2004) “Limacon” http://www.2dcurves.com/roulette/roulettel.html

Jan Wassenaar (2005) “Rhodonea” http://www.2dcurves.com/roulette/rouletter.html

Xah Lee (2004) „Epicycloid and hypocycloid“ http://xahlee.org/SpecialPlaneCurves_dir/EpiHypocycloid_dir/epiHypocycloid.html

Xah Lee (1998) „Limacon of Pascal“ http://xahlee.org/SpecialPlaneCurves_dir/LimaconOfPascal_dir/limaconOfPascal.html

Xah Lee (2004) „Lemniscate of Gerono“ http://xahlee.org/SpecialPlaneCurves_dir/LemniscateOfGerono_dir/lemniscateOfGerono.html

Appendix 1 Lissajous figures created conventionally in Excel

The tool: Excel

The definition: by the system of parametric equations which describes complex harmonic motion:

$$\begin{cases} x = A\sin(\alpha t + \delta) \\ y = B\sin(\beta t) \end{cases}$$

	A	B	C	D	E	F	G	H	I
1	t	Asin(alpha* t+D)	Bsin(beta* t)	tp =	-3,141593	A =	1		
2	-3,141593	-0,258819045	2,4503E-16	tk =	3,141593	alpha =	3		
3	-3,078761	-0,435231099	0,125333234	ht =	0,062832	B =	1		
4	-3,015929	-0,596224875	0,248689887			beta =	2		
5	-2,953097	-0,736097087	0,368124553			D =	0,262	15	
6	-2,890265	-0,849892693	0,481753674						
7	-2,827433	-0,933580426	0,587785252						
8	-2,764602	-0,984195608	0,684547106						
9	-2,70177	-0,999945169	0,770513243						
10	-2,638938	-0,980271175	0,844327926						

Table 1 Part of the table presenting numbers

	A	B	C	D	E	F	G	H	I
1	t	Asin(alpha* t+D)	Bsin(beta* t)	tp =	=-PI()	A =	1		
2	=tp	=A*SIN(alpha*$A2 + D)	=B*SIN(betha*$A2)	tk =	=PI()	alpha =	3		
3	=A2+ht	=A*SIN(alpha*$A3 + D)	=B*SIN(betha*$A3)	ht =	=(E2-E1)/100	B =	1		
4	=A3+ht	=A*SIN(alpha*$A4 + D)	=B*SIN(betha*$A4)			beta =	2		
5	=A4+ht	=A*SIN(alpha*$A5 + D)	=B*SIN(betha*$A5)			D =	=H5*PI()/180	15	
6	=A5+ht	=A*SIN(alpha*$A6 + D)	=B*SIN(betha*$A6)						
7	=A6+ht	=A*SIN(alpha*$A7 + D)	=B*SIN(betha*$A7)						
8	=A7+ht	=A*SIN(alpha*$A8 + D)	=B*SIN(betha*$A8)						
9	=A8+ht	=A*SIN(alpha*$A9 + D)	=B*SIN(betha*$A9)						
10	=A9+ht	=A*SIN(alpha*$A10 + D)	=B*SIN(betha*$A10)						

Table 2. Part of the table presenting formulas

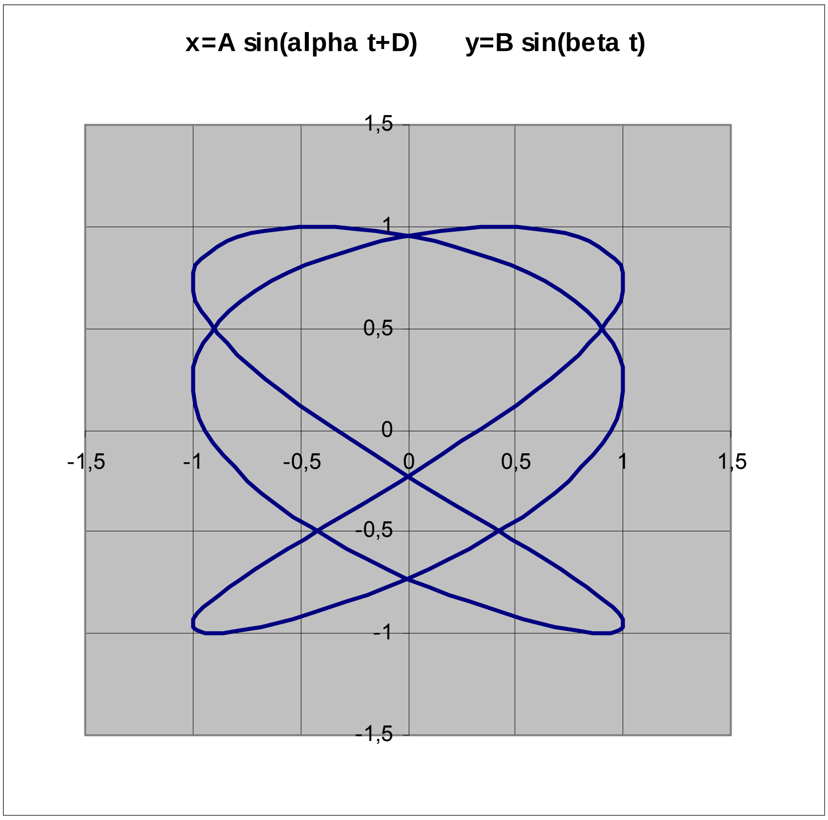
The worksheet data selected for the plot: B1:C102.

Spinner buttons: select View/Toolbars/Forms, create the Spinner button. Right-click the Spinner, then click Format Control and enter the following information: Minimum value, Maximum value, Incremental change and Cell link.

Spinner button A: 1 10 1 G2

Spinner button B: 1 10 1 G4

Spinner button D: 0 360 15 H5


x=A sin(alpha t+D)   y=B sin(beta t)
1,5
1
0,5
0
-0,5
-1
-1,5
-1,5
-1
-0,5
0
0,5
1
1,5

Appendix 2 Polar coordinates and polar plots. Sine function family.

Full name:

Read carefully description of the exercise and then write down or paste answers for questions with red pen color.

1) Introduction

The exercise is meant as a step, for some students perhaps the first one, towards familiarisation with polar plots of mathematical functions. Understanding of the polar plots is necessary to grasp the meaning of orbitals in chemistry and to understand what the geographic coordinates are. The choice of the optimum coordination system to describe a given phenomenon is determined by symmetry. The polar and spherical plots (spherical plots are the three-dimensional correspondents of the polar ones) are more suitable to solve the problems of spherical symmetry (the atom, the globe) than the Cartesian ones. In the real life situations we more often use the polar then the Cartesian coordinates, whether we are aware of it or not. Unfortunately, in the high school curricula the Cartesian coordinates are prevalent, so familiarisation with the polar coordinates is much beneficial.

Let's begin with the simplest example of the equation of a circle of a given radius r in the polar coordinates; it is

$$r = \text{const.}$$

The most illustrative examples of polar functions are spirals of different types and trigonometric functions:

$$r = \sin (n\, \varphi)$$

$$r = \cos (n\, \varphi)$$

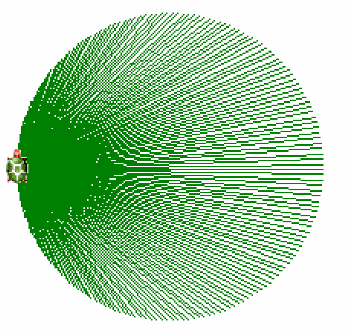
for integer and fractional values of parameter n.

The polar plots are made by drawing a vector of a value equal the value of the function (r) calculated for the angle (φ) made by the vector with a distinguished direction. It is worth mentioning here that the distinguished direction can be that of the Cartesian axis y (analogy to the north-south direction N-S) or the direction of the x axis. Moreover, the direction with respect to which the angle φ increases must be specified. In LOGO it would be the most natural to distinguish the direction N-S – the initial direction of the turtle movement. We assume that the angle increases in the clockwise direction (sometimes the opposite convention is assumed, e.g. in some dialects of the Lisp language).

Even the simplest iteration program for drawing the polar plot of the function r = sin φ brings out the problems we will have to solve.

```
to sinp0
  cs st setPC 2
  repeat 360 [let „r 200 * sin repc fd :r bk :r rt 1]
end
```

? sinp0



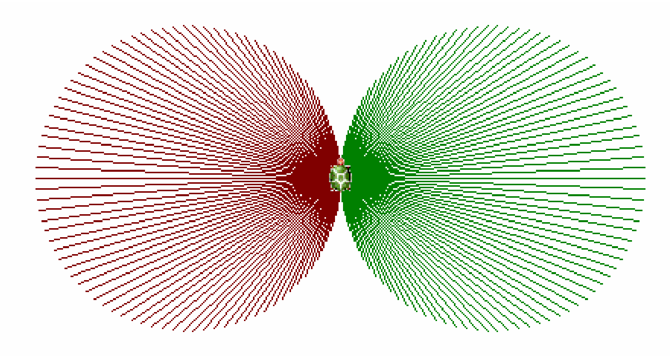
The angle φ varies from 1 to 360 at a step 1; 200 pixels correspond to the function value of 1, so the number of 200 is used for scaling of the plot. The first remark is that it would be preferable to have the angle φ varying from 0 to 360. As the procedure parameters we would like to have: the scaling factor, step of changes in the angle φ (it does not have to be 1), the initial value of the angle and certainly the final value of angle φ. There is another problem – when the function assumes negative values the turtle moves back instead of forward (the sign of the vector r changes) and the fragment of the plot for the negative values of the function r = sin φ overlaps the fragment for the positive values of the function, which gives a misleading impression that these two fragments are the same. Let's call this usually used convention as the "mathematical convention" To deal with this problem we can accept the frequently used convention of drawing the radii (vectors r) corresponding to the negative values of the function as positive and in different colour. Let us call it the "two-colour convention" (unfortunately we are using two colours in every convention!). The above example in the "two-coloured convention", still without parameters but for the angle φ varying from 0 to 360 at a step 0.1 looks like this:

```
to sinp
 cs st
 repeat 3601 [let "r 200 * sin 0.1 * (repc - 1) ifelse :r < 0 [setPC 4 bk :r fd :r ]
                                                              [setPC 2 fd :r bk :r ]
                                                               rt 0.1]
end
```

The same but with the parameters and assuming that the function and the variation range of the angle φ are the same and only the step of its changes is different:

```
to sinp1 :s :kfi
 cs st
 repeat 360 / :kfi + 1 [let „r :s * sin :kfi * (repc - 1)
                        ifelse :r < 0 [setPC 4 bk :r fd :r ]
                                      [setPC 2 fd :r bk :r ]
                        rt :kfi]
end
```

? sinp1 200 2



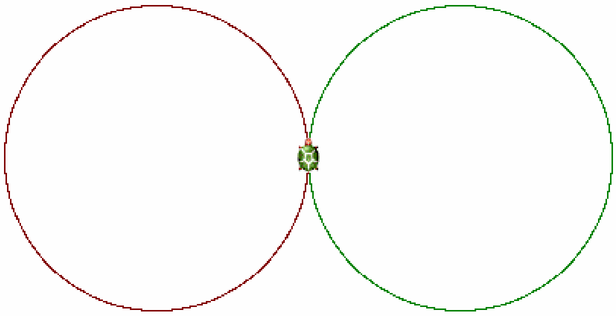
Additionally with the initial value of fi and the function as a parameters:

```
to polar1 :s :fi :kfi :fun
 cs st pd
 repeat 360 / :kfi + 1 [let "r :s * run :fun
                        ifelse :r < 0 [setPC 4 bk :r fd :r] [setPC 2 fd :r bk :r]
                        rt :kfi let "fi :fi + :kfi]
end
```

? polar1 200 0 1 [sin :fi]

As follows from the above, we have achieved a high degree of generality of the procedure.

2) Now, it would be more convenient to use the loop **while** instead of the **repeat** and let it be the first task for students. Call the procedure **polar2**. We assume that the angle fi always varies from 0 to 360 degrees. Paste the working procedure.
3) Modify the last procedure so that only the curves were drawn, not the radius-vectors to particular points of the curve. In Logo there is the primary procedure **dot** working when pen is up (**pu**). Paste the working procedure.



Below there is the recurrent way of drawing a circle in the polar coordinates.

```
to polarcircle :r :fi :kfi
 if :fi > 360 [stop]
 fd :r bk :r rt :kfi
 polarcircle :r :fi + :kfi :kfi
end
```

? cs polarcircle 200 0 1

The procedure permits drawing a circle of the radius of 200 pixels for the angle fi varying from 0 to 360 at a step 1.

4) Write in Imagine the recurrent version of the procedure polar2 (**polarek**). Tested procedure paste below.
5) Using the recurrent program you wrote draw the polar plots of the function
   r = sin (n_1/n_2 φ)
   for different integer values of n_1 and n_2 (for n = n_1/n_2 > 1 as well as n < 1). In the second case it is convenient to assume the final value of the angle being a multiple of 360 degree. Introduce additional parameters if needed. Paste the commands with an explanation saying which function is drawn, in which scale, what is the range of the argument and at which step it changes. An example of such a description:

   ? polar2 200 0 0.1 [sin :fi]
   draws the function sin fi, the unit value of the function corresponds to 200 pixels, the function is drawn for the angle varying from 0 to 360 degrees at a step 0.1

   Compare the polar plots with the Cartesian ones (Excel).
   Please write down these of yours discoveries, which turned to be especially interesting for you.