A **Spirograph** is a geometric drawing toy that produces mathematical roulette curves of the variety technically known as **Hypotrochoids** and **Epitrochoids**.
When moving circle is inside, it's called Hypotrochoid
When moving circle is outside, it's called Epitrochoid
The difference between two is sign for radius for moving circle.

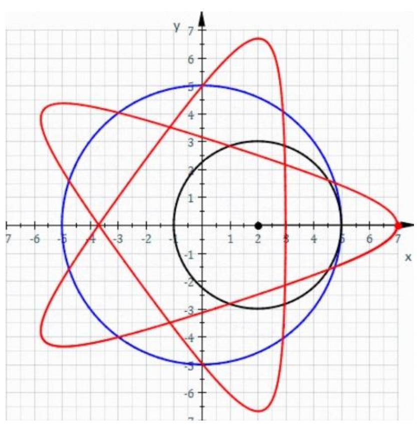
Hypotrochoids / Hypocycloid (Special type of Hypotrochoid where d==r)

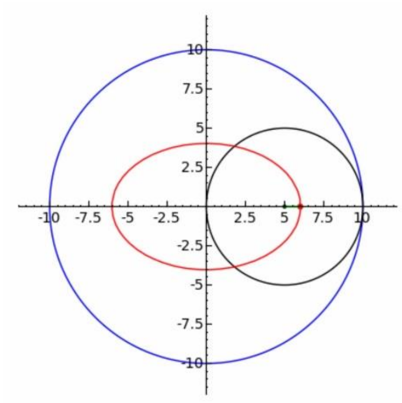
A hypotrochoid is a roulette traced by a point attached to a circle of radius r rolling around the inside of a fixed circle of radius R, where the point is a distance d from the center of the interior circle.
The parametric equations for a hypotrochoid are:

$$x(\theta) = (R-r)\cos\theta + d\cos\left(\frac{R-r}{r}\theta\right)$$
$$y(\theta) = (R-r)\sin\theta - d\sin\left(\frac{R-r}{r}\theta\right)$$

where θ is the angle formed by the horizontal and the center of the rolling circle (these are not polar equations because θ is not the polar angle). When measured in radian, θ takes values from 0 to $2\pi \times \frac{LCM(r,R)}{R}$ where LCM is least common multiple.

Special cases include the hypocycloid with $d = r$ is a line or flat ellipse and the ellipse with $R = 2r$ and d > r or d < r (d is not equal to r). (see Tusi couple).





Epitrochoids / Epicycloid (Special type of Epitrochoids where d==r**)**

An epitrochoid (/ɛpɪˈtrɒkɔɪd/ or /ɛpɪˈtroʊkɔɪd/) is a roulette traced by a point attached to a circle of radius r rolling around the outside of a fixed circle of radius R, where the point is at a distance d from the center of the exterior circle.

The parametric equations for an epitrochoid are

$$x(\theta) = (R + r)\cos\theta - d\cos\left(\frac{R+r}{r}\theta\right),$$

$$y(\theta) = (R + r)\sin\theta - d\sin\left(\frac{R+r}{r}\theta\right).$$

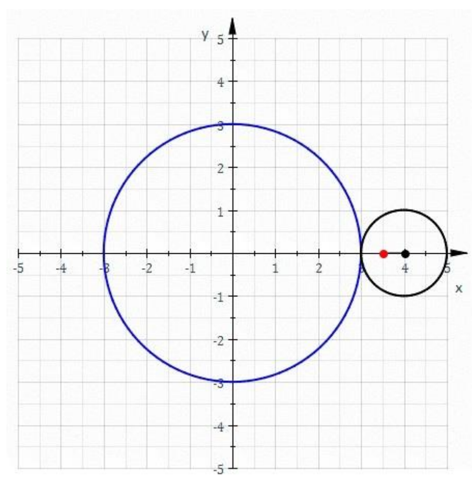
where θ is a parameter (not the polar angle).

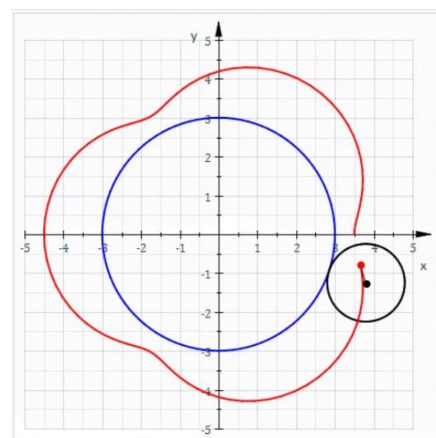
Special cases include the limaçon with $R = r$ and the epicycloid with $d = r$.

The classic Spirograph toy traces out epitrochoid and hypotrochoid curves.

The orbits of planets in the once popular geocentric Ptolemaic system are epitrochoids.

The combustion chamber of the Wankel engine is an epitrochoid.





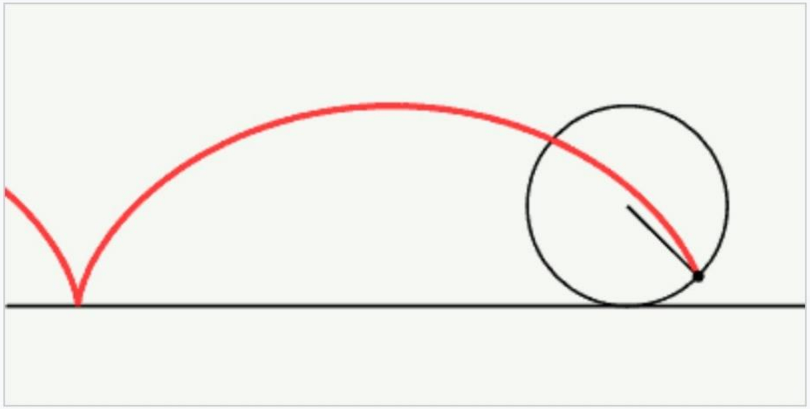
The epitrochoid with $R = 3$, $r = 1$ and $d = 1/2$

Cycloid

A cycloid is the curve traced by a point on the rim of a circular wheel as the wheel rolls along a straight line without slipping.

A cycloid is the curve traced by a point on the rim of a circular wheel as the wheel rolls along a straight line without slipping. A cycloid is a specific form of trochoid and is an example of a roulette, a curve generated by a curve rolling on another curve.

The cycloid, with the cusps pointing upward, is the curve of fastest descent under constant gravity, and is also the form of a curve for which the period of an object in descent on the curve does not depend on the object's starting position.



A cycloid generated by a rolling circle

Cyclogon

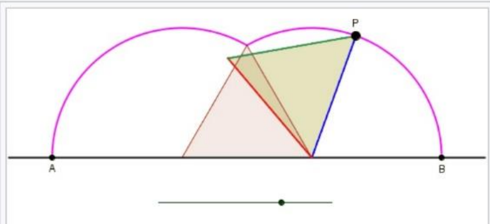
In mathematics, in geometry, a cyclogon is the curve traced by a vertex of a polygon that rolls without slipping along a straight line. There are no restrictions on the nature of the polygon. It can be a regular polygon like an equilateral triangle or a square. The polygon need not even be convex: it could even be a star-shaped polygon. More generally, the curves traced by points other than vertices have also been considered. In such cases it would be assumed that the tracing point is rigidly attached to the polygon. If the tracing point is located outside the polygon, then the curve is called a prolate cyclogon, and if it lies inside the polygon it is called a curtate cyclogon.

In the limit, as the number of sides increases to infinity, the cyclogon becomes a cycloid.

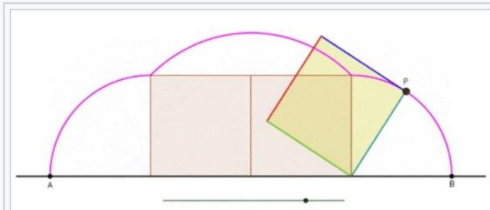
The cyclogon has an interesting property regarding its area. [3] Let A denote the area of the region above the line and below one of the arches, let P denote the area of the rolling polygon, and let C denote the area of the disk that circumscribes the polygon. For every cyclogon generated by a regular polygon,

$A = P + 2C$

Cyclogons generated by an equilateral triangle and a square

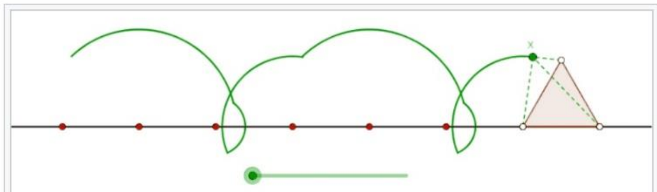


Animation showing the generation of one arch of a cyclogon by an equilateral triangle as the triangle rolls over a straight line without slipping.



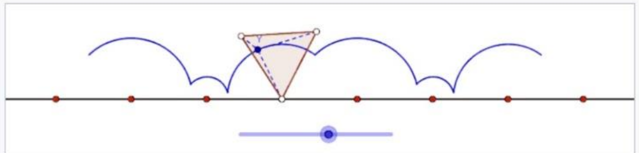
Animation showing the generation of one arch of a cyclogon by a square as the square rolls over a straight line without slipping.

Prolate cyclogon generated by an equilateral triangle



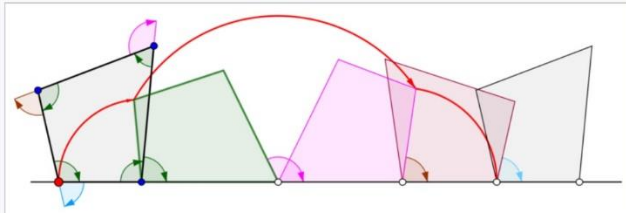
Animation showing the tracing of a prolate cyclogon as an equilateral triangle rolls over a straight line without skipping. The tracing point X is outside the disk of the triangle. (Click on the image to see the animation.)

Curtate cyclogon generated by an equilateral triangle

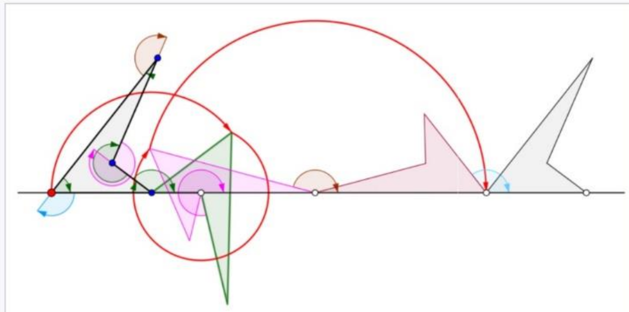


Animation showing the tracing of a curtate cyclogon as an equilateral triangle rolls over a straight line without skipping. The tracing point Y is inside the disk of the triangle. (Click on the image to see the animation.)

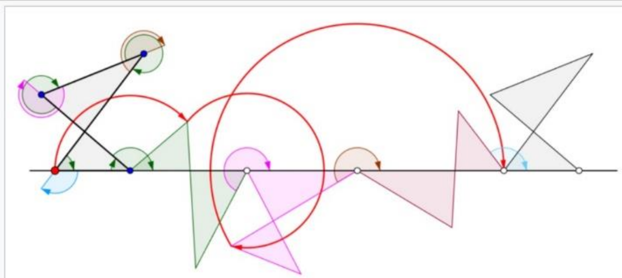
Cyclogons generated by quadrilaterals



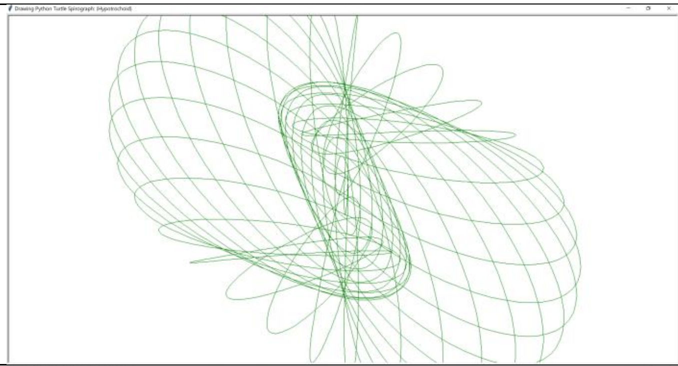
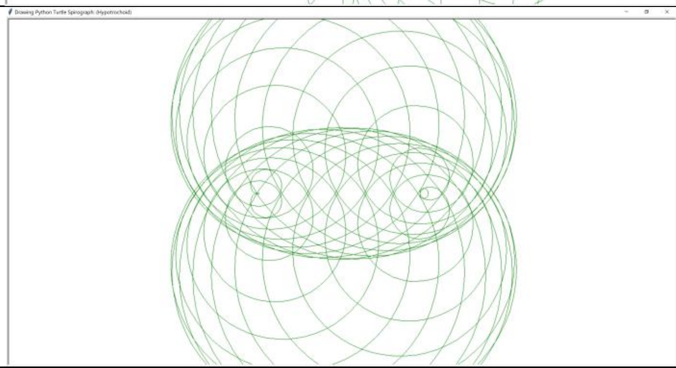
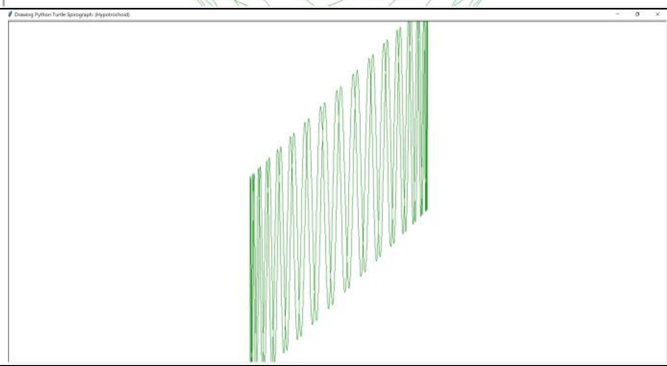
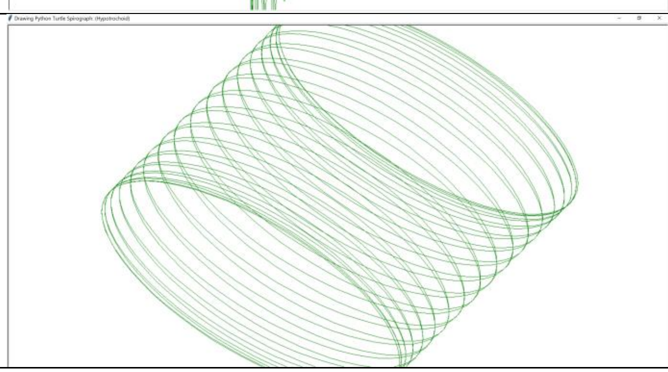
Cyclogon generated by a convex quadrilateral

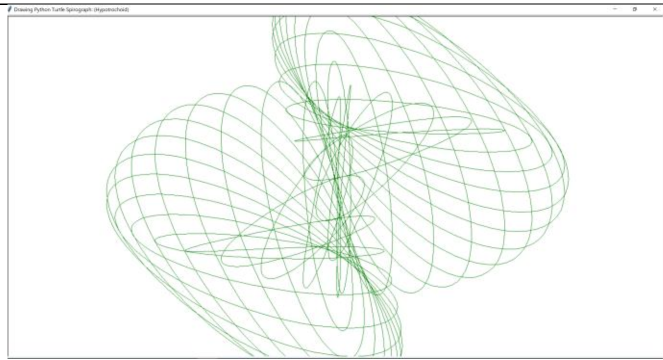
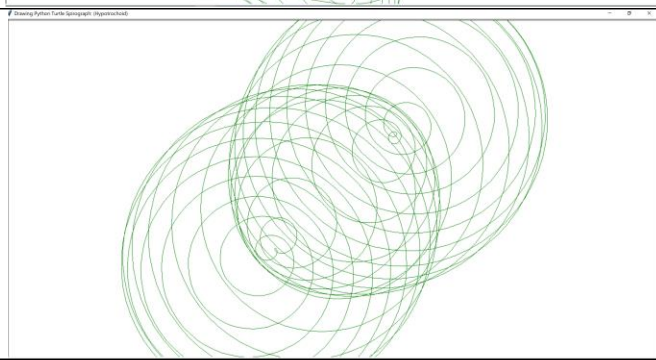


Cyclogon generated by a non-convex quadrilateral



Cyclogon generated by a star-like quadrilateral

	x = (R-r)*math.cos(Radian)+ d * math.cos ((R - r) / r * Radian) y = (R-r)*math.sin(Radian)- d * math.cos ((R - r) / r * Radian) x = (R-r)*math.cos(Radian)+ d * math.cos ((R - r) / r * Radian) y = (R-r)*math.sin(Radian)- d * math.cos ((R - r) / r * Radian)
	x = (R-r)*math.cos(Radian)- d * math.cos((R - r) / r * Radian) y = (R-r)*math.sin(Radian)- d * math.sin((R - r) / r * Radian) x = (R-r)*math.cos(Radian)- d * math.cos((R - r) / r * Radian) y = (R-r)*math.sin(Radian)- d * math.sin((R - r) / r * Radian)
	x = (R+r)*math.cos(Radian)- d * math.cos((R + r) / r * Radian) y = (R+r)*math.cos(Radian)- d * math.cos((R + r) / r * Radian) x = (R+r)*math.cos(Radian)- d * math.cos((R + r) / r * Radian) y = (R+r)*math.cos(Radian)- d * math.cos((R + r) / r * Radian)
	x = (R+r)*math.cos(Radian)+ d * math.cos((R + r) / r * Radian) y = (R+r)*math.cos(Radian)- d * math.cos((R + r) / r * Radian) x = (R+r)*math.cos(Radian)+ d * math.cos((R + r) / r * Radian) y = (R+r)*math.cos(Radian)- d * math.cos((R + r) / r * Radian)

	x = (R-r)*math.cos(Radian)+ d * math.cos((R - r) / r * Radian) y = (R-r)*math.cos(Radian)- d * math.cos((R - r) / r * Radian) x = (R-r)*math.cos(Radian)+ d * math.cos((R - r) / r * Radian) y = (R-r)*math.cos(Radian)- d * math.cos((R - r) / r * Radian)
	x = (R-r)*math.cos(Radian)+ d * math.sin((R - r) / r * Radian) y = (R-r)*math.cos(Radian)- d * math.cos((R - r) / r * Radian) x = (R-r)*math.cos(Radian)+ d * math.sin((R - r) / r * Radian) y = (R-r)*math.cos(Radian)- d * math.cos((R - r) / r * Radian)