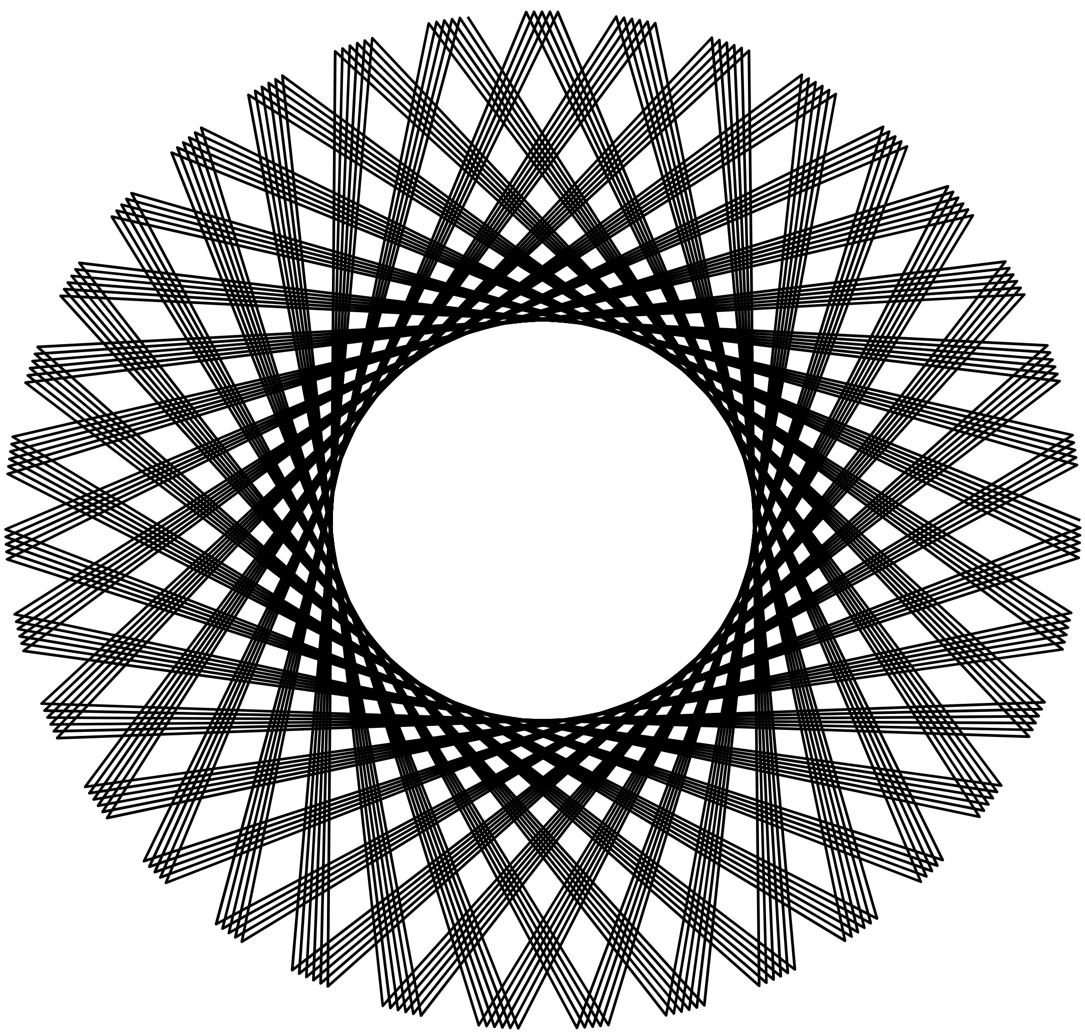
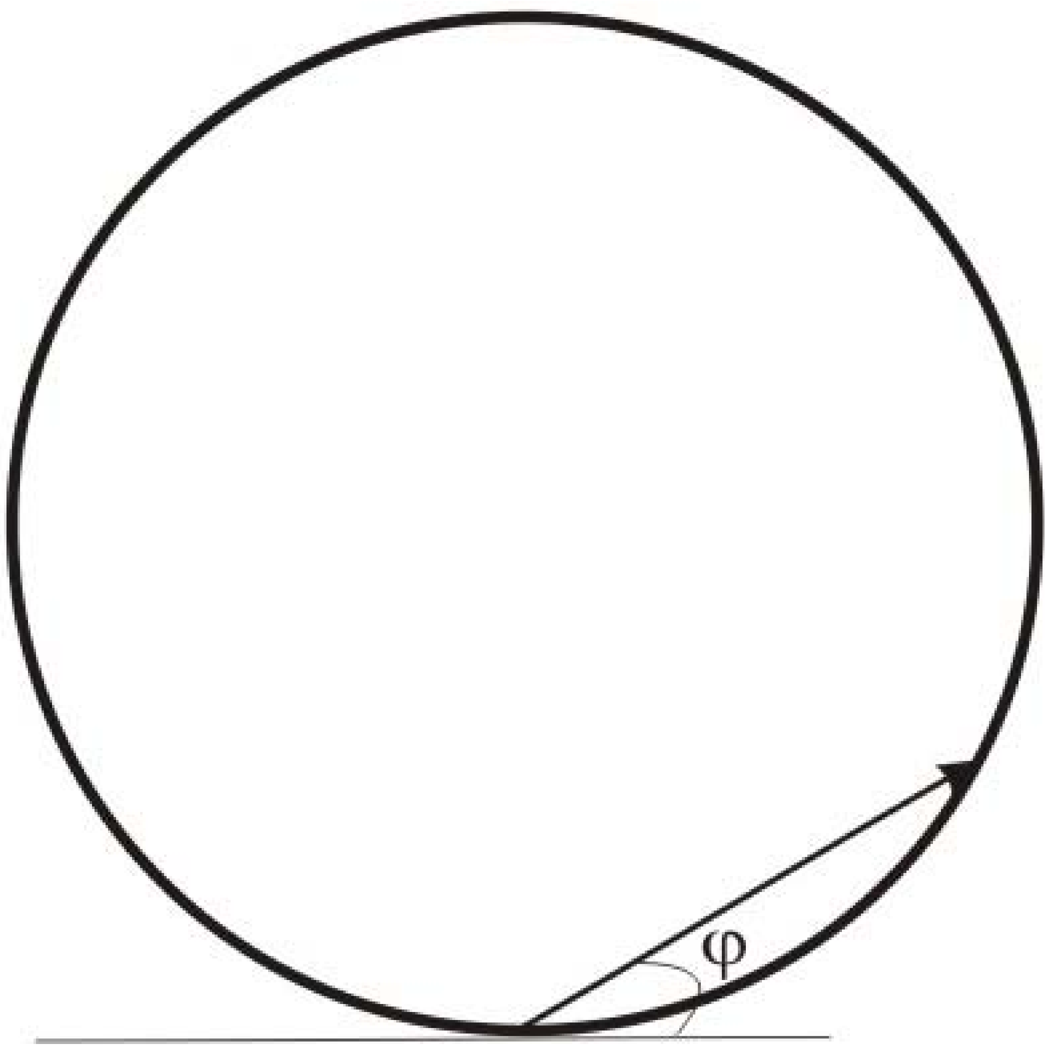
3: ***Alternate (Simpler Physics) Exercise Demonstrating Round-Off Error;***

Endless Reflections in a Circular Mirror

Note: It is hard to find a simple physics problem that illustrates round-off error in a function evaluation. The problem in the text dealing with special relativity is based on physics that some students (or their instuructors) may feel uncomfortable with since they have not covered it in class. In addition, the error is small because there is no chance for it to accumulate over multiple steps. Our view is that the problem is good because it requires only the simple evaluation of some algebraic formulas, but not necessarily an understanding of their derivation, the point being that level of precision can affect the final answer. However, for those who prefer simpler physics, a larger and more graphical effect, we here present an alternative problem. The price paid for these improvements is that we must employ some Maple commands and constructs that the students will not study until later. If you can think of a better example, please let me know!   RHL

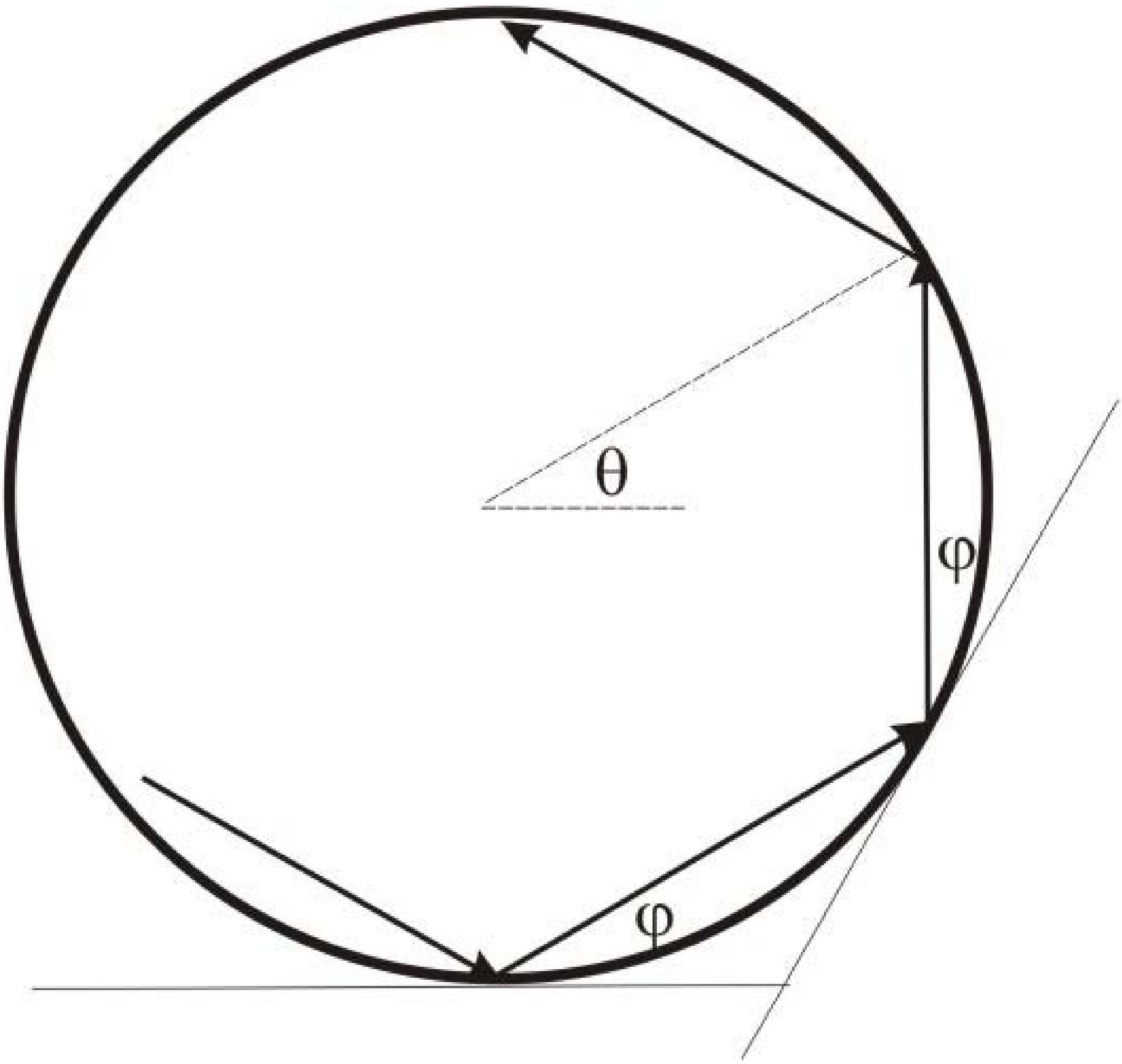


Problem: A ray of light is introduced into the interior of a circular mirror that reflects the light around its interior. As seen in the figure below, the ray is started at an angle ψ with respect to a horizontal tangent. Your problem is to determine the path followed by this ray for a perfectly reflecting mirror. In addition, you are to determine how the computed path changes if the calculation is done with only four places of precision.



Theory

For a perfectly reflecting surface, the basic law of optics is that the angle of incidence equals the angle of reflection. We indicate this in the figure below. Since no light is absorbed by the surface, the ray continues to reflect endlessly, but may, or may not, follow the same path around the circle each time. With an origin placed at the center of the circle, the location of the ray within the mirror is given by the angle θ.



For an initial angle $\psi < \pi$, geometry tells us that after each reflection the angle θ increases by 2ψ :

$$\theta_{new} = \theta_{old} + 2\,\psi$$

Although this may appear to indicate that θ increases endlessly, the addition or subtraction of 2π to θ does not change the location on the circle, and so it is possible to have the light ray's path fall on top of itself. Furthermore,

if $\dfrac{\psi}{\pi}$ is a rational number $\left(\text{the ratio of two integers } \dfrac{n}{m}\right)$,

$$\frac{\psi}{\pi} = \frac{n}{m} \qquad (\textit{closed trajectory})$$

then the ray will fall upon itself and form a geometric figure (like a Spirograph).

Implementation

We would like to have Maple calculate and plot these orbits for arbitrary values of the initial angle ψ. Also, we need to determine whether it makes a difference in our plots if the initial angle is represented by a number with only four places of precision. (We can do the calculation exactly in Maple if we do it algebraically, however since we want to plot up the results we must use floating point numbers, and then precision matters.)

We start by clearing the slate and by loading the packages needed to make plots:

```
> restart: with(plots): with(plottools):
                          # Initialization
```

```
> psi :=  Pi/3;                                  # Set
  the initial angle as Pi/3
```

Now that we have set the ray's initial angle, let us set a value for the variable θ that represents a location on the circular mirror :

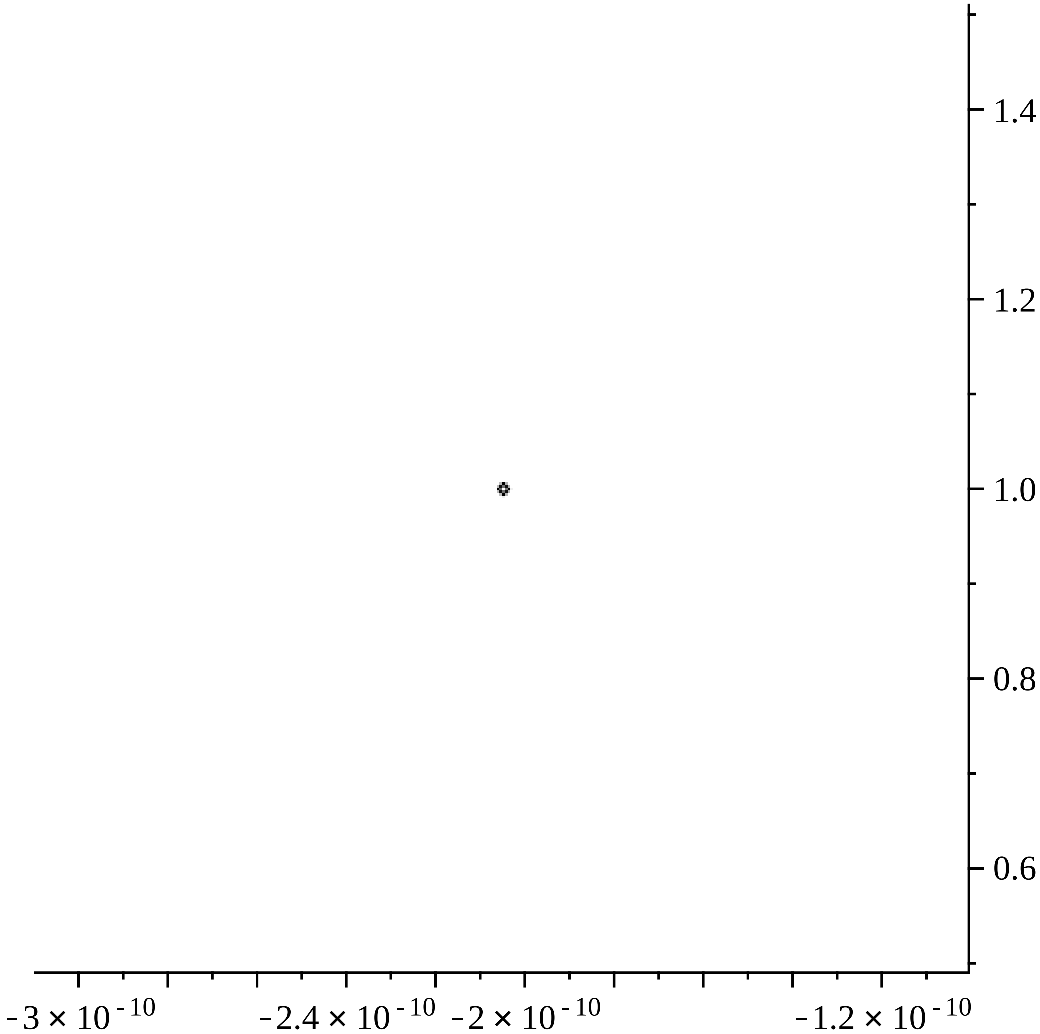
$$\psi := \frac{1}{3}\pi \tag{1}$$

```
> theta := Pi/2;
  # Location on circle
```

Well that's not too exciting. So let's make a graph that plots this point. Although we have yet to introduce graphics, we will use the command to plot a point in space, and ask that you just use it without further explanation. Since we are dealing with a circular geometry, it makes sense to use polar coordinates, and we do that by including the *coords* = *polar* option. We indicate the radius of the mirror equals 1 (since no size was given) and the value for θ as a list in square brackets $[1, \theta]$:

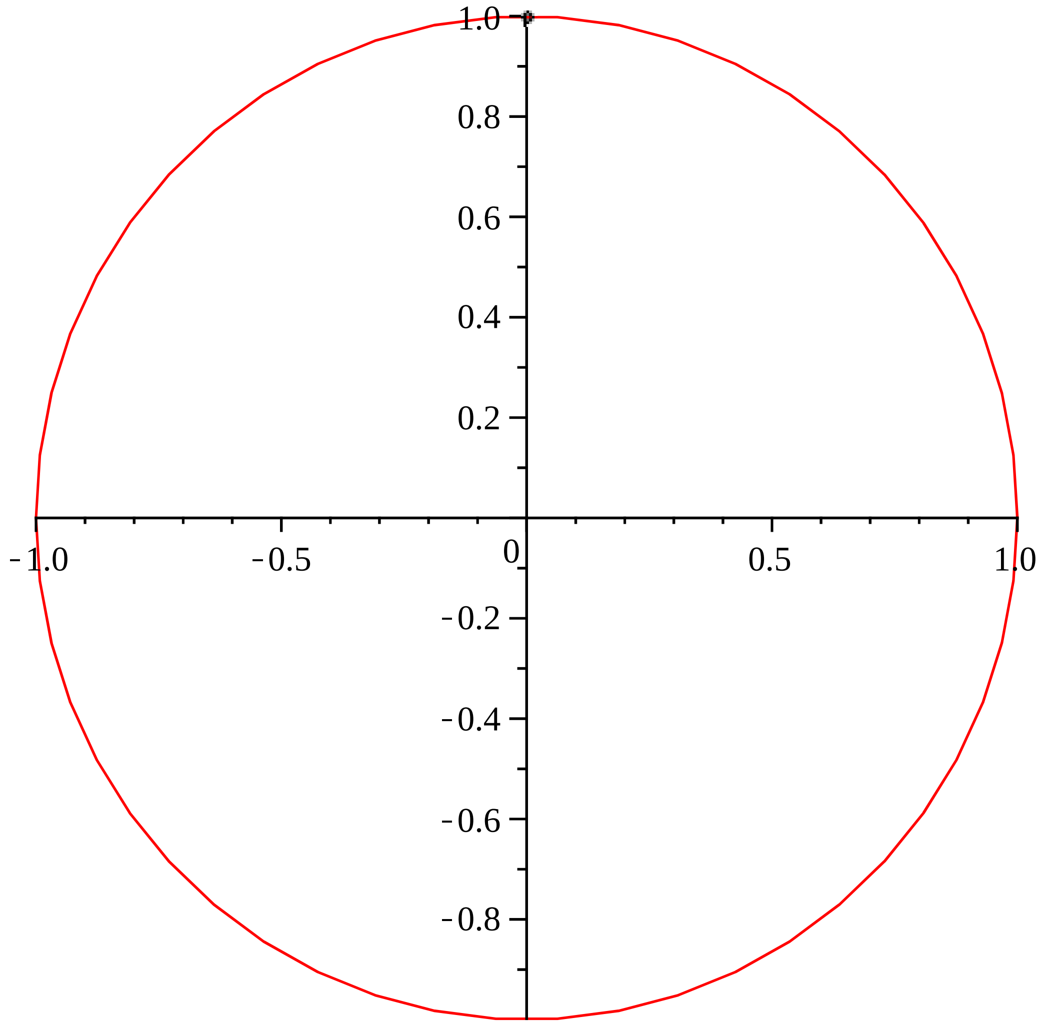
$$\theta := \frac{1}{2}\pi \tag{2}$$

```
> pointplot([1, theta], coords = polar);
                                           #` `Plot a point
```



OK, so now we have a point at the top of the circle, except we do not have any circle. So we use Maple's *circle(color=red)* command to draw a red circle. However, we must display the *pointplot* and the circle together on the same graph, and to do that we use the *display* command:

```
> display(circle(color = red),   pointplot([1, theta],  coords = polar) );
       #  Display circle and plot on same graph
```



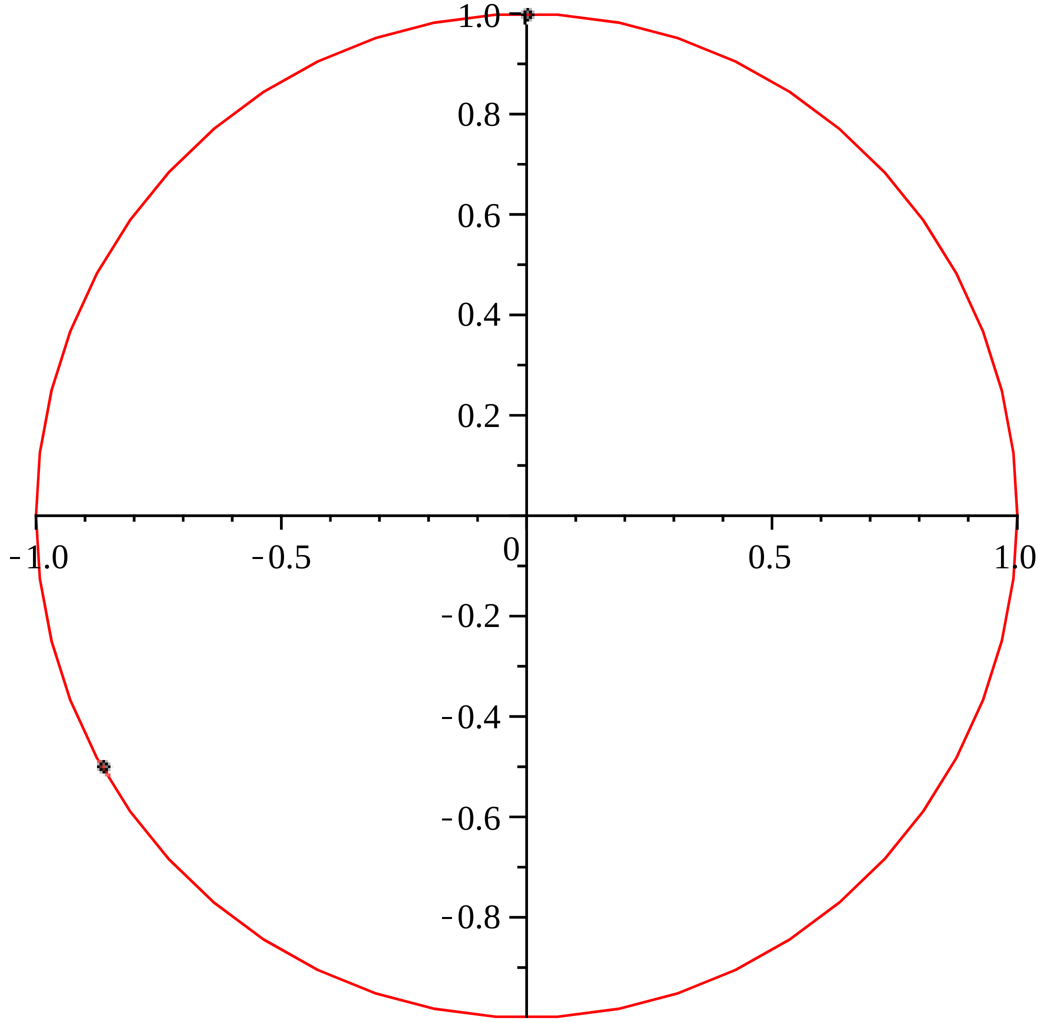
That's good enough for starters. To plot multiple reflections we will need to plot multiple points connected together. We start by plotting two points, one at θ and the other at $\theta + 2\psi$. We do that by defining a variable *points* that is a list containing two points, with each point also represented by a list (a list of lists):

```
> points := [[1,theta], [1,theta+2*psi]];     # Define a list of
  lists containing 2 points
```

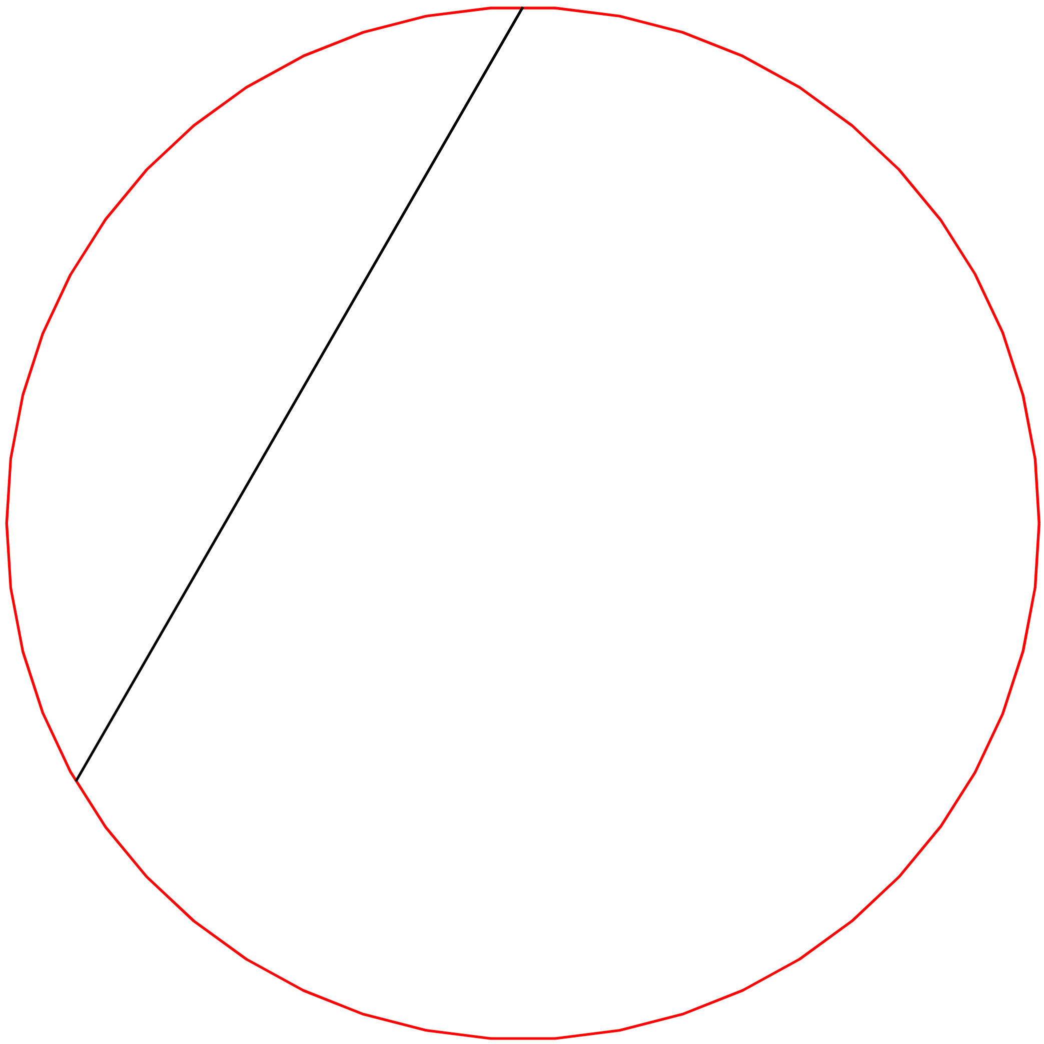
Now we plot these points and display them along with the circle, and then again with no axes displayed but with the points connected:

$$points := \left[\left[1, \frac{1}{2}\pi\right], \left[1, \frac{7}{6}\pi\right]\right] \tag{3}$$

```
>  display(circle(color = red), pointplot(points, coords = polar))
  ;  # 2 points on a circle
```




```
> display(circle(color=red), pointplot(points, coords=polar,
  connect=true), axes=none); # No axes, points connected
```

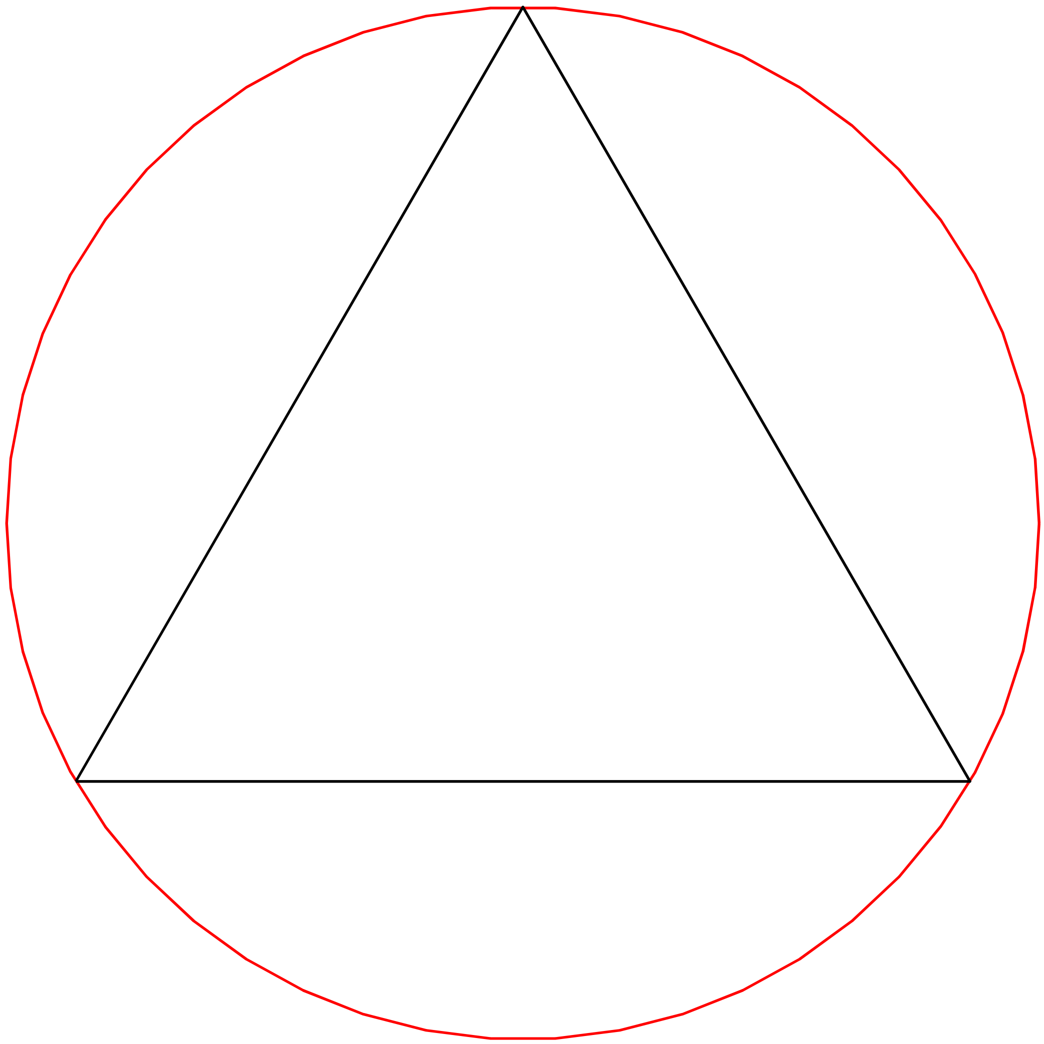


This is good. We are one third of the way around the circle. So we extend the list to include two more reflections, adding 2ψ to the value of theta for each reflection. If this works, we should be back where we started (since the initial angle divided by Pi is rational):

```
> points := [ [1,theta+2*psi], [1,theta+4*psi], [1,theta+6*psi],
  [1,theta+8*psi] ] ; # Lists
```

$$points := \left[\left[1, \frac{7}{6}\pi\right], \left[1, \frac{11}{6}\pi\right], \left[1, \frac{5}{2}\pi\right], \left[1, \frac{19}{6}\pi\right]\right] \tag{4}$$

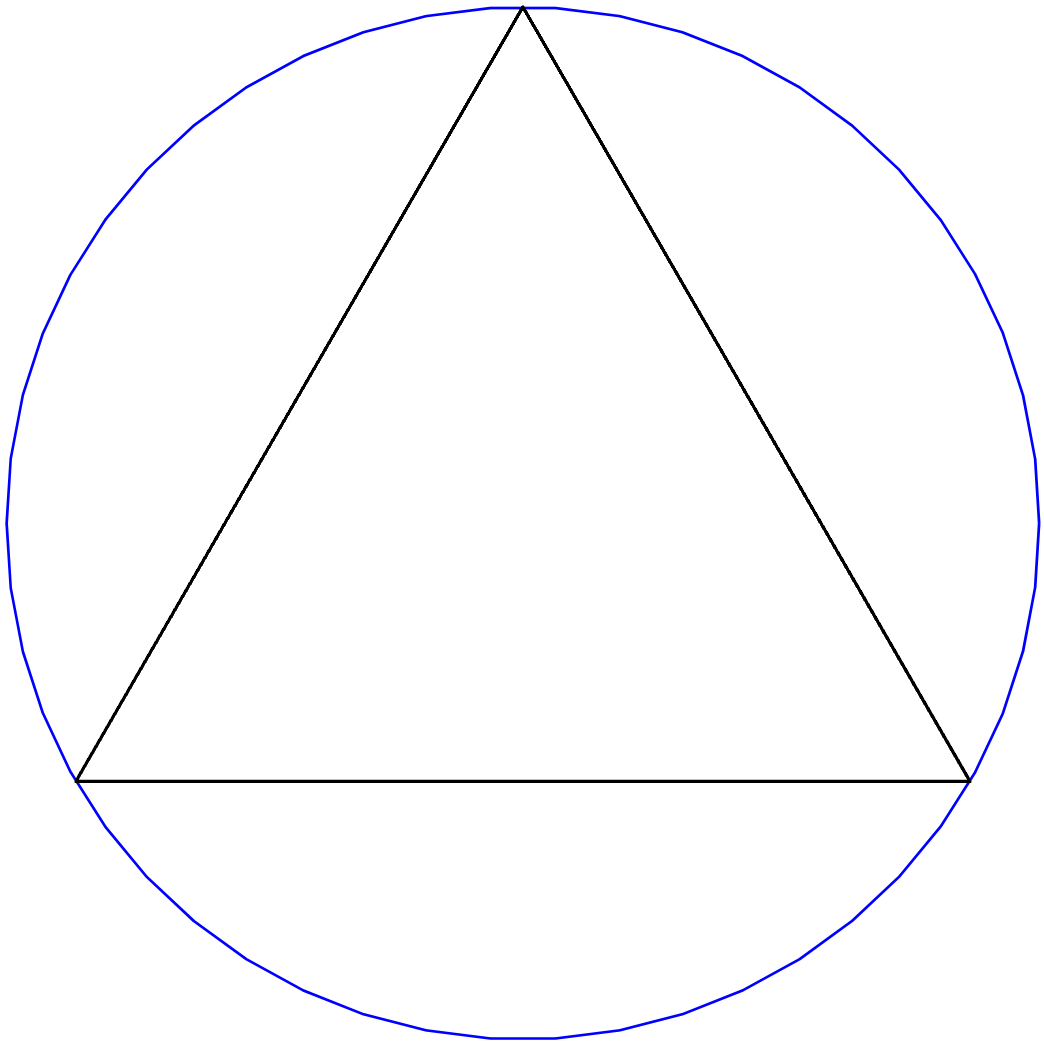
> $display(circle(color = red), pointplot(points, coords = \mathrm{polar}, connect = true), axes = none);$



This is excellent. It shows that for an initial ray angle

$\psi = \frac{1}{3}\pi$, the ray returns to its original position after three reflections, and then continues along the same path presumably forever. OK, we know that forever is a long time and that computers aren't perfect. So let's see explicitly what happens after some 400 reflections . We could do this by making the list of lists contain 400 sublists. We have better things to do with our time and so we will have Maple automatically extend the list to include 400 points, and then plot the results :

```
> for i from 1 to 400 do                        # Begin a loop that
  repeats 400 times
> points := [ op(points), [1., theta + 2.*psi*i] ];
> end do:                                  # End the loop
> display(circle (color=blue), pointplot(points, coords=polar,
  connect=true, axes=none));
```



Note that this *for* loop, which will be discussed later in the text, repeats the command between *for* and *end do* 400 times. The points are then connected with lines in the order in which they occur. The *op* command inserts the previous list of points as the first argument in the list, and thus makes the list grow longer with each repetition of the loop. We notice that all the reflections fall upon themselves, yielding the nice triangle once again. (Actually, some of the lines may look darker, which tells us that in our calculation the ray does not exactly follow the same path. We shall see the reason for this next.)

The second part of the problem asks us to investigate the results obtained using just four places of precision. Maple has a built in constant called *Digits* whose value equals the number of decimal places of precision being kept for floating point calculations. Its value is normally 10, which is what we have been using in the calculation so far. As a check, let's ask Maple what it has stored as the value of Digits:

```
> Digits;
```

$$10$$

While 10 is the default value for *Digits*, you can change it. We will do that to show the effect of reduced precision, first by declaring a new value for *Digits* and then by repeating our previous calculation of reflection for the same initial angle $\psi = \frac{1}{3}\pi$:

```
> Digits := 4;
```

$$Digits := 4$$

```
> theta := Pi/2;  psi := Pi/3;
```

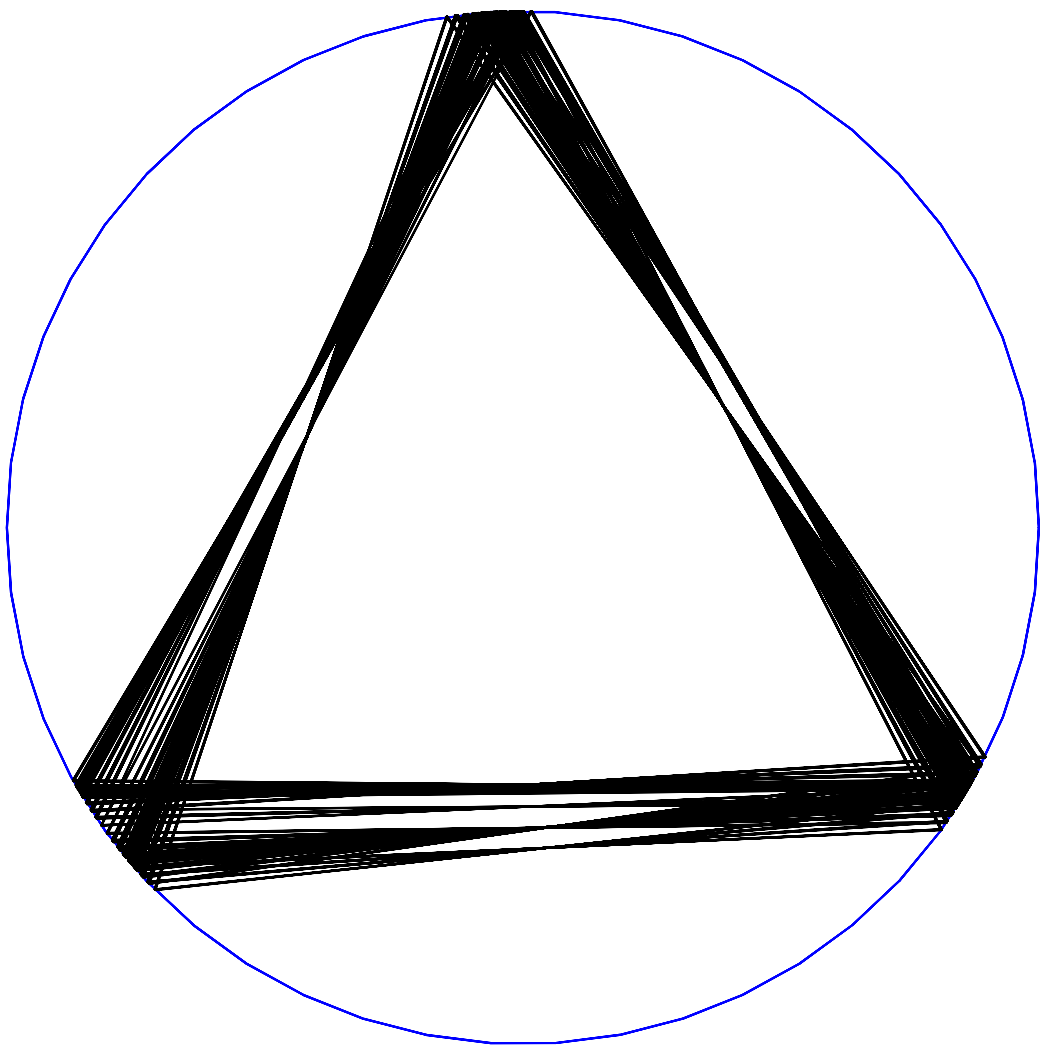
$$\theta := \frac{1}{2}\pi$$

$$\psi := \frac{1}{3}\pi$$

> *points* := [[1, theta + 2 * psi], [1, theta + 4 * psi]];

$$points := \left[\left[1, \frac{7}{6}\pi\right], \left[1, \frac{11}{6}\pi\right]\right]$$

```
> for i from 1 to 400 do
> points := [ op(points), [1., theta + 2.*psi*i] ];
> end do:
> display(circle (color=blue), pointplot(points, coords=polar,
  connect=true, axes=none));
```



> *Digits* := 10;

What we see here is that using only four places of precision leads to significant error. The exact answer, as well as our computation with 10 places of precision, shows that the light ray follows a triangular path. The calculation with four places of precision, in contrast, shows a triangle that is rotating or precessing slightly with each set of reflections. So even though we may not seean error in single step, when the calculation is repeated several hundred times, the error accumulates and becomes evident.

Exercises

Make sure that *Digits* is again set to its default value of 10.

1. Compute and plot the path of a ray of light starting at an angle of psi = 5 Pi/3. Since psi/Pi is rational, the trajectory should close on itself, but since psi is smaller than our worked example, it should take more reflections to get around.

2. Compute and plot the path of a ray of light starting at a value of psi/Pi that is irrational. You should get a beautiful path.

3. Now repeat problems 1 and 2 for four places of precision. Estimate the percentage difference that this causes in the position of the final path of the light ray.