Graphs and Objects linked to Graphs

Cabri can provide a very powerful means of exploring graphs – and a unique means of linking objects with graphs.

The first part of this handout will teach you how to draw the graph of an algebraic function using Cabri. If you are already familiar with using graphs with Cabri, then I suggest that you go directly to the section on objects linked to graphs.

Cabri tools enable you to draw lines, circles and conics and to find the equation of these relative to a particular coordinate grid. If you want to draw a graph that is determined by an algebraic relationship rather than by a geometrical object, creating a locus is usually the best way to do it. This is rather more complicated and it's useful to set up macros to do most of the work for you. Once you have created appropriate macros, drawing a graph with Cabri is almost as easy and far more flexible than drawing a graph with a graphics calculator or graph-plotting package.

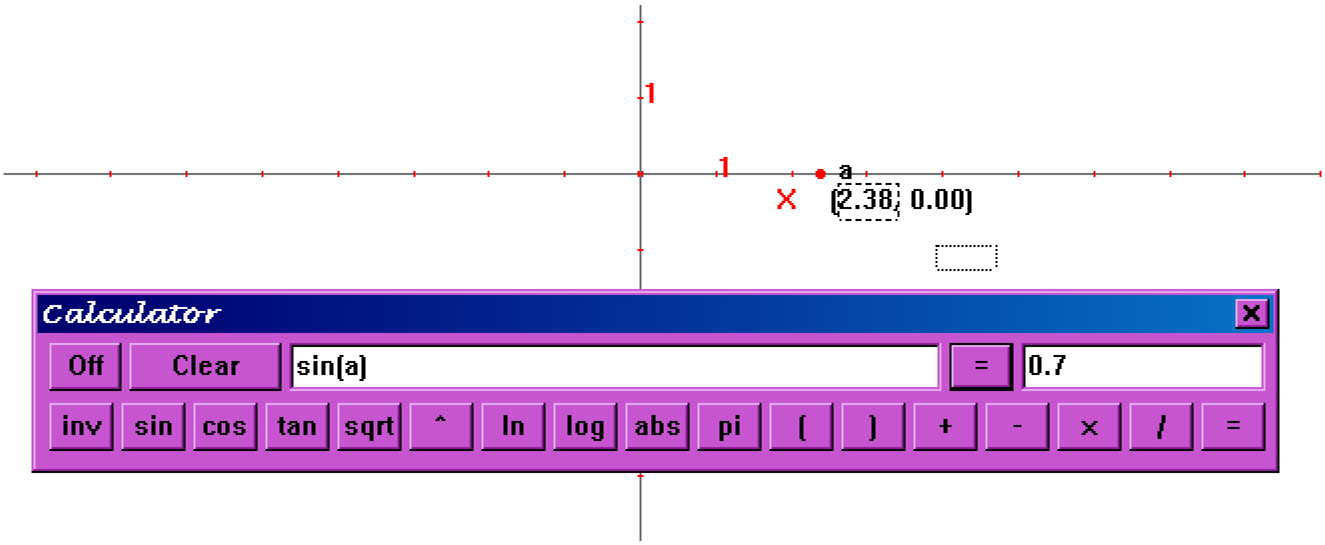
The following instructions will enable you to draw the graph of y = sin x and can be easily modified to enable other graphs to be drawn.

<u>1. Finding x and y coordinates of a point on the graph</u>

Show Axes [1R]. Create ***(Point on Object [2L])*** a point on the x axis and mark its coordinates ***(Equations&Coordinates [3R]).***

It's useful to ***Label [2R]*** this point X and possibly to ***Modify Appearance [1R]*** so that it stands out. The X coordinate of this point will be the x coordinate of the point on the graph.

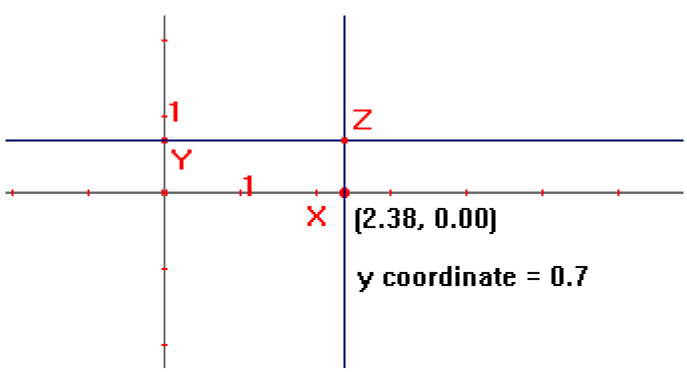
Get out the calculator ***(Calculate [3R]).*** Click on sin (within the calculator) and then on the x coordinate of X. An "a" will appear in the calculator box. Close the bracket. Now click on "=' and also on the result. A dotted rectangle will appear: position this on the screen and click. The result is the y coordinate of the point on the graph. (Note that Cabri uses radians for trig functions). The screen print below is taken just before the final positioning click. It's useful to use either Numerical Edit or Comments to change the text "result" to "y coordinate".



2. Plotting the point on the graph

There is already a point on the x axis with the required x coordinate: you are now going to create a point on the y axis with the required y coordinate. Choose ***Measurement Transfer [5L],*** select the y coordinate value on the screen and the y axis. A point will appear. ***Label [2R]*** this point Y.

Now to use points X and Y to plot the point on the graph. Draw a line through point X perpendicular to the x axis ***(Perpendicular Line [5L])*** and a line through Y perpendicular to the y axis. The ***Intersection Point [2L]*** of these lines is the required point on the graph.
Label[2R] it Z.

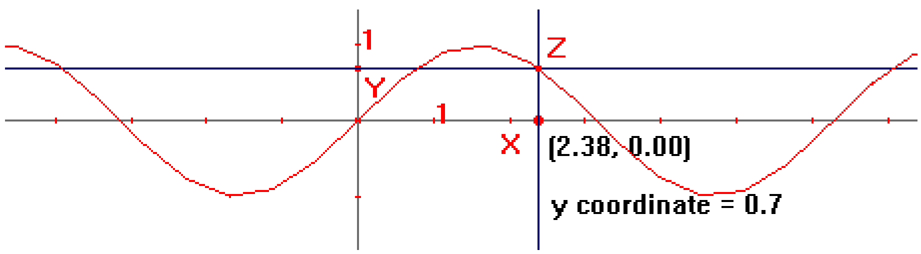


It's useful to create a macro that, when given axes and values for the x coordinate and the y coordinate, plots the point. You can use what you've done above to create such a macro if you like.

3. Drawing the entire graph

To get an idea of what's involved here, move point X along the x axis – what happens to Y and Z? You could also put a trace on Z via ***Trace On/Off [2R]*** . What happens now when you move point X? (To get rid of the trace, use **Edit - Refresh Drawing.)**

To create the graph as an entity in itself, use ***Locus [5L]:*** choose Z and then X and the locus of Z as X moves will be drawn. This is shown in the diagram below:



It's useful to create a macro that given axes, a movable point such as X and x and y coordinates that are related to the movable point creates the locus directly. Note that you may not always want the value for x to be the x coordinate of the movable point – for example for a parametric graph where both the x coordinate and the y coordinate are calculated.

Improving the graph

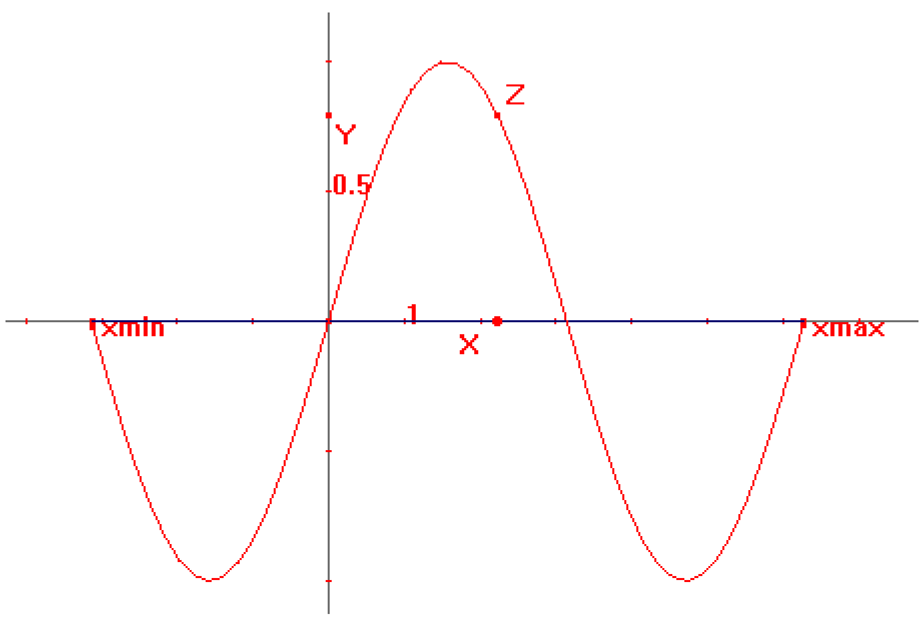
First, and most obvious: hide the lines (***Hide/Show [1R]***) whose intersection determines point Z. One of the advantages of using a macro to plot points is that these lines don't appear.
You might also want to hide points X, Y or Z and the coordinates and calculated y value.

Second, adjust the scale on the axes. Find the unit at 1 on the x or y axis (move point X if necessary) and explore moving these units. Note that moving the unit on the x axis also moves the unit on the y axis, but not vice-versa.

Third, add more points to the locus. You may notice that it is made up of connected line segments and, particularly when you change the scale on the y axis, does not look like a smooth curve. There are two ways to do this. One way is by changing the number of points used to determine the locus (select (***Pointer [1L]***) the locus and then **Options - Preferences - Loci Options** and change the number 50 to a higher number). This may make the locus rather slow to change, however – if you are exploring the effect of changing various parameters on a graph, it's a good idea to keep this number fairly low.

The second way is to limit the range of possible x values: Create two points on the x axis (***Point on Object [2L]***). ***Label [2R]*** these "xmin" and "xmax" and create a line ***Segment [3L]*** joining them. Redefine X (***Redefine Object [5L]***) as a point on this segment: the locus will then be drawn for a narrower range of x and will look much smoother. This is of course also very useful when you only want the locus defined for particular x values.

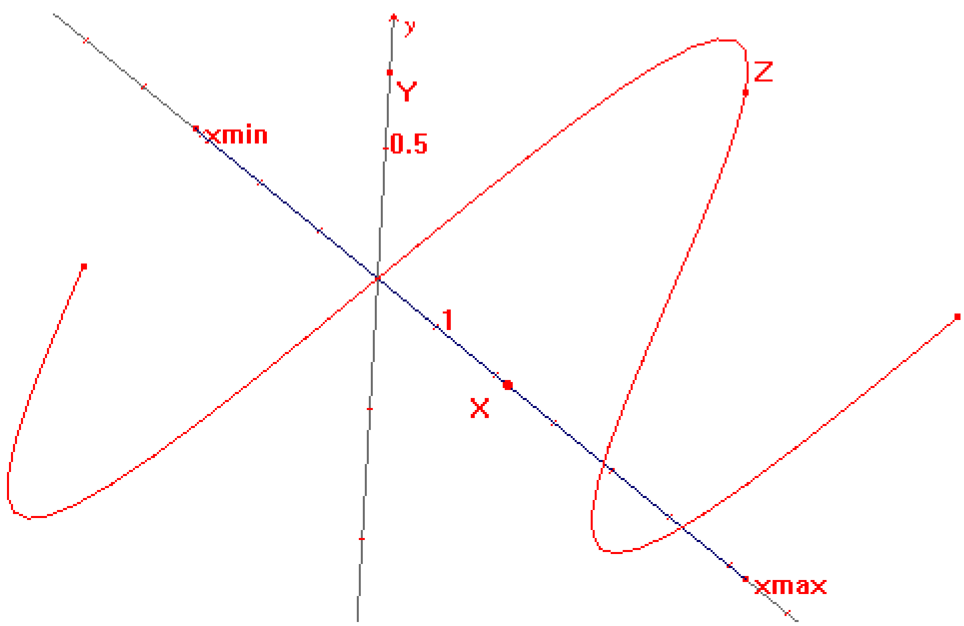
In the diagram below, lines and coordinates have been hidden, the unit on the y axis has been modified and X has been limited to the segment between "xmin" and "xmax".



Problems:

1. The original axes

The diagram below shows what can happen if you let your students loose on a graph that uses the original axes: the origin can be shifted, the axes can be rotated (move the x axis) and the angle between the axes can be changed (move the y axis). A fascinating graph can result– but this may not be quite what you wanted your students to be exploring.



Note that Z does not now represent the point with the required x and y coordinates – what would you need to change in the procedure for creating Z to enable the plotting of points with oblique axes?

When I am setting up a graph now, I never use the original axes – unless I intend to hide them eventually. It's better to set up new axes, less amenable to manipulation. One way to do this is detailed below:

1. ***Show Axes [1R].*** Create (***Point on Object [2L]***) a point anywhere on the positive x axis and a

second point anywhere on the positive y axis and ***Label [2R]*** these "x=1" and "y=1".

2. Select ***New Axes [1R]*** , and then select the points (0, 0), "x=1" and "y=1".

3. Now ***Hide Axes [1R]*** . Note that the point (0,0) disappears when you do this: this can be quite irritating!

4. Move the points "x=1" and "y=1" to the scale that you want. If you want this scale to remain fixed, then hide them (***Hide/Show [1R]***).

To redraw your graph using the new axes, display the coordinates of X in the new axes (***Equations&Coordinates [3R]***), edit the y coordinate calculation (get into ***Calculate [3R]*** and double-click on the number) and replot Y and Z.

2. Redefining points

HEALTH WARNING: Redefining objects with Cabri may cause the file to become unusable if you try to do this towards the end of a complex construction – there is an intrinsic bug somewhere here.

If you don't want to risk losing the work you've done on a complex file, save the file under a different name immediately after any redefinition. Close the file and then open it again: if it opens, then you're okay to proceed. If it doesn't, you'll have to go back to your original and find another way of proceeding without doing a redefinition.

<u>Further refining the locus</u>

1. Cosmetics: You can change the ***Colour [1R]*** or ***Thickness [1R]*** of a locus, or make it ***Dotted [1R].***

2. You can create points on a locus and hence on the graph: unfortunately you cannot find the point(s) of intersection between a locus and another object.

3. Further options for limiting the range of x or y values

It's possible to limit the range of x for a particular locus in various ways without confining the point X – this can be useful if a function is defined differently for different values of x.

One possibility is to set up a calculation (see Logic, discussed in the final session) which has the value 1 when x is within a particular interval, and 0 otherwise. In the calculation of the y coordinate, divide by this number: the locus will be defined within the interval and will not exist outside.

Another way is to create separate line segments for each x interval of interest and duplicate the point X in each segment i.e. create a point which is in the same position as X when X is within the interval, and which doesn't exist when X is outside it. Use the coordinates of the duplicate point rather than of X in the calculation of the y coordinate – or simply make the vertical line defining the x coordinate of Z run through the duplicate point rather than X. A macro for duplicating will be discussed in the final session.

Tracing a graph

The simplest way to trace a graph is to create a point on the locus and display the coordinates (***Equations&Coordinates [3R]***) of this point (you can move the coordinates away from the point with the ***Pointer [1L]*** and change the format from e.g. (2,5) to x=2,y=5 via ***Numerical Edit [2R]***). (BEWARE: do not remove either number if you do this: you may not be able to re-open your file). You can of course use the point Z for this – though Z can't be moved directly, unlike a point on the locus.

It's also possible to create a rather nice effect by overlaying a second locus of a different colour, which is identical, but defined between "xmin" and X, rather than "xmin" and "xmax". This is very useful when a graph is linked to a moving object and the variable x represents time: you can watch the graph being drawn as the object moves. We will use this idea later in creating a graph linked to an object.

Using parameters in a graph.

Cabri is particularly useful here: creating a basic graph can be quite fiddly compared with using a graphics calculator or specific graph plotting package. The ease with which parameters can be changed more than makes up for this.

We are going to modify the graph of y = sin x to become the graph of y = a sin(bx + c), where a, b and c are parameters that can be changed on the screen.

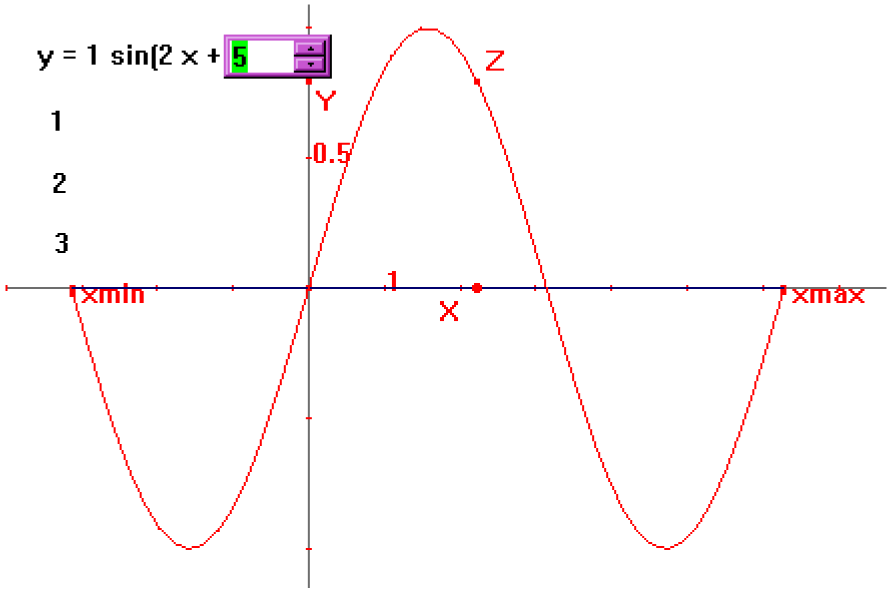
The first step is to create values for a, b and c on the screen.

The simplest way to do this is to use ***Numerical Edit [2R]*** and put three numbers on the screen (1, 2 and 3 have been used below).

Now use ***Comments [2R]*** to put in "y = *include first number* sin(*include second number* x + *include third number*)".

Delete the original three numbers – note that the included numbers stay put – and can be changed by ***Numerical Edit [2R].***

The diagram below illustrates this (the original numbers have not yet been deleted) – the value of c is in the process of being changed from 3 to 5.



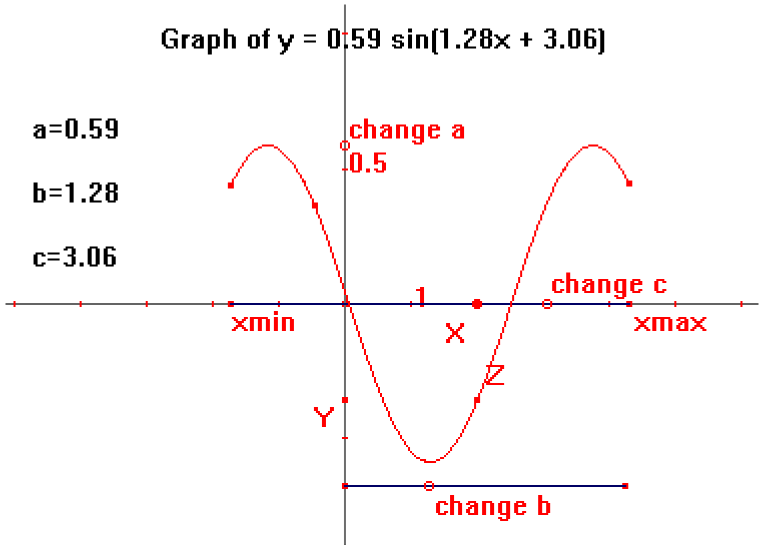
A second possibility is to use three points on lines or line segments.

A point is created on each line: either the distance from one end of the segment or the coordinates of the point are used to determine the value of the parameter. Parameter values are changed simply by moving the point.

The lines can be distinct from the graph or can form part of it: in the diagram below, a is represented by a point on the y axis, c is a point on the x axis and b is a point on an independent segment. This can facilitate identifying the effects of the various parameters on the graph (or can create an unusable mess!).

The coordinates of the points “change a”, “change b” and “change c” have been displayed, incorporated into the comments “a=….” etc. and then hidden.

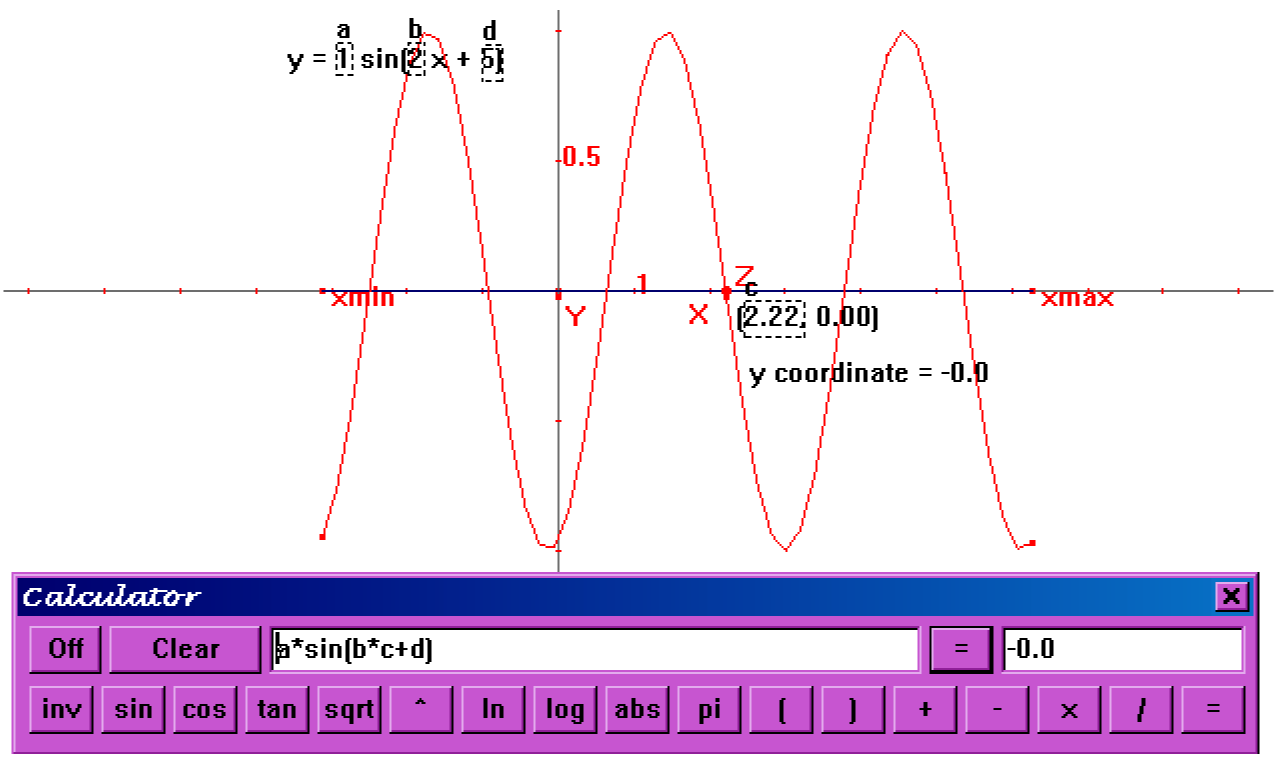
If you are using independent line segments it can be useful to define new axes just to enable you to use coordinates relative to these axes as parameter values, rather than having to do any calculations to determine these values.



The next step is to modify the calculation of the y coordinate to incorporate these parameters.

Open the calculator, double-click on the y coordinate calculation, and then modify as shown.

As soon as “=” is pressed in the calculator, the new graph will appear.



Note that I have ensured that the variable names appear in alphabetical order in the calculation (and note that a, b, and c in the calculation do not entirely correspond to the parameters a, b and c), even though this has meant reselecting the x coordinate – Cabri can mess up calculations in which this is not done. If ever you get really peculiar results, check your calculations to make sure that Cabri is using your variables in the correct sequence.

Animation

You can easily change the graph by either editing numbers or moving points on line segments. What's also very effective is to animate these changes so that the parameters change at a steady rate.

Numbers:
Try animating either a, b or c (***Animation [2R]***). Interesting, but perhaps not so useful. A better way of creating steady change is by means of the up and down arrows in ***Numerical Edit [2R].*** Note that you can put the cursor after any of the digits: clicking (and holding) the up or down arrow will change the number at the cursor, enabling you to fine-tune the rate at which the parameter and hence the graph changes.

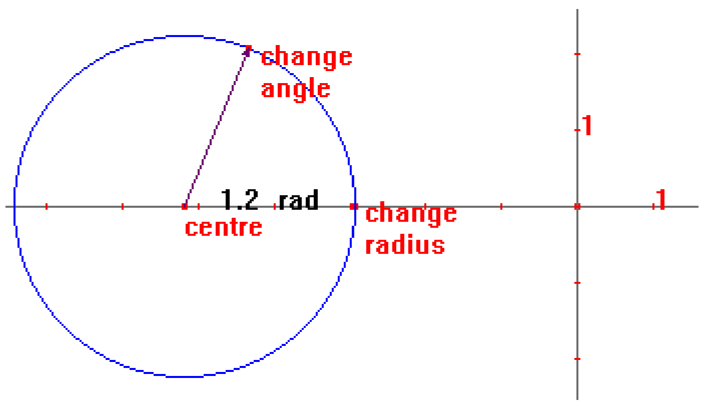
Sliders:
Use ***Animation [2R]*** on the points determining the parameter value. It's useful to have limited the parameter value (as is the case with b) – it can easily become very large and you can waste a lot of time searching for determining points that have shot off the screen. (**File-Show Page** is very useful if you do lose points this way). Clicking on a point to animate it makes for quite a rapid change: I prefer to click and drag the spring that appears.

Further animation possibilities:
Animation is wonderful for calculus: make point Y very visible (perhaps a large dark-coloured dot – or even surround it with a filled circle). Now animate X (***Animation [2R]***) and watch the way Y changes. You are seeing dy/dx in motion!

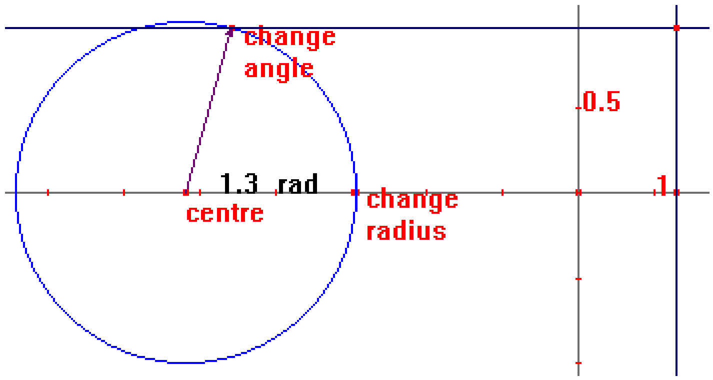
<u>Linking graphs and objects</u>

This is another area where Cabri shines. We are going to create an example of such a link, linking the definition of sin x with the graph of this function. This will also involve looking at problems with angles and hiding and showing graphs.

Open a new file and ***Show Axes [1R].*** Create two points on the negative x axis (***Point on Object [2L]***) and ***Label [2R]*** these "centre" and "change radius" as shown on the diagram below. Create the corresponding ***Circle [4L]*** and a point on the circle (***Point on Object [2L]***) with ***Label [2R]*** "change angle". Join the circle centre to this point with a ***Vector [3L].*** Make sure that angles are measured in radians (**Options – Preferences – Display Precision and Units**) and measure the ***Angle [3R]*** at the centre of the circle determined by the three points.

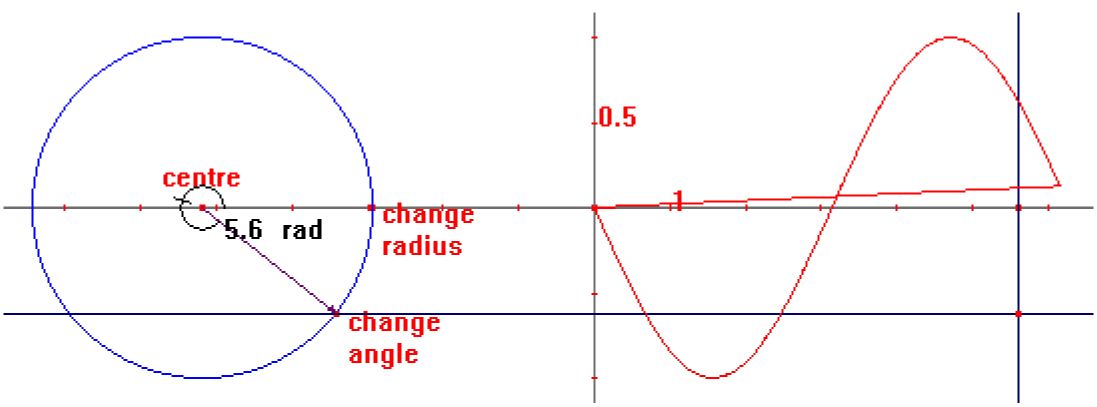


You're all set to plot the graph of sin x! All you need to do is plot the height of the point "change angle" above the x axis against the angle measurement. The height of the point is easily transferred via a line through this point ***Parallel [5L]*** to the x axis and the angle measurement can be transferred to the x axis via ***Measurement Transfer [5L].*** A vertical line through the point on the x axis (***Perpendicular Line [5L]***), ***Intersection Point [2L]*** of the vertical and horizontal lines and drag (***Pointer [1L]***) the unit 1 on the y axis to correspond to the circle radius and you're almost there. The diagram below shows the file at this stage.



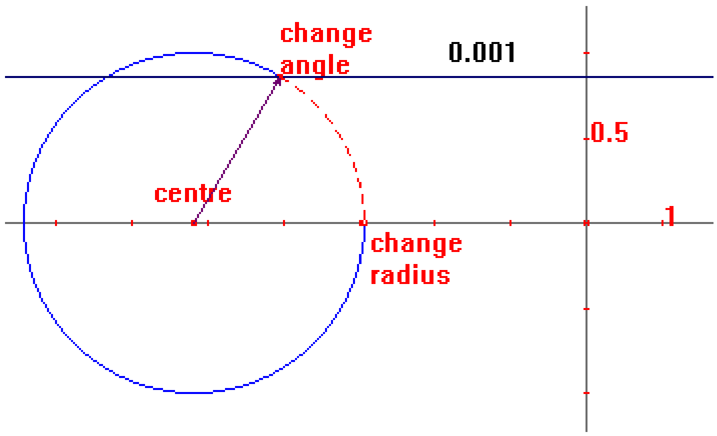
Or are you almost there? Create the ***Locus [5L]*** of the new point as "change angle" moves. What's gone wrong? Move (***Pointer [1L]***) "change angle" around the circle and watch the angle measurement.

A feature of Cabri that's useful in basic geometry is that, unless an angle is marked, angle measurements are always less than 180º. It's not useful here. Mark the angle and measure the marked angle – you can get the measure of a reflex angle now. Create the locus. The angle measured depends on the direction that "change angle" is moving around the circle and the situation hasn't improved much. A diagram of this is shown below.

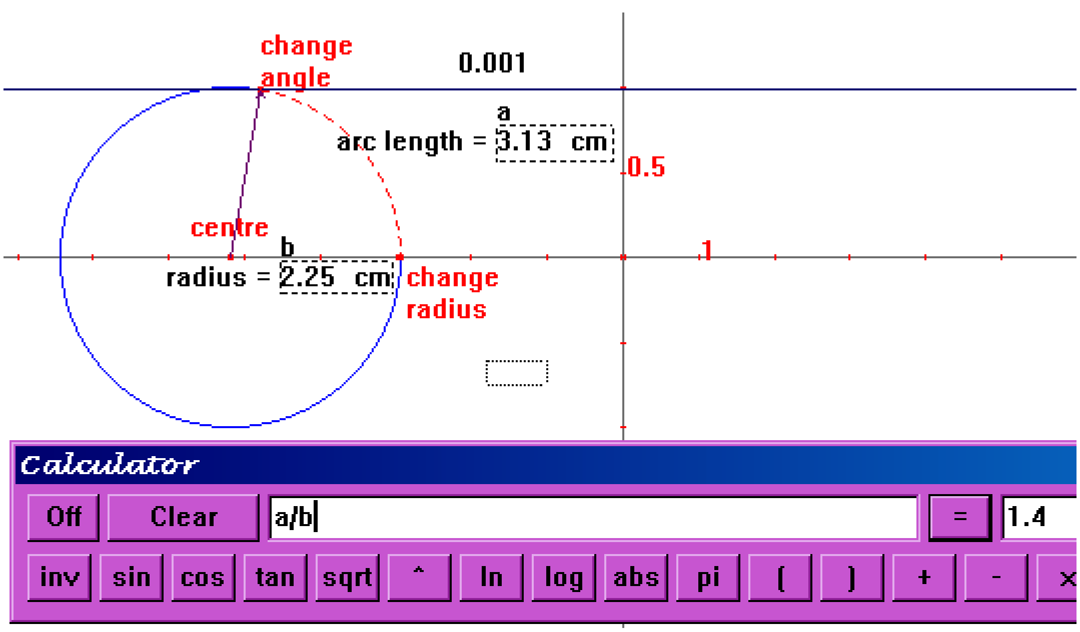


I've basically stopped using angle measurements with Cabri if I am going to be dealing with angles that might become reflex. What I do instead is to use arc length ÷ radius.

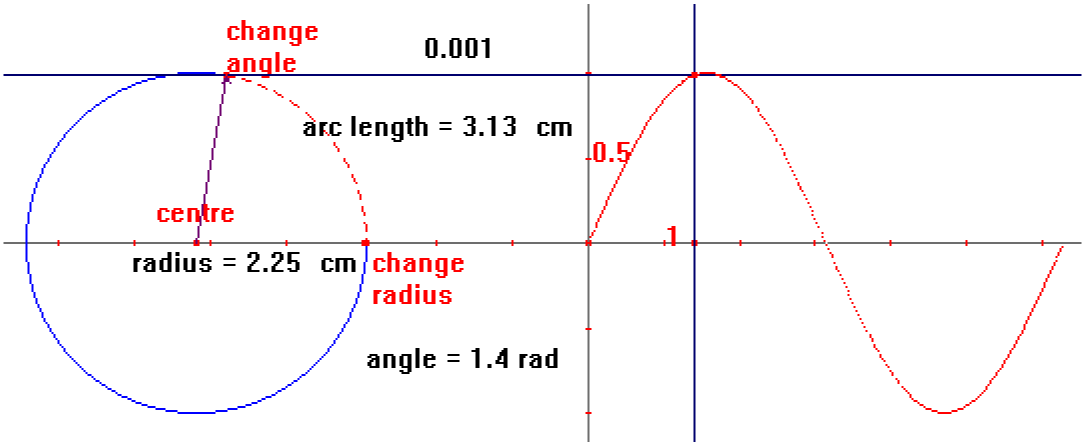
Get rid of all the angle measurements and try to create an ***Arc [4L]*** running from the point "change radius" to "change angle". Problem: an arc needs three points to define it. One way to solve this problem is to create a point on the circle between "change radius" and "change angle". The simplest way to do this is to create a very small number (***Numerical Edit [2R]***) and transfer this length to the circle (***Measurement Transfer [5L]***), starting at point "change radius" (note that to transfer a measurement to a circle you need to choose the circle before you choose the point from which the measurement is measured). Now create an ***Arc [4L]*** through "change radius", the new point and "change angle" and make the arc ***Dotted [1R].***



Measure the circle radius and the arc length (***Distance and Length [3L]***) and then ***Calculate [3L]*** the angle as shown below:

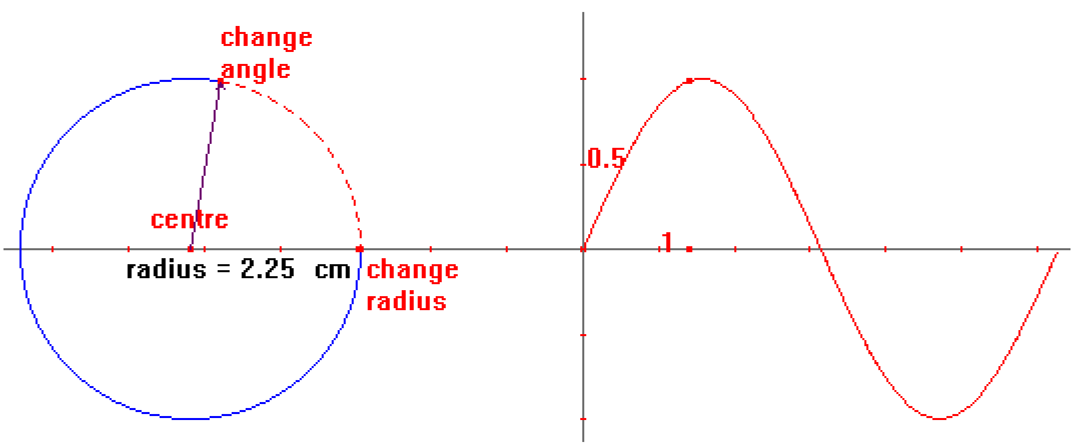


Move "change angle" around the circle – note that the angle measurement disappears when "change radius and "change angle" coincide. Not ideal, but better than the previous mess. Now transfer this angle measurement to the x axis (***Measurement Transfer [5L]***) and follow the previous instructions to draw the locus – much better! You will find that your locus does not quite reach the x axis: in this case it's worth increasing the number of points used to define the locus (**Options – Preferences – Loci Options**). The diagram below shows the locus defined with 250 points.

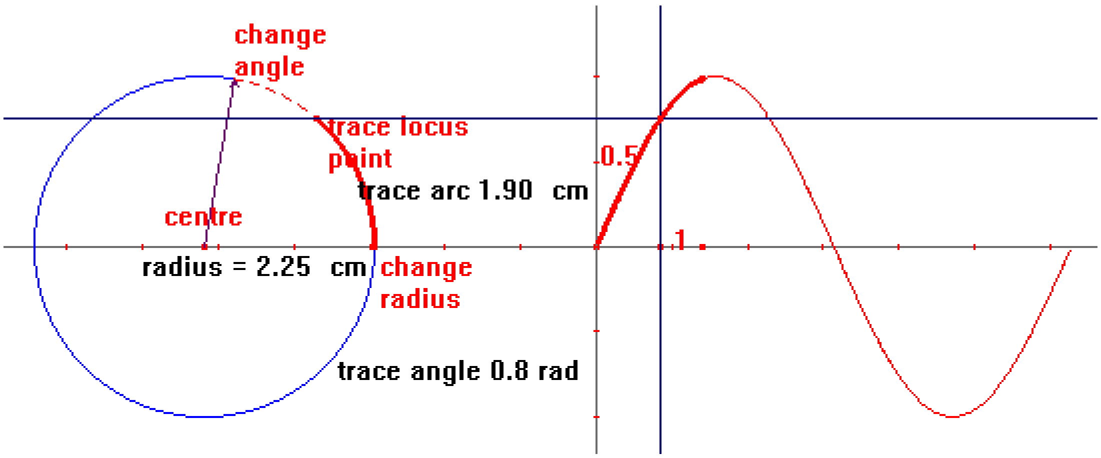


We are now going to create a fancy trace for the graph, by overlaying the locus which you have just created with a locus which is identical, but only defined for angles up to the one currently defined by "change angle". This will enable us to animate the point "change angle" and in effect watch the graph draw itself.

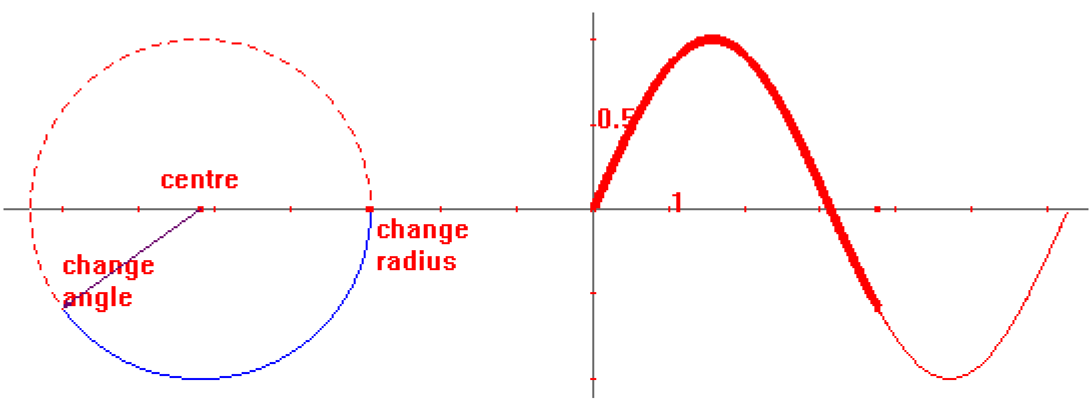
First of all, hide (***Hide/Show [1R]***) the number 0.001, the angle measurement, arc length and the horizontal and vertical lines.



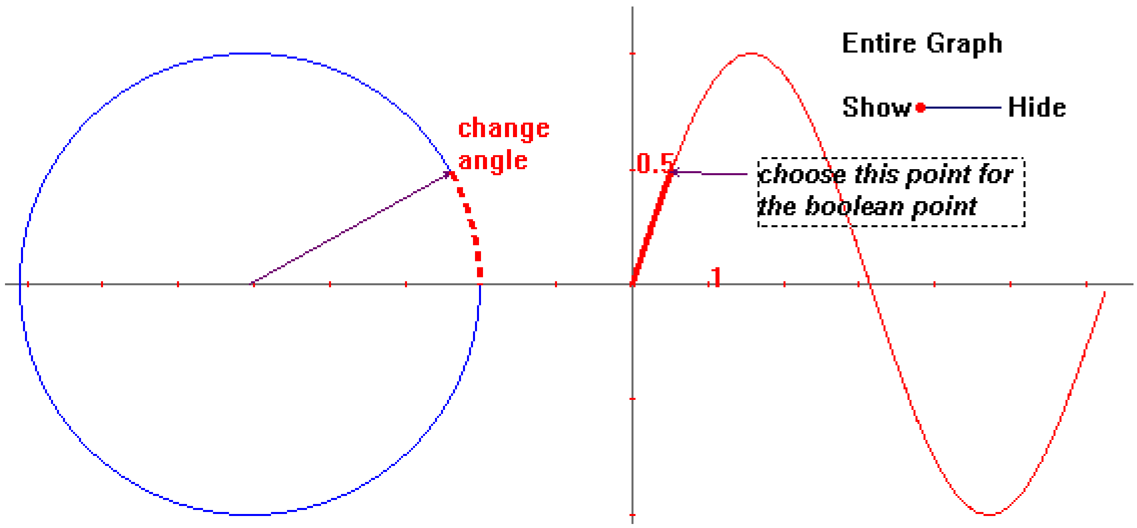
Create a point (***Point on Object [2L]***)(I've labelled it "trace locus point") on the arc between "change radius" and "change angle". Now treat this point exactly as you did "change radius": draw a line through the point parallel to the x axis, create an arc between the point and "change radius" and convert the length of this arc to a measurement of angle. Transfer the angle measurement to the x axis, find the point of intersection of vertical and horizontal lines and create the locus of this point – as "trace locus point" moves. This is illustrated in the screen shot below:



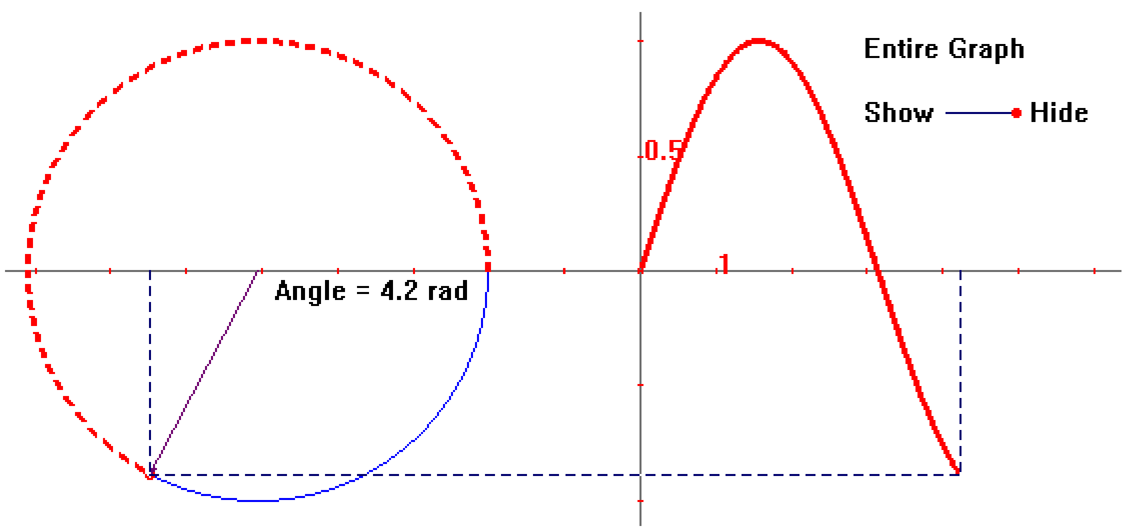
Now hide the trace locus point, the trace arc, the measurements, the points and lines used to construct the trace locus.



We are now going to revise the original locus so that it can be hidden by means of a slider. (Sliders were discussed in the Fishing Game). **File - Open** the Button macro and create a ***Button [5R]***, with the two slider points wherever you like – but the third point where the boolean point will exist chosen to be the intersection point which determined the original locus. This is shown in the diagram below. Move the slider so that the boolean point exists – and delete the original locus. Now create a new ***Locus [5L]*** of the boolean point as "change angle" moves. Adjust the number of points that define this locus, and you will have a locus that is identical to the original one – but which disappears as you move the slider button. You now have a choice of the trace locus overlaying the original locus – or being drawn on its own.



Time to tidy up – hide whatever you feel is unnecessary (and perhaps show the angle), add a few line segments to clarify what's going on, etc. Now animate "change radius" – and watch the graph draw itself!



As an extension, add the graphs of tan x and cos x. You might want to revise the trace graph so that it can be hidden by a slider before you do this.

Using graphs to model motion

Quite often, in modelling motion, a graph can be a direct "picture" of the motion. An example is a projectile: a basketball can be thrown at a basket using Cabri, by creating it as a point on an underlying graph. In this case, the graph is created – and then hidden.

Graphs defined by recursion

In such graphs, the y coordinate of a point is at least partially defined by the coordinates of the previously plotted point or points. These are rather harder work, as macros must be created to plot individual points and then combined to plot a series of points.

Further possibilities:

Graphs (endless possibilities – these are a few to get you started):

1. Explore polynomial graphs such as the graph of $y = ax^2 + bx + c$. How close can you get to $y = \sin x$, by adding polynomials?

2. Calculus: plot the graph of a function together with the tangent to the curve. Animate to see y move as x moves.

3. Graphs using polar coordinates – again, use arc length rather than angle.

4. A parametric curve: try $x = \sin a\, t$ and $y = \cos b\, t$ for different values of a and b. Link this graph with graphs of y against t and x against t.

Graphs linked with objects (I've created the following ones, so I know they're possible and effective!)

1. The maxbox problem: what is the largest open-topped box you can make from a rectangular piece of paper?

2. Distance – time graph linked with a moving object.

3. Motion of a pendulum linked with a graph of its angular displacement.

4. Collision between two moving objects linked to a distance-time graph.

Using graphs to model motion

Set up a simulation of a sport that involves projectile motion.

Graphs defined by recursion

1. Recurring decimals.

2. The Verhulst sequence ($y_{n+1} = Py_n (1-y_n)$)

3. Differential equations representing motion– for example skydiving.