

# 3d point plotter

## Introduction

This plugin allows you to generate points on a grid according to a formula that you input. You can then create a polygon mesh from the points, or a network of splines, or both. You can also add a 'cage' around the plot, and axis labels if you wish.

The plugin is similar in some ways to the Formula deformer supplied with Cinema 4D. This is not a deformer, however, but a collection of generators producing several objects. There are significant advantages over the Formula deformer, including:

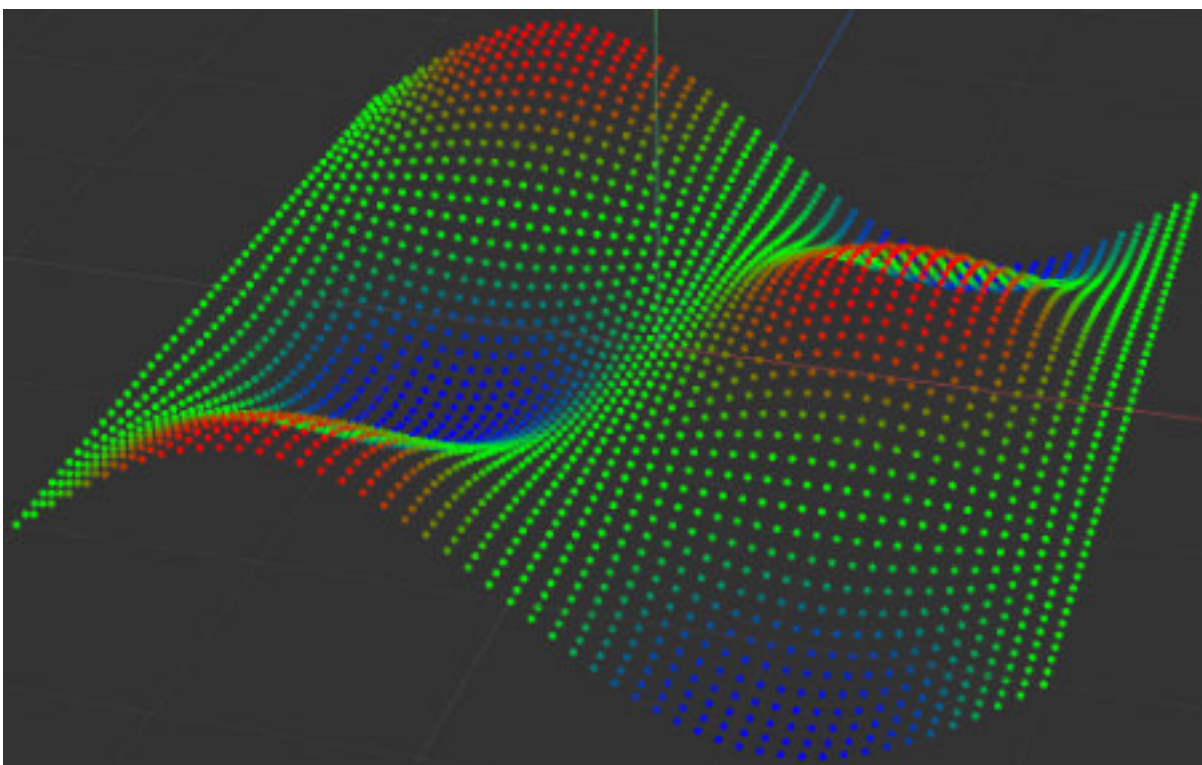
- the formula can be animated using time parameters
- no additional geometry is needed, unlike the Formula deformer
- a polygon mesh in various forms can be generated from the plotted points
- a grid of splines can also be generated from the points
- the points and meshes can be coloured using a Vertex Map shader (for flat geometry) or directly if point objects are produced
- an enclosing cage and/or axis labels can be added

The result is to produce a 3D graph from the supplied formula which can then be rendered.

## How it works

With its default settings, the 3D Plotter object creates a grid of points along the Plotter's X and Z axes. Each point has an X and Z coordinate assigned to it, but the Y coordinate is calculated using the formula you input. For example, here is the resulting grid using the default formula when the Plotter is created, which is:

$$\sin(x) * \cos(z)$$





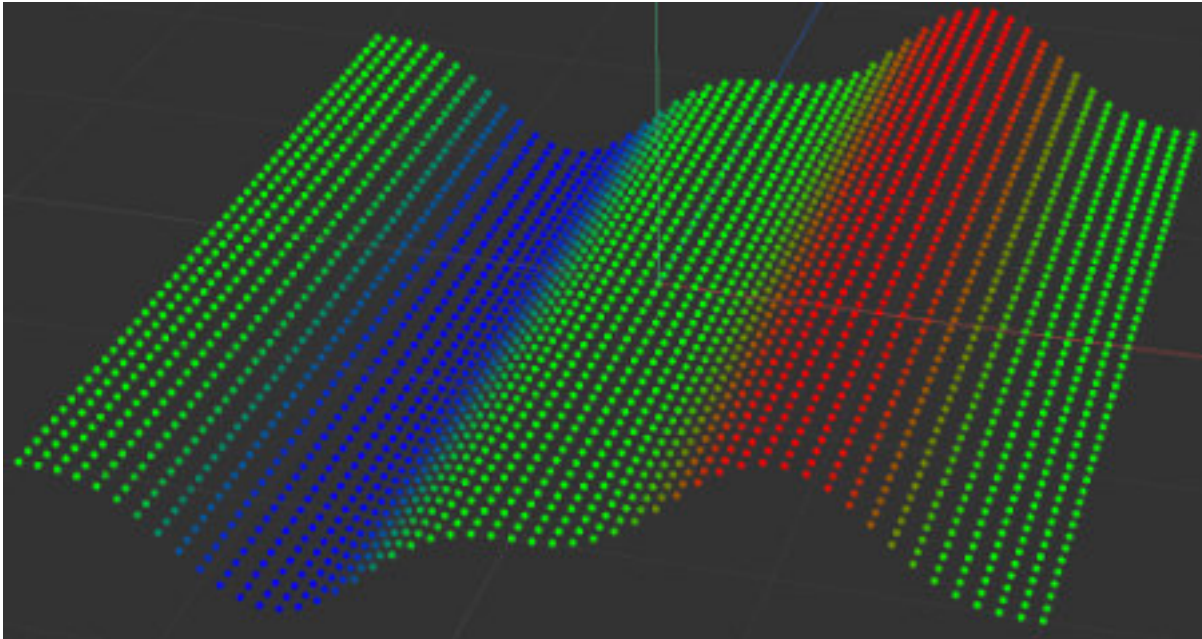
# 3d point plotter

As you can see, for each point the sine of its X coordinate and the cosine of its Z coordinate are multiplied together to give the Y coordinate.

You don't have to use the X and/or Z coordinates at all although typically you would use at least one of them. For example, this formula uses only the X coordinate:

```
Pow(Sin(x); 3)
```

and produces this:



The plugin has several preset formulae built-in to get you started.

## Disclaimer and legal notes

1. This plugin is free to use and may be distributed to anyone who would like a copy. Copyright of the plugin and this documentation remains with the author, Steve Pedler.
2. The latest version may be downloaded from my website at <https://microbion.co.uk/html/plotter.htm>.
3. This program is distributed in the hope that it will be useful, but it carries no warranty of any kind. There is no implied warranty of fitness for any purpose and you use it at your own risk.

## Installation

**Note: this plugin requires C4D R23 or greater. It will not run on earlier versions.**

Separate versions are provided for R2025, R2024, R2023, R25/S26, and R23/S24. Make sure you download the correct one.

### **Installing**

To install the plugin unzip the supplied archive into the 'plugins' folder of your user data folder. After restarting Cinema 4D the plugin will be found in the "Extensions' menu.

# 3d point plotter

## Using the plugin

To use the plugin, add the 3D Plotter object to the scene. This will generate points using the default size, resolution, and formula for the grid. These will show as coloured dots in the viewport (you can alter the viewport display settings if required) but by themselves they will not render.

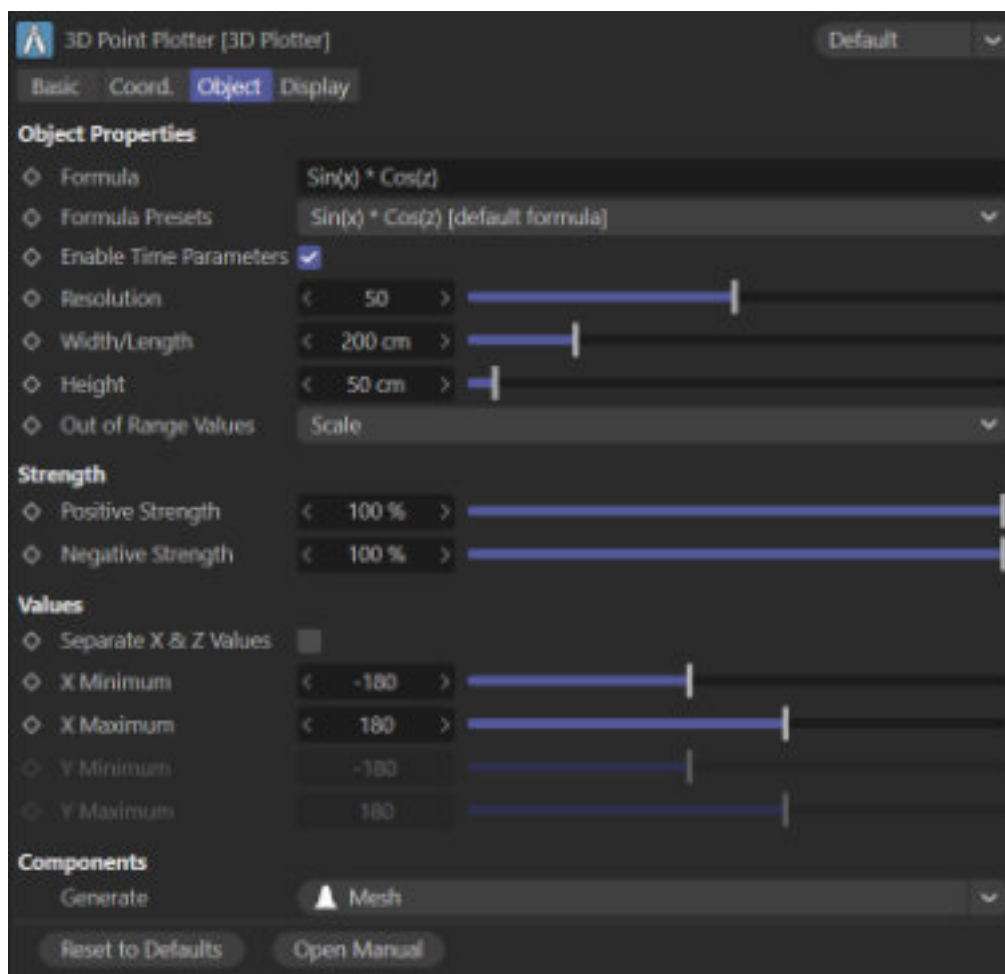
You can alter the settings as desired and then, if you wish, add a Mesh object to generate a variety of polygon meshes from the points. In addition to or instead of the Mesh object you can add a Grid object to generate a grid of splines. If desired, you can also add a Cage object to produce a cage of splines around the point grid and/or text labels on the axes with a Labels object.

## 3D Plotter Object

The 3D Plotter object interface has two tabs, Object and Display. It can also create additional objects, each of which has its own interface: the Mesh, Grid, Cage, and Labels objects.

### The Plotter: Object tab settings

The tab interface looks like this:



The settings are as follows:

### Formula

This is the formula used to calculate the Y-coordinate of each point's position. What you enter





here makes a huge difference to the result. What you can do with these formulae is discussed in more detail below.

## *Formula Presets*

This is a menu of several interesting presets to get you started. Try each of these and see what results you get.

## *Enable Time Parameters*

It is possible to animate the point grid using the scene time or frame as a parameter for the formula, but to do that you must turn on this switch. If the switch is off, time parameters are ignored.

There is a reason for this switch. If you are using time parameters in the formula, the Y-coordinate of each point must be recalculated each frame. This causes a slight slowdown in the animation frame rate, but if you then add a Mesh object, and if that is set to 'Points (Objects)' mode the slowdown due to the need to regenerate the objects each frame may be significant. Therefore, this switch can be used to turn off the time animation while developing the scene, then turn it back on when ready.

## *Resolution*

Simply, how many segments the X and Z axes of the grid are divided into. The default is 50 but what this actually means is that the X and Z axes are divided into 50 segments, not that 50 points are used along the axis. Fifty segments require one more point along each axis giving a total of  $51 \times 51 = 2,601$  points. This is often enough but for complex formulae with lots of detail or if you make the grid size very large, you might need to increase this value. If you decrease it too much you risk making any generated mesh too coarse.

## *Width/Length*

The actual size of the grid in scene units. The grid is always square, so the width and length are the same. Increasing the resolution does not affect the size of the grid, it just means that more points are packed into the same grid size.

## *Height*

The height of the grid. This is split evenly above and below the centre of the grid, so if the height is 50 units, the top of the grid will be at Y +25 and the lowest at Y -25.

However, there is another factor to consider. Depending on the formula used, the calculated Y coordinate may exceed – possibly very greatly – the height value. These are considered 'out-of-range values' and dealing with these is what the 'Out of Range Values' menu is for.

## *Out of Range Values*

Suppose you have this very simple formula:

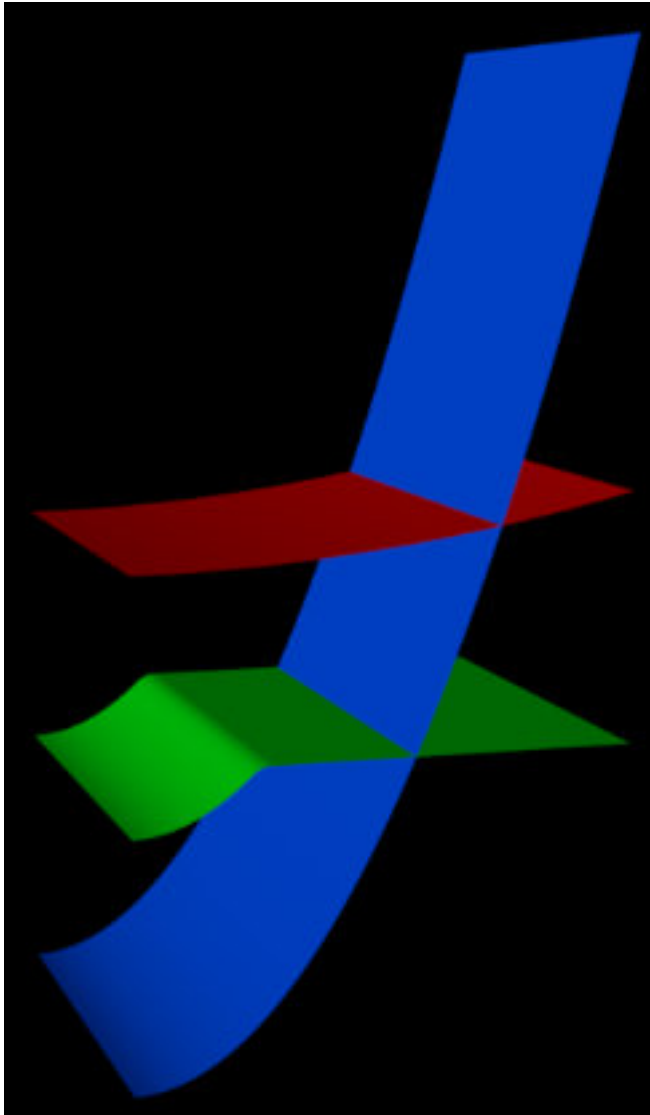
$$x * x$$

You'll probably recognise this as a simple formula producing a parabolic curve. As 'x' increases in value, the calculated Y value will become very large. The default setting for this menu is 'Scale'. What this does is take the maximum calculated Y value and any points with this Y value are located at the maximum height of the grid, which is the value in the 'Height' parameter. All other values of Y are scaled down accordingly. This formula with the 'Scale' mode and all other default values produces a rather flat parabolic curve.

# 3d point plotter

The next option is 'Clamp'. Instead of scaling the Y values, each Y value is given the value calculated for it without scaling. Taking the above example again, suppose 'X Maximum' is set to 20 units. This produces a maximum Y value well in excess of the maximum grid height, so with 'Clamp' selected any Y value over 50 (the default maximum height) is automatically set to 50. This produces a steeper curve which then ends abruptly when the calculated Y value exceeds 50, producing a flat plateau.

The final option is 'Ignore'. With this, there is no restriction placed on the Y value and the maximum height setting is not used. In the image below, the minimum and maximum X values are set to 0 and 4 respectively. There are three plotters showing each curve that results (they have been moved apart on the Y axis so you can see the results). The red curve uses 'Scale', the green curve is 'Clamp' and the blue curve is 'Ignore'.



Note that when using 'Ignore' the size and location of the labels and cage, if they are used, do not follow the actual height of the plotted points. They will still use the maximum height setting in the Plotter to place the generated splines and labels.

## *Positive and Negative Strength*

Suppose you have a curve such as the default formula. Some points have positive Y coordinates while some have negative coordinates. But you would like the negative values not to be used – that is, if the value is negative, it should be set to zero. You can do this by setting the negative strength to zero, and then all the negative Y coordinates will be set to zero. You don't have to set it to zero though, you can use values between zero and 100% and this will simply reduce the negative displacement along Y.

The same of course can be done by changing the positive strength, which will reduce the positive displacement along Y.

## *Coordinate Values (X Minimum/X Maximum/Z Minimum/Z Maximum)*

In order to calculate the Y coordinate of a point, that point will usually have a value for X and/or Z assigned to it in the formula. If the grid length is 200 units, and the resolution is 50 segments, each point will be spaced evenly along the length and width of the grid, around the centre of the 3D Plotter object. Don't forget that 50 segments require 51 points. So the first point would have an X coordinate of -100 and the 51st point would be at X +100. Then the 52nd point would have an X coordinate of -100 again (but with a different Z coordinate), and so on.

It doesn't make sense to use these 3D world coordinate values directly though. For one thing, moving the plotter would alter the X and/or Z coordinates of all the points and you'd get a different curve. More importantly, you could never alter the range of values, and the range limits would always depend on the size and location of the grid.



Instead, each point is assigned an internal value which ranges from the 'X Minimum' value to the 'X Maximum' value, depending on where the point is located in the grid. So with the default settings the first point would have an X coordinate of -100 but the X value used in the formula would be -180; and the 51<sup>st</sup> point would be located at X +100 but the formula value would be +180. The formula values will be the same no matter where the grid is located or what its width and height might be.

You can adjust these settings to whatever you need. As in the 'Out of Range Values' example above, the minimum and maximum values don't need to be centred around the centre of the 3D Plotter object – if you want min/max values of -3 and +1237 you can have them!

For interest only, the default values of -180 and 180 were chosen because when these are slotted into a trigonometric function such as Sin() or Cos(), they are treated as degree values and these values produce a nicer curve than if they were set to (say) -200 and 200 instead.

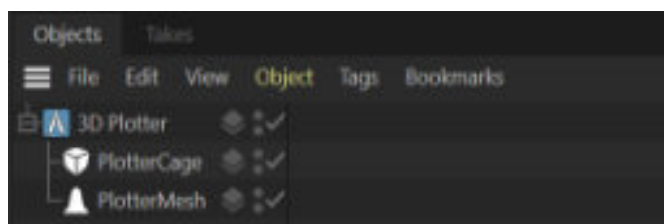
## Separate X & Z Values

If this switch is off, which it is by default, then you only set the X point values and the Z values are set to be the same as the X values. You aren't forced to do this, however. If you turn this switch on, you can set the minimum and maximum Z values independently of the X values. With the default curve, try turning this switch on and setting the Z values to 0 and 1440 and you will see the effect immediately.

## Generate

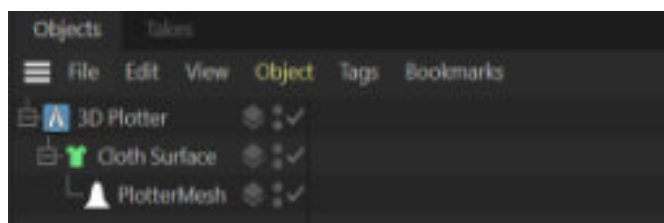
This menu button lets you create one of the additional components for the Plotter. You can select the object you want from the list and the component will be created as a child of the 3D Plotter object. Note that, if the Plotter already has a child object of that type, you will be asked if you want to create another one.

These components are separate objects and can be accessed from the C4D object manager:



The components must be in the hierarchy of a Plotter object but they don't have to be direct child objects of the 3D Plotter. (The reason for being in the Plotter's hierarchy is that in some cases they need to obtain data from the Plotter object, so they need to find their parent Plotter – which they can't do if you have removed the

component object from its position as a child object of the Plotter.) For example, if you wanted to add some thickness to a generated mesh, you could use a C4D Cloth Surface object like this:



This works fine because the Mesh object is still in the Plotter object's hierarchy.

To remove a component, simply delete it in the object manager. To disable a component, click the tick switch in the object manager to turn it off; click again to turn it back on.

## Reset to Defaults

Click this button to reset the interface to the default values.

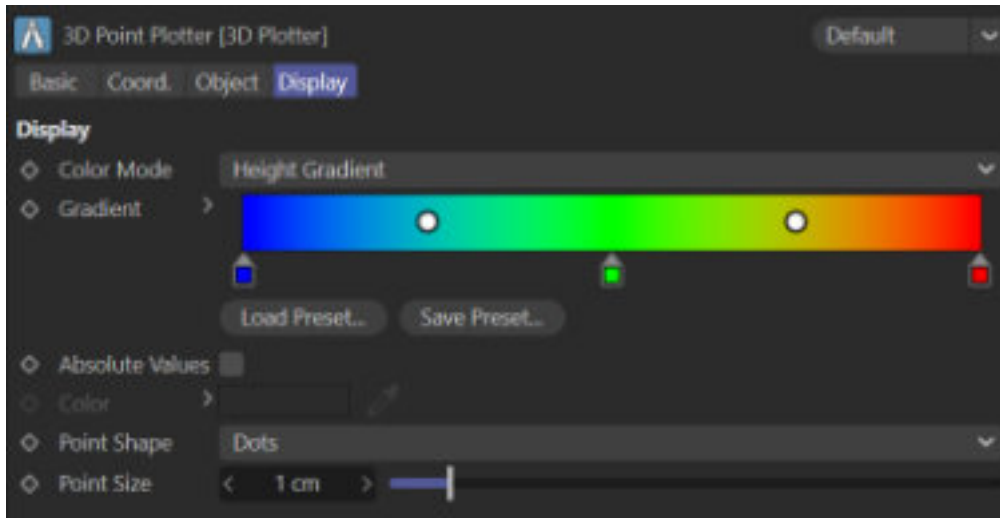
## Open Manual

Click this button to open this manual,

# 3d point plotter

## The Plotter: Display tab settings

This is the tab interface:



This tab controls the display of the points in the viewport.

### *Color Mode*

This menu selects between two modes. The default is 'Height Gradient'. In this, points with the lowest Y coordinate value take their colour from the left hand end of the gradient; those with the highest Y value have the colour at the right hand end. Negative values are taken into account, so a value of -50 is lower than +50 (fairly obviously!) and would result in different colours being used. However, this changes if the switch 'Absolute Values' is turned on – see below.

The second option is 'Single Color'. This lets you choose one colour for all points, which you can do from the 'Color' control.

### *Gradient*

The gradient to use when 'Color Mode' is set to 'Height Gradient'. You can change this as required.

### *Absolute Values*

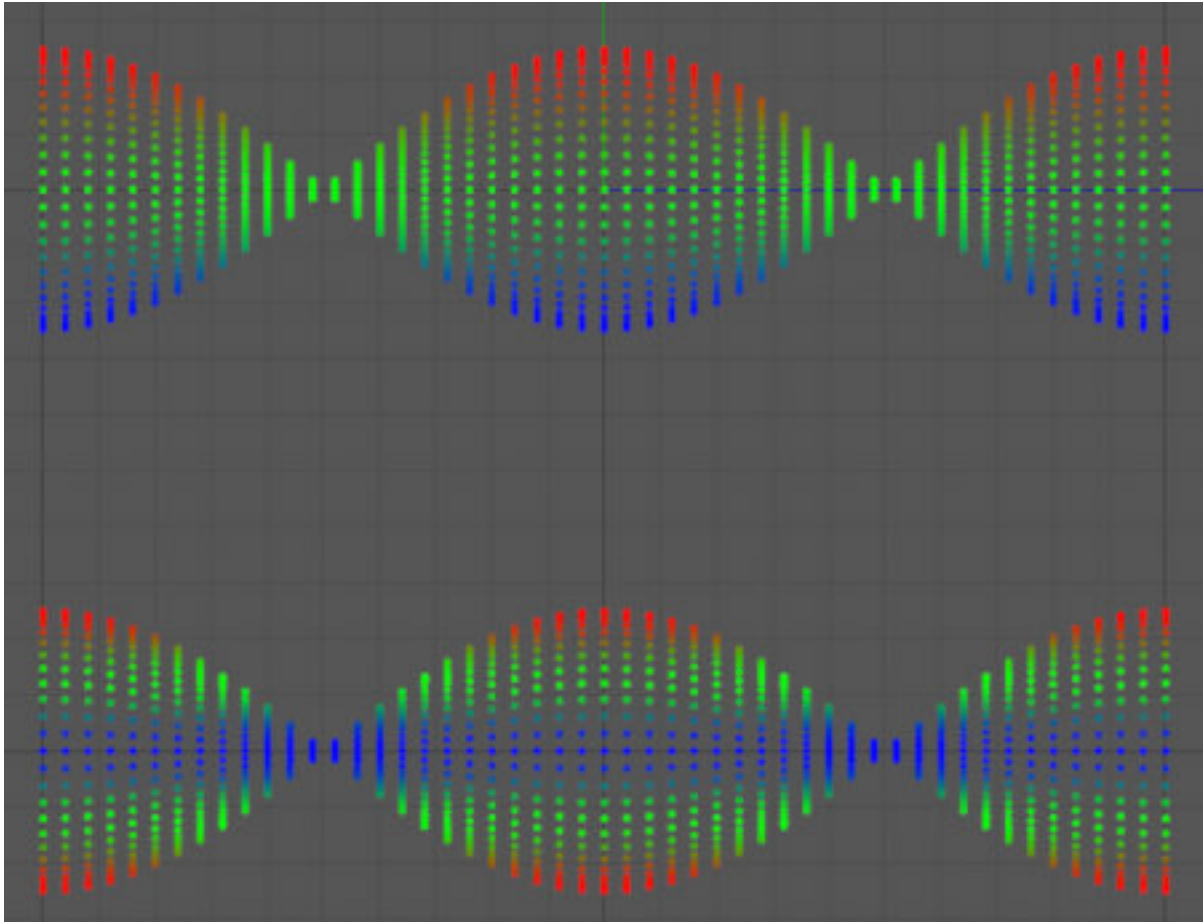
This switch is only available if the height gradient is used. In this case, the difference between negative and positive Y coordinate values are ignored and all values are treated as positive. This means that -50 and +50 are treated in the same way and would get the same colour.

The difference can be seen in the screenshot below. The top plotter has the 'Absolute Values' switch turned off, which is the default setting. You can see that the lowest points are coloured blue, from the left hand end of the gradient; the highest points are coloured red.

In the lower plotter, absolute values are turned on. Now both the lowest and highest points are coloured red, because both have the greatest Y coordinate values once the negative sign is ignored. The blue points are those closest to a Y coordinate of zero, which is the lowest possible value when absolute values are used.



# 3d point plotter



## *Color*

The colour to use when 'Color Mode' is set to 'Single Color'.

## *Point Shape*

This menu gives three options for the points display, Dots (the default), Spheres, or Ticks. These shapes don't render so they are purely for the viewport display.

## *Point Size*

The size of the shape in the viewport (this does not affect the size of any point objects, which are controlled in the Mesh object). Again, this is for the viewport display only.

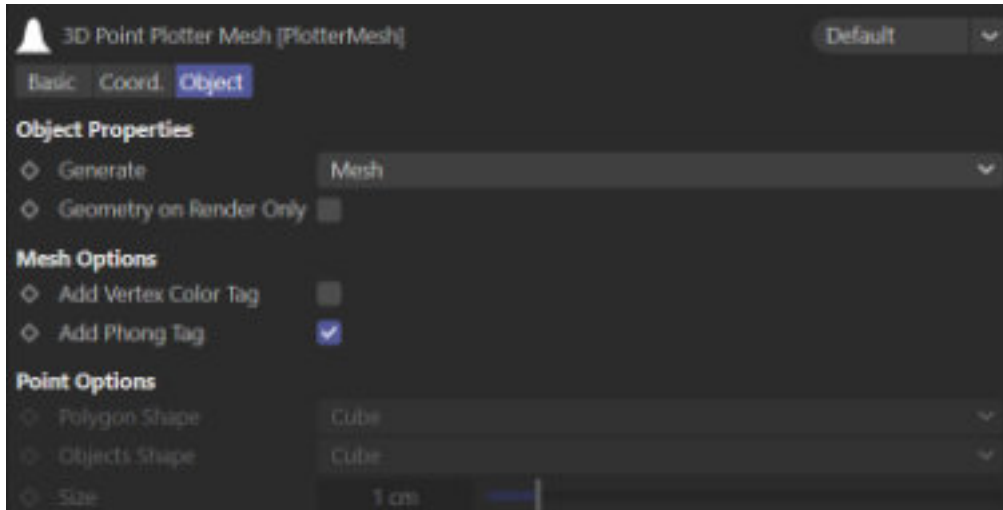
## The Plotter Mesh Object

This component generates a polygon mesh from the points. There are several different choices for the type of mesh, as explained below. The mesh itself is fully procedural and can be deformed with C4D deformers in the usual way.

The interface looks like this:



# 3d point plotter



## *Generate*

This menu controls what will be generated. The options are:

### *Mesh*

The default setting; a polygon mesh is generated from the points. The mesh is simply a flat grid of polygons and is always constructed the same way – that is, the same points are always used to build the same polygon. No account is taken of how the point grid might be distorted by deformers.

### *Points (Polygon)*

A small object is generated for each point in the grid. These are all part of a single polygon object. This lets you render the points if required. There are some limitations on the shape of each point, which you can choose from the 'Polygon Shape' menu. These shapes are:

- Cube (a simple cube shape, 6 polygons per point)
- Pyramid (a pyramid shape, 8 polygons per point)
- Icosahedron (an icosahedron, each point using 20 polygons)

### *Points (Objects)*

This also generates a small object for each point but in this case each point object is separate, one for each point. The available shapes are the basic Cinema 4D primitive objects and there are currently four implemented. You can choose the shape from the 'Objects Shape' menu which has these options:

- Cube
- Sphere
- Pyramid
- Icosahedron (a Platonic primitive of Icosahedron type)

Which option should you choose for generating geometry from the points? The pros and cons are shown in this table.

# 3d point plotter

Feature	Points (Polygons)	Points (Objects)
Speed of generation	Faster	Slower
Animation speed	Faster	Slower
Effect on viewport	Very little	May slow the viewport
Sphere shape available	No (use Icosahedron)	Yes
Colour each point individually	No	Yes

As you can see, the main advantages to using 'Points (Objects)' is that you can generate true sphere shapes but especially that each object will be coloured using the colour of the point with which it is associated. This is not possible with 'Points (Polygon)' since only a single object is produced. In both cases, of course, you can texture the objects with whatever material you like.

## *None*

The final option is that nothing is generated; this can be useful if you find that you are generating a very polygon-heavy mesh and you want to turn it off temporarily. Or you could just disable the Mesh object itself, of course.

Note: the point shapes selectable in the Plotter's Display are only used for the viewport display. They are not rendered; if you need to render the points, use the Mesh object and one of the 'Points' modes to generate the required geometry.

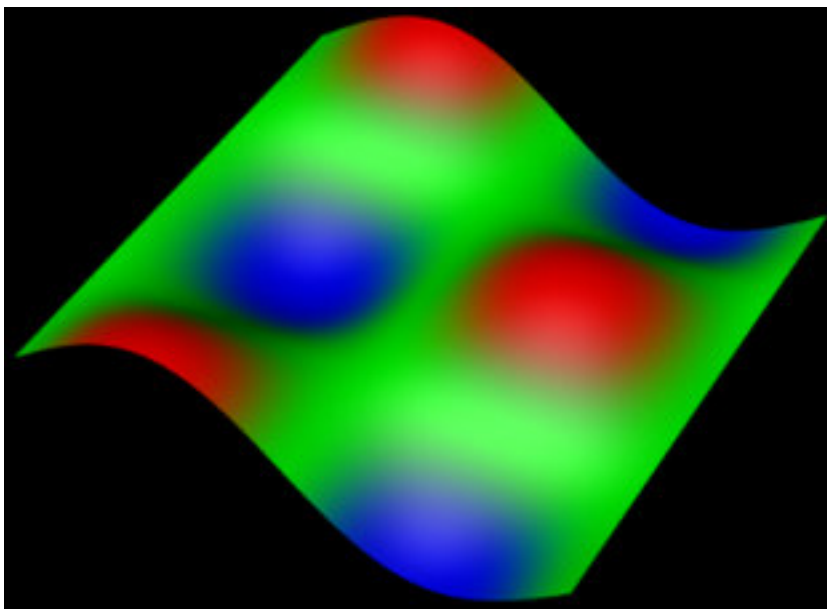
## *Geometry on Render Only*

If this switch is turned on, the mesh and/or point objects will only be generated on rendering to the Picture Viewer (not to the C4D editor).

## *Add Vertex Colour Tag*

When checked, this will add a vertex colour tag to the generated mesh. The switch is only available if the Generate menu is set to Mesh.

The vertex colour tag uses the colour from the point corresponding to each vertex in the mesh. You can then use the tag in a Vertex Map shader. With the default settings the rendered result looks like this:



# 3d point plotter

## Add Phong Tag

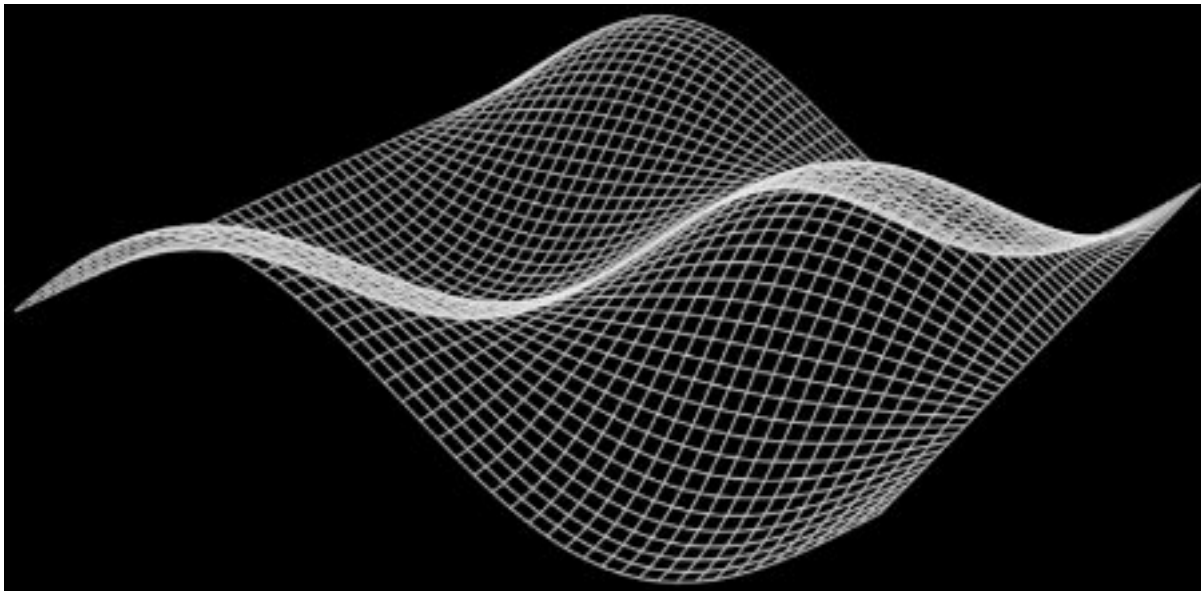
This switch is on by default and adds a Phong tag to the mesh if the 'Generate' menu is set to 'Mesh'. If you don't want that for some reason, turn this switch off.

## Size

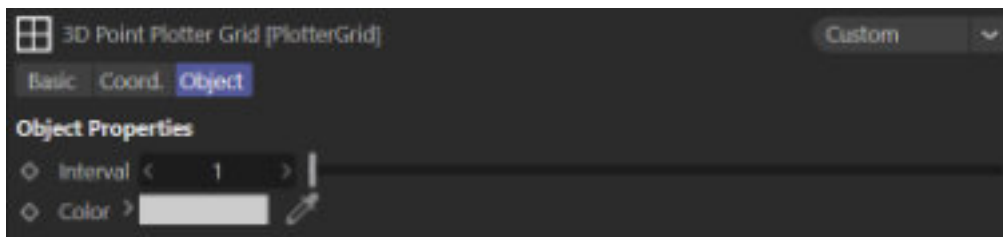
This setting is only used if the Generate menu is set to 'Points (Polygon)' or 'Points (Objects)'. It sets the size of the objects which are generated.

## The Plotter Grid object

This component, instead of a mesh, creates a grid of splines from the point positions. Like the mesh, the grid is always constructed from the same points, regardless of any distortion. This is a typical result (rendered in this case with the Hair material):



The interface for this component is very simple:



## Interval

This setting lets you build a grid with fewer splines than if you used all the points in the grid. The default setting is 1, which means that a spline segment is added for every row of points. If it is set to 2, every other row of points is used, if 3, then there are two 'gaps' between each spline segment, and so on. It's easier to try it out than to have it explained!

## Color

This the spline display colour and you can see it in the Basic tab of the Grid interface. It is the colour used for the spline in the viewport and depending on what method you use to render the spline, this can also be the rendered colour. You can use the Hair material to render the splines



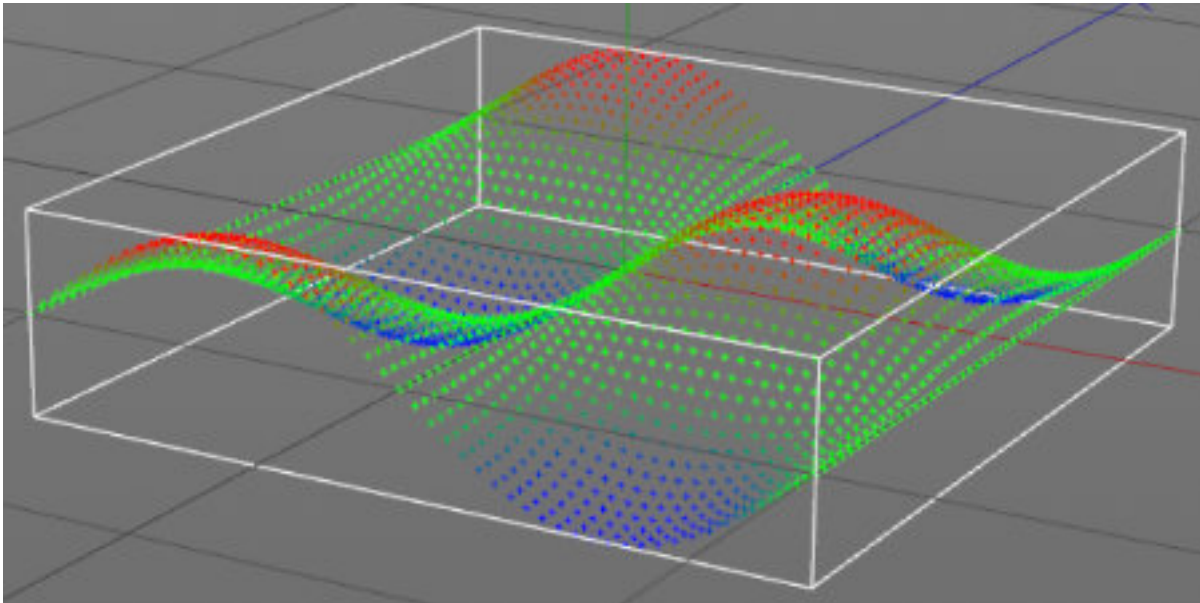


instead, but that doesn't use the display colour so this is ignored by that material.

## The Plotter Cage object

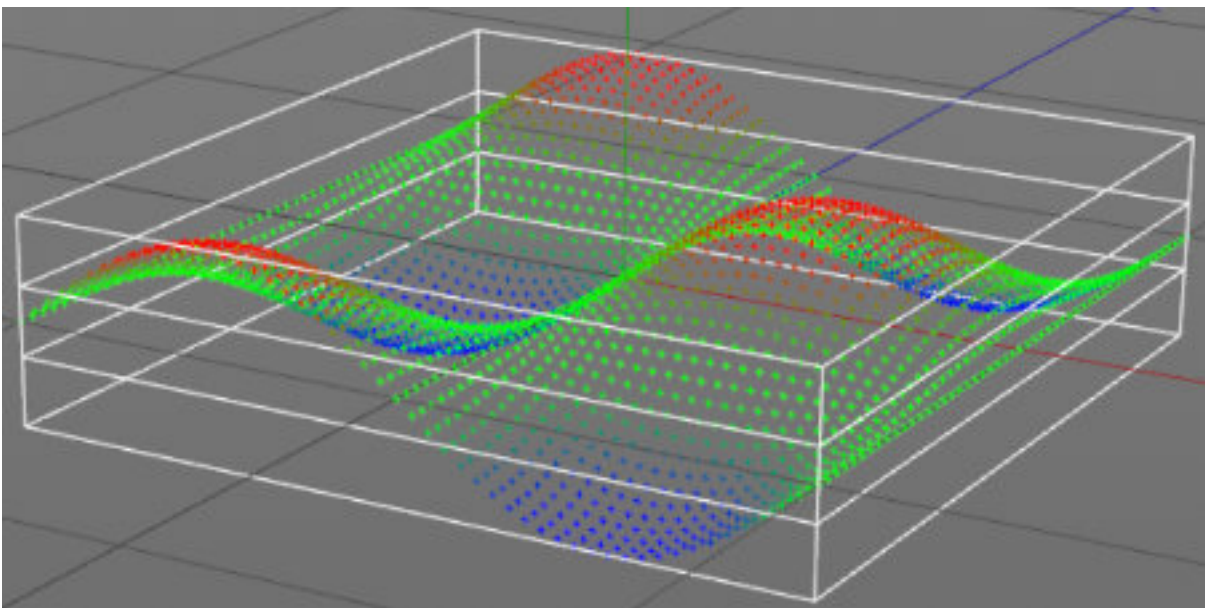
This component draws a 'cage' around the point grid. At its simplest this is just a wireframe box drawn round the points. It uses the dimensions (width, length and height) from the Plotter object to set the size. With the default settings you would see this:

The interface is simple:



### *Intermediate (XZ)*

As you can see from the screenshot above, the cage is just a single wireframe box. You can add more divisions, however. If you set this value to 2, you would see this:

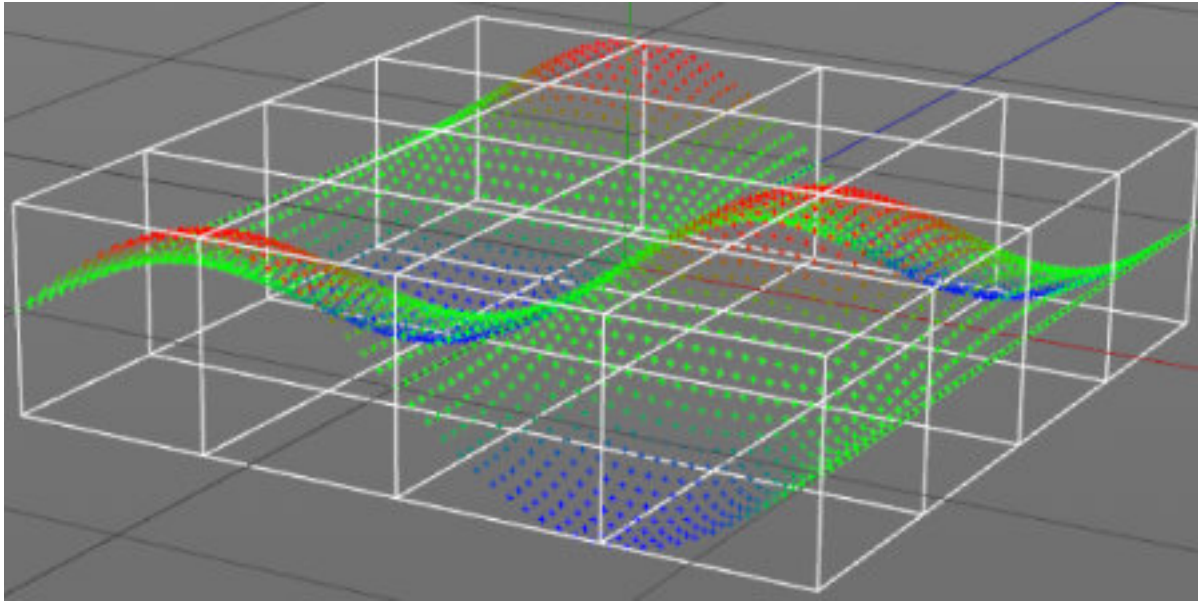


This adds extra divisions on the XZ plane of the plotter.

### *Intermediate (XY/ZY)*

This is the same but for the XY and ZY planes. Setting this value to 3 would give this:

# 3d point plotter



Of course, you can combine the two and have divisions in all three planes, as desired.

## *Color*

The display colour of the cage splines. As with the Grid object, depending on what method you use to render the spline, this can also be the rendered colour

## The Plotter Labels object

The final component lets you add labels to the plotter. These are actually Extrude objects with a Text spline child object, so they will appear in any rendered image.

Note that the Labels object generates *all* the labels all the time, but simply sets the editor/render dots in the object manager to control which ones you see and render.

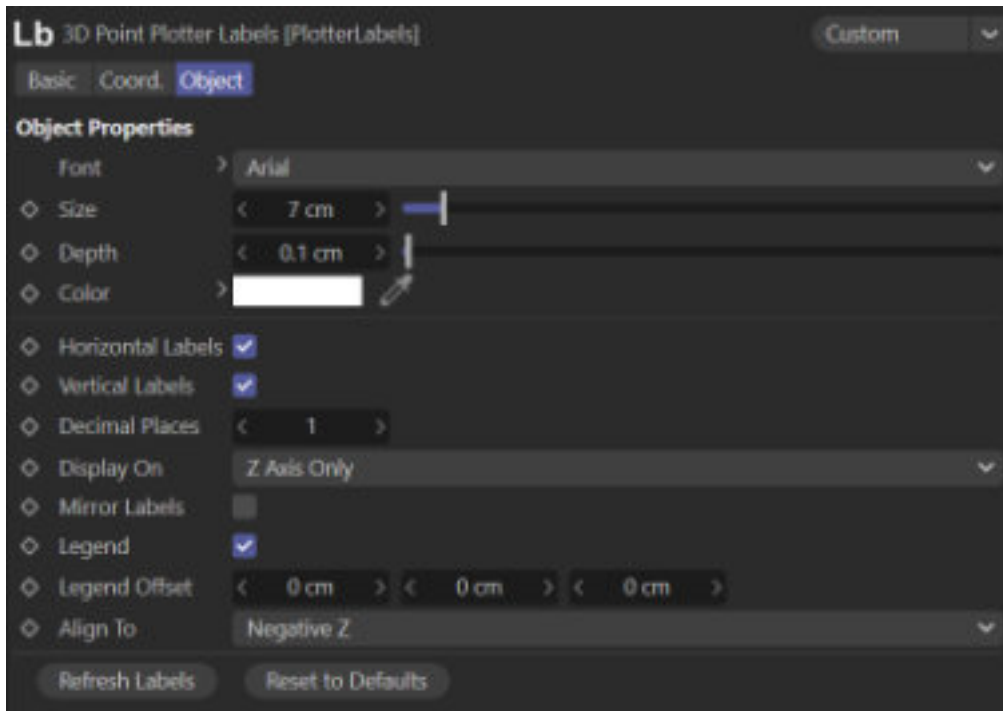
Note that adding labels does not impact the viewport performance but it does slow down the scene playback. You can disable the Labels object in the object manager to turn them all off during scene editing.

**Important: DO NOT rename the Extrude objects which are child objects of the Labels object!** The Labels object uses the name to find the relevant label to show or hide it, mirror it, move it and so on. If you rename the label, the Labels object will not work correctly. The most likely thing is that when the scene is played you will get a duplicate of the label because the Labels object will assume that the label does not exist, so recreates it.

This is the object's interface:

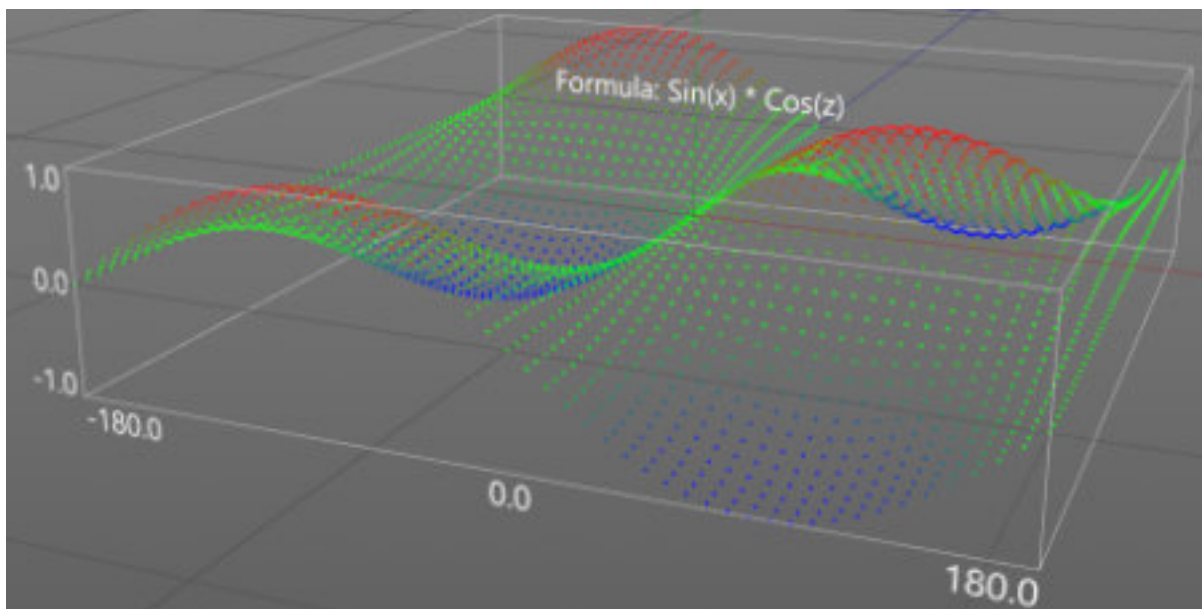


# 3d point plotter



(You may wonder why the Extrude/Text spline combination is used for labels. Wouldn't MoText be simpler and give the same results? Well, it would but for one thing: it isn't possible to set the colour of a MoText object by setting the Display Color in the object's Basic tab. You can do this, however, with the Extrude object.)

With the default settings this is what you would see in the viewport:



As you can see, there are horizontal labels (running from -180.0 to 180.0 in this case) which you will recognise as the Maximum X and Z values from the Plotter interface.

Then you have vertical labels representing the calculated Y values. You see that these run from -1.0 to 1.0 in this example, because sine and cosine values can never exceed 1.0 or be smaller than -1.0. Other formulae would show different ranges.

There is also a legend above the cage showing the actual formula.





# 3d point plotter

## *Font*

The font used for the labels.

## *Size*

The size of the label text. The default setting works well enough with the default size grid but you may need to increase or decrease it if you alter the size.

## *Depth*

The 'depth' of the label - how 'thick' it is along the object's Z-axis.

## *Color*

You can set a colour here to colour all the labels. Alternatively, you can add whatever material you like to the Labels object and they will all render with that material, or you can attach different materials to different labels (that is, to the individual Extrude objects).

## *Horizontal Labels*

Turn off this switch to remove the labels on the horizontal (X and Z) axes.

## *Vertical Labels*

Turn off this switch to remove the vertical (Y axis) labels.

## *Display On*

To avoid cluttering the appearance, the axis labels are only displayed on one side of the grid. By default the labels face the 3D world negative Z axis, since unless the camera or object has been rotated this is the side which faces the default camera. However, you can alter that with this menu. The options are:

- Z Axis Only – the default settings, the labels face negative Z
- X Axis Only – the labels face the positive X axis
- Both X and Z Axes – there are two sets of labels, one facing -Z and the other +X

## *Decimal Places*

The number of decimal places to show in the horizontal and vertical labels.

## *Mirror Labels*

Since you may want to rotate the object, or the camera, or both, the labels could then face away from the camera, which isn't very helpful. In some cases you can use the 'Display On' menu to correct this, but if that doesn't do what is needed, turn on this switch. The axis labels which faced the negative Z axis will then face the positive Z axis. Likewise, if you have altered the 'Display On' setting, using this switch will mirror the displayed labels according to the chosen axis. The vertical labels are also mirrored with this switch but the legend is not (use the legend's 'Align To' menu for this).

## *Legend*

Turn off this switch to hide the legend.



### *Legend Offset*

The legend is placed just on top of the grid and in the centre of the X and Z axes. If you need to move the legend, simply alter these values so that it is positioned where you want it.

### *Align To*

This is the counterpart for the legend to 'Mirror Labels' for the axis labels. By default the legend points towards the negative Z axis. But if, as a result of object or camera rotation, the legend is no longer readable, you can use this menu to realign the legend to one of the other faces.

### *Refresh Labels*

If you need to regenerate the labels for some reason - perhaps you deleted one or more labels in the object manager by accident - or if the labels don't display as you think they should, you can reset them by clicking this button. This will delete the existing labels and regenerate them using the settings in the Labels object.

### *Reset to Defaults*

Clicking this button will reset the Labels object settings to their default and regenerate the labels.

## Rendering the result

So, you have a good formula and you've added all the required components and it looks good in the viewport. How do you render it all?

As discussed above, you can't render the viewport display directly but you can generate a small object to represent each point. You do this with a Mesh object set to one of the 'Points' modes. Don't forget that if you use 'Points (Objects)' mode, each object will have as its display colour the colour of the underlying point.

If you generate a Mesh object using 'Mesh' mode, you can add whatever materials you like to the generated mesh. Remember the trick of adding a vertex color tag though, if you want to render the mesh using the colours of the points.

Rendering the Grid or Cage objects requires rendering the splines in some way. You could use a Sweep object to generate renderable geometry, for example. Alternatively, you could render the splines directly. There are inbuilt and third-party options for this. In Cinema 4D itself, the Hair material is a very good way to render splines, although this does not use the spline colour from the Cage or Grid objects so you will have to set that yourself. There are also several third-party objects and render engines which can render splines.

Finally, the Labels object produces geometry which can be rendered. Since it uses Extrude objects, the labels will render in the colour assigned to them in the Labels object, unless you use your own material to override this, of course.

## Formulae

What can you put in the formula? Cinema 4D has a rather good formula parser with plenty of functions, operators, and constants to generate many different curves. You can find the complete list of formula arguments in the C4D help. The list is in the help appendix under 'Formula'. If you can't find it, type 'formula' into the search bar in the help, and you'll get a number of hits – one of them is the argument list.



# 3d point plotter

The Plotter also provides some additional arguments that C4D does not. These are as follows:

Argument	Meaning
x	The value of the internal x axis value (not the actual physical coordinate, the internal value generated from the X Minimum and X Maximum values)
z	As for 'x' but for the internal z value
t	The time of the animation in seconds*
f	The current frame of the animation*
r	The grid resolution
h	The grid height

\* = ignored if the 'Enable Time Parameters' switch is turned off in the Plotter interface

Sometimes you can enter what you think is a valid formula but nothing happens and the points remain flat with no Y displacement. Possible causes are:

- you entered an argument that doesn't exist – for example,  $\sin(x) * \cos(z) * y$  won't work, because the 'y' argument doesn't exist, so the formula can't be parsed
- you misspelt an argument, which has the same effect as entering a non-existent one, e.g.  $Ab(x)$  instead of  $Abs(x)$
- and by far the commonest mistake, you entered the wrong number of parentheses – for example, this:  $\sin(x)) * \cos(z)$  will fail because there are too many ')' characters

The missing/extra parentheses error is very easy to make because typically the font used in the text entry fields is small and the parenthesis characters are difficult to see, so if your formula doesn't work that's the first thing to look for.

## Summary

This plugin provides what you need to plot mathematical formulae in 3D space and render the results. It could be useful, for example, in visualising maths formulae or simply for making pretty pictures. The choice is yours.

## Contact

Comments, suggestions and (hopefully not too many!) bug reports are welcome. You can contact me through my website at <https://microbion.co.uk/html/contact.htm> and the latest version of the plugin can be obtained from <https://microbion.co.uk/html/plotter.htm>.

Steve Pedler  
May 31<sup>st</sup> 2022