An Introduction to Scripting with CalcPlot3D

This tutorial will help you begin using and creating scripts for **CalcPlot3D**. Scripts allow you to save your work in CalcPlot3D, particularly when there are multiple related views you wish to save in a particular sequence. Note that it is also possible to simply save a single view to a custom URL. Each of these options has its uses.

You can use scripts to create: **Classroom Demonstrations**
Guided Student Exploration Activities

You can use a custom URL to create**:** **A plot to share with a student or colleague**
A link with your own default initial settings
A plot to show in class, on a website, or as a OER textbook figure

Classroom Demonstrations: It is often useful to demonstrate how to enter functions, space curves, etc. in CalcPlot3D during class, but sometimes it would be nice to quickly step through a series of visual examples, each of which can be manipulated and explored as much or as little as time allows and you desire.

Guided Student Exploration Activities: Students gain the most understanding from playing with and exploring the concepts visually themselves. The more directly students can be engaged in the visualization and exploration of calculus concepts the more they will internalize and the clearer will be their geometric intuition.

A plot to share with a student or colleague: A custom URL can be used to share a link to the CalcPlot3D app that opens with your particular functions, graph settings, and 3D viewing window. You can use this feature to paste the link in an email to share a quick example with a student, to show an interesting plot to a colleague, to report a bug in CalcPlot3D, or even to create an interactive figure for an online textbook or webpage. You may also choose to use one or more of these saved plots as part of your classroom instruction.

Your own default initial settings: It's worth noting that you can use a custom URL to simply set the default startup settings you wish to have always in place when you start the app in your browser. Just create the view you want and use this as your link to open CalcPlot3D. Note that you can save these special URLs in Favorites on your browser.

Custom URLs load a single view into CalcPlot3D when it initially opens. Scripts are loaded by the user from the app once it is already running, using the Scripts submenu of the CalcPlot3D main menu. For a limited number of pre-made scripts stored on the CalcPlot3D server, there is also a way to get the script to load automatically using a tag in the URL (e.g., the script we'll start with in our first activity below).

Typically, we will use CalcPlot3D to create a script by saving views one at a time to form a desired sequence of steps. It's even possible to choose at which step to add new ones and we can delete unwanted steps. Although this process should allow us to obtain most of what we want from within CalcPlot3D, it can still be very useful to edit the script in a text editor to quickly reorder the steps and/or to fine tune the settings without having to juggle steps or create a new script. Editing the script is the only way to add text instructions or descriptions to your steps, and there are even a few scripting parameters that can only be adjusted in the script itself.

CalcPlot3D is still being developed, so you may find some features are not yet saved in a script. Please let the presenter know if you have requests for something that does not save or if you discover errors in CalcPlot3D. Saving to a URL using the Encode View in URL option on the File menu and sharing this URL is a great way to demonstrate a bug and share it with us as it makes the bug easy to see and hopefully fix.

Part I. Now let's take a look at an example script. We will start with a demonstration script since it will give a general picture of what can be done with scripting in **CalcPlot3D**, without including as much text as an exploration script might.

1. Enter the following URL in your browser, noting the short query string at the end. This loads the example script from the CalcPlot3D server. If you already have CalcPlot3D open, just add the last part from the question mark to the end.
   https://c3d.libretexts.org/CalcPlot3D/index.html?example=DomainDemo
2. Exploring the current version of the script, you'll see that it contains a series of examples of functions of two variables with interesting domains. Now open the main menu of CalcPlot3D, select the **Scripts** submenu and then select **Save script**. You should see the file **DomainDemo.txt** saved to your computer.
3. Locate this file and open it in your simplest text editor, such as Notepad (Windows) or TextEdit (Mac). Below is a simplified version of the first step of this script.

```
<init steps="6">

   <step 1>
      <function>
            type="z=f(x,y)"
            z="(x+y)/(x-y)"
```

Init must come first and must specify the number of steps in the script. Note the steps all start with <step #> and end with </step>. This is a form of XML.

```
            umin="-2"
            umax="2"
            vmin="-2"
            vmax="2"
```

These values set the specific viewable domain for this function, with u = x, and v = y. If these values are different from the ranges specified in the window element, then the surface will stay fixed with the specified domain even if the zoom level of the plot is changed.

```
            grid="30"
            alpha="-1"
            format="normal"
            view="0"
            constantColor="rgb(255,0,0)"
            visible="true"
      </function>
      <window>
            hsrmode="3"
            nomidpts="true"
            anaglyph="-1"
```

These three properties determine what hidden surface mode is used, if each rectangle in the grid is broken into triangles at its midpoint, and if a 3D glasses mode is used.

The window object contains all the properties that format the whole plot.

```
            transparent="false"
            alpha="140"
            twoViews="false"
            unlinkViews="false"
            axisExtension="0.7"
            xaxislabel="x"
            yaxislabel="y"
            zaxislabel="z"
            xmin="-2"
            xmax="2"
            xscale="1"
            xscalefactor="1"
            ymin="-2"
            ymax="2"
            yscale="1"
            yscalefactor="1"
            zmin="-2"
            zmax="2"
            zscale="1"
            zscalefactor="1"
            zminClip="-4"
            zmaxClip="4"
            zoom="0.7"
            edgesOn="tr
```

Sets the scale of the 3D plot.

```
            facesOn="true"
            showBox="true"
            showAxes="true"
            showTicks="true"
            perspective="true"
            centerxpercent="0.5"
            centerypercent="0.5"
            rotationsteps="30"
            autospin="true"
            xygrid="false"
            yzgrid="false"
            xzgrid="false"
            gridsOnBox="true"
            gridPlanes="false"
            gridColor="rgb(128,128,128)"
      </window>
      <viewpoint>
```

The viewpoint can be set with either rectangular or spherical coordinates.

```
            center="7.76,-4.22,4.69,1"
            focus="0,0,0,1"
            up="-0.275,0.443,0.853,1"
      </viewpoint>
   </step>
```

Center sets the location of the eye.
Focus sets the center of the view.
Up sets the vector for the up direction.

4. Now let's go back to the CalcPlot3D app to look at the steps of this script in action.
   Click through the 6 steps using the Next button on the exploration box located at the right side of the screen. This script is meant to be used in class as a visual demonstration after specifying and drawing the domains and ranges of various functions with some limits on their natural domains.

5. You may notice that the plots in the steps at the end should be placed with their corresponding surfaces from earlier in the exploration. These steps were recorded using the app, and similar to how a macro works in Excel or Word, we can edit them after they have been recorded. Let's reorder the steps in the script in your text editor so that they show each surface first and then its view from above the *xy*-plane. Be sure to copy and paste the entire step for each one. Then you can renumber the steps, if you like. The actually step numbers are ignored by the app. They are just there for our benefit.

6. Save the script to your computer. Now let's load it into CalcPlot3D to check that it works and has the correct order. In CalcPlot3D, select **Load a script** from the Scripts menu and then locate this text file.

7. Now let's add some text to the exploration box for a step. In the text editor, add a line just after <step 1> and just before <window>. On this line, enter:

   <text> This is an interesting discontinuous function of two variables.
   It's function is $ f(x, y) = \frac{x+y}{x-y} $.</text>

   Note that everything between the two delimiters <text> and </text> will be displayed on the exploration box, including spaces, carriage returns, and empty lines. Word-wrapping should work well and you have the option of using LaTeX to format math in your text using $ (in-line math) or $$ (display math) as delimiters. In fact, many basic HTML commands should also work including <p>, <b>,
,<ol>, <li> and <ul>.

8. There are two ways we can add a new example to this script. We can either add it in CalcPlot3D or we can add it in the text editor. First let's add a step using the text editor. Copy an entire step showing a normal view of a function, like step 1. Be sure to start at the <step 1> and end by including the </step> statement. Paste this step at the end of the script.

9. Go back to the top of the script and change the number of steps to 7. Otherwise the step we just added will be ignored by the applet.

10. Now edit the function in the new step to be **arcsin(y – x)**. Edit any other parameters you wish to and then save the script and load it into CalcPlot3D to try it out.

Next we will add a step in CalcPlot3D.

11. First let's create the view we want in the app. Clear the functions shown and enter the function, **z = 1/(y – x^2),** as the first function. Press Enter when you finish, or select the checkbox next to function. Adjust the viewpoint using the mouse and make any changes to how the surface appears using the **surface settings** menu (click the gear symbol ⚙ to find this menu).

12. Now from the **Scripts** menu in the CalcPlot3D menu, choose the option to **Add current view to active script**. After a second, you should see the plot window briefly show the first step of the script and then it should display the step you just added. Notice that there is now an additional step (8 steps total if you

added the step in the text editor above).

13. Next we should save the script. Scripts are not automatically saved when you add a new step. Be sure to save it, if you wish to keep your additions. To save the script from CalcPlot3D, select the **Save script** option on the **Script** submenu. Each time it is saved, it is downloaded like any other file in your browser, likely placed in your **Downloads** folder.

14. Open this new script in your text editor. I usually either double-click on the script file or I right-click it and select the editor I wish to use. Check to see that the new step is there and notice that it lists all parameters, line by line.

15. Now try changing the functions in the script. You can also vary the window and viewpoint parameters, if you wish. You might try changing the third step function to **ln(y - x^2)** for example. Save the script after each change and load it again into CalcPlot3D.

Part II. Next we will create a script from scratch using the applet and edit it in the text editor.

1. The purpose of this script is to examine a series of examples of surface intersections. The first thing we need to do is exit the earlier exploration. We can either reload the page (easiest) or use the Exit option on the Exploration control panel. Now we need to set up the first example for our script!

2. Let's graph the surfaces defined by: $z = x^2 + y^2$ and $y = x$. To enter the second function, you can select the option, **Function: y = f(x, z)** from the Add to graph dropdown menu.

3. To display more of the plane, click the **Format Axes** button. This will reveal the **Format Axes** dialog options. Use these to change the **z-max** value to **4**. This will automatically raise the **zClip-top** value to **8**. Change this value back to **4** and press enter to make these settings take effect. Then click the **Format Axes** button again to hide these options.

4. Make the surfaces appear semi-transparent by clicking on the transparency button (or by typing Ctrl-T). This will make it easier to see the intersection.

5. Now it is time to create our new script with this as its first step. Choose the option to **Create new script with current view as 1st step** on the **Script** submenu in CalcPlot3D. You will be asked for a script name. A default name of **myScript** is supplied. Let's call it **Intersections**. When you click OK, nothing special seems to happen, except that the exploration box should appear to the right, and the first step is recorded.

6. Next see if you can determine a space curve that will trace this intersection. Hint: Let $x = t$. Graph this space curve to verify it goes through the intersection. Add this view as a second step in the script.

7. Next clear the plot and let's create the intersection of the surfaces defined by: $x^2 + y^2 = 4$ and $z = x^2$. Here you may wish to solve the first equation for x or y and graph the cylinder in two pieces, or you can use cylindrical coordinates or an implicit surface equation to graph it with just one. (To graph in cylindrical coordinates, select **Function: r = f(θ, z)** from the **Add to graph** dropdown menu.)

   Add this view as a third step in your script.

8. Next see if you can determine a space curve that will trace this intersection. Hint: Let $x = 2\sin t$. Graph this space curve to verify it goes through the intersection.

9. Add this step to the script and then **save the script**.

10. Close the CalcPlot3D app and then open it again and load in your script to see if it worked!

This concludes this part of the tutorial. Now try to create an exploration or demonstration script of your own! You can try any of the following ideas or come up with something of your own.

An exploration/demonstration of:

- velocity and acceleration similar to the one I have provided in the scripts folder.
- functions in spherical and/or cylindrical coordinates.
- Parametric surfaces
- More intersections of surfaces
- Lagrange Multiplier Optimization
- Level Surfaces
- Gradient vectors for functions of 2 and/or 3 variables.
- Taylor polynomials of functions of 2 variables

Try the following built-in scripts. Save them to your computer with the Save Script menu option to explore how they add text to the steps.

Velocity-Acceleration Exploration: https://c3d.libretexts.org/CalcPlot3D/index.html?exploration=Velocity-Acceleration
Lagrange Multiplier Exploration: https://c3d.libretexts.org/CalcPlot3D/index.html?exploration=LagrangeMultExp1
Cycloid Example Script: https://c3d.libretexts.org/CalcPlot3D/index.html?exploration=Cycloids

Appendix: Some additional scripting example code that you may find helpful.

Setting the View Point

<viewPoint>	These parameters set the viewpoint. Each should be entered as a 3D point.
center	Center sets the location of the eye in rectangular coordinates. (ex. center = “(5, 0, 0)”]
focus= "(0,0,0)"	Focus sets the center of the view.
up= "(0,0,2)"	Up sets the vector for the up direction.
eyeR="10" theta phi	If center is not specified, these three parameters can be used to set the viewpoint using spherical coordinates. If center is specified above, these three parameters are ignored.
</viewPoint>	

Examples:

```
<viewpoint center = "(10, 0, 0)" focus = "(1, 0, 0)" up = "(0, 2, 0)" />
<viewpoint eyeR = "10" theta = "pi/2"  phi ="pi/4" />
```