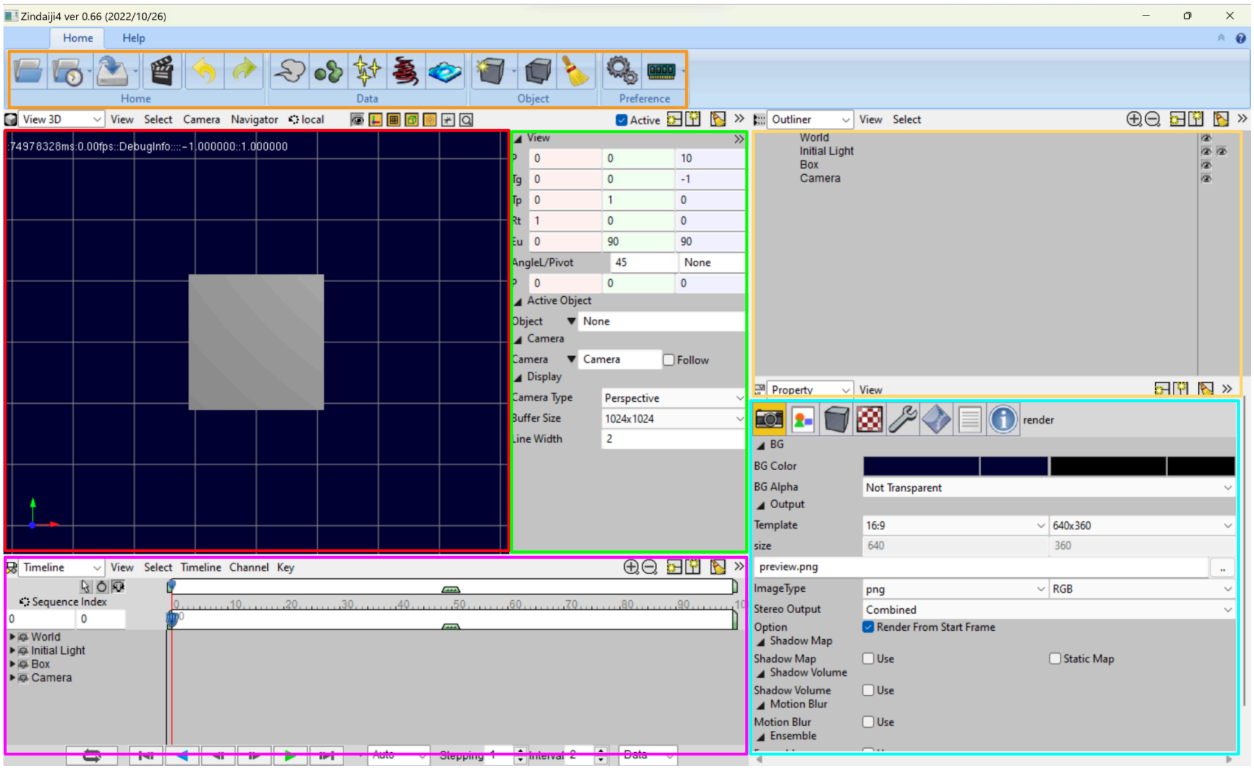
Zindaiji Visualization Manual

Part 0 - Panels



The above is what the Zindaiji interface will look like when you open it. The UI consists of 5 main panels.

The “View 3D” panel is the one in the top left which is where you will view the different timesteps of your simulation. As of right now, the only object in that panel is the box. However, this will change when you input the data which plots each particle in 3D space.

The panel to the right of this is where you control your view/angle at which you look at the object in the View 3D panel. For starters, you can look around in the view 3D panel by dragging with your mouse. You should see that this changes the numbers in the panel too. Alternatively, instead of looking around with your mouse, you can type numbers into the panel directly. The row you'll use most is the top one: "P". You can also use this panel to control the camera settings.

This panel is for controlling which object in the simulation you are changing the settings for. In its default state, the objects are "world," "initial light," "box," and "camera," however, there will also be an object called "particle" when you upload your data. You can hide these objects in the View 3D panel by clicking the eye icon.

This panel is where you will do most of the work. Here, you will upload your data, change the simulation settings, add color coding, and export the final simulation.

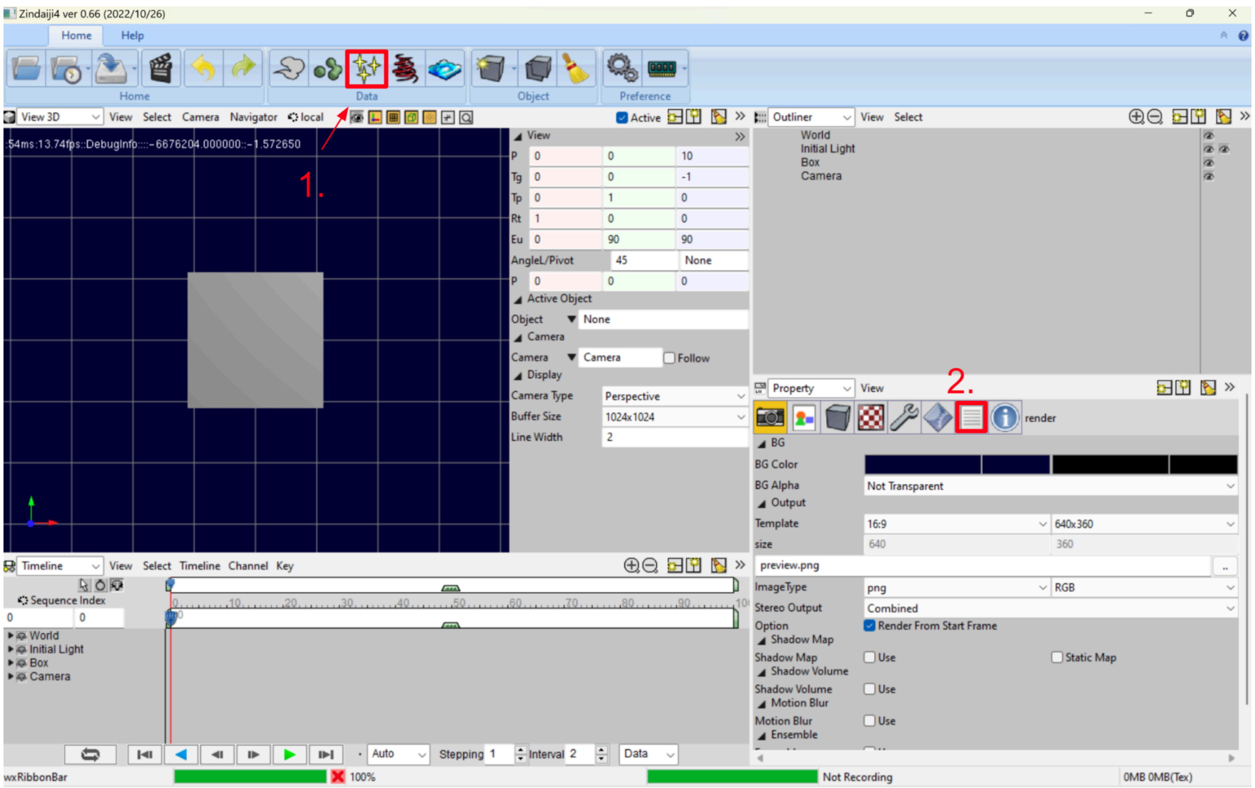
This panel is the timeline. It's where you can skip to different timesteps in the simulation. Additionally, here, you can control the camera settings depending on the timestep.

This is your toolbar. Hovering over each icon will tell you what does, however the ones you will use most are the first three for saving the simulation file, as well as the scene button next to them for controlling how many timesteps the simulation runs. Additionally, you will use the 3-star button for adding a particle.

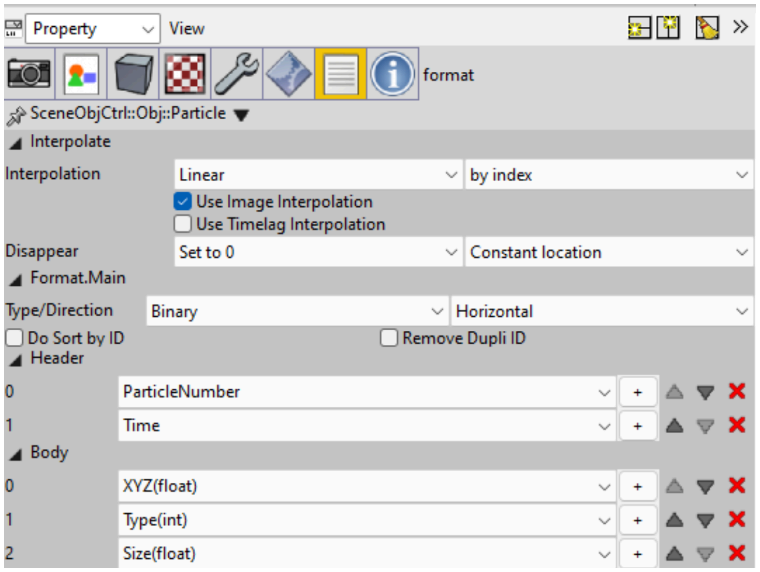
Part 1 – Inputting Data

The first step is to input the data the Python program was used to write.

1. Click the 3-star button at the top labeled “Add:Particle”
2. Click the text icon in the bottom right panel



When you click on the text icon, you should see the following settings in that panel:



First, you are going to want to change the interpolation settings. Interpolation means the simulation will guess where each particle should be in between timesteps and create an additional frame to make the animation look smoother. However, the default "Linear" setting is glitchy with this data set. So, using the drop down menu, **change it to "Step"** which runs much smoother.

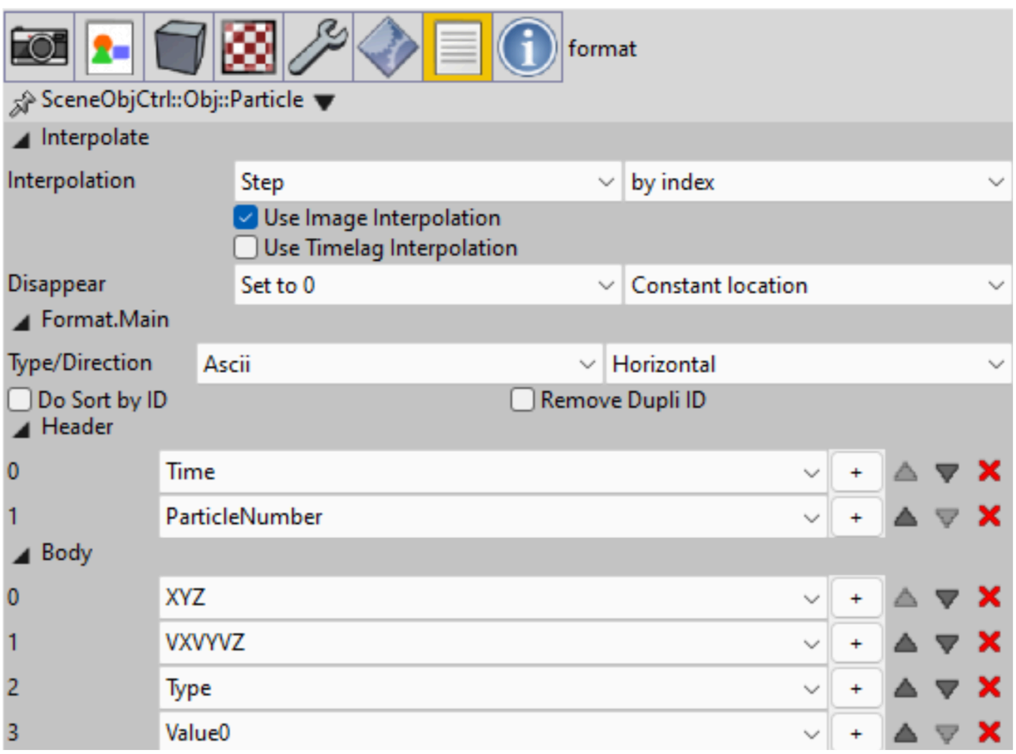
After the interpolation, this interface switches to asking the user how it should read the data set. The default setting "Binary" means it's going to interpret the data set you import later on as binary, however, our data set is not recorded in this format. So, using the drop down menu, **change it to "Ascii"** (American Standard Code for Information Interchange).

Next to Ascii, the drop down menu will ask if it should read the data horizontally or vertically. **Leave it as horizontal.**

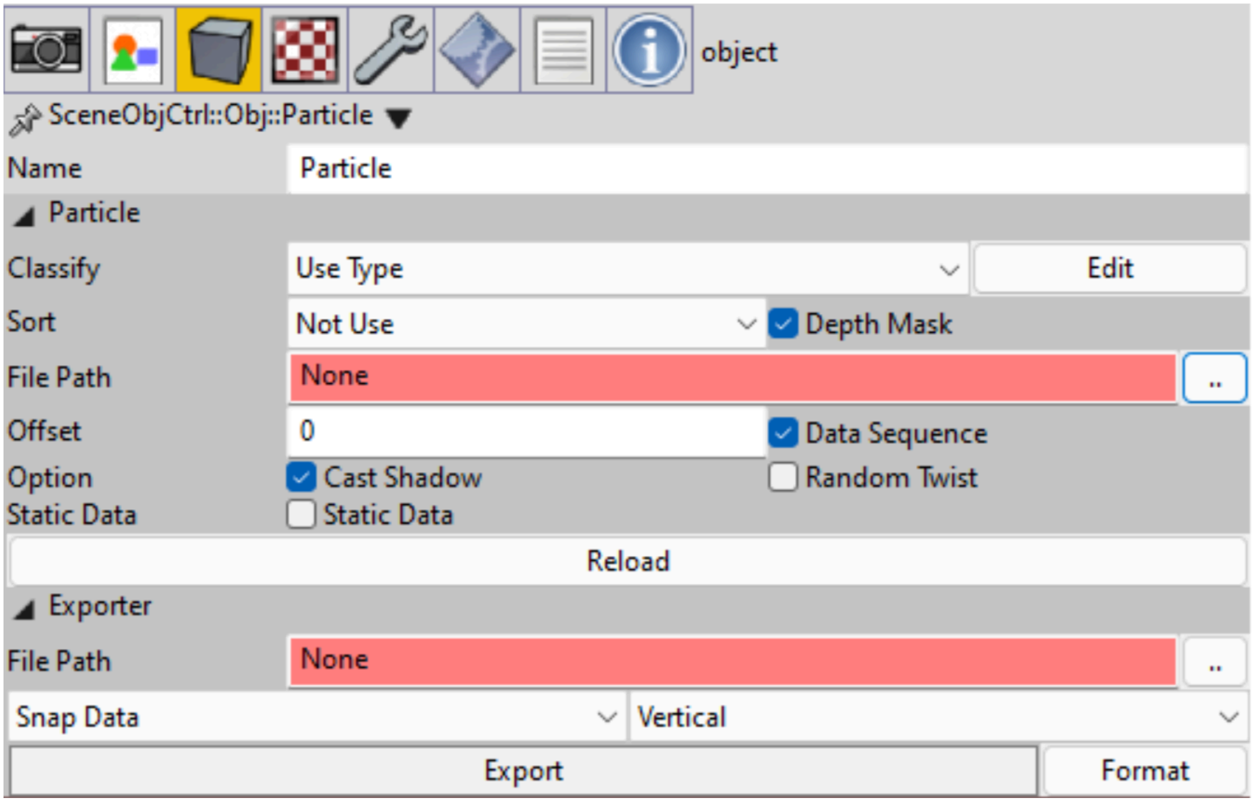
After that, it's going to ask what the two numbers printed at the top of each timestep document mean. This will depend on your Python code, however, for 2024, each timestep has *time* printed first, and then particle number on the second line. **So, switch the order of these in the drop down menus.**

Next, Zindaiji will ask how to interpret the body of code. Again, this will depend on your code output. However, for 2024, the format is x, y, z positions, x, y, z velocities, material, entropy, density, energy, and pressure, read from left to right. So, **leave the first drop down as xyz and change the second to vxvyvz**. Then, **change the third line to "type"** (type refers to the type of material), and add one row with the "+" button and **change it to "value0."** "Value0, value1, value2…" etc. refers to a general placeholder; Zindaiji will read whatever value is in that position in the document and store it as that value. In this case, value0 refers to entropy. If you added a value1 after value0, value1 refers to density.

By the end of this step, your panel should look like this:



After you do this, click on the box icon in the same bar as the text document icon. This is where you are going to import the data. Your screen should look like this:



Now, click on the two dots button next to the *first* red bar titled "None" (In the above picture, it is outlined in blue). This should open up your folder directories. Navigate to where you stored your output data from the Python program and click on the very first timestep, AKA the 0th timestep. When running the simulation, Zindaiji will read that program, then move on to the 1st timestep in that folder, then the 2nd, then the 3rd, etc. If there is a jump in the naming of timesteps (for example, timestep 158 is missing so the names jump from 156, 157 to 159), then Zindaiji will register an error when arriving at that timestep. To fix this, see Part 4 of the Python Manual, or read the doc string at the bottom of the file ReadSPH2024.py.

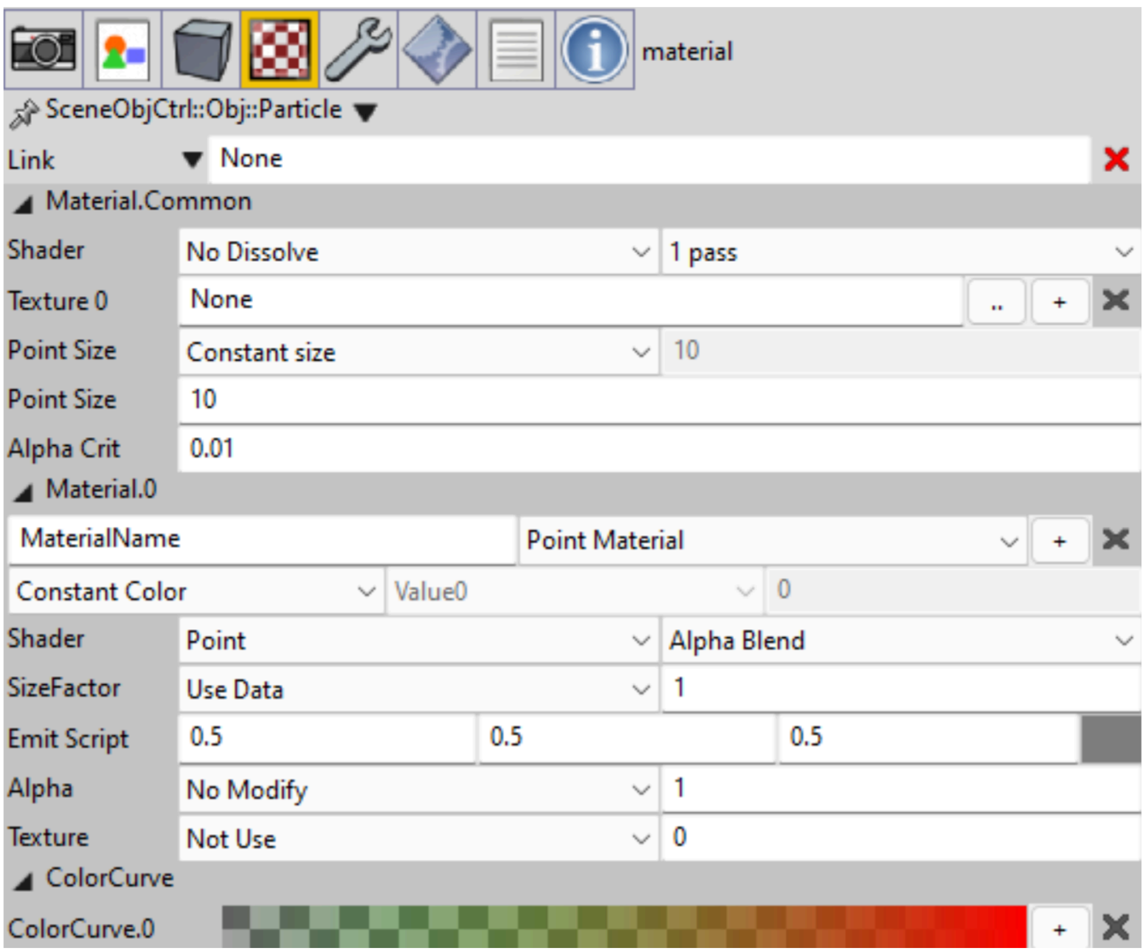
The data is now successfully imported into Zindaiji. After doing so, the View 3D panels should turn gray. If not, hit reload under the red "None" box. Then, go to the View 3D panel and zoom out with the mouse. You should see Earth and the impactor rendered in that window.

Part 2 – Color Coding and Aesthetics

Right now, all particles in the simulation are the same color which isn't very interesting. However, with Zindaiji, you can color code each particle according to different parameters. This manual will cover the basics and show 2 different color coding methods.

Basics

First, click on the picnic basket icon next to the box icon you used in the previous part. Your screen should look like this:



Ignore the options under "Material.Common;" we won't be using it to color code.

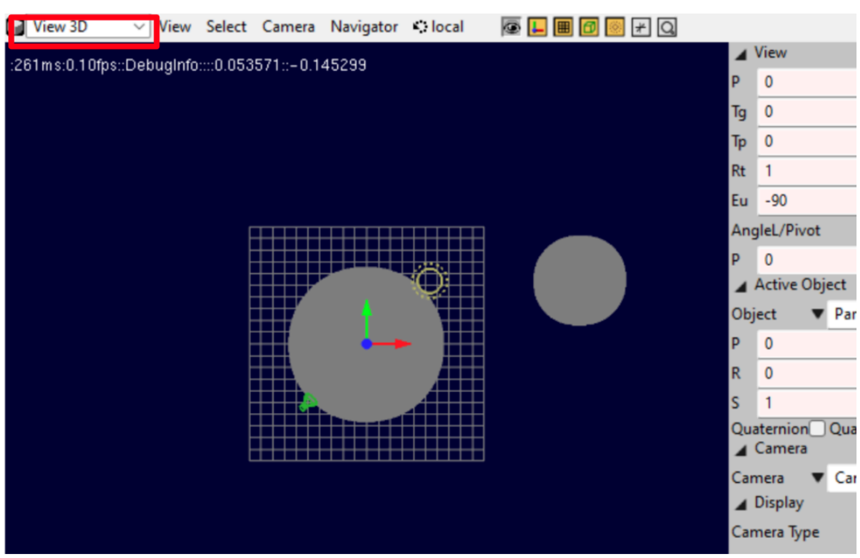
Instead, look at Material.0. The MaterialName is a user controlled parameter and does not have any effect on the simulation. Next to that, the drop down menu controls the shape of each particle in the simulation. Leave it as “Point Material” as the other options don’t work well with this data set.

Constant Color and Color Curves

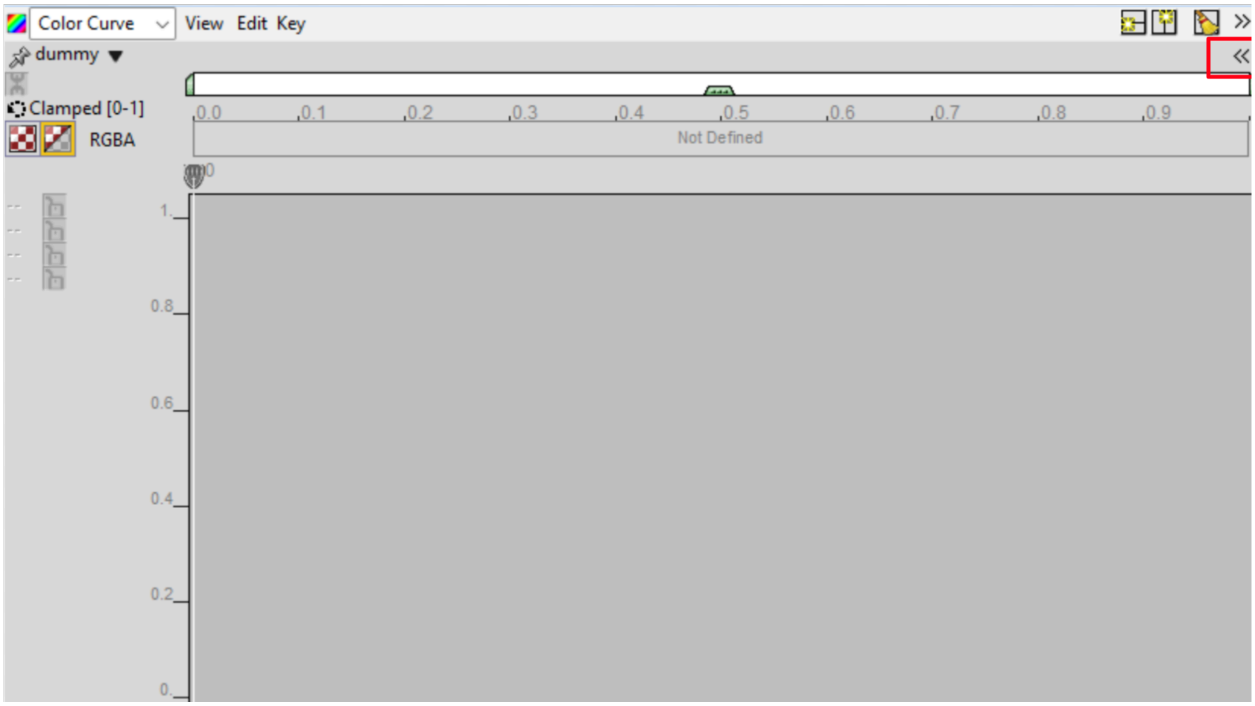
Below that, there is an option to set the objects in the simulation to a constant color. You can change this constant color by clicking on the gray box at the end of the row starting with “Emit Script.”

However, there are other options. Select the drop down menu on “Constant Color” and choose “Color Curve.” Color curve is a setting that allows Zindaiji to read the data files and look for a specified data value. It then goes through all the data files, finds the minimum and maximum of that value, and remaps each particle’s value to a float between 0 and 1. Using this range from 0 to 1, it assigns the particle a color specified by the color curve you choose. You can specify the value Zindaiji will read with the drop down menu next to color curve. Choosing Value0, for example, with color code each particle by the entropy it has at each timestep. Lastly, you can choose which color curve you are using by typing in its number next to that drop down menu.

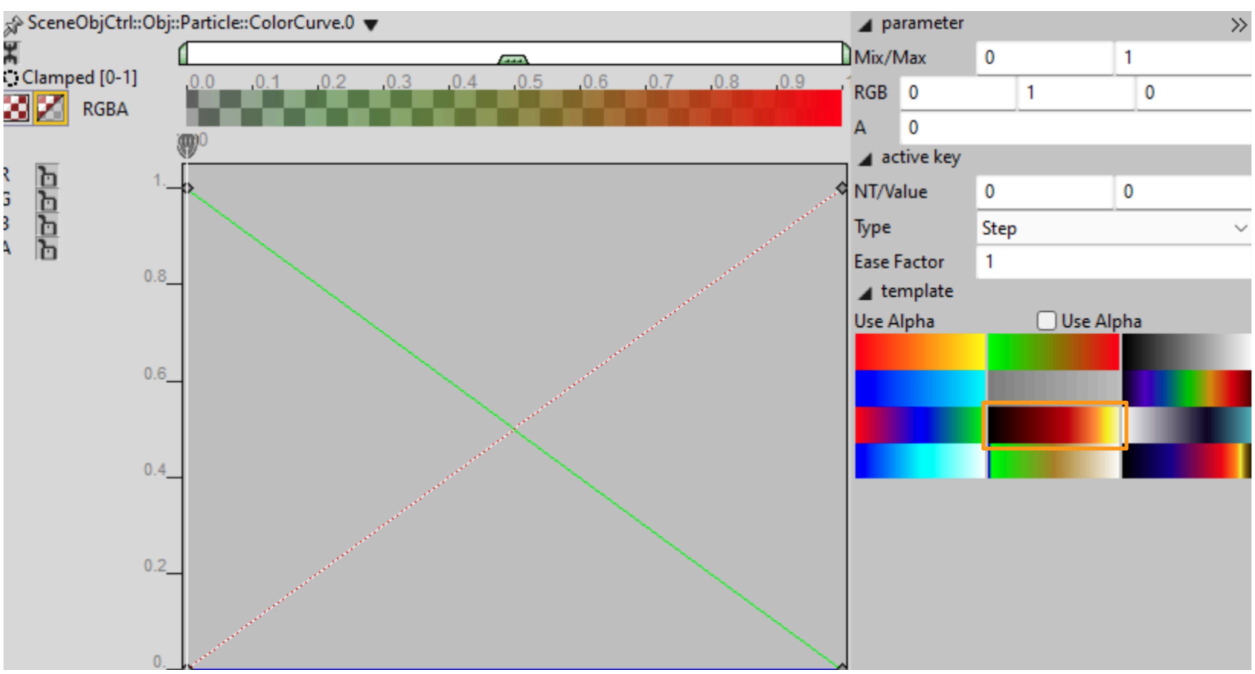
Zindaiji creates a default color curve at the bottom of this menu, however, it is not useful for this dataset. To change it, go to the View 3D panel and switch it to the Color Curve menu with the drop down box.



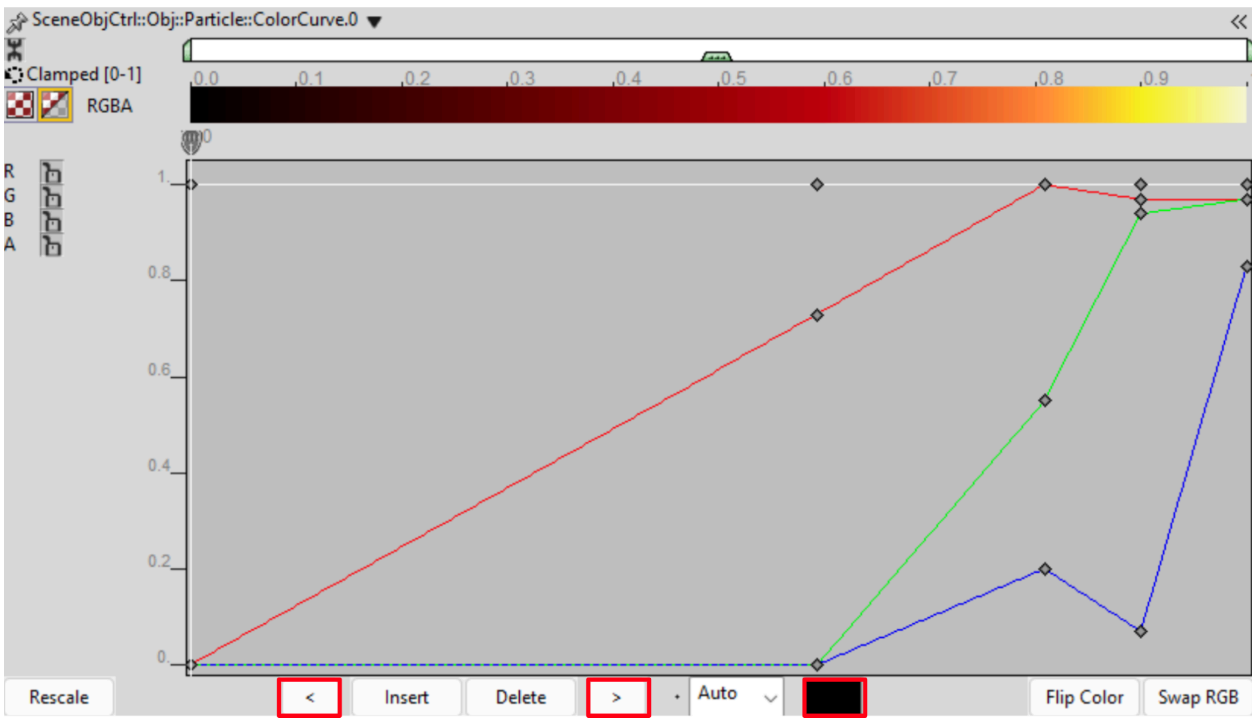
Your View 3D panel should turn into this:



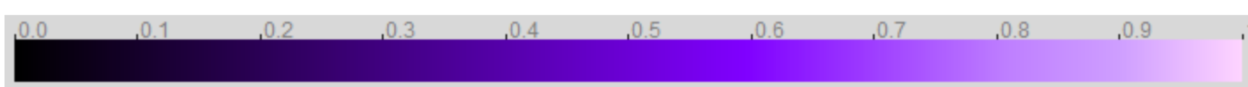
To start, go to the menu with the color settings and click on the default ColorCurve0 at the bottom. This tells Zindaiji that you want to change it. From there, click the arrow outlined in red in the above image. This should pull up color curve templates:



You can simply choose one of these templates if you want, or if your needs are more specific, you can modify them. This process can get very complex, so this manual will show one good way to do it. Using the templates given, choose one of them that is generally what you want, but maybe has a few differences. For example, I want a curve that fades from black to several different shades of purple. So, I'll choose the one highlighted in orange in the above image. The screen should look like this:

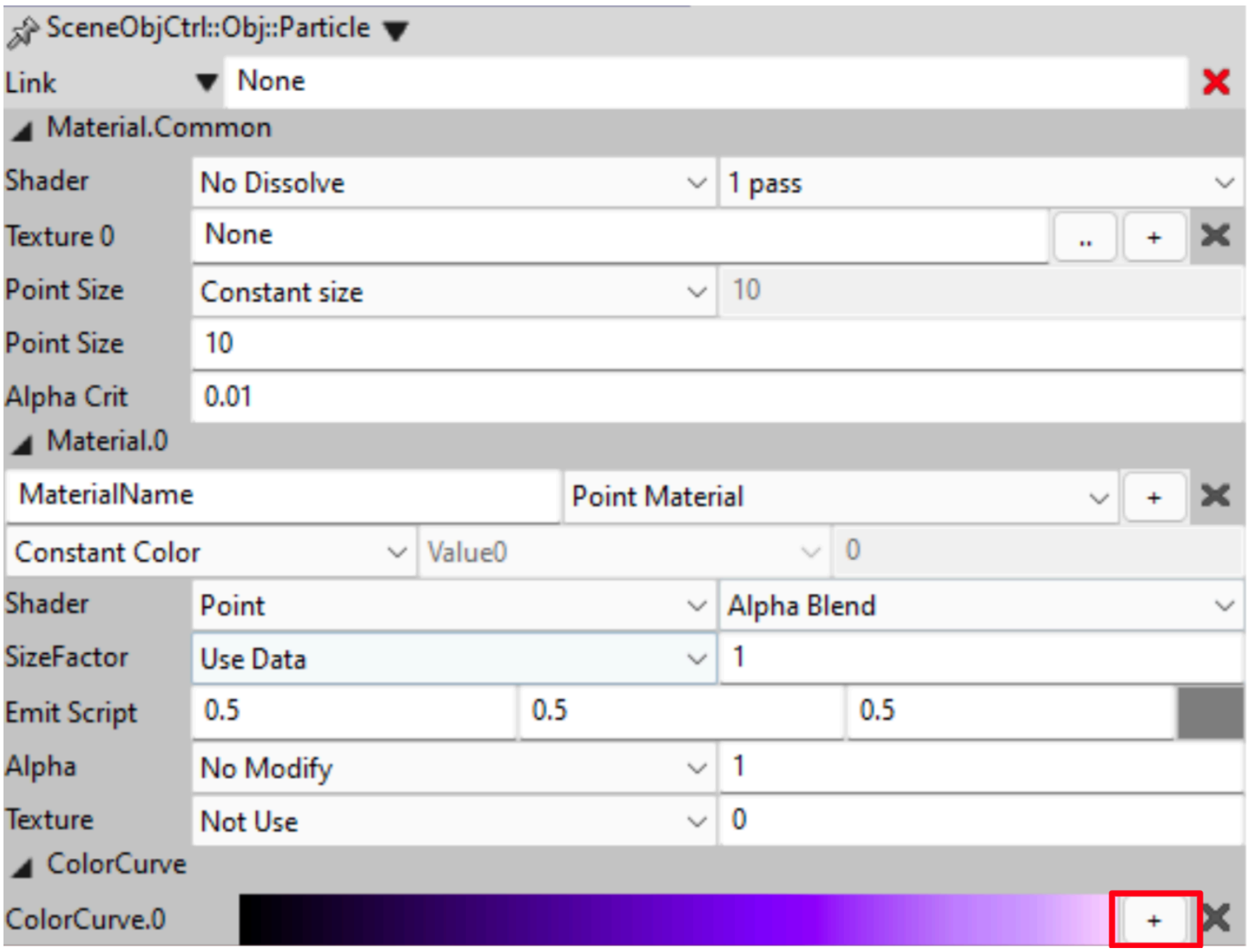


Now, clicking on the arrows highlighted in red in the above image will switch between pivots in the color curve where the color is explicitly defined. Between these pivots, Zindaiji creates a gradient to blend them together. Now, clicking on the black box highlighted in red will pull up a color menu. Here, you can change the color of the pivots, meaning you can create a gradient between any two colors. Here's an example of a violet color curve you can create with this method:



This process will work similarly for the other templates in the menu.

If you want to create additional color curves, you can do so by clicking the plus button (highlighted in red below) and repeating the above steps. Note that even if you assign a color curve to a material, differences in the color may not show up in the first few timesteps since no collision has occurred - you will have to skip around in the timeline to see the effects of the color curve.



Particle Size and Shaders

Besides color, you can also modify the particles' appearance by their size. From the row titled "SizeFactor," change the dropdown menu to "Constant(1)." Initially, Zindaiji was trying to use the inputted data to determine the size however, we never specified what the size should be. Instead, you can change the size by altering the constant next to the drop down menu. I find that sizes between 0.1-0.3 work the best.

You can also change the shape of the particle in the row titled "Shader." The default is "Point" which is a square. However, you can also change the particle to look like a circle, a sphere, or be invisible with the "No Rendering" option. Play around with it to see the different lighting effects.

Color Coding Method 1: Entropy

The first method we'll cover is color coding the simulation solely by the amount of entropy it has. To do this, select "Value0" as the parameter the color curve is measuring and choose the color curve option. From there, choose an appropriate color curve to show entropy (a common one is the one from the templates shown below).

This is a very simple color coding method, however, has very revealing results when the simulation is completed:

https://drive.google.com/file/d/1GBE_U4CwnUUU9_Hw3pZvZTqN4mxbYES1/view?usp=sharing

Color Coding Method 2: Material

The second method involves assigning a different color to each material. To do this, first make sure that in the text document icon menu, Zindaiji will read the material data as "Type." From there it will read through all the files, determine that there are 4 different "types" of materials indexed by 0, 1, 2, and 3. Then, add 3 additional materials in the menu under the picnic basket icon for a total of 4 materials. Zindaiji will interpret this as you wanting to show the different types of material in the simulation. From there, you can select constant color and change the way Earth's core, the impactor core, Earth's mantle, and the impactor mantle look.

Alternatively, you can also assign a separate color curve to each material. To do this, type in the number of the color curve next to the drop down menu where you specify the value. The steps will be the same as in the first method of color coding, however, you will repeat it for each material so that each one is curving “Value0,” for example, with a different color curve (if you use the same color curve for each, then this method is no different than the first). If you use Value0, then Zindaiji will curve the color of each material distinctively so that you can visualize both entropy and material at the same time. Note, you can of course do this with other parameters in the Python files, such as pressure, energy, or density.

Part 3 - Pseudo-Camera Controls

One of the final things we'll want to control for our simulation is the camera controls, or what angles and zoom we'll film the animation from.

The way Zindaiji recommends doing this is with the buttons that directly control the camera. However, I found that this is very hard to operate in a controlled manner; the camera seems to have a mind of its own.

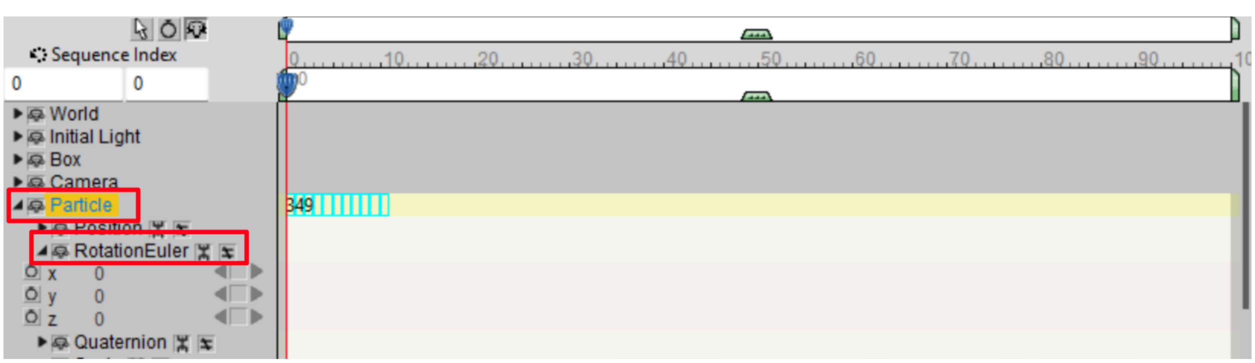
Instead, the best method I have found to control the camera isn't to control it at all, but instead to control the particle. For example, let's say we want to make the camera rotate around the particle. Using the camera settings makes this difficult, so instead we can just set the *particle* to rotate. This gives the illusion that the camera is actually the one rotating. To understand this better, view the video below (it's the same one from the entropy example). In this simulation, I have set the particle to rotate 360 degrees, however, this motion gives the impression that it's actually the camera moving.

https://drive.google.com/file/d/1GBE_U4CwnUUU9_Hw3pZvZTqN4mxbYES1/view?usp=sharing

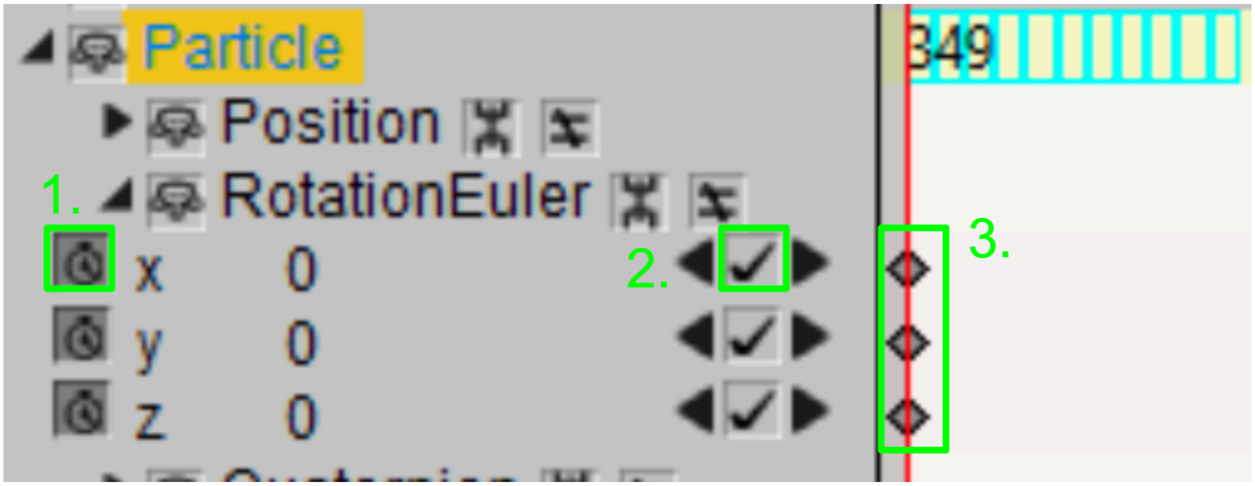
You can do a similar trick for zooming in and out - just change the scale of the object. A higher scale is the same as zooming in, and a lower scale is the same as zooming out.

Rotation and Scale Pivots

First, we'll show how to create a rotation.



First, navigate to the timeline, and select “Particle” on the left hand menu. Then, from the drop-down, select “RotationEuler.”

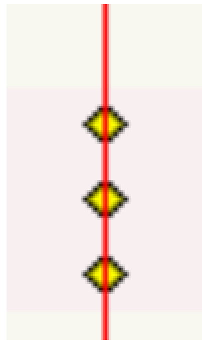


Click the clock icon next to x,y,z (you can click any of the three). It should turn dark and create two new items on the timeline, labeled 2 and 3 in the picture above. The three dots in box 3 represent a rotation pivot which is where you can enter an angle you want the particle to be at (we’ll get to that later). The three check marks in box 2 is the on and off switch - click one of them and it will turn on/off the pivot. If you scroll on the timeline to a later timestep and click the check marks again (this time they should be empty since you’re not hovering on a pivot), and click one of the three empty boxes where the check marks should be, it will create another pivot. Now if you enter a new angle for the particle, the particle will rotate to that new angle as time progresses between the pivots.

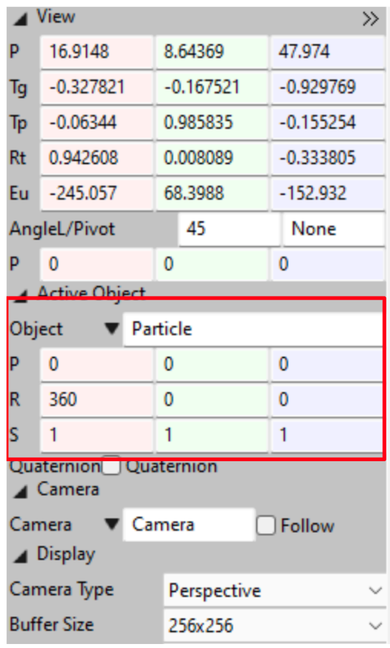
The process for creating pivots to control the zoom in/out is the exact same. Under “particle” on the menu at the left, go to scale. Use the same process to create pivots for the zoom.

Entering Values for Rotation and Scale

To enter the values for what the rotation and scale should be at certain pivots, you’ll first want to hover over the pivot on the timeline and click one of the three diamonds so it’s yellow:



Next, go to the panel to the right of the View 3D window and find the boxes highlighted in red below:



You'll first want to make sure the drop down menu for the object is set to "Particle." For rotation, you'll want to enter values to the "R" row, and for scale, you'll want to enter values to the "S" row.

For scale, you'll want to make sure all three numbers you enter are equal - otherwise it will stretch/deform the particle. For rotation, you can enter whatever values you want. Each box corresponds to the angle rotated around its colored axis. You can see these axes in the bottom left corner of the View 3D panel.

Besides scale and rotation, you can also control the position of the particle on your screen, however, I find this is less useful.

Part 5 - Cross Section Simulations

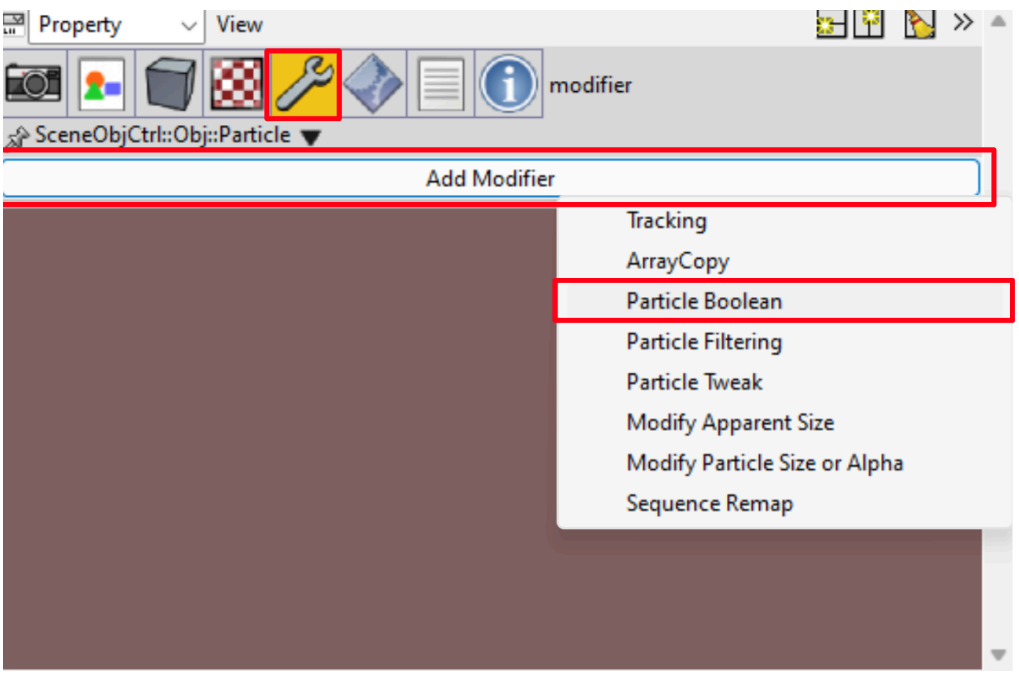
Besides 3-dimensional simulations, simulations of the cross section of the impact are a great way to show the mixing that occurs during the collision:

https://drive.google.com/file/d/1PU0JbYQoHE3j-gXbkaql-wugo1htdbeU/view?usp=sharing

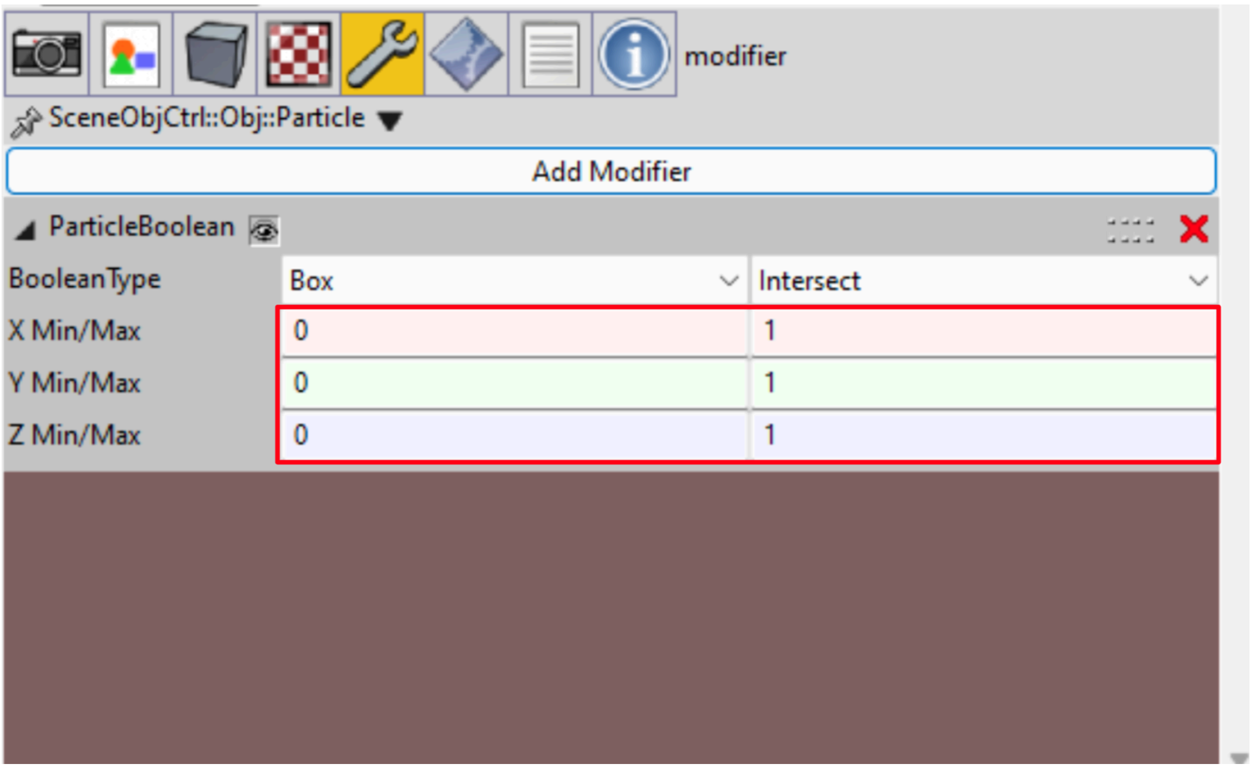
To create this in Zindaiji, we'll use a particle Boolean which will make it so certain particles are invisible.

First, you'll want to set each material in the simulation to a different color so you can see the mixing clearly.

Now, go to the bottom left panel and click the wrench icon, then "Add Modifier," then "Particle Boolean."



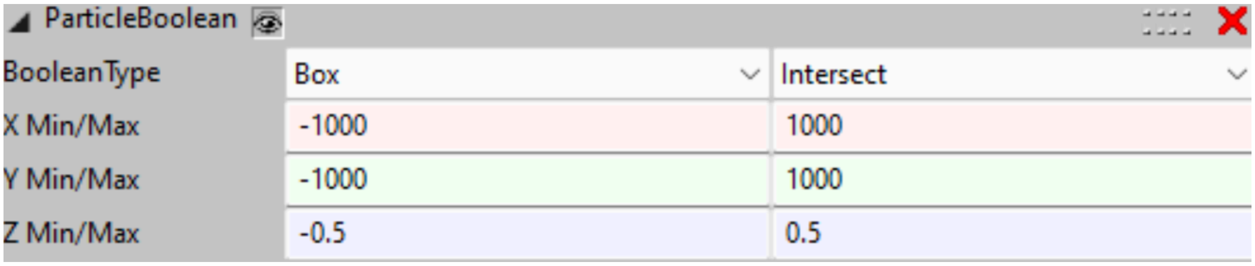
Your screen should look like this:



Under the “Intersect” drop-down menu, you will find 3 options: “Intersect,” “Difference,” and “Add.” The way that this modifier works is it creates a 3D shape in the View 3D panel. With the “Intersect” option, only the particles that are inside the shape, or intersect with it are shown. With “Difference,” it is only the particles that are outside the shape. “Add” will show particles hidden by a separate particle boolean modifier. To display the cross section, we will want to use “Intersect.”

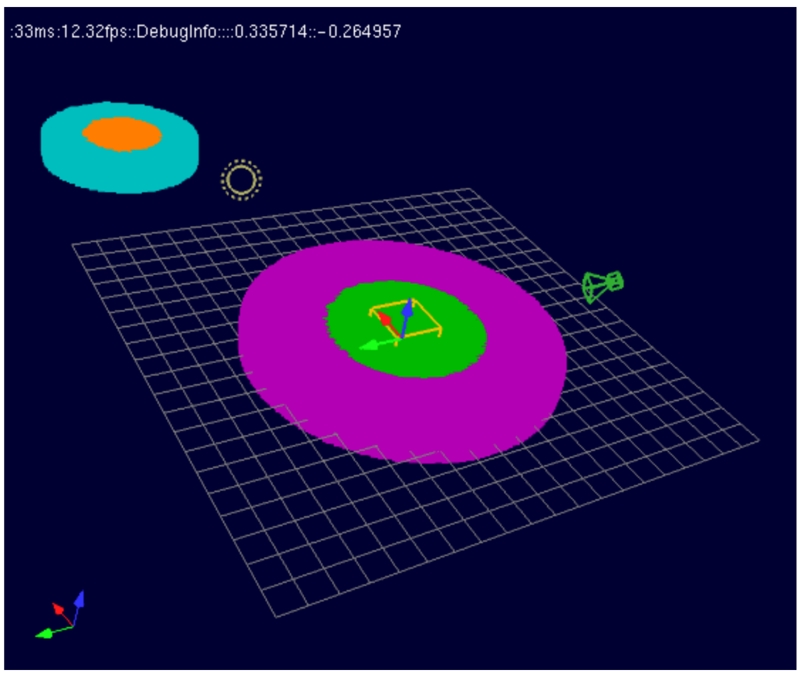
You can control the shape that the particle boolean filter is under the “Box” drop down menu. The default options are “Box,” “Sphere,” and “Object” (for object, I believe you would have to create another object involving importing more data). “Box” is the easiest option to use for cross sections.

In the box outlined in red in the image above, you can control the dimensions of the box by inputting the minimum and maximum x,y, and z values its respective dimensions should have. You may have to play around a little, however, I found that the dimensions that worked for me were these:



For the x and y dimensions, you just need to input arbitrarily large values since we want every particle in that plane to be shown. The z dimension is where we control the thickness of the cross section.

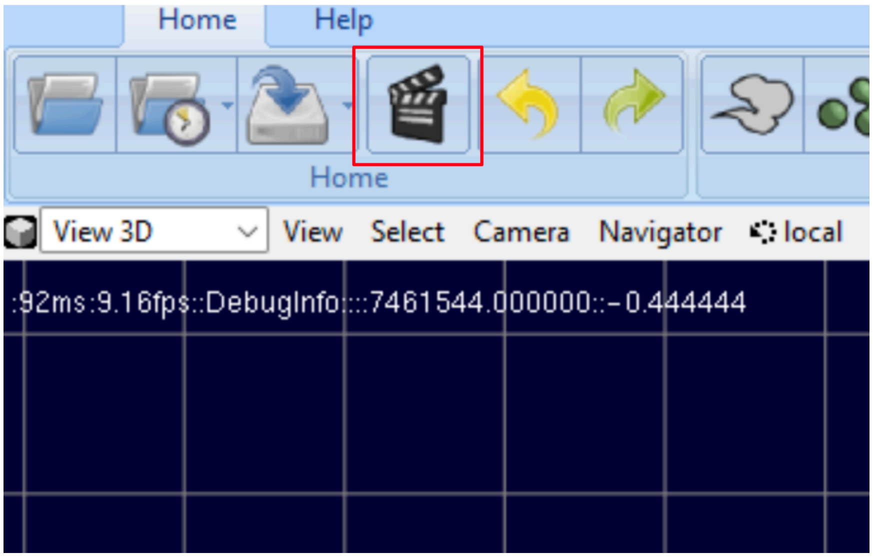
Now, in the view 3D panel, you should see something like this:



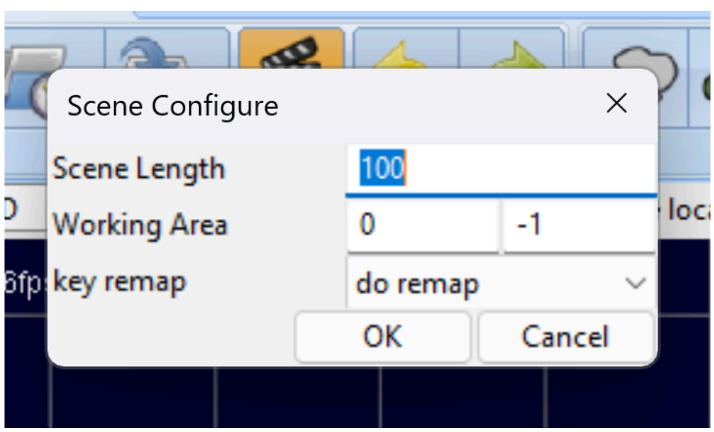
You can now run the simulation just as before.

Part 4 – Adding More Timesteps

This is a short segment. To add more timesteps to visualization, click on the scene icon at the top toolbar.



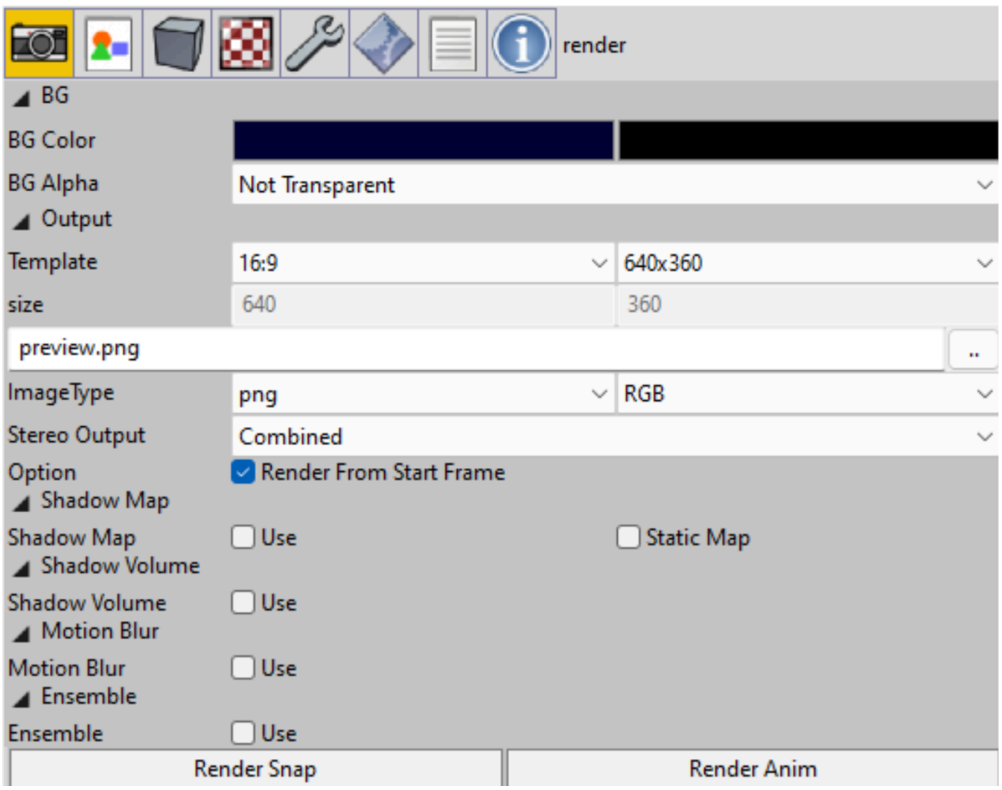
You should be greeted with the following pop-up:



The default timestep amount is 100; you can change this by editing “Scene Length.” If you enter more timesteps than you have data for, Zindaiji will just fill up the missing timesteps with the frame from the last timestep. Don’t change “Working Area” or “key remap.”

Part 6 – Exporting the Animation

Now, it's finally time to export the animation. To do this, first click on the camera icon on the toolbar on the right panel. Your screen should look like this:



In the first colored box in the row titled "BG Color," you can change the background color of the simulation. Underneath that in the row titled "Template," you can control the aspect ratio and resolution of the simulation.

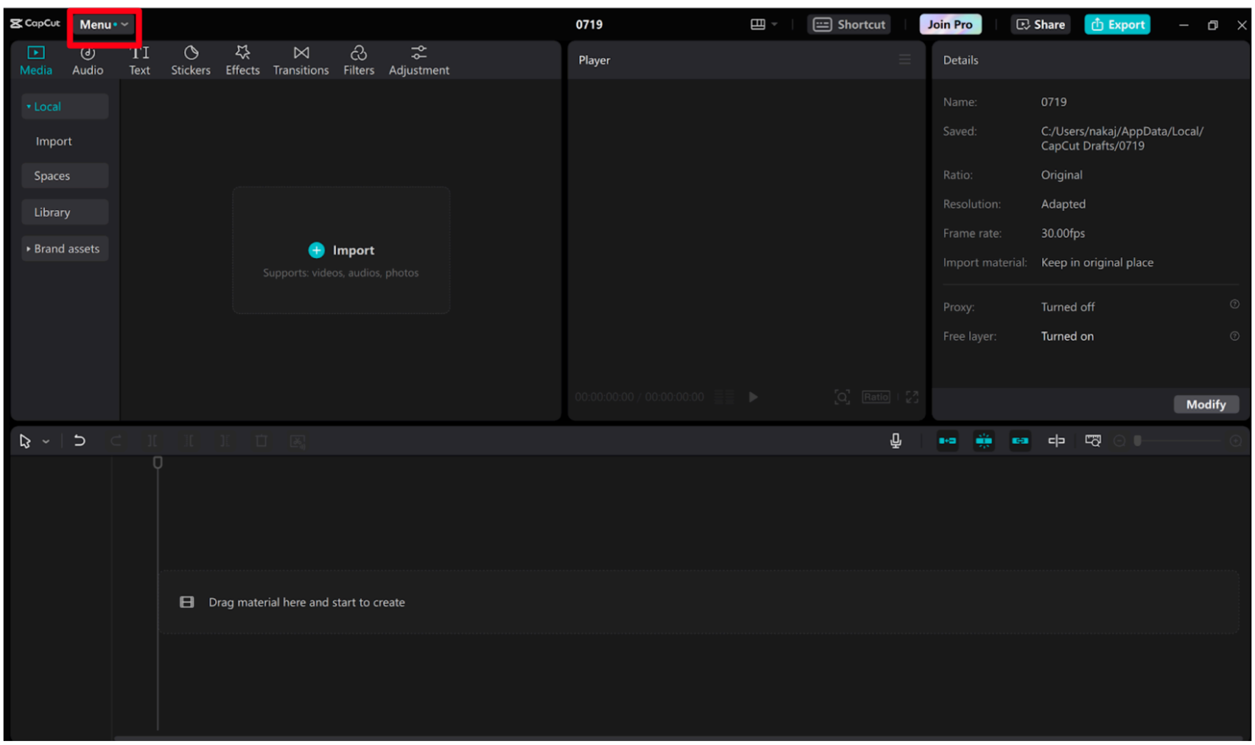
In the bar underneath that, with the text, "preview.png," you control which folder the simulation will be outputted to by clicking on the two dots. Once you have your desirable settings, make sure the "Render From Start Frame" box is checked, and click "Render Anim." This will take screenshots at each timestep in the simulation, and output them to the folder you selected.

The next step is to compile these photos into a video, much like you would a stop-motion animation. There are several softwares that can do this for you, however, I find the easiest to be CapCut.

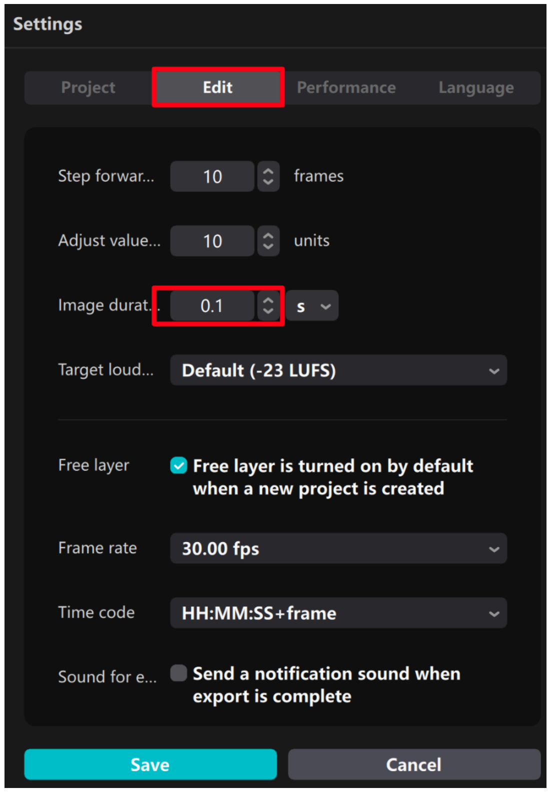
If the computer does not already have CapCut, you can find the download here:

https://www.capcut.com/tools/desktop-video-editor?utm_medium=sem&utm_source=googleadwords_int&pid=359289&af_c_id=21157337217&adset_id=162157605753&ad_id=697948663363&placement=&keyword_name=capcut&targetid=kwd-1458048424717&matchtype=e&gad_source=1&gclid=CjwKCAjwnei0BhB-EiwAA2xuBk96Vkn79hW1SJzN0LUo4ZhNDm6siqNrhNHHpJLmqVn6IQASLbJ0exoCWEMQAvD_BwE

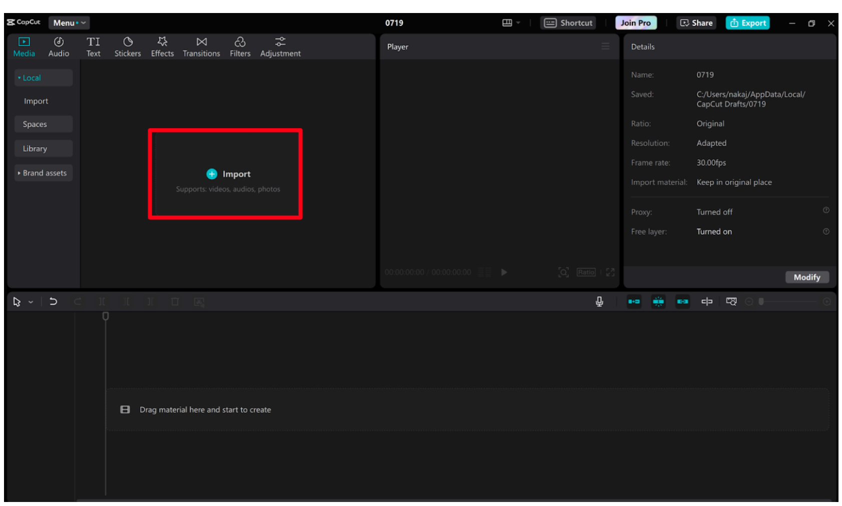
After downloading, open CapCut and create a new project, so that your screen looks like this:



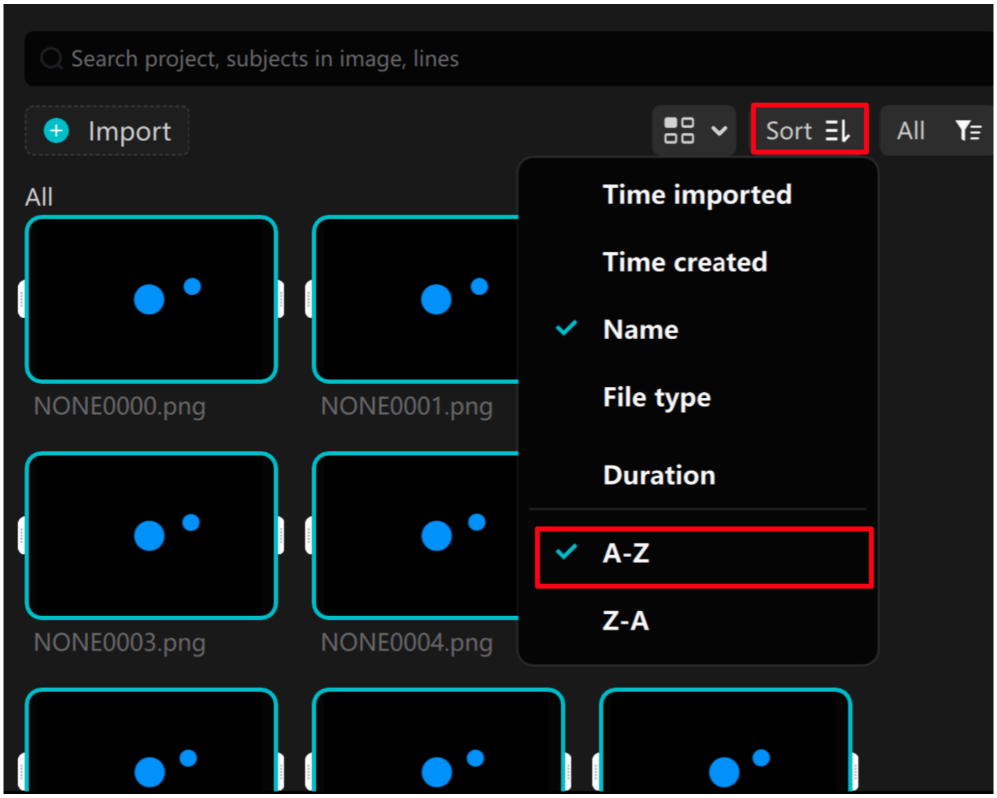
Click the menu button highlighted in red, and then click on settings. Then, in the settings pop-up, click on “Edit.” Your screen should look like this:



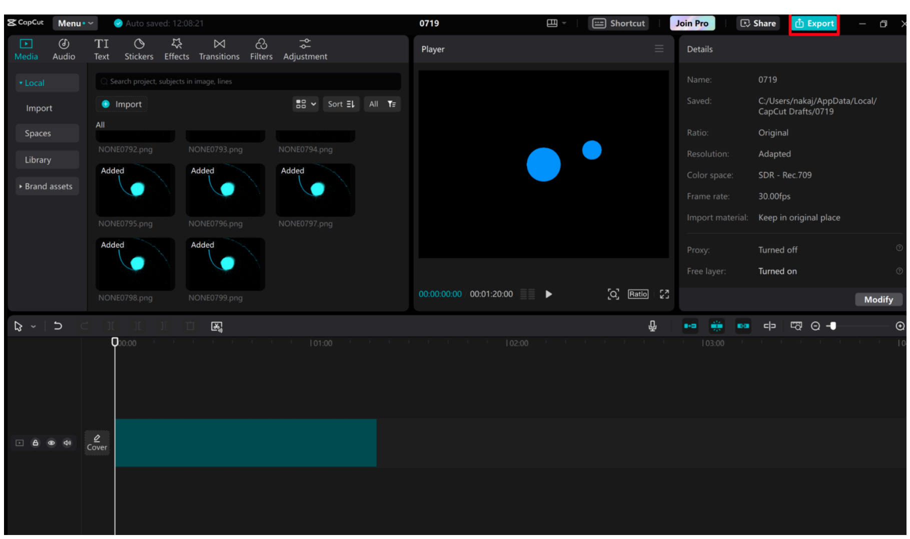
Now, make sure your image duration is set to 0.1 seconds, and then hit "save." Next, click "import" back on the front page, and select all of the images Zindaiji created when rendering the animation.



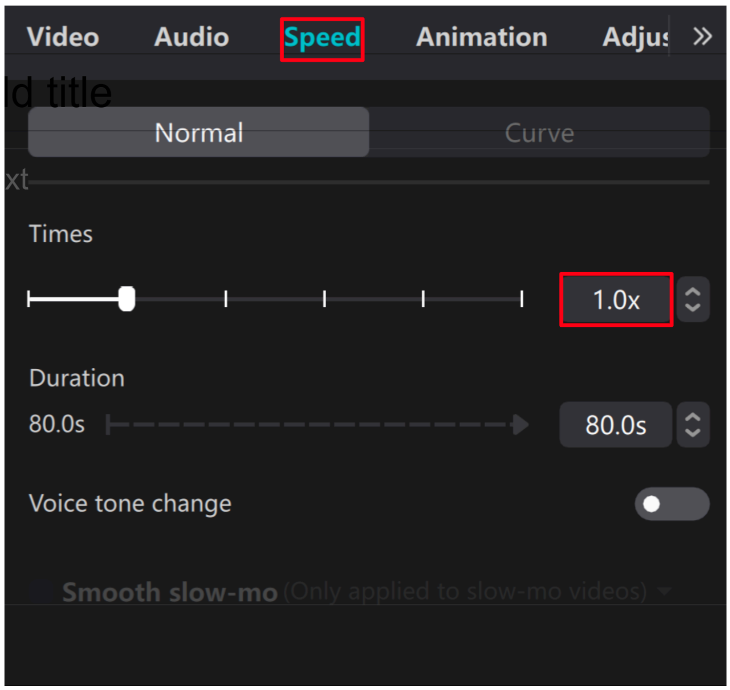
Next, go to the "sort" button, and click "name." Then, click "sort" again and click "A-Z" at the bottom:



This will put all of the timestep images in order. If all of the images are not already selected, select them all by clicking on the first image, then scrolling down to the last image and clicking it while holding down shift. Then, click and drag one of the images onto the timeline; CapCut will automatically bring the rest with it since you have them all selected:



Now, click “Export” at the top left and choose a location to store the video. Once it has finished exporting, you’re now going to *import* the video that was just exported back into the project. Delete all the images on the timeline and drag the imported video onto it instead (you can delete all the images easily by clicking on a blank spot on the timeline and dragging over the images). Now, make sure you have the video selected on the timeline and go over to the panel on the right hand side. Click “Speed.” Your screen should look like this:



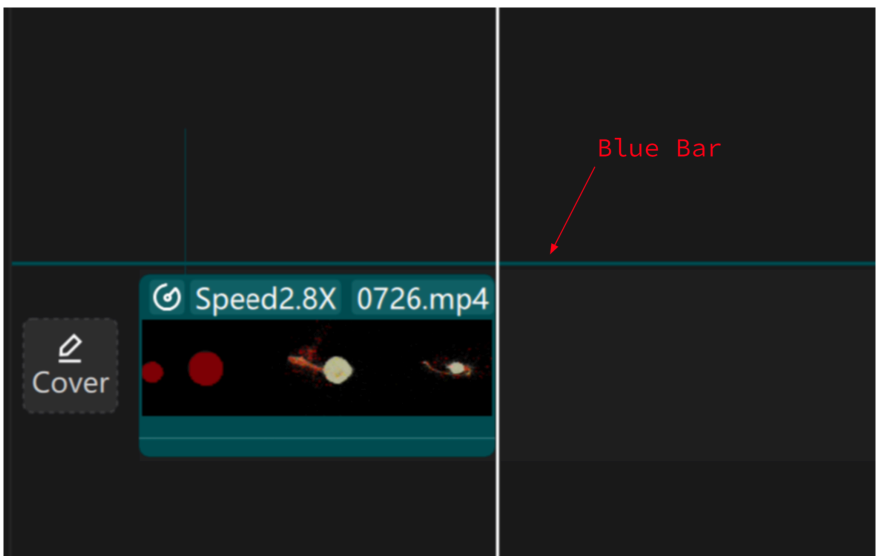
The red box under “Times” can be used to change how fast the simulation plays. A faster speed results in a smoother animation. After you have your desired speed, export the simulation.

You are now finished, congratulations!

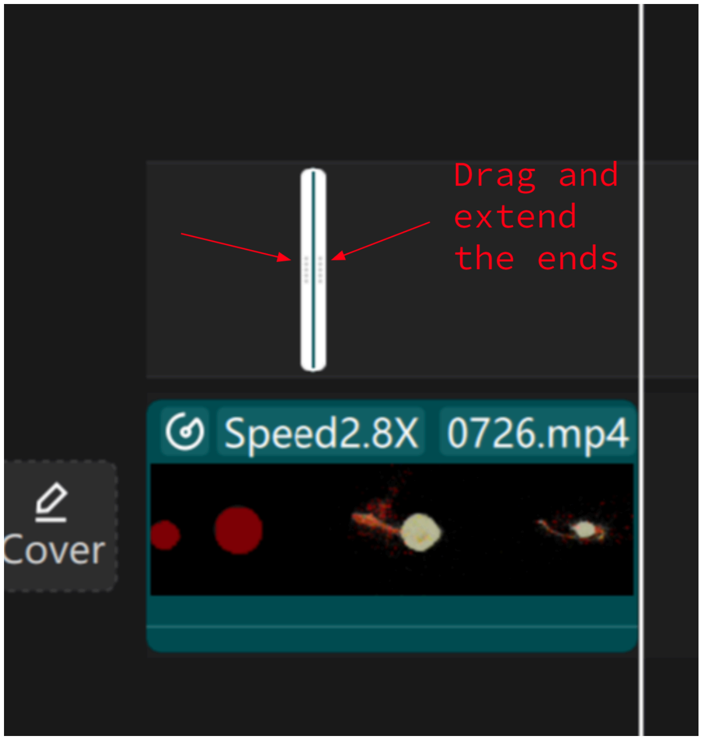
Part 7 - Adding Extra Elements

For some simulations, you may want to add in some extra elements afterwards, such as a scale for the parameter being visualized. For example, if I was visualizing entropy, I might want to overlay an image of the color curve with values on either end to indicate the scale.

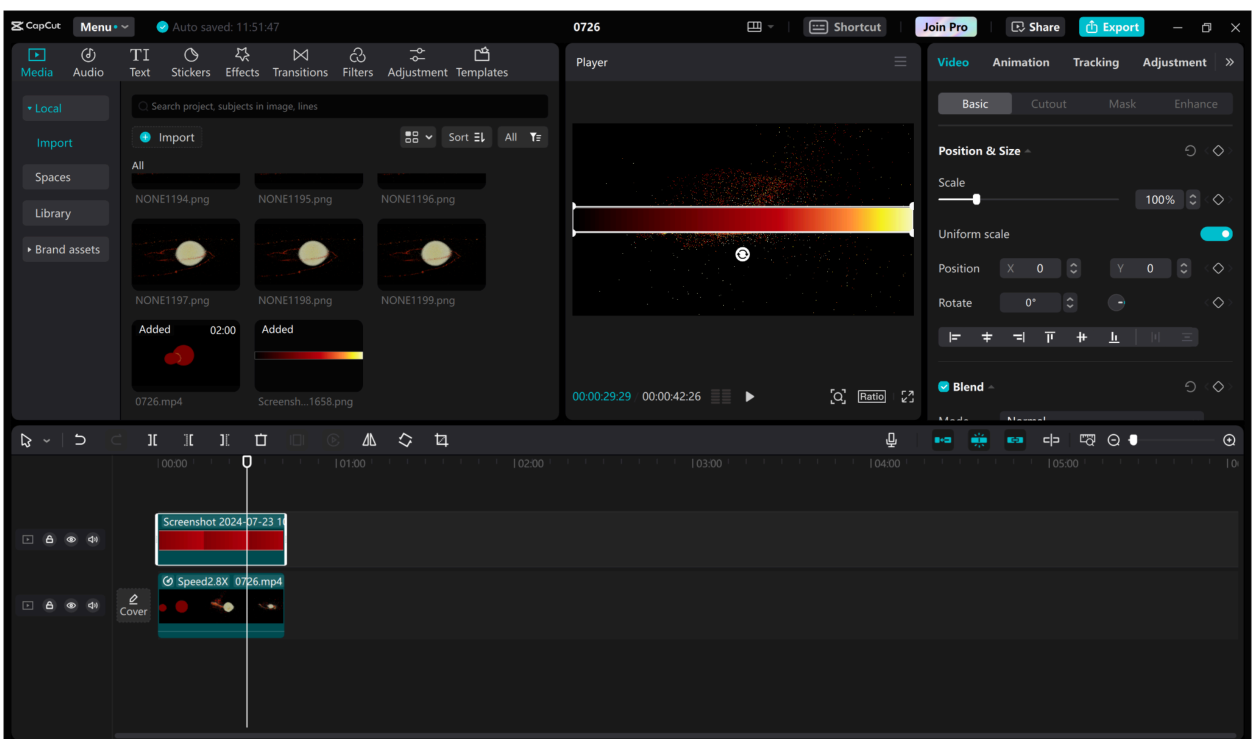
To do this, you can use CapCut once again. First, in the media section of your project, import a screenshot of the color curve. Then, drag it onto the timeline *above* where the video is. A blue bar should appear above the video indicating that the screenshot will be placed on a timeline above it, and will be overlaid:



Now, when you let go of the mouse the screenshot will appear somewhere above the video on the timeline. Click and drag the ends of the screenshot block so that it is aligned with the video:

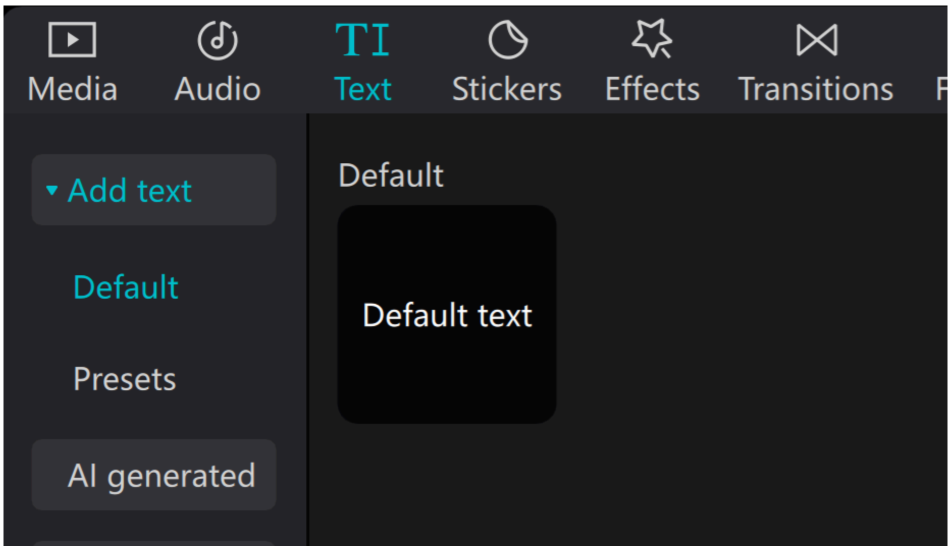


Your screen should now look something like this:

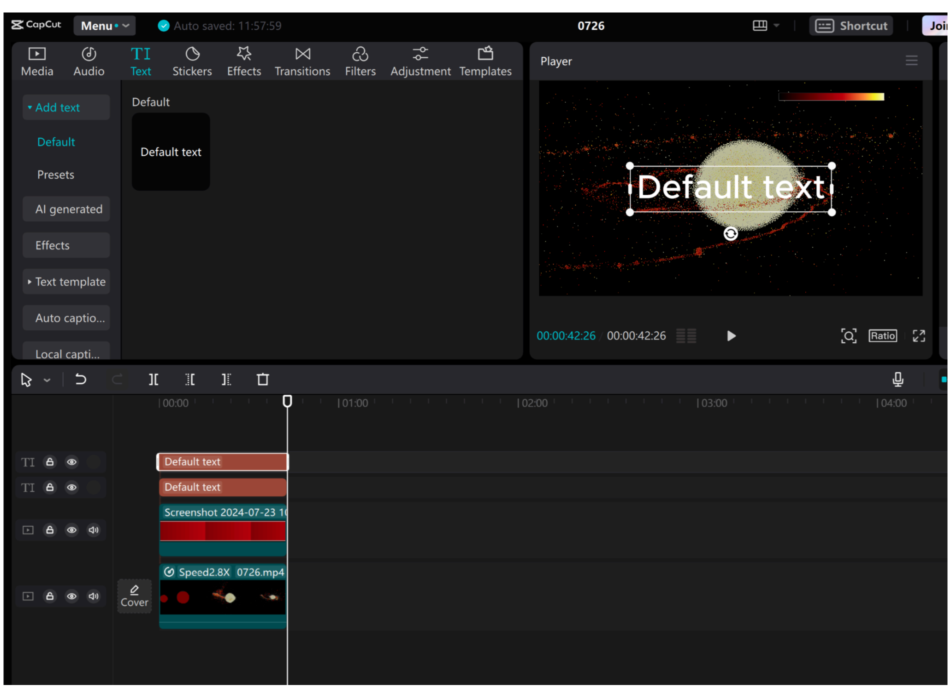


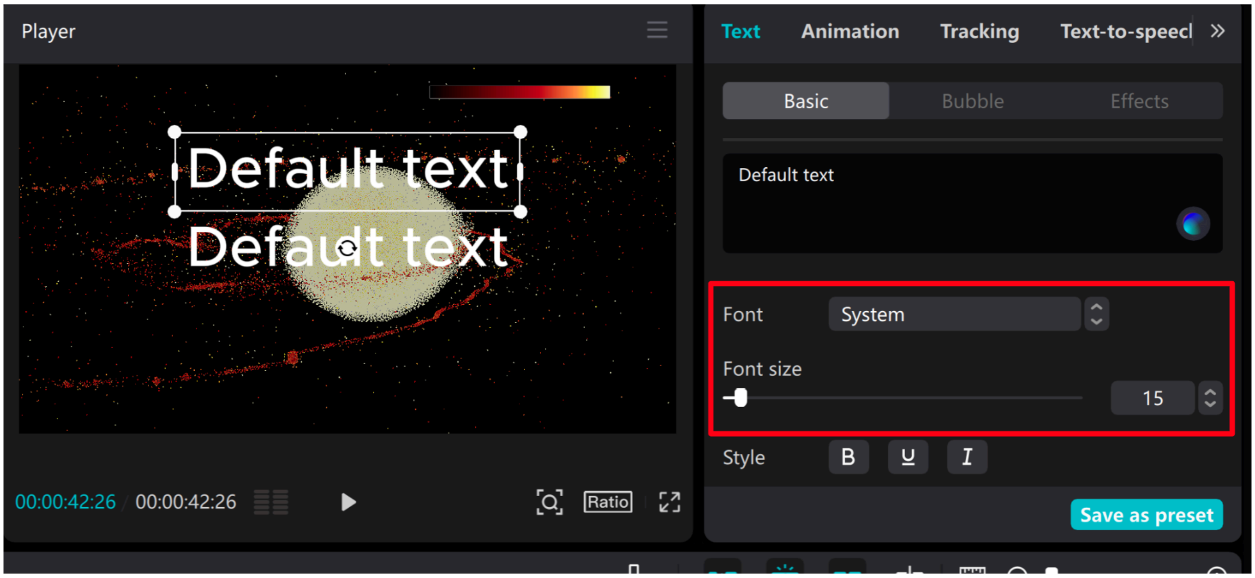
In the panel with the video, you can resize the color curve and drag it to your desired position. Additionally, if you find you cannot shrink the screenshot anymore, you can zoom in on the panel which should allow you to shrink it more.

The next step is to add the numbers on the scale. Go over to the import media panel and click “Text”:



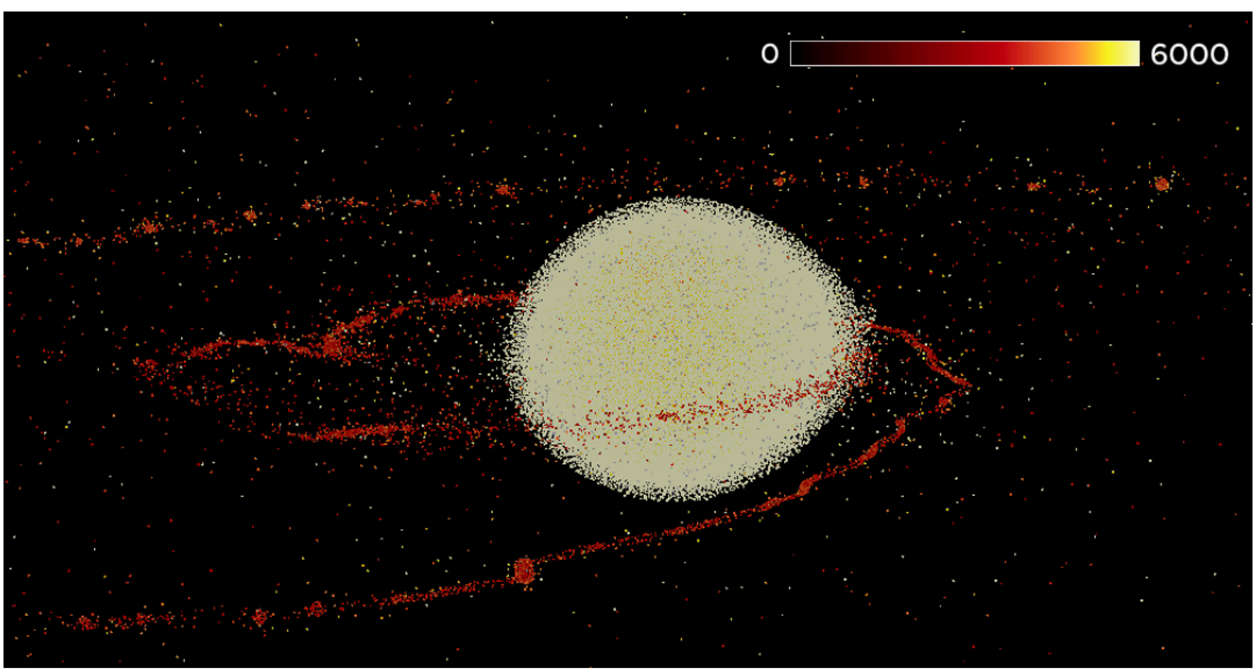
Click and drag “Default text” onto the timeline twice and put them stacked on top of each other, above the screenshot just as you did before. Then, extend their ends to fill the whole video:





If the textbox is selected on the timeline, you should be able to edit the font settings in the box highlighted in red above. You can also edit the text by clicking on the textbox directly, or by clicking in the “Default text” box above the red highlighted box.

Enter in the appropriate numbers and drag the text boxes to the desired location. An example could look something like this:



Note: it can be tedious to move around the text boxes and get them aligned. An alternative is to use one text box instead of two, type in the first number, add a bunch of spaces, and type in the second number so that it comes out at the other end of the scale. This way, you don't have to fiddle with the text boxes until they are aligned.