Water Flow Tutorial

From VisItusers.org

This page contains an in-depth visualization tutorial for a water flow simulation.

Contents

Description of Simulation Data

This tutorial uses the **dbreak3d** dataset -- available at: http://www.visitusers.org/index.php?title=Tutorial_Data

The dataset simulates the evolution of water and air in a water tank after an interface holding a column of water is instantaneously removed. The data was generated using the OpenFOAM open source CDF simulator, using a two-phase volume of fluid (VOF) method to resolve the interface between the water and air.

Our case is a 3D variant of the 2D Dam Break OpenFOAM tutorial, which is described in detail at: http://www.openfoam.org/docs/user/damBreak.php. Several changes to the 2D tutorial input deck were necessary to create the 3D simulation. We started with guidance from http://www.calumdouglas.ch/openfoam-example-3d-dambreak/. Our OpenFOAM input deck is available at: https://github.com/cyrush/dbreak3d_vtutorial. We ran the simulation on a single node of the insight cluster at Clemson (Special thanks to Vetria Byrd and Galen Collier of Clemson for access to this resource)

After running the simulation we converted the OpenFOAM output to Silo data files. The Silo files are compressed and contain the subset of the variables of interest for this tutorial. The final dataset has 160 silo files holding 4 seconds of the simulation at time intervals of 0.025 seconds [40 timesteps per second].

You can open the entire time series in VisIt using the **dbreak3d_fluid.visit** file.

Variables

The simulation is run in an Eulerian fashion: The mesh itself is static -- but the variables evolve to reflect the physics.

Each of the 160 time steps contains three variables:

- ***U*** {u,v,w} -- Velocity vector
- ***alpha1*** -- Volume fraction of water (0.0 - 1.0)
- ***p_rgh*** -- A pressure term

Boundaries

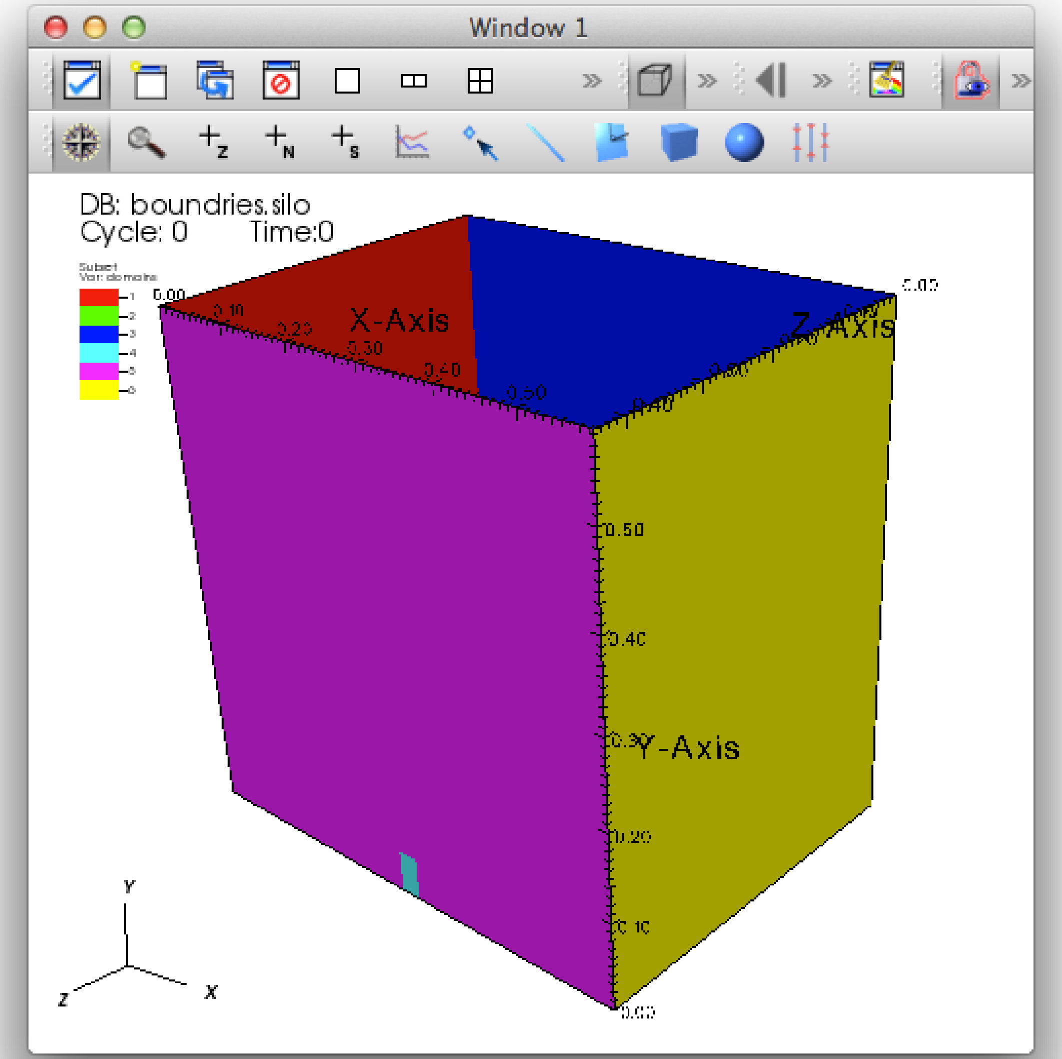
In addition to the fluid data the **dbreak3d_boundaries.silo** data file contains the boundaries of the water tank. We will use this additional file to bring context to our visualizations.

Exploring the Fluid Data

Viewing the Tank Boundaries

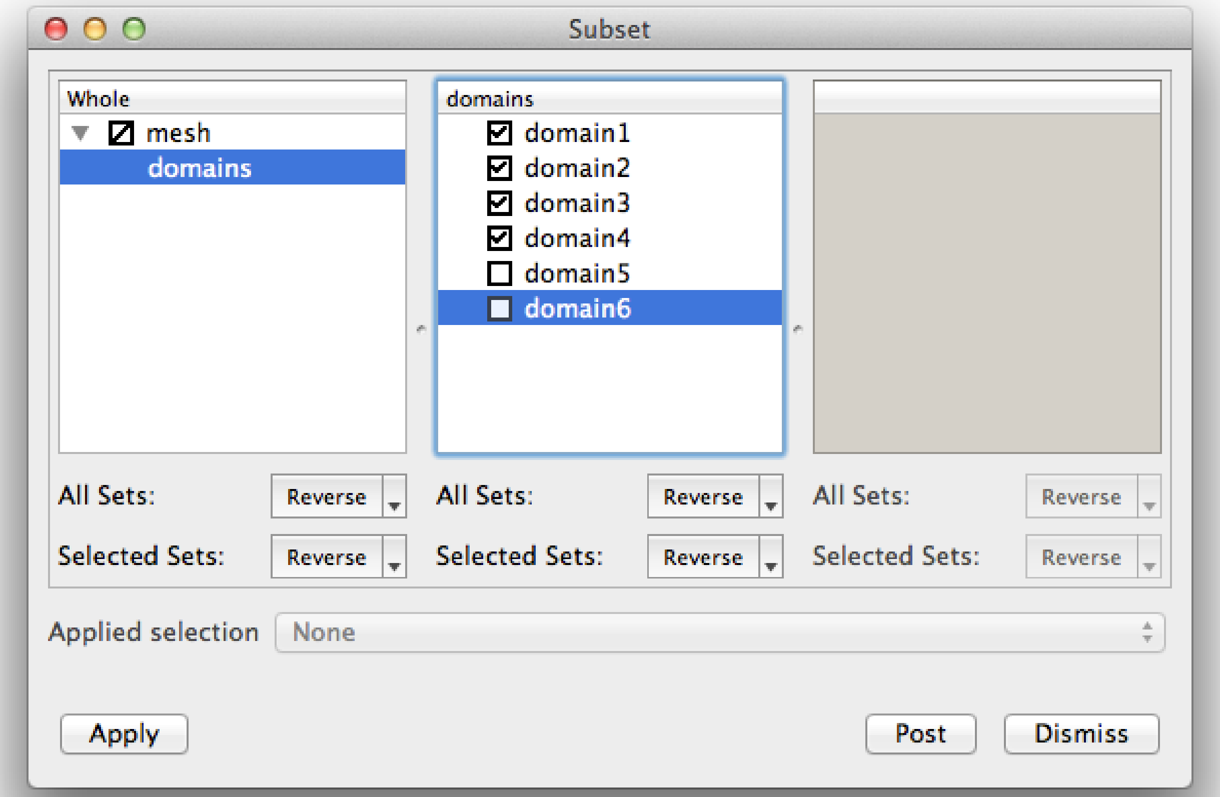
We will start by creating a plot that shows the water tank boundaries.

- Open **dbreak3d_boundaries.silo** file.
- Create a **Subset** Plot of **domains**
- Click Draw



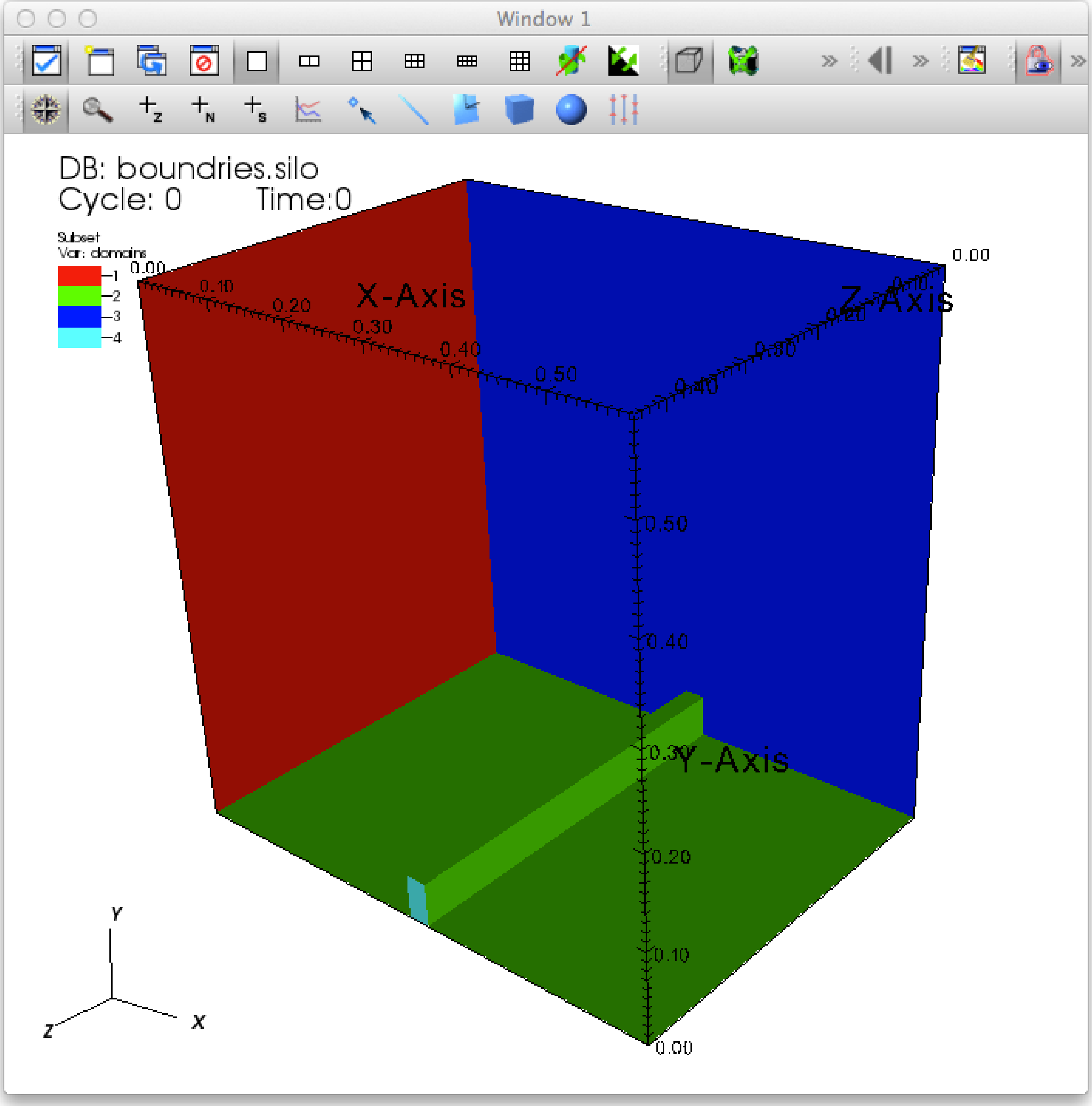
This subset plot shows the different faces that comprise the water tank. We do not want to view all of the boundaries because they will block the fluid data, so next we turn off a few of the boundary faces that are identified as subsets or domains in the data file.

- Open the Set Selection Window (click on the Ven Diagram next to **domains**)
- Under **domains**: Turn off **domain5** and **domain6**



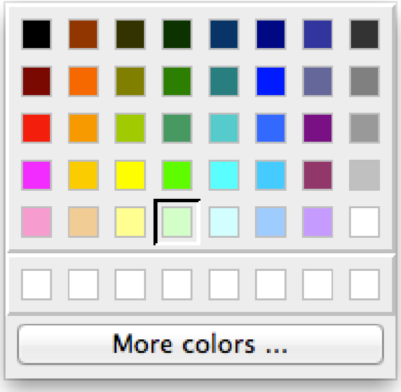
domains
☑ domain1
☑ domain2
☑ domain3
☑ domain4
☐ domain5
☐ domain6

- Click Apply

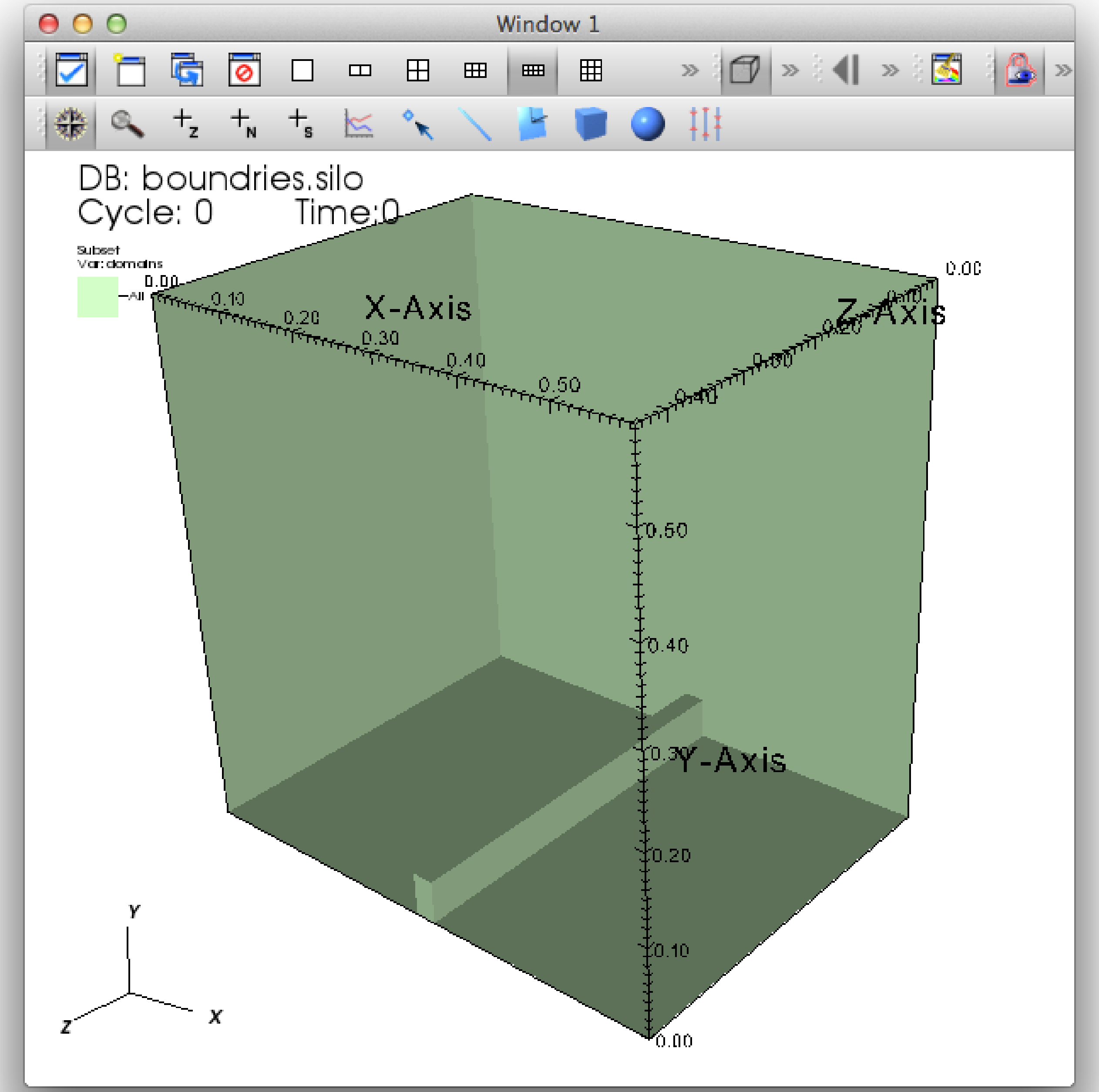


Finally, lets color all the faces with a uniform color:

- Open the **Subset** plot attributes window
- Select **Single** and choose the light pastel green color



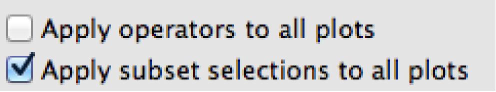
- Click Apply and Dismiss



Viewing the Water sub-volume

Now we will plot the water in the tank.

- Uncheck **Apply operators to all plots**

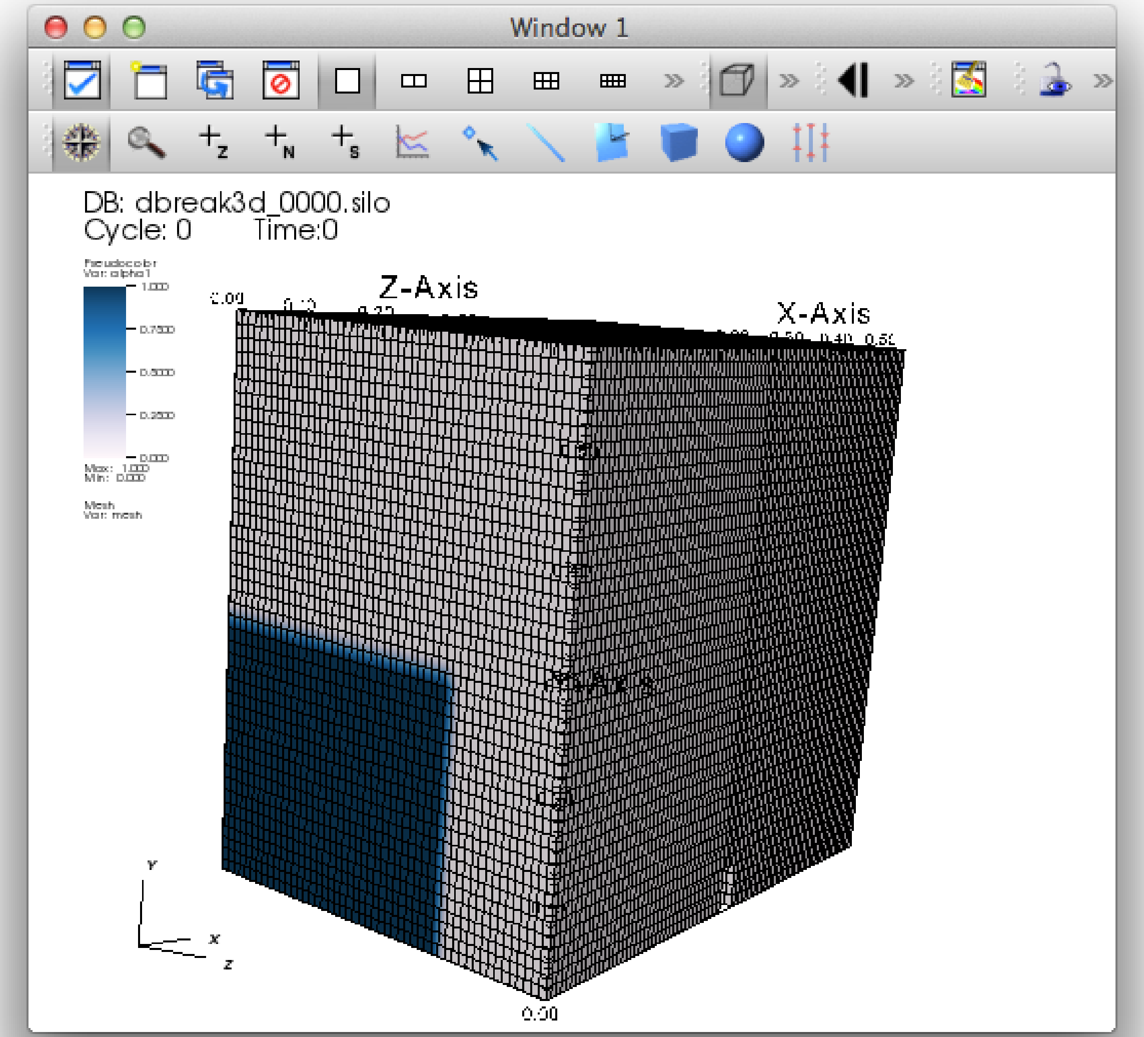


☐ Apply operators to all plots
☑ Apply subset selections to all plots

- Open **dbreakd3d_fluid.visit**
- Create a **Pseudocolor** Plot of **alpha1**
- Double click on the **Pseudocolor** plot to bring up its attributes.

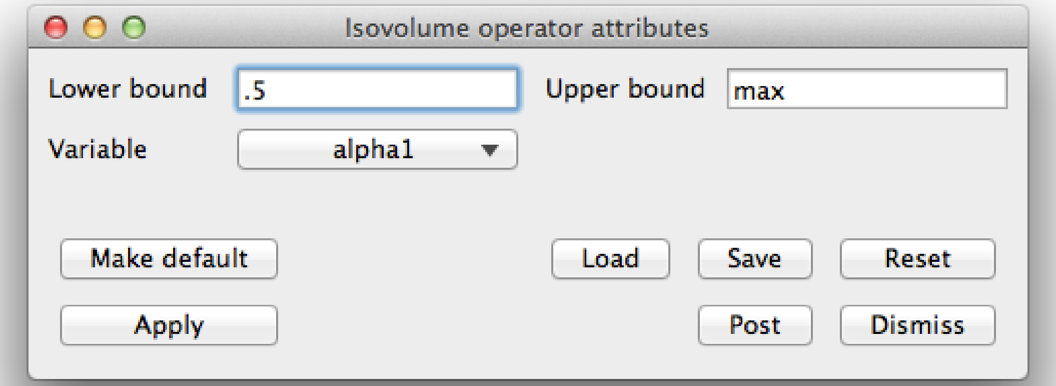
- Change the color table to **PuBu**
- Click Apply and Dismiss
- Create a **Mesh** Plot of **mesh**
- Click Draw
- Rotate to view the back face of the water column.

These are the initial conditions of the simulation.



The **alpha1** variable defines the fraction of each cell's volume that is water. We are interested in the portion of the simulation mesh that are at least 50% water. To extract this sub-volume we use an **Isovolume** Operator:

- Delete the **Mesh** plot
- Add an **Isovolume** Operator (Operators->Selection->Isovolume)
- Double click on the **Isovolume** operator to bring up its attributes
- Set the **Lower bound** to .5
- Select **alpha1** as the **Variable** option

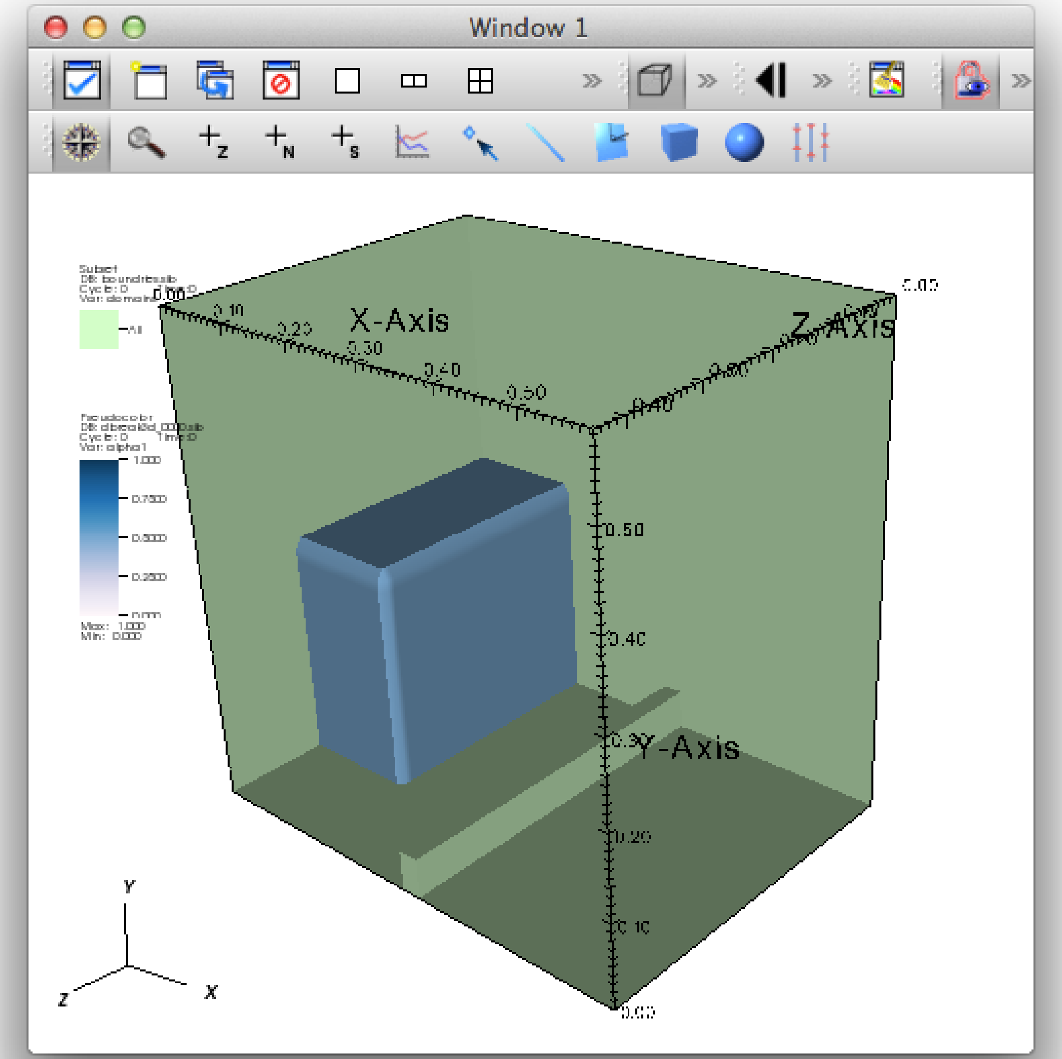


Lower bound .5

Upper bound max

Variable alpha1

- Click Apply and Dismiss
- Create the **Subset** plot of the boundaries dataset as before.
- Click Draw
- Rotate the plot to view the column of water in front of the tank boundaries.



Save Session File with Basic Visualization Setup

We will use these two basic plots as foundation for the rest of the tutorial, so it will be helpful to have a way to easily recreate them.

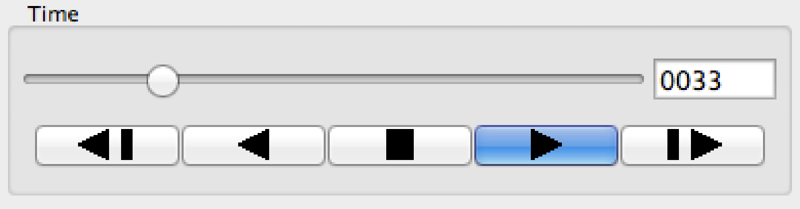
- Save the session (File->Save session as)
    - Set the file name to **dbreak3d_plots_basic.session**

You can use this session file to easily restore these two plots in the future.

You can also use the **dbreak3d_plots_basic.py** python script to reproduce these plots if you have both the **dbreak3d_fluid.visit** and **dbreak3d_boundaries.silo** databases opened.

Animate the simulation

- Click the **Play** button on the timer slider control.



Watch the water in the simulation move according to gravity and the constraints at the tank boundaries.

Later in the tutorial we will make a movie displaying water evolution in the tank.

Exploring Time Varying Properties of the Water

Next we will analyze two time varying aspects of the water flow.

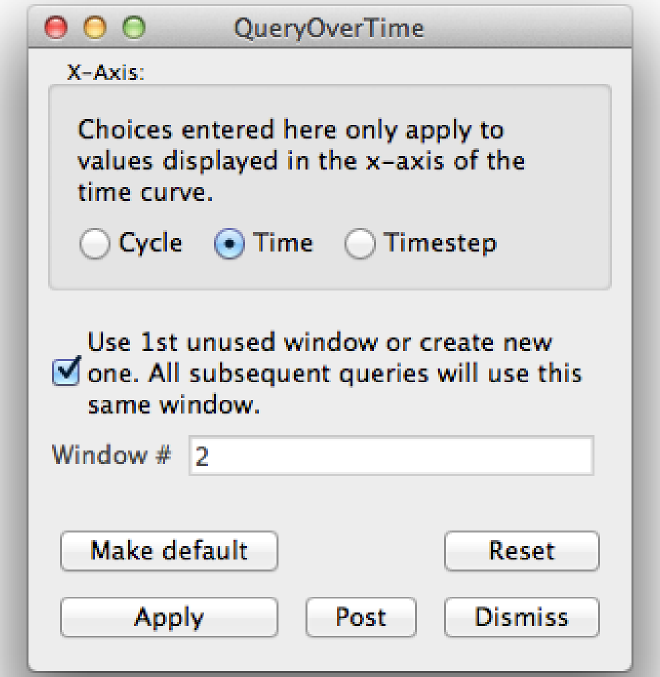
Before we begin, there are two preparation items:

Import dbreak3d Tutorial Expressions

- Open the Expressions Window (Controls->Expressions)
- Click **Load**
- Select the **dbreak3d_exprs.xml** file (available in the dbreak3d folder in the tutorial examples folder.)
- Click Apply and Dismiss

Set Query Over Time Options

- Open the **Query over time options** Window (Controls->Query over time options)
- Select **Time** as the X-axis variable.
- Click Apply and Dismiss


QueryOverTime

X-Axis:

Choices entered here only apply to values displayed in the x-axis of the time curve.

○ Cycle ● Time ○ Timestep

☑ Use 1st unused window or create new one. All subsequent queries will use this same window.

Window # 2

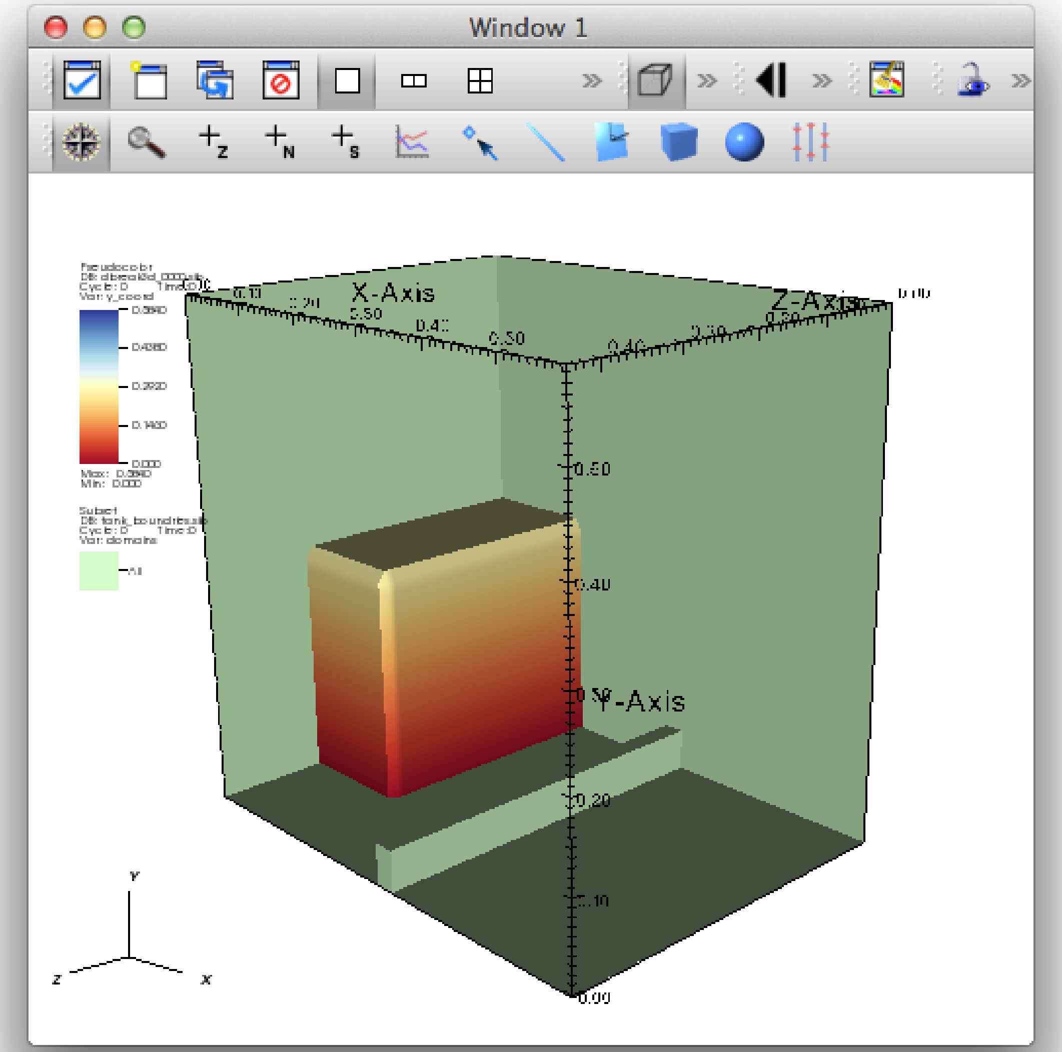
Height of Water Over Time

The first question we will explore is:

What is the highest point in the tank that some volume of water reaches?

We will use VisIt's Expression and Query building blocks to answer the question. First, we need a way to extract the height of the water. We will start with the basic water and tank plot setup described previously and create a plot of **y_coord**. The definition for this expression is **coord(mesh)[1]**, which extracts the second spatial coordinate from the mesh geometry and exposes it as a scalar variable.

- Change the active plot variable to **y_coord**
- Double click on the **Pseudocolor** plot to bring up its attributes.
- Change the color table to **RedYlBlu**
- Click Apply and Dismiss



The color of the plot now varies according to the height of the mesh cells.

Now we use the **Max** query to capture how the highest value varies over the duration of the simulation.

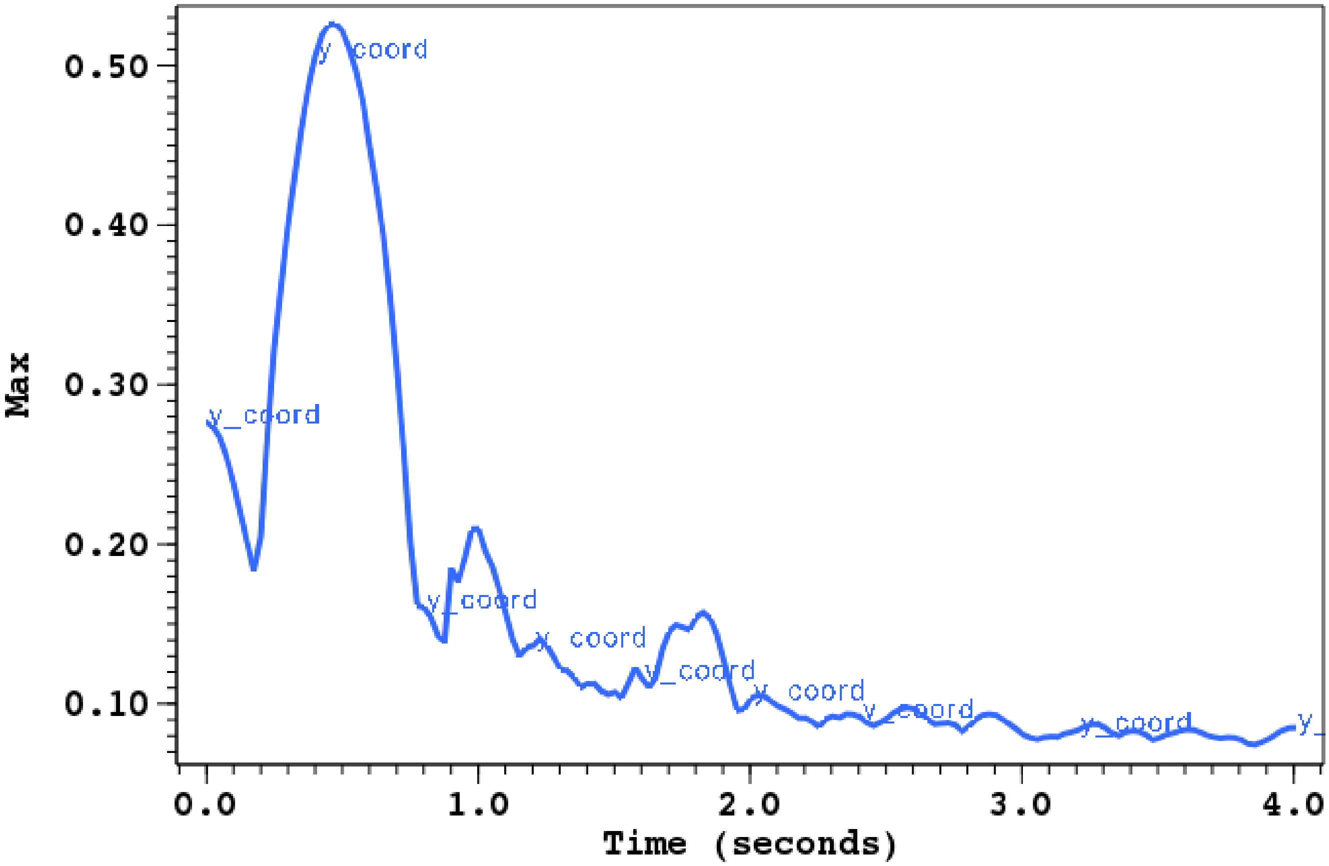
- Open the **Query** Window (Controls->Query)
- Select that **Max** Query

In the query options:

- Select **Actual Data**
- Check **Do Time Query**
- Click Query

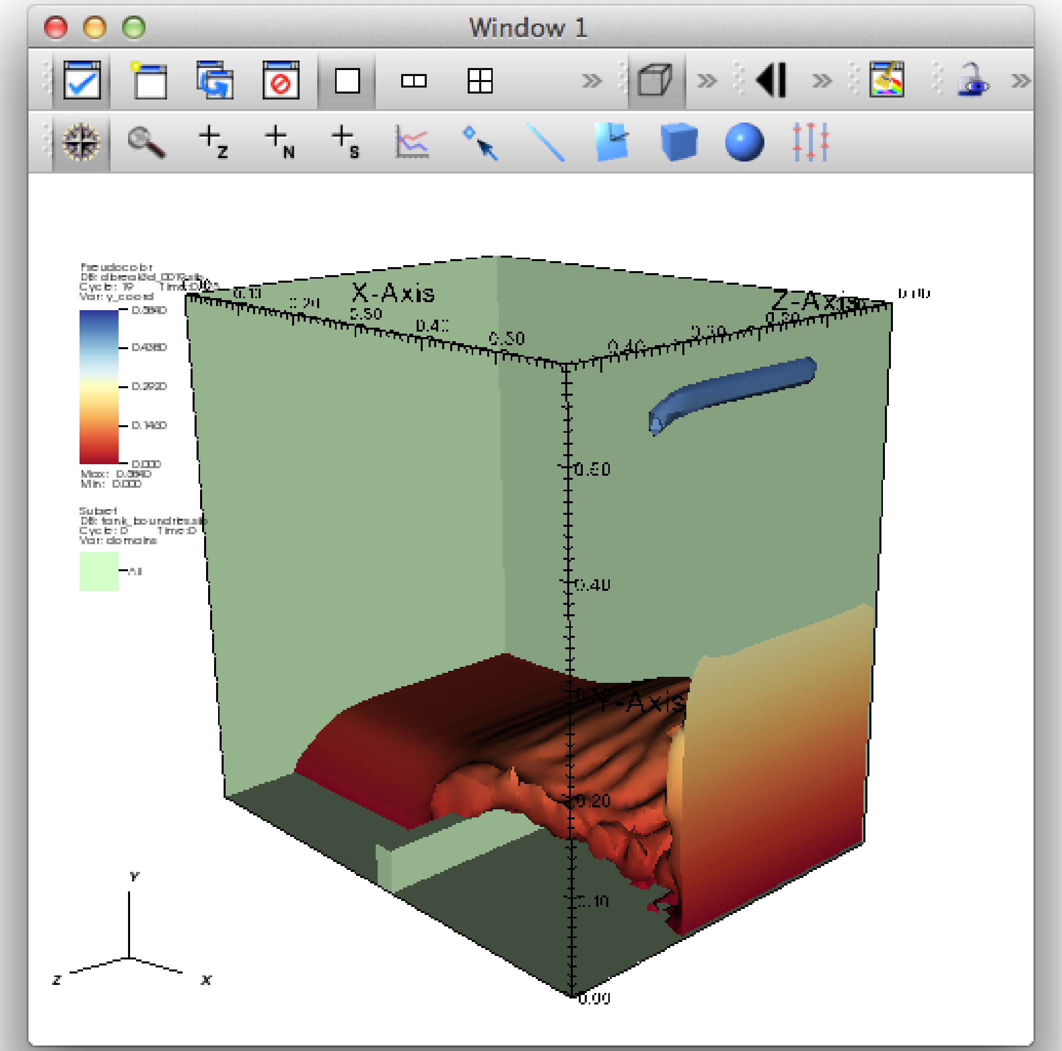
This will process all 160 time steps and extract the max y-coordinate of the mesh cells that are more than 50% water. (It may take a few seconds to complete)

When the query completes a new window will appear with a curve plot that displays the max value vs time.



The curve shows us that the max height decays over time as gravity pulls the water to the bottom of the tank. From this curve we also see that the max height of any volume of water in the simulation happens at 0.475 seconds [timestep =19].

- Return to the window with your **y_coord** plot, and change the time slider to timestep=19.



At this time step we can see a blob of water near the top of the tank reach its maximum height as gravity overcomes its upward velocity.

- Use the time slider to view time steps before and after timestep = 19.

Number of Water Droplets Over Time

The next question we will explore about the water is:

How do blobs of water break away from the main water volume as the water column splashes and settles?

Again, we will use VisIt's Expression and Query building blocks to answer the question.

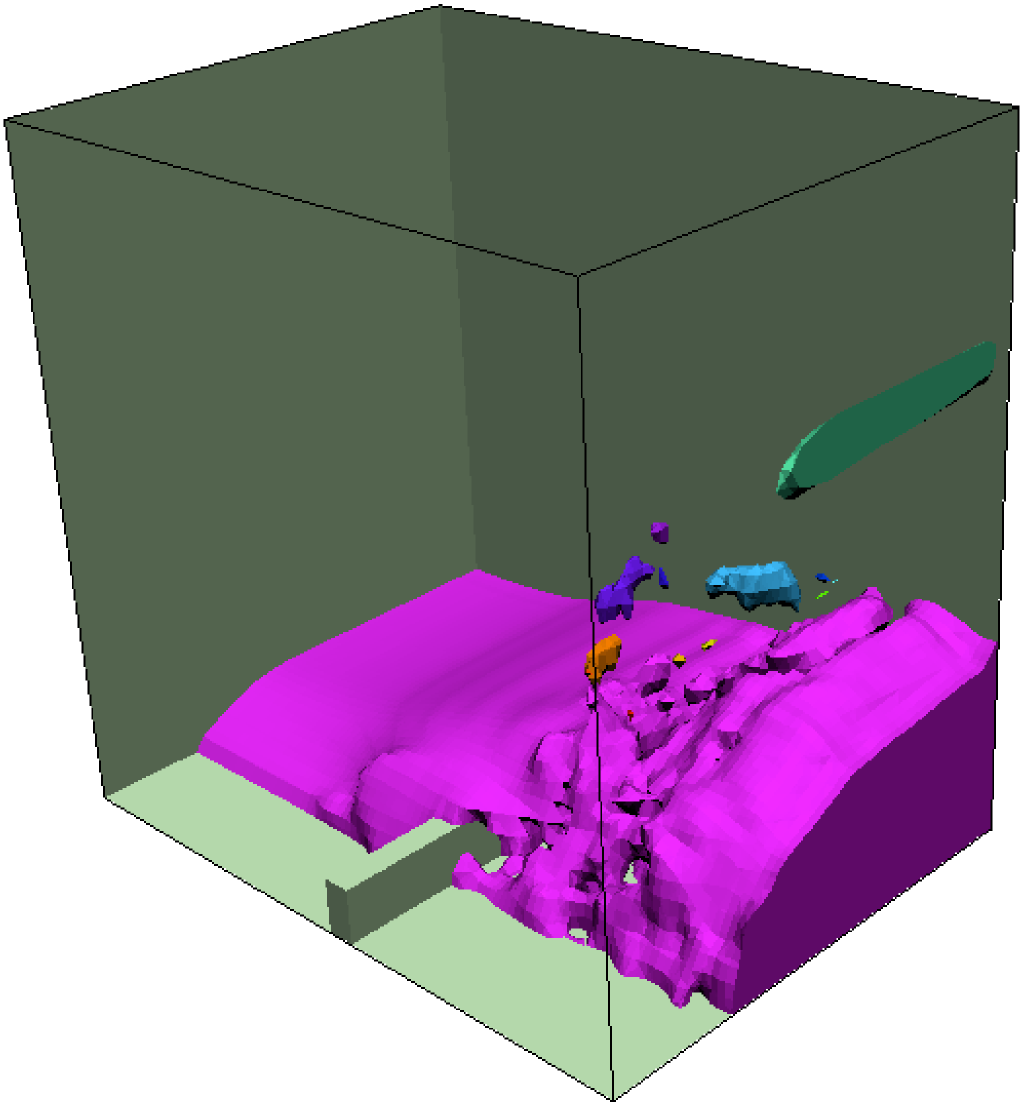
VisIt's connected components algorithm allows us to identify topologically distinct sub-volumes of a mesh. In the case of our water flow simulation we will use it to identify blobs or droplets of water that separate from the main water volume.

We will use the **cc_label** expression to view the droplets. First we need to make sure execute the **cc_label** expression after the **Isovolume** selects the water sub-volume.

- Select time step 25
- Add a **DeferExpression** Operator (Operators->Analysis->DeferExpression)
- Double click on the **DeferExpression** operator to bring up its attributes
- Enter **cc_label** in the **Variables** field.
- Click Apply

Select a different color table too improve the contrast of the labels:

- Double click to open the **Pseudocolor** attributes.
- Select the **rainbow** color table
- Click Apply
- Click Draw



The colors identify the different connected sub-volumes of the water. Next, lets execute the time varying query to see how the number of connected blobs evolves over time:

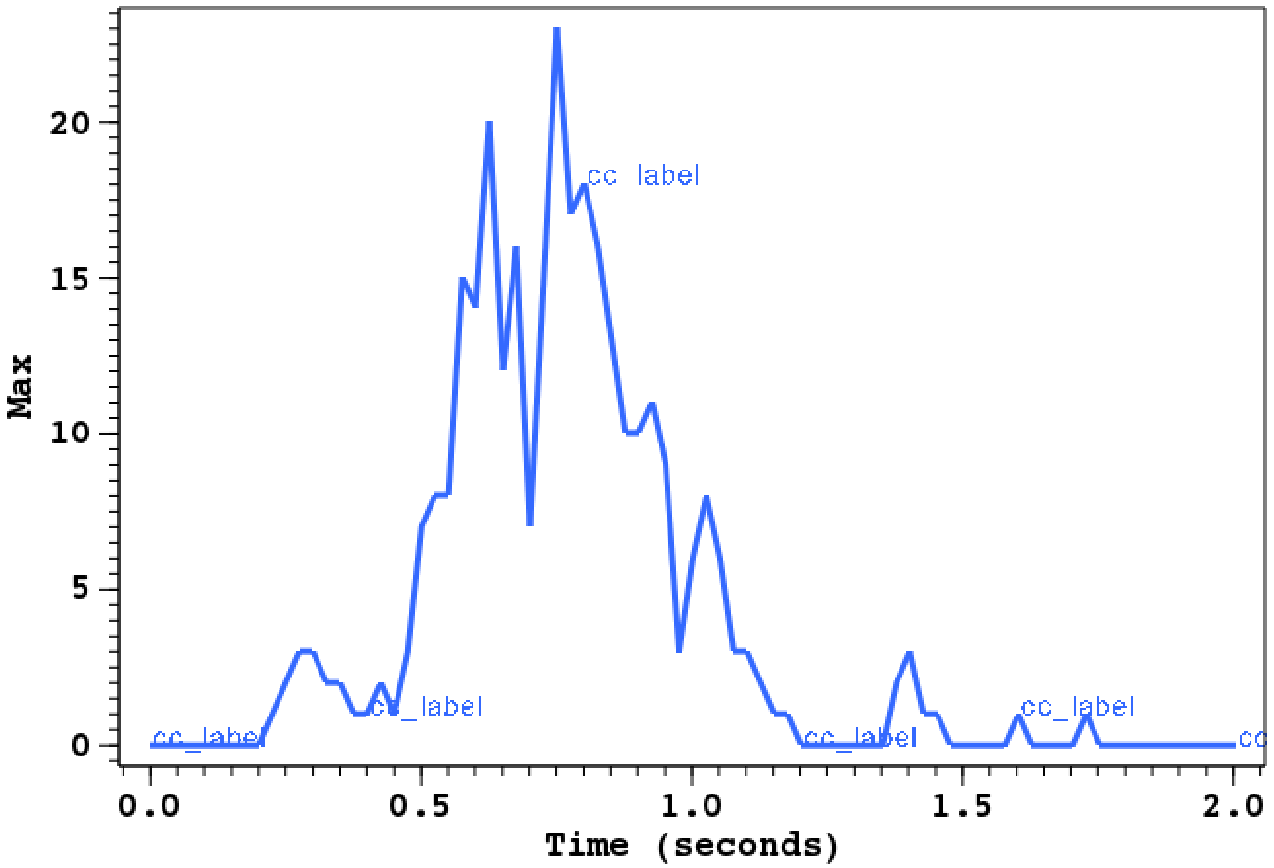
- Open the **Query** Window (Controls->Query)
- Select that **Max** Query

In the query options:

- Select **Actual Data**
- Check **Do Time Query**
- Set the **Ending timestep** to 80 (this is after that water settles down back into one connected sub-volume)
- Click Query

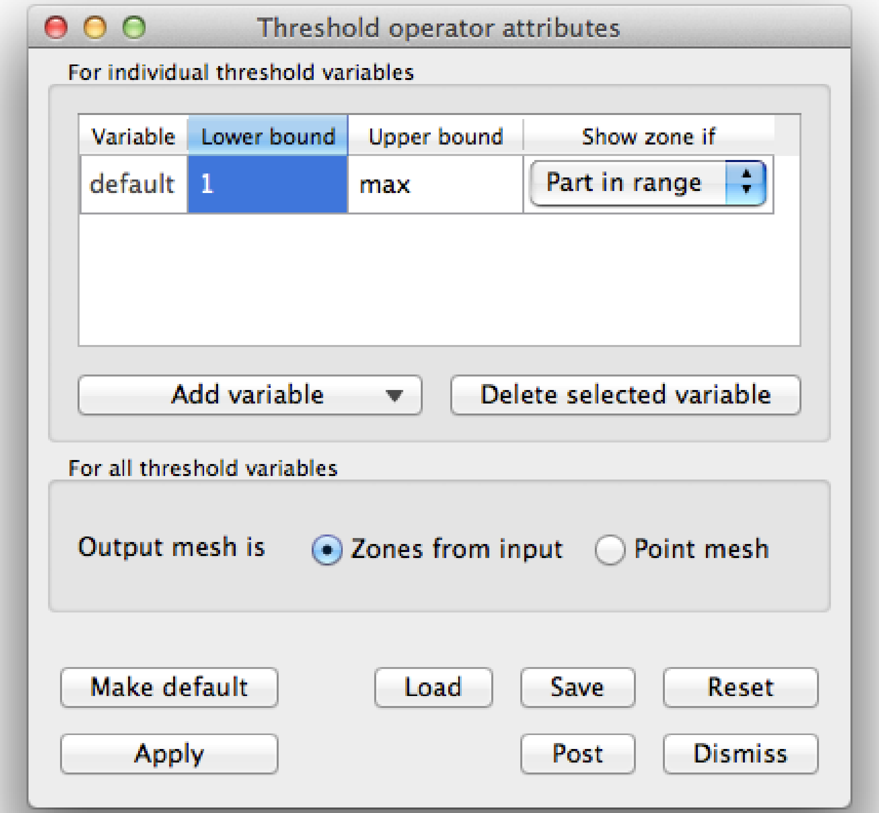
This will process 80 timesteps and extract the max connected component id. (It may take a few seconds to complete)

When the query completes a new window will appear with a curve plot that displays the max value vs time.



We can see that the water breaks into multiple chunks after the column splashes into the right wall of the tank. The maxim number of components occurs at t=0.75 seconds [timestep = 30] We can use a threshold operator to isolate the smaller components from the bulk of the water.

- Select time step 30
- Add a **Threshold'** ***operator to the current*** *cc_label* plot (Operators->Selection->Threshold)
- Double click on the **Threshold** operator to bring up its attributes
- Set the **Minimum** to be 1



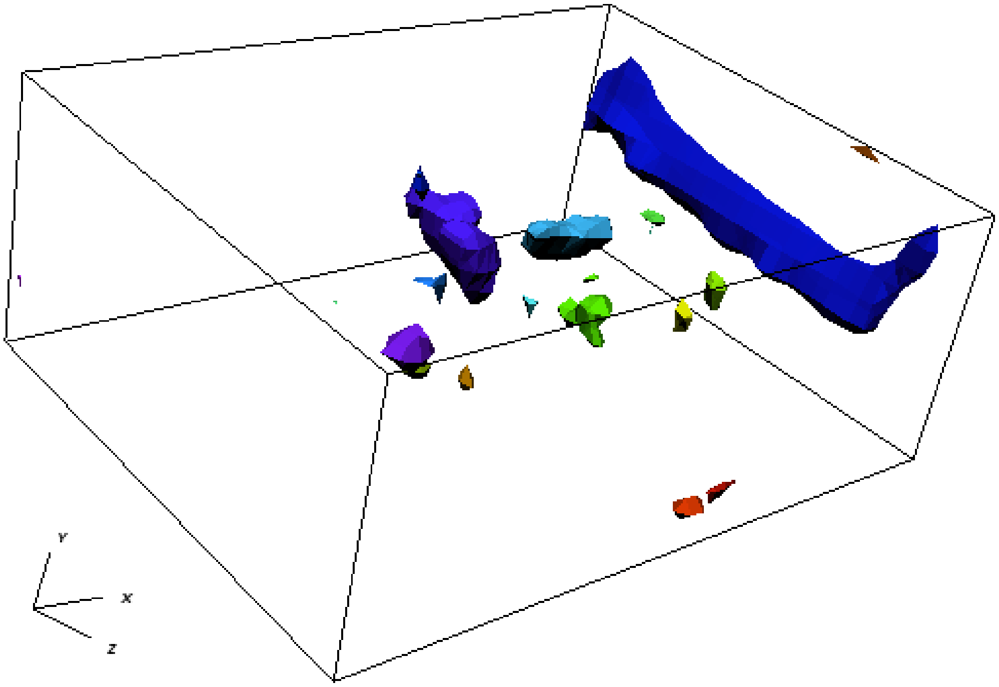
Threshold operator attributes
For individual threshold variables
Add variable
Delete selected variable
For all threshold variables
Make default
Load
Save
Reset
Apply
Post
Dismiss

Variable	Lower bound	Upper bound	Show zone if
default	1	max	Part in range

Output mesh is ● Zones from input ○ Point mesh

- Click Apply and Dismiss
- Click Draw

There are 24 components at this time step (Labeled 0 though 23) -- we removed the largest which happens to be labeled 0. We can see that many of the components are quite small volumes of water:



Exploring the Velocity Vector Field

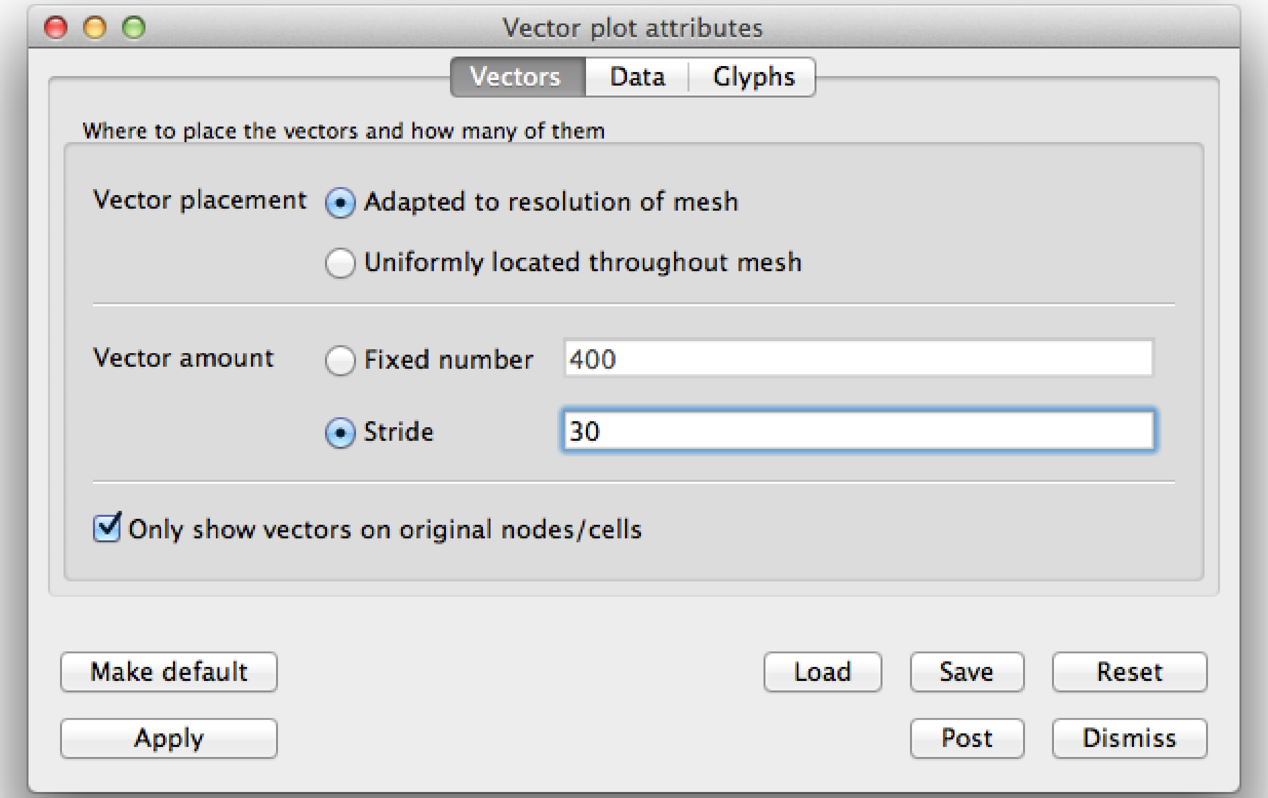
We will now explore three ways to visualize the velocity vector field of the simulation.

Plotting the Velocity Field using Vector Plots

First we will directly plot the velocity vectors that exist on the mesh nodes.

- Start with the basic tank boundaries and water sub-volume plots outlined previously.
- Use the time slider to select timestep = 1 (The velocity at timestep = 0 is fully zero)

- Add a **Vector** plot of **U**
- Open the **Vector** plot attributes
- On the 1st tab (Vectors), set the **Stride** to 30
- On the 2nd tab (Data), select the **RdYlBu** color table
- On the 3rd tab (Glyphs)
  - Set **Scale** to **0.75**
  - Set **Arrow body** to **Cylinder**
  - Set **Geometry Quality** to **High**

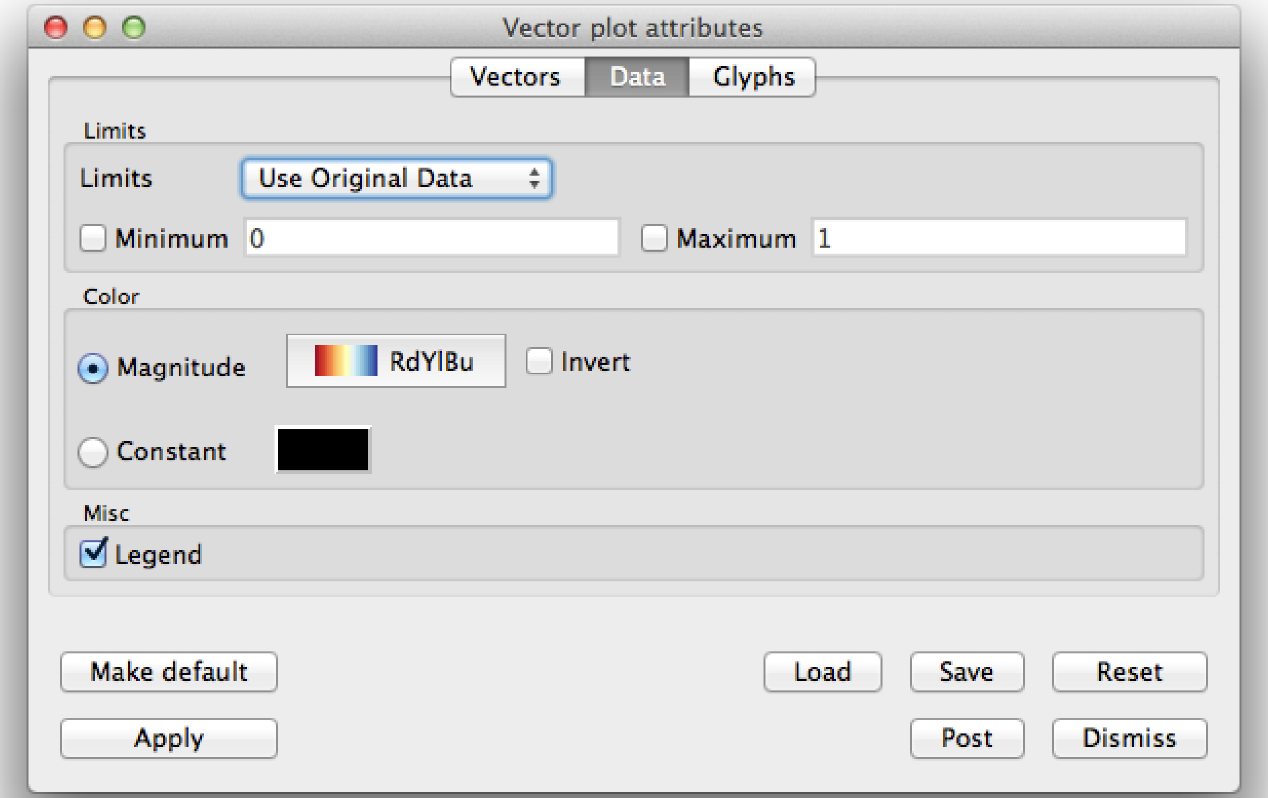


Where to place the vectors and how many of them

Vector placement ● Adapted to resolution of mesh
○ Uniformly located throughout mesh

Vector amount ○ Fixed number 400
● Stride 30

☑ Only show vectors on original nodes/cells



Limits

Limits: Use Original Data

☐ Minimum 0 ☐ Maximum 1

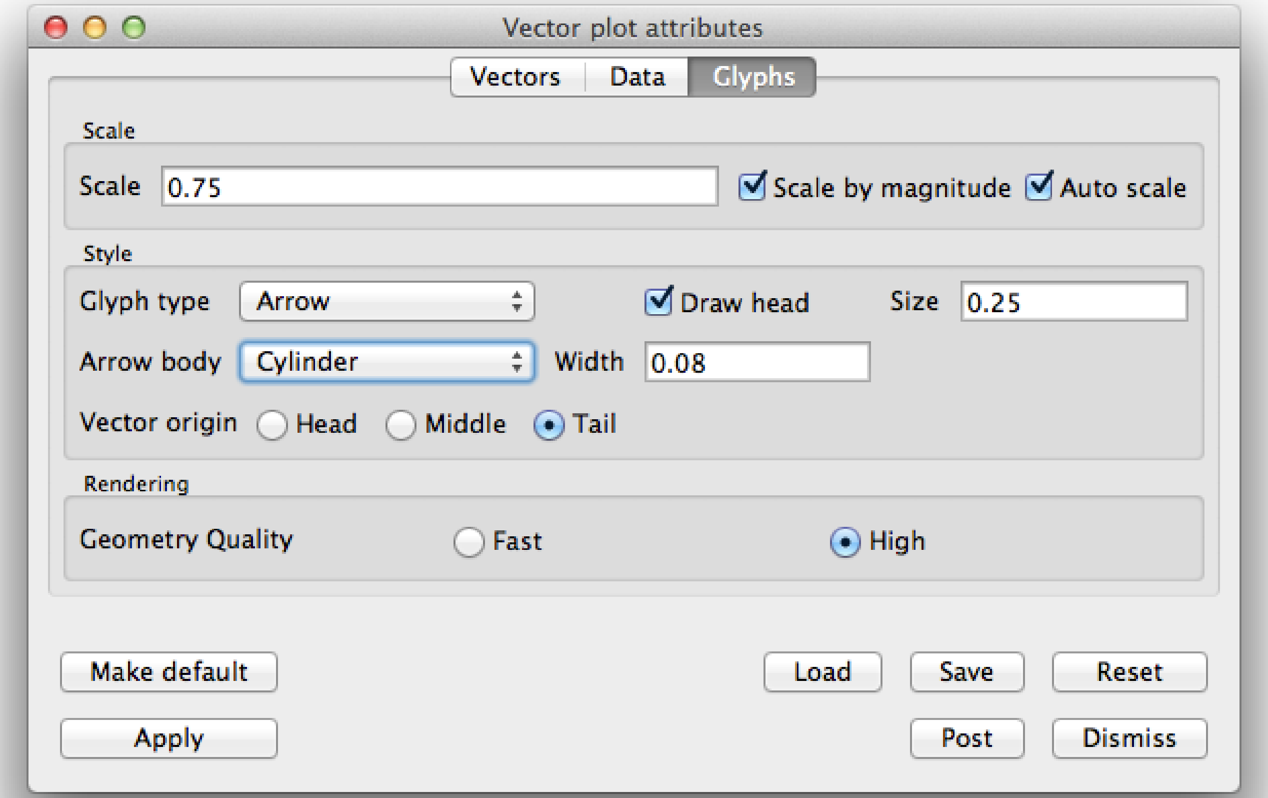
Color

● Magnitude RdYlBu ☐ Invert

○ Constant

Misc

☑ Legend



Scale

Scale 0.75 ☑ Scale by magnitude ☑ Auto scale

Style

Glyph type Arrow ☑ Draw head Size 0.25

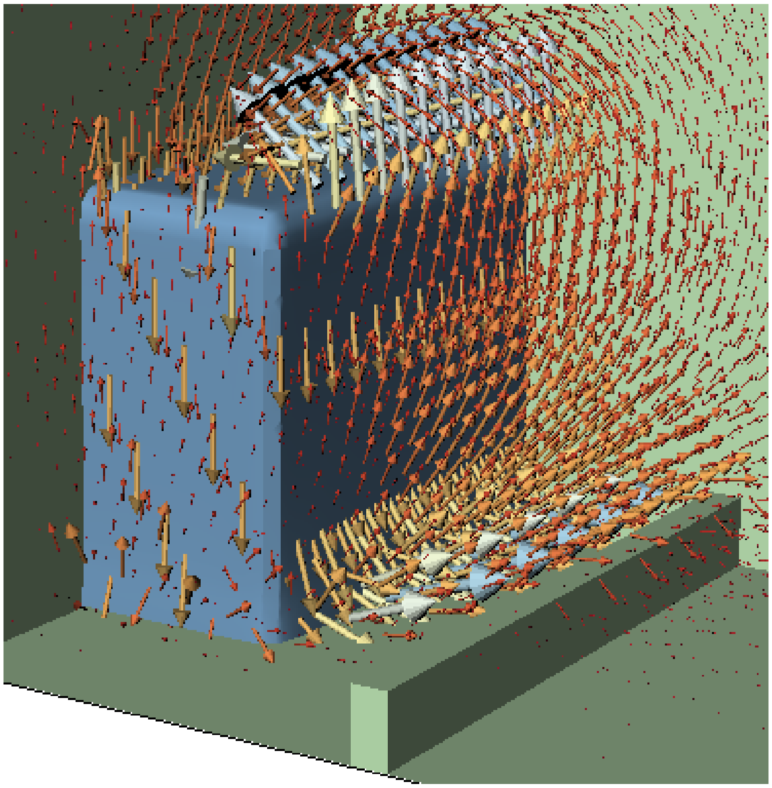
Arrow body Cylinder Width 0.08

Vector origin ○ Head ○ Middle ● Tail

Rendering

Geometry Quality ○ Fast ● High

- Click Apply and Dismiss
- Click Draw
- Zoom in to explore the plot results.



The vector plot uses glyphs to draw portions of the instantaneous vector field. The arrows are colored according to the speed at each point (the magnitude of the velocity vector). Next we will look at a more advanced technique to visualize the flow of the velocity vector field.

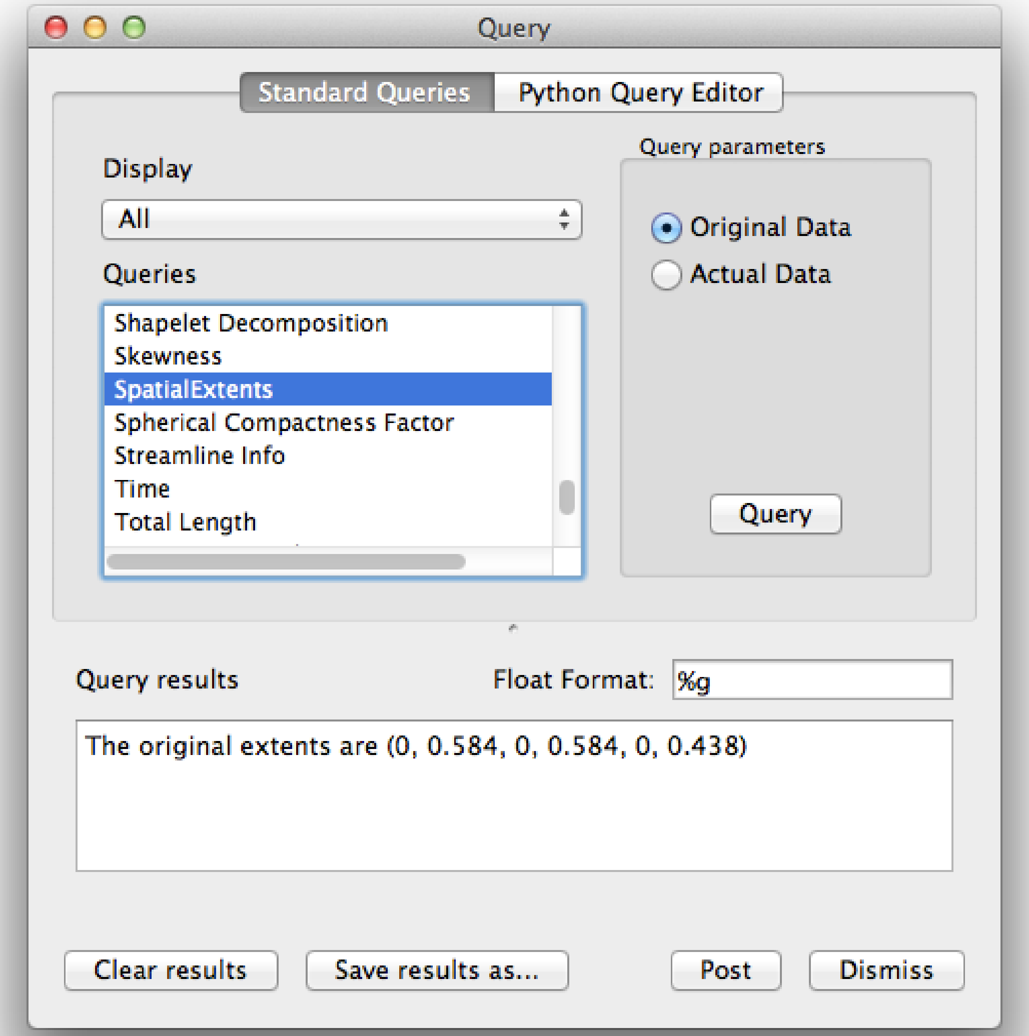
Streamlines at the Advancing Interface

Streamlines visualize flow by advecting a set of massless particles, initially placed at user selected seed points, through an instantaneous vector field. In our simulation, as the column of water falls it quickly pushes the water at the bottom outward and it collides with a small ledge. We can use a **Streamline** plot to explore the water velocity field at the advancing water front.

First we want to get an idea of where the advancing portion of the water is:

- Start with the basic tank boundaries and water sub-volume plots outlined previously.

- Select the **Pseudocolor** plot of **alpha1**
- Open the **Query** Window (Controls->Query)
- Execute a **SpatialExtents** Query with **Original Data** selected



Display: All

Queries: SpatialExtents

● Original Data

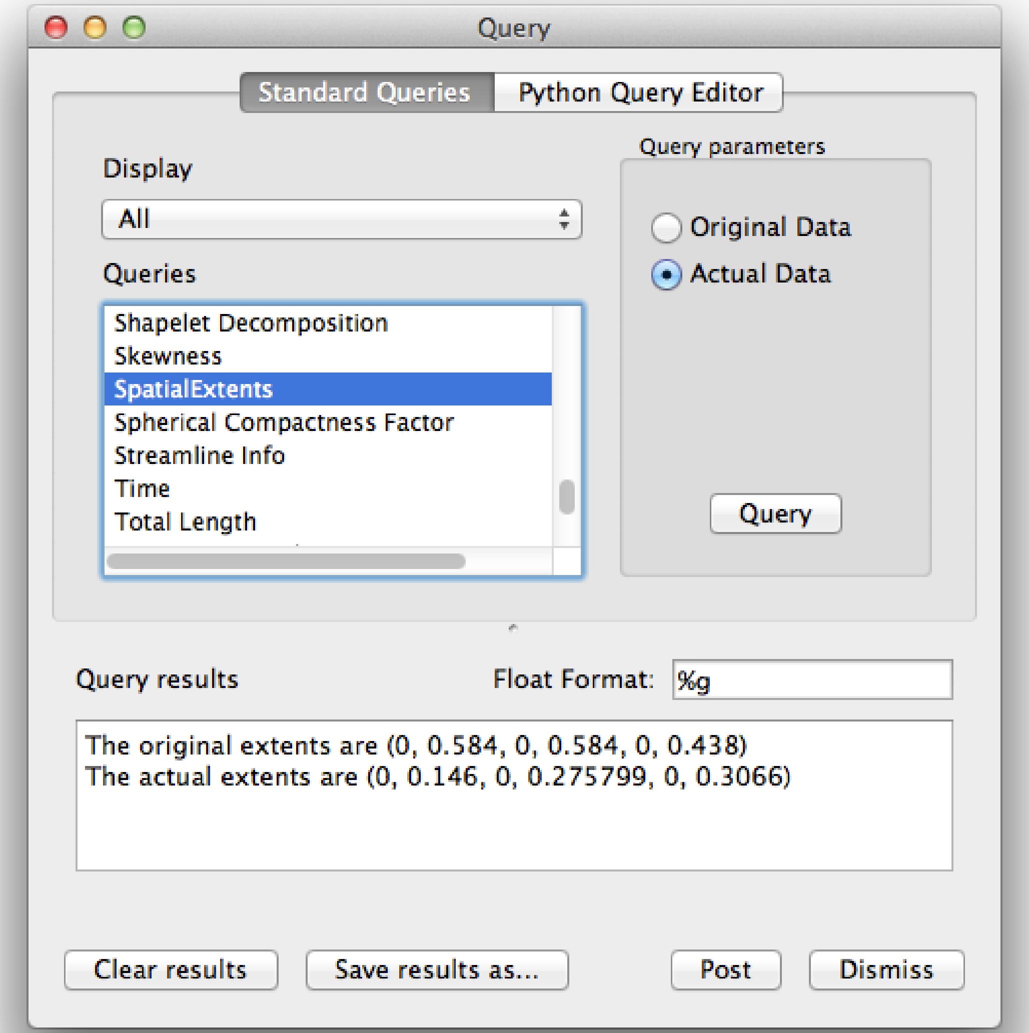
Float Format: %g

The original extents are (0, 0.584, 0, 0.584, 0, 0.438)

This gives us the full extent of the fluid simulation mesh:

```
(0, 0.584, 0, 0.584, 0, 0.584)
```

- Now execute a **SpatialExtents** Query with **Actual Data** selected.



○ Original Data
● Actual Data

Float Format: %g

The original extents are (0, 0.584, 0, 0.584, 0, 0.438)
The actual extents are (0, 0.146, 0, 0.275799, 0, 0.3066)

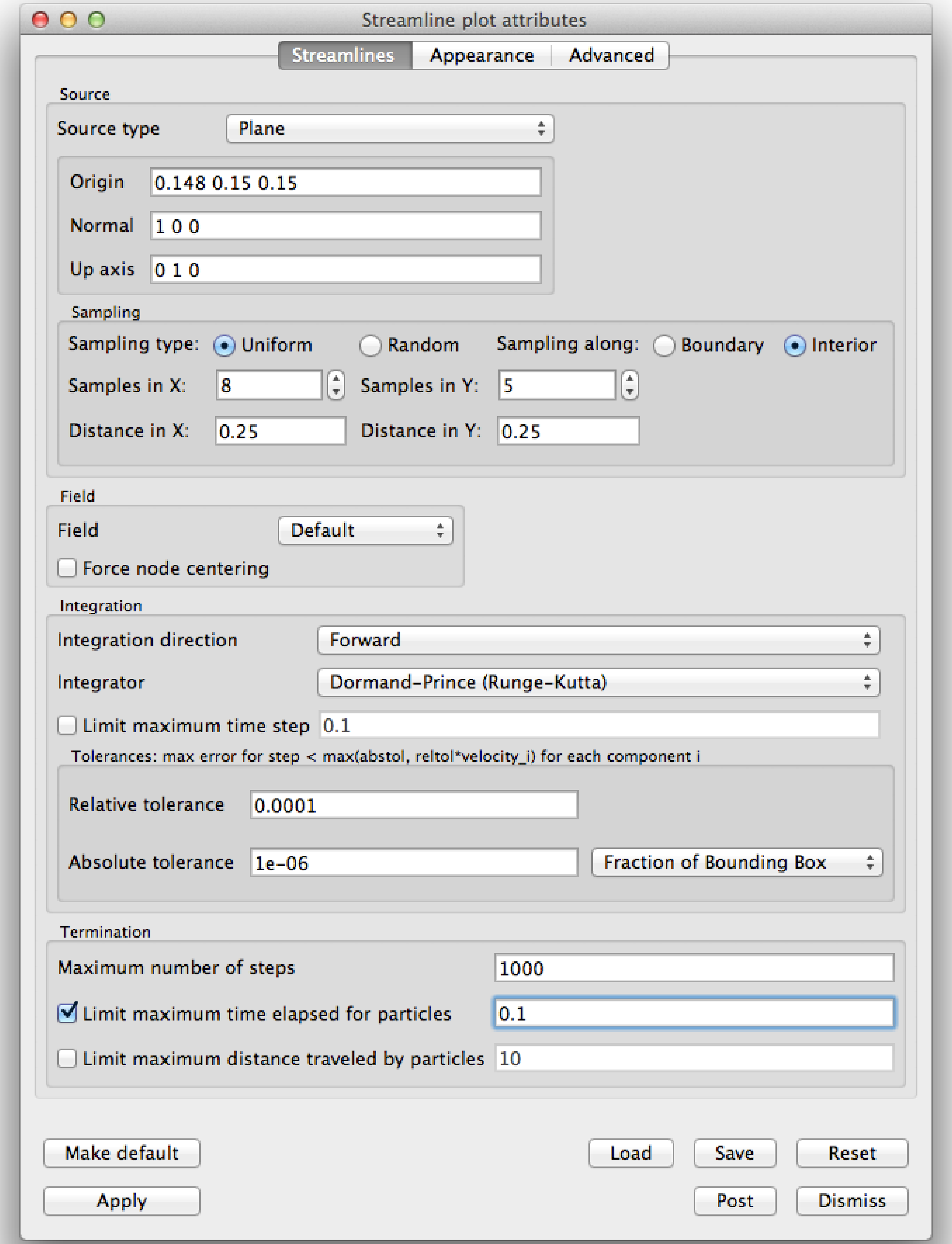
This gives us the extents of just the water column:

```
(0, 0.146, 0, 0.275799, 0, 0.3066
```

With this info in hand, we can create a **Streamline** plot with seed points near this interface.

- Use the time slider to select time step = 1 (The velocity at time step = 0 is fully zero)
- Add a **Streamline** plot of **U**
- Open the **Streamline** attributes
  - Under the 1st tab (Streamlines)
    - Create a Plane source at

        - Origin: 0.148 0.15 0.15
        - Normal: 1 0 0
        - Up axis: 0 1 0
        - Samples in X: 8
        - Samples in Y: 5
        - Distance in X: 0.25
        - Distance in Y: 0.25
    - For termination:
        - Set **Limit maximum time elapsed for particles** to .1
- Under the 2nd tab (Appearance)
    - Select the **RdYlBlu** color table
    - Draw as **Tubes**
    - Select **Show heads** and **Display as Cone**
    - Set the heads **Radius** to 0.012
    - Set the heads **Height:Radius Ratio** to 4
- Under the 3rd tab (Advanced)
    -
        - Turn off all warnings



Source

Source type: Plane

Origin: 0.148 0.15 0.15

Normal: 1 0 0

Up axis: 0 1 0

Sampling

Sampling type: ● Uniform ○ Random

Sampling along: ○ Boundary ● Interior

Samples in X: 8

Samples in Y: 5

Distance in X: 0.25

Distance in Y: 0.25

Field

Field: Default

☐ Force node centering

Integration

Integration direction: Forward

Integrator: Dormand-Prince (Runge-Kutta)

☐ Limit maximum time step: 0.1

Tolerances: max error for step < max(abstol, reltol*velocity_i) for each component i

Relative tolerance: 0.0001

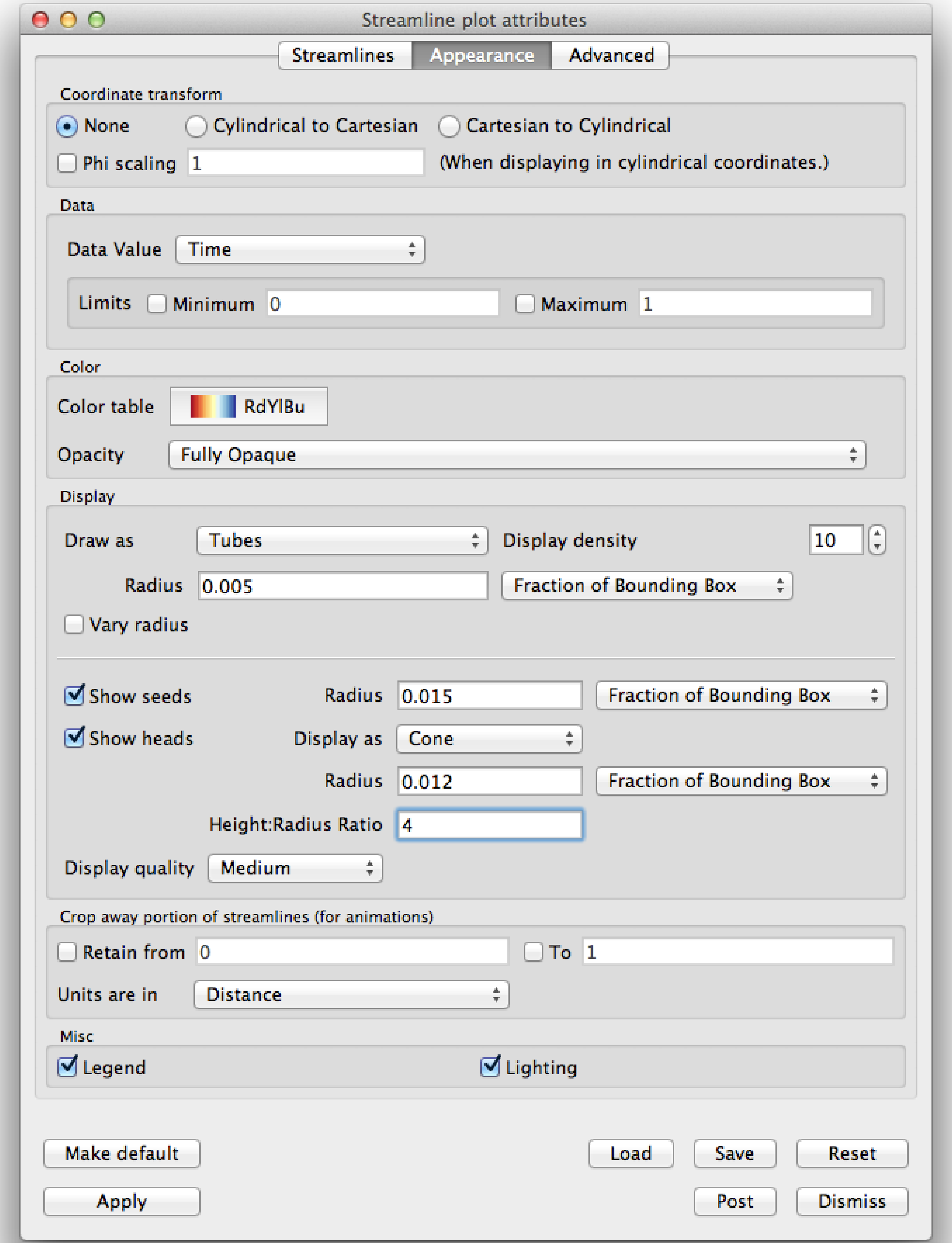
Absolute tolerance: 1e-06 Fraction of Bounding Box

Termination

Maximum number of steps: 1000

☑ Limit maximum time elapsed for particles: 0.1

☐ Limit maximum distance traveled by particles: 10



Coordinate transform

● None ○ Cylindrical to Cartesian ○ Cartesian to Cylindrical

☐ Phi scaling 1 (When displaying in cylindrical coordinates.)

Data

Data Value: Time

Limits ☐ Minimum 0 ☐ Maximum 1

Color

Color table: RdYlBu

Opacity: Fully Opaque

Display

Draw as: Tubes

Display density: 10

Radius: 0.005 Fraction of Bounding Box

☐ Vary radius

☑ Show seeds Radius: 0.015 Fraction of Bounding Box

☑ Show heads Display as: Cone

Radius: 0.012 Fraction of Bounding Box

Height:Radius Ratio: 4

Display quality: Medium

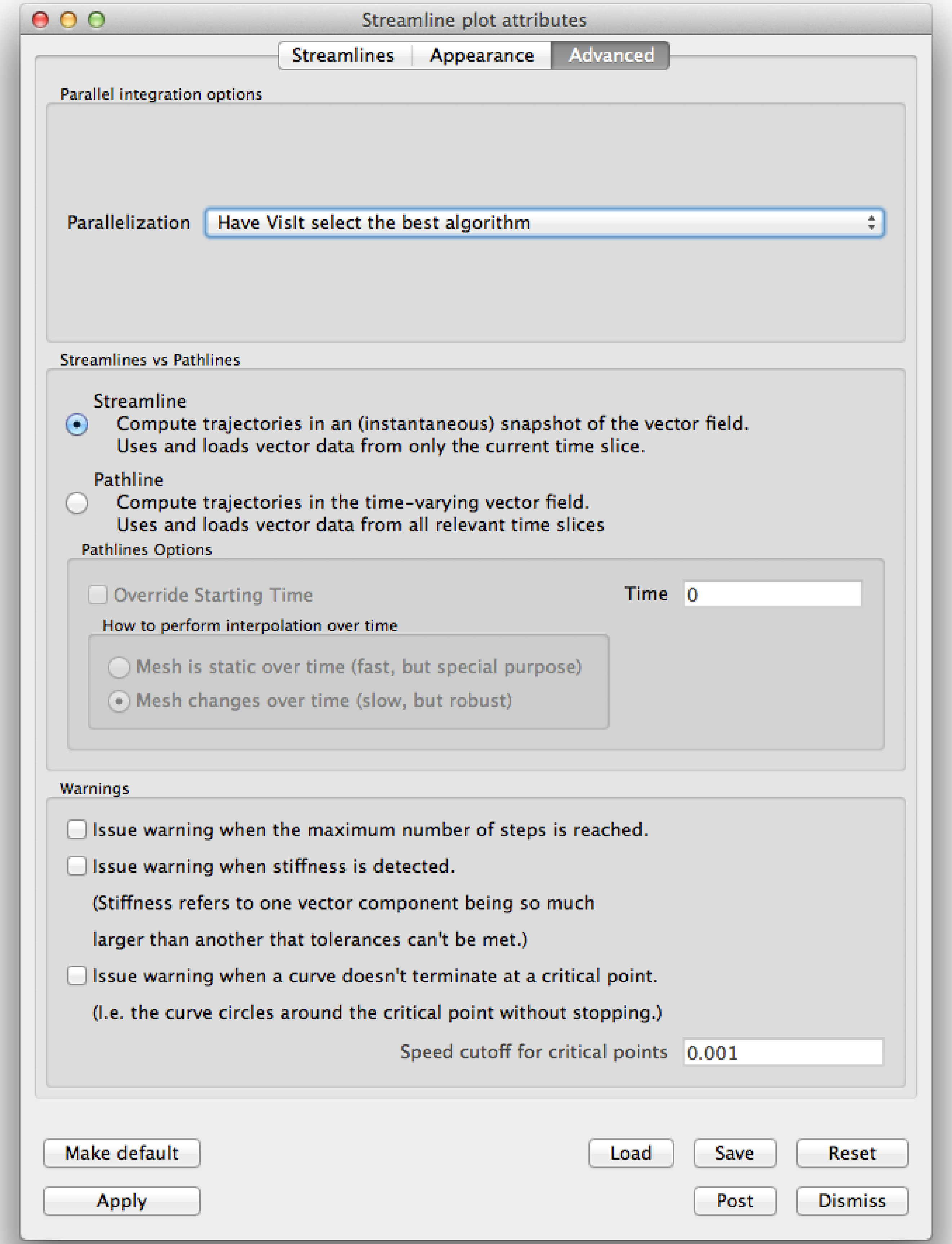
Crop away portion of streamlines (for animations)

☐ Retain from 0 ☐ To 1

Units are in: Distance

Misc

☑ Legend ☑ Lighting



Parallel integration options

Parallelization: Have VisIt select the best algorithm

Streamlines vs Pathlines

● Streamline
Compute trajectories in an (instantaneous) snapshot of the vector field.
Uses and loads vector data from only the current time slice.

○ Pathline
Compute trajectories in the time-varying vector field.
Uses and loads vector data from all relevant time slices

Pathlines Options

☐ Override Starting Time

Time: 0

How to perform interpolation over time

○ Mesh is static over time (fast, but special purpose)
● Mesh changes over time (slow, but robust)

Warnings

☐ Issue warning when the maximum number of steps is reached.

☐ Issue warning when stiffness is detected.
(Stiffness refers to one vector component being so much
larger than another that tolerances can't be met.)

☐ Issue warning when a curve doesn't terminate at a critical point.
(I.e. the curve circles around the critical point without stopping.)

Speed cutoff for critical points: 0.001

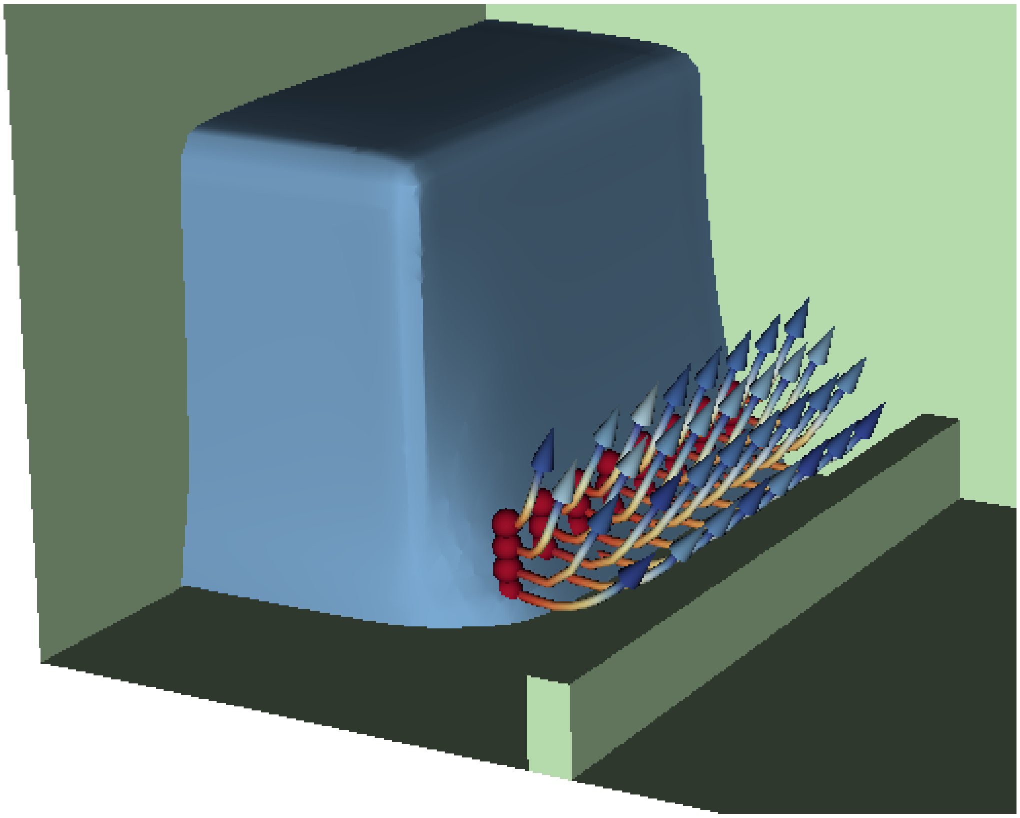
- Click Apply and Dismiss
- Click Draw

At this point save your visualization session:

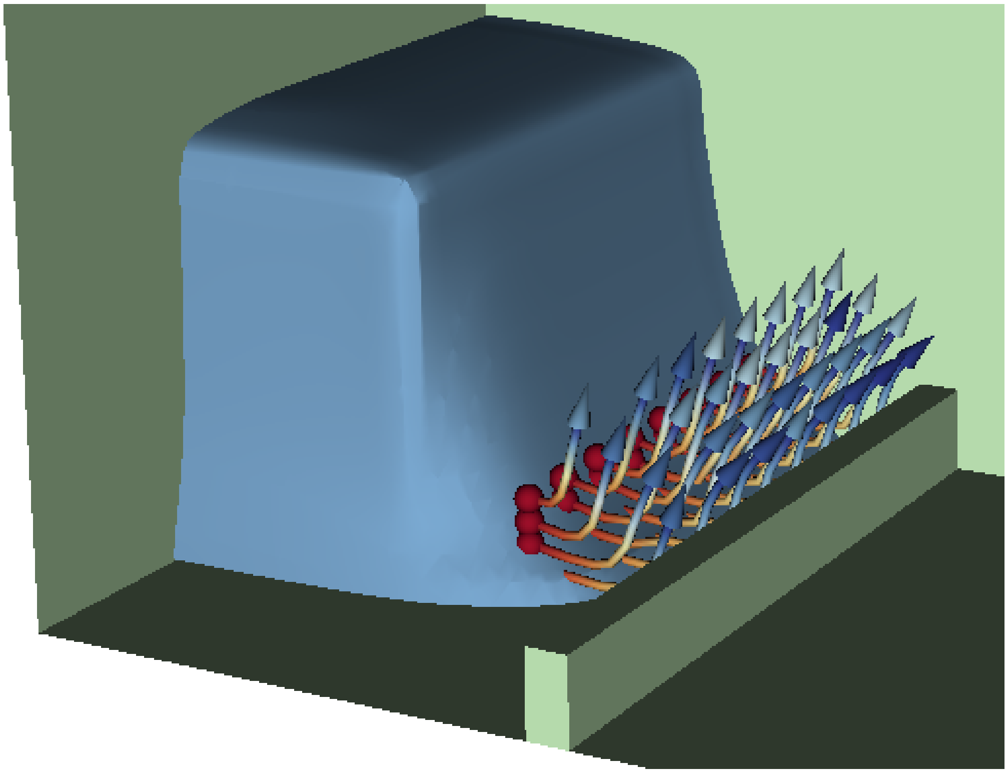
- Save the session (File->Save session as)
    - Set the file name to **dbreak3d_plot_streamlines.session**

You can experiment with both advancing the time slider and moving the X-origin of the plane used to define the seed points.

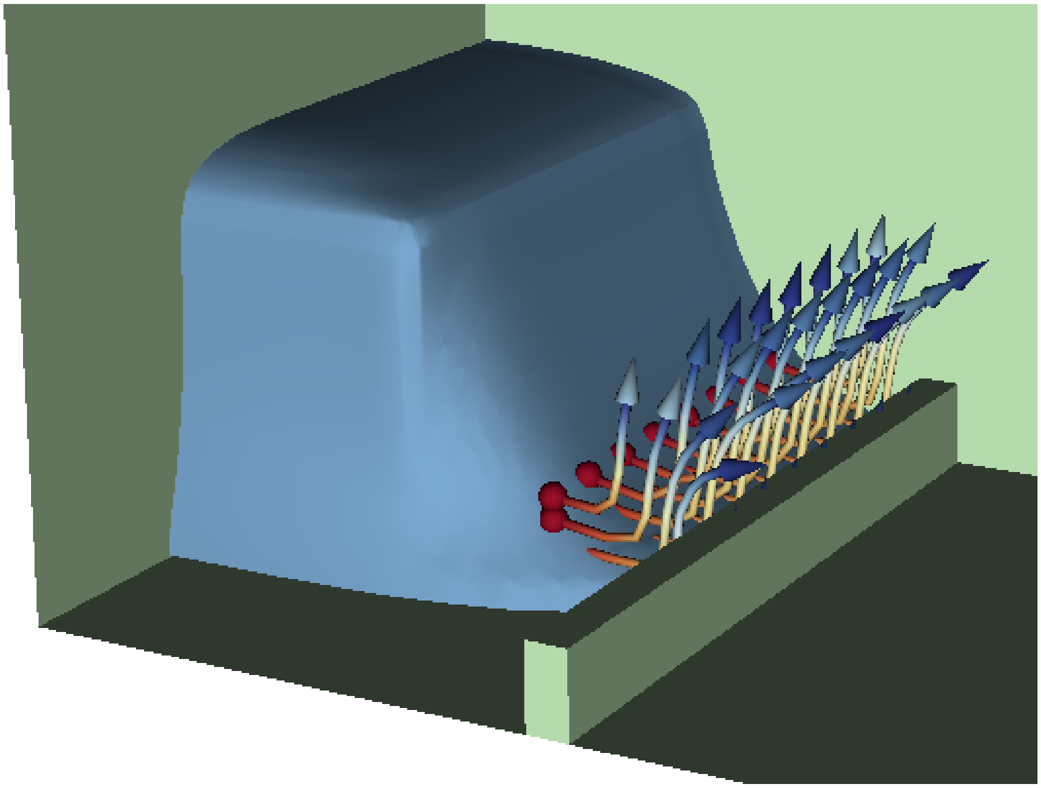
- Open the **Streamline** attributes
- Select time step = 3
- Under the 1st tab (Streamlines) set:
    - Origin: 0.18 0.025 0.15
    - Distance in Y: 0.05
- Click Apply



- Select time step = 4:
- Under the 1st tab (Streamlines) set:
    - Origin: 0.195 0.025 0.15
    - Distance in Y: 0.05
- Click Apply



- Select time step = 5:
- Open the **Streamline** attributes
- Under the 1st tab(Streamlines) set:
  - Origin: 0.210 0.025 0.15
  - Distance in Y: 0.05
- Click Apply



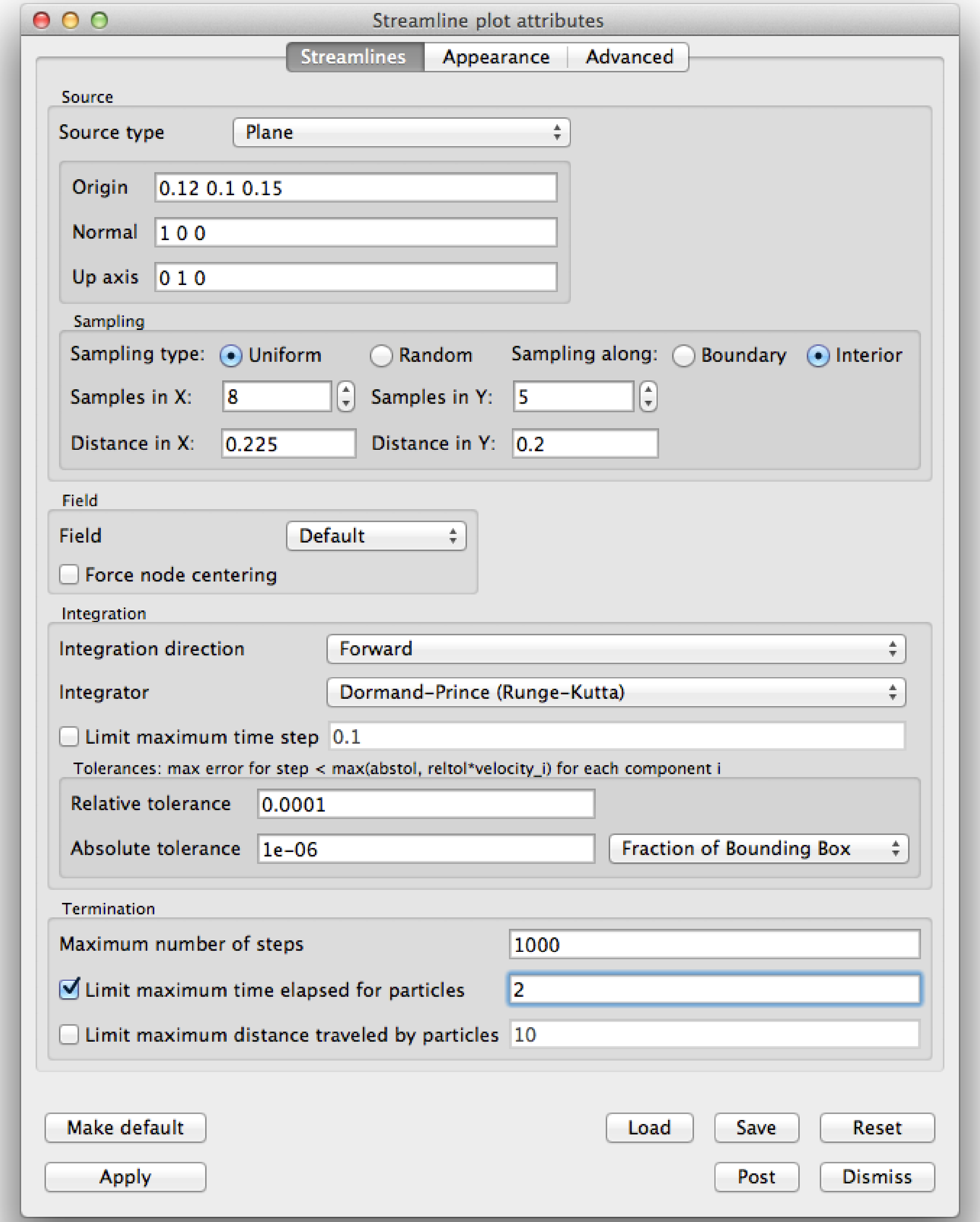
You can also exaggerate the length by increasing the termination time -- however recall there are only 0.025 seconds between each timestep. To be a better visualize of how seed points travel though the vector field as it evolves over time, we will use a pathline technique.

Using Pathlines to Understand Time Varying Flow

Pathlines extend the advection / integral curve concept of streamlines to a time varying vector field. Using pathlines we can trace the paths massless particles would take from the start to the end of the simulation, using the velocity vector field data from all output files.

To do so, we first adjust to select seed points that are well embedded in the fluid column at the beginning of the simulation.

- Start with the **Streamline** plot previously outlined (you can use your **dbreak3d_plot_streamlines.session** or the **dbreak3d_plot_streamlines.py** python script)
- Set the timer slider to time step = 0
- Open the **Streamline** attributes
  - Under the 1st tab (Streamlines)
    - Create a Plane source at
      - Origin: 0.12 0.1 0.15
      - Distance in X: 0.225
      - Distance in Y: 0.2
    - For termination:
      - Set **Limit maximum time elapsed for particles** to 2
  - In the 3rd tab (Advanced)
    - Select **Pathline**
    - Select **Mesh is static over time**
- Click Apply and Dismiss



Source

Source type: Plane

Origin: 0.12 0.1 0.15

Normal: 1 0 0

Up axis: 0 1 0

Sampling

Sampling type: ● Uniform ○ Random

Sampling along: ○ Boundary ● Interior

Samples in X: 8

Samples in Y: 5

Distance in X: 0.225

Distance in Y: 0.2

Field

Field: Default

☐ Force node centering

Integration

Integration direction: Forward

Integrator: Dormand-Prince (Runge-Kutta)

☐ Limit maximum time step: 0.1

Tolerances: max error for step < max(abstol, reltol*velocity_i) for each component i

Relative tolerance: 0.0001

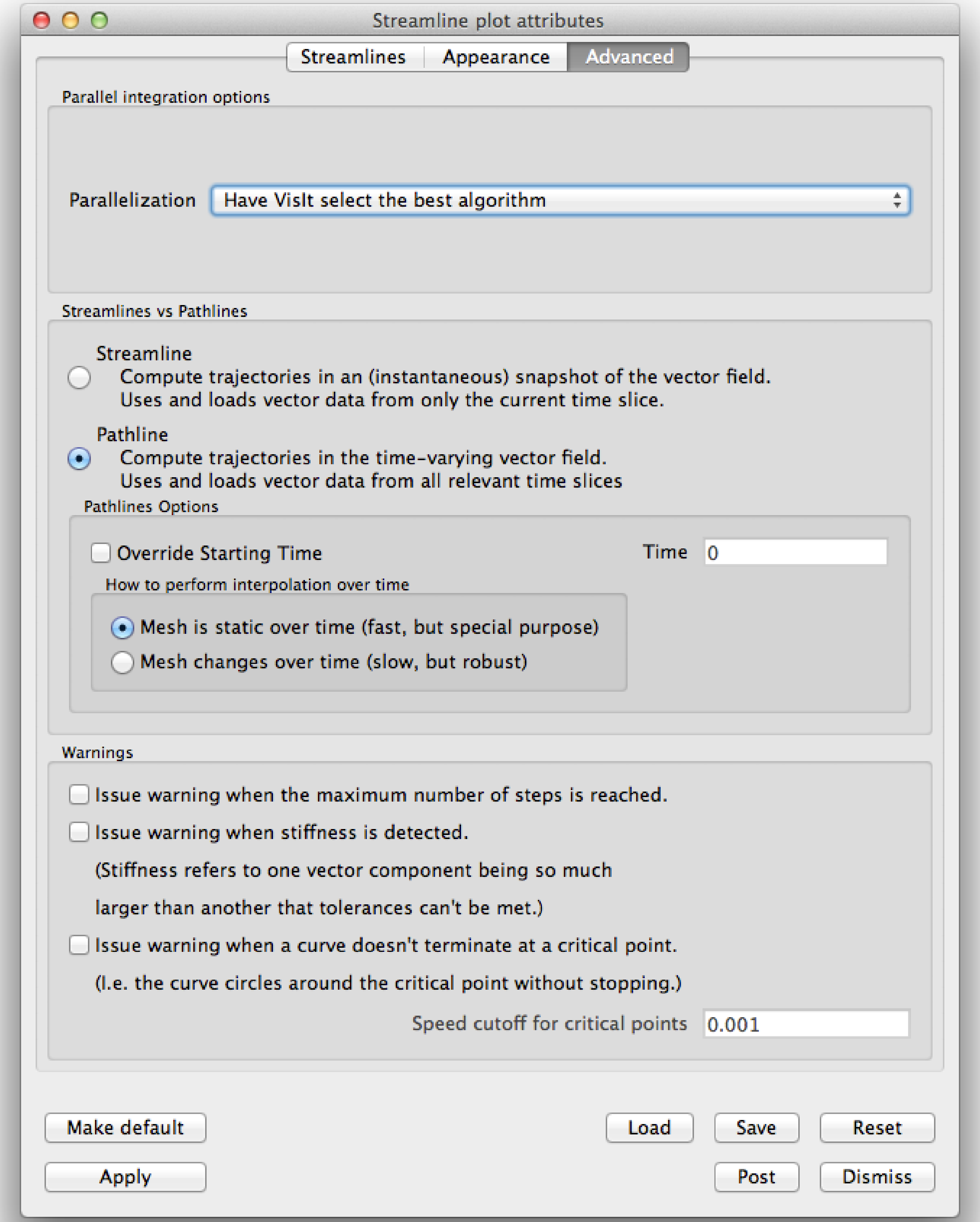
Absolute tolerance: 1e-06 Fraction of Bounding Box

Termination

Maximum number of steps: 1000

☑ Limit maximum time elapsed for particles: 2

☐ Limit maximum distance traveled by particles: 10



Parallel integration options

Parallelization: Have VisIt select the best algorithm

Streamlines vs Pathlines

○ Streamline
Compute trajectories in an (instantaneous) snapshot of the vector field.
Uses and loads vector data from only the current time slice.

● Pathline
Compute trajectories in the time-varying vector field.
Uses and loads vector data from all relevant time slices

Pathlines Options

☐ Override Starting Time

Time: 0

How to perform interpolation over time

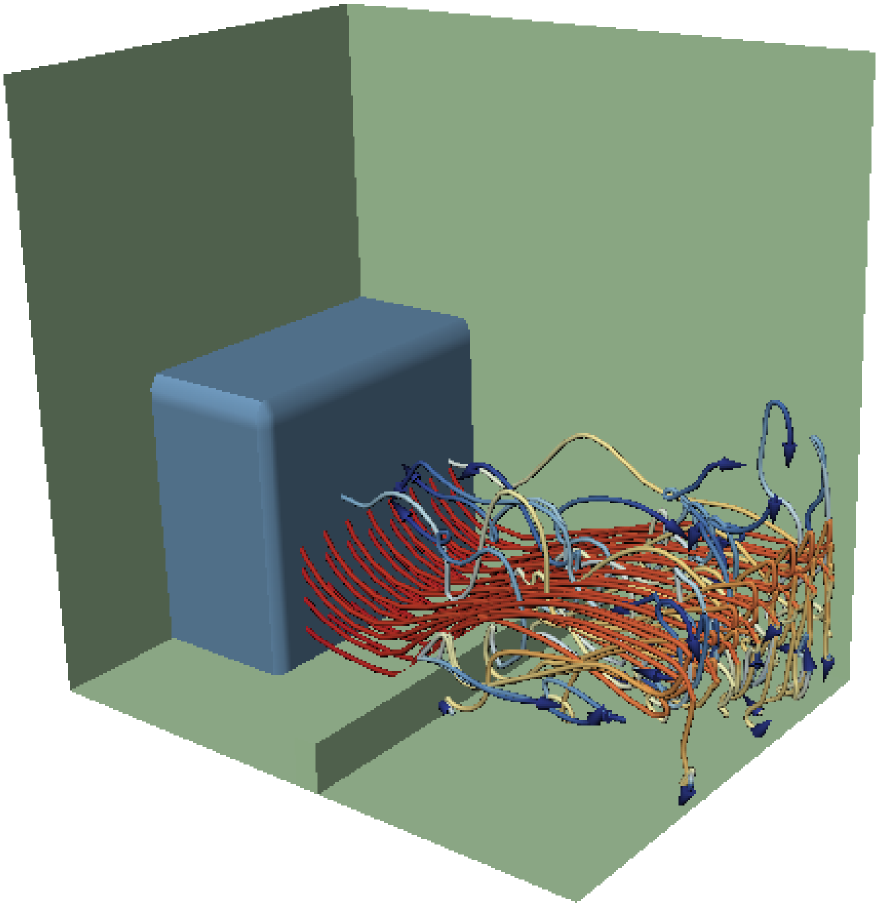
● Mesh is static over time (fast, but special purpose)
○ Mesh changes over time (slow, but robust)

Warnings

☐ Issue warning when the maximum number of steps is reached.
☐ Issue warning when stiffness is detected.
(Stiffness refers to one vector component being so much
larger than another that tolerances can't be met.)
☐ Issue warning when a curve doesn't terminate at a critical point.
(I.e. the curve circles around the critical point without stopping.)

Speed cutoff for critical points: 0.001

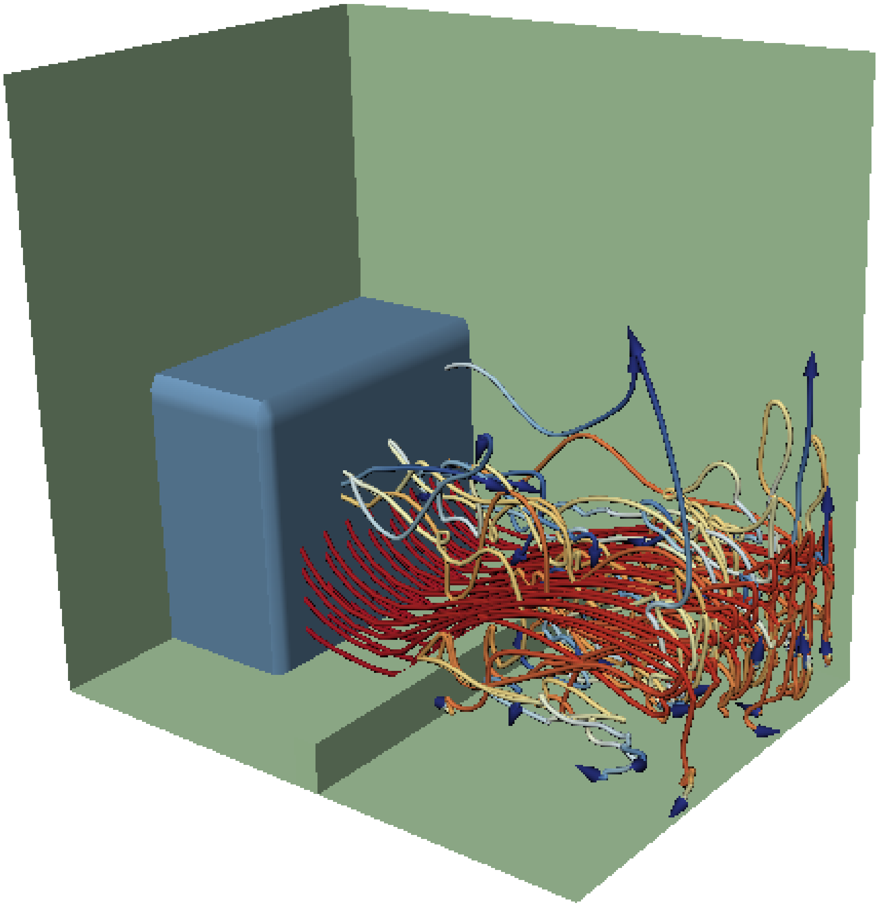
This will create paths that start at timestep=0 at our seed points and travel with the flow for 2 seconds to simulation time. (This will process may take a few seconds because it will process 80 data files)



Pathlines expose the complex flow behavior of the velocity vector field.

Lets extend the termination time to the final time of the simulation.

- Open the **Streamline** attributes
  - Under the 1st tab(Streamlines)
    - For termination:
      - Set *Limit maximum time elapsed for particles* to 4



At this point save your visualization session:

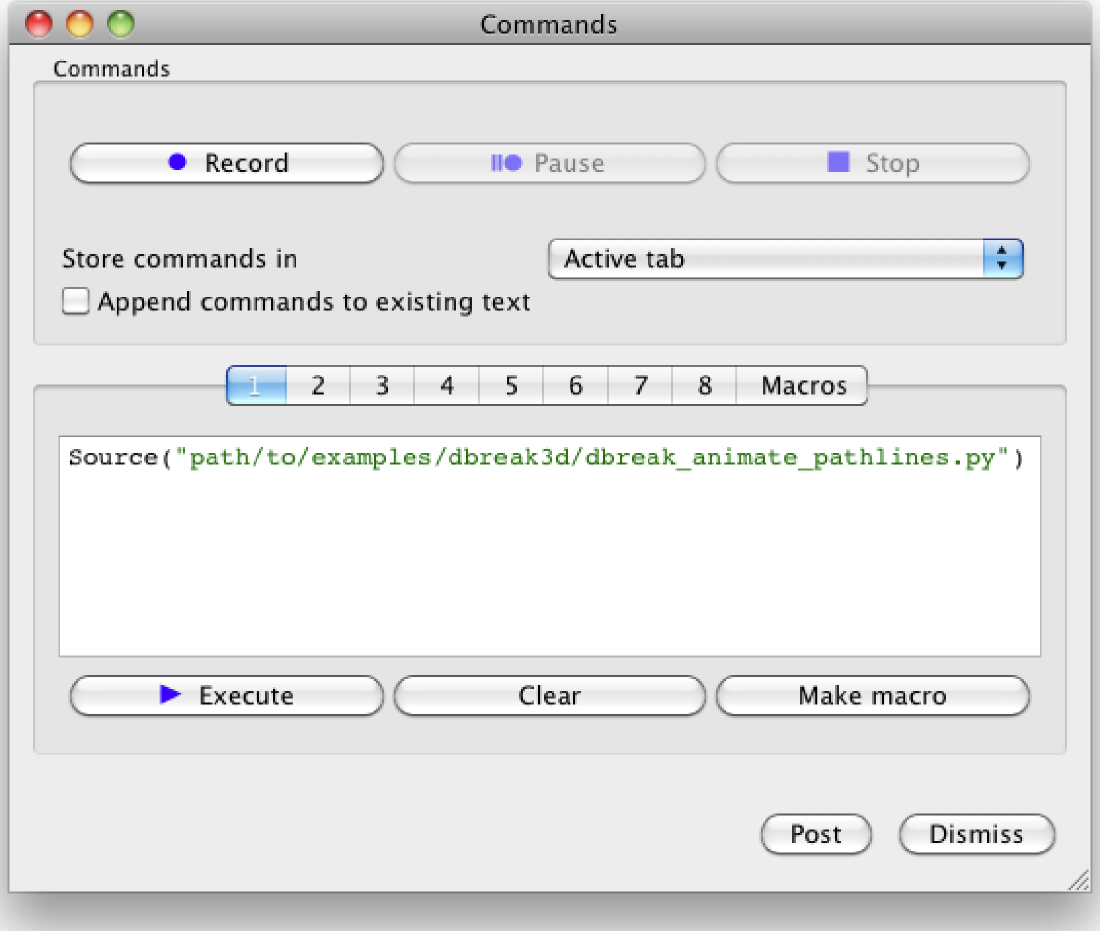
- Save the session (File->Save session as)
    - Set the file name to **dbreak3d_plot_pathlines.session**'

Animating pathlines with a python script

The **Streamline** plot allows you to crop away portions of paths without recomputing the streamlines or pathlines to help support integral curve animations. You can use the **dbreak3d_pathlines_animate.py** to animate our pathline visualization from 0 to 4 seconds.

```
Source("path/to/examples/dbreak3d/dbreak_animate_pathlines.py")
```

You can easily execute this script using the **Commands** Window (Controls->Command)

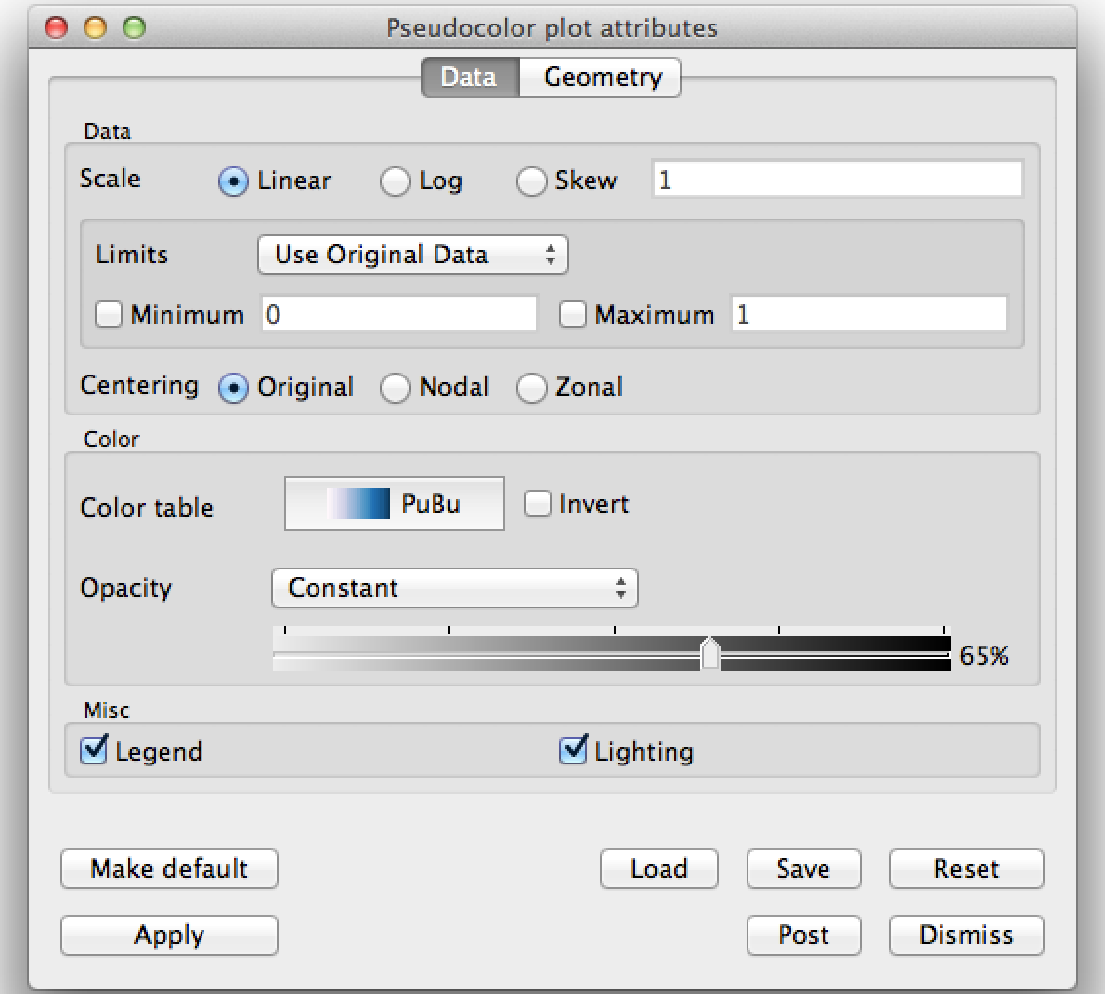


```
Source("path/to/examples/dbreak3d/dbreak_animate_pathlines.py")
```

Rendering a Movie of the Fluid Data

To finish this in-depth tutorial we will create a a simple movie from the water flow simulation data. The movie will display the movement of the water sub-volume and leverage a few more bells and whistles to create a polished visualization.

- Start with the basic tank boundary and water isovolume plot setup (you can restore your dbreak3d_plot_basic.session file or use the dbreak3d_plot_basic.py python script)
- Open the **Pseudocolor** plot attributes
- Under Color, for **Opacity** select **Constant**
- For **Opacity** select **Constant**
- Set the Opacity slider value to 65%
- Click Apply and Dismiss



Pseudocolor plot attributes

Data

Scale ● Linear ○ Log ○ Skew 1

Limits Use Original Data

☐ Minimum 0 ☐ Maximum 1

Centering ● Original ○ Nodal ○ Zonal

Color

Color table PuBu ☐ Invert

Opacity Constant

65%

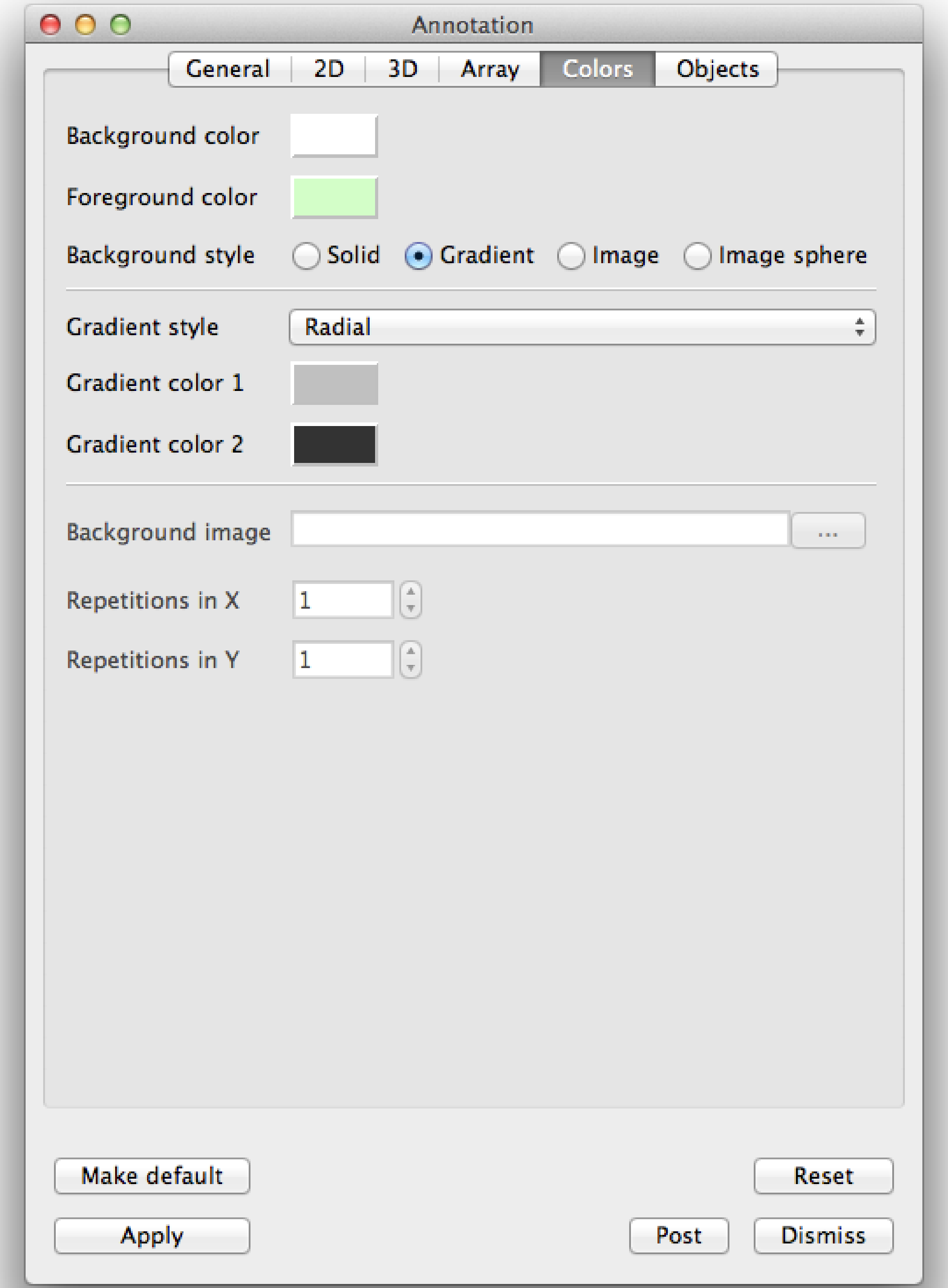
Misc

☑ Legend ☑ Lighting

Setting the Window Annotations

To make the visualization look more polished, we will change the window annotations and background, lighting, and add a time slider annotation:

- Open the *Annotations* Window (Controls->Annotation)
- On the 1st Yab:
  - Click **No Annotations**
  - Click Apply
- On the 3D Tab:
  - Recheck **Show bounding box**
- On the Color Tab:
  - Set the **Foreground color** to be the same color as our tank boundaries plot.
  - Select a **Gradient** background with the **Radial** style.
  - Set **Gradient color 1** to be light gray
  - Set **Gradient color 2** to be very dark gray



Background color

Foreground color

Background style ○ Solid ● Gradient ○ Image ○ Image sphere

Gradient style: Radial

Gradient color 1

Gradient color 2

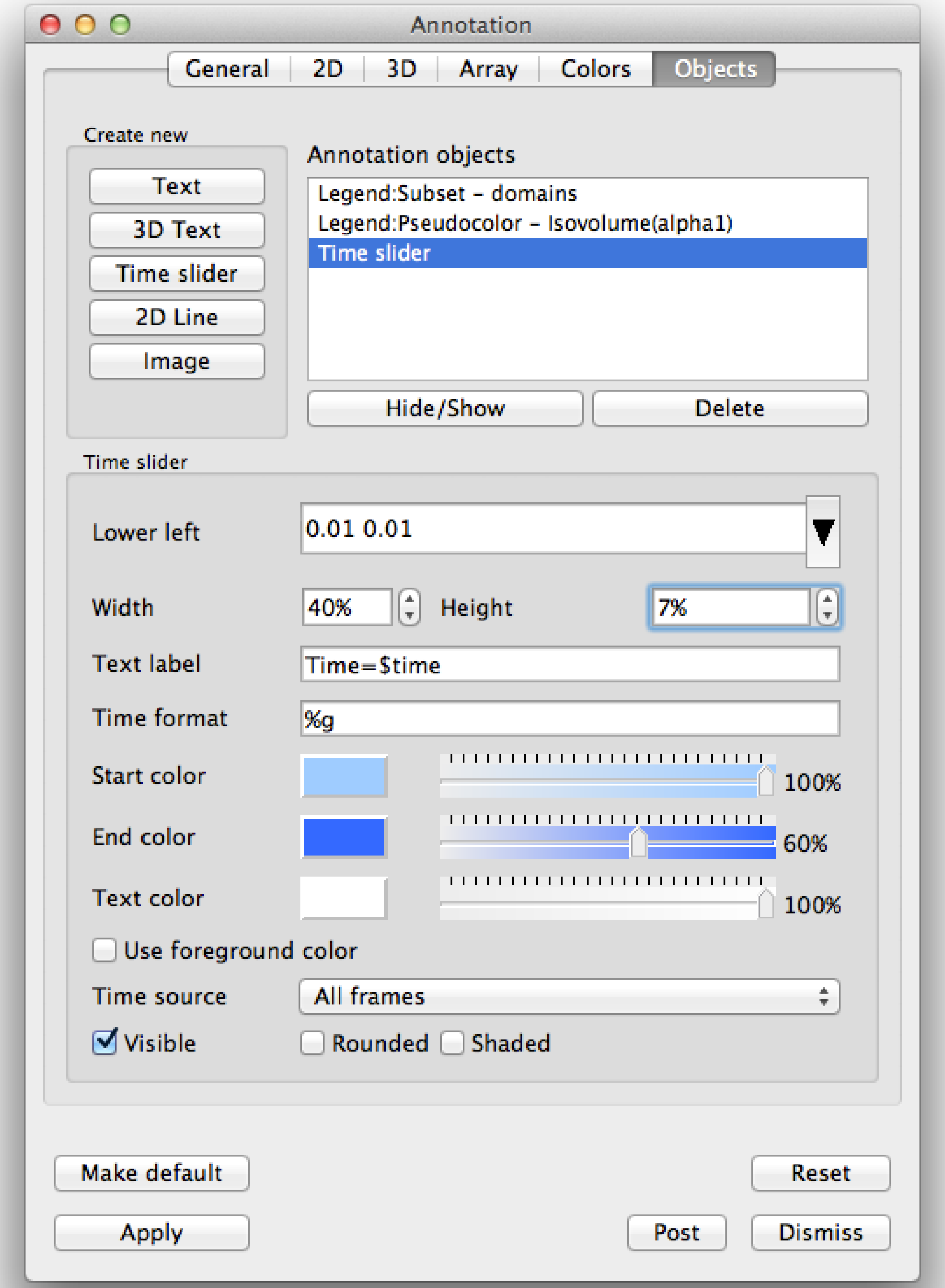
Background image

Repetitions in X: 1

Repetitions in Y: 1

- On the Objects Tab:
  - Create a new **Time slider** annotation.
  - For the name dialog type **ts** and click Ok
    - Set the **Width** to 40%
    - Set the **Height** to 7%
    - Set the **Start color** to Light Blue
    - Set the **End color** to a darker Blue
    - Set the **Text color** to White

    - Uncheck **Rounded** and **Shaded**



Annotation objects
Legend:Subset - domains
Legend:Pseudocolor - Isovolume(alpha1)
Time slider

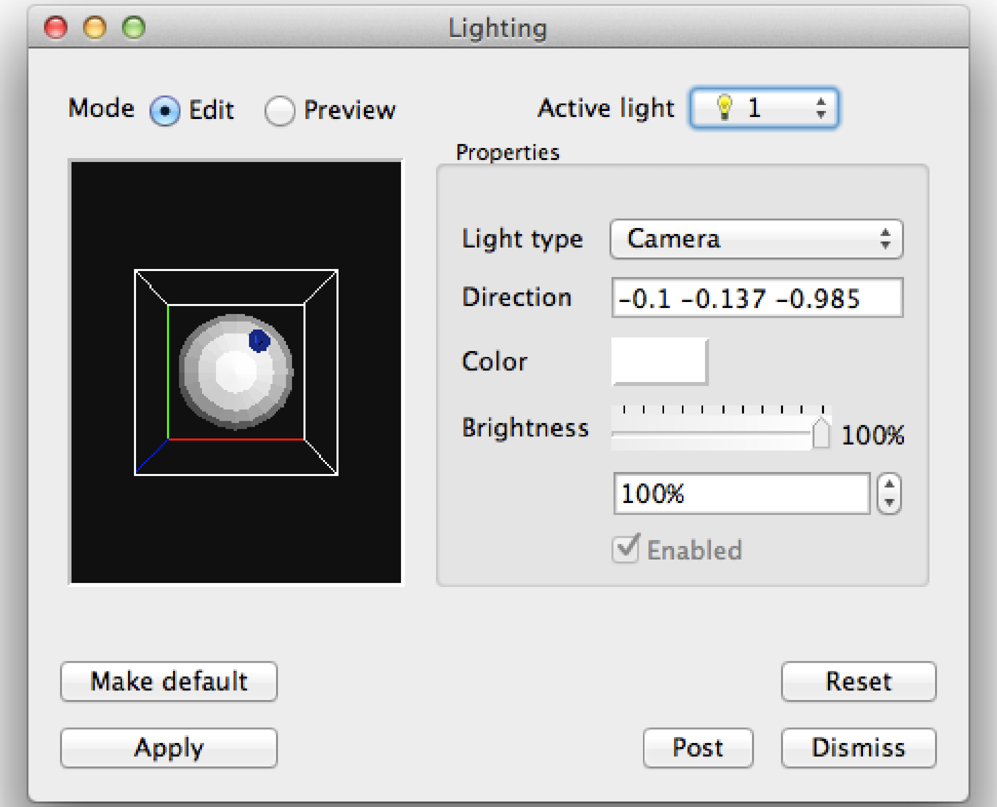
Time slider
Lower left 0.01 0.01
Width 40%
Height 7%
Text label Time=$time
Time format %g
Start color 100%
End color 60%
Text color 100%
☐ Use foreground color
Time source All frames
☑ Visible
☐ Rounded
☐ Shaded

The default light shines straight on to the scene, which is not as intuitive for humans who are primarily used to overhead lighting.

- Open the **Lighting** Window (Controls->Lighting)
  - Move the light vector up and to the right.
  - Click Apply

Try a few different settings until you find a setting you are satisfied with.

- Click Dismiss



Mode ● Edit ○ Preview

Active light 1

Light type: Camera

Direction: -0.1 -0.137 -0.985

Brightness: 100%

100%

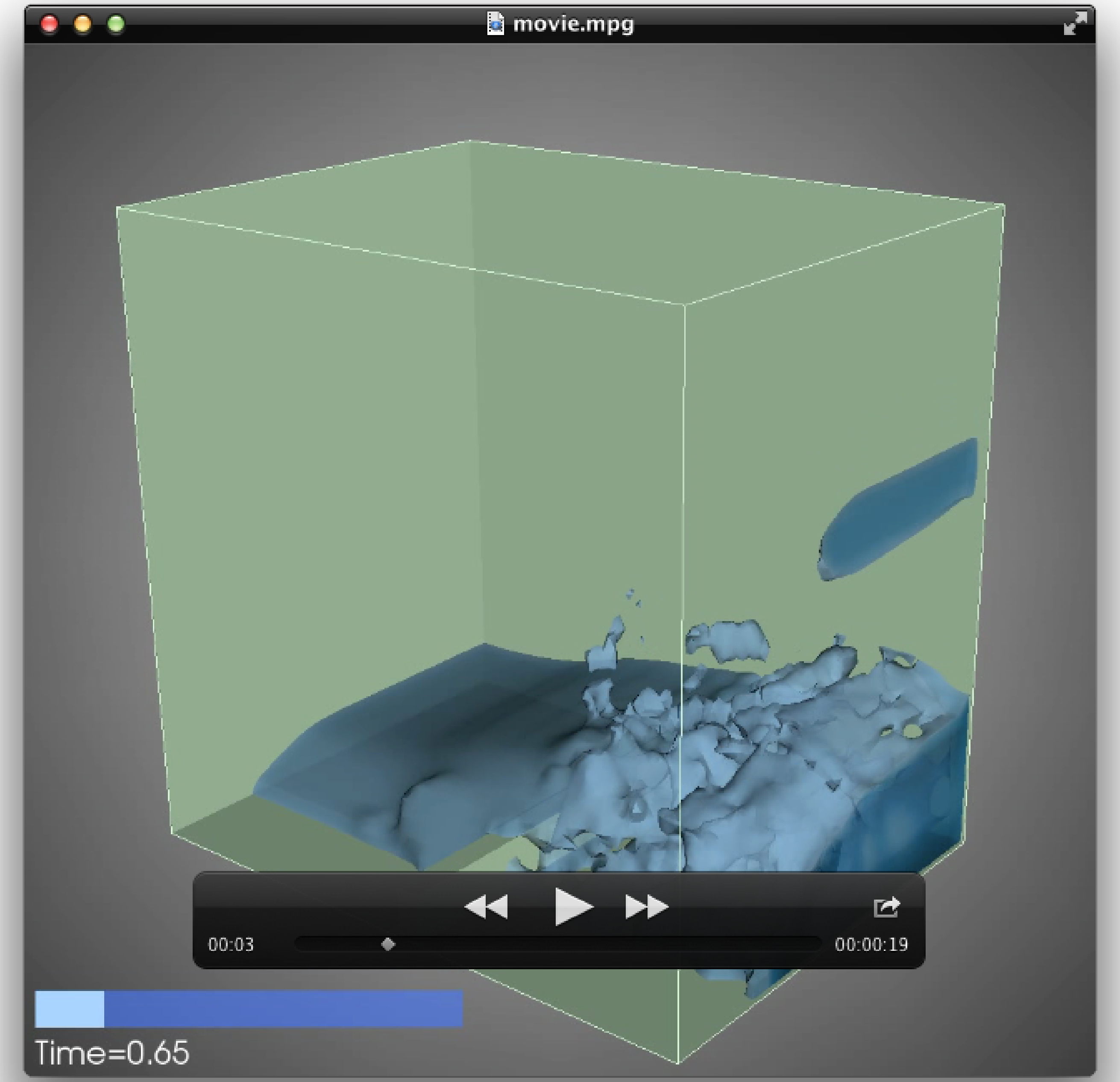
☑ Enabled

Encoding the movie

Now we are ready to create an mpeg of the visualization.

- Open the **Save Movie** wizard (File->Save movie)
- Follow the prompts to create a MPEG movie with a 1000x1000 output resolution.

This will render all of the timesteps of the simulation data and create an mpeg in your home directory. You can also save a sequence of images in addition to encoding to a movie format.



The end!

Retrieved from "http://visitusers.org/index.php?title=Water_Flow_Tutorial"

- This page was last modified 22:14, 22 November 2013.