TWO-STEP GUIDE for creating 3D radar reflectivity isosurface using VAPOR

Author: ©Hongpei Yang 2025, Sun Yat-sen University
Email: yanghp731@gmail.com; yanghp9@mail2.sysu.edu.cn

Software:

VAPOR (Visualization and Analysis Platform for Ocean, Atmosphere, and Solar Researchers)

Platform:

macOS (Apple Silicon, M4 chip)

VAPOR webpage and documentation:

https://www.vapor.ucar.edu/

Required data:

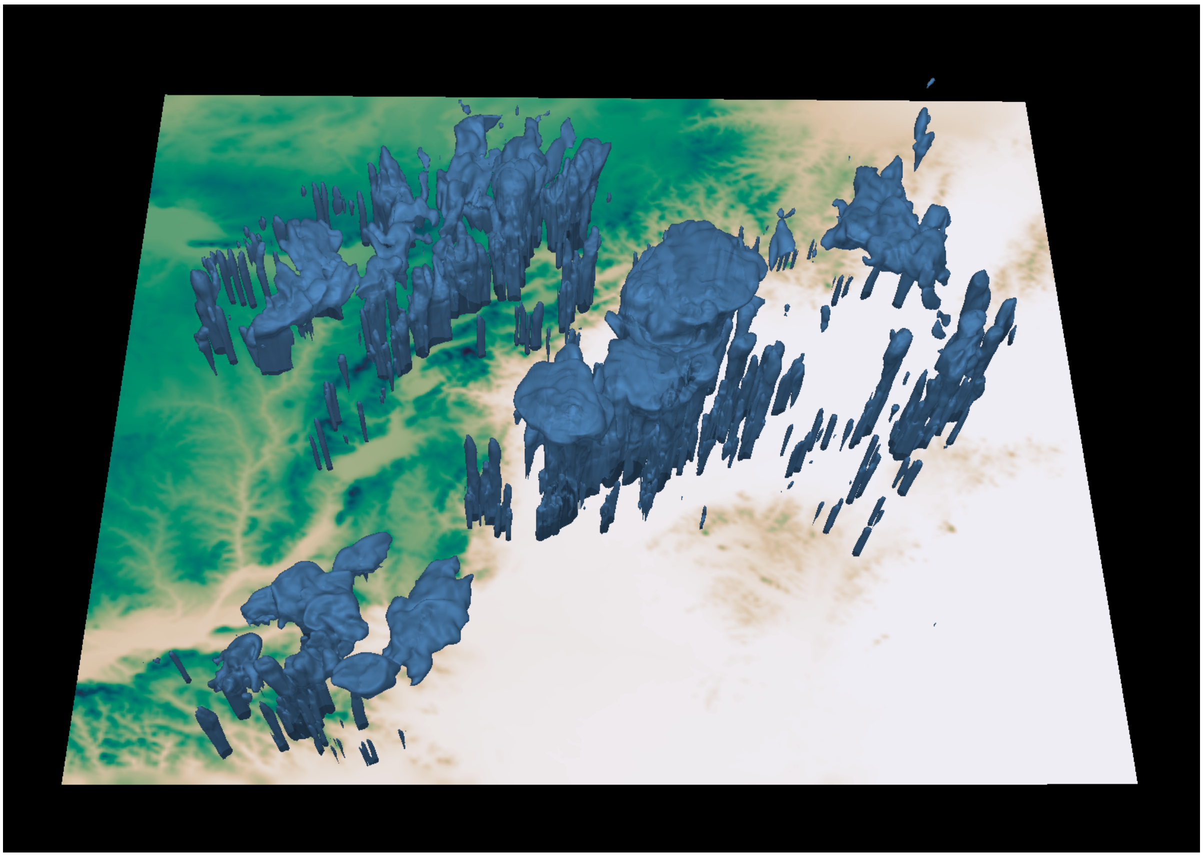
WRF-ARW model output

Preprocessing is performed using Python with the wrf-python package to extract and reformat variables (e.g., reflectivity and topography) into a VAPOR-compatible NetCDF file.

Installing wrf-python on Apple Silicon (M-series chips) may require special handling. A helpful workaround and detailed discussion can be found here:

https://github.com/NCAR/wrf-python/issues/199#issuecomment-1949394544

Sample:



Caption: 3D radar reflectivity of severe storms over North China.

Step 1: Exporting 3D Reflectivity (**REF**) and **TOPOgraphy** from WRF to a VAPOR-Compatible NetCDF File

Input: WRF output NetCDF file containing at least one time step (e.g., wrfout_d02_2024-08-04_0000)
Output: A VAPOR-compatible NetCDF file containing 3D reflectivity (REF / DBZ) and 2D TOPOgraphy (e.g., topo_dbz3d_vapor_ready.nc)

Python Code:

The output file is prepared following VAPOR requirements, including:

- True geographic coordinates (latitude/longitude)
- Physical vertical coordinate
- 3D reflectivity field without time dimension

```
### import packages
import numpy as np
from netCDF4 import Dataset
import wrf

### read wrf output file
# Open the NetCDF file
ncfile1   =   Dataset('/Volumes/T7-yanghp/proj_ongoing/downhill_CI/WRF_data/wrfout_d02_2024-08-04_0000_ctrl')
# Get the WRF variables
z = wrf.getvar(ncfile1,"z",0)
it = 22 # set the time
dbz = wrf.getvar(ncfile1,"dbz",it) # radar reflectivity
xlat = wrf.getvar(ncfile1,"XLAT",it)
xlon = wrf.getvar(ncfile1,"XLONG",it)
topo = wrf.getvar(ncfile1,"ter",it)
znu    = wrf.getvar(ncfile1,"ZNU",it)

dbz_3d = np.array(dbz)      # (nz, ny, nx)
topo2d = np.array(topo) # (ny, nx)
lat2d    = np.array(xlat)    # (ny, nx)
lon2d    = np.array(xlon)    # (ny, nx)
znu1d    = np.array(znu)      # nz

nz, ny, nx = dbz_3d.shape

### Create and open NetCDF file
out_nc = Dataset('topo_dbz3d_vapor_ready.nc', 'w', format='NETCDF4')

out_nc.createDimension('z', nz)
out_nc.createDimension('y', ny)
out_nc.createDimension('x', nx)
```

```
z_var = out_nc.createVariable('z', 'f4', ('z',))
y_var = out_nc.createVariable('y', 'f4', ('y',))
x_var = out_nc.createVariable('x', 'f4', ('x',))

z_var[:] = znu1d.astype('f4')          # physical layers
y_var[:] = lat2d[:,0].astype('f4')
x_var[:] = lon2d[0,:].astype('f4')

z_var.units = 'm'
y_var.units = 'degrees_north'
x_var.units = 'degrees_east'
z_var.positive = 'up'

### write variables
dbz_var = out_nc.createVariable('DBZ', 'f4', ('z','y','x'), fill_value=np.nan)
dbz_var[:] = dbz_3d.astype('f4')
dbz_var.units = 'dBZ'
dbz_var.long_name = 'Radar reflectivity'
dbz_var.valid_range = np.array([-10.0, 70.0], dtype='f4')

topo_var = out_nc.createVariable('TOPO', 'f4', ('y','x'))
topo_var[:] = topo2d.astype('f4')
topo_var.units = 'm'

lat_var = out_nc.createVariable('XLAT', 'f4', ('y','x'))
lon_var = out_nc.createVariable('XLONG','f4', ('y','x'))
lat_var[:] = lat2d.astype('f4')
lon_var[:] = lon2d.astype('f4')
lat_var.units = 'degrees_north'
lon_var.units = 'degrees_east'

### Close NetCDF file
out_nc.close()
print("NetCDF file written: topo_dbz3d_vapor_ready.nc")
```

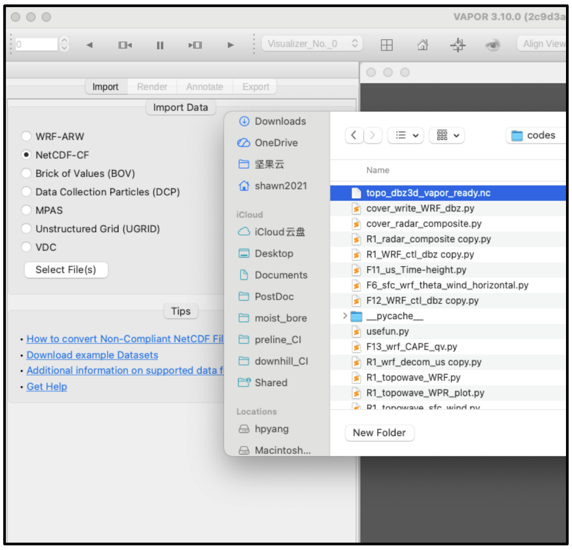
This python code file can also be downloaded from:
https://github.com/hongpei-yang/met-tools/tree/main

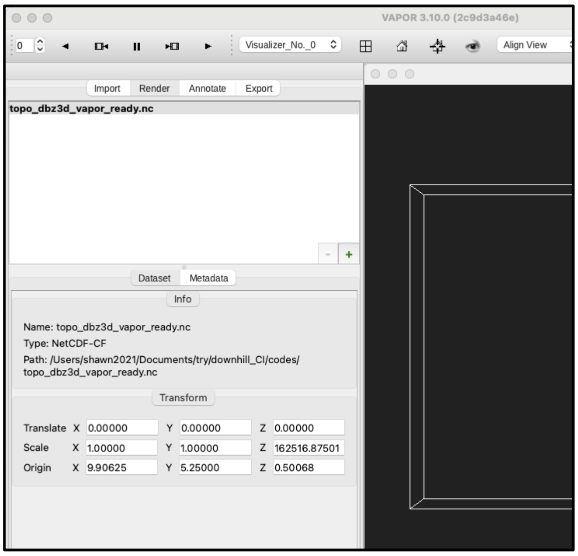
Step 2: Visualizing REF and TOPO using VAPOR

VAPOR version: 3.10.0

a) Importing the data

1. Launch **VAPOR**.
2. Select **NetCDF-CF** → **Select Files**.
3. Choose the NetCDF file prepared in Step 1.
4. Click **Open**.
5. Click on the loaded file in the **Render** panel to inspect the metadata.



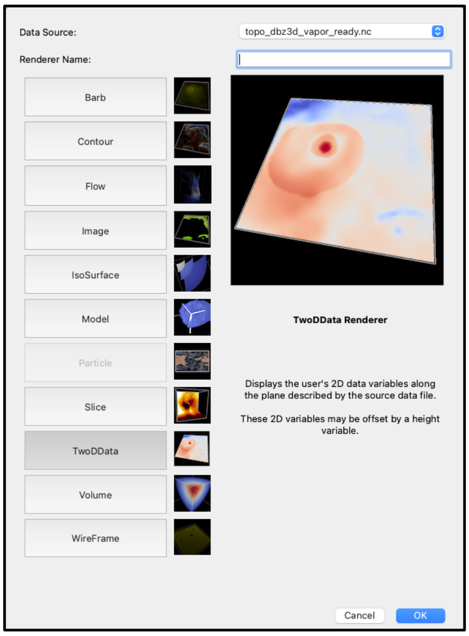


Name: topo_dbz3d_vapor_ready.nc
Type: NetCDF-CF
Path: /Users/shawn2021/Documents/try/downhill_CI/codes/topo_dbz3d_vapor_ready.nc

Translate	X	0.00000	Y	0.00000	Z	0.00000
Scale	X	1.00000	Y	1.00000	Z	162516.87501
Origin	X	9.90625	Y	5.25000	Z	0.50068

b) Visualizing 2D Topography (TOPO)

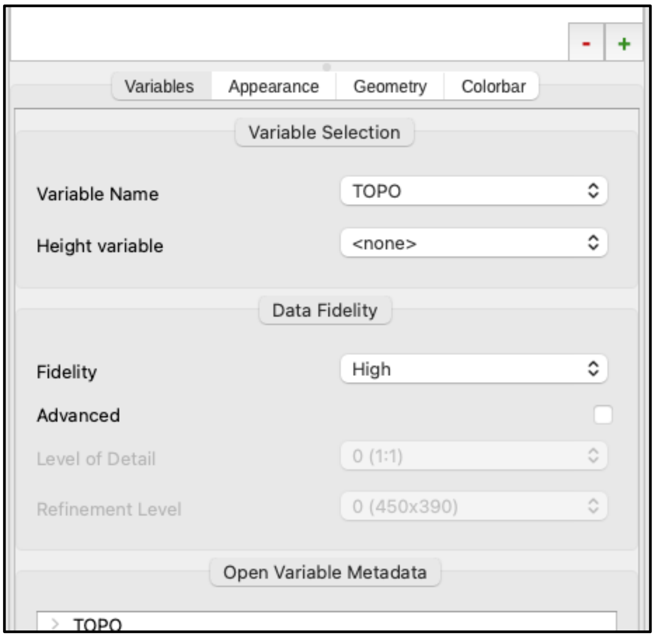
1. In the **Render** section, click **"+"** and select **TwoDData**, then click **OK**.
2. In the **Appearance** section:
   - Select a colormap of your choice.
   - Adjust the data range to enhance terrain contrast.



Data Source: topo_dbz3d_vapor_ready.nc
Renderer Name:

Displays the user's 2D data variables along the plane described by the source data file.

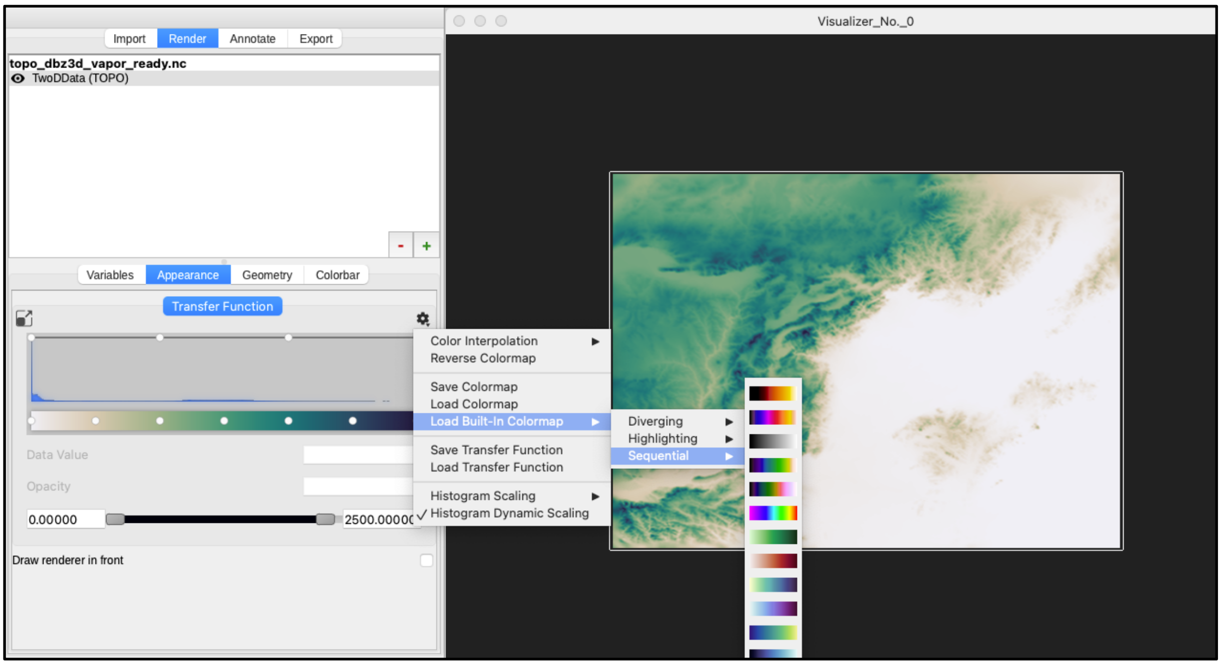
These 2D variables may be offset by a height variable.



Variable Name: TOPO
Height variable: <none>
Fidelity: High
Advanced □
Level of Detail: 0 (1:1)
Refinement Level: 0 (450x390)

3. Click the **eye icon** in the **Render** panel to display the topography.

Once the inspection is complete, it is recommended to click the **eye icon again to hide the TOPO layer**, especially before rendering 3D variables, as visualization can consume additional memory.

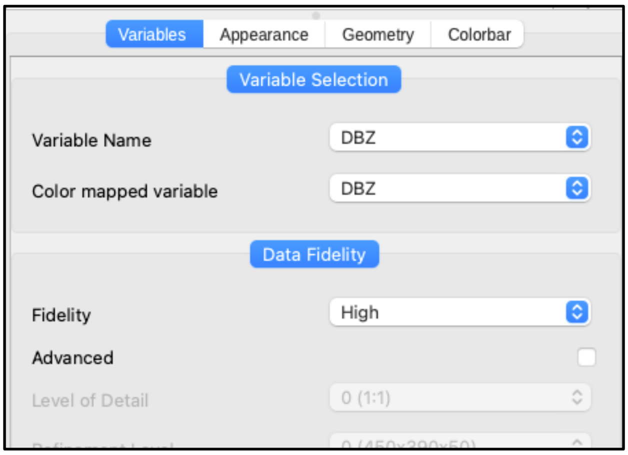


0.00000 — 2500.00000

c) Visualizing 3D Reflectivity (REF)

1. Return to the **Render** section.
2. Click "**+**" and select **IsoSurface**, then click **OK**.

3. Make sure **DBZ (REF)** is selected as the variable and that it is color-mapped by itself.
4. Adjust the data range (e.g., **0–55 dBZ** in this case).
5. Choose **Iso Regular** as the **Raytracing Algorithm**.



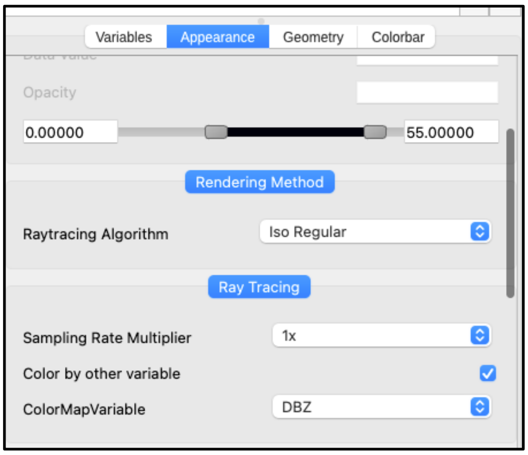
Variable Name: DBZ

Color mapped variable: DBZ

Fidelity: High

Advanced □

Level of Detail: 0 (1:1)



0.00000 — 55.00000

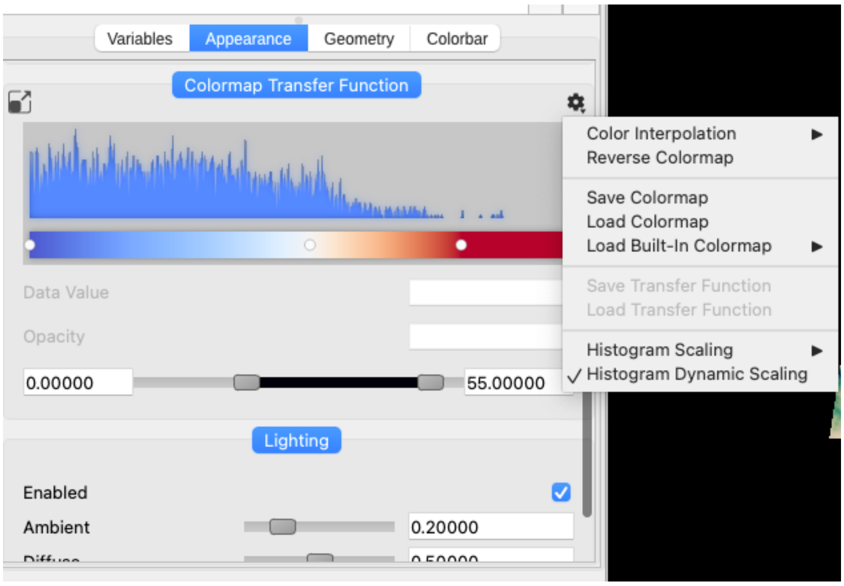
Raytracing Algorithm: Iso Regular

Sampling Rate Multiplier: 1x

Color by other variable ☑

ColorMapVariable: DBZ

6. In the **Colormap Transfer Function**:
    - Select a suitable colormap.
    - Ensure the colormap range matches the data range defined in the **Transfer Function**.
7. Optionally enable and adjust lighting effects in the **Lighting** section to enhance 3D perception.



0.00000 – 55.00000

Enabled ☑

Ambient 0.20000

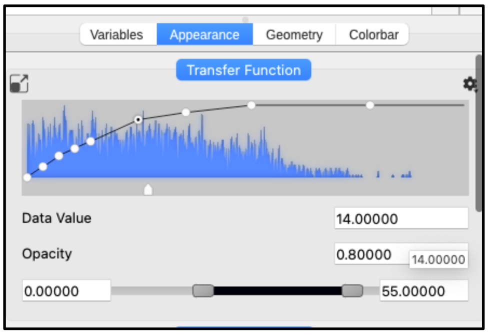
Diffuse 0.50000

Opacity adjustment (critical step)

The most important—and often most time-consuming—step is adjusting **Opacity** for each **Data Value** to achieve a see-through effect.

For example, lower reflectivity values can be set to higher transparency so that intense convective cores with higher reflectivity remain visible inside the storm.

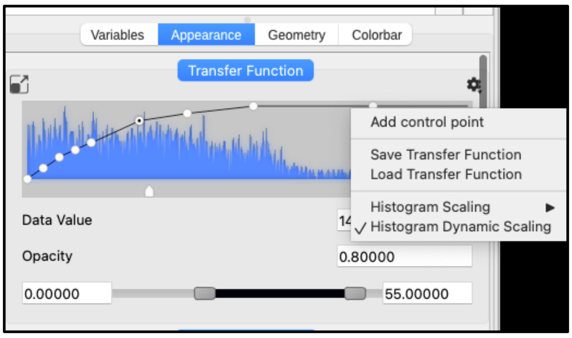
- You can add opacity control points by clicking on the black opacity curve and selecting **“Add control point”**.
- Additional refinements can be made by adjusting control points in the **Colormap Transfer Function** to fine-tune color distribution.



Data Value 14.00000

Opacity 0.80000

0.00000 – 55.00000



Data Value 14

Opacity 0.80000

0.00000 – 55.00000

⚠ Performance tip:

During opacity tuning, it is recommended to temporarily disable real-time rendering by clicking the **eye icon** in the **Render** panel, as 3D rendering is memory-intensive and may cause the software to shut down.

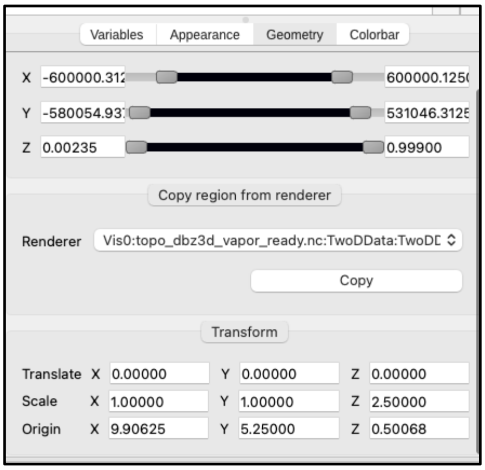
d) Adjusting plotting range, axis scaling, and focus region

These adjustments are mainly performed in the **Geometry** section:

- You may directly modify the data range along each axis.
- Alternatively, drag the red bounding box in the visualization window (similar to image cropping) to focus on the region of interest.
- To enhance vertical structure, you can increase the **Z-axis scale** (e.g., 2.5 in this example).
- The plot center can also be repositioned as needed.

To rotate the 3D scene, drag the mouse upward in the visualization window.

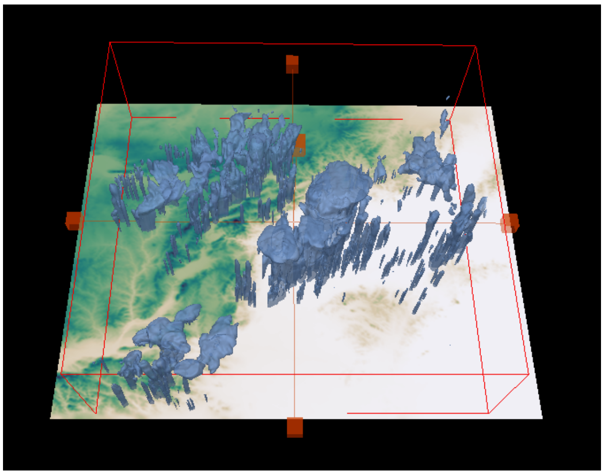
To return to a default planar view, click the home or eye icon at the top of the window.



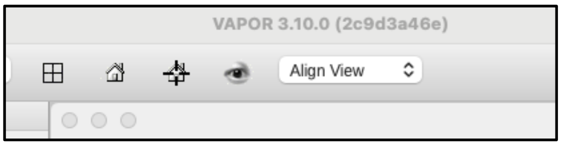
X -600000.312 … 600000.125
Y -580054.93 … 531046.3125
Z 0.00235 … 0.99900

Renderer Vis0:topo_dbz3d_vapor_ready.nc:TwoDData:TwoDD

Translate	X	0.00000	Y	0.00000	Z	0.00000
Scale	X	1.00000	Y	1.00000	Z	2.50000
Origin	X	9.90625	Y	5.25000	Z	0.50068



To return to a default planar view, click the home or eye icon at the top of the window.

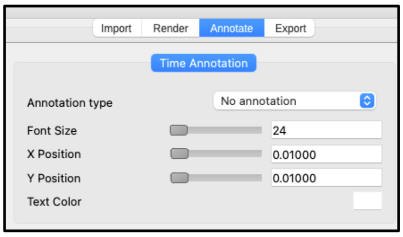


e) Additional customization

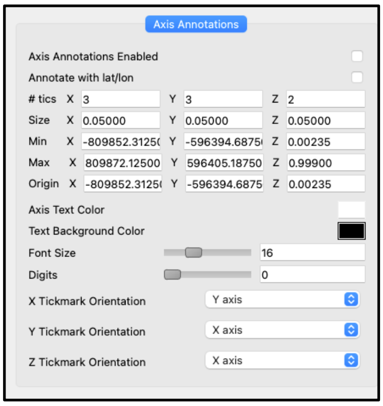
In the **Annotate** section, you may:

- Add text annotations
- Modify axis colors and orientations
- Change background and frame colors

These options help tailor the visualization for presentations or publications.



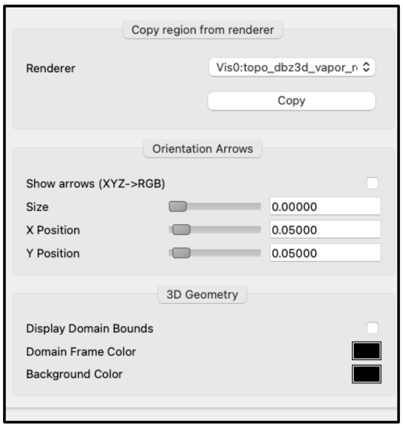
Annotation type: No annotation
Font Size: 24
X Position: 0.01000
Y Position: 0.01000
Text Color



Axis Annotations Enabled □
Annotate with lat/lon □

# tics	X	3	Y	3	Z	2
Size	X	0.05000	Y	0.05000	Z	0.05000
Min	X	-809852.3125(	Y	-596394.6875(	Z	0.00235
Max	X	809872.12500	Y	596405.18750	Z	0.99900
Origin	X	-809852.3125(	Y	-596394.6875	Z	0.00235

Axis Text Color
Text Background Color
Font Size: 16
Digits: 0
X Tickmark Orientation: Y axis
Y Tickmark Orientation: X axis
Z Tickmark Orientation: X axis



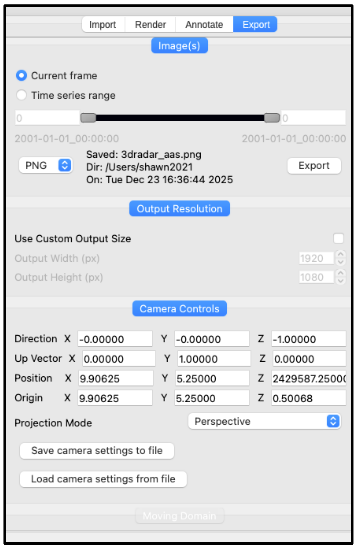
Renderer: Vis0:topo_dbz3d_vapor_r

Show arrows (XYZ->RGB) □
Size: 0.00000
X Position: 0.05000
Y Position: 0.05000

Display Domain Bounds □
Domain Frame Color
Background Color

f) Exporting figures

Figures can be exported using the **Export** section, where you may specify output preferences. Alternatively, screenshots can be taken directly from the visualization window.



● Current frame
○ Time series range

0 — 0

2001-01-01_00:00:00 — 2001-01-01_00:00:00

PNG

Saved: 3dradar_aas.png
Dir: /Users/shawn2021
On: Tue Dec 23 16:36:44 2025

Use Custom Output Size □
Output Width (px) 1920
Output Height (px) 1080

Direction	X	-0.00000	Y	-0.00000	Z	-1.00000
Up Vector	X	0.00000	Y	1.00000	Z	0.00000
Position	X	9.90625	Y	5.25000	Z	2429587.2500(
Origin	X	9.90625	Y	5.25000	Z	0.50068

Projection Mode: Perspective

---END--