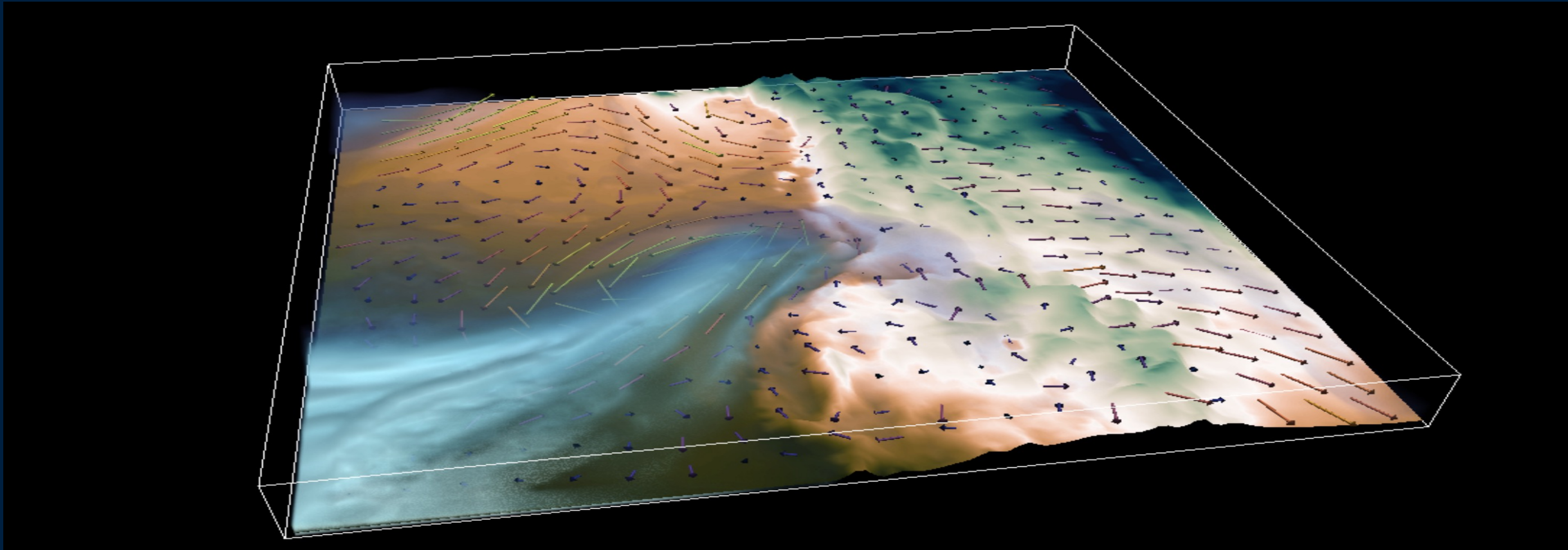
INTERACTIVE VISUALIZATION OF WRF DATA WITH VAPOR 3.5

VAPOR: Data Exploration Tool Designed for Earth System Science

DECEMBER 7, 2021



OVERVIEW

INTRO: VAPOR SOFTWARE

- what is VAPOR?
- what makes it different from other visualization tools?
- examples

HANDS-ON WORKSHOP

- importing data
- twoddata with elevation projection
- wind barbs
- volume rendering

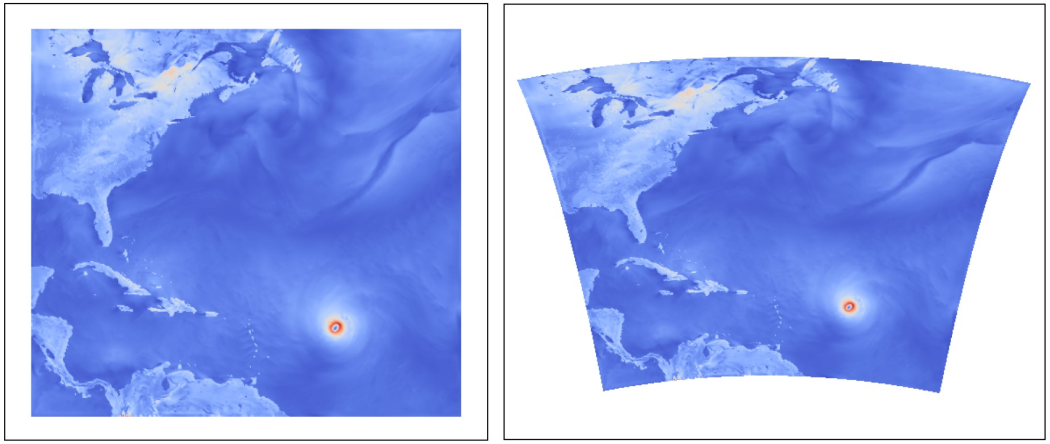
INTRO: VAPOR SOFTWARE

INTRODUCTION:
WHAT IS VAPOR?

- **Stand alone visualization package designed specifically for Earth Systems Science (ESS) developed by NCAR**

- **Focus:**
    - Interactive data exploration

- **Main Challenges:**
    - Large 3D numerical data
    - ESS-specific datasets/geo-referencing



(**a**) Lambert Conformal Conic Projection

(**b**) No Projection

INTRODUCTION:
WHAT MAKES IT SPECIAL?

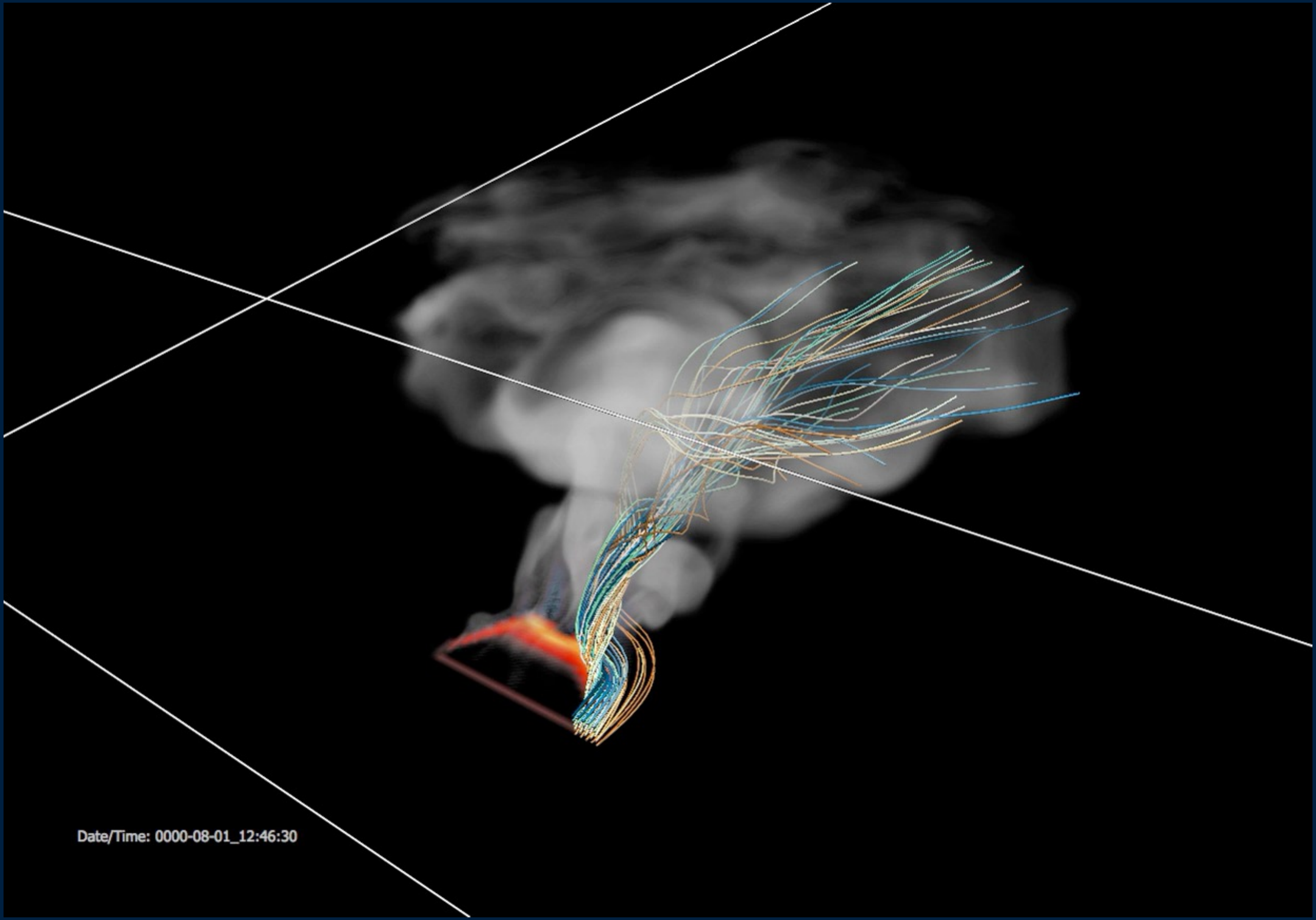
- **How is it different from other tools?**
    - VisIt, ParaView – interactive, but general-purpose, computationally demanding
        - support distributed memory, parallel rendering often performed on a cluster
    - NCL, Python (MetPy, CDAT) – ESS specific, but not interactive, 2D focus
        - challenging for data exploration

- **VAPOR = fills the gap**
    - **allows interactive data exploration on commodity hardware**
    - **addresses issues specific to ESS**
        - georeferencing
        - vertical coordinate systems, common grids (staggered, eta levels)
        - missing data
        - common ESS file formats (NetCDF, MPAS, WRF)

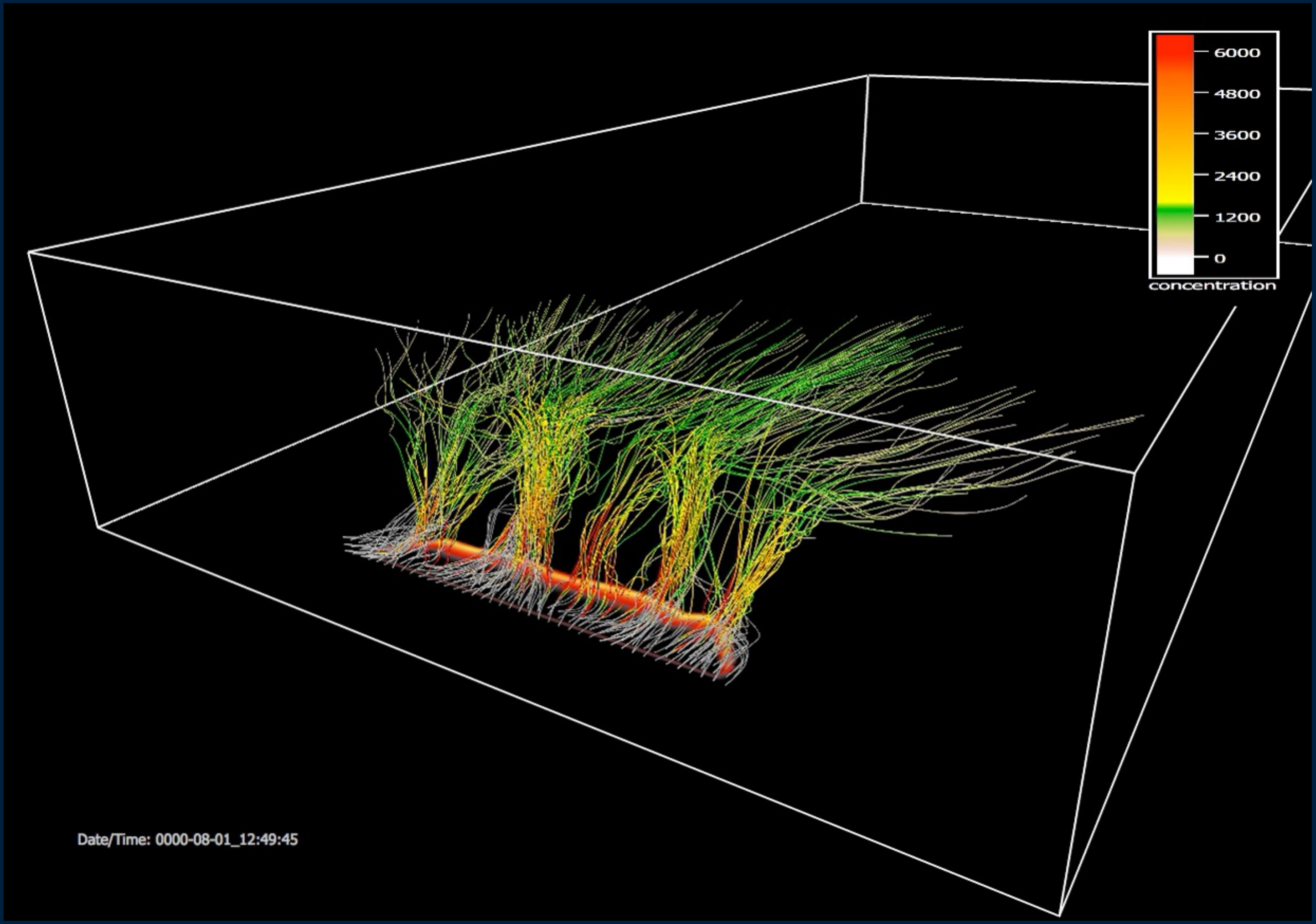
VAPOR DATA COLLECTION (VDC)

- **What makes VAPOR fast?**
  - **progressive data access**
    - **multi-resolution**
      - eg. GoogleMaps – data hierarchy
    - **lossy compression**
      - order data based on its "information content" and use only the important bits
      - eg. JPEG, streaming
  - both possible using **discrete wavelet transform**
    - transform data into frequency space and store coefficients only (VDC format)
    - combine both controls above into a single **fidelity control**
  - **THIS IS OPTIONAL!**

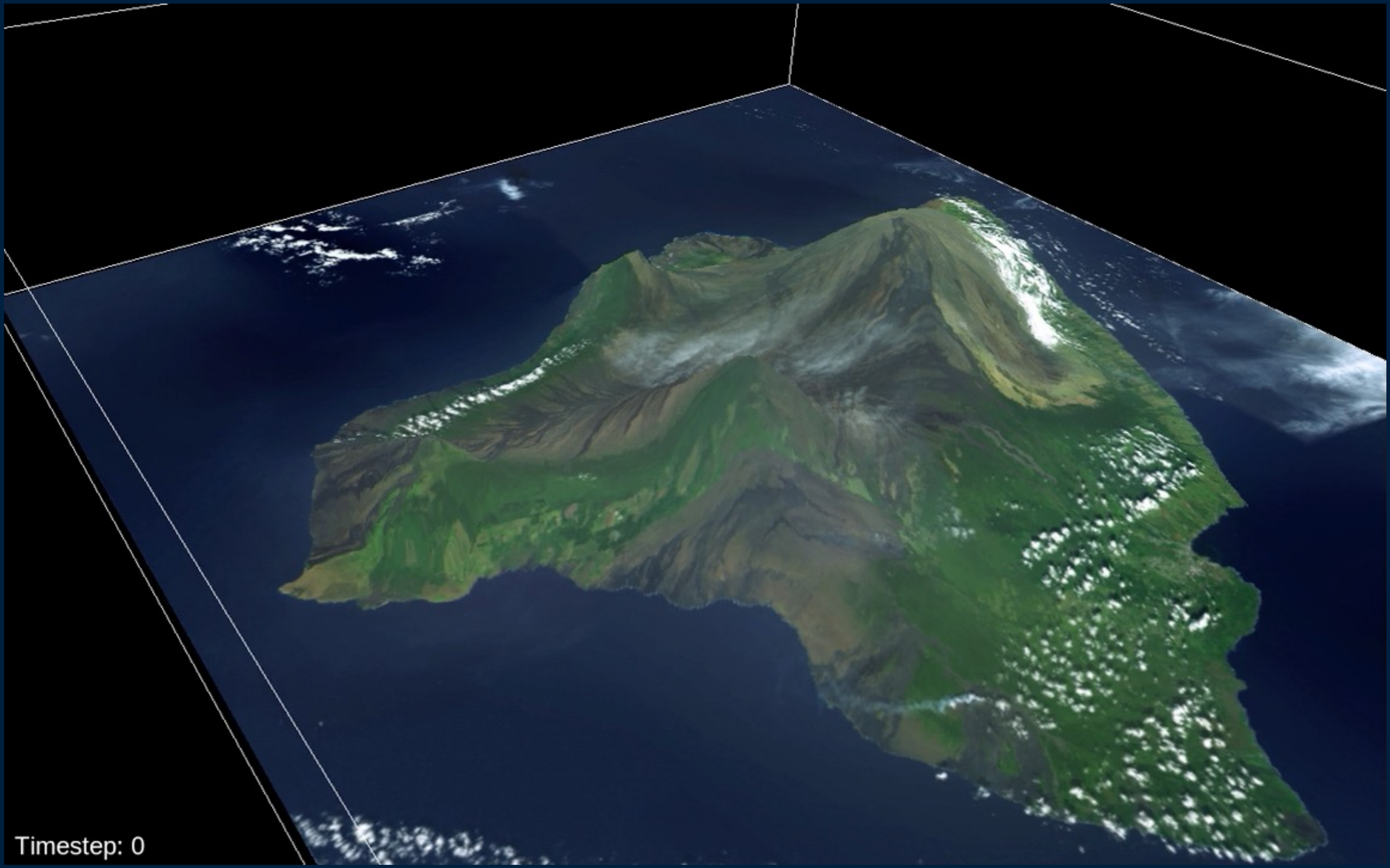
EXAMPLES: WILDFIRE SMOKE AND TURBULENCE



EXAMPLES: WILDFIRE SMOKE AND TURBULENCE



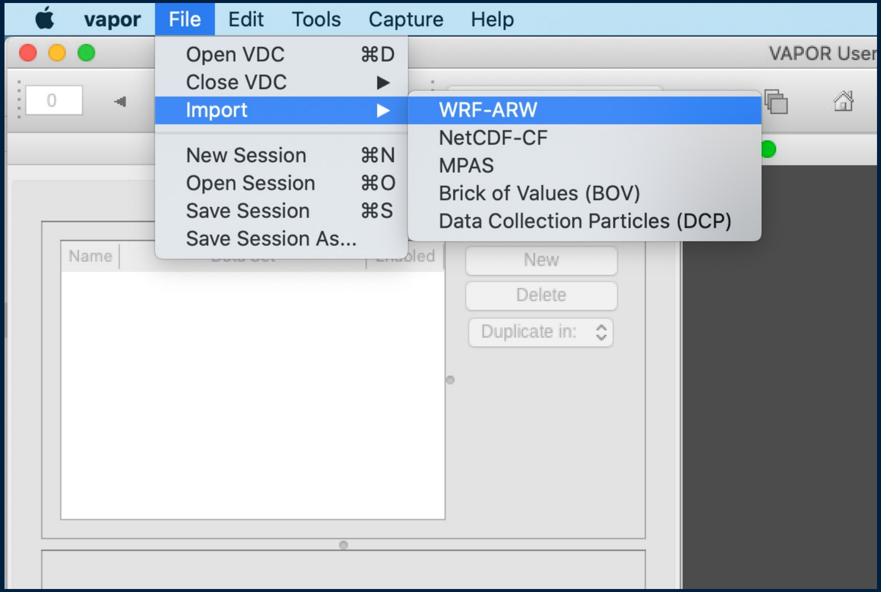
EXAMPLES: VOG DISPERSION



HANDS-ON WORKSHOP

HANDS-ON WORKSHOP: IMPORTING DATA

- **VAPOR is able to directly import:**
  - NetCDF files that follow the CF Convetion (NetCDF-CF)
  - WRF-ARW
  - MPAS
- Other data types need to be converted to VDC manually through command line tools
- VDC allows user to control fidelity



HANDS-ON WORKSHOP: CREATING A RENDER

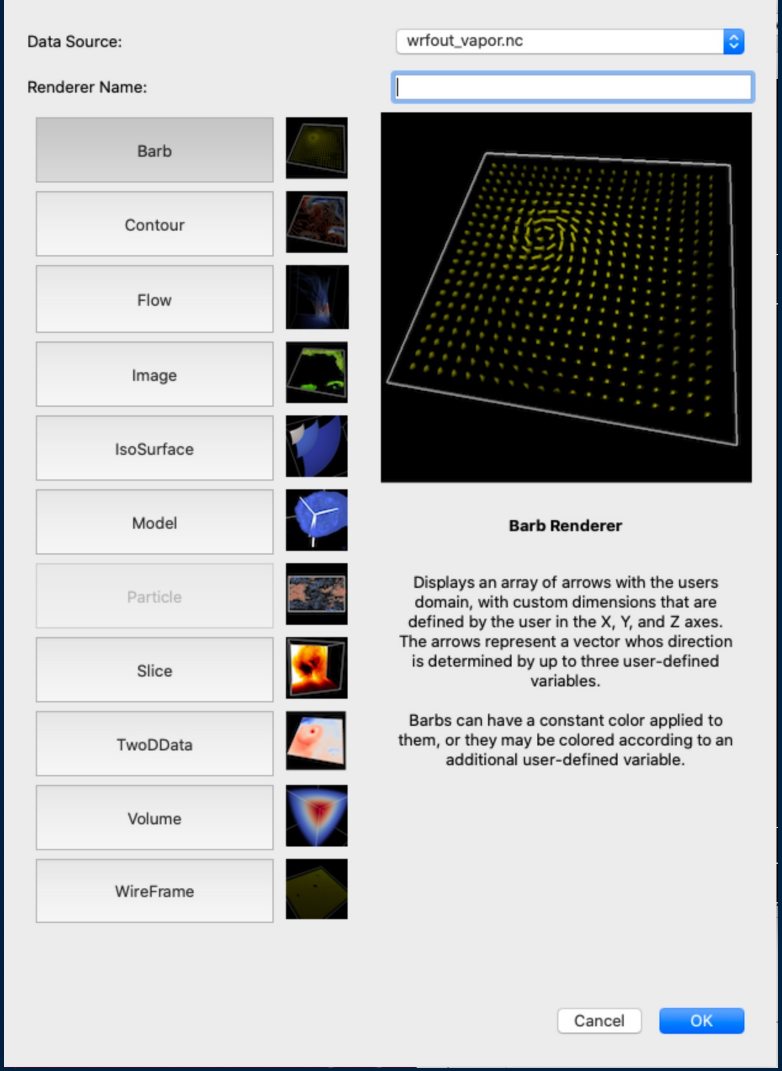
- **VAPOR's main utility = RENDERERS**

- **We WILL NOT:**

  •Perform basic visualization (slicing, contouring etc.)

- **We WILL:**

  •Focus on data exploration



Data Source: wrfout_vapor.nc

Renderer Name:

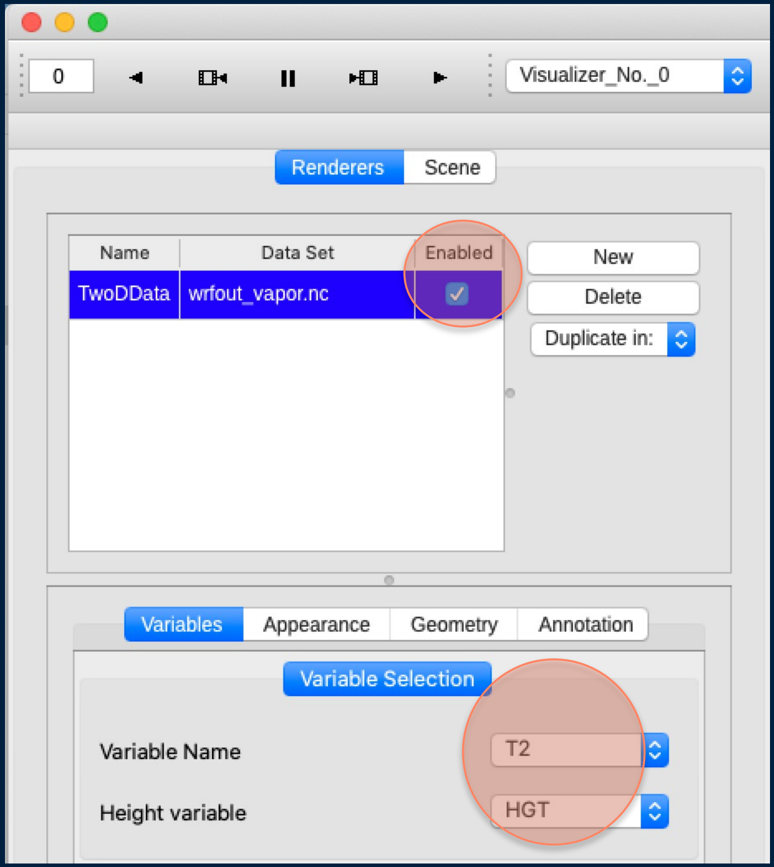
Barb Renderer

Displays an array of arrows with the users domain, with custom dimensions that are defined by the user in the X, Y, and Z axes. The arrows represent a vector whos direction is determined by up to three user-defined variables.

Barbs can have a constant color applied to them, or they may be colored according to an additional user-defined variable.

HANDS-ON WORKSHOP: SURFACE DATA

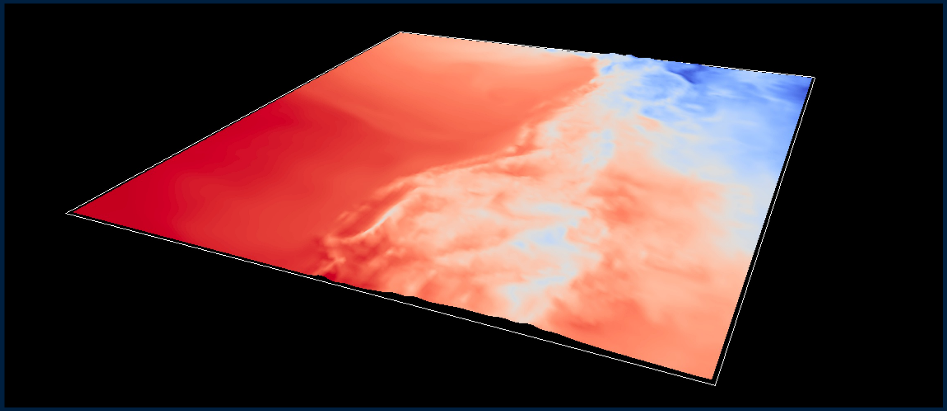
- Create **New** Renderer
- Select **TwoDData**
- Click **Enabled**
- Variable Name **T2**
- Height Variable **HGT**



Name	Data Set	Enabled
TwoDData	wrfout_vapor.nc	☑

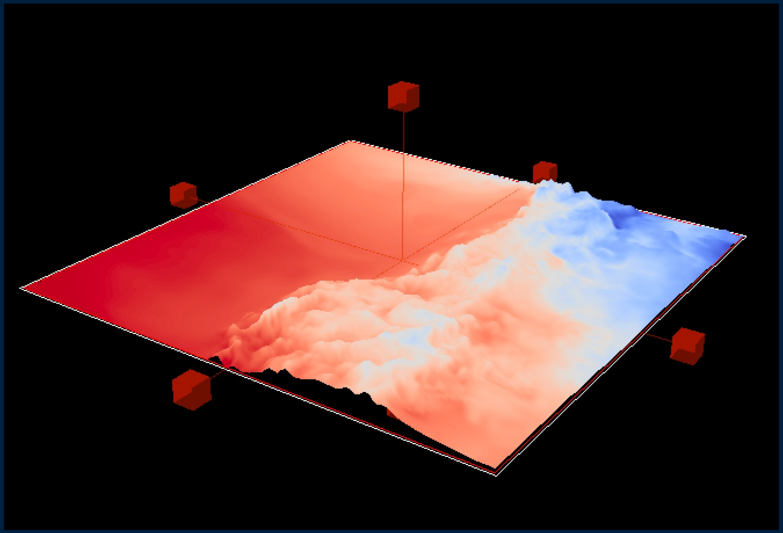
Variable Name T2

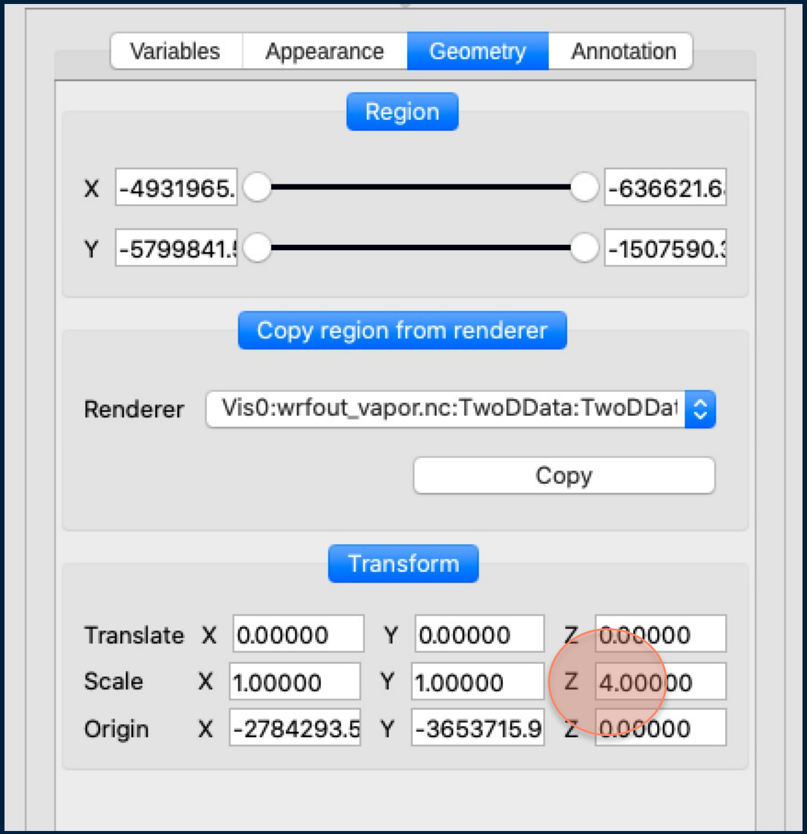
Height variable HGT



HANDS-ON WORKSHOP: SURFACE DATA

- Click **Geometry**
- Set **Scale** to **4** for the vertical (z)





X -4931965. -636621.6

Y -5799841.! -1507590.:

Renderer Vis0:wrfout_vapor.nc:TwoDData:TwoDDat

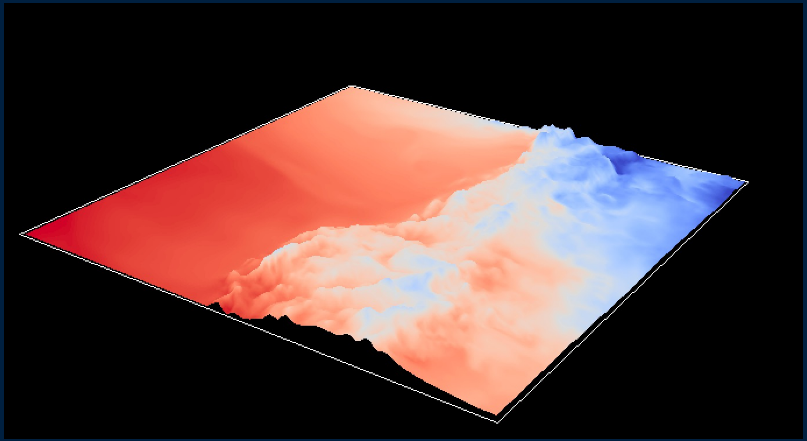
	X		Y		Z	
Translate	X	0.00000	Y	0.00000	Z	0.00000
Scale	X	1.00000	Y	1.00000	Z	4.00000
Origin	X	-2784293.5	Y	-3653715.9	Z	0.00000

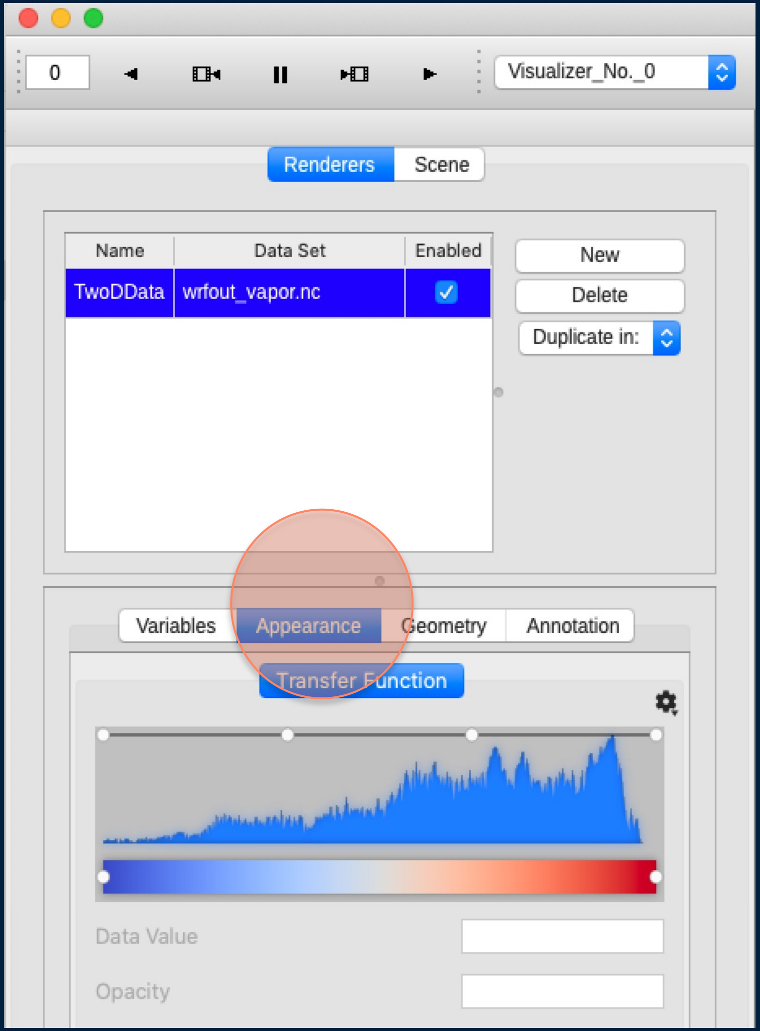
HANDS-ON WORKSHOP: COMMON DISPLAY CONTROLS

- Click **Appearance**

- Explore **Transfer Function**

- **What's wrong with this picture?**
  - **Colors should diverge at 0C for temperature**





Name	Data Set	Enabled
TwoDData	wrfout_vapor.nc	☑

HANDS-ON WORKSHOP: COMMON DISPLAY CONTROLS

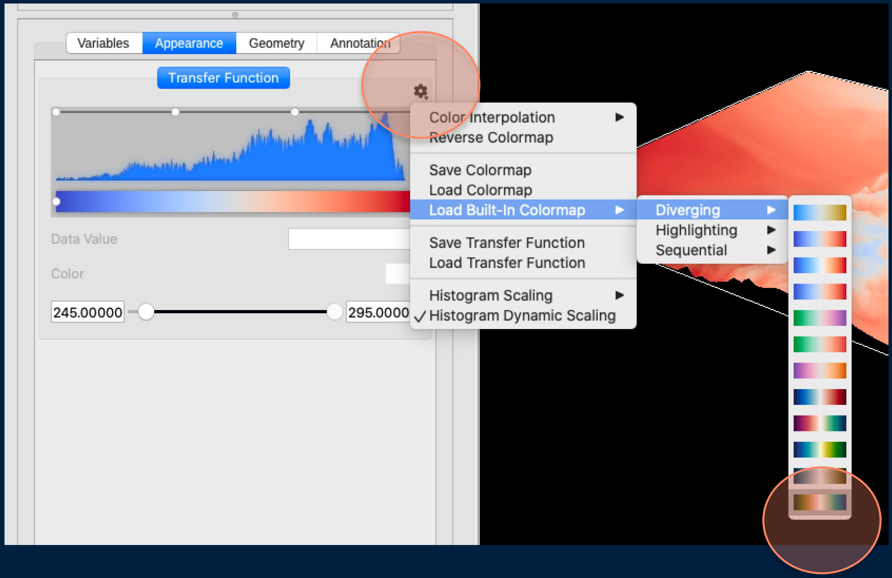
- Let’s select a different colormap and adjust it for our purpose

- Click **Settings** – **Load Built-in Colormap** – **Diverging** – select the last one

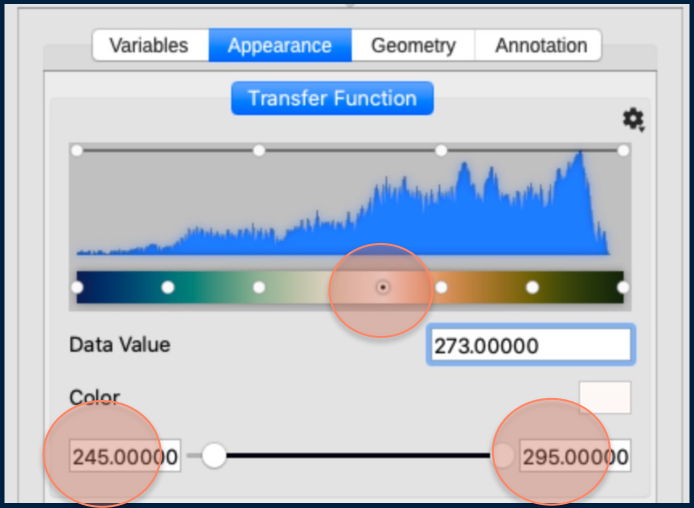
- Click **Settings** again and select **Reverse Colormap**

- Adjust range: set values to go from **245** to **295**

- Click on the middle **Control Point** and set it’s value to **273**



245.00000 – 295.0000



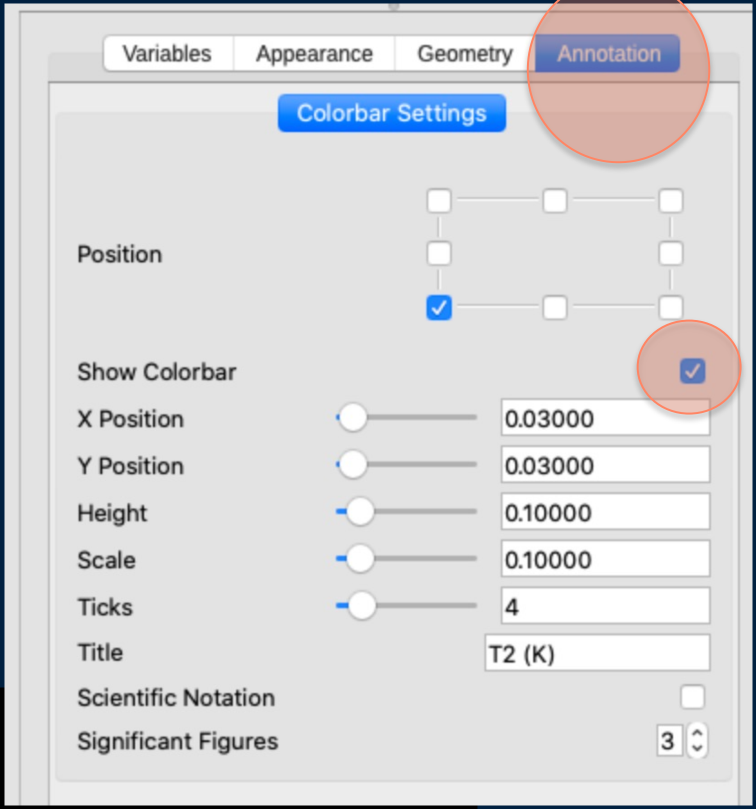
Data Value 273.00000

Color

245.00000 – 295.00000

HANDS-ON WORKSHOP: COMMON DISPLAY CONTROLS

- **Annotate your plot**
  - Select **Annotations**
  - Click **Show Colorbar**
  - Adjust range, position, appearance



Position: ☑ bottom left

Show Colorbar ☑

X Position 0.03000

Y Position 0.03000

Height 0.10000

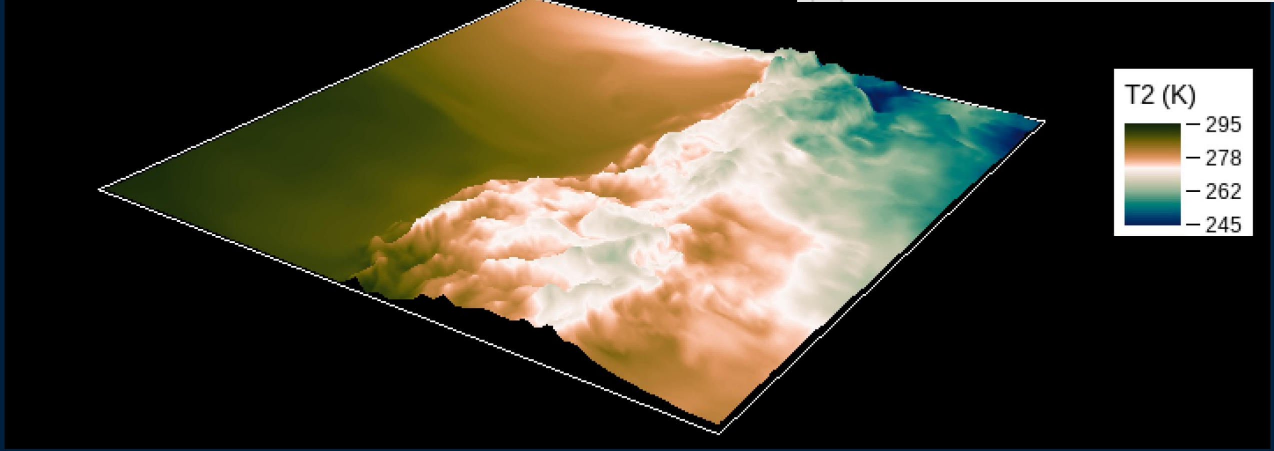
Scale 0.10000

Ticks 4

Title T2 (K)

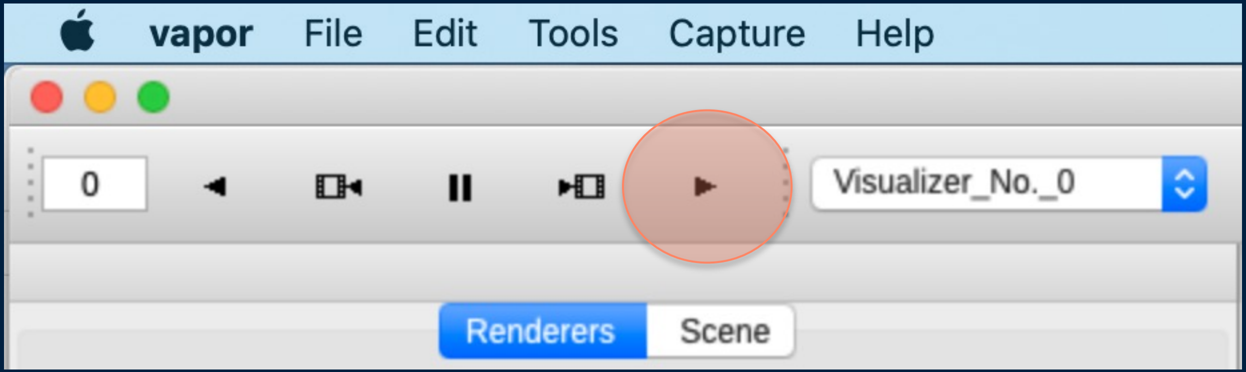
Scientific Notation □

Significant Figures 3



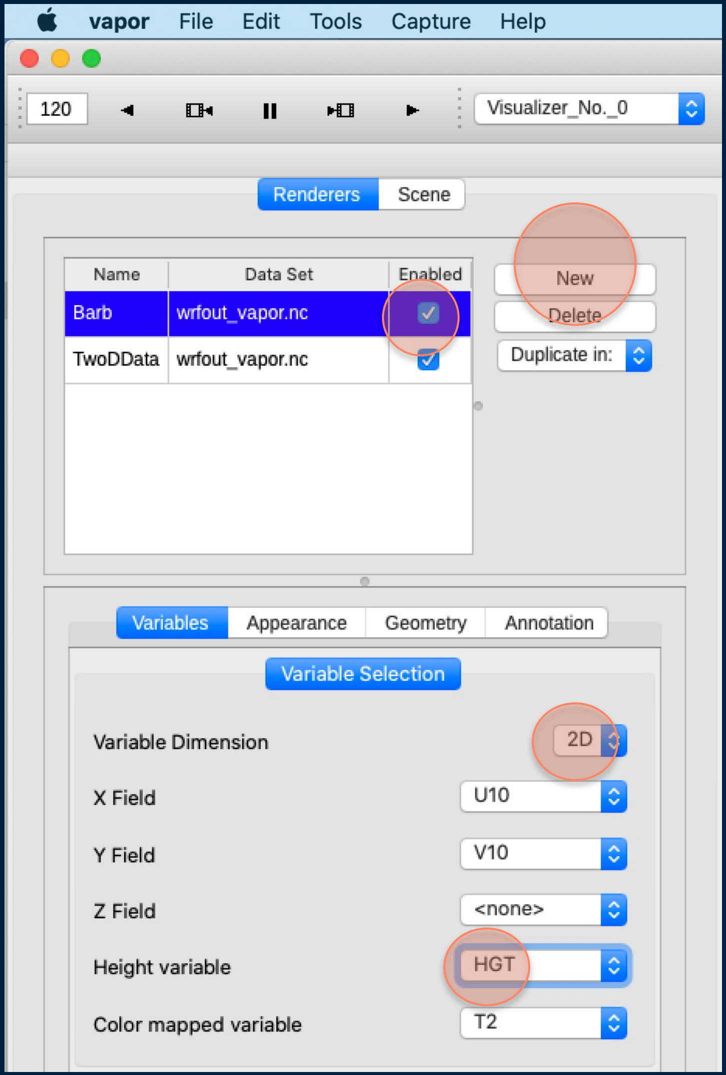
HANDS-ON WORKSHOP:
TIME CONTROL

- **Now lets enjoy what we’ve made so far**
  - In top menu click **Play button**



HANDS-ON WORKSHOP: BARB RENDERER

- Create **New** Renderer
- Select **Barb**
- Click **Enabled**
- Set **Variable Dimension** to **2D**
- Set **Height Variable** to **HGT**



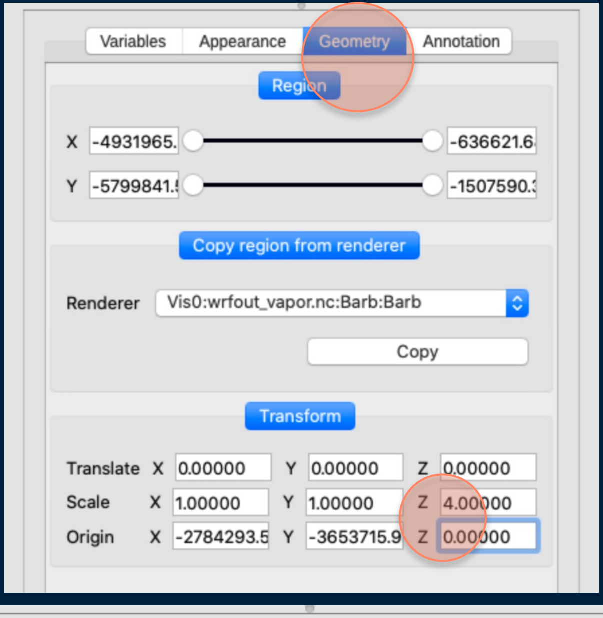
Name	Data Set	Enabled
Barb	wrfout_vapor.nc	☑
TwoDData	wrfout_vapor.nc	☑

Variable Dimension	2D
X Field	U10
Y Field	V10
Z Field	<none>
Height variable	HGT
Color mapped variable	T2

HANDS-ON WORKSHOP: BARB RENDERER

- Under **Geometry**
  - Set **Scale (Z)** to **4**
  - Set **Origin (Z)** to **0**

- Select **Appearance**
  - Set **X Dimension** to **20**
  - Set **Y Dimension** to **20**
  - Set **Thickness Scale** to **0.6**
  - Change range max to **15**
  - Unclick **Use Constant Color**
  - Change colormap to **Sequential - Thermal**

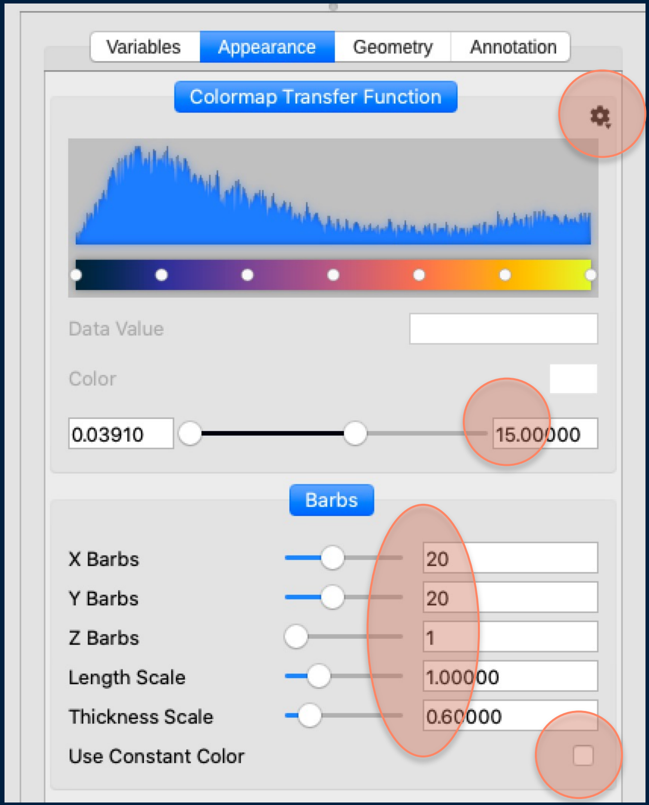


X -4931965. -636621.6

Y -5799841.! -1507590.:

Renderer Vis0:wrfout_vapor.nc:Barb:Barb

Translate	X	0.00000	Y	0.00000	Z	0.00000
Scale	X	1.00000	Y	1.00000	Z	4.00000
Origin	X	-2784293.5	Y	-3653715.9	Z	0.00000



0.03910 15.00000

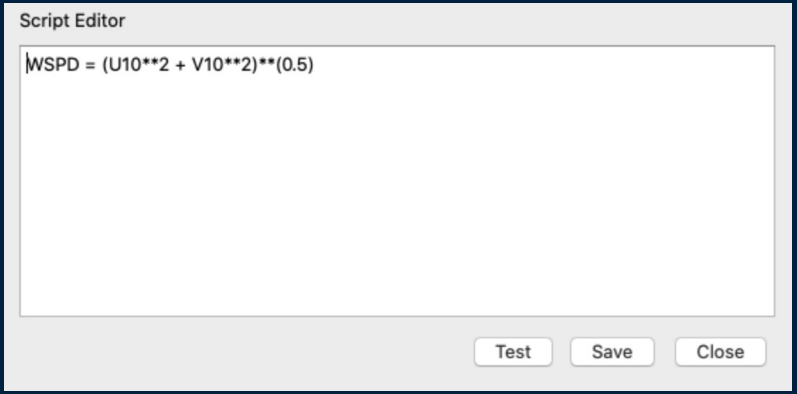
X Barbs	20
Y Barbs	20
Z Barbs	1
Length Scale	1.00000
Thickness Scale	0.60000
Use Constant Color	☐

HANDS-ON WORKSHOP: PYTHON VARIABLE EDITOR

- In **Tools** select **Python Variables**
- Click **New** to create a new script **GetWSPD**
- Select **U10** and **V10** as Input Variables
- Create new output variable **WSPD**
- Write Python code in Script Editor:

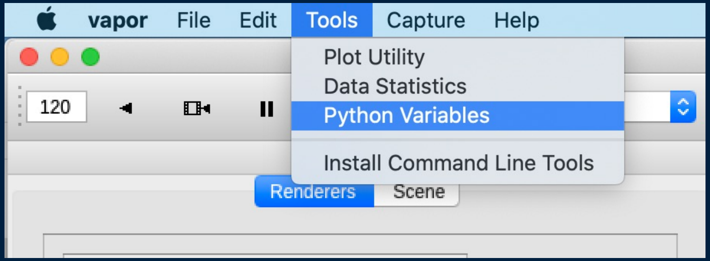
  WSPD = (U102 + V10**2)**(0.5)**

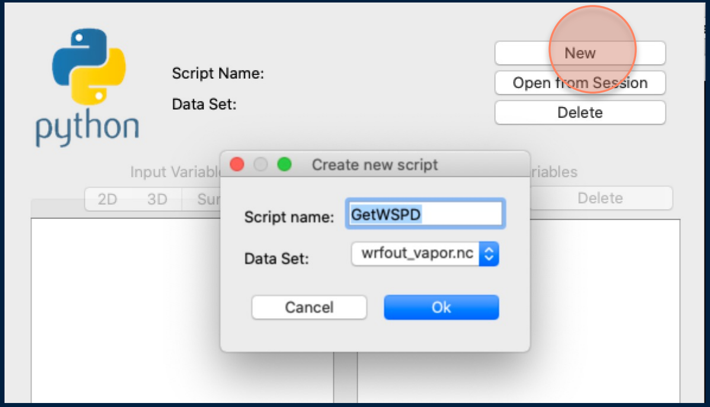
- Click **Test - Save - Close**



Script Editor

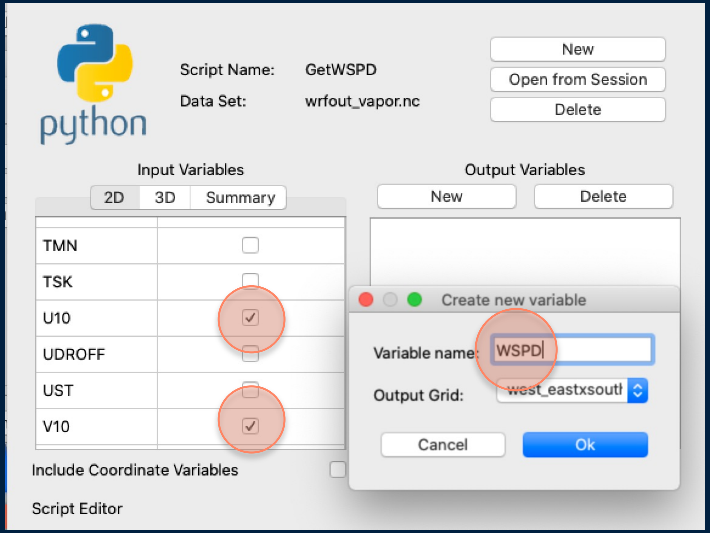
```
WSPD = (U10**2 + V10**2)**(0.5)
```





Script name: GetWSPD

Data Set: wrfout_vapor.nc



Script Name: GetWSPD

Data Set: wrfout_vapor.nc

TMN	□
TSK	□
U10	☑
UDROFF	□
UST	□
V10	☑

Include Coordinate Variables □

Variable name: WSPD

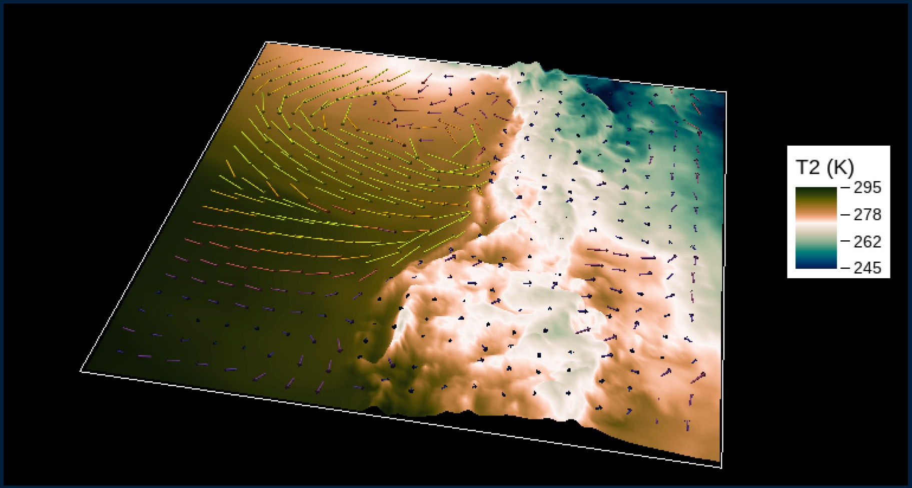
Output Grid: west_eastxsouth

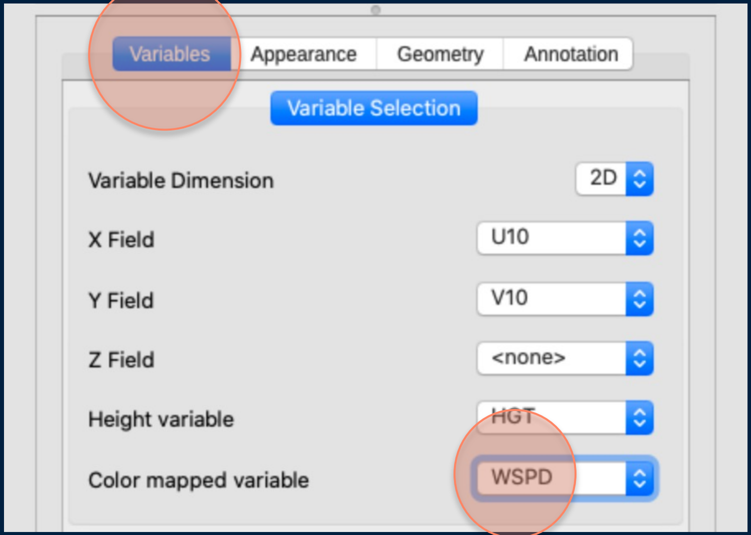
HANDS-ON WORKSHOP: COLORIZING THE BARBS

- Select **Variables**
  - change **Color mapped variable** to **WSPD**

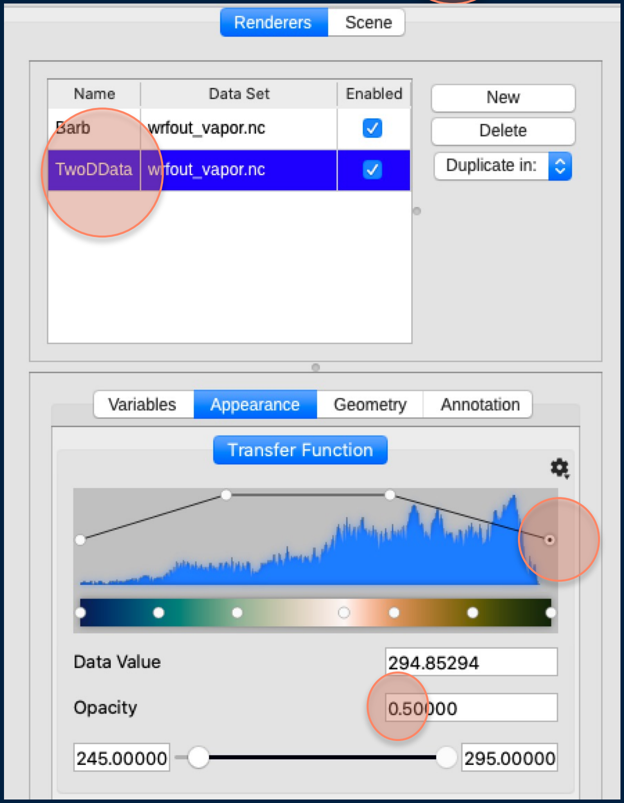
- Change Renderer to **TwoDData** and under **Appearance** set the opacity of first and last points to **0.5**

- **Play**





Variable Dimension	2D
X Field	U10
Y Field	V10
Z Field	<none>
Height variable	HGT
Color mapped variable	WSPD

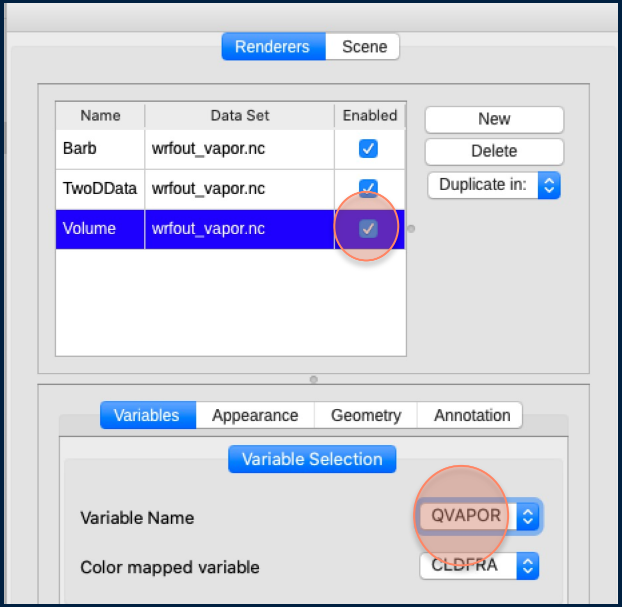


Name	Data Set	Enabled
Barb	wrfout_vapor.nc	☑
TwoDData	wrfout_vapor.nc	☑

Data Value	294.85294
Opacity	0.50000
245.00000	295.00000

HANDS-ON WORKSHOP: VOLUME RENDERING

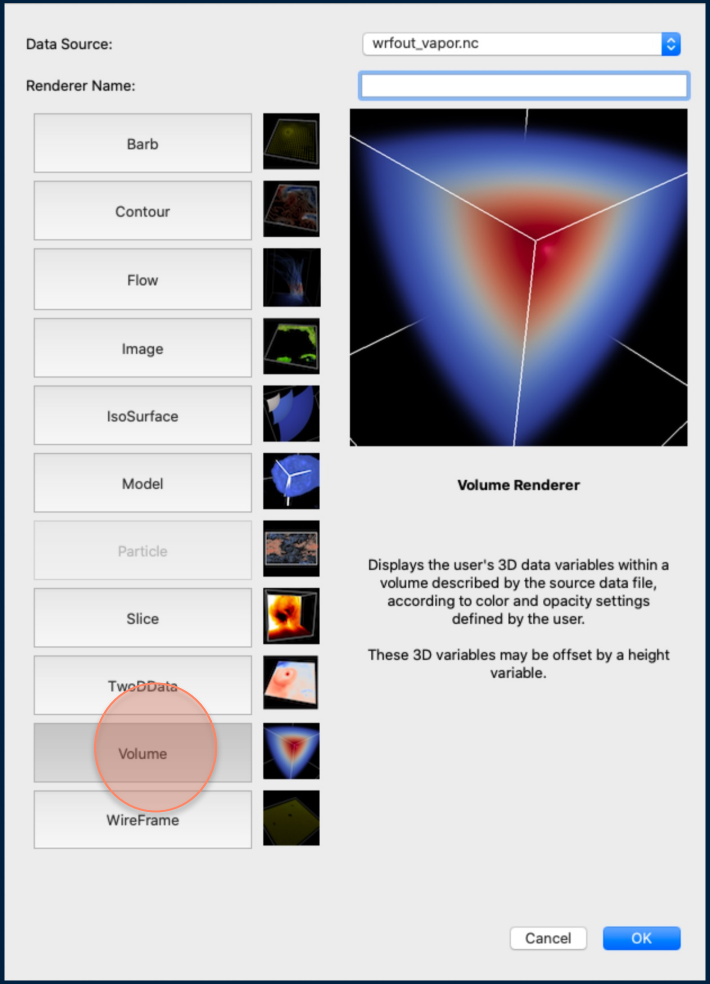
- Create **New** Renderer
- Select **Volume**
- Click **Enabled**
- Set **Variable Name** to **QVAPOR**



Name	Data Set	Enabled
Barb	wrfout_vapor.nc	☑
TwoDData	wrfout_vapor.nc	☑
Volume	wrfout_vapor.nc	☑

Variable Name: QVAPOR

Color mapped variable: CLDFRA



Data Source: wrfout_vapor.nc

Renderer Name:

Volume Renderer

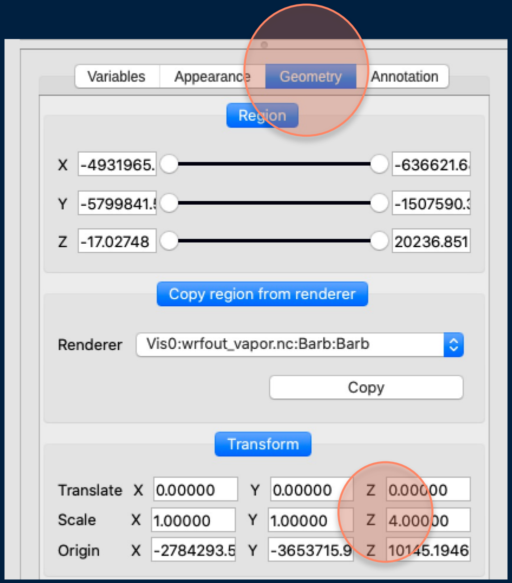
Displays the user's 3D data variables within a volume described by the source data file, according to color and opacity settings defined by the user.

These 3D variables may be offset by a height variable.

HANDS-ON WORKSHOP: VOLUME RENDERING

- Under **Geometry**
  - Set **Scale (z)** to **4**

- Under **Appearance**
  - Change colormap to **Sequential – Ice**
  - Set data range from **0** to **0.01**
  - Set Opacity points to **0, 0.5, 0.1, 0.4**
  - Experiment with **Lighting Parameters**



X -4931965. — -636621.6

Y -5799841.! — -1507590.:

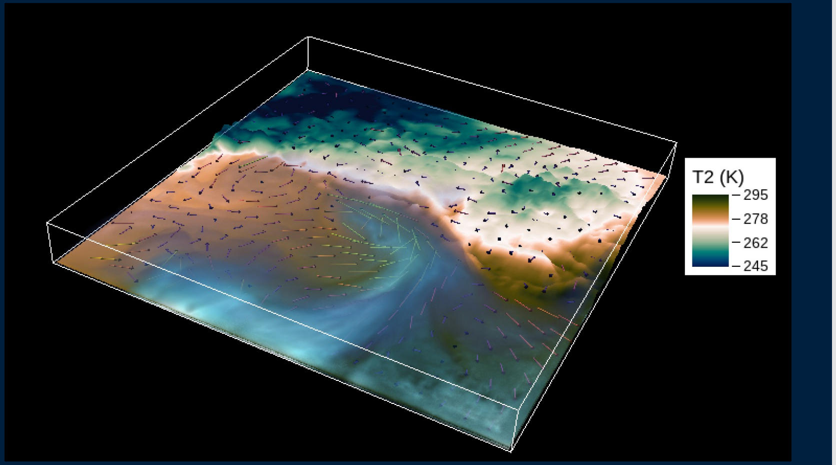
Z -17.02748 — 20236.851

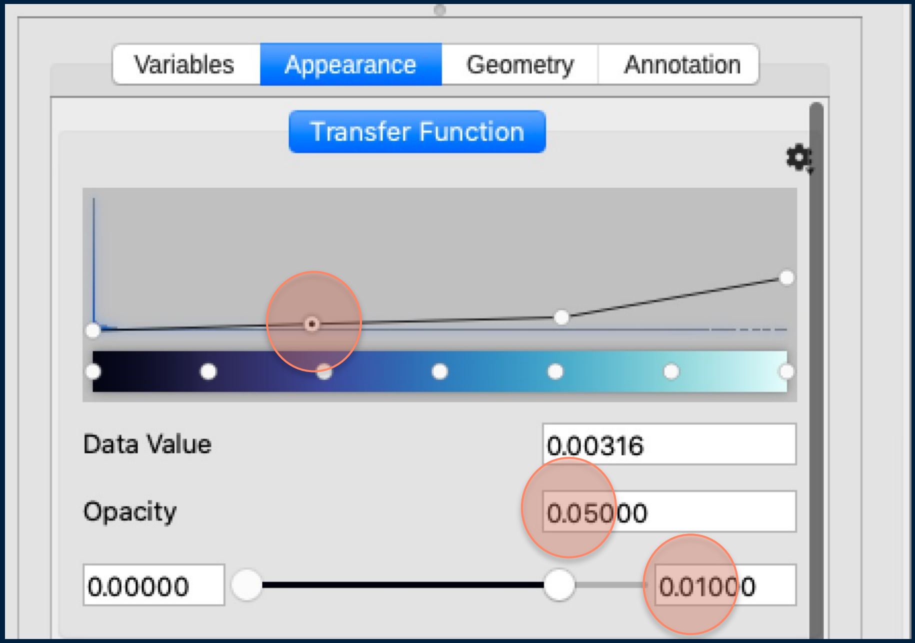
Renderer Vis0:wrfout_vapor.nc:Barb:Barb

Translate X 0.00000 Y 0.00000 Z 0.00000

Scale X 1.00000 Y 1.00000 Z 4.00000

Origin X -2784293.5 Y -3653715.9 Z 10145.1946





Data Value 0.00316

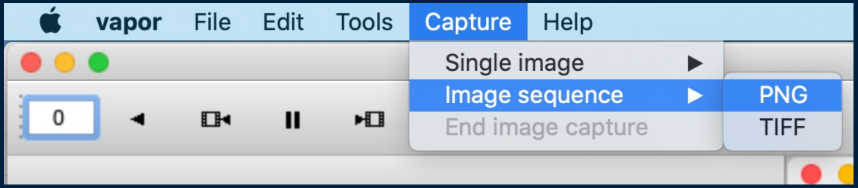
Opacity 0.05000

0.00000 — 0.01000

HANDS-ON WORKSHOP:
EXPORTING YOUR WORK

- Set frame number to **0**

- From **Capture – Image Sequence** select **PNG**



- Select save location and click **Play**

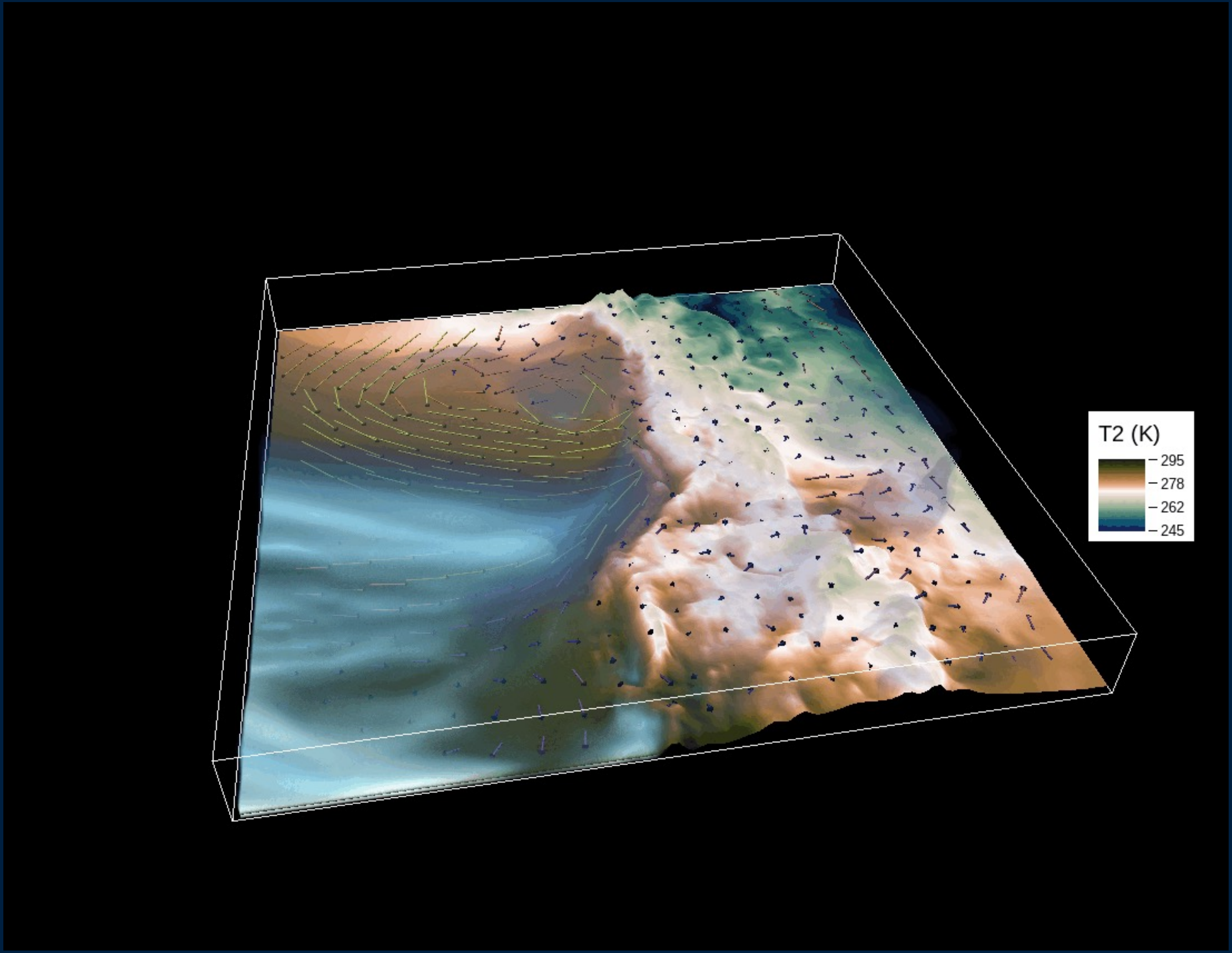
- Animate your frames using program of choice, for example:

```
convert *.png vapordemo.gif
```

OR

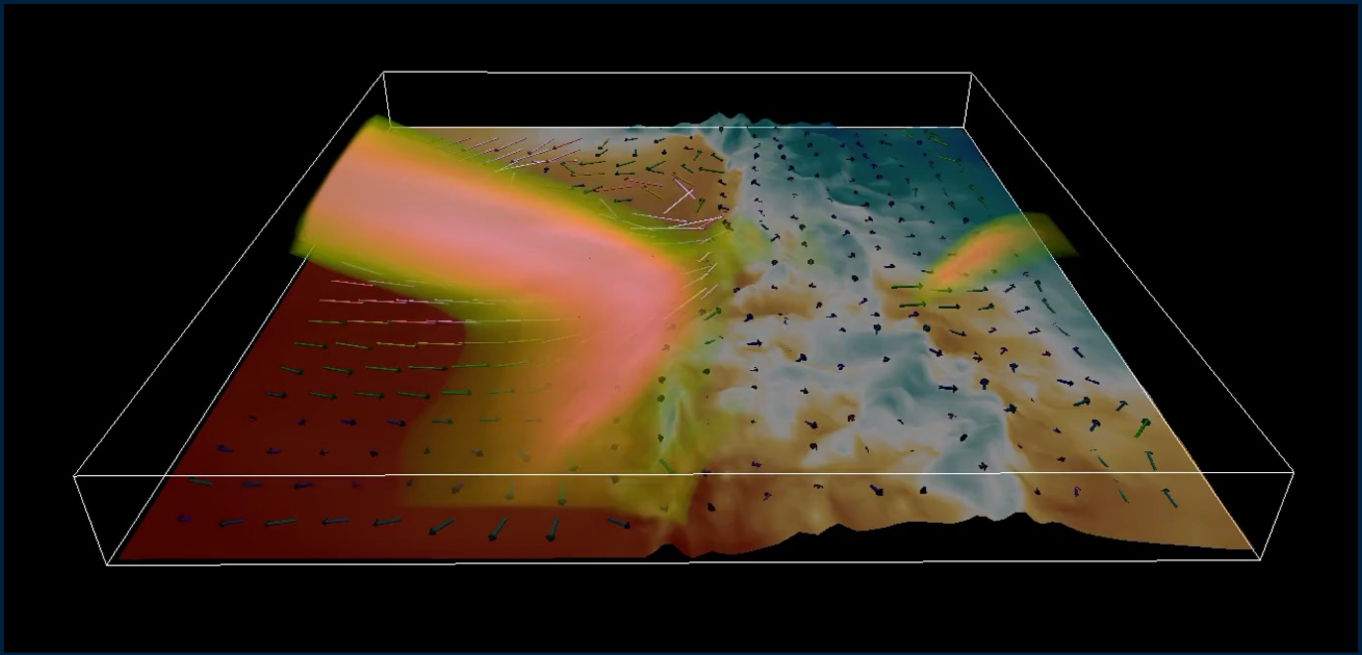
```
ffmpeg -framerate 15 –I demo%04d.png -vf scale=-2:836 -pix_fmt yuv420p  demo.mp4
```

ENJOY



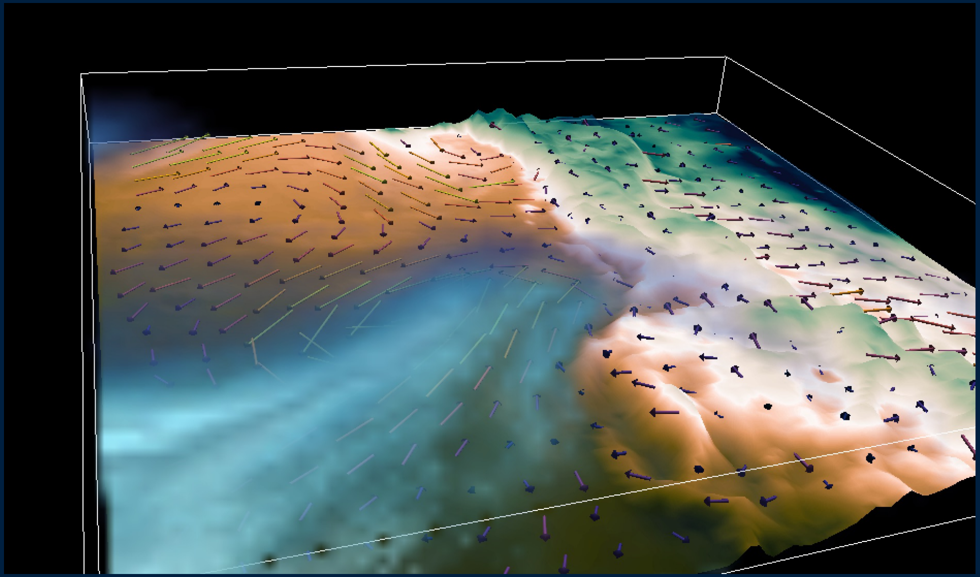
A [HOMEWORK?] CHALLENGE:

- **Try to visualize the jet in three dimensions**
- **Hints:**
  - Create a volume renderer of 3D wind fields and play around with ranges, opacity and colormaps
    - or test out an Isosurface renderer



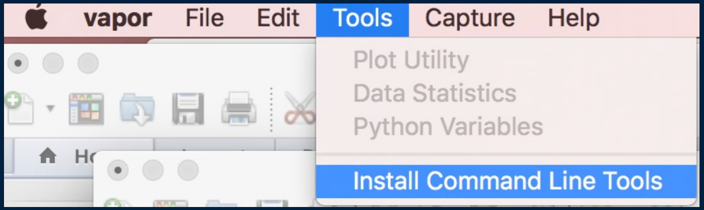
HANDS-ON WORKSHOP:
SUMMARY

- What we [hopefully] learned:
  - **Importing data into VAPOR**
  - **Using Renderers:**
    - TwoDData
    - Barbs
    - Volume
  - **Creating new Python Variables**
  - **Using Display Controls**
    - Working with colormaps
    - Working with data and opacity ranges
  - **Exporting your work**



APPENDIX: CONVERTING DATA TO VDC

- Install Command Line Tools

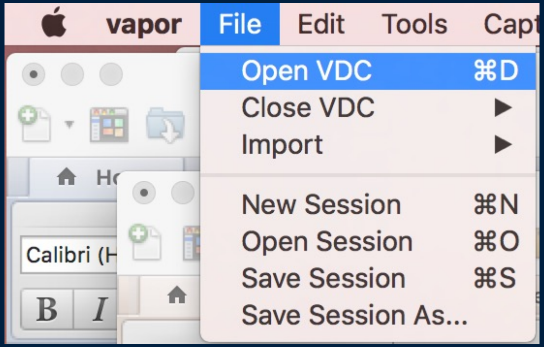


- Log out/in
- Open a *new* terminal window and navigate to your data folder
- Run the following commands:

```
wrfvdccreate wrfout_vapor.nc wrfout_vapor.vdc

wrf2vdc wrfout_vapor.nc wrfout_vapor.vdc
```

- **File – Open VDC**


vapor
File
Edit
Tools
Capt
Open VDC ⌘D
Close VDC ▶
Import ▶
New Session ⌘N
Open Session ⌘O
Save Session ⌘S
Save Session As...