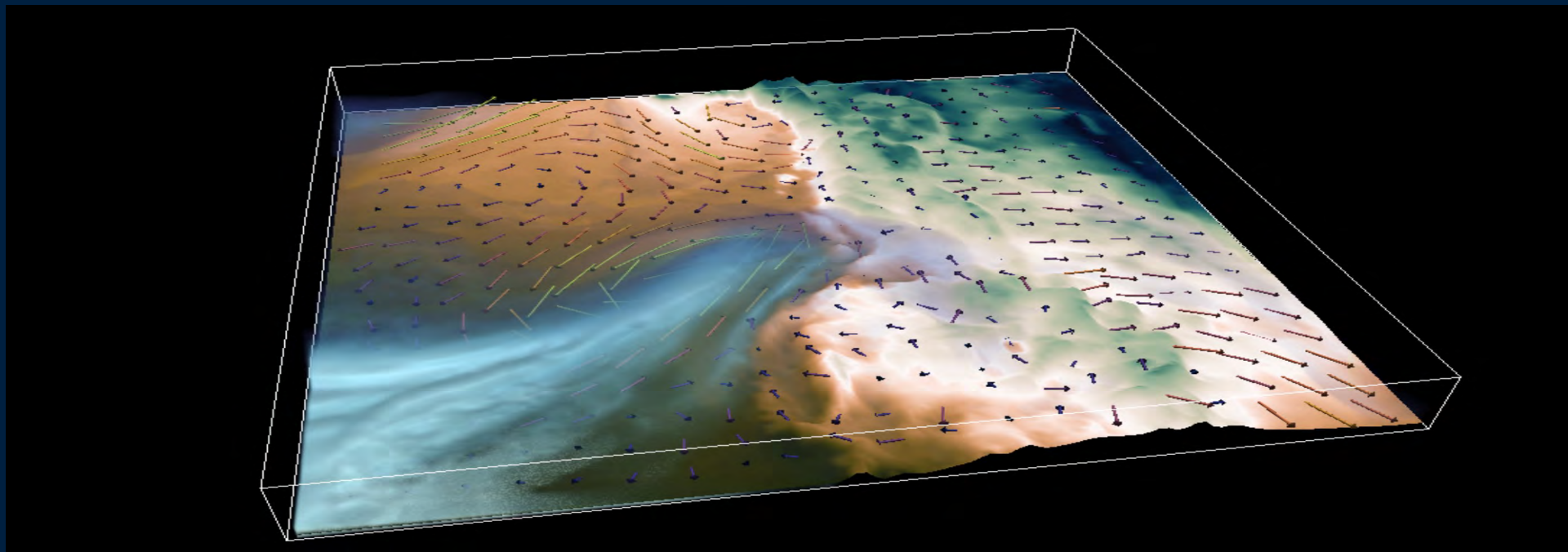
INTERACTIVE VISUALIZATION OF WRF DATA WITH VAPOR 3.9

VAPOR: Data Exploration Tool Designed for Earth System Science

MARCH 11, 2024



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

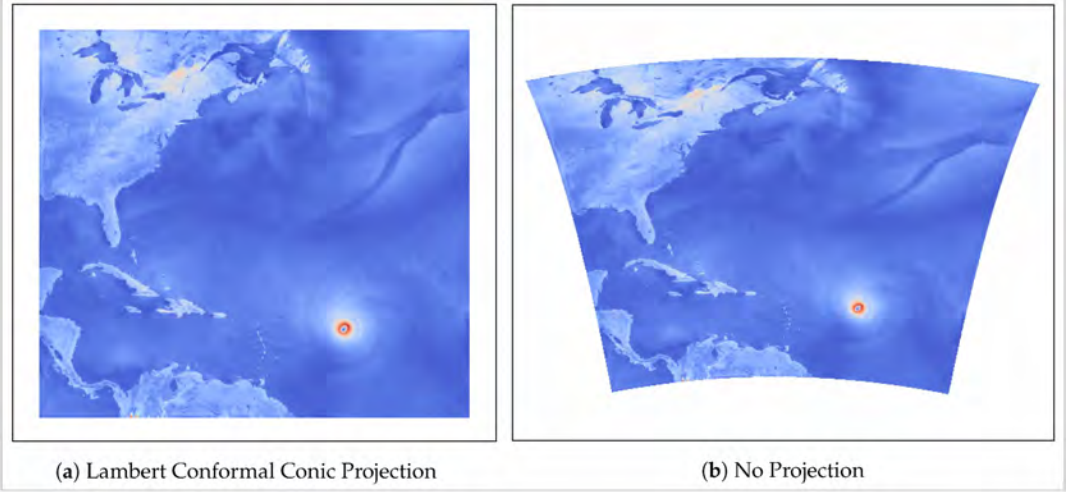
INTRO: VAPOR SOFTWARE

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)

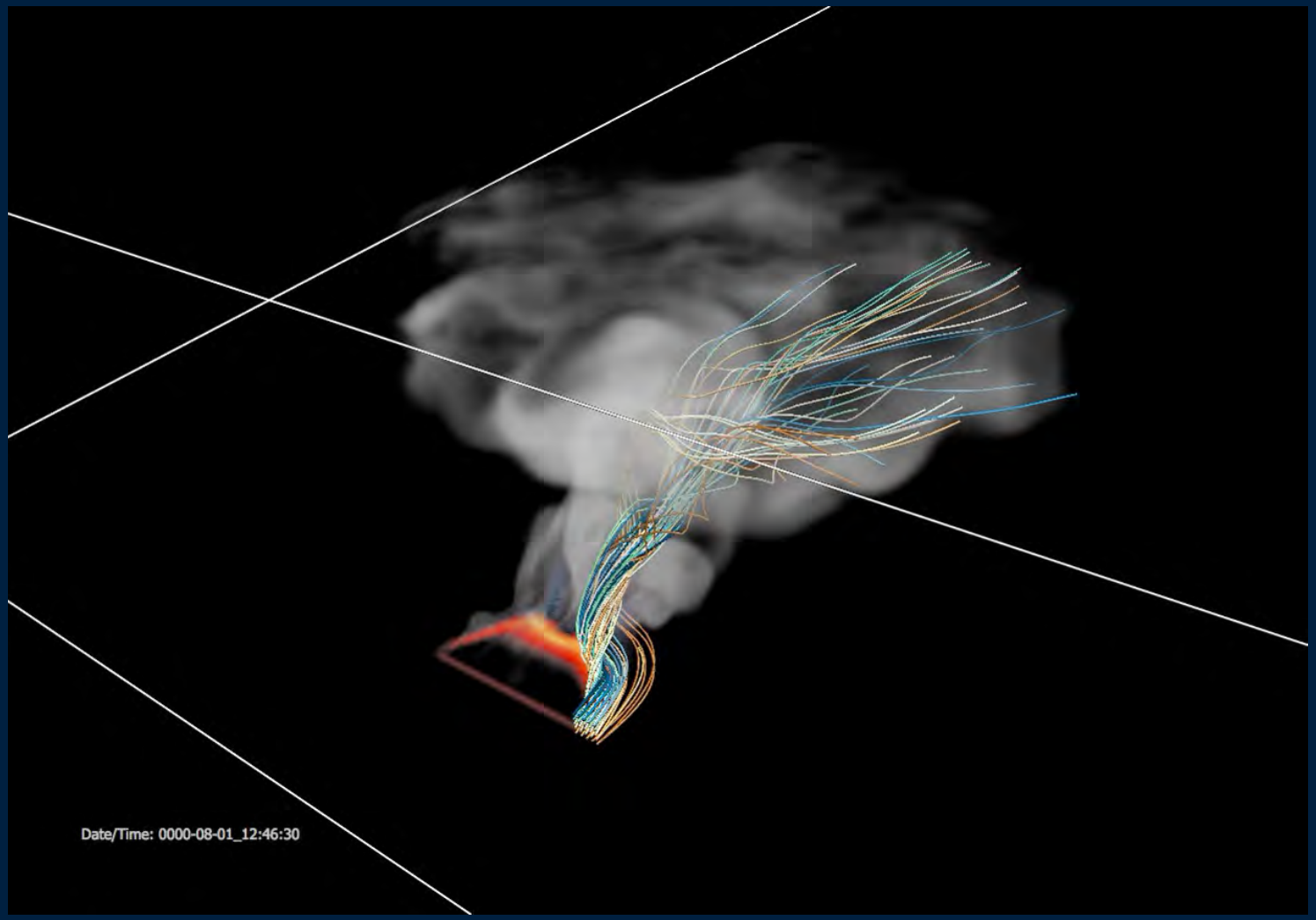
- **What makes VAPOR fast?**

  - **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

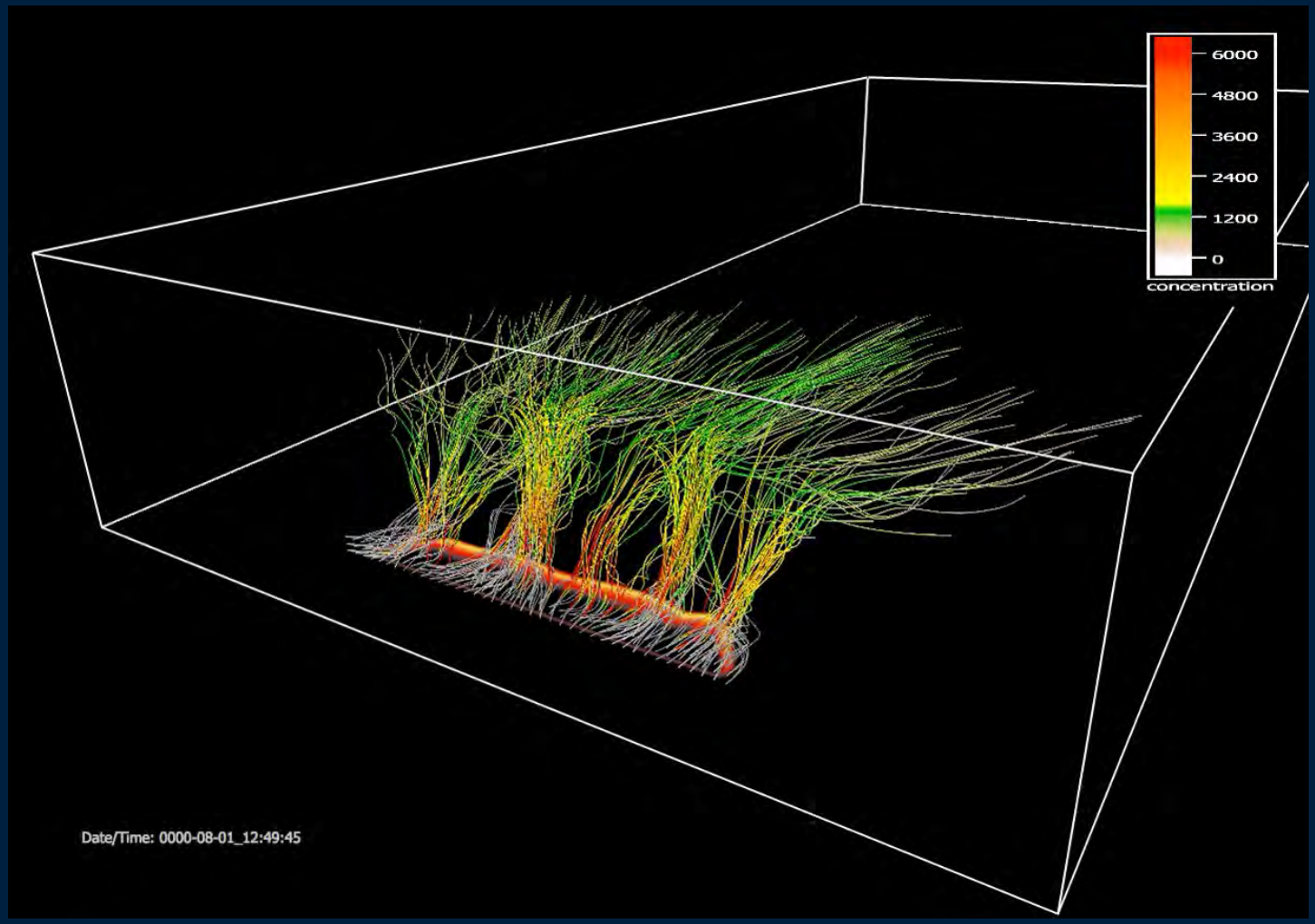
  - both possible using **discrete wavelet transform**
    - transform data into frequency space and store coefficients only (VDC format)
    - combine both controls above into a single **fidelity control**

  - **THIS IS OPTIONAL!**

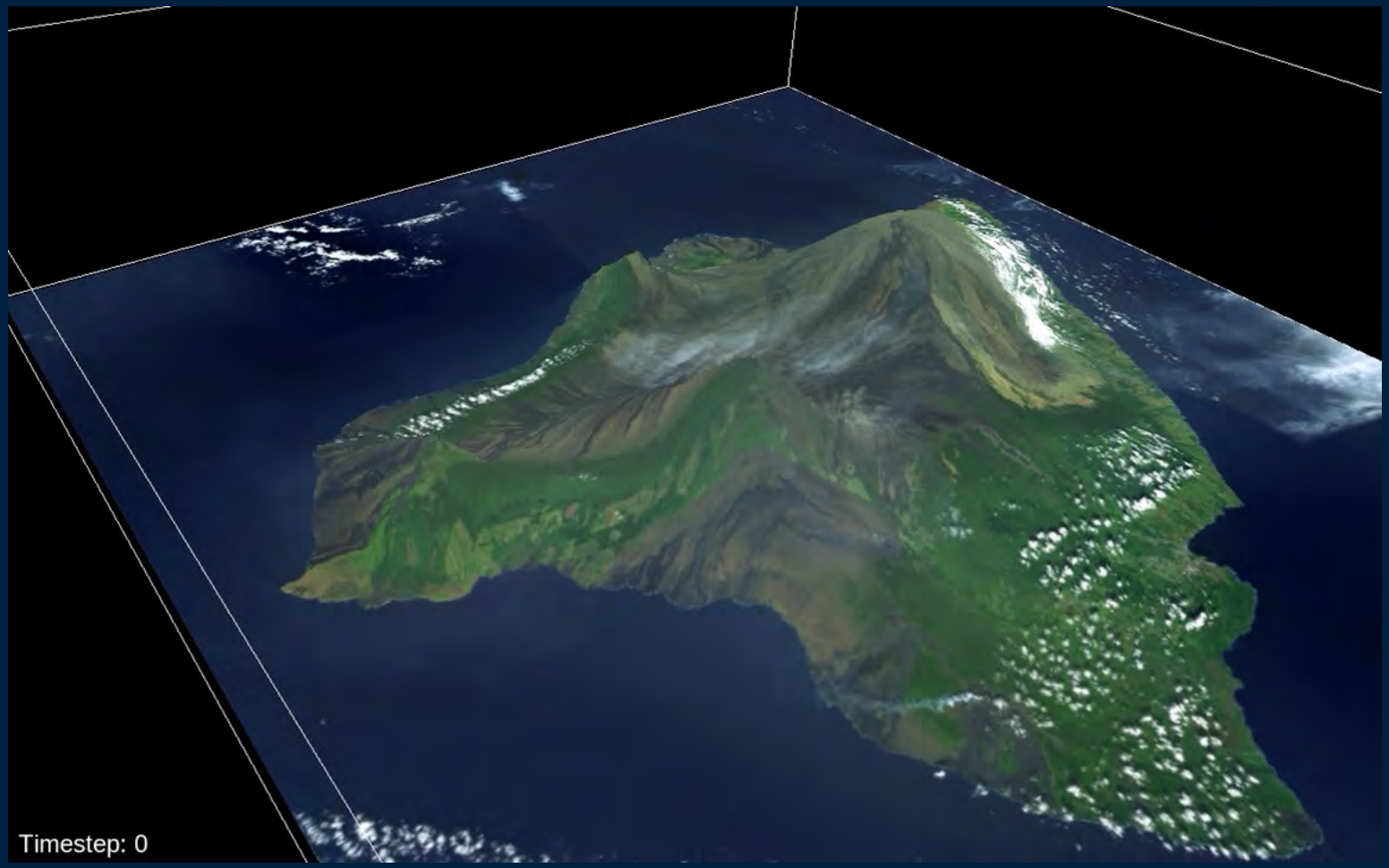
EXAMPLES: WILDFIRE SMOKE AND TURBULENCE



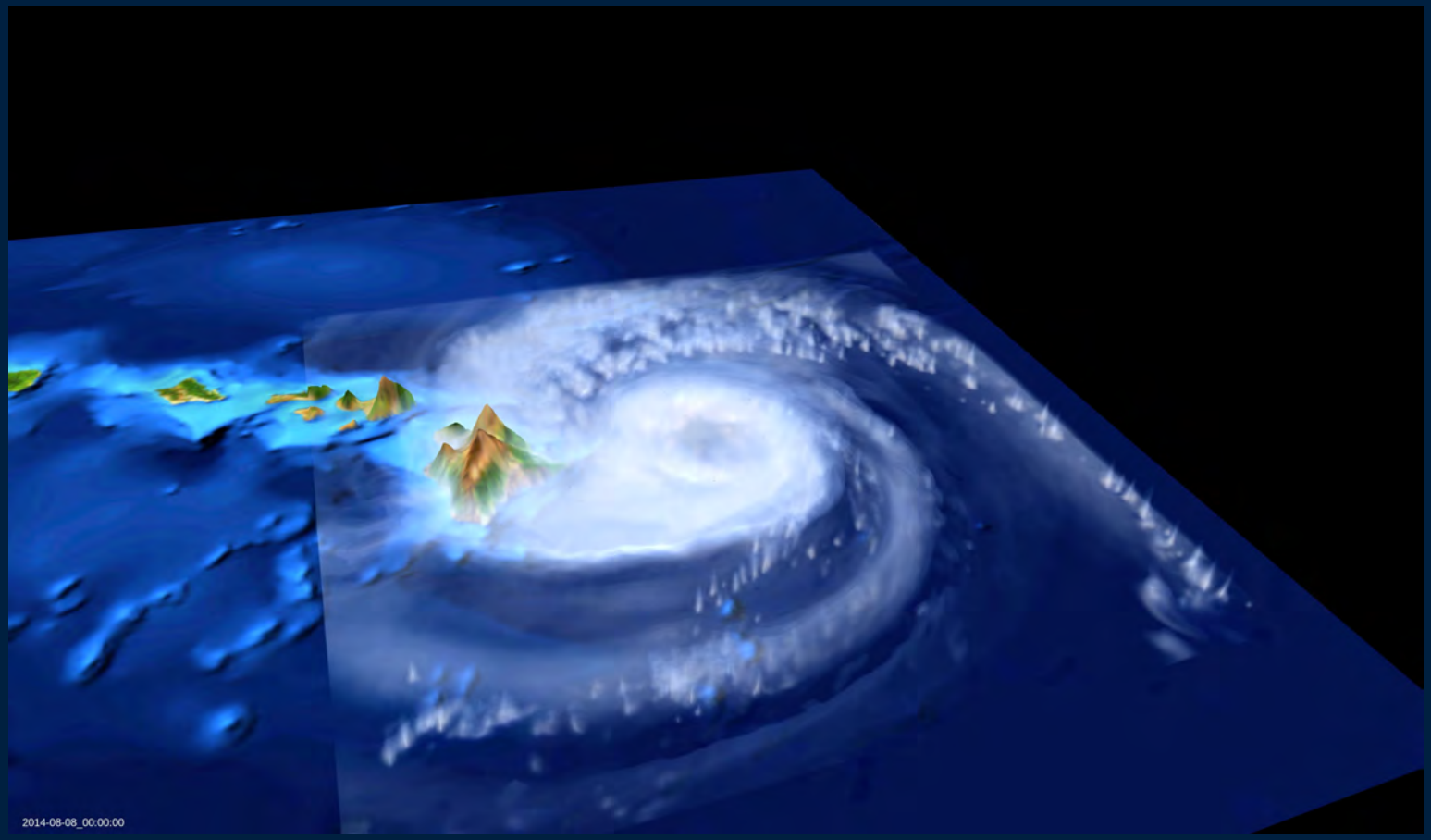
EXAMPLES: WILDFIRE SMOKE AND TURBULENCE



EXAMPLES: VOG DISPERSION



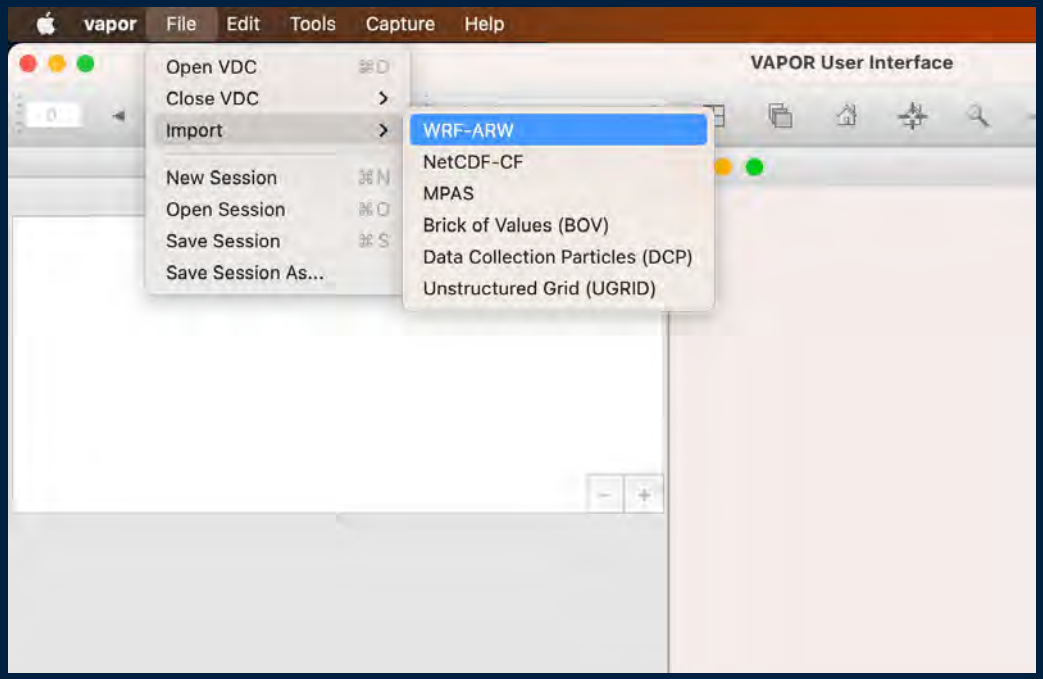
EXAMPLES: HURRICANE ISELLE



HANDS-ON WORKSHOP

HANDS-ON WORKSHOP: IMPORTING DATA

- **VAPOR is able to directly import:**
  - NetCDF files that follow the CF Convention (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

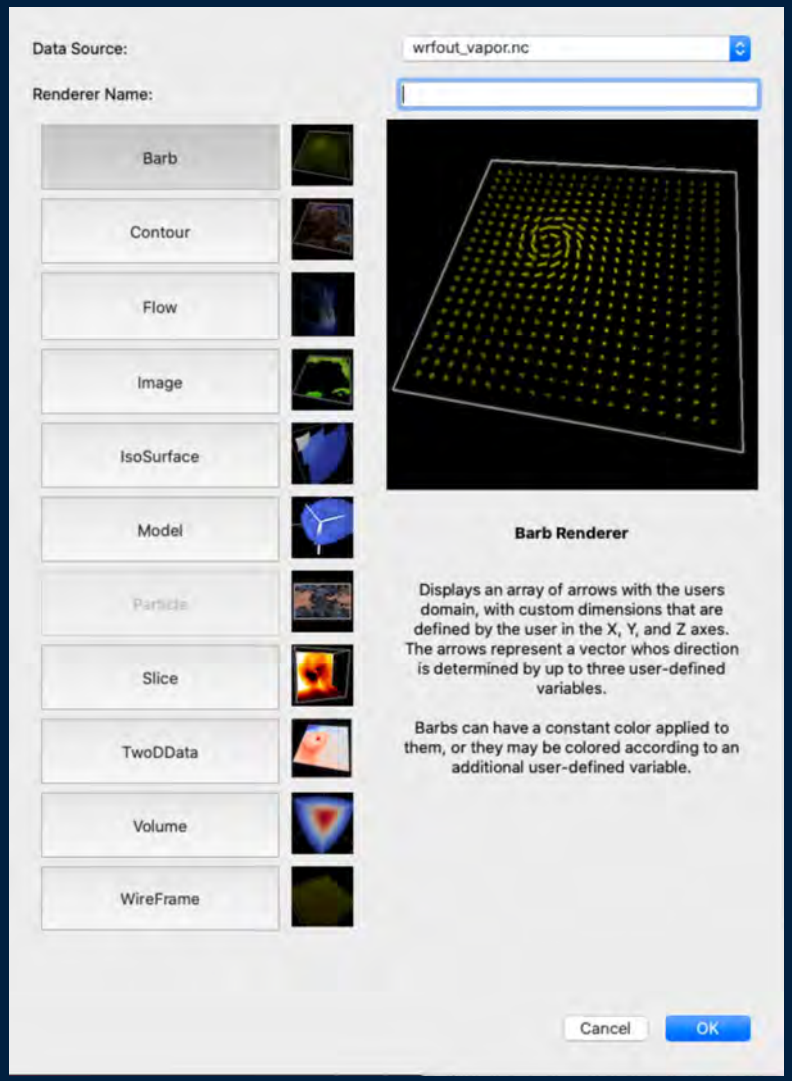


HANDS-ON WORKSHOP: CREATING A RENDER

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

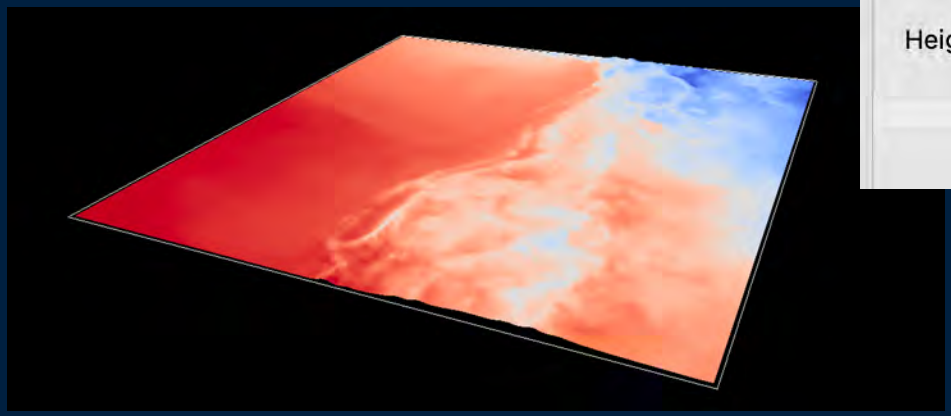
- **We WILL NOT:**
  - Perform basic visualization (slicing, contouring etc.)

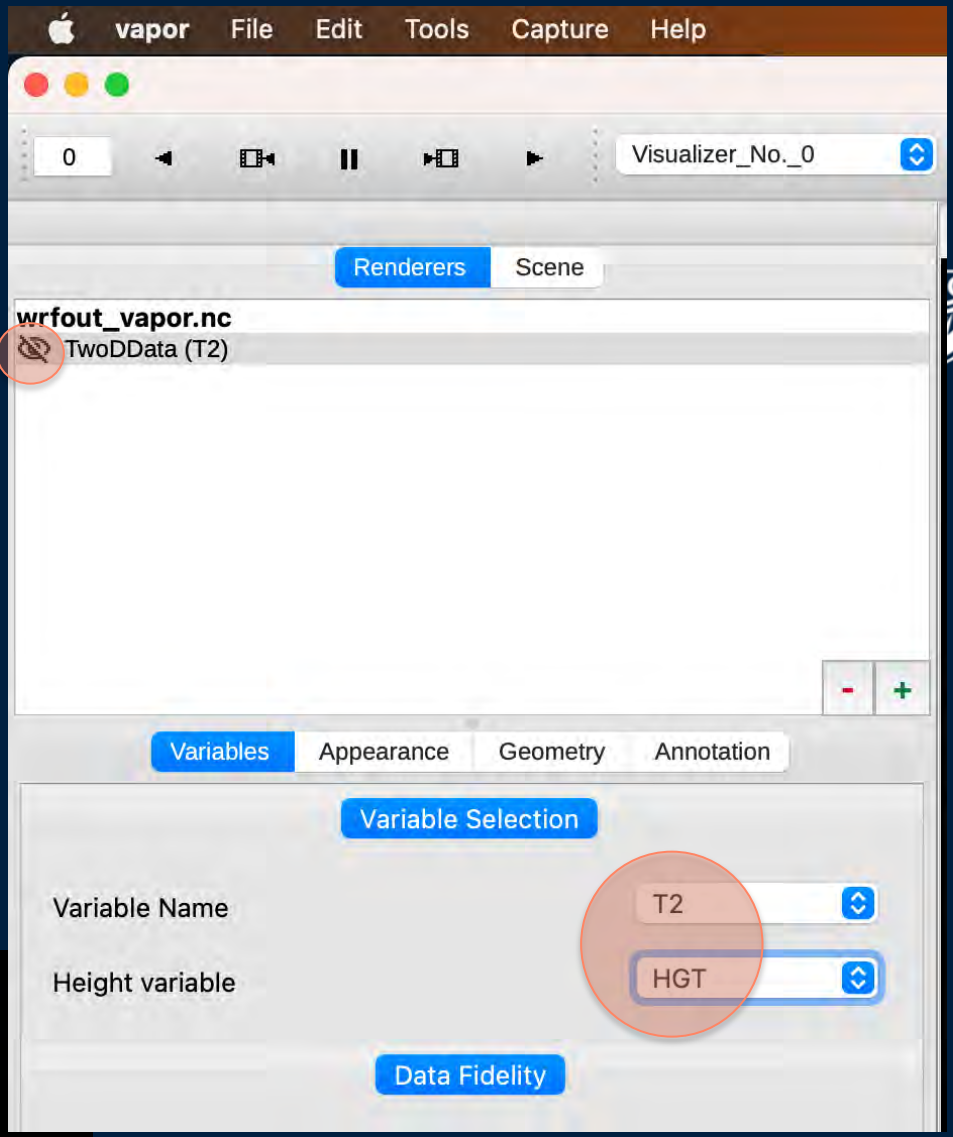
- **We WILL:**
  - Focus on data exploration



HANDS-ON WORKSHOP: SURFACE DATA

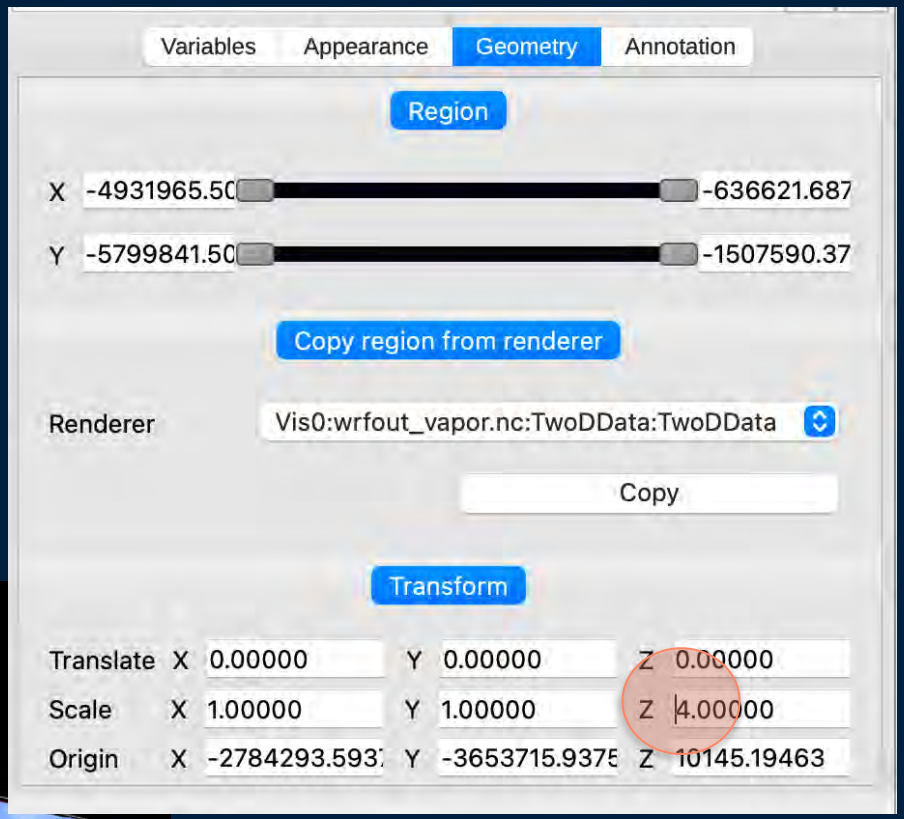
- Create **New** Renderer
- Select **TwoDData**
- Variable Name **T2**
- Height Variable **HGT**
- Enable viewing

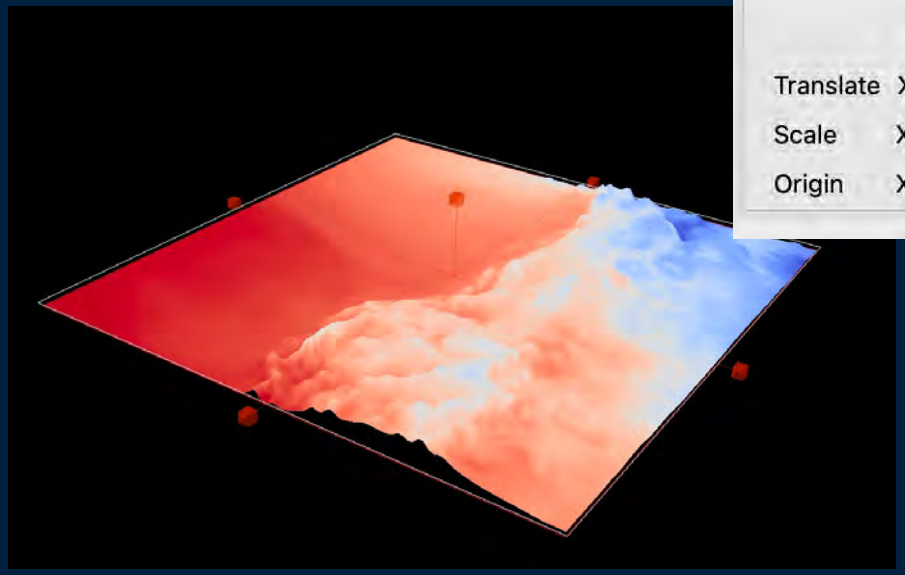




HANDS-ON WORKSHOP: SURFACE DATA

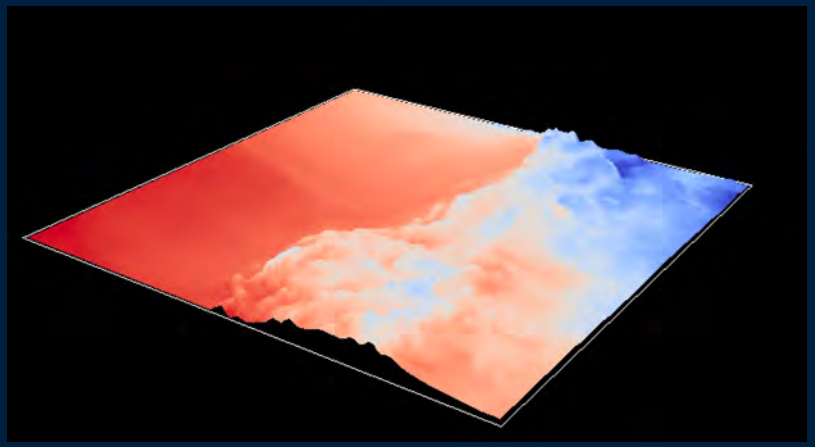
- Click **Geometry**
- Set **Scale** to **4** for the vertical (z)





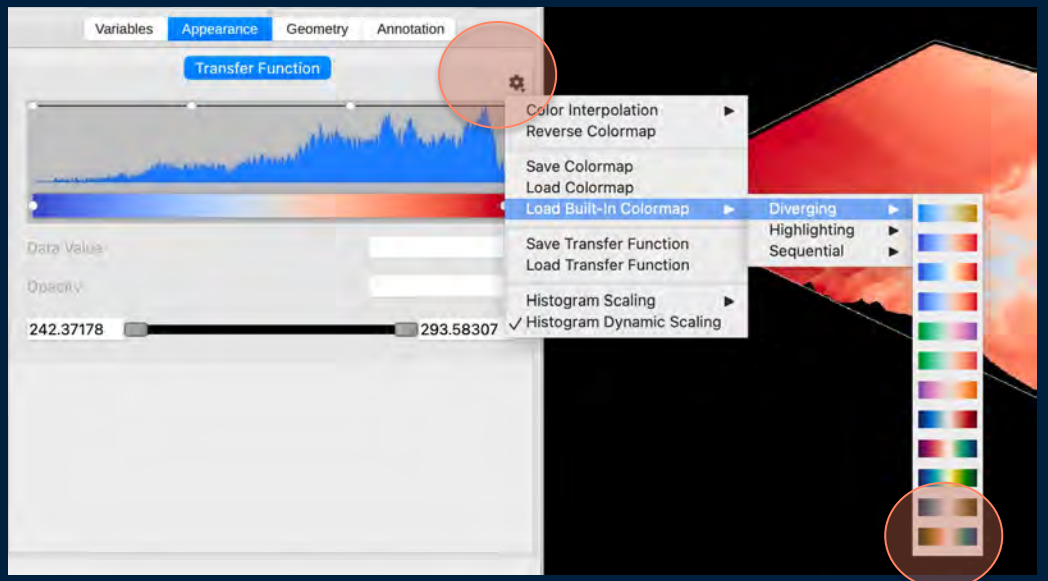
HANDS-ON WORKSHOP: COMMON DISPLAY CONTROLS

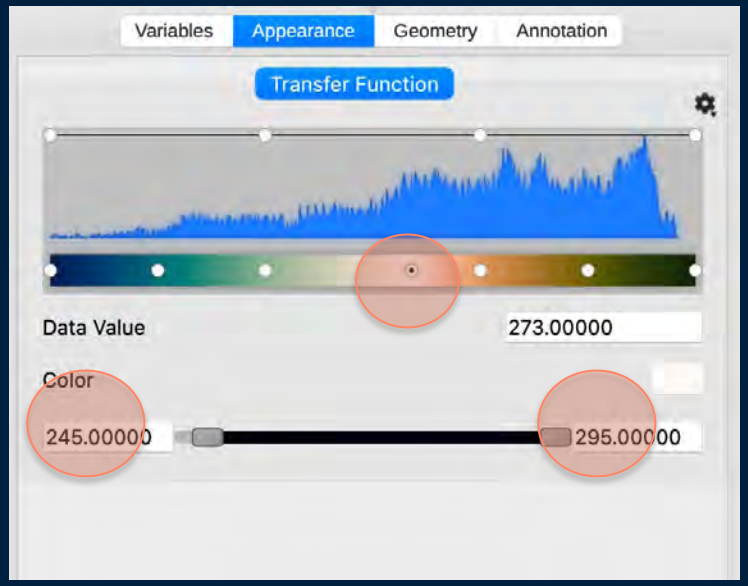
- Click **Appearance**
- Explore **Transfer Function**
- Are default colormap setting appropriate**?**
  - **Colors should diverge at 0C for temperature**



HANDS-ON WORKSHOP: COMMON DISPLAY CONTROLS

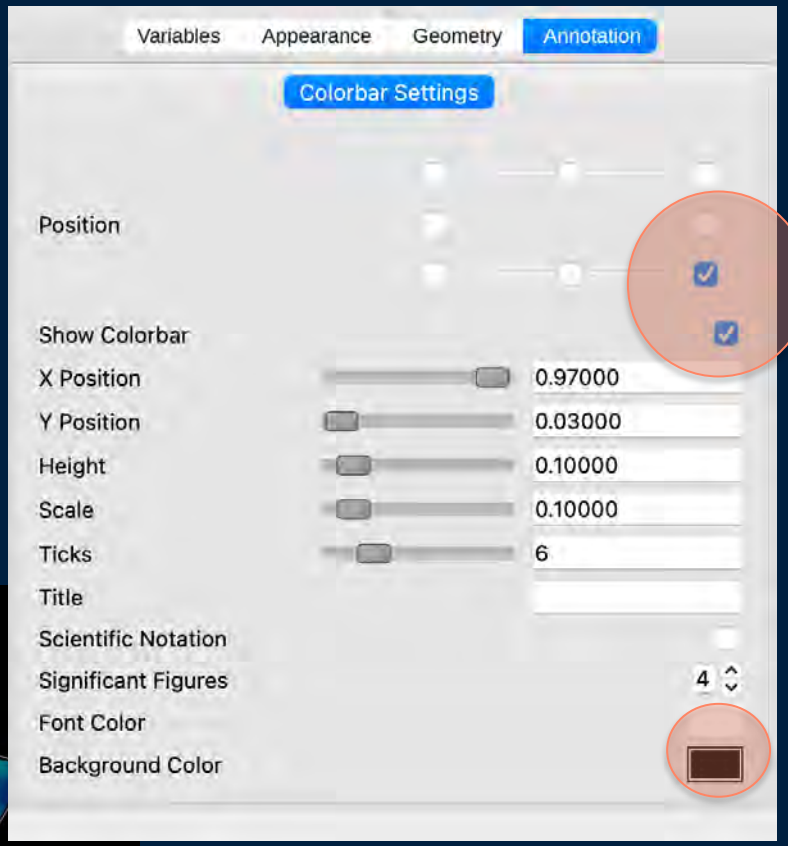
- Let's select a different colormap and adjust it for our purpose
- Click **Settings** – **Load Built-in Colormap** – **Diverging** – select the last one
- Click **Settings** again and select **Reverse Colormap**
- Adjust range: **s**et values to go from **245** to **295**
- Click on the middle **Control Point** and set its value to **273**

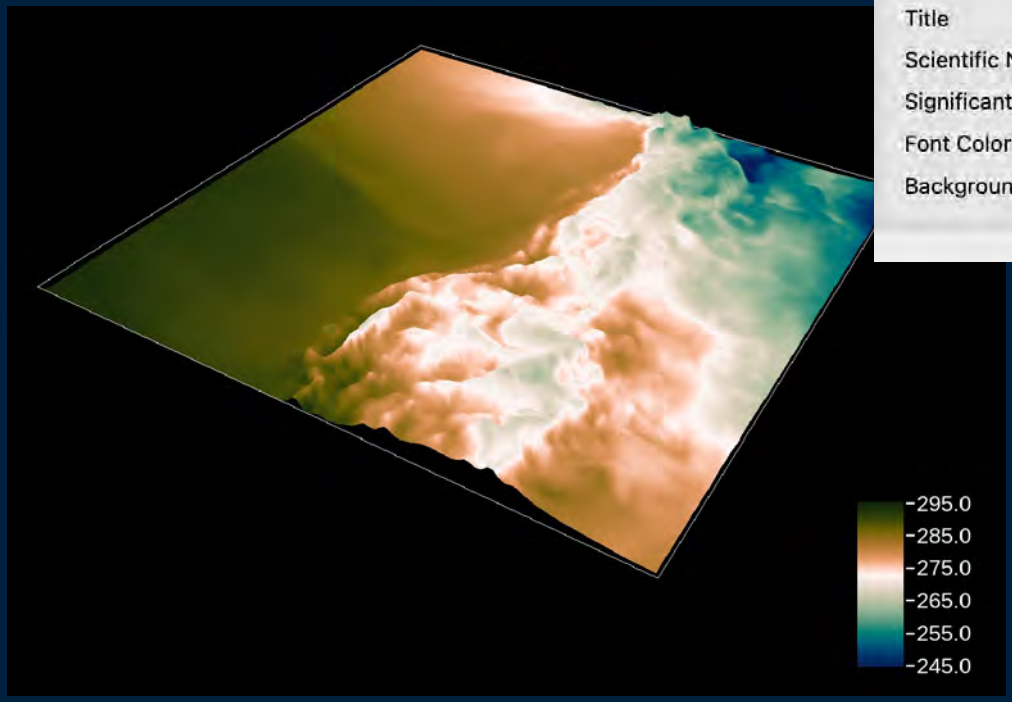




HANDS-ON WORKSHOP: COMMON DISPLAY CONTROLS

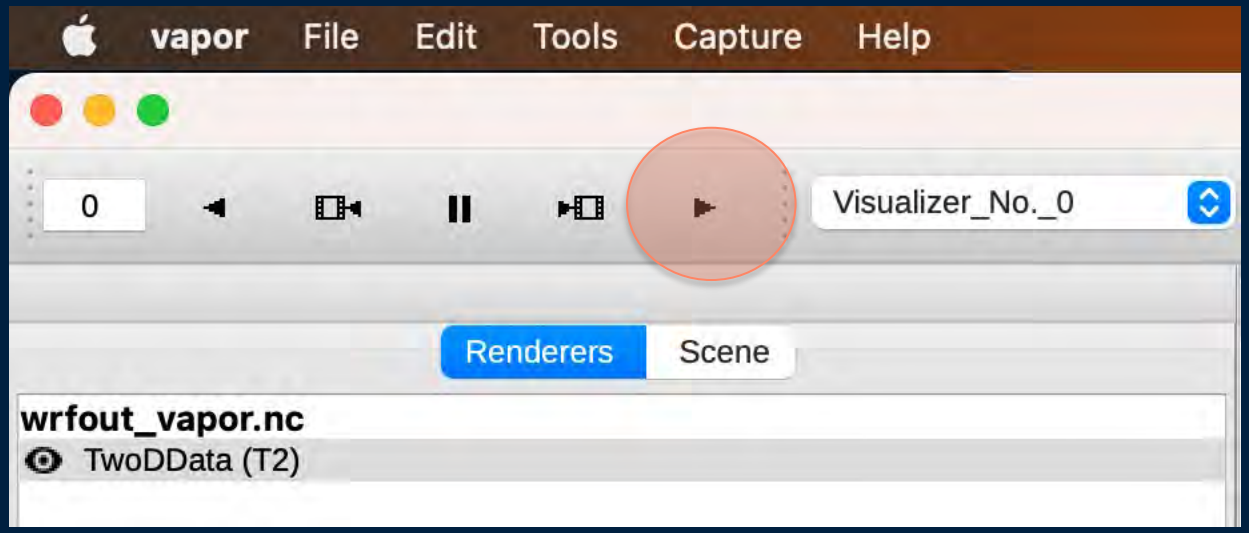
- **Annotate your plot**
  - Select **Annotations**
  - Click **Show Colorbar**
  - Adjust range, position, appearance





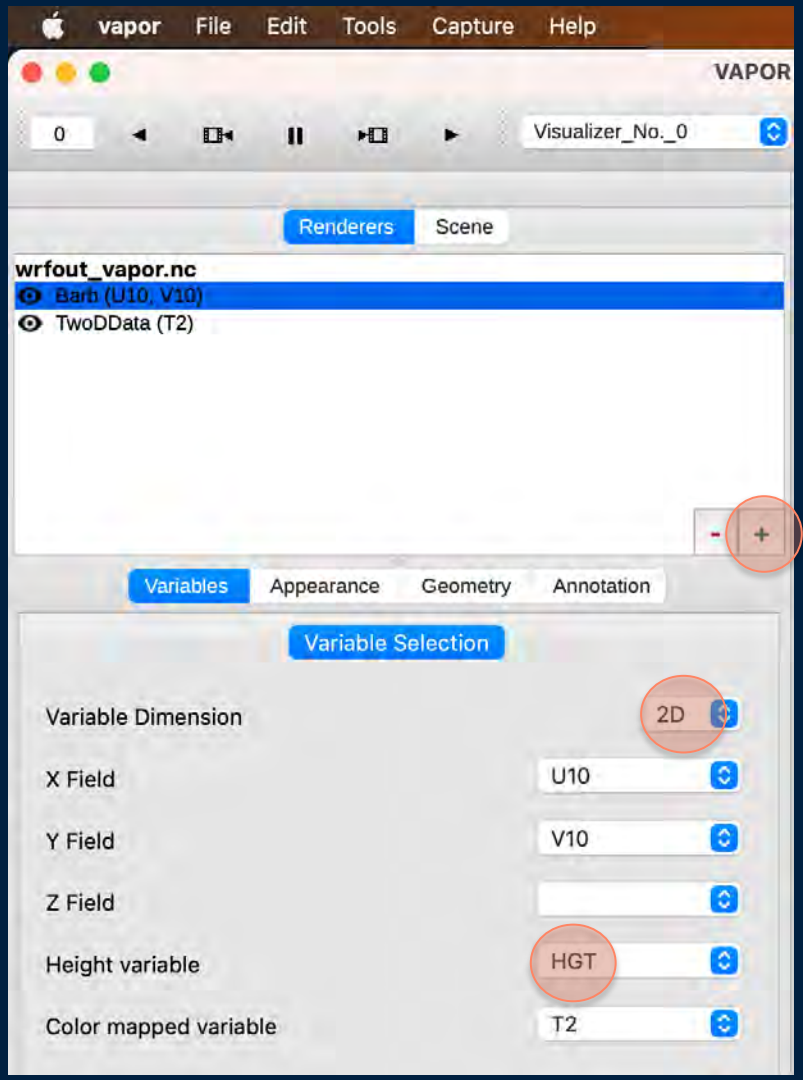
HANDS-ON WORKSHOP: TIME CONTROL

- **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**
- Set **Variable Dimension** to **2D**
- Set **Height Variable** to **HGT**
- Enable viewing

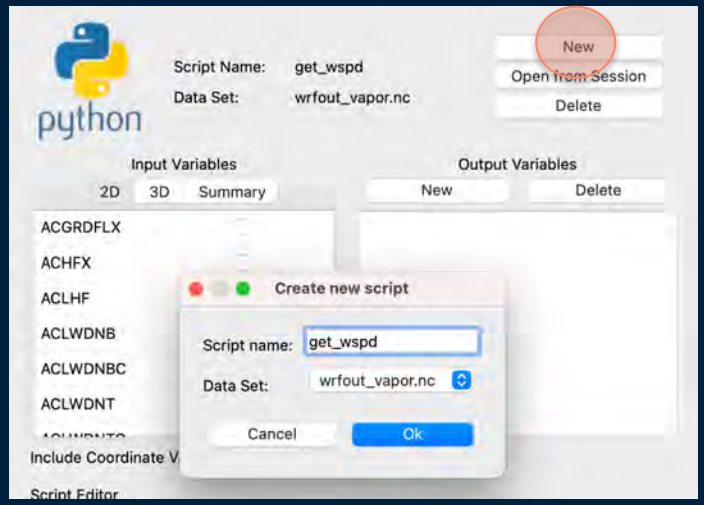


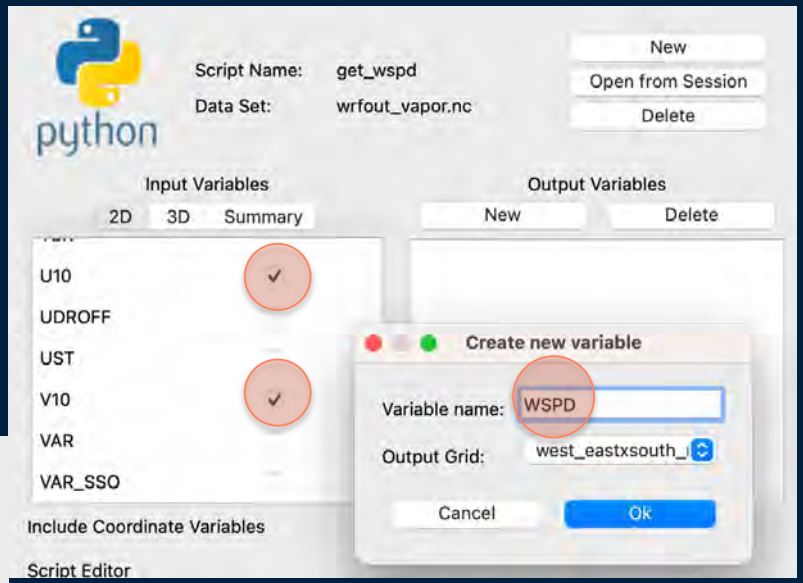
HANDS-ON WORKSHOP: PYTHON VARIABLE EDITOR

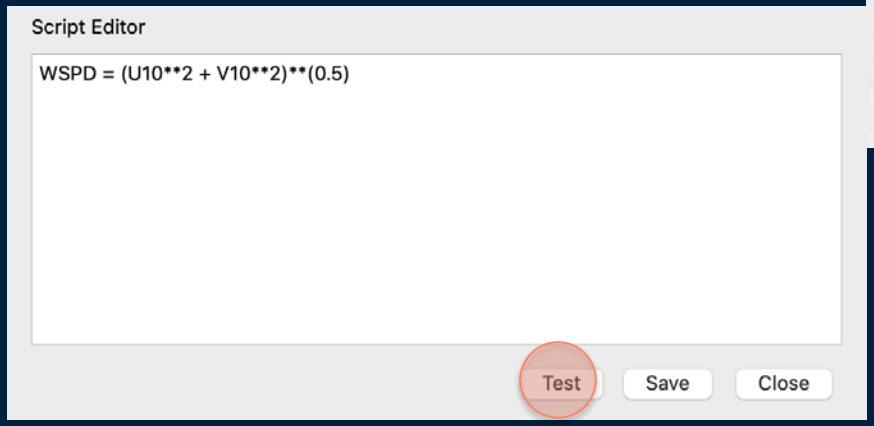
- In **Tools** select **Python Variables**
- Click **New** to create a new script **get_wspd**
- Select **U10** and **V10** as Input Variables
- Create new output variable **WSPD**
- Write Python code in Script Editor:

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

- Click **Test - Save - Close**

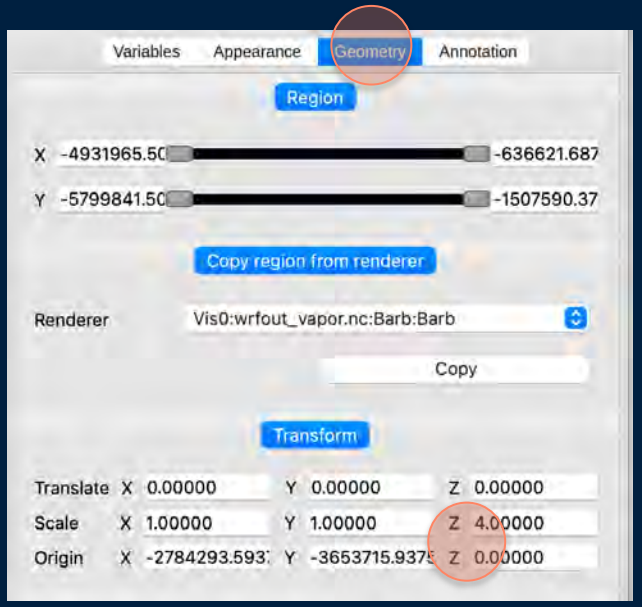






HANDS-ON WORKSHOP: BARB RENDERER

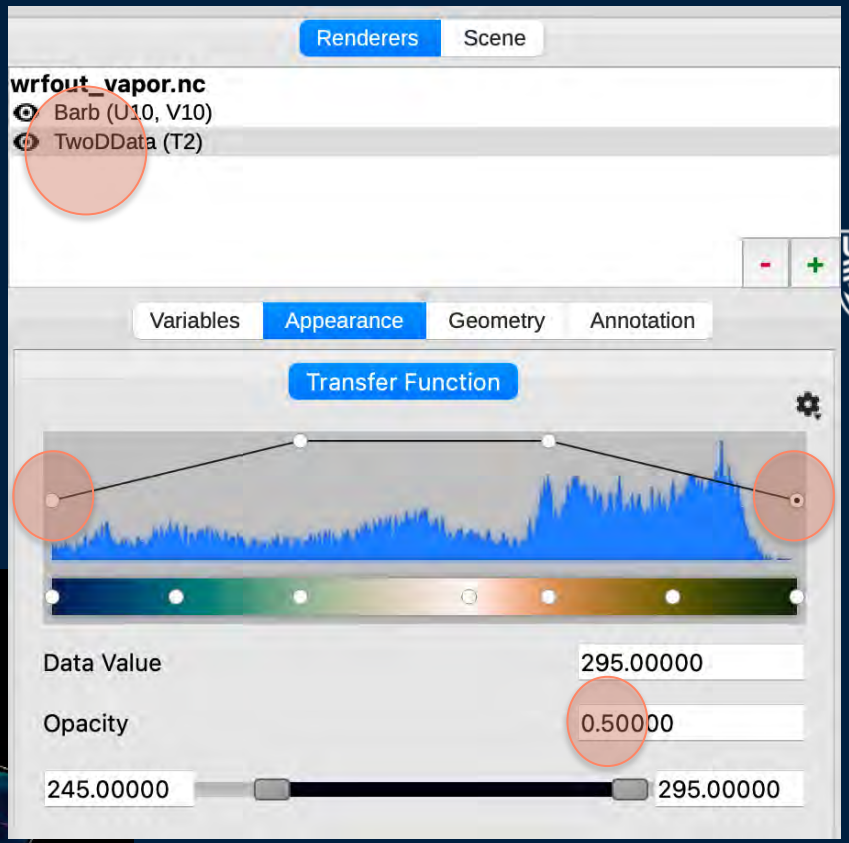
- Under **Variables**
  - change **Color mapped variable** to **WSPD**
- Under **Geometry**
  - Set **Scale (Z)** to **4**
  - Set **Origin (Z)** to **0**
- Under **Appearance**
  - Change colormap to **Sequential – Thermal** (second last option)
  - Change range max to **10**
  - Set **X Barbs** to **20**
  - Set **Y Barbs** to **20**
  - Set **Thickness Scale** to **0.6**
  - Unclick **Use Constant Color**

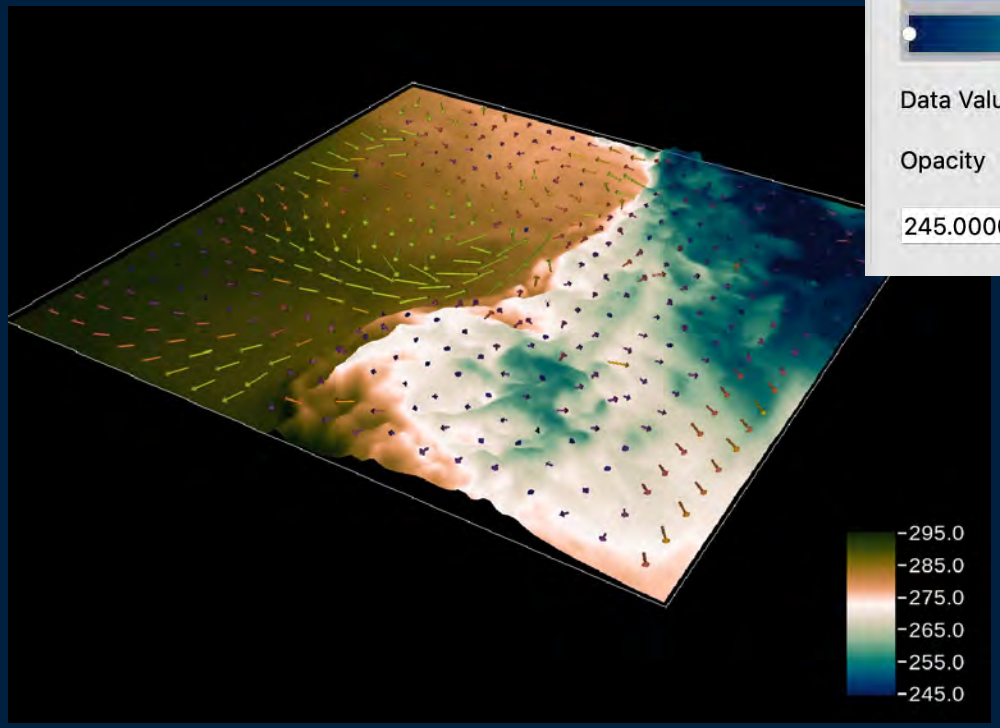


HANDS-ON WORKSHOP: ADJUSTING OPACITY

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

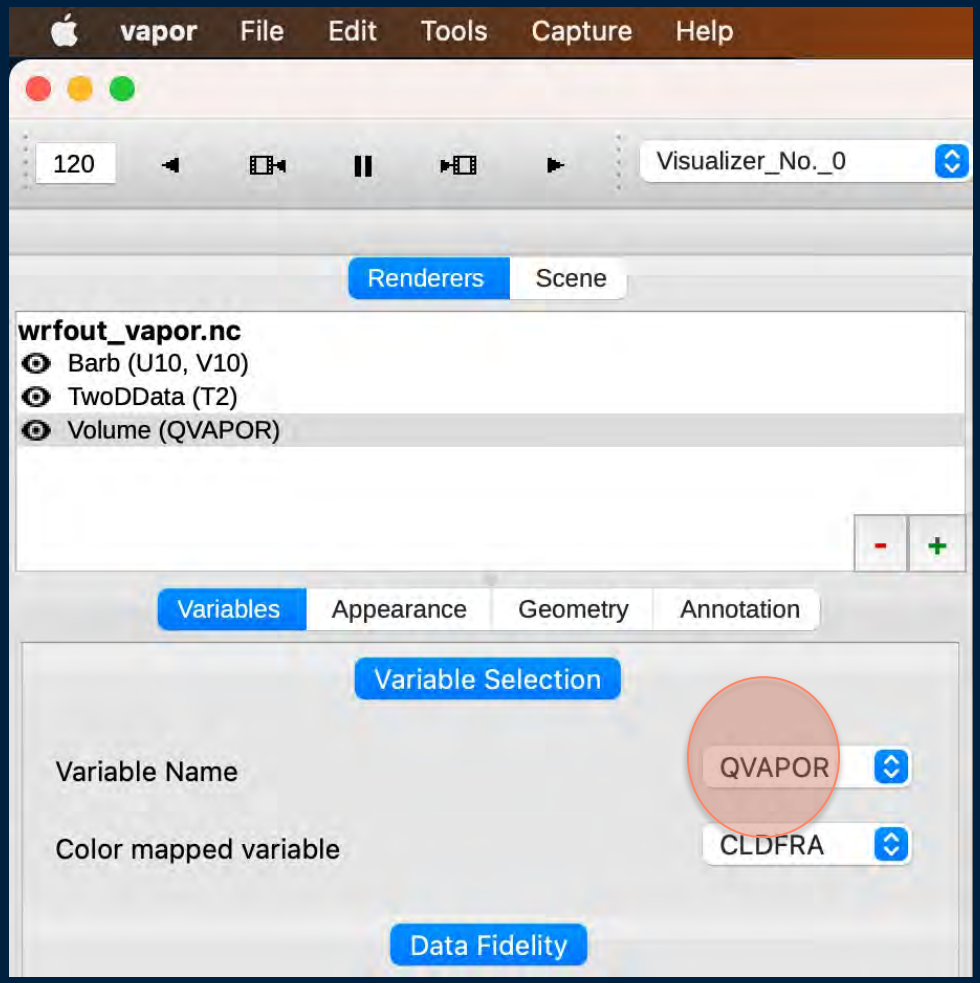
- **Play**





HANDS-ON WORKSHOP: VOLUME RENDERING

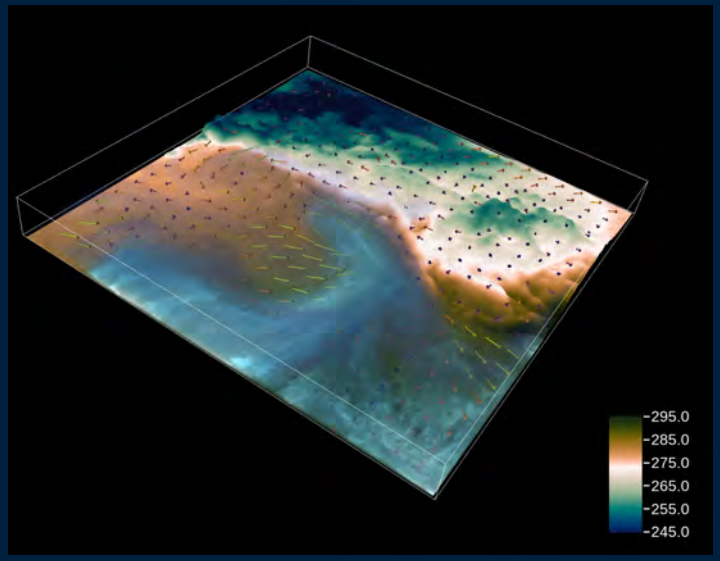
- Create **New** Renderer
- Select **Volume**
- Set **Variable Name** to **QVAPOR**
- Enable viewing

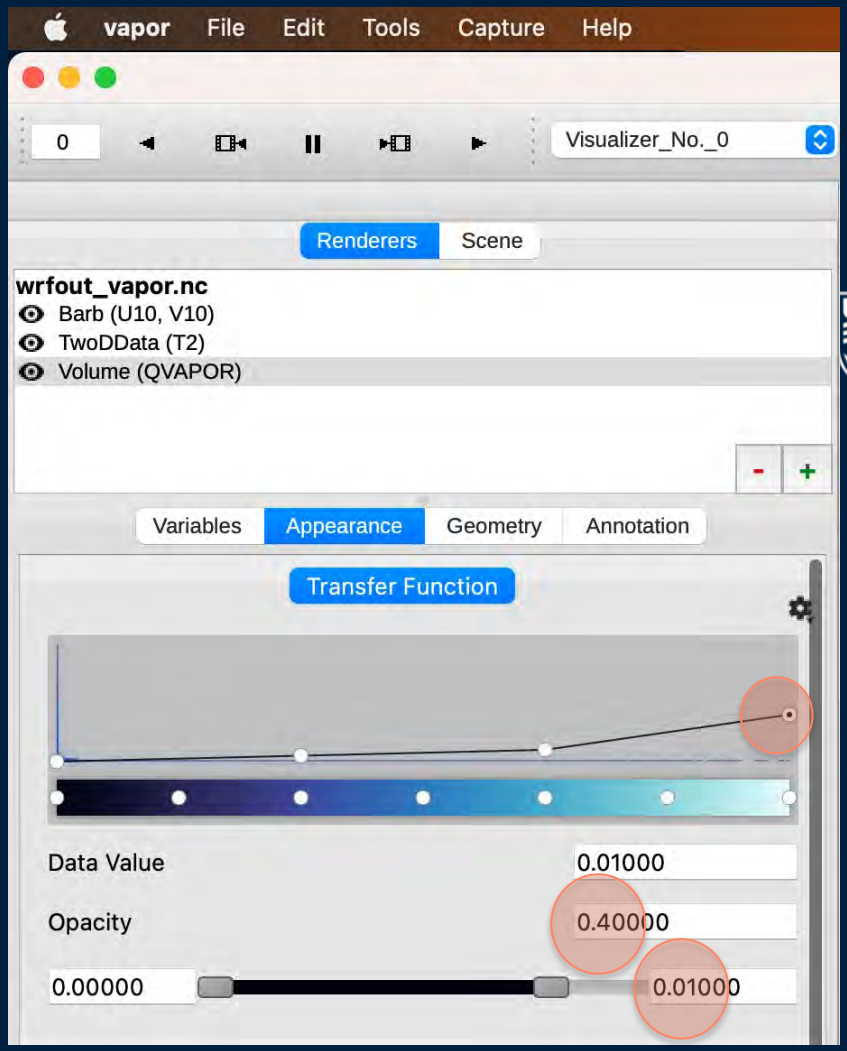


HANDS-ON WORKSHOP: VOLUME RENDERING

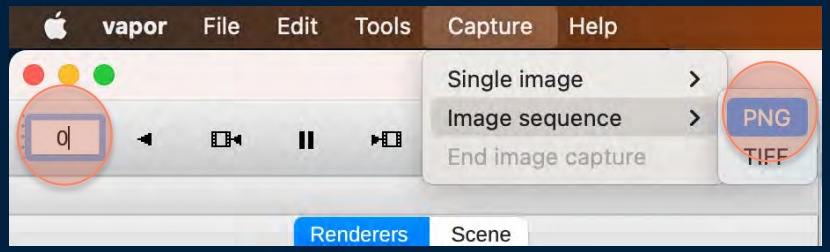
- Under **Geometry**
  - Set **Scale (z)** to **4**

- Under **Appearance**
  - Change colormap to **Sequential – Ice**
  - Set data range from **0** to **0.01**
  - Set Opacity points to **0, 0.05, 0.1, 0.4**
  - Experiment with **Lighting Parameters**





HANDS-ON WORKSHOP: EXPORTING YOUR WORK



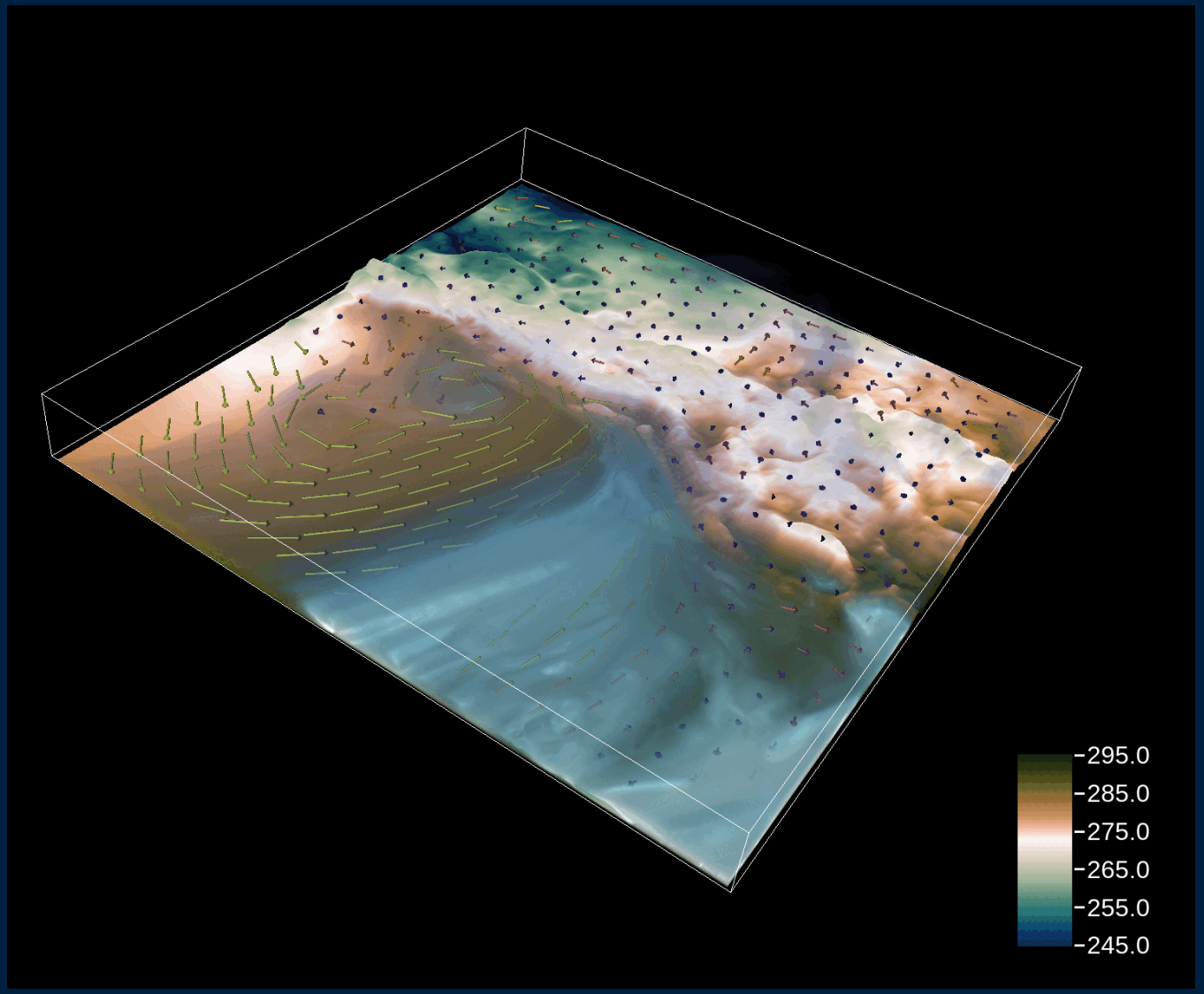
- Set frame number to **0**
- From **Capture – Image Sequence** select **PNG**
- Select save location and click **Play**
- From **Capture** select **End image capture**
- Animate your frames using program of choice, for example:

convert *.png vapordemo.gif

OR

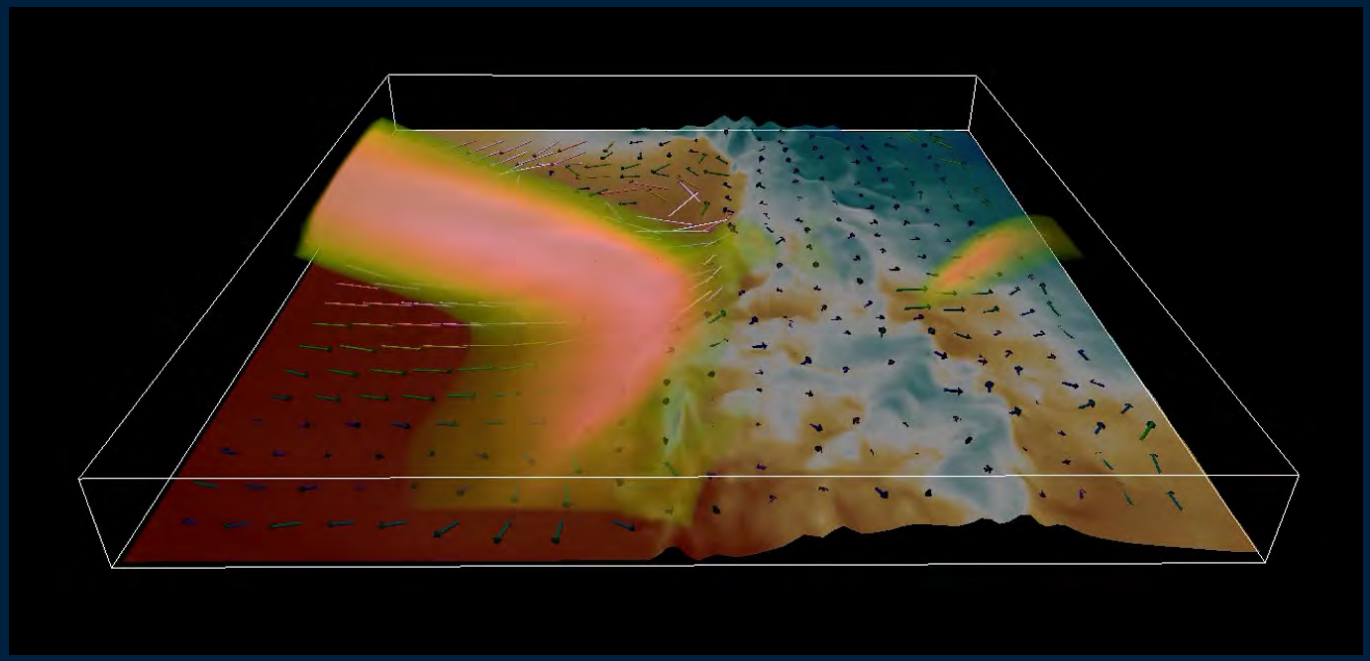
ffmpeg -framerate 15 –I demo%04d.png -vf scale=-2:836 -pix_fmt yuv420p demo.mp4

ENJOY



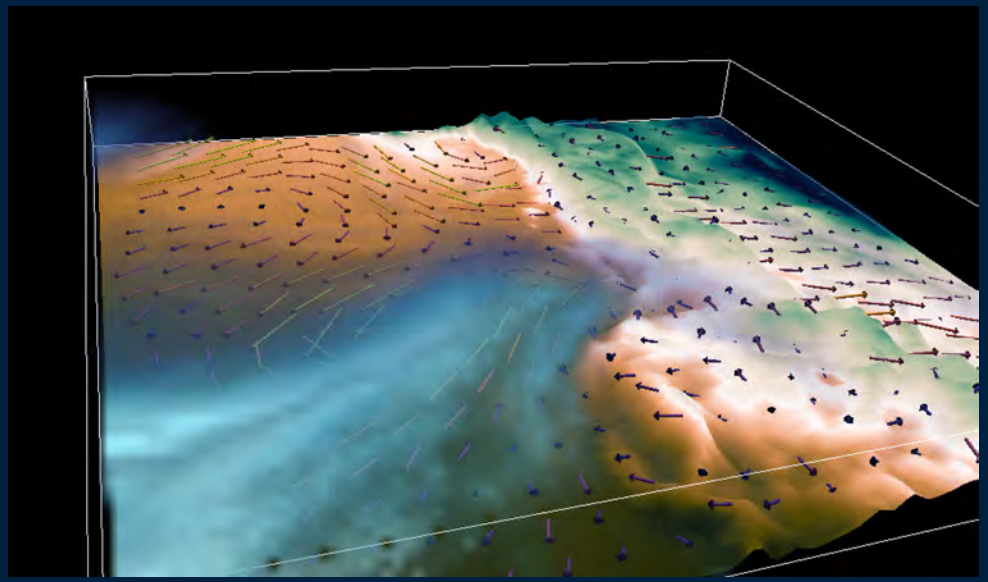
A [HOMEWORK?] CHALLENGE:

- **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
  - Alternatively, test out an Isosurface rendere



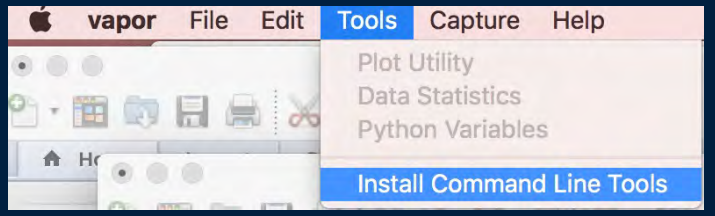
HANDS-ON WORKSHOP:
SUMMARY

- 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**