How to install and use the Python NetCDF Profiles Reader Plugin

Felicia Brisc, CEN/Universität Hamburg

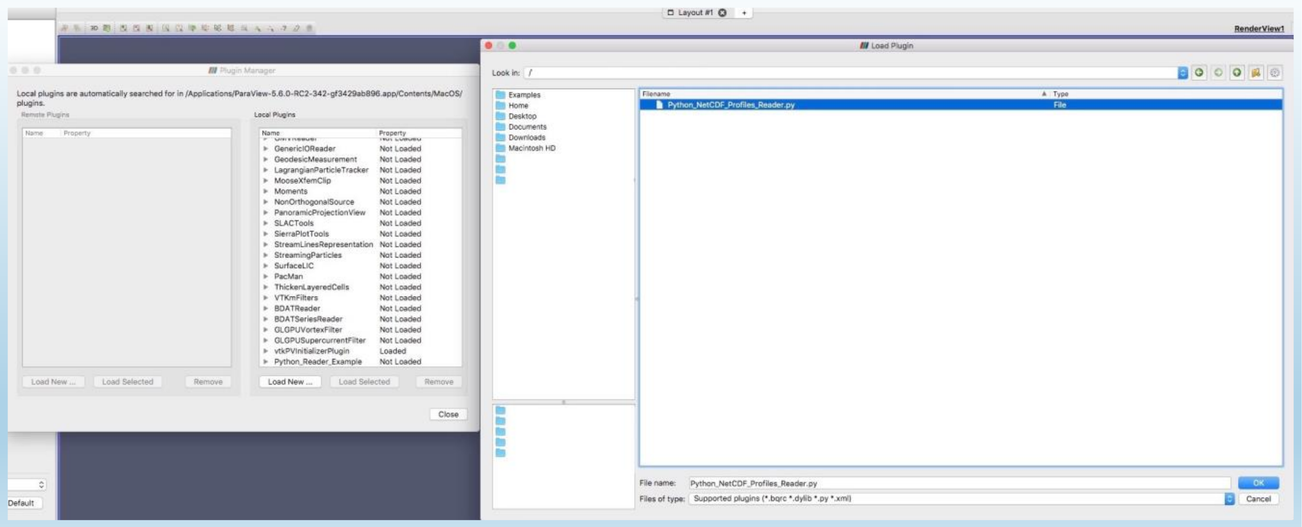
Source code and example NetCDF file can be downloaded at:
https://github.com/FeliciaBrisc/Python-NetCDF-vertical-trajectory-profiles-reader-for-ParaView

A. Installation

A 1. First install the netcdf4-python module, and either make sure its libraries are in the system path or add the libraries to the system path directly in the plugin Python script with sys.path.append(...)

A 2. Copy the **Python_NetCDF_Profiles_Reader.py** in a local folder

A 3. Start ParaView and load the plugin via the Plugin Manager: **Tools -> Manage Plugins -> Load New -> select the Python_NetCDF_Profiles_Reader.py** file in the opening **Load Plugin** window.



Local plugins are automatically searched for in /Applications/ParaView-5.6.0-RC2-342-gf3429ab896.app/Contents/MacOS/plugins.

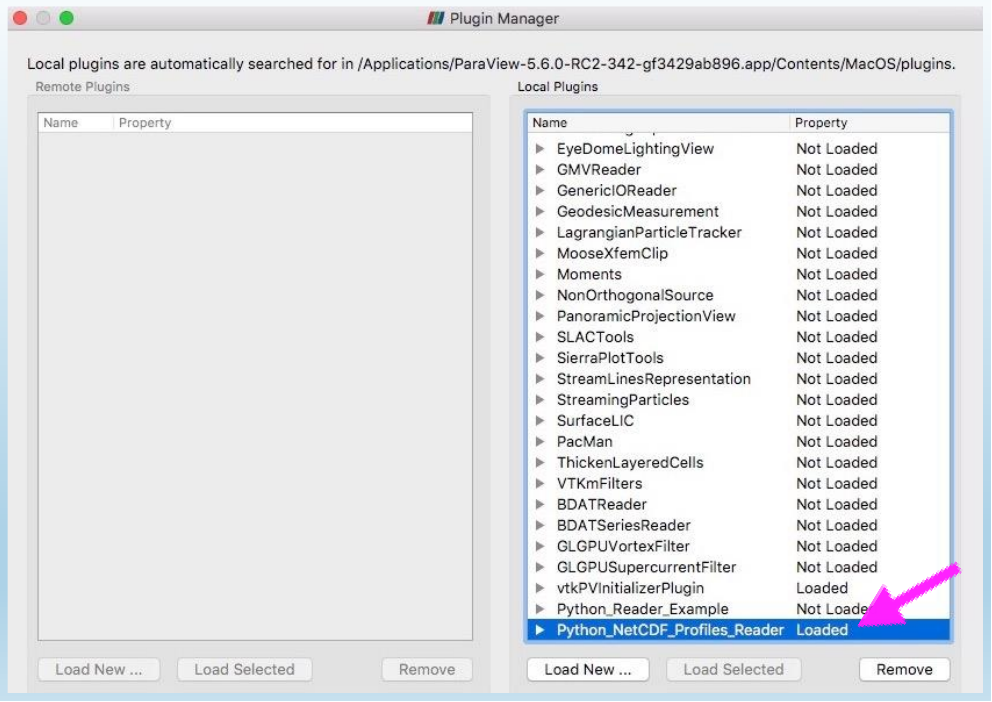
Name	Property
GenericIOReader	Not Loaded
GeodesicMeasurement	Not Loaded
LagrangianParticleTracker	Not Loaded
MooseXfemClip	Not Loaded
Moments	Not Loaded
NonOrthogonalSource	Not Loaded
PanoramicProjectionView	Not Loaded
SLACTools	Not Loaded
SierraPlotTools	Not Loaded
StreamLinesRepresentation	Not Loaded
StreamingParticles	Not Loaded
SurfaceLIC	Not Loaded
PacMan	Not Loaded
ThickenLayeredCells	Not Loaded
VTKmFilters	Not Loaded
BDATReader	Not Loaded
BDATSeriesReader	Not Loaded
GLGPUVortexFilter	Not Loaded
GLGPUSupercurrentFilter	Not Loaded
vtkPVInitializerPlugin	Loaded
Python_Reader_Example	Not Loaded

Look in: /

Filename	Type
Python_NetCDF_Profiles_Reader.py	File

File name: Python_NetCDF_Profiles_Reader.py
Files of type: Supported plugins (*.bqrc *.dylib *.py *.xml)

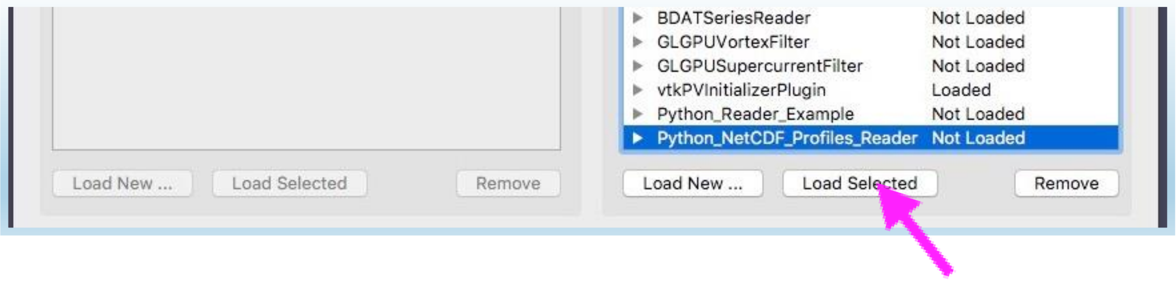
A 4. The plugin will now appear as **Loaded** in the **Plugin Manager**.



Local plugins are automatically searched for in /Applications/ParaView-5.6.0-RC2-342-gf3429ab896.app/Contents/MacOS/plugins.

Name	Property
EyeDomeLightingView	Not Loaded
GMVReader	Not Loaded
GenericIOReader	Not Loaded
GeodesicMeasurement	Not Loaded
LagrangianParticleTracker	Not Loaded
MooseXfemClip	Not Loaded
Moments	Not Loaded
NonOrthogonalSource	Not Loaded
PanoramicProjectionView	Not Loaded
SLACTools	Not Loaded
SierraPlotTools	Not Loaded
StreamLinesRepresentation	Not Loaded
StreamingParticles	Not Loaded
SurfaceLIC	Not Loaded
PacMan	Not Loaded
ThickenLayeredCells	Not Loaded
VTKmFilters	Not Loaded
BDATReader	Not Loaded
BDATSeriesReader	Not Loaded
GLGPUVortexFilter	Not Loaded
GLGPUSupercurrentFilter	Not Loaded
vtkPVInitializerPlugin	Loaded
Python_Reader_Example	Not Loaded
Python_NetCDF_Profiles_Reader	Loaded

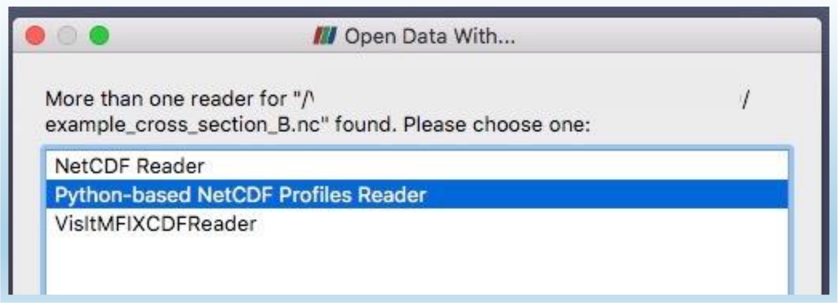
NOTE: Every time you will restart ParaView, the plugin will still be visible in the **Plugin Manager**, but as **Not Loaded**. Therefore first you will need to select the plugin, then press the **Load Selected** button.



▶ BDATSeriesReader	Not Loaded
▶ GLGPUVortexFilter	Not Loaded
▶ GLGPUSupercurrentFilter	Not Loaded
▶ vtkPVInitializerPlugin	Loaded
▶ Python_Reader_Example	Not Loaded
▶ Python_NetCDF_Profiles_Reader	Not Loaded

B. How to use the Plugin

B 1. Open your NetCDF trajectory profile file: **File -> Open -> Select your .nc file**. A window will open listing the available ParaView readers for NetCDF files, select the **Python-based NetCDF Profiles Reader**.

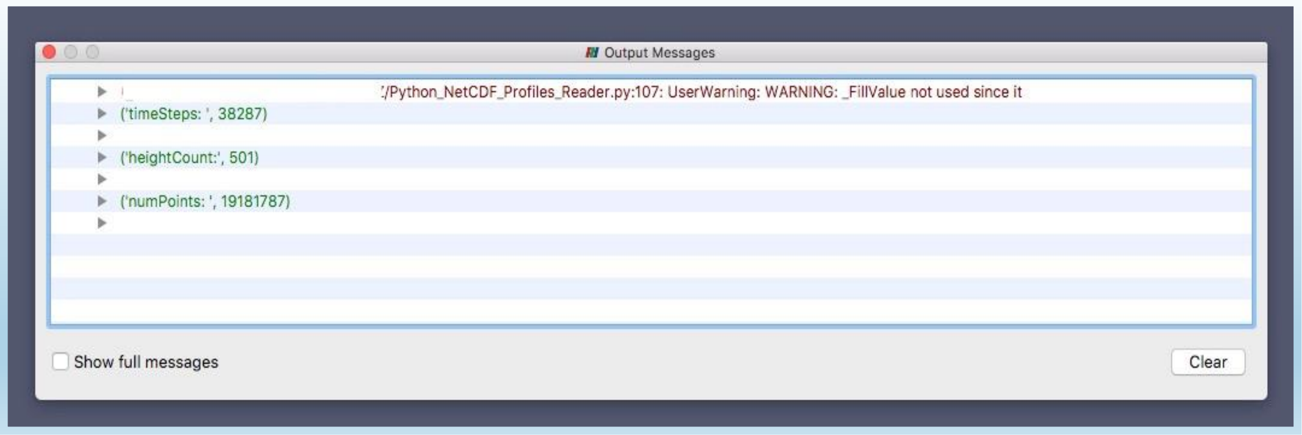


Open Data With...

More than one reader for "/\ /
example_cross_section_B.nc" found. Please choose one:

NetCDF Reader
Python-based NetCDF Profiles Reader
VisItMFIXCDFReader

B 2. With your NetCDF file open, ParaView will most likely open the **Output Messages** window, which displays warnings, errors and other various messages. Drag this window out of the way, do not close it, it will reopen when playing the timesteps.

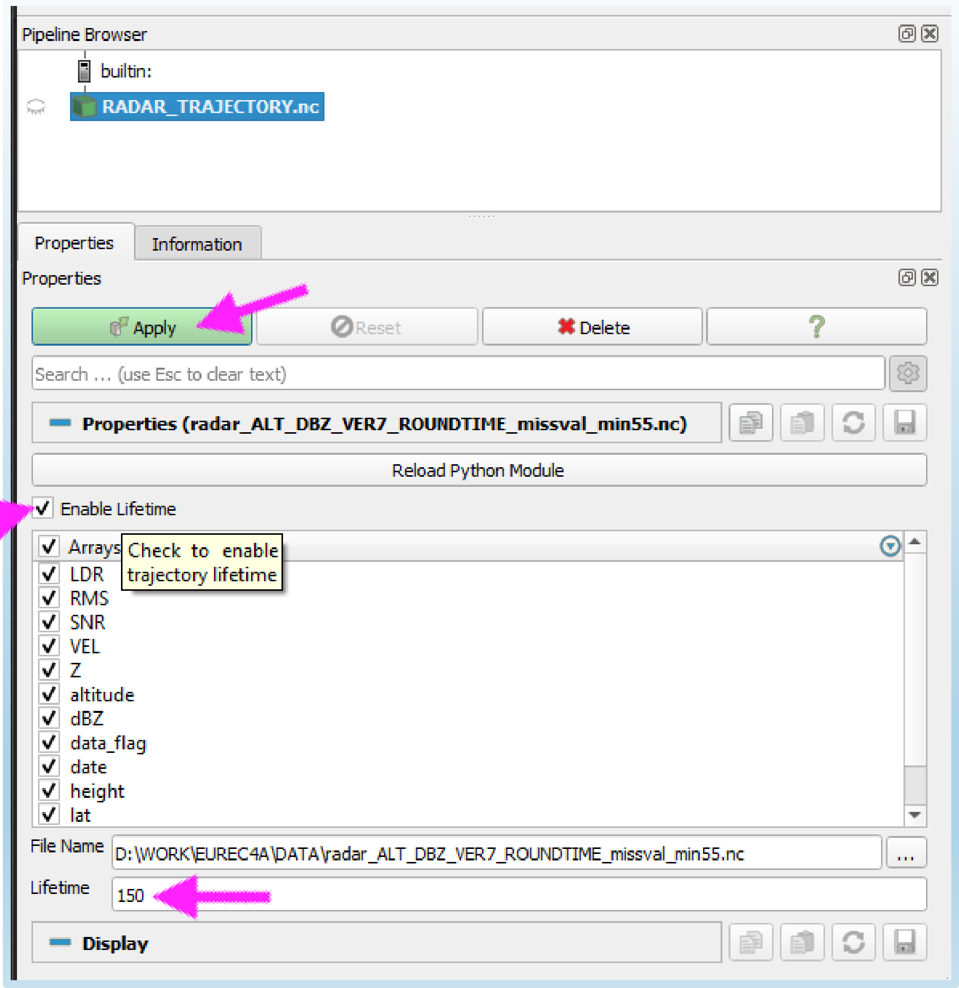


Output Messages

```
▶ '/Python_NetCDF_Profiles_Reader.py:107: UserWarning: WARNING: _FillValue not used since it
▶ ('timeSteps: ', 38287)
▶
▶ ('heightCount:', 501)
▶
▶ ('numPoints: ', 19181787)
▶
```

B 3. In case you would like the trajectory to have a **lifetime** - that is, only a number equal to the desired lifetime of the most recent time steps will be displayed – then write the selected lifetime in the **“Lifetime” text box**, also make sure to check the **“Enable Lifetime” checkbox** and click **“Apply”**. Displaying only recent time steps with the help of the lifetime feature is useful for example when the trajectory has overlapping portions, which would lead otherwise to view self-obstructing.

In the current source code there’s a parameter **`lifetimeMin`** that sets the minimum allowed value for the lifetime (currently 30) – if you wish to allow smaller values, please modify this parameter in the source code.



Pipeline Browser

builtin:

RADAR_TRAJECTORY.nc

Properties | Information

Properties

Apply | Reset | ✖ Delete | ?

Search ... (use Esc to clear text)

Properties (radar_ALT_DBZ_VER7_ROUNDTIME_missval_min55.nc)

Reload Python Module

☑ Enable Lifetime

☑ Arrays

Check to enable trajectory lifetime

☑ LDR
☑ RMS
☑ SNR
☑ VEL
☑ Z
☑ altitude
☑ dBZ
☑ data_flag
☑ date
☑ height
☑ lat

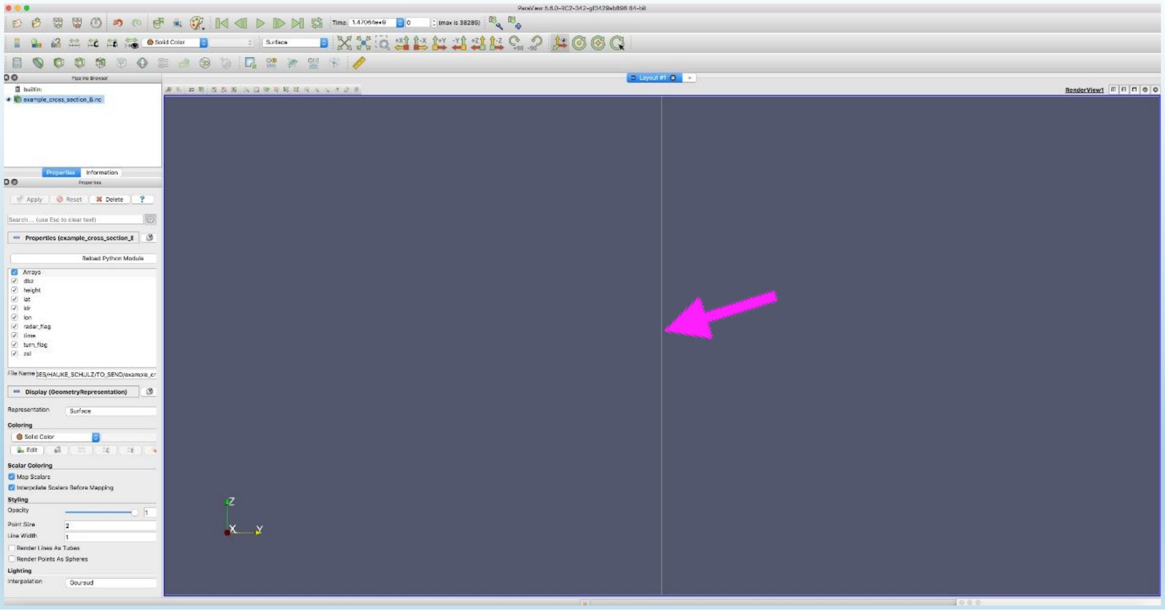
File Name D:\WORK\EUREC4A\DATA\radar_ALT_DBZ_VER7_ROUNDTIME_missval_min55.nc ...

Lifetime 150

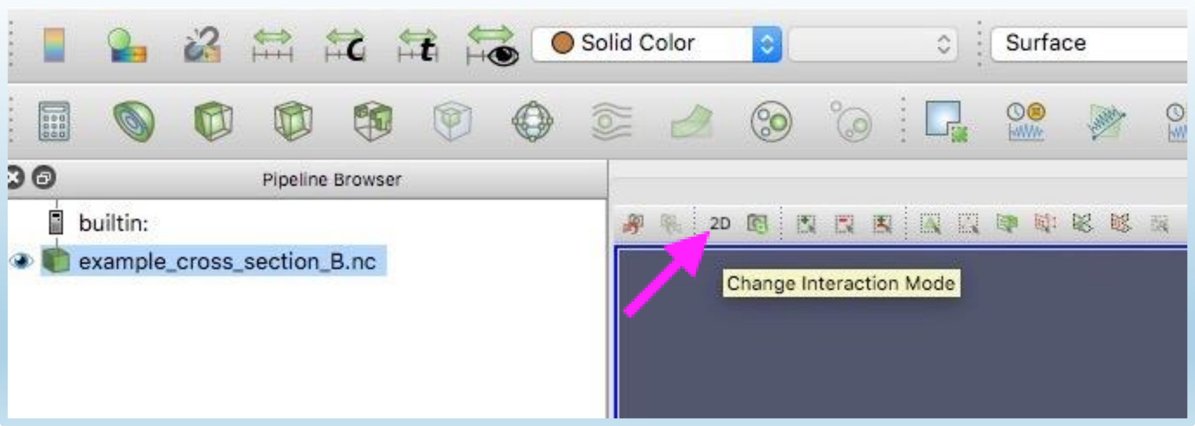
Display

B4. After you click the **Apply** button, the first time step will be displayed in the rendering window.

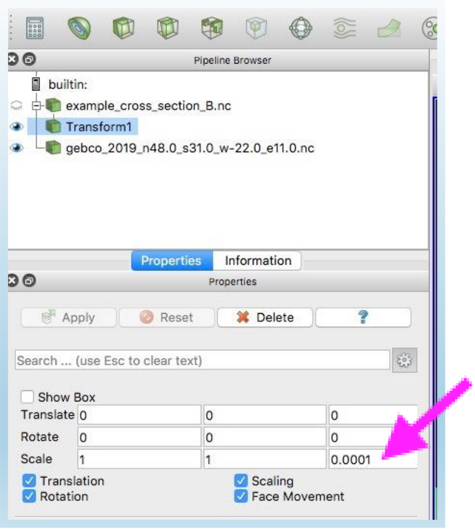
NOTE: In case your ParaView project contains other temporal data sets with various time steps, please make sure to **advance to the first time step of the trajectory data set** in order to initialize it and display it in the rendering window. Coloring the trajectory by a variable is possible **only after the trajectory display was initialized**.



B 5. Make sure you are in 3D mode - if needed, change the interaction mode from 2D to 3D, by clicking the **2D/3D switch button**



B 6. You will likely need to change the vertical scale of the profile. In order to do this, either go directly to the **Transforming** area in the **Properties tab**, or apply a **Transform filter** on your data set, and modify the scale on the Z axis. It might be useful to load in the project for example topology data, which could give you by comparison a better idea on how your data needs to be scaled. You could for example scale by the height dimension relative to the Earth radius, or just by any convenient scale.

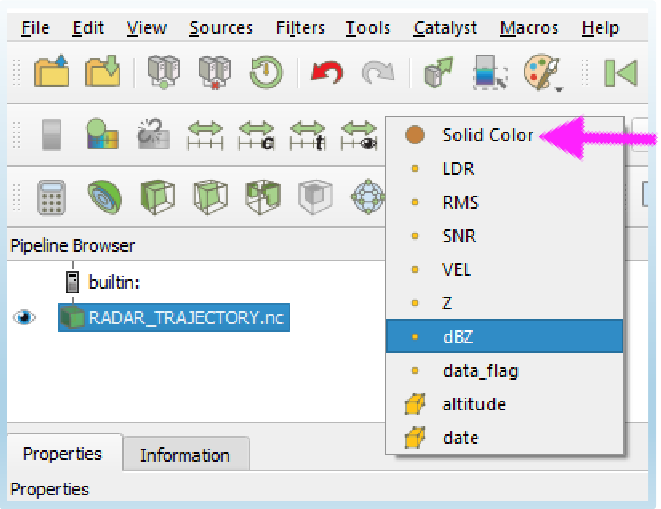

Pipeline Browser
builtin:
example_cross_section_B.nc
Transform1
gebco_2019_n48.0_s31.0_w-22.0_e11.0.nc
Properties
Information
Properties
Apply
Reset
Delete
?
Search ... (use Esc to clear text)

□ Show Box

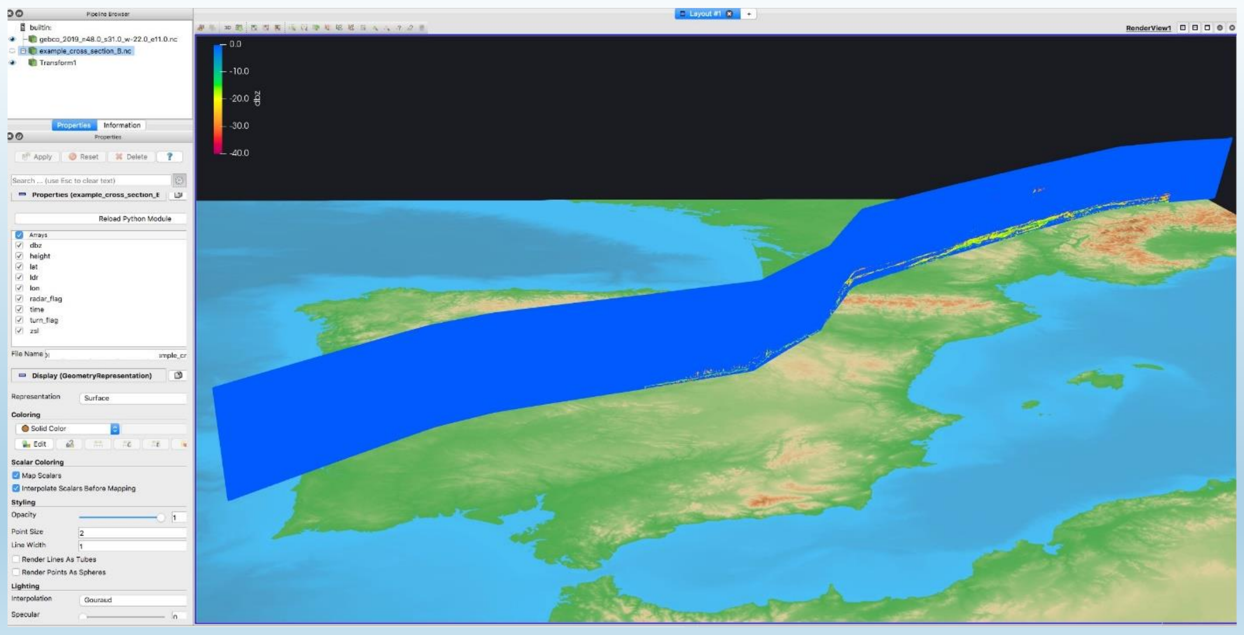
Translate	0	0	0
Rotate	0	0	0
Scale	1	1	0.0001

☑ Translation ☑ Scaling
☑ Rotation ☑ Face Movement

B 7. Before playing any further time steps, please consider again if the lifetime feature needs to be enabled (see §B 3). Enabling the lifetime at a later point might not work well. Change as needed the variable by which the trajectory profile will be colored, then press the **Play** button to advance the time steps.



B 8. Let the time steps play until the last one, or stop the animation at the desired time step. The Reader will be able to further advance the time steps, **but will not be able to go back to previous timesteps (see also § C)**.



builtin:
gebco_2019_n48.0_s31.0_w-22.0_e11.0.nc
example_cross_section_B.nc
Transform1

Search ... (use Esc to clear text)

Properties (example_cross_section_B

Reload Python Module

☑ Arrays
☑ dbz
☑ height
☑ lat
☑ ldr
☑ lon
☑ radar_flag
☑ time
☑ turn_flag
☑ zsl

File Name

Display (GeometryRepresentation)

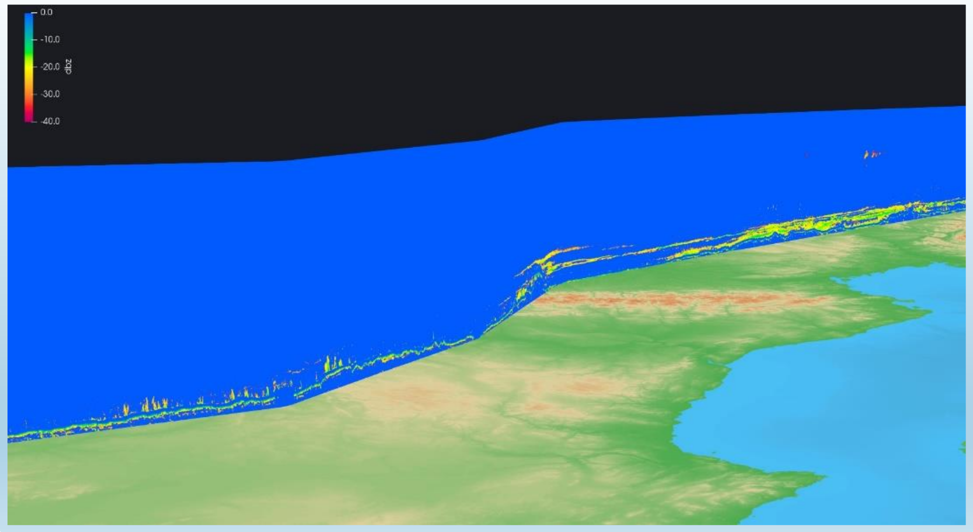
Representation: Surface

Coloring
Solid Color

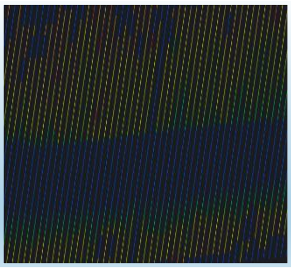
Scalar Coloring
☑ Map Scalars
☑ Interpolate Scalars Before Mapping

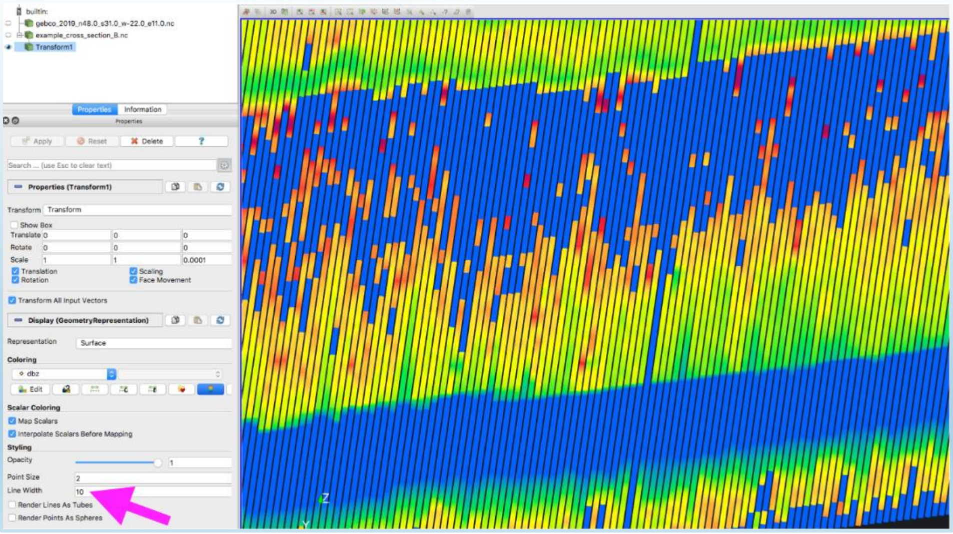
Styling
Opacity: 1
Point Size: 2
Line Width: 1
☐ Render Lines As Tubes
☐ Render Points As Spheres

Lighting
Interpolation: Gouraud
Specular: 0



B 9. The output of the Plugin is a **polyline**, which is displayed by ParaView at the default thickness of 1. Hence, depending on the density of your grid and the zoom-in level in the 3D rendering window, there will be visible gaps between consecutive time steps. These gaps can be diminished by increasing the **Line Width** in the **Properties tab**.





builtin:
gebco_2019_n48.0_s31.0_w-22.0_e11.0.nc
example_cross_section_B.nc
Transform1

Search ... (use Esc to clear text)

Properties (Transform1)

Transform: Transform
☐ Show Box
Translate 0 0 0
Rotate 0 0 0
Scale 1 1 0.0001
☑ Translation ☑ Scaling
☑ Rotation ☑ Face Movement
☑ Transform All Input Vectors

Display (GeometryRepresentation)

Representation: Surface

Coloring
dbz

Scalar Coloring
☑ Map Scalars
☑ Interpolate Scalars Before Mapping

Styling
Opacity: 1
Point Size: 2
Line Width: 10
☐ Render Lines As Tubes
☐ Render Points As Spheres

C. Important notes: current restrictions and limitations of the Plugin

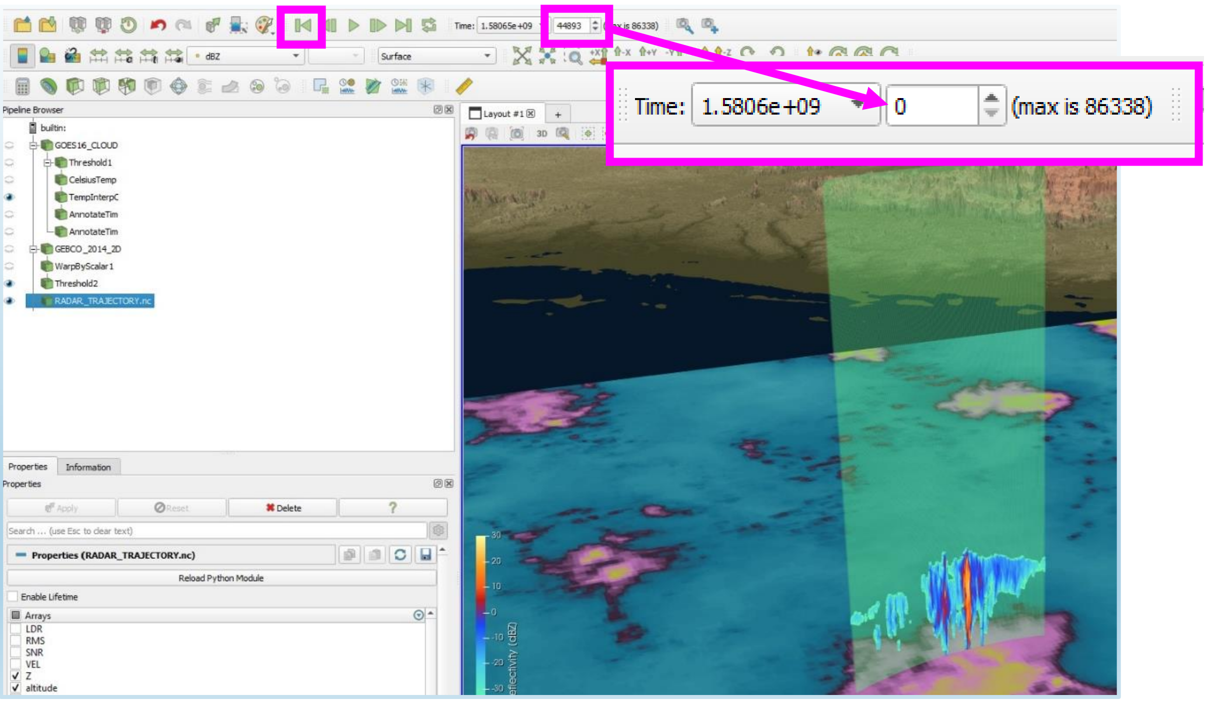
C 1. The **time, height, latitude and longitude** dimensions are mandatory.

C 2. Users are able to **initialize** the trajectory from an arbitrary time step and advance the time steps in **increments of 1** with the help of either the **“Play/Pause” button** or the **“Next Frame” button**. **Any other subsequent time jumps will not work** (see § C3).

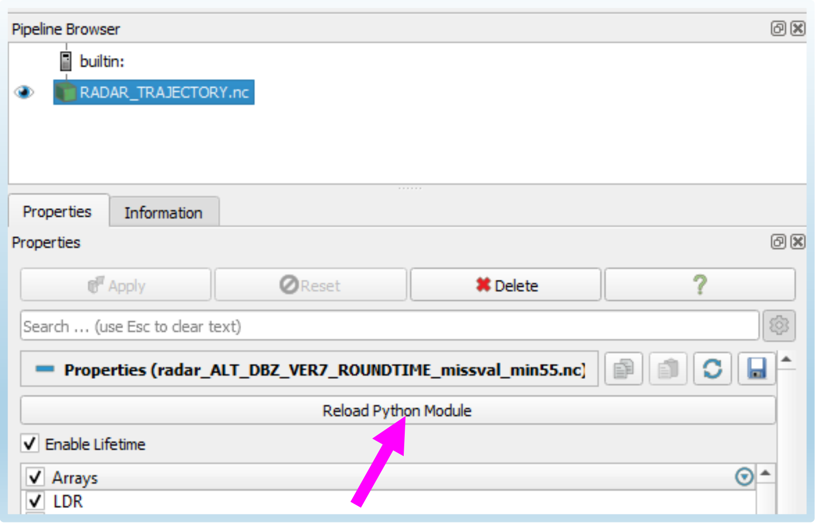
However, it is possible to **initialize the trajectory again, starting from an arbitrary time step**. This will enable users to easily examine time intervals of interest. Two steps are necessary to re-initialize:

1. Animation time line reset: the time steps have to be set back to 0. Either rewind directly to the first frame, or write in the animation time field “0”.

2.Python module reset: select the trajectory data in the Pipeline Browser and press the **“Reload Python Module” button**



Time: 1.5806e+09 0 (max is 86338)



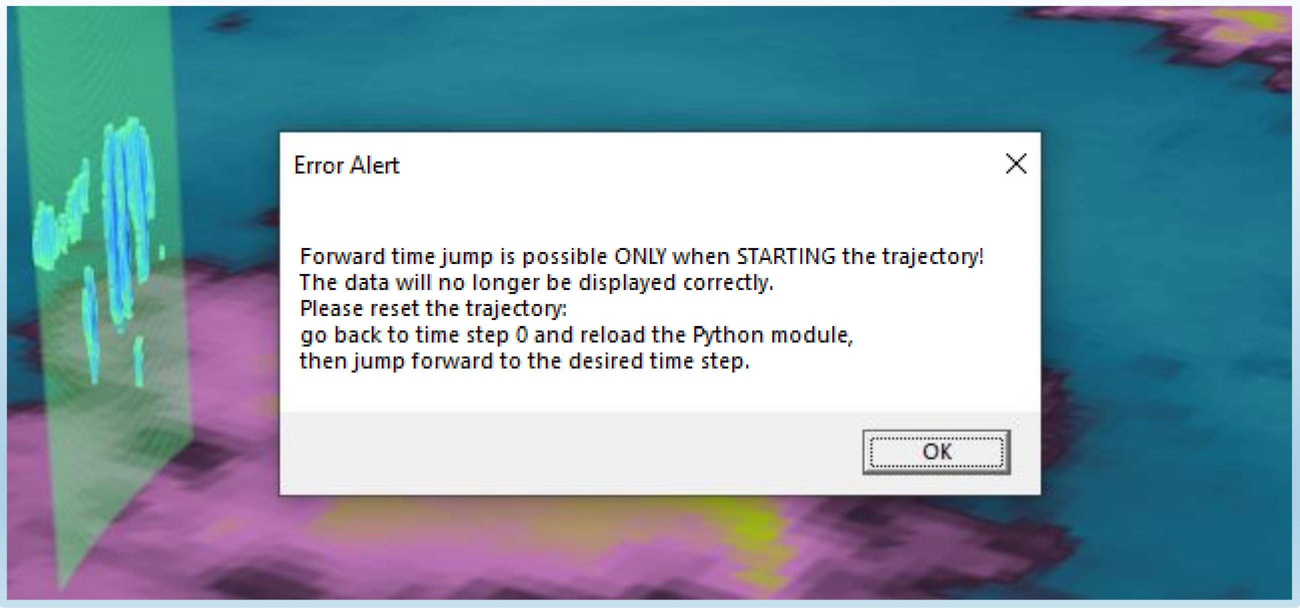
☑ Enable Lifetime

☑ Arrays

☑ LDR

After executing these two steps, which will delete any current trajectory polyline, the trajectory can be initialized again from the desired time step.

C 3. The Plugin initializes the trajectory by reading the current animation time, then the corresponding NetCDF variables values and creating the first cell of the polyline. When the time steps are advancing, the Plugin adds incrementally, step-by-step, a new cell to the existing polyline. Hence, **the Plugin will only work when the animation is played forward by time steps increments of 1**. Playing the animation backward or with time jumps will not work and an error message box will be displayed:

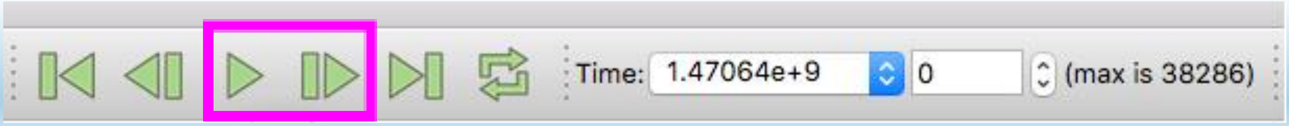


Error Alert

Forward time jump is possible ONLY when STARTING the trajectory!
The data will no longer be displayed correctly.
Please reset the trajectory:
go back to time step 0 and reload the Python module,
then jump forward to the desired time step.

This means that users should keep in mind a few things:

- The only animation controls users should use during viewing their data are the **“Play/Pause” button** and the **“Next Frame” button**.



Time: 1.47064e+9 0 (max is 38286)

- When saving a project before closing it, **first rewind the time steps back to 0 or to the desired initial time step**, **then save your project file**. This way, when you reopen the project, the first/desired time step will be read and you can play again the time steps.

- If you want to stop your animation at a certain time step, it is recommended to consider the response latency when hitting the play buttons. It is good practice **to pause the animation a few time steps earlier**, and then advance with the **“Next Frame”** button to the desired time step.