PyVista
Visualizing CAE Results with Python

Alex Kaszynski

November 5, 2022

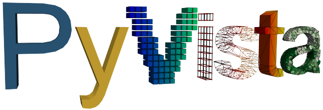
Ansys

Table of Contents

1. PyVista - Introduction
    - Where is it already used?
    - Quick Example
    - Comparison - VTK vs. PyVista
2. Getting Started
    - Installation
3. Examples
    - Basic Plot
    - Basic Volumetric Plot
    - Filters
    - PyInstaller and PyQT
    - Tutorial

PyVista - Introduction

```
>>> from pyvista import demos
>>> demos.plot_logo(
...     window_size=(1920, 1080), off_screen=False,
... )
```



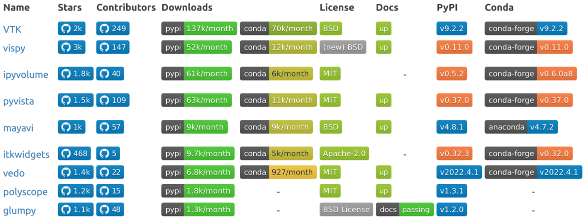
PyVista allows you to rapidly load meshes and handles much of the “grunt work” of setting up plots, connecting classes and pipelines, and cleaning up plotting windows.

PyVista allows you to:

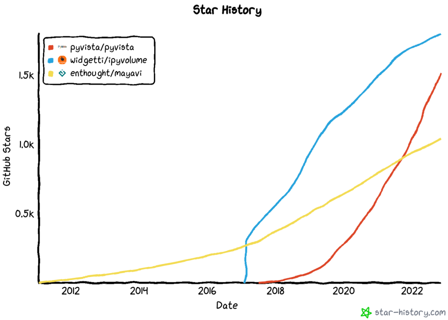
- Easily load a wide variety of datasets and file types.
- Leverage powerful VTK filters and perform complex data operations.
- Quickly set up simple or complex plots.

PyVista - Popularity and Growth

- Already the most popular 3d visualization library on PyPI.
- Designed not just for visualization, but for scientific visualization focused on data post-processing, file IO, and interoperability with other libraries.



Name	Stars	Contributors	Downloads		License	Docs	PyPI	Conda
VTK	2k	249	pypi 137k/month	conda 70k/month	BSD	up	v9.2.2	conda-forge v9.2.2
vispy	3k	147	pypi 52k/month	conda 12k/month	(new) BSD	up	v0.11.0	conda-forge v0.11.0
ipyvolume	1.8k	40	pypi 61k/month	conda 6k/month	MIT	-	v0.5.2	conda-forge v0.6.0a8
pyvista	1.5k	109	pypi 63k/month	conda 11k/month	MIT	up	v0.37.0	conda-forge v0.37.0
mayavi	1k	57	pypi 9k/month	conda 9k/month	BSD	up	v4.8.1	anaconda v4.7.2
itkwidgets	468	5	pypi 9.7k/month	conda 5k/month	Apache-2.0	-	v0.32.3	conda-forge v0.32.0
vedo	1.4k	22	pypi 6.8k/month	conda 927/month	MIT	up	v2022.4.1	conda-forge v2022.4.1
polyscope	1.2k	15	pypi 1.8k/month	-	MIT	up	v1.3.1	-
glumpy	1.1k	48	pypi 1.3k/month	-	BSD License	docs passing	v1.2.0	-

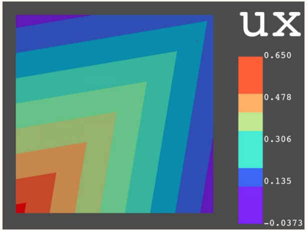


PyVista - Current Usage

PyVista is already being used by:

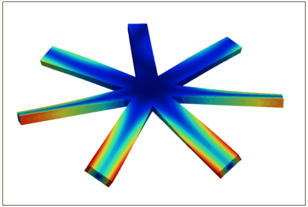
ACE & Partners

```
p = pv.Plotter()
p.add_mesh(
    copygrid,
    scalars='ux',
    n_colors=9
)
p.camera_position='xy'
p.show()
```



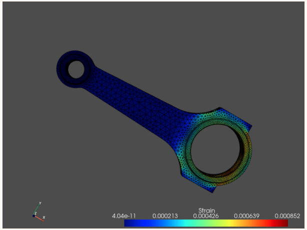
PyAnsys

```
result.animate_nodal_displacement(
    36,
    displacement_factor=2e-4,
    loop=False,
    add_text=False,
    show_scalar_bar=False,
    cmap="jet",
)
```

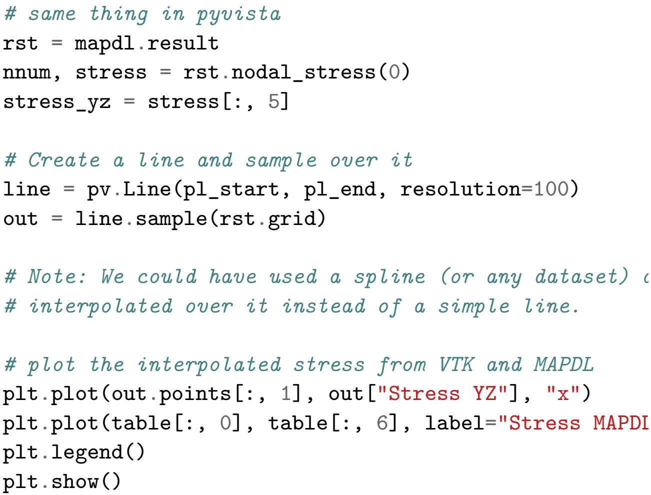


OnScale

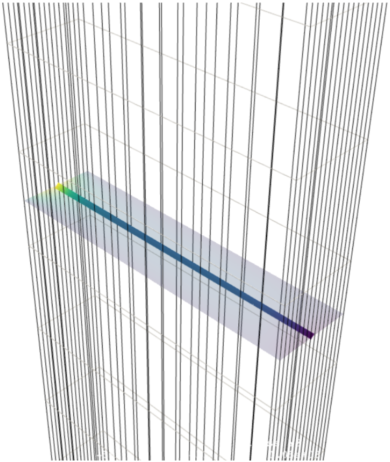
```
import pyvista

# read and plot a result
result = read('result.vtu')
result.plot(
    scalars='strain',
    cmap='jet',
)
```



Quick Example - Path Operation

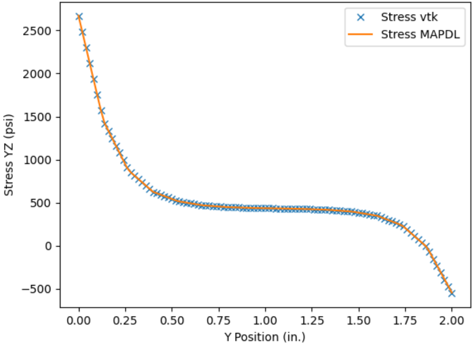


```
# same thing in pyvista
rst = mapdl.result
nnum, stress = rst.nodal_stress(0)
stress_yz = stress[:, 5]

# Create a line and sample over it
line = pv.Line(pl_start, pl_end, resolution=100)
out = line.sample(rst.grid)

# Note: We could have used a spline (or any dataset) a
# interpolated over it instead of a simple line.

# plot the interpolated stress from VTK and MAPDL
plt.plot(out.points[:, 1], out["Stress YZ"], "x")
plt.plot(table[:, 0], table[:, 6], label="Stress MAPDL
plt.legend()
plt.show()
```



Quick Example - Path Operation vs PyMAPDL

```python
# mapping components of interest to path.
mapdl.pdef("Sx_my_path", "s", "x")
mapdl.pdef("Sy_my_path", "s", "y")
mapdl.pdef("Sz_my_path", "s", "z")
mapdl.pdef("Sxy_my_path", "s", "xy")
mapdl.pdef("Syz_my_path", "s", "yz")
mapdl.pdef("Szx_my_path", "s", "xz")

path_out = mapdl.prpath(
    "Sx_my_path",
    "Sy_my_path",
    "Sz_my_path",
    "Sxy_my_path",
    "Syz_my_path",
    "Szx_my_path",
)
table = np.genfromtxt(path_out.splitlines()[1:])
```



Comparison - VTK vs. PyVista

```python
import vtk
reader = vtk.vtkSTLReader()
reader.SetFileName("bunny.stl")
mapper = vtk.vtkPolyDataMapper()
output_port = reader.GetOutputPort()
mapper.SetInputConnection(output_port)
actor = vtk.vtkActor()
actor.SetMapper(mapper)
ren = vtk.vtkRenderer()
renWin = vtk.vtkRenderWindow()
renWin.AddRenderer(ren)
iren = vtk.vtkRenderWindowInteractor()
iren.SetRenderWindow(renWin)
ren.AddActor(actor)
iren.Initialize()
renWin.Render()
iren.Start()
```

```python
from pyvista import examples
mesh = examples.download_bunny()
mesh.plot(cpos='xy')
```

Getting Started

Getting Started

1. PyVista - Introduction
    - Where is it already used?
    - Quick Example
    - Comparison - VTK vs. PyVista
2. Getting Started
    - Installation
3. Examples
    - Basic Plot
    - Basic Volumetric Plot
    - Filters
    - PyInstaller and PyQT
    - Tutorial

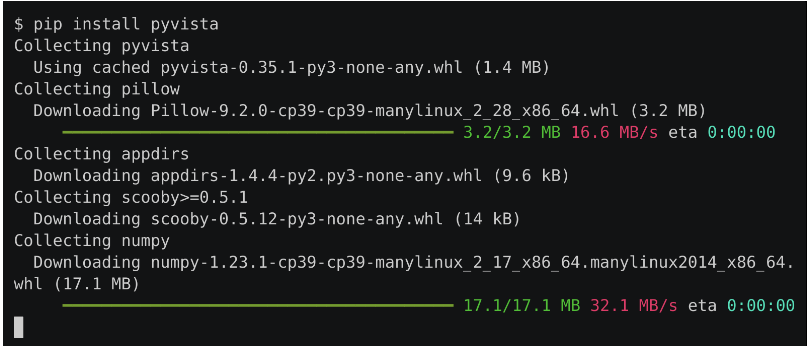
Installation

pip

```
pip install pyvista
```

conda

```
conda install -c conda-forge pyvista
```



```
$ pip install pyvista
Collecting pyvista
  Using cached pyvista-0.35.1-py3-none-any.whl (1.4 MB)
Collecting pillow
  Downloading Pillow-9.2.0-cp39-cp39-manylinux_2_28_x86_64.whl (3.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.2/3.2 MB 16.6 MB/s eta 0:00:00
Collecting appdirs
  Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Collecting scooby>=0.5.1
  Downloading scooby-0.5.12-py3-none-any.whl (14 kB)
Collecting numpy
  Downloading numpy-1.23.1-cp39-cp39-manylinux_2_17_x86_64.manylinux2014_x86_64.
whl (17.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 17.1/17.1 MB 32.1 MB/s eta 0:00:00
```

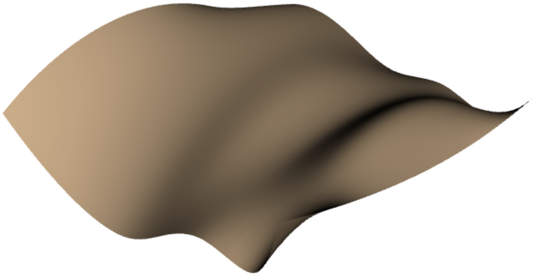
Examples

Examples

1. PyVista - Introduction
   - Where is it already used?
   - Quick Example
   - Comparison - VTK vs. PyVista
2. Getting Started
   - Installation
3. Examples
   - Basic Plot
   - Basic Volumetric Plot
   - Filters
   - PyInstaller and PyQT
   - Tutorial

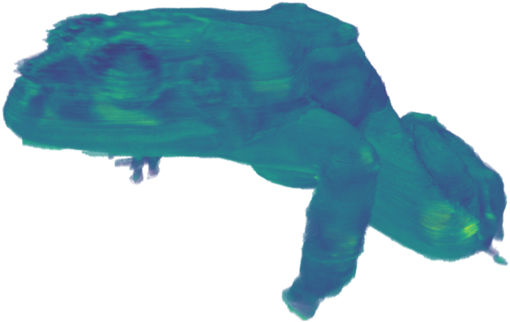
Examples - Basic Plot

```
>>> from pyvista import examples
>>> dataset = examples.download_saddle_surface()
>>> dataset
PolyData (0x7f4d81806c40)
  N Cells:      5131
  N Points:     2669
  N Strips:     0
  X Bounds:     -2.001e+01, 2.000e+01
  Y Bounds:     -6.480e-01, 4.024e+01
  Z Bounds:     -6.093e-01, 1.513e+01
  N Arrays:     0
>>> dataset.plot(color='tan')
```



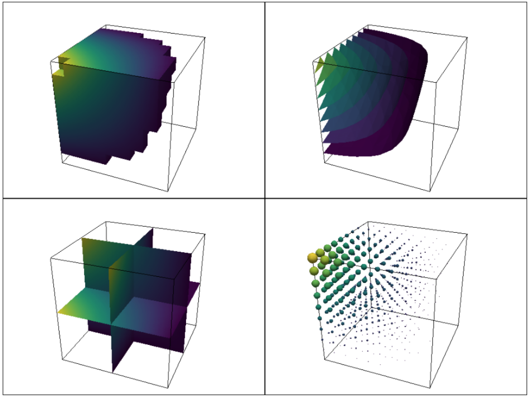
Examples - Basic Volumetric Plot

```
>>> from pyvista import examples
>>> dataset = examples.download_frog()
>>> dataset
UniformGrid (0x7f4d81806700)
  N Cells:      31594185
  N Points:        31960000
  X Bounds:        0.000e+00, 4.990e+02
  Y Bounds:        0.000e+00, 4.690e+02
  Z Bounds:        0.000e+00, 2.025e+02
  Dimensions:        500, 470, 136
  Spacing:        1.000e+00, 1.000e+00, 1.500e+0
  N Arrays:        1
>>> dataset.plot(volume=True)
```



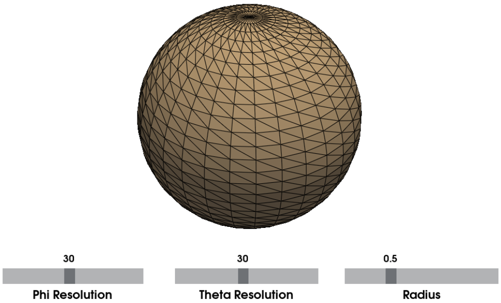
Examples - Filters

```python
import pyvista as pv
from pyvista import examples

dataset = examples.load_uniform()
outline = dataset.outline()
threshed = dataset.threshold([100, 500])
contours = dataset.contour()
slices = dataset.slice_orthogonal()
glyphs = dataset.glyph(
    factor=1e-3, geom=pv.Sphere()
)

pl = pv.Plotter(shape=(2, 2))
pl.add_mesh(outline, color="k")
pl.add_mesh(threshed, show_scalar_bar=False)
```



Examples - Widgets

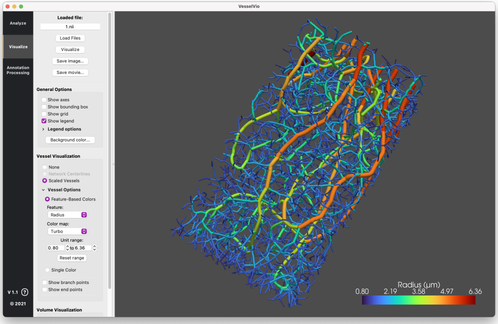
```
p = pv.Plotter()
p.add_mesh(starting_mesh, show_edges=True)
p.add_slider_widget(
    callback=callback,  # callable
    rng=[3, 60],
    value=30,
    title="Phi Resolution",
    pointa=(0.025, 0.1),
    pointb=(0.31, 0.1),
    style='modern',
)
```



Examples - PyInstaller and PyQT

- Use PyInstaller and PyQT or PySide to create a standalone application.
- Multi-platform. Build on the OS you intend to deploy.
- Compatible with GitHub Actions and can be automated.
- Deploy as using an installer like NSIS.

```
pip install -r requirements_build.txt
pyinstaller \
    --add-data=Library;Library \
    --additional-hooks-dir=Hooks \
    --icon library\icons\icon.ico \
    --windowed VesselVio.py
```



Loaded file: 1.nii

General Options

☐ Show axes
☐ Show bounding box
☐ Show grid
☑ Show legend
› Legend options

Vessel Visualization

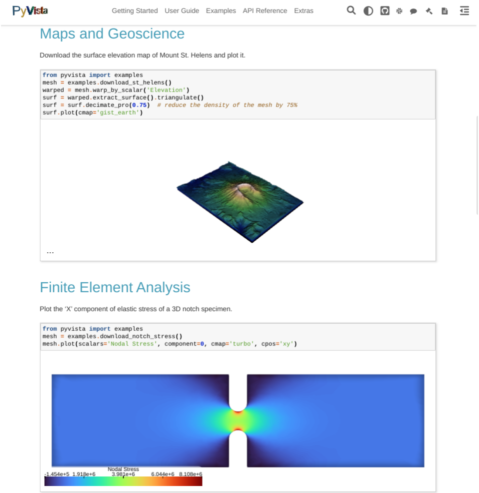
○ None
○ Network Centerlines
● Scaled Vessels
⌄ Vessel Options
● Feature-Based Colors
Feature: Radius
Color map: Turbo
Unit range: 0.80 to 6.36
○ Single Color
☐ Show branch points
☐ Show end points

Volume Visualization

Examples - Documentation with Sphinx

- PyVista supports the Sphinx documentation generator.
- Allows you to generate static and interactive documentation.
- Place code snippets directly in the documentation as examples.

```
.. jupyter-execute::

    from pyvista import examples
    mesh = examples.download_st_helens()
    warped = mesh.warp_by_scalar('Elevation')
    surf = warped.extract_surface().triangulate()
    surf = surf.decimate_pro(0.75)
    surf.plot(cmap='gist_earth')
```



Maps and Geoscience

Download the surface elevation map of Mount St. Helens and plot it.

```
from pyvista import examples
mesh = examples.download_st_helens()
warped = mesh.warp_by_scalar('Elevation')
surf = warped.extract_surface().triangulate()
surf = surf.decimate_pro(0.75)  # reduce the density of the mesh by 75%
surf.plot(cmap='gist_earth')
```

...

Finite Element Analysis

Plot the ‘X’ component of elastic stress of a 3D notch specimen.

```
from pyvista import examples
mesh = examples.download_notch_stress()
mesh.plot(scalars='Nodal Stress', component=0, cmap='turbo', cpos='xy')
```

Tutorial

The PyVista Tutorial contains a variety of lessons to help you get started with PyVista. The first lessons include:

- Introduction - Using PyVista for 3D Visualization within Python.
- Reading and plotting 3D data using the PyVista module and external files.
- Learn the basics of the PyVista data types and how to open common 3D file formats to visualize the data in 3D.
- Demonstrate many features of the PyVista plotting API to create compelling 3D visualizations and touch on animations.
- Demonstrate the PyVista filters API to perform mesh analysis and alteration.

Ansys