

PyVista is:

- PyVista is a helper library for the Visualization Toolkit (VTK) that takes a different approach on interfacing with VTK through NumPy and direct array access. This package provides a Pythonic, well-documented interface exposing VTK's powerful visualization backend to facilitate rapid prototyping, analysis, and visual integration of 3D data and referenced datasets.

<https://docs.pyvista.org>

Share this project on X: [X Tweet](#)

Want to test-drive PyVista? Check out our live examples on MyBinder: [launch](#) [binder](#)

PyVista is a NumFOCUS affiliated project



Brief Examples

Here are some brief interactive examples that demonstrate how you might want to use PyVista:

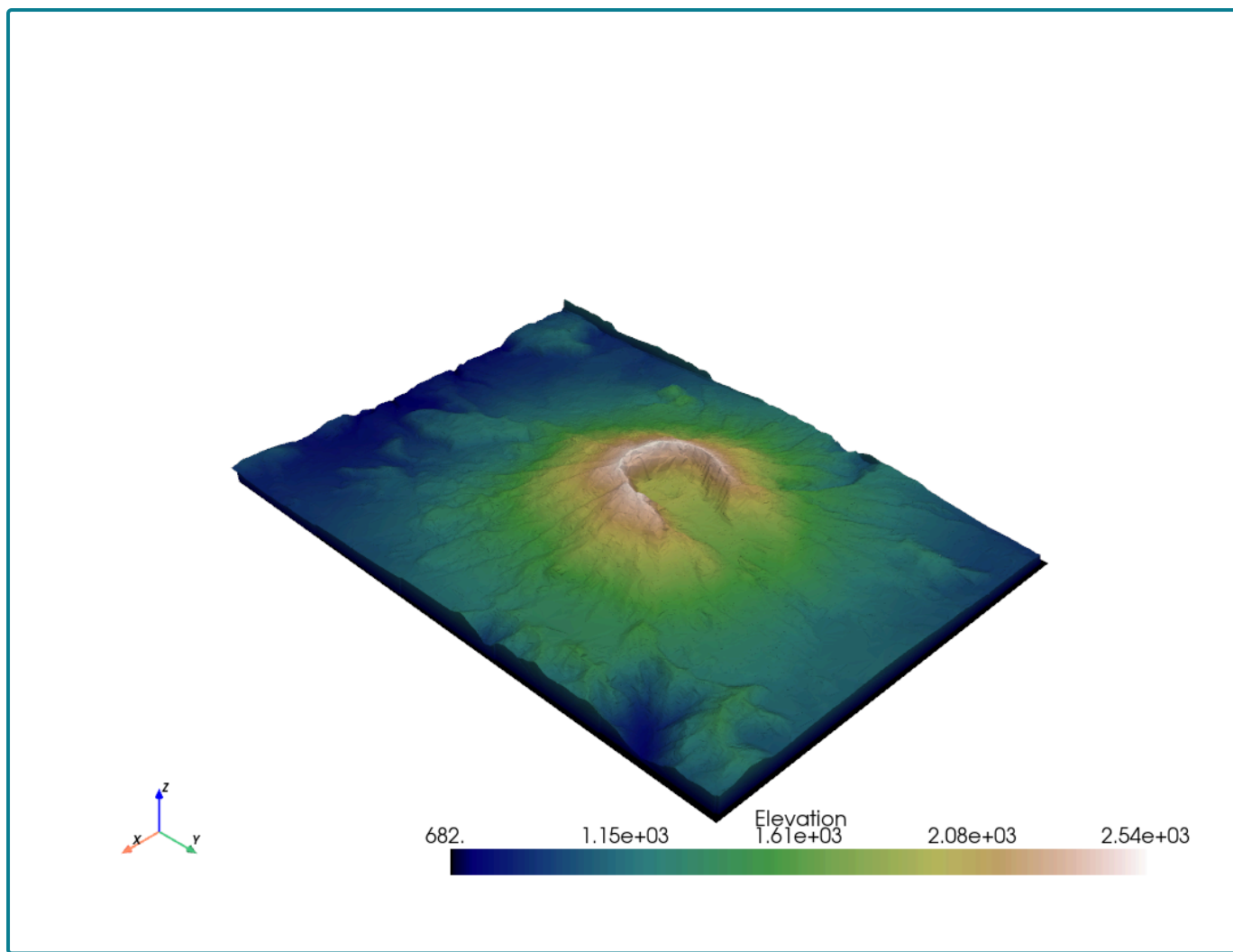
Maps and Geoscience

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')
```

Static Scene

Interactive Scene



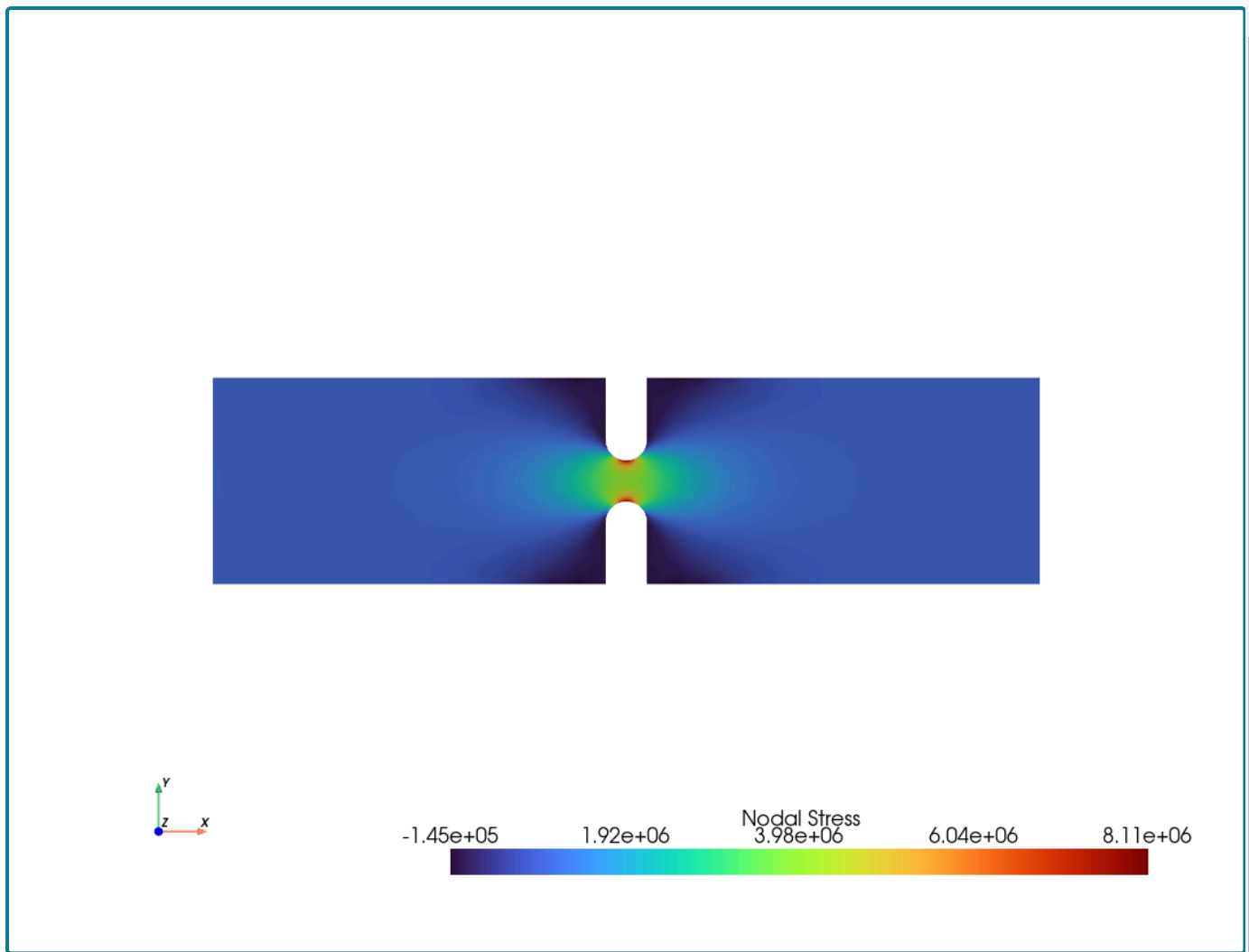
Finite Element Analysis

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')
```

Static Scene

Interactive Scene



Simple Point Cloud with NumPy

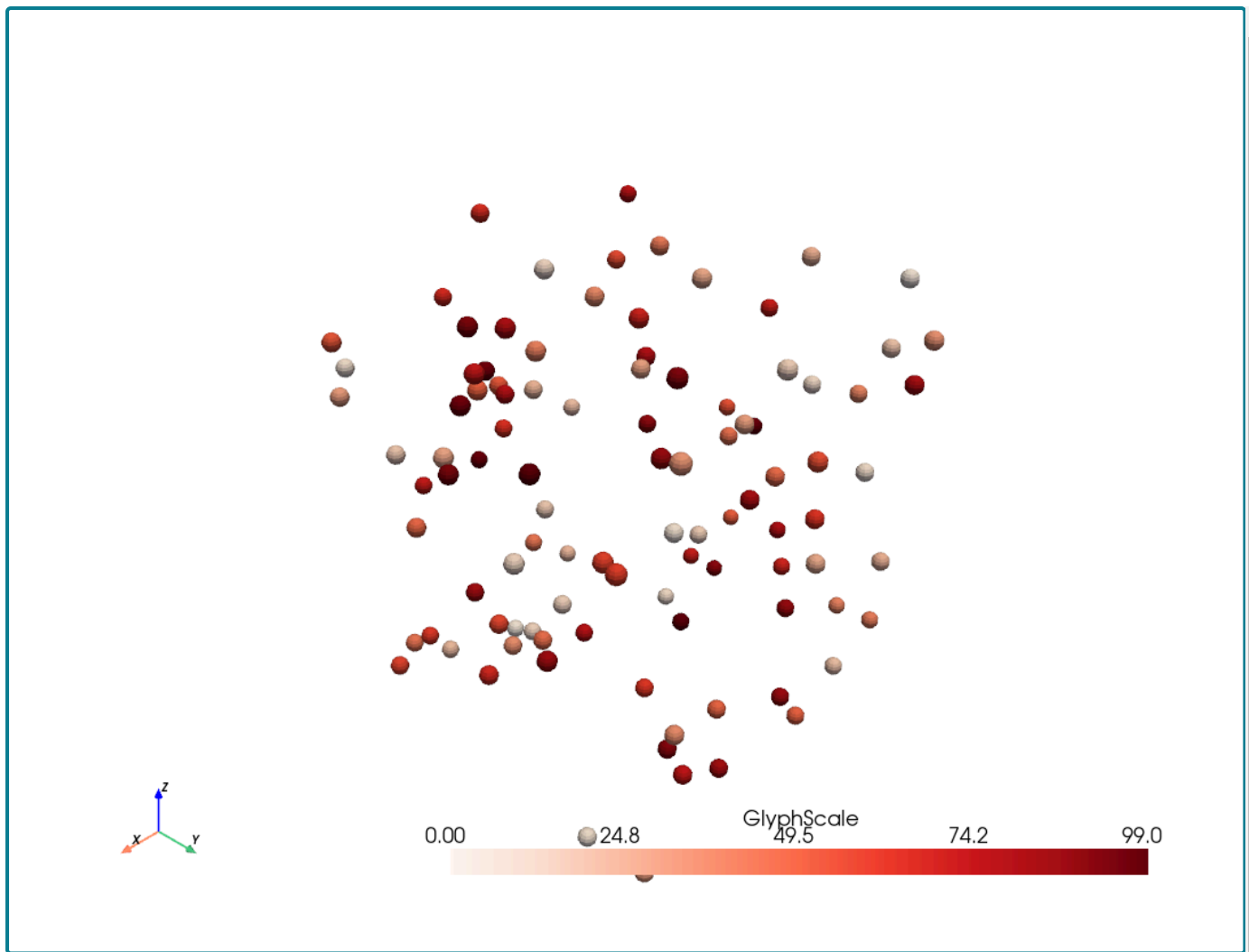
Easily integrate with NumPy and create a variety of geometries and plot them. You could use any geometry to create your glyphs, or even plot the points directly.

```
import numpy as np
import pyvista

rng = np.random.default_rng(seed=0)
point_cloud = rng.random((100, 3))
pdata = pyvista.PolyData(point_cloud)
pdata['orig_sphere'] = np.arange(100)

# create many spheres from the point cloud
sphere = pyvista.Sphere(radius=0.02, phi_resolution=10, theta_resolution=10)
pc = pdata.glyph(scale=False, geom=sphere, orient=False)
pc.plot(cmap='Reds')
```

Static Scene**Interactive Scene**



Plot a Spline

Generate a spline from an array of NumPy points.

```
import numpy as np
import pyvista

# Make the xyz points
theta = np.linspace(-10 * np.pi, 10 * np.pi, 100)
z = np.linspace(-2, 2, 100)
r = z**2 + 1
x = r * np.sin(theta)
y = r * np.cos(theta)
points = np.column_stack((x, y, z))

spline = pyvista.Spline(points, 500).tube(radius=0.1)
spline.plot(scalars='arc_length', show_scalar_bar=False)
```

Static Scene

Interactive Scene



Boolean Operations on Meshes

Subtract a sphere from a cube mesh.

```
import pyvista
import numpy as np

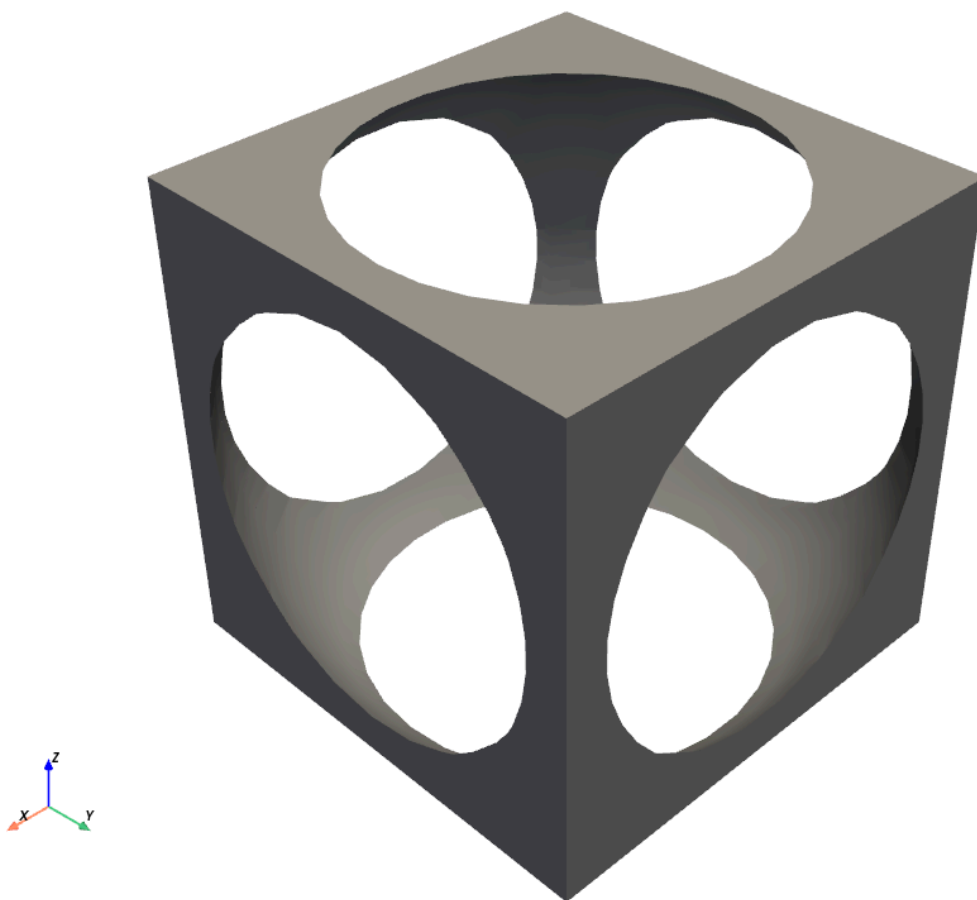
def make_cube():
    x = np.linspace(-0.5, 0.5, 25)
    grid = pyvista.StructuredGrid(*np.meshgrid(x, x, x))
    surf = grid.extract_surface().triangulate()
    surf.flip_normals()
    return surf

# Create example PolyData meshes for boolean operations
sphere = pyvista.Sphere(radius=0.65, center=(0, 0, 0))
cube = make_cube()

# Perform a boolean difference
boolean = cube.boolean_difference(sphere)
boolean.plot(color='darkgrey', smooth_shading=True, split_sharp_edges=True)
```

Static Scene

Interactive Scene



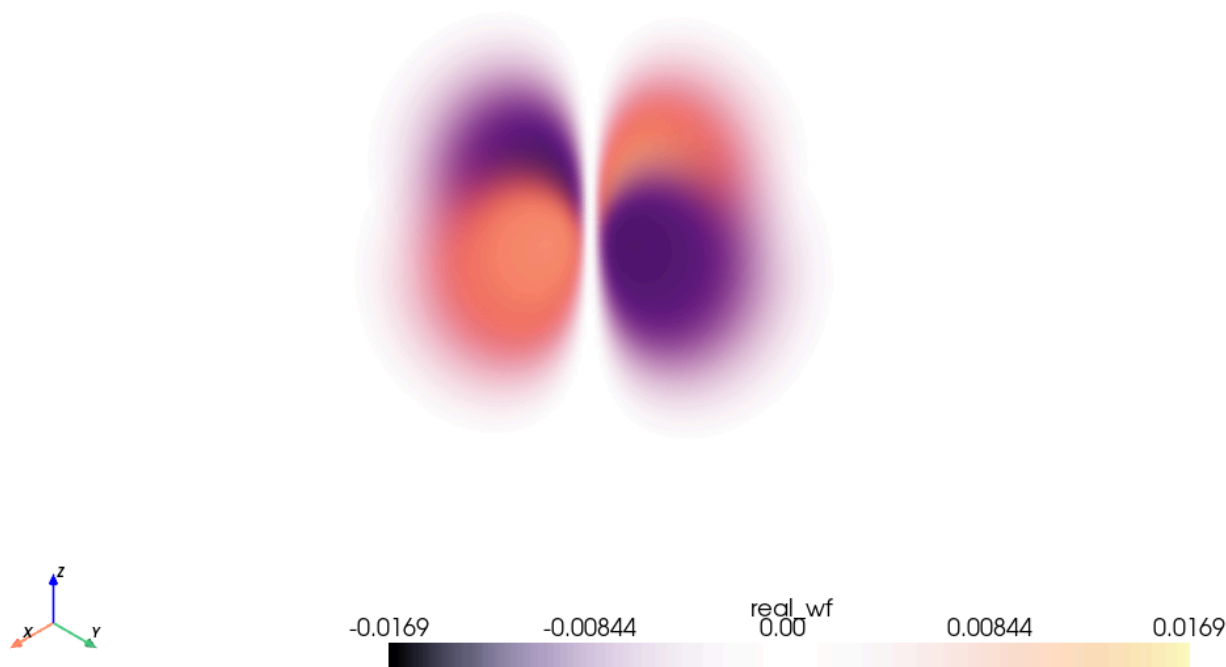
Plot Volumetric Data

Plot the $3d_{xy}$ orbital of a hydrogen atom.

Note

This example requires [sympy](#).

```
from pyvista import examples
grid = examples.load_hydrogen_orbital(3, 2, -2)
grid.plot(volume=True, opacity=[1, 0, 1], cmap='magma')
```



Translating

The recommended way for new contributors to translate PyVista's documentation is to join the translation team on Transifex.

There is a [pyvista translation page](#) for pyvista (main) documentation.

1. Login to [transifex](#) service.
2. Go to [pyvista translation page](#).
3. Click **Request language** and fill form.
4. Wait acceptance by transifex pyvista translation maintainers.
5. (After acceptance) Translate on transifex.
6. We can host the translated document in [GitHub Pages](#) by creating [GitHub repository](#).
7. Translation is backed up in [pyvista-doc-translations](#).

Details can be found here: <https://help.transifex.com/en/>

Status

© Copyright 2017-2024, The PyVista Developers.

Built with the [PyData Sphinx Theme](#) 0.15.4.