LA-UR-24-28796

Approved for public release; distribution is unlimited.

Title:	Using ParaView to Visualize MCNP6 Fission Matrix Eigenmodes
Author(s):	Vaquer, Pablo Andres
Intended for:	2024 MCNP® User Symposium, 2024-08-19/2024-08-22 (Los Alamos, New Mexico, United States)
Issued:	2024-08-30 (rev.1)

Los Alamos NATIONAL LABORATORY

Using ParaView to Visualize MCNP6 Fission Matrix Eigenmodes

Pablo A. Vaquer

MCNP User Symposium 2024

August 22, 2024

Introduction

- This presentation will demonstrate how to generate MCNP6 [1] fission matrix eigenmodes for a nuclear reactor, which includes:
  - Using the KOPTS and HSRC card in MCNP6
  - Using h5py to parse the MCNP6 runtape.h5
  - Using Python to compute the eigenvalues and eigenvectors of the fission matrix, and to generate a VTI file
- This presentation will also demonstrate how to use ParaView to visualize the eigenvectors, stored in the VTI file
- The Texas A&M University Nuclear Science Center Reactor (NSCR) was used as an example

Theory: the MCNP6 fission matrix equation

The k-eigenvalue transport equation can be expressed as

$$\widehat{M}\psi\left(\vec{r},\widehat{\Omega},E\right)=\frac{1}{k}\frac{\chi(E)}{4\pi}S(\vec{r})$$

where $\widehat{M}$ is the migration operator (which includes streaming collision, and scattering) and $\boldsymbol{S(\vec{r})}$ **is the fission source** [2][3][4]

A fission-matrix eigenvalue equation can be derived via spatial discretization

$$kS_i=\sum_j^J F_{ij}S_j$$

where F_{ij} is a component of the fission matrix, which indicates the probability that a **fission neutron born in region *j*** leads to the production of a **fission neutron in region *i***

The KOPTS and HSRC cards

- Adding the following KOPTS card to an MCNP6 input deck will instruct MCNP6 to compute the fission matrix

```
KOPTS FMAT=yes
```

- The fission matrix will be evaluated for the spatial grid that is provided by the HSRC card

```
HSRC nx xmin xmax ny ymin ymax nz zmin zmax
```

- Note, the KOPTS and HSRC cards are meant to be used in conjunction with the KCODE card

Python scripts for creating VTI file (1/2) fission_matrix_eigs.py

Imports the other Python script

The **h5py** Python package is used to extract fission matrix data from the MCNP6 runtape.h5 file

The **SciPy** Python package is used to compute eigenvalues and eigenvectors

Eigenmodes are saved to a **VTI** file

```python
import h5py
import numpy as np
import scipy.sparse as sparse
import scipy.sparse.linalg as sla
import convert_numpy_to_vtk
SUPPORTED_RUNTAPES = ([1, 0, 0], [1, 0, 1])
#-----------------------------------------------------------------------
def extract_fmat(runtape: str):
    """Returns the last saved fission matrix as a scipy.sparse.csr_matrix"""
    with h5py.File(runtape, "r") as handle:
        version_file = handle["config_control"].attrs["version_file"]
        if any(SUPPORTED_RUNTAPES[0] != version_file) and any(SUPPORTED_RUNTAPES[1] != version_file):
            print("Warning: possibly incompatible runtape detected.")
        fmat = handle["results/fission_matrix"]
        n_dim = fmat["n"][()]
        indices = fmat["indices"][:]
        indptr = fmat["indptr"][:]
        data = fmat["data"][:]
        n_xyz = fmat["n_xyz"][:]
        delta_xyz = fmat["delta_xyz"][:]
        origin = fmat["origin"][:]
    return (sparse.csr_matrix((data, indices, indptr), shape=(n_dim, n_dim)),
        n_xyz, delta_xyz, origin)
#-----------------------------------------------------------------------
def get_eigs(matrix, n_xyz, n_tot=30):
    """Retrieve eigenvalues/eigenvectors, sort, reshape to 3D"""
    eigenvalues, eigenvectors = sla.eigs(matrix, k=n_tot)
    indices = np.argsort(-np.abs(eigenvalues))
    sorted_eigvals = np.sort(eigenvalues[indices].real)[::-1]
    sorted_eigvecs = [eigenvectors[:, i].reshape(n_xyz[::-1]).transpose().real for i in indices]
    sorted_eigvecs[0] = np.abs(sorted_eigvecs[0])
    return sorted_eigvals, sorted_eigvecs
#-----------------------------------------------------------------------
def eigs_to_vti_file(origin, delta_xyz, eigvals, eigvecs, output_file: str):
    """Generate VTI file containing eigenvalues/eigenvectors"""
    data_dict = dict()
    for i, eigval in enumerate(eigvals):
        data_dict[f'{i} eigenmode, eigenvalue = {eigval}'] = eigvecs[i]
    convert_numpy_to_vtk.image_data_to_vti_file(origin, delta_xyz, data_dict, output_file)
#-----------------------------------------------------------------------
fission_matrix, n_xyz, delta_xyz, origin = extract_fmat("nscr.r.h5")
eigvals, eigvecs = get_eigs(fission_matrix, n_xyz, n_tot=30)
eigs_to_vti_file(origin, delta_xyz, eigvals, eigvecs, "nscr_eigs.vti")
```

** adapted from [4]*

Python scripts for creating VTI file (2/2)

convert_numpy_to_vtk.py

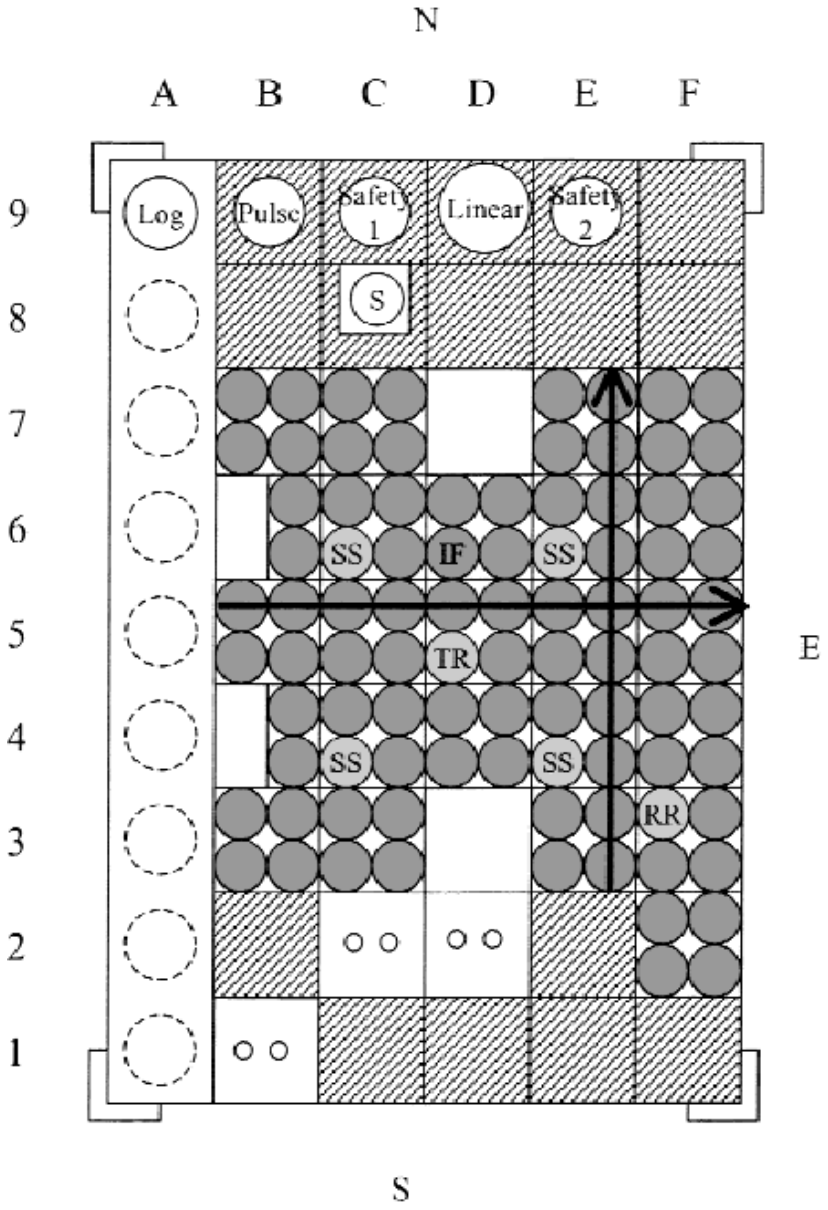
Writes a VTI **file**

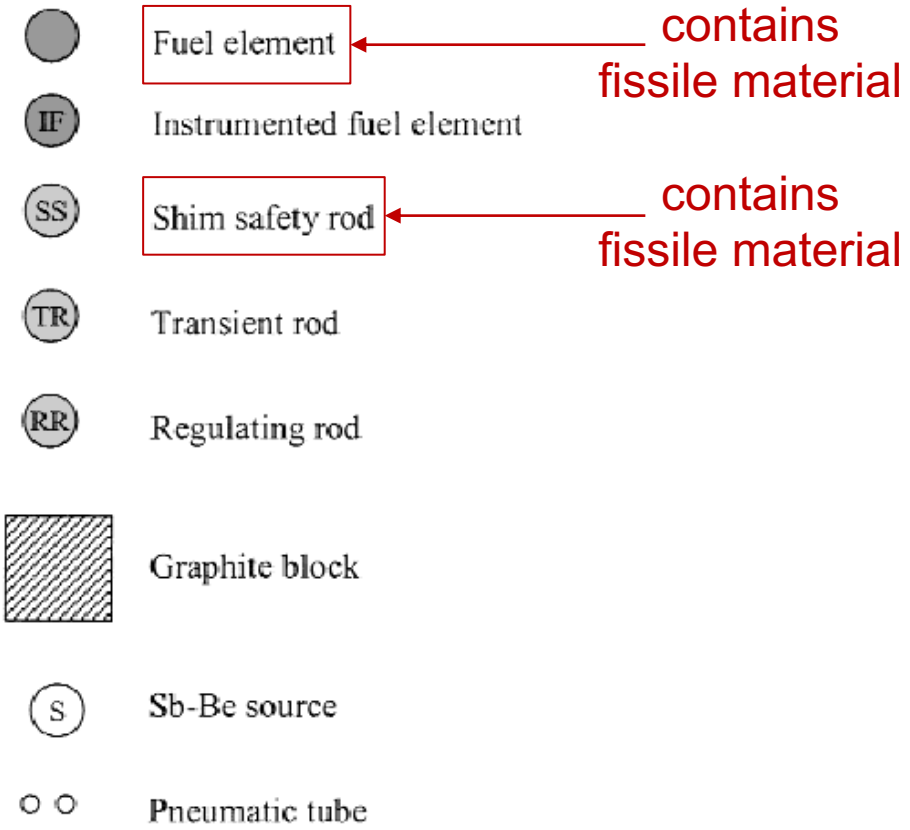
Uses image data, stored as NumPy arrays and dictionaries, and converts them in a VTI **object**

Combines the two functions above to produce a VTI **file**

```
import vtk
import vtk.util.numpy_support as numpy_support
#-----------------------------------------------------------------------------
def write_vti_file(vti, output_file: str):
    """Write VTI file"""
    writer = vtk.vtkXMLImageDataWriter()
    writer.SetFileName(output_file)
    writer.SetInputData(vti)
    writer.Write()
#-----------------------------------------------------------------------------
def image_data_to_vti_object(origin, spacing, data_dict: dict):
    """Convert image data stored as numpy arrays to a VTI object"""
    assert len(origin) == 3
    assert len(spacing) == 3
    vti = vtk.vtkImageData()
    vti.SetOrigin(origin[0], origin[1], origin[2])
    vti.SetSpacing(spacing[0], spacing[1], spacing[2])
    for field_name, data in data_dict.items():
        vti.SetDimensions(data.shape[0]+1, data.shape[1]+1, data.shape[2]+1)
        vtk_data = numpy_support.numpy_to_vtk(num_array=data.transpose().flatten(),
            deep=True, array_type=vtk.VTK_FLOAT)
        vtk_data.SetName(field_name)
        vti.GetCellData().AddArray(vtk_data)
    return vti
#-----------------------------------------------------------------------------
def image_data_to_vti_file(origin, spacing, data_dict: dict, output_file: str):
    """Convert image data stored as numpy arrays to a VTI file"""
    vti = image_data_to_vti_object(origin, spacing, data_dict)
    write_vti_file(vti, output_file)
#-----------------------------------------------------------------------------
```

Nuclear Science Center Reactor (NSCR) Layout



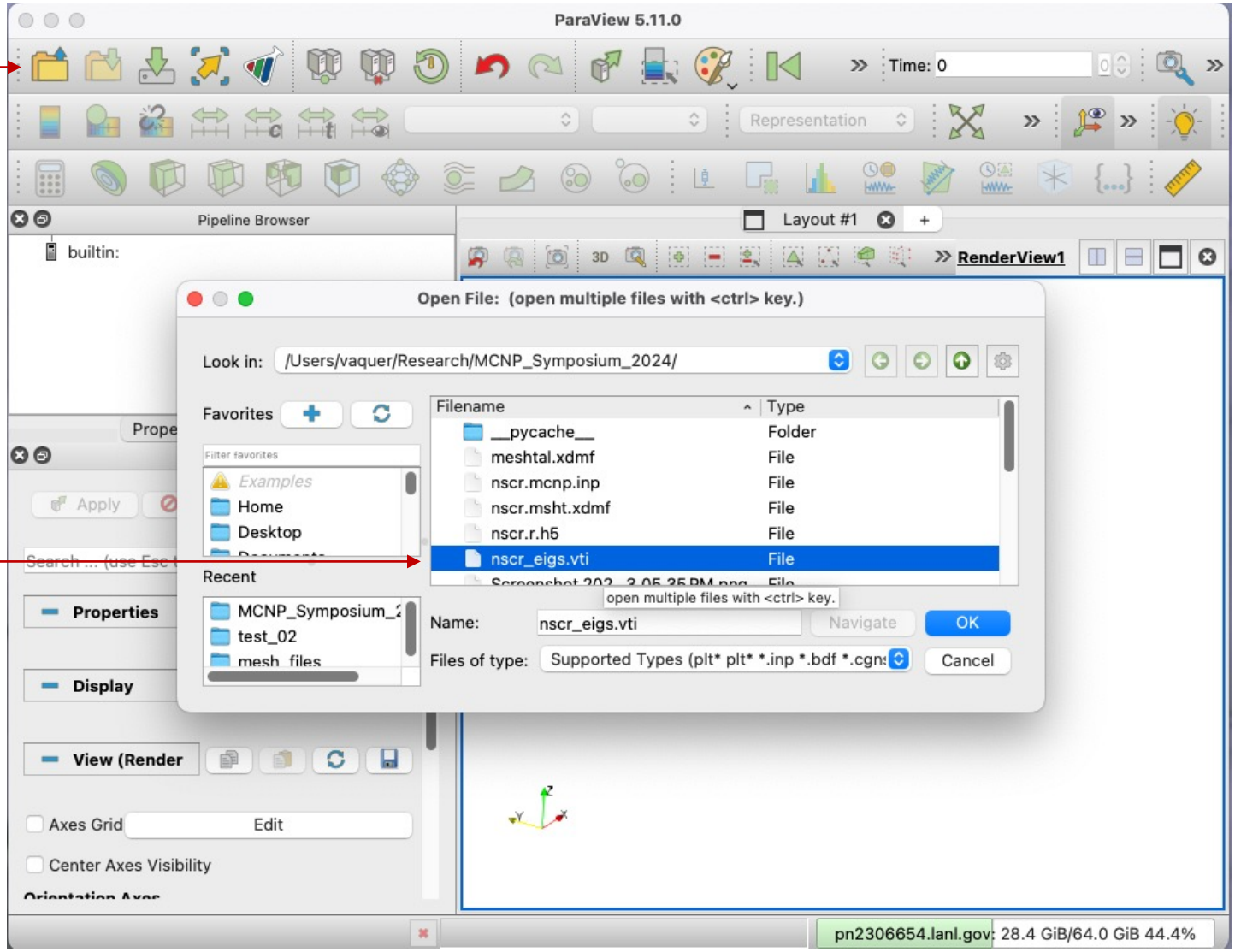


* *image from [5]*

Opening the VTI file with ParaView

Click to open file

Select VTI file



Plot of the 0th eigenmode (k_0=1.005) with ParaView

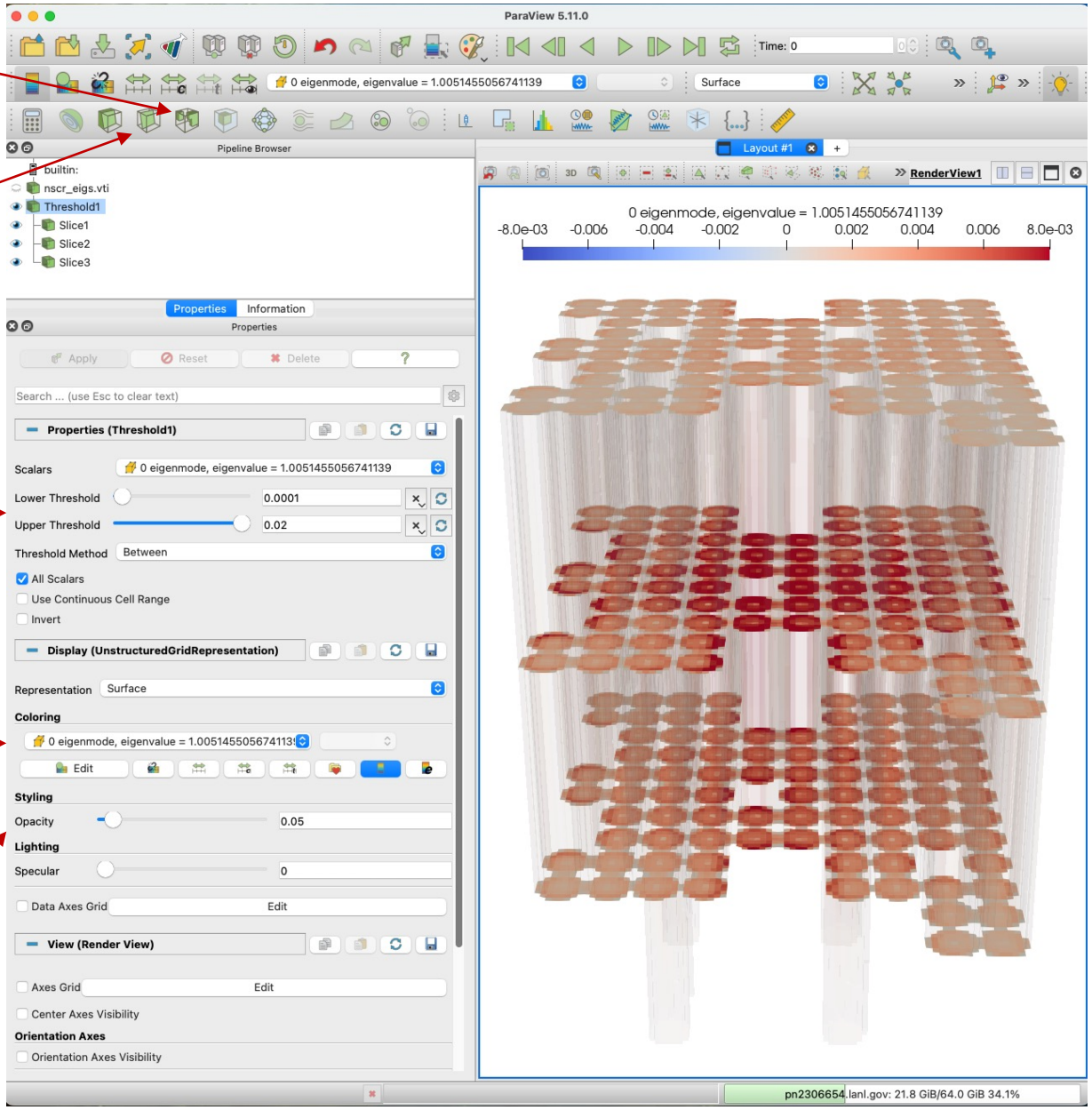
Add a threshold

Add multiple slices along Z dimension

Vary threshold values here

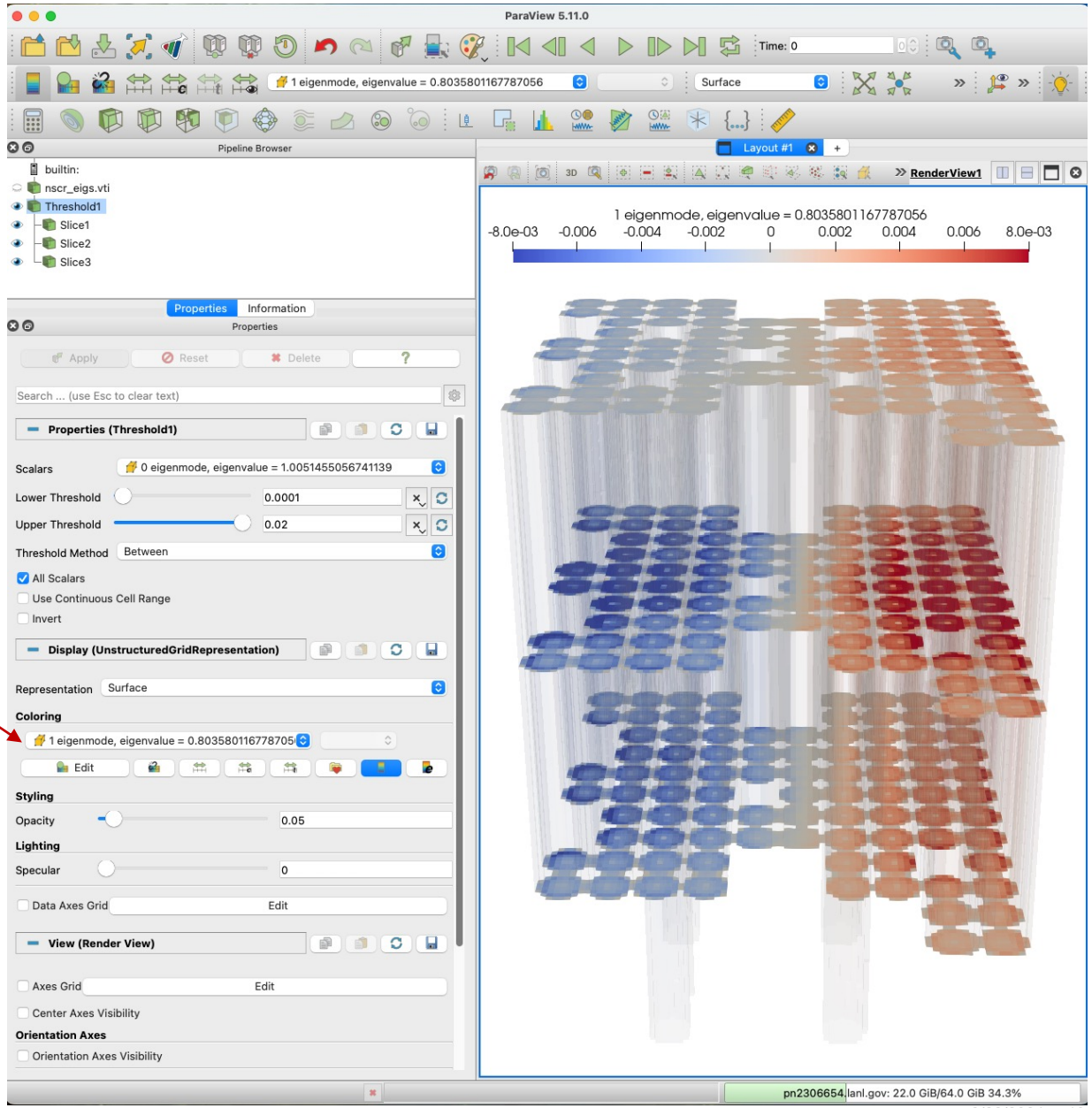
Select 0th eigenmode

Vary opacity here



Plot of the 1st eigenmode (k_1=0.804) with ParaView

Select 1st eigenmode (this must be done for all slices)



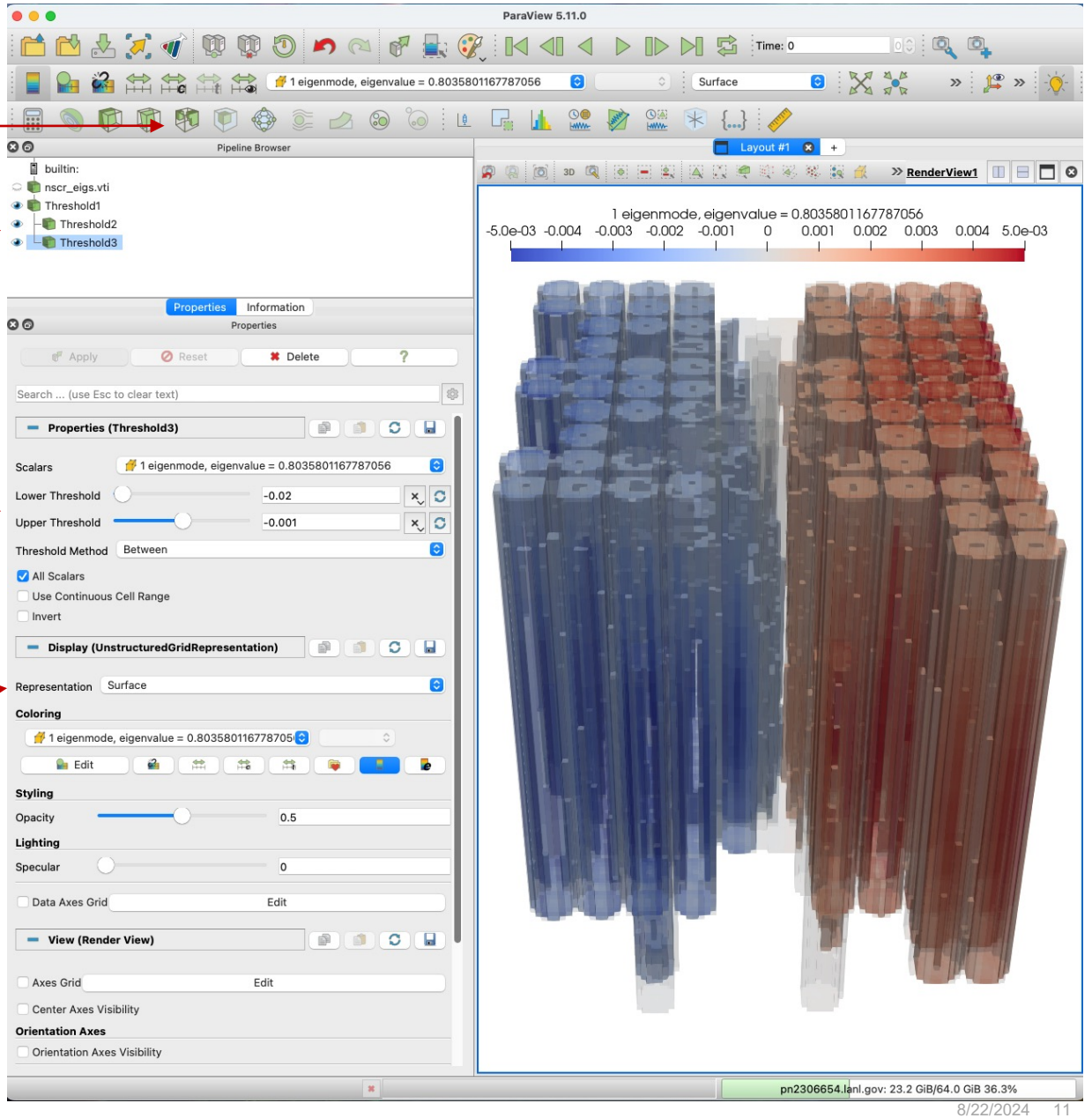
Alternative plot of the 1st eigenmode (k_1=0.804)

Add separate thresholds for negative and positive values

Vary threshold values here

Many other representations can be plotted:

- points
- wireframes
- volumes



References

1. J. A. Kulesza, T. R. Adams, J. C. Armstrong, S. R. Bolding, F. B. Brown, J. S. Bull, T. P. Burke, A. R. Clark, R. A. Forster, III, J. F. Giron, A. S. Grieve, C. J. Josey, R. L. Martz, G.W. McKinney, E. J. Pearson, M. E. Rising, C. J. Solomon, JR., S. Swaminarayan, T. J. Trahan, S. C. Wilson, and A. J. Zukaitis, "MCNP. Code Version 6.3.0 Theory & User Manual," Tech. Rep. LA-UR-22-30006, Rev. 1, Los Alamos National Laboratory, Los Alamos, NM, USA. Sep. 2022.
2. S. E. Carney, F. B. Brown, B. C. Kiedrowski, W. R. Martin. Higher-mode Applications of Fission Matrix Capability for MCNP. Los Alamos National Laboratory Tech. Rep. LA-UR-13-27078. Los Alamos, NM, USA. 2013.
3. F. B. Brown, S. E. Carney, B. C. Kiedrowski, W. R. Martin. Fission Matrix Capability for MCNP Monte Carlo. Los Alamos National Laboratory Tech. Rep. LA-UR-13-26962. Los Alamos, NM, USA. 2013.
4. J. A. Kulesza, C. J. Josey. Fission Matrix Processing Using the MCNP6.3 HDF5 Restart File. Los Alamos National Laboratory Tech. Rep. LA-UR-22-20980, Rev. 1. Los Alamos, NM, USA. May 2022.
5. Kim, C. H., Jang, S., & Reece, W. D. (2004). Monte Carlo Modeling of the Texas A&M University Research Reactor. *Nuclear Technology*, *145*(1), 1–10. https://doi.org/10.13182/NT04-A3455