Using the VTK Library: A Summary

Peter Mortensen

May 2021

Overview

The Visualization Toolkit (VTK) is open source software for manipulating and displaying scientific data. It comes with state-of-the-art tools for 3D rendering, a suite of widgets for 3D interaction, and extensive 2D plotting capability.

VTK can be installed with pip:
`pip install vtk`.

Necessary Python Libraries

This overview will use only the vtk library and common essential libraries.

```
import numpy as np
import os
import sys
import vtk
from vtk.util.numpy_support import vtk_to_numpy
from vtk.numpy_interface import dataset_adapter as dsa
```

The below code will use

A useful collection of examples is found at https://kitware.github.io/vtk-examples/site/Python/

Reading files

Read .vtk file to access polydata:

```
fname = 'bav02_root_med-prop-recon_f1.vtk'

reader = vtk.vtkPolyDataReader()
print('Reading Reference Frame:', ref)
# Read the source file.
reader.SetFileName(fname)
reader.ReadAllScalarsOn()
reader.ReadAllVectorsOn()
reader.Update()
polydata = reader.GetOutput()
```

Accessing file data

A summary of the dataset can be accessed by using `print(polydata)`

The polydata object can also give you the data values directly.

```
#Get the number of points and cells
NP = polydata.GetNumberOfPoints()
NC = polydata.GetNumberOfCells()
print('Number of Points = ',NP)
print('Number of Cells = ',NC)

#Get the array of points
Points = vtk_to_numpy(polydata.GetPoints().GetData())
```

Output:

```
>> Number of Points =  900
>> Number of Cells =  1728
```

Check if there is array data and is there is save it to an array.

```
#Get array data
if not polydata.GetPointData().GetArray(0):
    print('Warning: No saved data')
else:
    NumOfArr = polydata.GetPointData().GetNumberOfArrays()
    Data = np.zeros((NumOfArr,NP))
    print('List of Array Data:')
    for i in range(NumOfArr):
        ArrayName = polydata.GetPointData().GetArrayName(i)
        print(ArrayName)
        if i in range(1,11):
            Data[i,:] = vtk_to_numpy(polydata.GetPointData().
    GetArray(i))
```

Plotting data

The VTK library can be very cumbersome when plotting vtk data, thus it is recommended that the library vtkplotlib is used which has all the functionality of the VTK library but with a simpler format, similar to the commonly use matlibplot library.

vtkplotlib simply requires the appropriate numpy arrays.

To plot the mesh create an array of the cells in the mesh of dimension (NC,3,3), where NC is the number of cells in your mesh

```
Cells = np.zeros((NC,3,3))
for i in range(NC):
    Cells[i,:,:] = vtk_to_numpy(polydata.GetCell(i).GetPoints().
    GetData())

plot = vpl.plots.MeshPlot.MeshPlot(Cells)

vpl.show()
```

The points of the mesh can coloured to give a colourmap of the data at each points, for example, the wall thickness at each point. This requires a lookup table for the colour range.

In the below example, the second array of Data

```
# Colour transfer function.
ctf = vtk.vtkColorTransferFunction()
ctf.SetColorSpaceToDiverging()
p1 = [0.0] + list(colors.GetColor3d('MidnightBlue'))
p2 = [1.0] + list(colors.GetColor3d('DarkOrange'))
ctf.AddRGBPoint(*p1)
ctf.AddRGBPoint(*p2)
cc = list()
for i in range(256):
    cc.append(ctf.GetColor(float(i) / 255.0))

# Lookup table
lut = vtk.vtkLookupTable()
lut.SetNumberOfColors(256)
for i, item in enumerate(cc):
    lut.SetTableValue(i, item[0], item[1], item[2], 1.0)
lut.SetRange(np.amin(Data[1,:]),np.amax(Data[1,:]))
lut.Build()

Cells = np.zeros((NC,3,3))
TD = np.zeros((NC,3))
for i in range(NC):
    Cells[i,:,:] = vtk_to_numpy(polydata.GetCell(i).GetPoints().
    GetData())
    #Assign point id
    t = np.array([polydata.GetCell(i).GetPointId(j) for j in range
    (3)])
    #For each cell assign thickness for each point
    TD[i,:] = [Data[1,t[j]] for j in range(3)]

plot = vpl.plots.MeshPlot.MeshPlot(Cells, tri_scalars=None, scalars
    =TD, color=None, opacity=None, cmap=lut, fig='gcf', label=None)

# Optionally the plot created by mesh_plot can be passed to
    color_bar
vpl.color_bar(plot, "Thickness")

vpl.show()
```

Writing Data

The ith point can be redefined to be a new point using 'polydata.GetPoints().SetPoint(i,NewPt)'.

In the following example, the points are redefined so that the centre of the mesh is at the point (0,0,0).

```
#Find Mid point of the mesh
Mids = np.zeros((3))
Mids[0] = np.mean(Ranges[0:2])
Mids[1] = np.mean(Ranges[2:4])
Mids[2] = np.mean(Ranges[4:6])
```

```
6
7  #Redefine the point locations
8  for i in range(NP):
9      ptNew = np.array([Points[i,j] - Mids[j] for j in range(3)])
10     polydata.GetPoints().SetPoint(i, ptNew)
```

Also, array data can be added to the points, cells, and fields. Make sure that the new array of data has the correct dimensions. Below are examples for cell and point data.

```
1  #Create New Cell data
2  NewCellData = np.random.randn(NC)
3  # Add Cell Data
4  CellData  = [NewCellData]
5  CellNames = ['NewCellData']
6  for i in range(len(CellNames)) :
7      arrayCell = vtk.util.numpy_support.numpy_to_vtk(CellData[i],
       deep=True)
8      arrayCell.SetName(CellNames[i])
9      dataCells = polydata.GetCellData()
10     dataCells.AddArray(arrayCell)
```

Note that the data at the cells can be converted into point data using vtkCellDataToPointData().

```
1  # Convert Cell Data to Point Data
2  c2p = vtk.vtkCellDataToPointData()
3  c2p.AddInputData(polydata)
4  c2p.Update()
5  c2p.GetOutput()
6
7  #Create new point data
8  NumArr = polydata.GetPointData().GetNumberOfArrays()
9  NewPointData = vtk_to_numpy(c2p.GetPolyDataOutput().GetPointData().
       GetArray(NumArr))
10
11 # Add Point Data
12 PointData = [NewPointData]
13 PointNames = ['NewPointData']
14 for i in range(len(PointNames)) :
15     arrayPoint = vtk.util.numpy_support.numpy_to_vtk(PointData[i],
       deep=True)
16     arrayPoint.SetName(PointNames[i])
17     dataPoints = polydata.GetPointData()
18     dataPoints.AddArray(arrayPoint)
19     dataPoints.Modified()
```

Once the new data has been added the polydata can be saved as a new vtp or vtk file. This example asks you to choose either a vtp or vtk format, but others are available

```
1  #Save new data
2  #Choose format
3  opformat = 'vtp'
4
5  #define writer and new filename
6  if opformat == 'vtp':
7      fname = 'NewVTKSummaryData.vtk'
```

```
    writer = vtk.vtkXMLDataSetWriter()
elif opformat == 'vtk':
    fname = 'NewVTKSummaryData.vtk'
    writer = vtk.vtkDataSetWriter()

writer.SetFileName(Newfname)
writer.SetInputData(polydata)
print('Writing',fname)
writer.Write()
```