Visualizing BOUT++ data with VisIt

BOUT++ Mini Workshop 2014

October 7, 2014

Eric Brugger
Overview

- Overview of VisIt (10 min)
- Getting your BOUT++ data into VisIt (10 min)
- Visualizing BOUT++ data (20 min)
- Additional resources (3 min)
- Closing remarks and questions (2 min)
- Hands on tutorial (30 min)
VisIt is an open source, richly featured, turnkey application for scientific data analysis and visualization.

- Utilizes a plugin architecture for extensibility
  - Rendering methods
  - Data manipulation
  - Data readers
- Utilizes a client/server architecture
- Provides a scalable/parallel server
- Multiple interfaces
  - A graphical user interface
  - A python scripting interface
  - A java library interface
- Multi-platform support – Unix, Windows, MacOS
Major use cases

Data Exploration

Presentations

Comparative Analysis

Visual Debugging

Quantitative Analysis
VisIt employs a parallelized client-server architecture.
VisIt automatically switches to a scalable rendering mode for large data sets.

- Rendering Modes:
  - Local (hardware)
  - Remote (software or hardware)

- Beyond surfaces:
  - VisIt also provides scalable volume rendering.
VisIt scales well on current HPC platforms.

<table>
<thead>
<tr>
<th>Machine</th>
<th>Architecture</th>
<th>Problem Size</th>
<th># of Cores</th>
</tr>
</thead>
<tbody>
<tr>
<td>Graph</td>
<td>X86_64</td>
<td>$20,001^3 (8 \text{Tz})$</td>
<td>12K</td>
</tr>
<tr>
<td>Dawn</td>
<td>BG/P</td>
<td>$15,871^3 (4 \text{Tz})$</td>
<td>64K</td>
</tr>
<tr>
<td>Franklin</td>
<td>Cray XT4</td>
<td>$12,596^3 (2 \text{Tz})$</td>
<td>32K</td>
</tr>
<tr>
<td>JaguarPF</td>
<td>Cray XT5</td>
<td>$12,596^3 (2 \text{Tz})$</td>
<td>32K</td>
</tr>
<tr>
<td>Juno</td>
<td>X86_64</td>
<td>$10,000^3 (1\text{Tz})$</td>
<td>16K</td>
</tr>
<tr>
<td>Franklin</td>
<td>Cray XT4</td>
<td>$10,000^3 (1\text{Tz})$</td>
<td>16K</td>
</tr>
<tr>
<td>Ranger</td>
<td>Sun</td>
<td>$10,000^3 (1\text{Tz})$</td>
<td>16K</td>
</tr>
<tr>
<td>Purple</td>
<td>IBM P5</td>
<td>$8,000^3 (0.5 \text{Tz})$</td>
<td>8K</td>
</tr>
</tbody>
</table>

*Scaling Studies of Isosurface Extraction and Volume Rendering* (2009)
Getting your data into VisIt

- VisIt requires your data to be stored as a NETCDF file.

- The data must be stored with the appropriate conventions to be recognized as a BOUT++ file.
  - If you use the incorrect conventions then VisIt will treat the file as a generic NETCDF file and you will only be able to render the data as multi-dimensional arrays.
  - When you store it using the correct conventions VisIt will automatically transform it into your real geometry.
The format of the solution file

Grid file

Solution files

Necessary variables

Must be in t, x, y, z order

Name of grid file
The format of the grid file

```
edge83{bruger}43: ncdump -h D3D_144382.02510_x516y64.psi080to120mer.nc
netcdf D3D_144382.02510_x516y64.psi080to120mer {
  dimensions:
    y = 64 ;
    x = 516 ;
    x2 = 3 ;
    x3 = 2 ;
  variables:
    short nx ;
    short ny ;
    int ixseps1 ;
    short ixseps2 ;
    int jyseps1_1 ;
    int jyseps1_2 ;
    int jyseps2_1 ;
    int jyseps2_2 ;
    int ny_inner ;
    float dx(x, y) ;
    float dy(x, y) ;
    float ShiftAngle(x) ;
    float zShift(x, y) ;
    float pol_angle(x, y) ;
    double ShiftTorsion(x, y) ;
    float Rxy(x, y) ;
    float Zxy(x, y) ;
    float Bpxy(x, y) ;
    float Btxy(x, y) ;
    float Bxy(x, y) ;
    Float hthe(x, y) ;
```

Necessary variables
When things go wrong

- Use "ncdump -h field_file" to check that the required variables are present
- Use "ncdump -v gridname field_file" to check that the gridname is correct
- Use "ncdump -h grid_file" to check that the required variables are present
- Use "visit -debug 4" and look at the resulting "vlog" files for a message that may indicate the problem.
VisIt automatically detects the standard grid configurations (circular, 1X, 2X)

Circular grid

1 X grid

2 X grid
VisIt adaptively refines the grid based on zShift

Large jump in zShift leads to a lot of refinement

zShift  Original grid  Refined grid
VisIt will break the grid into several structured pieces

Circular grid

1 X grid

2 X grid
VisIt will also create structured blocks as it replicates the blocks around the taurus.

A 1 X grid with zPeriod = 8 gives 4 * 8 = 32 grids.
Opening a BOUT file

Select the open icon to open a file

Open the Solution file
Running in parallel

- When you open the first file, VisIt will prompt you to start a compute engine.
- Choose a parallel one.
- Don’t use more processors than blocks.
- E.g. 1 X grid with zperiod=8 gives 32 blocks, so use 32 processors (using more or less won’t hurt, but may result in lower performance or wasting cpus).
Selecting variables to display

Press the Add icon to bring up the list of plots and then the list of variables.

- P in 3d
- P in 3d with zShift applied
- zShift in 2d
Creating a simple plot

- This is a pseudo color plot of electron temperature with a clip applied
  - If you don’t either slice or clip the 3d data it will be very boring

- It has a non-standard color table

- It has some annotations removed, the color bar is enlarged from the default
Creating a simple plot – specifics

- Icon to add plots
- Icon to add operators
- Downward facing triangle expands the plot
- Time slider to change the time state
- Clip operator, double click to bring up the clip attributes
- Pseudo color plot, double click to bring up the pseudo color attributes
- Click to remove the clip operator
Creating a simple plot – the pseudo color attributes

Use the limits to set a specific range to color the data. Useful for maintaining the scale with time animations.

Click on the color table icon to select a different color table. You may want to invert it.

You may want to turn off lighting on slices.
Creating a simple plot – the clip attributes

Accurate is usually the best

Set the origin and normal for up to 3 planes.
- 1 plane removes a half
- 2 planes remove a quarter
- 3 planes remove an octant
Creating a simple plot – annotations

- Turn off the database and user information titles
- Turn off the axes and bounding boxes
- Set the legend attributes. You can also add lines and titles.
Creating a more sophisticated image
Step 1: Create a pseudo color of te with a clip to show the interior

Make sure these are unchecked
Start with a pseudo color plot of te
Adjust the pseudo color attributes

Use the limits to set a specific range to color the data. Useful for maintaining the scale with time animations.

Click on the color table icon to select a different color table. You may want to invert it.
Clip out the front third of the taurus
Clipping out the front third of the taurus

Accurate is usually the best

Set the origin and normal for 2 planes. 2 planes remove a quarter.
Remove the unnecessary annotations and change the view.
Removing the unnecessary annotations

Turn off the database and user information titles

Turn off the axes and bounding boxes
Adjust the view with the interactive tools and fine tune with the view attributes

Orientation settings

Image panning settings

Image zooming settings
Adjust the legend attributes
Adjusting the legend attributes

The legend position

The legend size

The font
Add the titles for the legend
Adding the titles for the legend

1st line of title

2nd line of title

Title properties

Title properties
Step 2: Add a pseudo color plot of \texttt{te\_zshift} on a slice
Set the pseudo color attributes

- Set the same limits as the other pseudo color plot.
- Set the same color table as the other pseudo color plot.
- Turn off the legend.
Set the slice and clip attributes

The slice attributes

The slice operator creates 2 intersections. Remove the unwanted one.
Step 3: Add a second pseudo color plot of te_zshift on the other slice
Set the pseudo color attributes

Set the same limits as the other pseudo color plot.

Set the same color table as the other pseudo color plot.

Turn off the legend.
Set the slice and clip attributes

The slice attributes

The slice operator creates 2 intersections. Remove the unwanted one.
Step 4: Add a pseudocolor plot with an iso surface operator
Set the pseudo color attributes

- Set the same limits as the other pseudo color plot.
- Set the same color table as the other pseudo color plot.
- Turn off the legend.
Set the iso surface attributes

Set the value of the iso surface

[Image of a window with settings for isosurface attributes, highlighting the value set to 0.26]
The final result

Electron temperature perturbation

-0.280
-0.140
0.000
0.140
0.280
How to get help when you run into trouble

- VisIt Users Mailing List
  - Address: visit-users@elist.ornl.gov
  - Info: https://elist.ornl.gov/mailman/listinfo/visit-users
  - Archive: https://elist.ornl.gov/pipermail/visit-users/

- VisIt Users Wiki
  - http://www.visitusers.org

- Reference Manuals
  - https://wci.llnl.gov/simulation/computer-codes/visit/
Manuals & Other Documentation

- Getting Started Manual
- Users Manual
- Python Interface
- Getting Data Into VisIt
- VisIt Class Slides
- VisIt Class Exercises
Hands on tutorial

- Run on NERSC
  - ssh -Y username@carver.nersc.gov
  - /usr/common/graphics/visit/bin/visit

- Download VisIt and install it on your local machine
  - https://wci.llnl.gov/simulation/computer-codes/visit/executables/