Visualizing BOUT++ data with VisIt

BOUT++ Mini Workshop 2015
December 18, 2014

Eric Brugger
Overview

- Overview of VisIt (10 min)
- BOUT data and VisIt (5 min)
- Tutorial (30 min)
VisIt is an open source, richly featured, turnkey application for scientific data analysis and vis

- 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.
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**

```
D3D_144382.02510_x516y64_psi080to120mer.nc
heatflux DIIID144382_2500_nosheath_t770.nc
ted_DIIID144382_2500_nosheath_t770.nc
```

**Solution files**

```
edge83{brugger}39: ncdump -h te_DIIID144382_2500_nosheath_t770.nc
netcdf te_DIIID144382_2500_nosheath_t770 {
    dimensions:
        x = 155 ;
        strlen = 42 ;
        strlen2 = 30 ;
        strlen3 = 3 ;
        z = 64 ;
        y = 64 ;
        x2 = 516 ;
        t = 155 ;
    variables:
        float timestep(x) ;
        double timescale ;
        char gridname(strlen) ;
        char variablename(strlen2) ;
        char unit(strlen3) ;
        float te(t, x2, y, z) ;
        short zperiod ;
}
```

**Name of grid file**

```
edge83{brugger}40: ncdump -v gridname te_DIIID144382_2500_nosheath_t770.nc
netcdf te_DIIID144382_2500_nosheath_t770 {
    data:
        gridname = "D3D_144382.02510_x516y64_psi080to120mer.nc" ;
}
```
The format of the grid file

```c
edge83{brugger}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
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.
Tutorial information

- The tutorial assumes some familiarity with VisIt.
  - The following tutorial goes over the basics of VisIt.

- Link to the tutorial content
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