VTK Regions tutorial

This tutorial illustrates how to save simulation results in VTK files, and define sub regions to limit the data range to slices and small rectangular regions. The idea is to save small regions data at a high frequency and large regions data at a lower frequency to make the post-processing more convenient. The method described here can be used only with Cartesian coordinate system (with and without cut-cells). The term VTK refers to a family of files that include a series of VTU (transient) files and a PVD file that stores the simulation time associated with each VTU file. Typically, the PVD file is opened with Paraview for post-processing. Once loaded, Paraview can loop through all VTU files. This tutorial assumes the user is familiar with basic MFIX operation.

A python script `pvd_ezplot.py` was developed to help post-process the data for very small regions, typically containing one or only a few cells. Transient data is extracted from the VTK files, and is plotted on the screen for quick inspection. This script was developed and tested with python 2.7.9, and requires the python vtk library.

The VTK region feature and python script was made available in the 2015-1 Release.

VTK Regions

Figure 1 shows the entire vessel geometry. The vessel has a rectangular shape with an internal obstacle. The definition of the overall geometry and wall boundary conditions are shown in the snippets accompanying Figure 1. The actual simulation setup is not important and details can be found in the mfix.dat file.

To activate the VTK file option, use **`WRITE_VTK_FILES = .TRUE.`** in mfix.dat. Since many VTU files will be generated, an option is given to save the VTU files in a dedicated folder. Setting **`VTU_DIR = 'VTU'`**, will save all VTU files in the subdirectory (below the run directory) **`VTU`**. This directory must be present before the simulation starts. If **`VTU_DIR`** is not defined, all files will be stored in the run directory. PVD files are always saved in the run directory.

VTK region keywords are arrays. The first index corresponds to the region being defined. Currently, there is a maximum of 100 regions that can be defined. If **`WRITE_VTK_FILES = .TRUE.`** and there are no explicit region defined, then a single region comprising the entire geometry is defined by default.

The extant of a VTK region is defined similar to initial or boundary conditions, with the keywords **`VTK_X_w, VTK_X_e, VTK_Y_s, VTK_Y_n, VTK_Z_b`**, and **`VTK_Z_t`**. The data will be saved in a series of files, which base name is defined with **`VTK_FILEBASE`**. The time interval at which data is saved is defined with **`VTK_DT`**.

For example, **VTK_DT(1) = 0.25** and **VTK_FILEBASE(1) = 'BUB3D'**, will generate the following files in the **VTU** directory (when **TSTOP=1.0**): **BUB3D_0000.vtu, BUB3D_0001.vtu, BUB3D_0002.vtu, BUB3D_0003.vtu,** and **BUB3D_0004.vtu**, corresponding to times 0.0, 0.25, 0.50, 0.75 and 1.0 seconds, respectively. The VTU files do not store the time information. Instead, the PVD file links each VTU file with its corresponding simulation time. The PVD file **BUB3D.pvd** is saved in the run directory. The idea is to hide the VTU files and leave the PVD file easily accessible. To visualize the data, the PVD file (here **BUB3D.pvd**) can be loaded into Paraview. This is not covered in this tutorial.

The bottom snippet in Figure 1 shows how to define VTK region#1 (entire domain), saved every 0.25 second. The list of variables saved in the VTU files is defined through **VTK_VARLIST(1,1:4) = 1 2 3 4**, where each item in the list correspond to one or several pre-defined variables:

1 : Void fraction (EP_g)
2 : Gas pressure, solids pressure (P_g, P_star)
3 : Gas velocity (U_g, V_g, W_g)
4 : Solids velocity (U_s, V_s, W_s)
5 : Solids density (ROP_s)
6 : Gas and solids temperature (T_g, T_s)
7 : Gas and solids mass fractions (X_g, X_s)
8 : Granular temperature (Theta_m)
9 : User defined scalars
10 : Reaction Rates
11 : Turbulence quantities (k and epsilon)
12 : Gas Vorticity magnitude and Lambda_2 (VORTICITY, LAMBDA_2)
100: Processor assigned to scalar cell (Partition)
101: Boundary condition flag for scalar cell (BC_ID)

Figure 2 shows how to define a vertical slice, i.e in the XY plane (same value for **VTK_Z_b** and **VTK_Z_t**) and Figure 3 shows how to define a series of horizontal slices with the keyword **VTK_NYS** (number of slices in the y-direction). Similar keywords would apply for slices in other directions (**VTK_NXS** and **VTK_NZS**).

Figure 3 shows examples of very small regions (1 cell and 8 cells) that can be used as probes. VTK region #4 (“PROBE1”) will save void fraction, gas pressure, gas and solids velocity at a single point, every 0.01 second. VTK region#5 (“PROBE2”) will save void fraction gas pressure over 8 cells every 0.01 second. Here the variables are defined explicitly with logical keywords for each variable.

Python script pvd_ezplot.py

For VTK regions containing a few cells, a python script called `pvd_ezplot.py` can be used to plot the transient data. The script is located in `/mfix/tools/python`. The script reads a PVD file, and plots the cell(s) data as a function of time for each variable in the VTU files. If there are more than one cell, the mean value is also plotted. The plots are saved in PNG files and the transient data is also saved into text files (one file per variable, one column per cell).

Various options are available to clip the time range, process the data and change the look of the plots. This script is not meant to be a comprehensive post-processing tool, but can be useful to quickly visualize data.

Once the simulation results are generated, the script is invoked from the command line. From the run directory, enter the following command to plot the PROBE1 data:

```
python ~/mfix/tools/python/pvd_ezplot.py PROBE1.pvd
```

This assumes mfix is installed in the home directory. The actual path may be adjusted. This will generate transient data for the void fraction, gas pressure, gas and solids velocity (Figure 5).

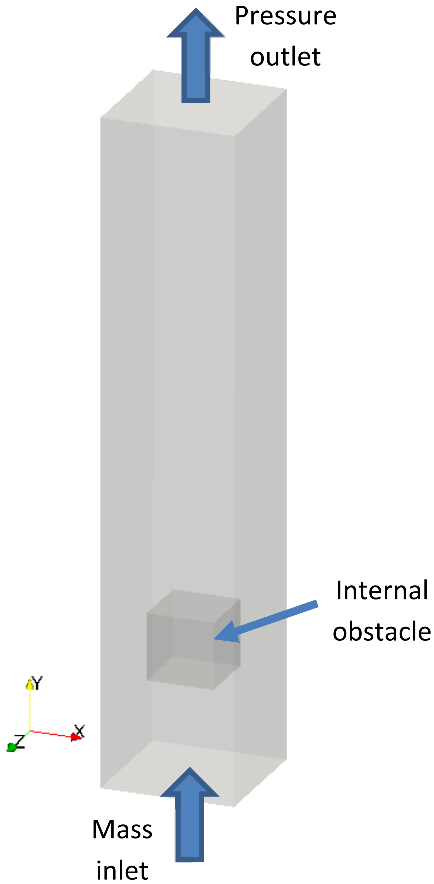
Similarly, enter the following command to plot the PROBE2 data:

```
python ~/mfix/tools/python/pvd_ezplot.py PROBE2.pvd
```

This will generate plots for the void fraction and gas pressure for all 8 cells contained in the probe, as well as the mean value (Figure 6).

The usage summary for the script is given below:

```
usage: pvd_ezplot.py [-h] [--tmin TMIN] [--tmax TMAX] [--maxcells MAXCELLS]
                     [--no-mean] [--no-fig] [--no-png] [--no-dat]
                     [--legend LEGEND] [--frame_alpha FRAME_ALPHA]
                     [--font-family FONT_FAMILY] [--font-weight FONT_WEIGHT]
                     [--font-size FONT_SIZE] [--linewidth LINEWIDTH]
                     filename

positional arguments:
  filename              PVD file to be parsed (include path).By default, the
                        data is plotted on the screen, saved in .png file, and
                        exported in .dat file (one file per variable, one
                        column per cell). The mean value of all cells is also
                        computed, plotted and saved.

optional arguments:
  -h, --help            show this help message and exit
  --tmin TMIN           Lower limit of time range (seconds).
  --tmax TMAX           Upper limit of time range (seconds).
  --maxcells MAXCELLS   Maximum number of cells. When the number of cells is
                        larger than maxcells, cell data is not stored in dat
                        files nor plotted (default =20). This limit is used to
                        prevent unreasonable data processing for large cell
                        count. Setting maxcells to 0 will result in only the
                        mean value to be stored and plotted unless --no-mean
                        is used (nothing is stored or plotted in that case).
  --no-mean             Do not compute mean.
  --no-fig              Do not show Figure.
  --no-png              Do not save Figure.
  --no-dat              Do not save Data.
  --legend LEGEND       Plot legend location (default='best', Acceptable
                        values = 'right', 'center left', 'upper right', 'lower
                        right', 'best','center', 'lower left', 'center right',
                        'upper left', 'upper center', 'lower center'.
  --frame_alpha FRAME_ALPHA
                        Plot legend frame transparency (0.0=tranparent,
                        1.0=opaque, default=1.0)
  --font-family FONT_FAMILY
                        Plot font family (default=sans-serif)
  --font-weight FONT_WEIGHT
                        Plot font weight (default=normal)
  --font-size FONT_SIZE
                        Plot font size (default=16)
  --linewidth LINEWIDTH
                        Plot line width (default=3)
```

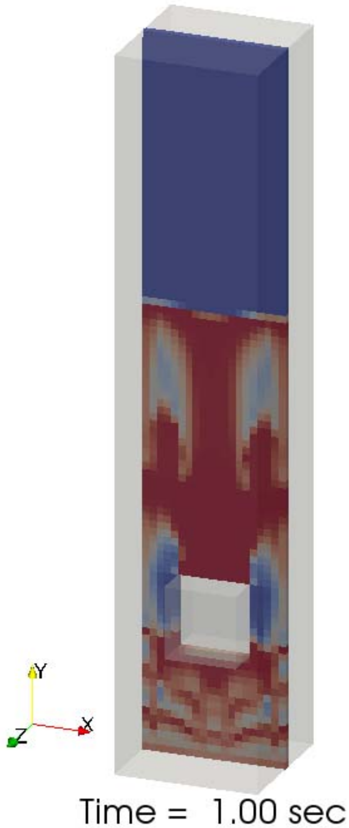



```
COORDINATES = 'CARTESIAN'
XLENGTH     = 0.20     ! (m)
IMAX        = 20
YLENGTH     = 1.00     ! (m)
JMAX        = 100
ZLENGTH     = 0.20     ! (m)
KMAX        = 20
```

```
! Vessel walls + internal obstacle
!-----------------------------------------------------------//
BC_X_w(3:7)  = 0.00   0.20   0.00   0.00   0.05     ! (m)
BC_X_e(3:7)  = 0.00   0.20   0.20   0.20   0.15     ! (m)
BC_Y_s(3:7)  = 0.00   0.00   0.00   0.00   0.15     ! (m)
BC_Y_n(3:7)  = 1.00   1.00   1.00   1.00   0.25     ! (m)
BC_Z_b(3:7)  = 0.00   0.00   0.00   0.20   0.05     ! (m)
BC_Z_t(3:7)  = 0.20   0.20   0.00   0.20   0.15     ! (m)

BC_TYPE(3:7) = 'NSW'  'NSW'  'NSW'  'NSW'  'NSW'
```

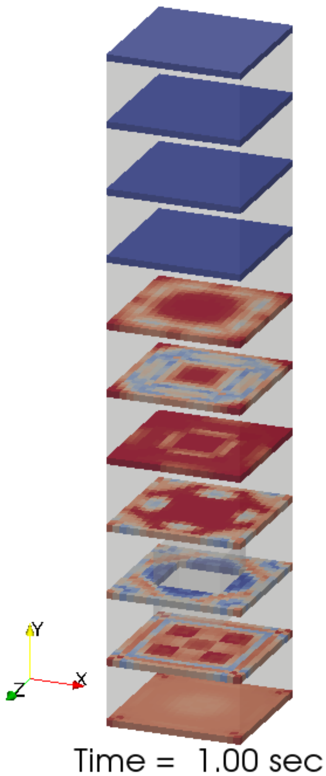
```
! First VTK region is the entire geometry, saved
! every 0.25 sec.
! The variables are specified by a variable list
! (1=EP_G, 2=P_G and P_S, 3=Gas velocity, 4=Solids Velocity)

VTK_DT(1)          = 0.25
VTK_FILEBASE(1)    = 'BUB3D'
VTK_X_w(1)         = 0.00
VTK_X_e(1)         = 0.20
VTK_Y_s(1)         = 0.00
VTK_Y_n(1)         = 1.00
VTK_Z_b(1)         = 0.00
VTK_Z_t(1)         = 0.20
VTK_VARLIST(1,1:4) = 1 2 3 4
```

Figure 1. Entire domain and definition of VTK region #1.



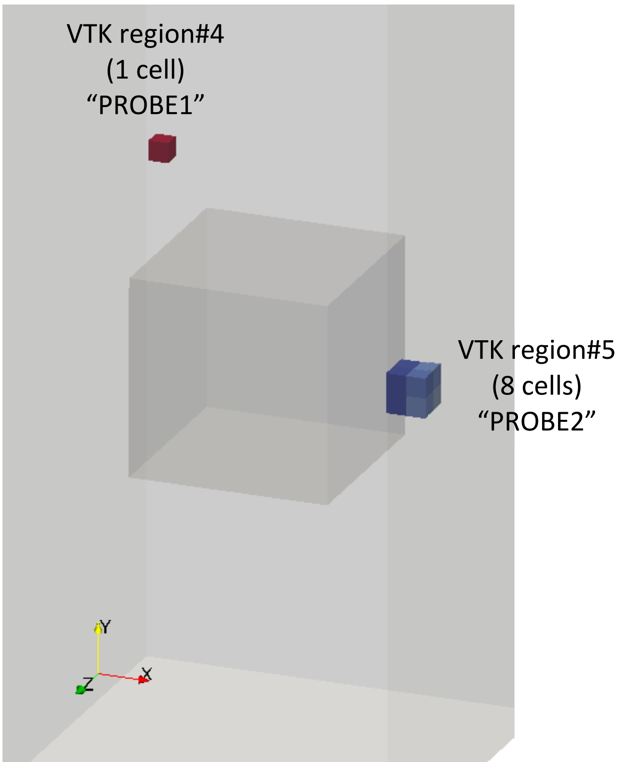

```
! Second VTK region is a vertical slice, saved every 0.10 sec.
! The variables are specified by a variable list
! (1=EP_G, 2=P_G and P_S, 3=Gas velocity, 4=Solids Velocity)
VTK_DT(2)          = 0.10
VTK_FILEBASE(2)    = 'VSLICE'
VTK_X_w(2)         = 0.00
VTK_X_e(2)         = 0.20
VTK_Y_s(2)         = 0.00
VTK_Y_n(2)         = 1.00
VTK_Z_b(2)         = 0.10
VTK_Z_t(2)         = 0.10
VTK_VARLIST(2,1:4) = 1 2 3 4
```

Figure 2. Definition of VTK region #2 (one vertical slice).




```
! Third VTK region is a collection of 11 horizontal slices,
! saved every 0.10 sec.
! The variables are specified by a variable list
! (1=EP_G, 2=P_G and P_S, 3=Gas velocity, 4=Solids Velocity)
VTK_DT(3)          = 0.10
VTK_FILEBASE(3)    = 'HSLICES'
VTK_X_w(3)         = 0.00
VTK_X_e(3)         = 0.20
VTK_Y_s(3)         = 0.00
VTK_Y_n(3)         = 1.00
VTK_Z_b(3)         = 0.00
VTK_Z_t(3)         = 0.20
VTK_NYS(3)         = 11
VTK VARLIST(3,1:4) = 1 2 3 4
```

Figure 3. Definition of VTK region #3 (11 horizontal slices).




```
! Fourth VTK region is a single cell probe,
! saved every 0.01 sec. The variables are
! specified explicitly.
VTK_DT(4)        = 0.01
VTK_FILEBASE(4) = 'PROBE1'
VTK_X_w(4)       = 0.05
VTK_X_e(4)       = 0.05
VTK_Y_s(4)       = 0.30
VTK_Y_n(4)       = 0.30
VTK_Z_b(4)       = 0.10
VTK_Z_t(4)       = 0.10
VTK_EP_G(4)      = .TRUE.
VTK_P_G(4)       = .TRUE.
VTK_VEL_G(4)     = .TRUE.
VTK_VEL_S(4,1)   = .TRUE.
```

```
! Fifth VTK region is an 8-cell probe,
! saved every 0.01 sec. The variables are
! specified explicitly.
VTK_DT(5)        = 0.01
VTK_FILEBASE(5) = 'PROBE2'
VTK_X_w(5)       = 0.175
VTK_X_e(5)       = 0.185
VTK_Y_s(5)       = 0.195
VTK_Y_n(5)       = 0.205
VTK_Z_b(5)       = 0.095
VTK_Z_t(5)       = 0.105
VTK_EP_G(5)      = .TRUE.
VTK_P_G(5)       = .TRUE.
```

Figure 4. Definition of VTK region #4 and #5 (probes).

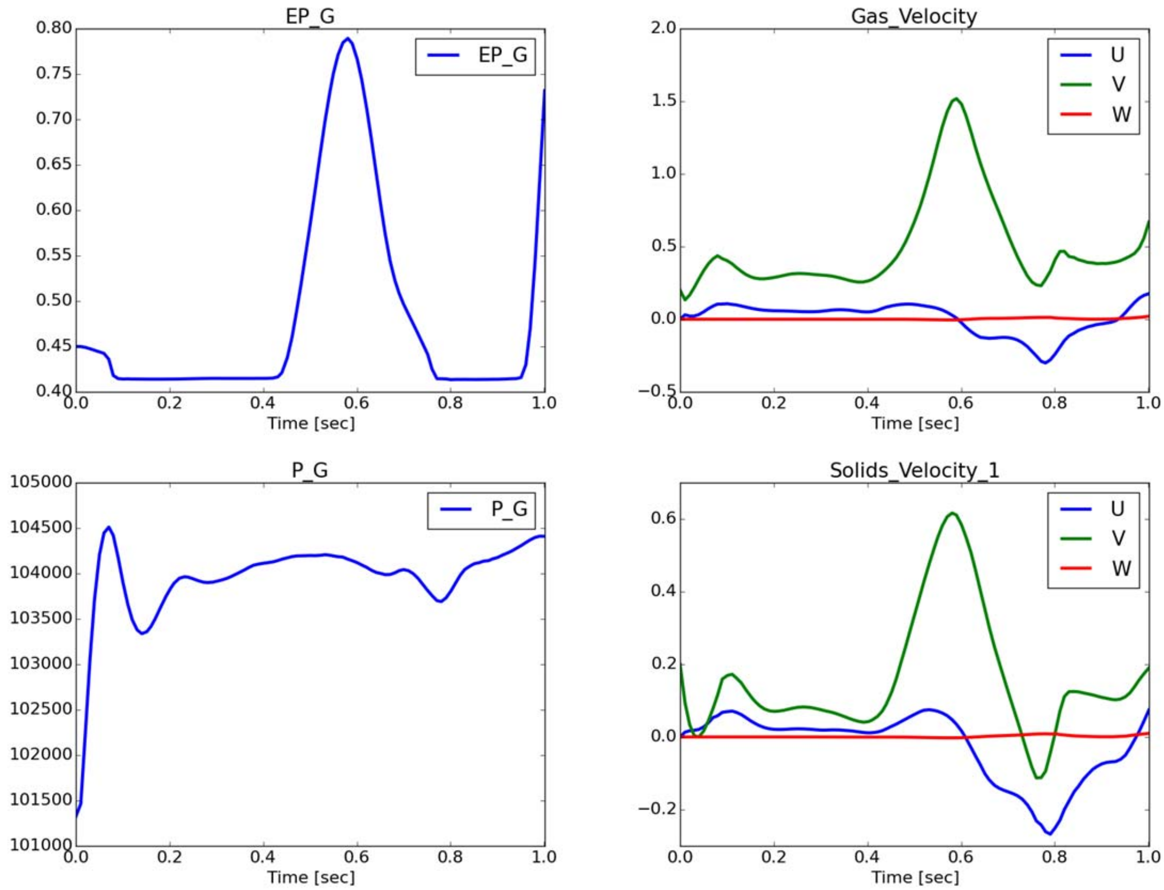


Figure 5. Transient data for PROBE1.

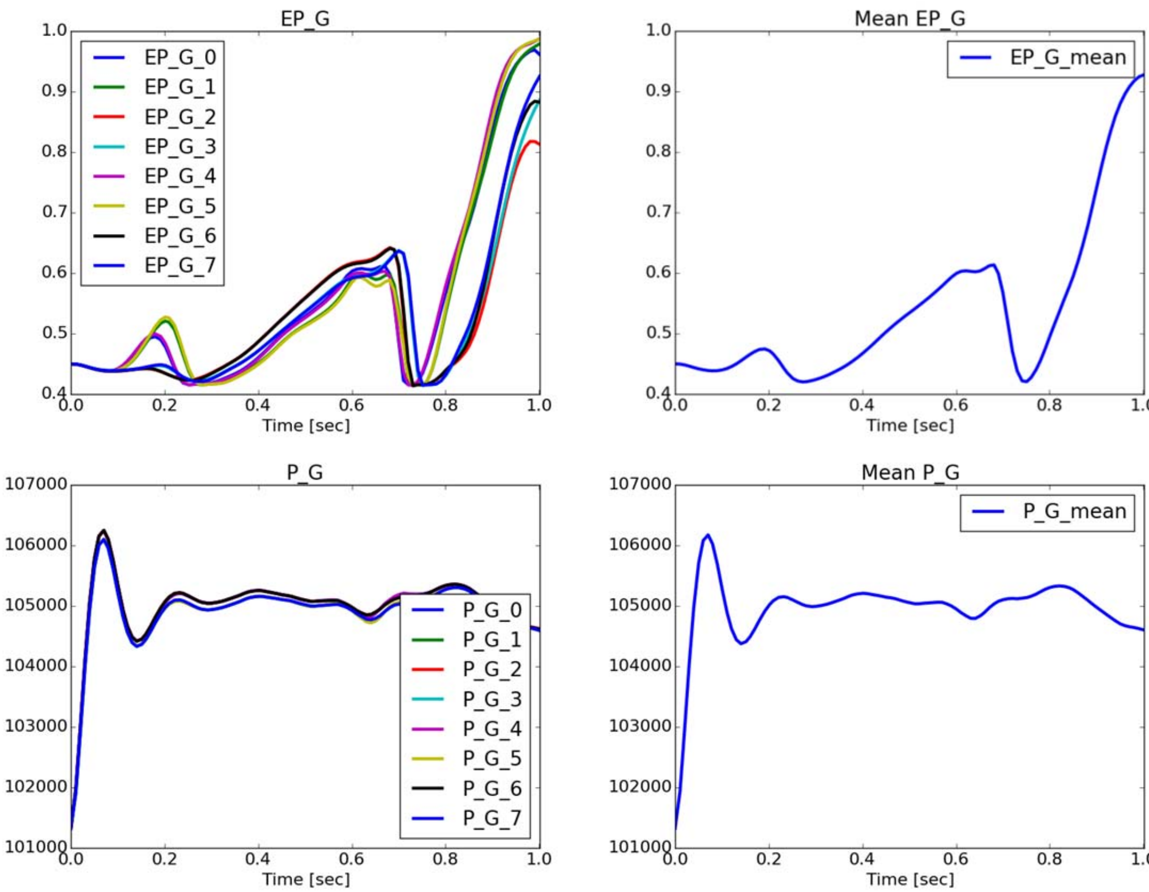


Figure 6. Transient data for PROBE2.