

# pkdgrav\_current ss2vtk.c & walls2vtp.c Documentation

Last Update: 11/7/16

Author: Yun Zhang

Contact Info:

Department of Astronomy

University of Maryland

College Park MD 20742

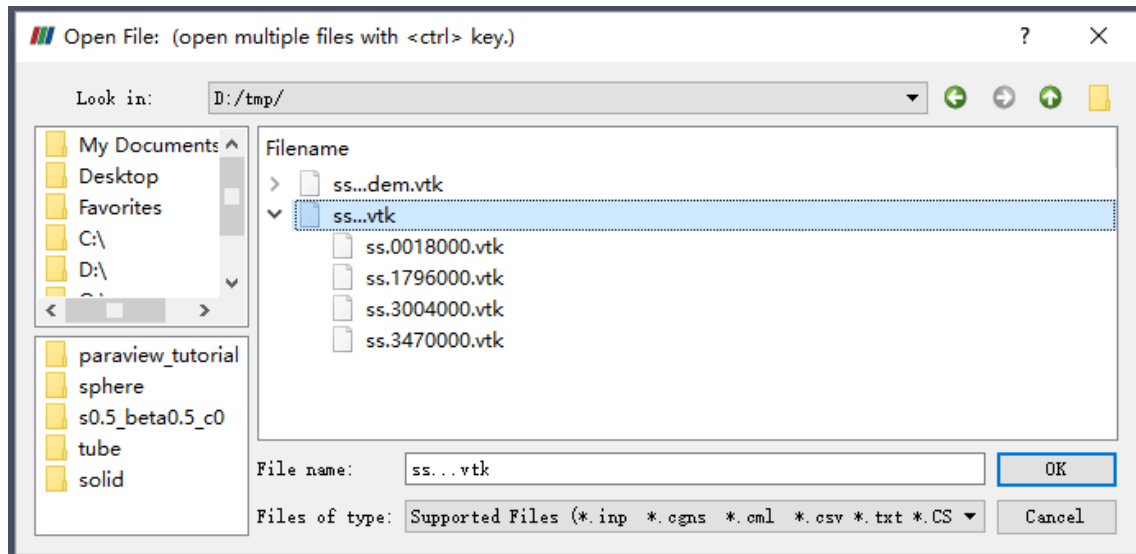
E-mail: [yun@astro.umd.edu](mailto:yun@astro.umd.edu)

## 1 OVERVIEW

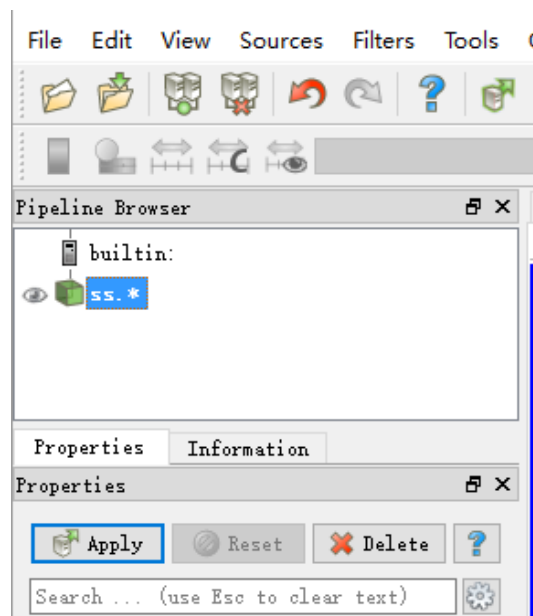
The specialized post-processing tool, *Paraview* is based on the *Visualization Toolkit*, which defines formats for saving various types of data. Some of them (with the *.vtk* and *.vtp* extension) are used here. The function *ss2vtk.c* is designed to convert binary data file *\*.ss* to *\*.vtk* format, for visualization of particle data and force chain distribution in *Paraview*, and the function *walls2vtp.c* is designed to convert wall data file to *\*.vtp* format, for visualization of wall data in *Paraview*. Every *\*.vtp* file is correlated to the same time as the *\*.vtk* file, which can be used to make a real-time movie in *Paraview*. The source codes are located in *pkdgrav\_current/src/ss*.


## 2 Analyze particles data

By typing “ss2vtk \*.ss” in the command line, the binary particle data file *\*.ss* can be converted to *\*.vtk* format with the same file name (which also works for a series of input files). The generated *\*.vtk* files can be directly open in *Paraview*. The open dialogue automatically collapses numbered files in one, making it easy to select all of them:



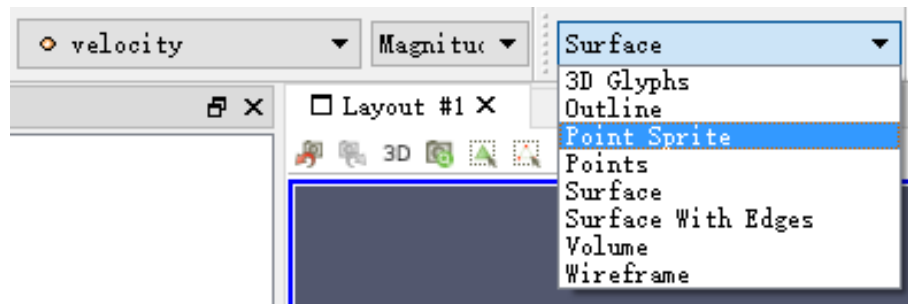
Click on the “Apply” button in the “Properties” sub-window to make loaded objects visible. You can see tree of displayed objects in the “Pipeline browser”:



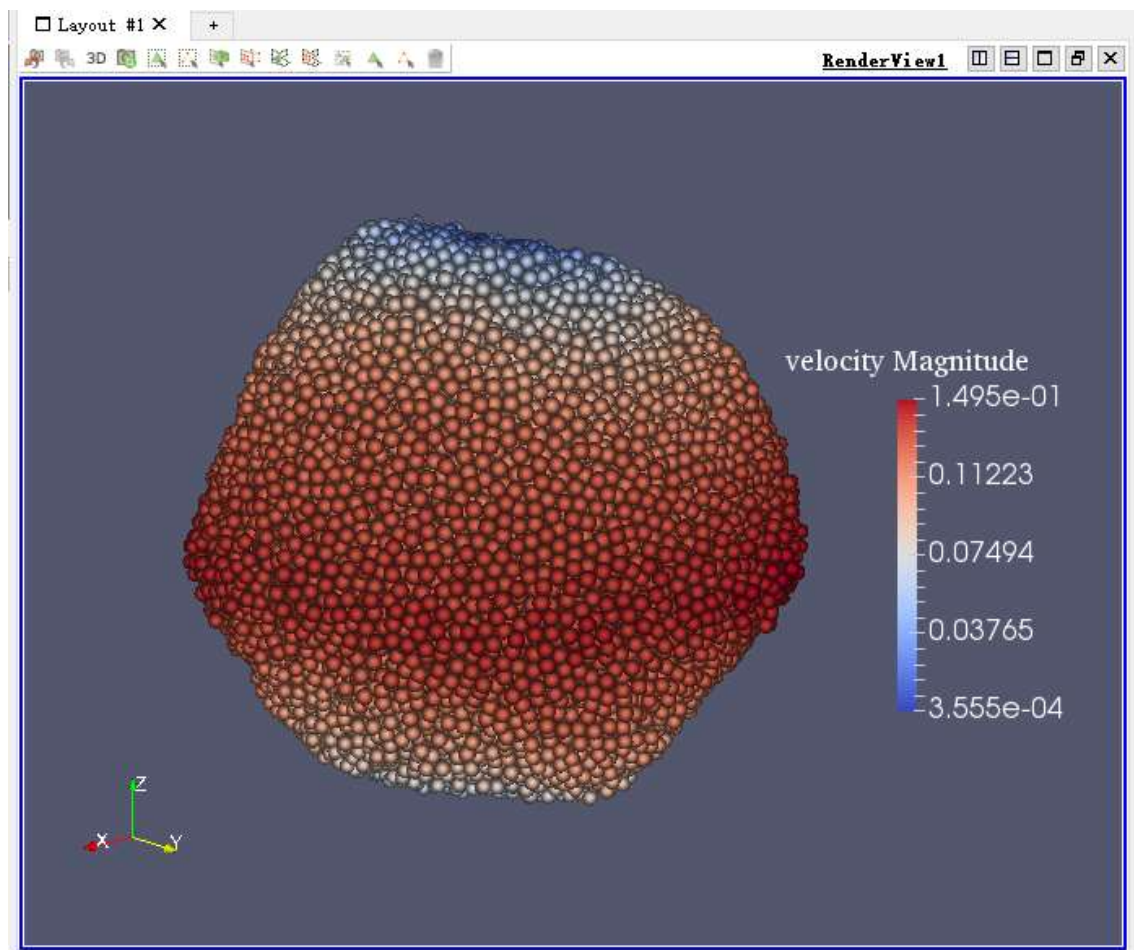
For now, you can get a 3D model in the “RenderView” sub-window. But, all the spheres will only appear as points. To make them look as spheres, you have to add “glyph” to the ss.\* item in the pipeline using the  icon. Then set (in the “Properties”): 1. “Glyph type” to Sphere; 2. “Scalars” to “Radius”; 3. “Scale mode” to Scalar (Scalar is set above to be the radii value saved in the file, therefore spheres with radius 1 will be scaled by their true radius); 4. “Set scale factor” to 1; 5. Select “All points” for “Glyph mode” in “Masking” (they make some particles not to be rendered for performance reasons, controlled by the “Maximum Number of Points”).

Another way to display spheres is to use *PointSprite* plugin (Paraview Version 4.2.0/4.4.0 contains the function; Other versions may also work, but need tests). This technique requires much less RAM in comparison to *Glyphs*: 1. “Tools -> Manage

Plugins”; 2. “PointSprite\_Plugin -> Load selected -> Close”; 3. Open vtk-files; 4. “Representation -> Point Sprite”; 5. “Point Sprite -> Scale By -> radii”; 6. “Edit Radius Transfer Function -> Proportional -> Multiplier = 1.0 -> Close”. Note that when you use “Scale By” radii, the state variable displaying in the Renderview need to be “radius”; otherwise, it cannot be scaled properly. Then, when you get the right radii for the particles in the Renderview, you can change back to the state variable you actually want to display.



After clicking “Apply”, spheres will appear. They will be rendered over the original white points, which you can disable by clicking on the eye icon next to “ss.\*” in the Pipeline browser. You can select the state variable you want to display (i.e., “velocity” in this case).

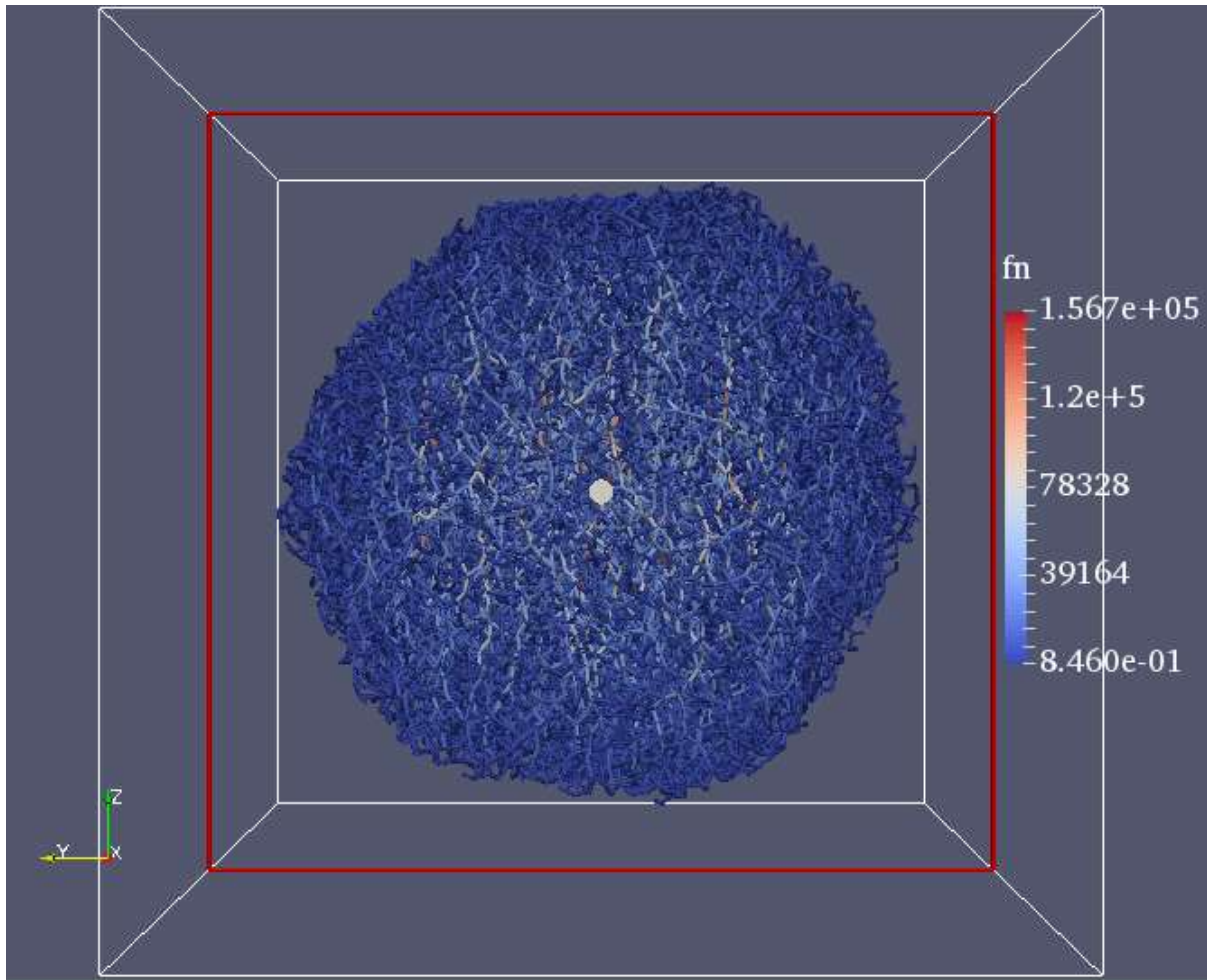


### 3 Analyze force chain distribution

By typing “ss2vtk -c \*.ss” for simulation without walls or “ss2vtk -w \*.ss” for simulation with walls in the command line, a corresponding \*.vtk file containing the force chain information for particles contact (\*pp.vtk) or particles and walls contact (\*pw.vtk) is generated based on the binary particle data file \*.ss and contact information file \*.demdiag (which also works for a series of input files). It should be noted that when using “ss2vtk -c \*.ss” or “ss2vtk -w \*.ss”, an additional file \*.vtk containing the particle information will also be generated (this file is the same as the \*.vtk file obtained from using “ss2vtk \*.ss” except that the total contact force acting on each particle and the coordination number of each particle are recorded in this version). To disable generate this \*.vtk file, you can add “-n” to the command, e.g., “ss2vtk -c -n \*.ss”.

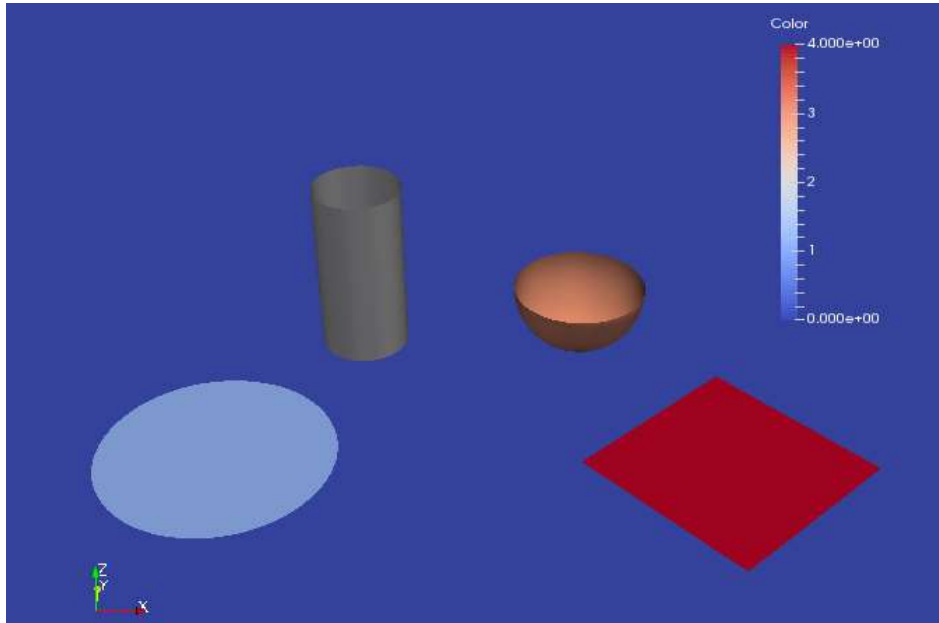
Following the same procedure as described in Section 2, open the generated force chain data file \*pp.vtk or \*pw.vtk in *Paraview* and click on the “Apply” button. For now, you can see the force chain distribution in the “RenderView” sub-window. But, all the chains will only appear as lines. To make them look better, you can use *Tube* filter: 1. “Filters -> Alphabetical -> Tube”; 2. set “Radius” to 3 (or any suitable value) in the “Properties”.

After clicking “Apply”, force chains will appear. To display the force network over the cross-section, choose the “Tube1” item in the Pipeline browser and use “Filters -> Alphabetical -> Clip”, and then click “Apply”.

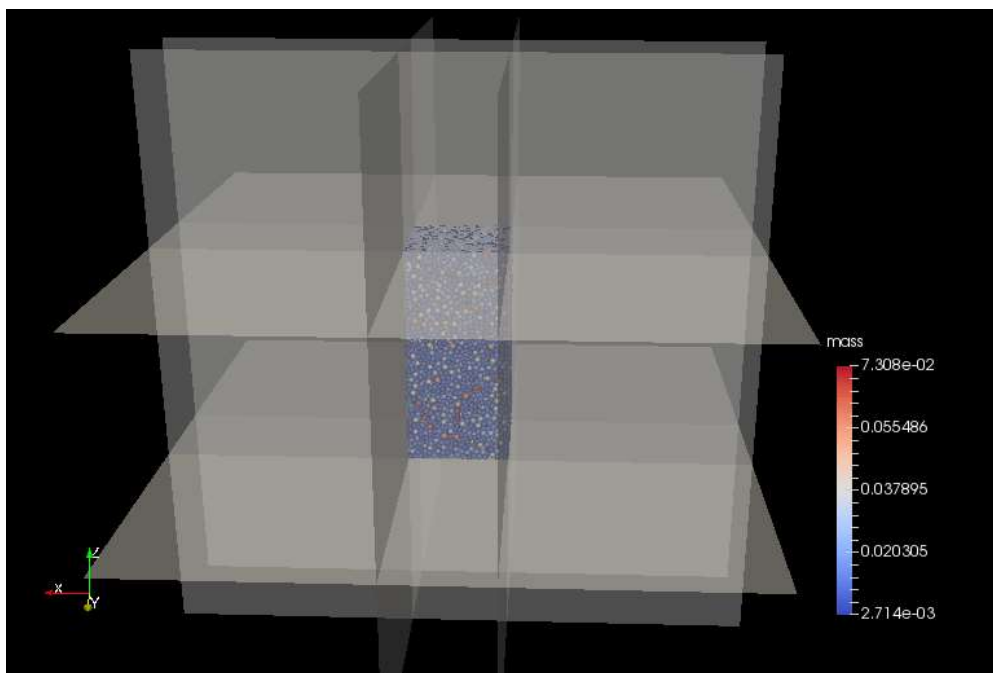


#### 4 Analyze walls data

By typing “walls2vtp walls.dat \*.ss” for simulation without using reacted walls or “walls2vtp -r walls.dat WALLSREACTviz.out \*.ss” for simulation including reacted walls in the command line, a corresponding \*.vtp file containing all the wall information is generated based on the wall information file *walls.dat* (i.e., the input wall file for *pkdgrav*) and the binary particle data file \*.ss (which provides the time and step information). This function also works for a series of input files. The generated \*.vtp files can be directly opened in *Paraview*: (the same example as used in walls.pdf)



You can select the state variable you want to display (i.e., “color” in this above case). The particle data \*.vtk can be visualized in the same time:



In order to observe particles inside of walls, the user can set “Styling -> Opacity” to 0.5 (or any suitable value) in the “Properties”.