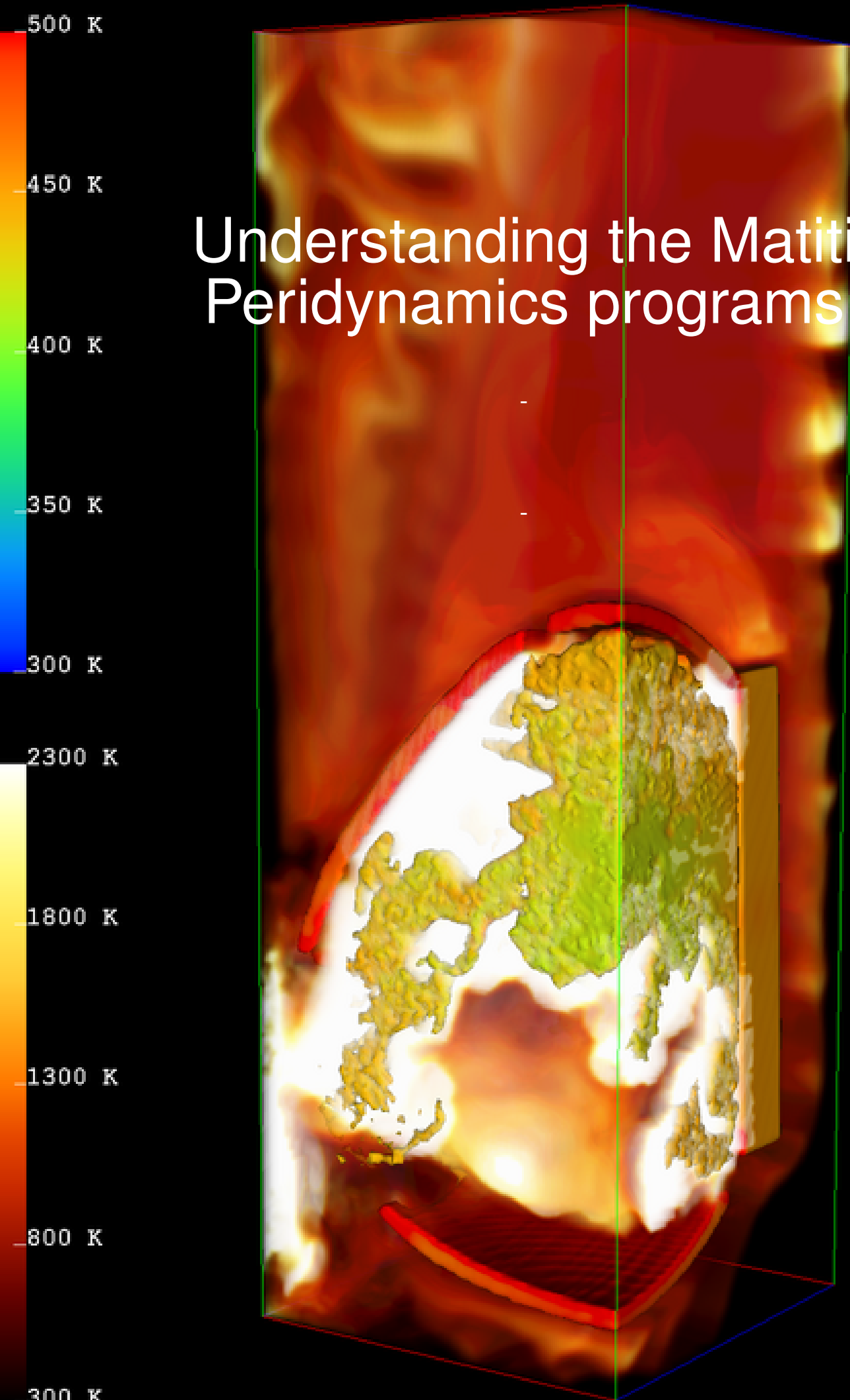
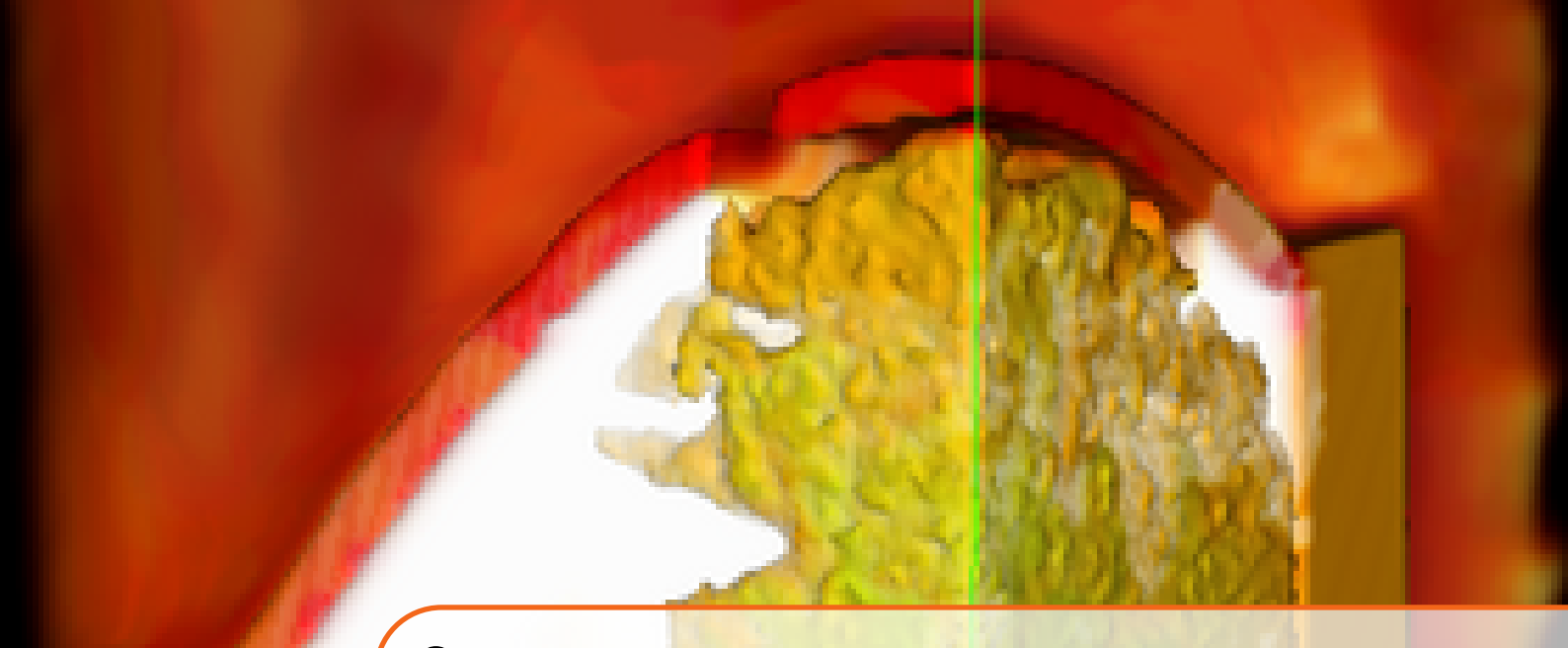


Understanding the Matiti Peridynamics programs



Copyright © 2013 Callaghan Innovation

The contents of this manual can and will change significantly over time. Please make sure that all the information is up to date.



Contents

1	Matiti Structure	5
2	Organization	13

1 — Matiti Structure

In the *ParticleFracturing* directory you can find a directory named as *Matiti*. *matiti* itself consists of other folders like: *src*, *opt*, *runs* and *triangle* as it can be seen in figure 1.1.

- From the *Triangle* directory you need a library with which you can discretize your surface into triangle shaped parts.
- *opt* is a directory that you make in order to make the needed files such as the executable files from your source directory so you can run the program.
- *runs* is where you run your program and save the out-put data (as vtk file format) to be visualised using a proper package like visIt.
- *src* is the source folder where all the input files and programs are gathered in it. This directory again includes other folders and files which are categorized in figure 1.1.

First group which are shown by the blue boxes are:

- *CMake* finds the runtimes for cmake so you can make all the needed files out of the *src* directory into the *opt* directory.
- *Geometry* is where simple geometries such as point, vector, box and polygon

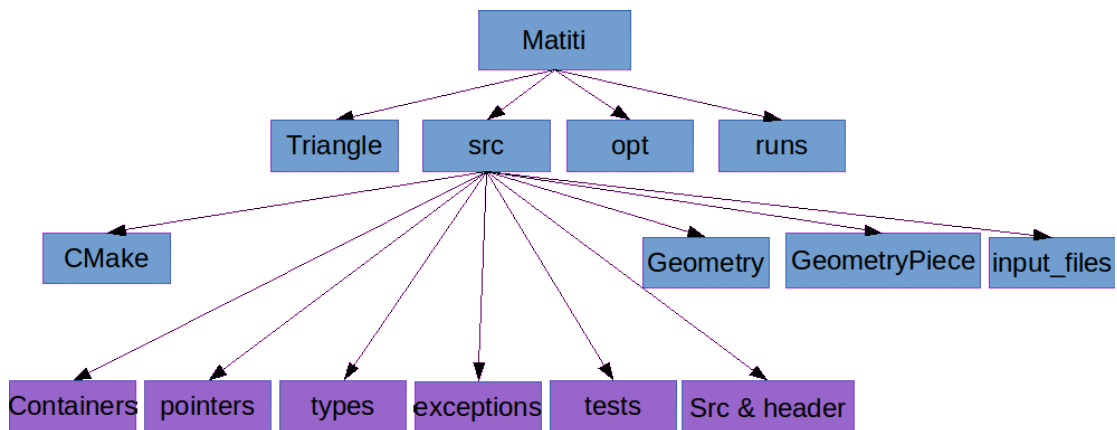


Figure 1.1: Directory Graph for Matiti

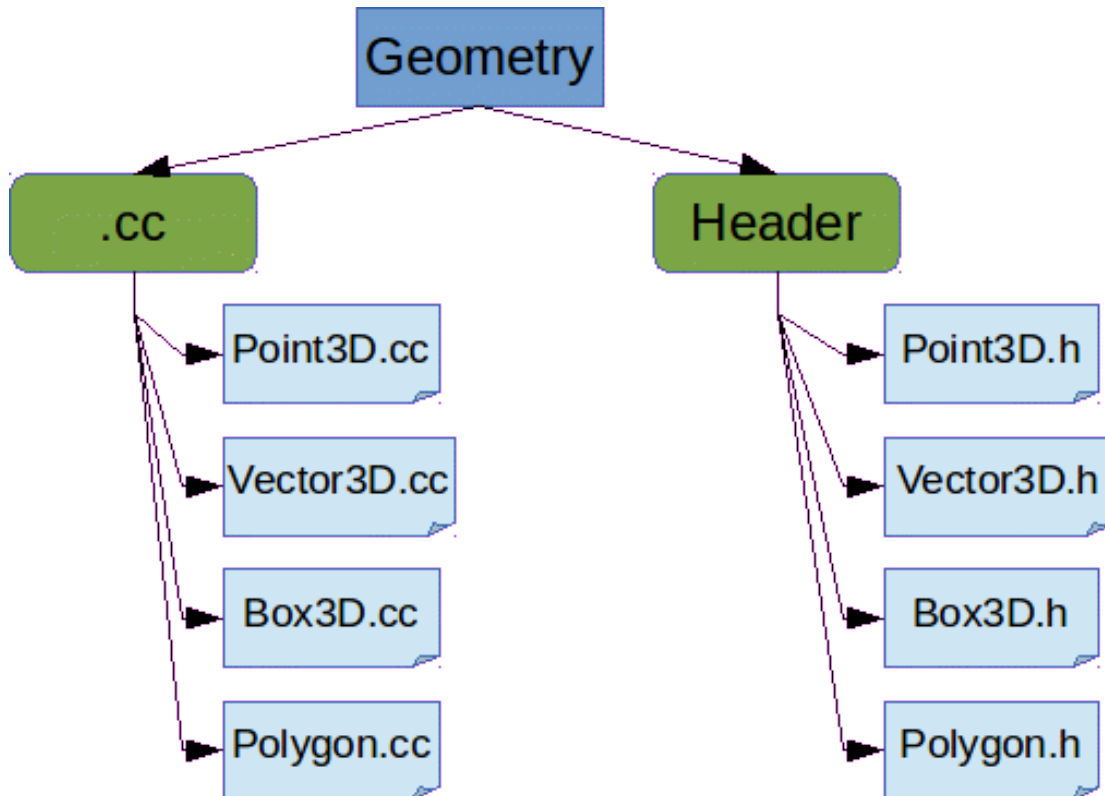


Figure 1.2: Graph for Geometry directory

are made, figure 1.2:

- *GeometryPiece* is a directory in which geometry pieces can be created using two input nodes (as lower and upper) and the elementary geometries introduced in the *Geometry* directory. This directory consists of files as shown in figure 1.3
- *input-files* is a directory in which all your input files have been gathered. These are the files that you need to copy them into the *runs* directory, whenever you want to run them, and from there run it using the executable file in the *opt* directory (\rightarrow *test_peri.*) Organization and the skeleton of these input files are described in the next chapter (\rightarrow **Chapter 2**)

In addition to these directories, there are so many files in the *src*. We have made some groups shown by the violet color. These files are divided and arranged in each group based on their properties and applications:

- *containers* as shown in figure 1.4
- *Pointers* as shown in figure 1.5
- *Types* as shown in figure 1.6
- *Exceptions* as shown in figure 1.7
- *Tests* as shown in figure 1.8
- *Src & header* as shown in figure 1.9

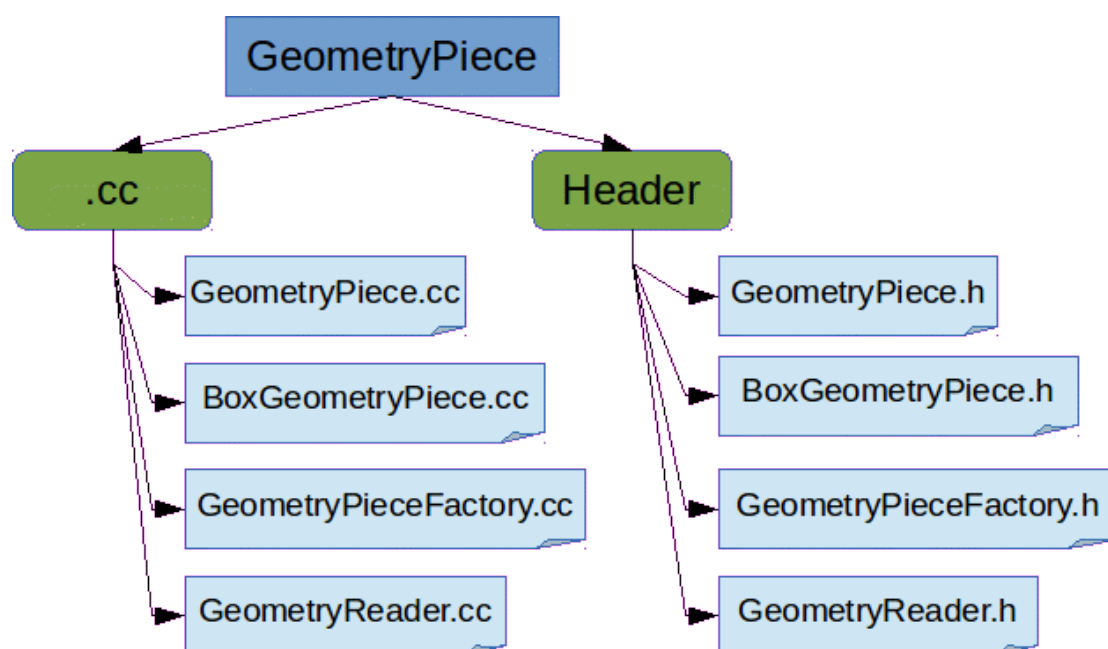


Figure 1.3: Graph for GeometryPiece directory

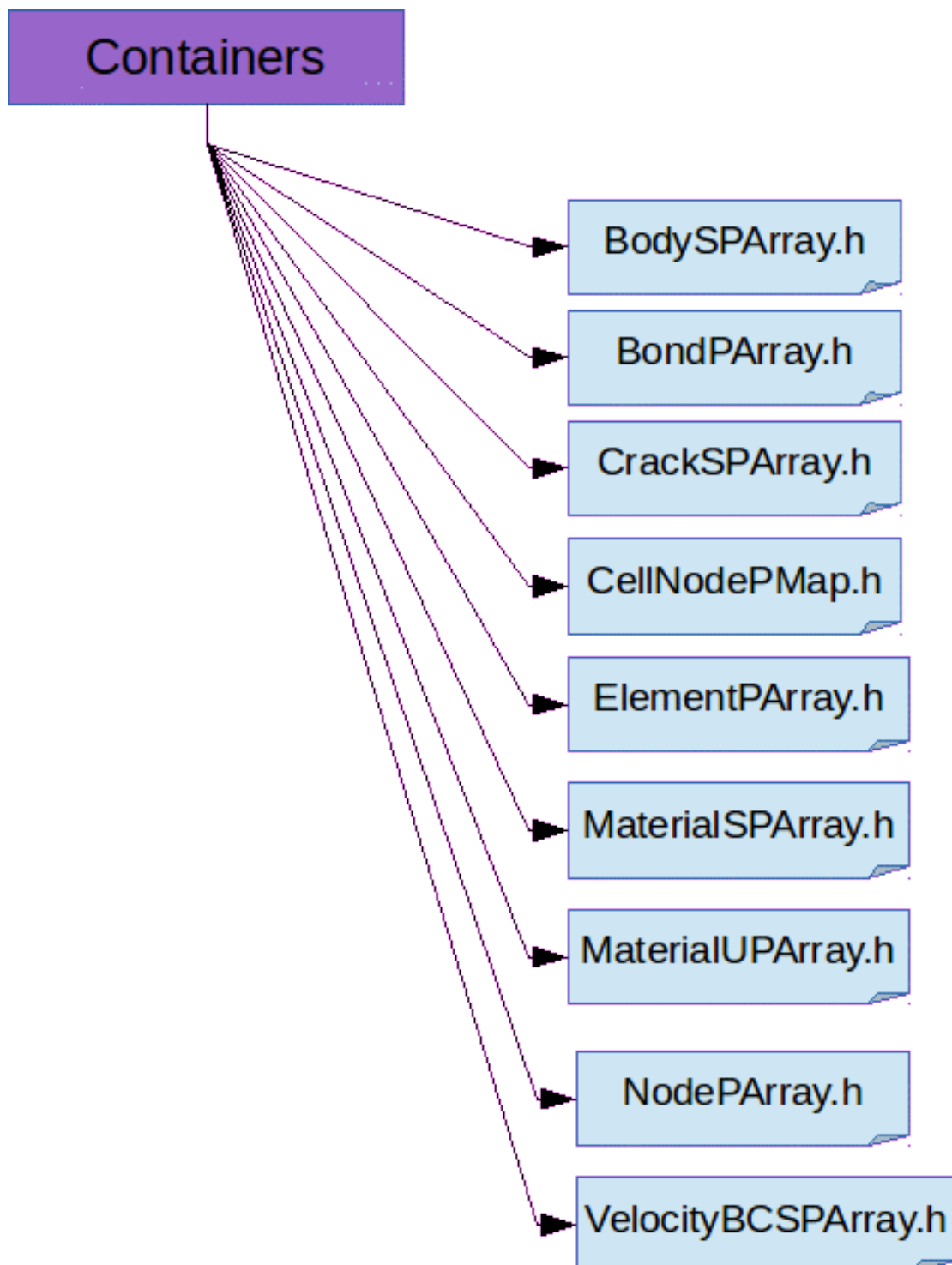


Figure 1.4: Graph for Containers Group

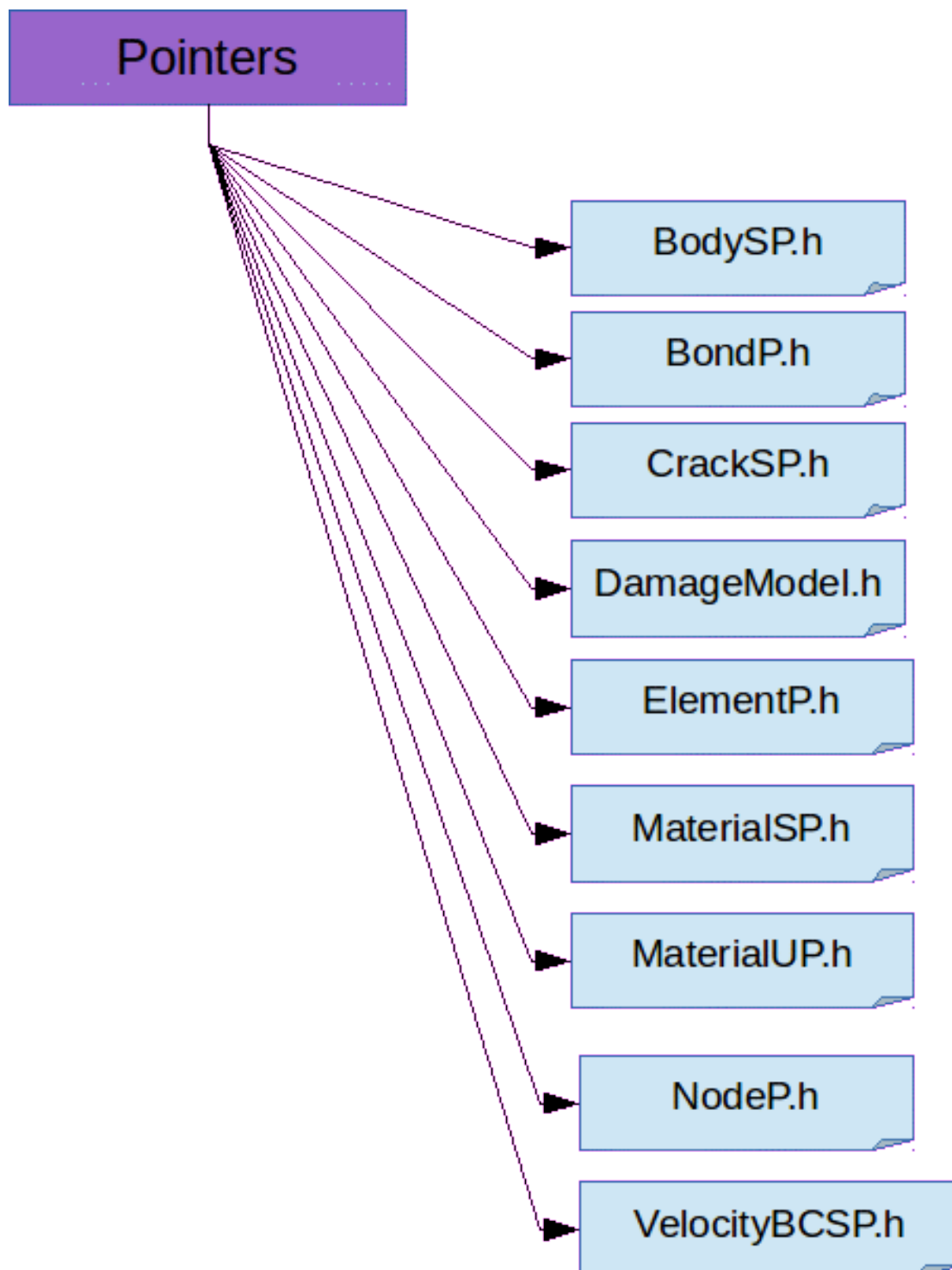


Figure 1.5: Graph for Pointers Group

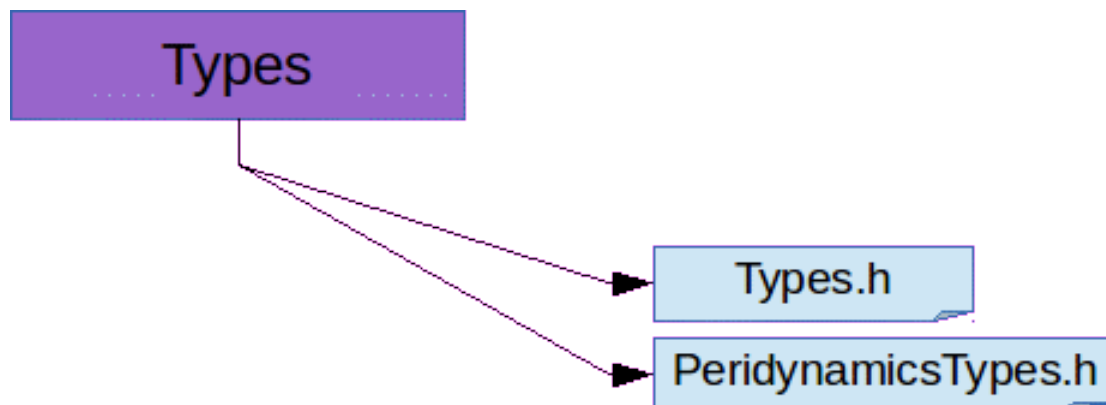


Figure 1.6: Graph for Types Group

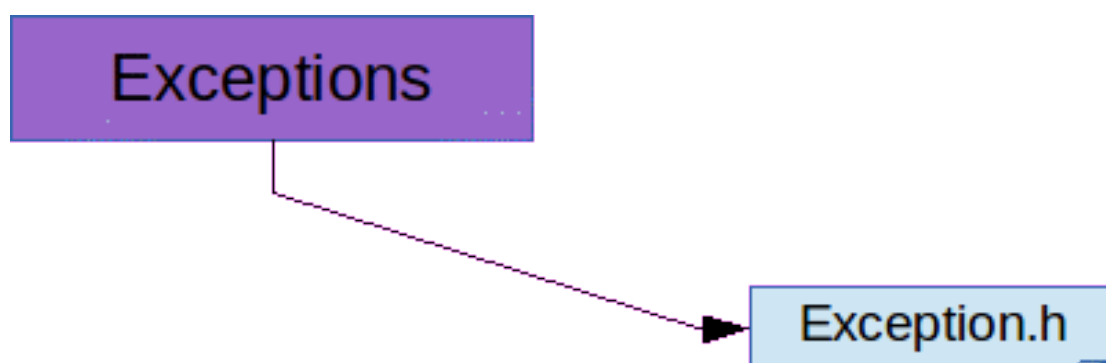


Figure 1.7: Graph for Exceptions Group

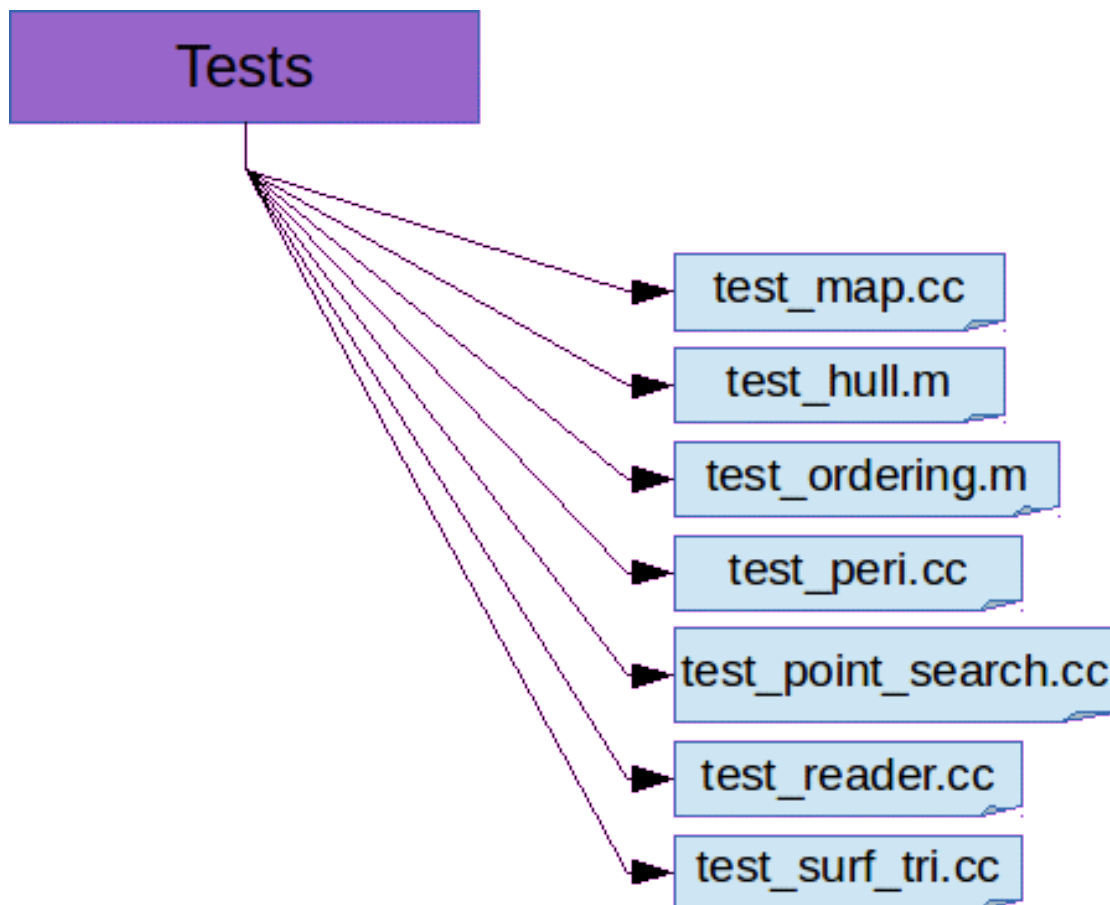


Figure 1.8: Graph for Tests Group

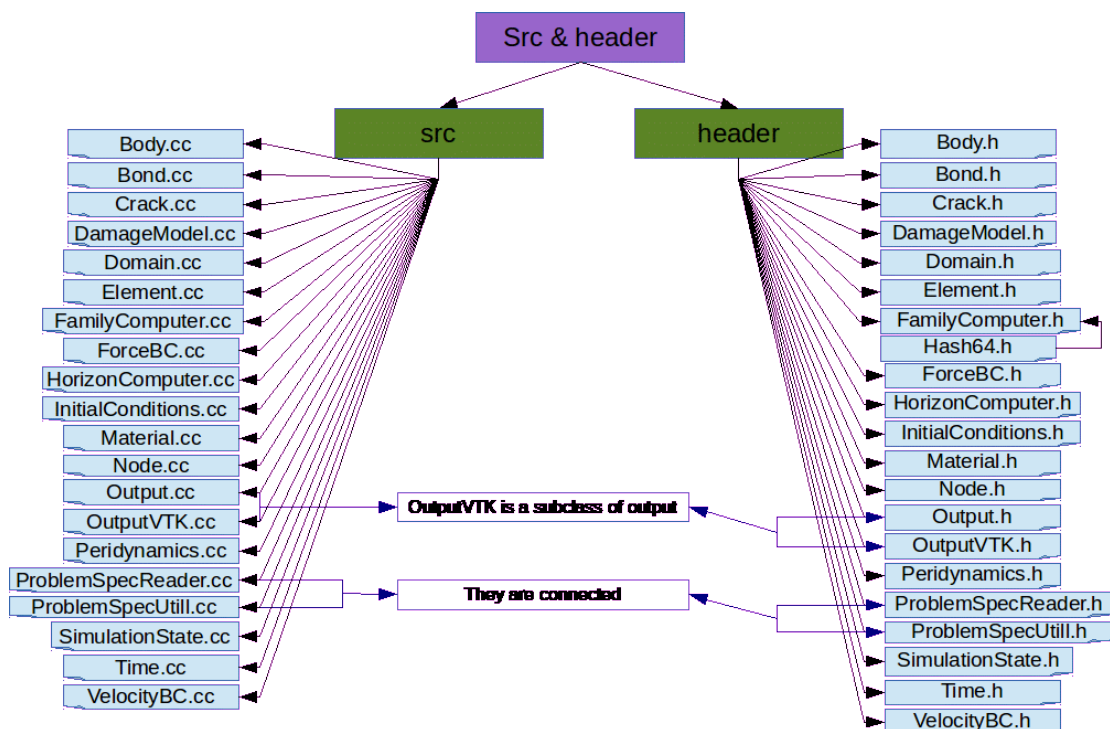


Figure 1.9: Graph for Src & header Group

2 — Organization

The Matiti Peridynamics programs work as described below:

- First it gets an input using *ProblemSpecReader* and *ProblemSpecUtil*
- All the calculations on the data would be done in a mainbody of the program (saved in the input files directory.) This part is called the **Domain**.
- and the out put data would be saved as VTK files using *Output* and *OutputVTK* programs.

All the above items are gathered in figure [2.1](#)

Domain is where you can make your bodies or objects. Before the domain part, in the main body, we set the *time* and *Simulation State* and in the domain part, having the domain points we set the *VelocityBC*. Now in the domain we can make our object(s) or the Body(s) but before that we need to make the *Nodes*.

Actually, the *GeometryPiece* makes *Elements* with which we can have our *Nodes*. Moreover *Material* should be made using the *Damagemodel* and be passed to the *Nodes* through the *Bond*. For making the nodes we also need *FamilyComputer* and *HorizonComputer* to find the neighbours of each node. (See figure [2.2](#))

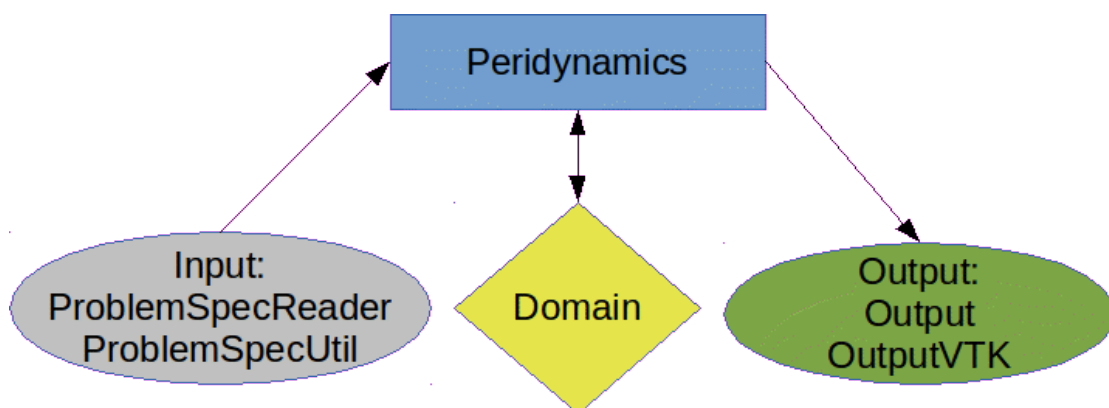


Figure 2.1: Graph for the Peridynamics schematic job

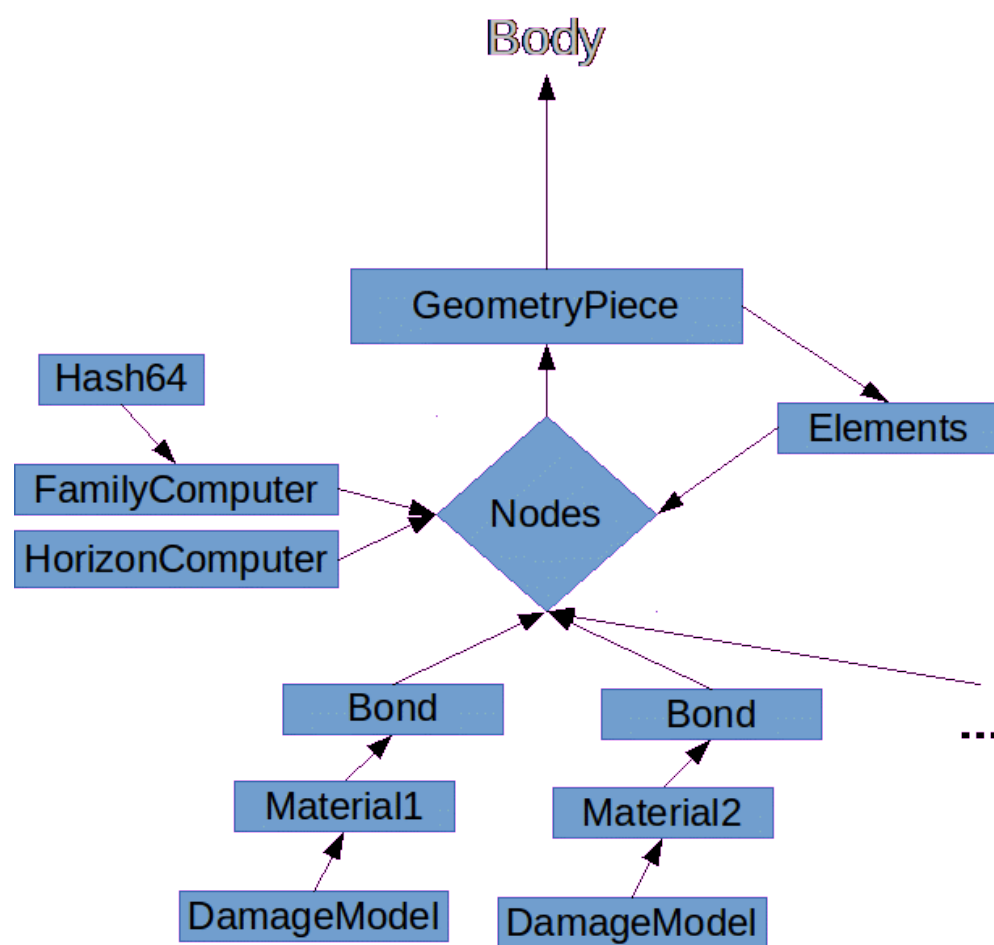


Figure 2.2: Graph for the Nodes

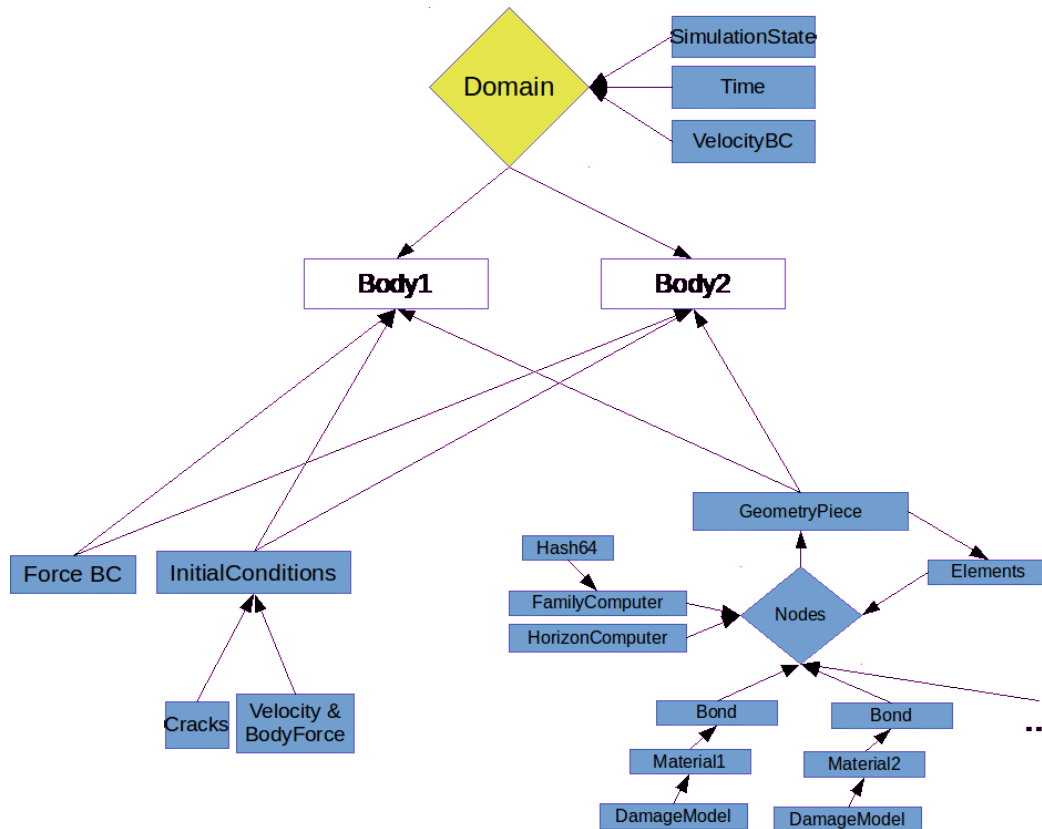


Figure 2.3: Graph for the Domain

Now that we have the nodes we just need to give the *Initial Conditions* and *Force Boundary Conditions* so we can have our body(s). The *Initial Condition* is made of *Cracks*, *initial Velocity* and the *Body Forces* like the gravity. All the information regarding the domain structure and organization is gathered in figure 2.3)