1. Material model
   1) Describe the material that constitutes the modeled component
      It is cubic mortar comprising of cement, graded sand and water
   2) Describe your selection for the material model
      Everything will be included because it is simple cubic
   3) Define the material model parameters and how their values were obtained
      It is possible to find individual properties about cement, graded sand, and water. However, I cannot find mixed material's properties in online. So, I will do experiment and get the material model parameters such as Young's modulus and Poisson ratio. On the same time, I will keep looking for material's parameters in online.
2. Boundary conditions
   1) Describe the loads
      Compression load will be applied on the top of the material such as $f_x = f_y = 0 \; and \; f_z = -p$, and the other side is fixed such as $u_x = u_y = u_z = 0$.
   2) Describe how these loads are being modeled as boundary conditions
      (1) Assign global node

```
def nodeVal(self):
    self.generate_mesh()
    self.coor = defaultdict(int)
    glb = 0
    for x in self.xl:
        for y in self.yl:
            self.nodeLocation[glb] = [x,y]
            self.coor[(x,y)] = glb
            glb+=1
```

Above figure is 2D example of the function for assigning global node number for each node. For example, global node number of [0, 0] will be 0, [0, 1] will be 1 and [0, 2] will be 2. Its relation will be saved in hash map named coor and nodeLocation.

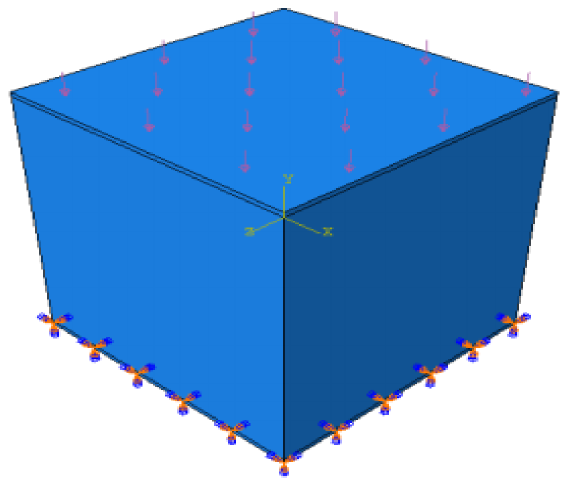
(2) Apply compression load

```
def defineBoundaryConditions(self):
    self.nodeVal()
    totalNodes = self.xNode*self.yNode*self.zNode
    for globalnode in range(totalNodes):
        #Dirichlet BC
        if self.nodeLocation[globalnode][2]== 0:
            self.boundary_values[3*globalnode+0] = 0
            self.boundary_values[3*globalnode+1] = 0
            self.boundary_values[3*globalnode+2] = 0
        #Neumann BC
        if self.nodeLocation[globalnode][2]== 1.5:
            self.NN_BC[3*globalnode+0] = 0
            self.NN_BC[3*globalnode+1] = 0
            self.NN_BC[3*globalnode+2] = -1e9
```

Above figure is the function of defining boundary conditions. In hash map named nodeLocation has an information of global nodes' location. So, I can know the locations of the global nodes. The hash map named boundary_values defines known deformation information. Since bottom surface is fixed, I define all deformation values are zero, where z-axis value is zero. The hash map named

NN_BC defines known force values. Since compression load is applied at top surface, I defined z-axis force as -1e9, and x-axis and y-axis forces as 0, where z-axis value is 1.5.

(3) Figure



Above figure is an example how to apply load on the material. Bottom surface is fixed and top surface is compressed by load.

*reference of the figure:
http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.736.5988&rep=rep1&type=pdf