**DEM model for NPH simulation**

**Geometry**

I have started using a simplified geometry for the brain.

I have generated a sphere (10 cm radius) and an .stl file using Gmsh.

I converted the .stl file into a .gts file and then imported into my code as a predicate.

I believe it is possible to create a more realistic model (for example 2 concentric sphere, where the inner one represents the ventricles) or even import a patient specific mesh.

In a previous email, you mentioned the fact that:

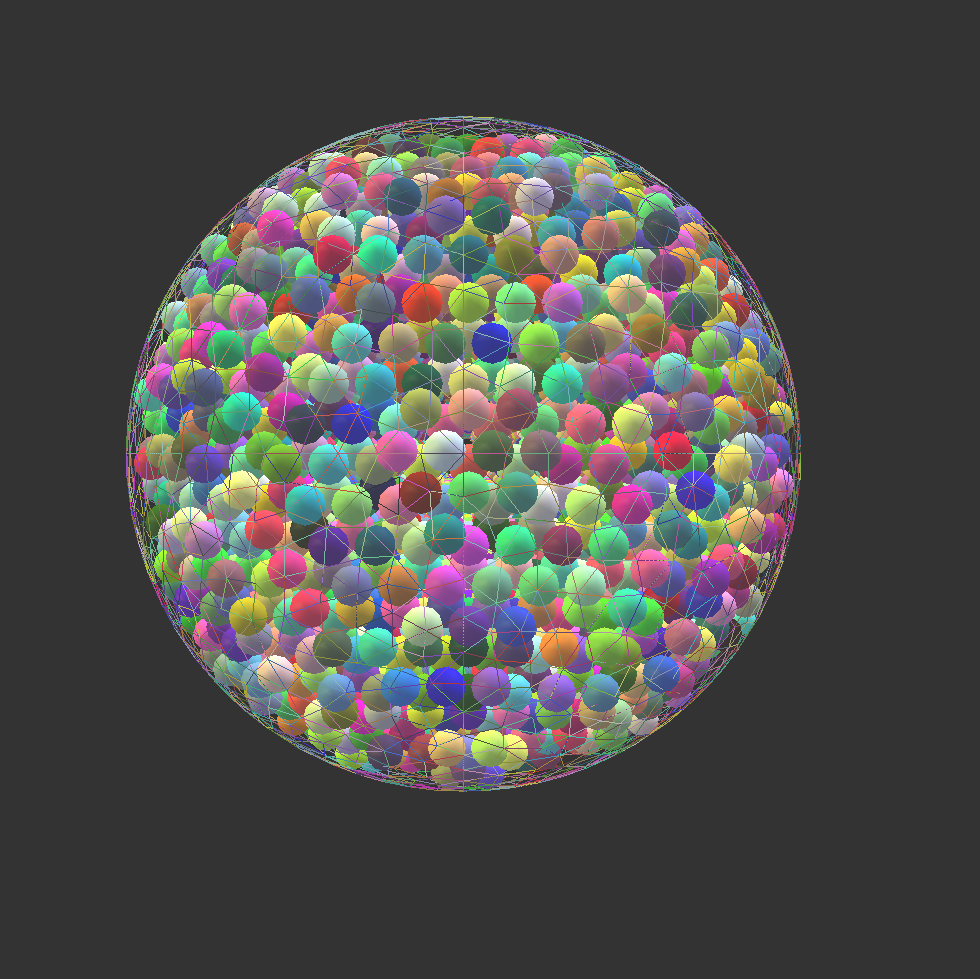
“ we need to define a row or two of spheres which align with the outside boundary circular wall in a very regular way like a chain, and then have separate "fluid channel" between two rows of particles”. I still did not figure it out how to do it. I can make more research and talk to Robert about it.

**Packing**

The predicate has been filled with spheres using the *regularHexa* function that return set of spheres in regular hexagonal grid,.

The spheres have a radius of 0.50 cm and the gap between the spheres is 0.0 cm.

These values can be easily adjusted.

  
Illustration 1: Snapshot of the model

**Material properties**

It is possible to

**Engines**

As concerns the engine and collider model I have used the one that were in the demo I have started from. I probably need more reading and studying about it.

O.engines=[

ForceResetter(),

InsertionSortCollider([Bo1\_Sphere\_Aabb(),Bo1\_Facet\_Aabb()],label='collider'),

InteractionLoop(

[Ig2\_Sphere\_Sphere\_ScGeom(),Ig2\_Facet\_Sphere\_ScGeom()],

[Ip2\_FrictMat\_FrictMat\_FrictPhys()],

[Law2\_ScGeom\_FrictPhys\_CundallStrack()],

),

NewtonIntegrator(damping=.1),

PyRunner(iterPeriod=1000,command='timing.stats(); O.pause();'),

PyRunner(iterPeriod=10,command='addPlotData()')

]

**Boundary conditions**

For what I have read from Yade manual it is possible to cycle over the spheres and impose displacement or velocity.

For example:

for b in O.bodies:

if b.shape.name!=Sphere: continue # skip non-spheres

# ask the predicate if we are inside

else:b.dynamic=False

In this way all the spheres are fixed.