**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:

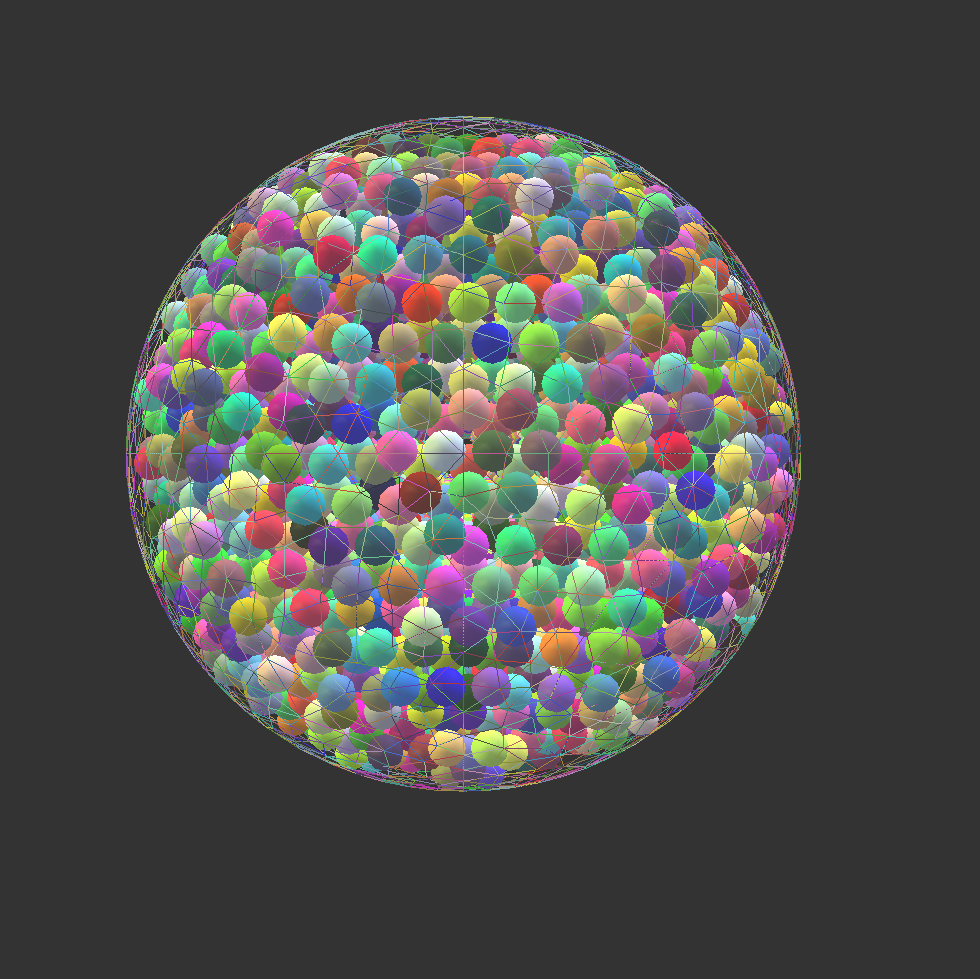
“ we need to define a row or two of spheres which align with the outside boundary circular wall in a very regular way like a chain, and then have separate "fluid channel" between two rows of particles”. I still did not figure it out how to do it. I can make more research and talk to Robert about it.

**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 there is no gap between the spheres.

These values can be easily adjusted.

  
Illustration 1: Snapshot of the model

**Material properties**

Unfortunately, there is not much agreement on the material properties of brain tissue. According to Li Et Al Influences of brain tissue poroelastic constants on intracranial pressure (ICP) during constant-rate infusion the Young modulus is around 10000 Pa. According to Tully and Ventikos Cerebral water transport using multiple-networkporoelastic theory: application to normalpressure hydrocephalus instead the Young modulus is closer to 500Pa.

As concerns the Poisson ratio this varies between 0.35 and 0.48.

I have defined the material properties as:

idTissue=O.materials.append(FrictMat(young=500.0,poisson=.35,frictionAngle=.6,label="concrete"))

I have to understand which is the best value for the friction angle.

**Simulation loop**

From Yade’s manual:

*In a typical DEM simulation, the following sequence is run repeatedly:*

*• reset forces on bodies from previous step*

*• approximate collision detection (pass 1)*

*• detect exact collisions of bodies, update interactions as necessary*

*• solve interactions, applying forces on bodies*

*• apply other external conditions (gravity, for instance).*

*• change position of bodies based on forces, by integrating motion equations.*

*Each of these actions is represented by an Engine, functional element of simulation. The sequence of engines is called simulation loop.*

We can use this setup. Eventually also a FlowEngine can be incorporated.

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=0.1,gravity=[0,0,0]),

# FlowEngine(label="flow"), commented at the moment

]

**ForceResetter():** itresets forces at each timestep

**InsertionSortCollider():** it manages collision between particles and facets. The functors Bo1\_Sphere\_Aabb() and Bo1\_Facet\_Aabb() are bound functors and define the Axis-aligned bounding boxes (Aabb) for spheres and for facets. At this stage we only have spheres and facets so it should be enough.

**InteractionLoop():** in this loop the interaction between the spheres.

The functor prefixed with Ig2

**Fluid Engine**

It is possible to incorporate the fluid dynamics with the Flow engines.

It seems possible to impose the pressure in one point like this:

flow.imposePressure(Vector3((0.0,0.0,0.0)), 10.0)

Unfortunately, it seems difficult to impose boundary conditions. I have already asked a question to Yade forum and waiting for an answer

Bibliography

"LiEtAl2013": , Influences of brain tissue poroelastic constants on intracranial pressure (ICP) during constant-rate infusion,

"TullyVentikos2010": , Cerebral water transport using multiple-networkporoelastic theory: application to normalpressure hydrocephalus,