Comets, exercise II

We will try to build ourselves a model for cometary dust particles. Some background and boundaries for particles from Evgenij's slides:

Modeling cometary dust particles

(a) dust particles have to be essentially irregular (non-spherical, non-cubical, etc.)
(b) dust particles have agglomerate structure. Packing density have to be chosen in order provide range of the measured density (from 0.3 to 3 g/cm3).

Refractive indices for cometary spices in visible:
Mg-rich silicate: Re(m) = 1.5 – 1.6 Im(m)=0.00001 – 0.01
Water ice: Re(m) = 1.31 Im(m)=0
Organic material Re(m) = 1.4 – 1.6 Im(m)=0.002 – 0.5

Creating the model shape

We need to create a shape for our particle that is (a) irregular and (b) have aggregated structure. Although the shape of the aggregate should be irregular, the constituents (basic shapes that make up the aggregate) can be simple and regular if they are small compared to the wavelength. Spheres or cubes are fine. Some algorithms for aggregate creation:

Ballistic particle-cluster aggregation:

- Add one sphere on the origin
- Take random direction and "shoot" another particle towards origin
- Check where it touches the first one and put it there
- …continue

Ballistic cluster-cluster aggregation:

- Create small clusters, with e.g. ballistic particle-cluster
- Add clusters together as in particle-cluster aggregation

Cloud of particles: (not exactly an aggregate, but…)

- Choose surrounding volume, e.g. large sphere, cube, …
- Put constituents (spheres) in random locations inside surrounding volume
- You can choose to let/not let the constituents to overlap, because overlapping can be neglected when converting the geometry to voxels for DDA

Random walk in 3D:

- Choose surrounding volume, e.g. cube or sphere
- Put one cube on the origin
- Choose randomly the next cube adjacent to previous
- Do not let the random walk escape the surrounding volume
- If you let the process to cross itself, remove duplicate cubes in the end
- Pros of random walk – automatically creates voxel geometry suitable for DDA.

Converting geometry to voxels

We need to convert the geometry consisting of larger geometrical shapes (e.g. spheres) to small voxels (dipoles) for DDA. This can be done e.g. with simple volume scanning algorithm:

- Define voxel (a.k.a cell, dipole) size and rectangular volume that envelopes your aggregate.
- Go through the voxel grid from (min(x),min(y),min(z)) to (max(x),max(y),max(z)) one by one
- For each (x,y,z) check if that voxel is inside any of the shapes in your aggregate. If inside, save the coordinates (x,y,z)

Once all the voxels containing material have been found, write them in ADDA format to a file. The file should have this format:

```
# comment rows with'#'
x1 y1 z1
x1 y1 z2
…
xn yn zn
```

The voxel coordinates (xi,yi,zi) need to be integers. The actual size is given with ADDA command line arguments (e.g. with *–dpl* and *–lambda*).

After finished, try to compute scattering using your shape (command line option *–shape read filename*).

Hints:

Random direction on the sphere

Spherical coordinates (θ,ϕ) can be created so that ϕ = rand(0,2π) and θ=arccos(rand(-1,1))

Point-line distance

Say you have a line going through origin and $\boldsymbol{x}$ and you need to compute the shortest distance from line to sphere center $\boldsymbol{c}$. The distance d is given by

$$d = \frac{|\mathbf{c} \times (\mathbf{c} - \boldsymbol{x})|}{|\boldsymbol{x}|}$$

where × is the cross product (see next figure). If $d <= 2r$, the new sphere approaching to origin from $\boldsymbol{x}$ will collide with sphere at $\boldsymbol{c}$. ps. Make sure that $|\boldsymbol{c} \cdot \boldsymbol{x}|$ is positive, so that $\boldsymbol{c}$ is on the direction of $\boldsymbol{x}$ and not in the opposite direction from origin.

Place for new sphere that collides with sphere at *c*

(See figure) You can get d from above, y_1 and y_2 from Pythagora's theorem. So, $\boldsymbol{x}$ should be replaced to $(y_1+y_2)\,\boldsymbol{x} / |\boldsymbol{x}|$, whatever the initial distance $|\boldsymbol{x}|$ was.