fastAerosol for GEANT4

N. MacFadden, A. Knaian

NK Labs, LLC

September 17, 2020

High-level introduction slides: A. Knaian

https://www.nklabs.com

Aerosols

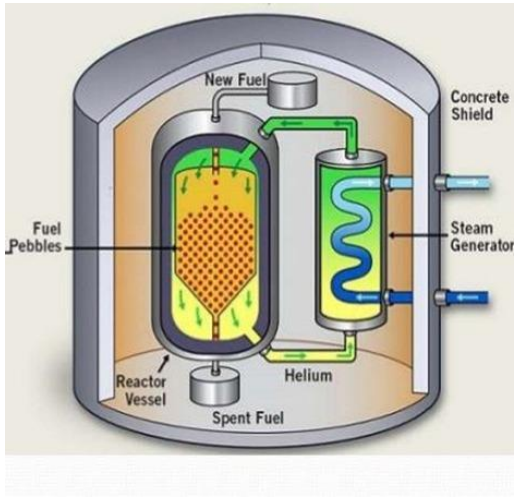
Atmospheric Cloud

Smoke Stack

Dust storm, Texas, 1935

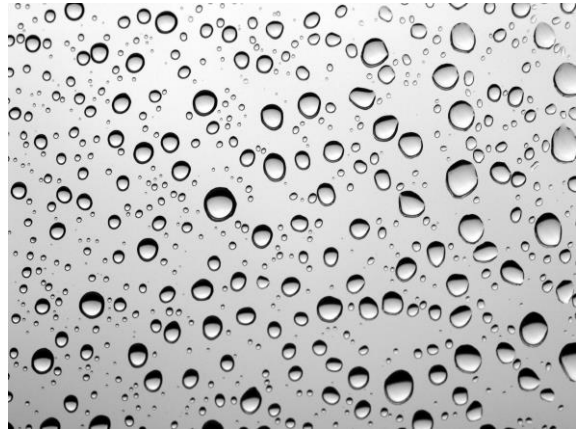
Carina Nebula

Droplet spray



Pebble Bed Nuclear Reactor

Bag of wooden pellets



Raindrops in the Atmosphere

Aerosols

- Overall aerosol has a size/shape
- Droplets have a size/shape distribution
- Defined number density of droplets per volume
- But --- exact positioning of each droplet -- the microstate – is unknown and for practical purposes not an experimental observable

Why did we write fastAerosol?

- Nuclear energy application
- Aerosol with billions of particles
- Using a parameterized volumes, the machine would run out of memory, or “take forever” to run.
- fastAerosol models the whole aerosol as a single GEANT4 geometry object. It is very fast, especially for small numbers of primaries relative to the number of droplets.
- Enables simulation of detectors that would not be possible to simulate otherwise, because the geometry would exceed the memory of the computer.

fastAerosol is Easy to Use!

```
cloud = new fastAerosol("cloud",
                        cloudShape,                 //cloud shape
                        dropletR,                   //bounding radius of droplets
                        minSpacing,                 //minimum spacing between droplets
                        dropletNumDens,             //approximate number of droplets in cloud
                        sphericalUncertainty);      //uncertainty in distance to droplet surface from outside using just droplet's origin as info
cloud->SetDropletsPerVoxel(4);


fastAerosolSolid* solidCloud =
    new fastAerosolSolid("cloudSV",                 //its name
                         cloud,                     //its shape
                         dropletShape);             //its droplets
```

2-3 lines in your DetectorConstruction::Construct() method creates your aerosol from its macro properties, and then you can start shooting primaries.

fastAerosol Methods and Results

N. J. L. MacFadden[1,2] A. N. Knaian[1]
Slides created and presented by N. J. L. MacFadden

[1]NK Labs, LLC

[2]Department of Physics
University of Waterloo

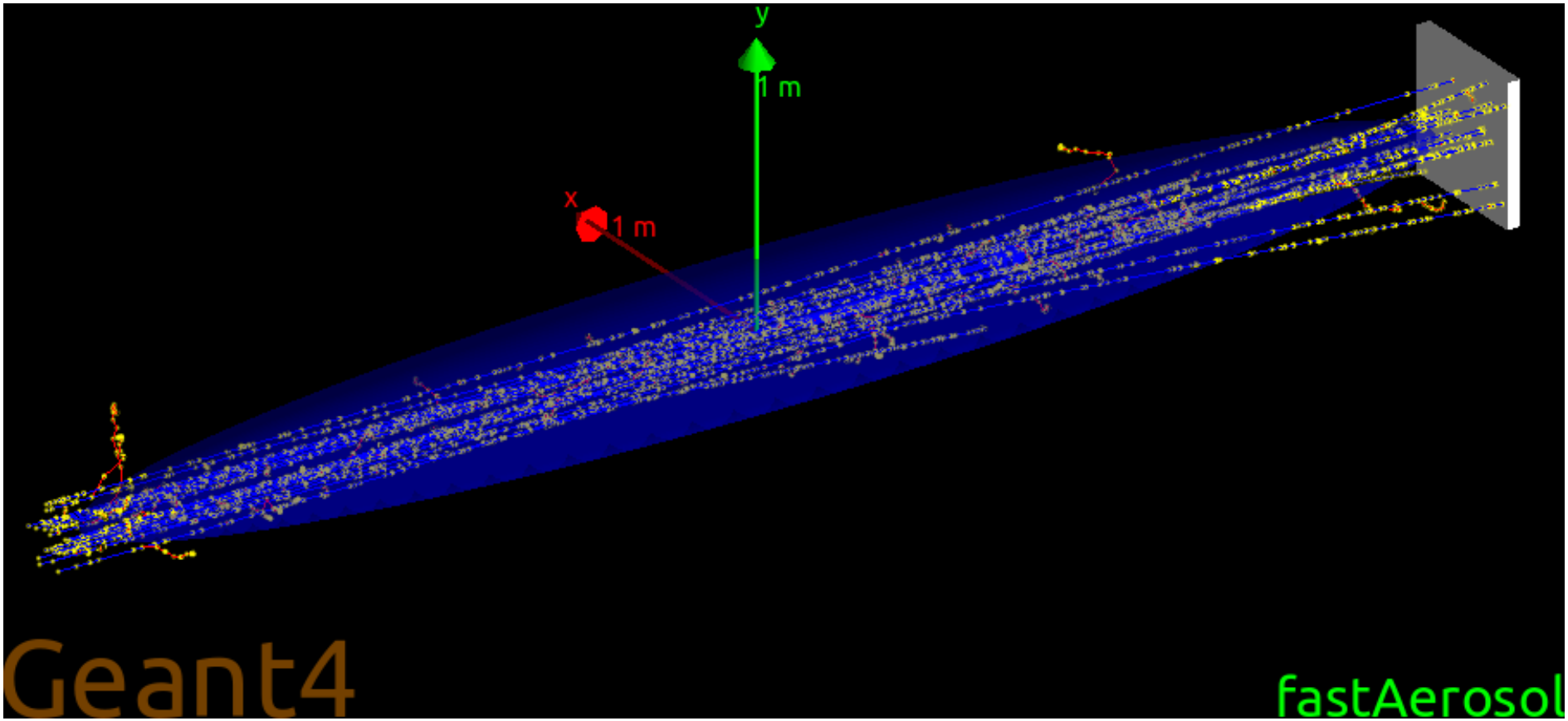
25th Geant4 Collaboration Meeting, September 2020

Outline

Need for Granularity

fastAerosol is modelled droplet-by-droplet ('granular') - **why**?

Consider shooting protons through an aerosol:

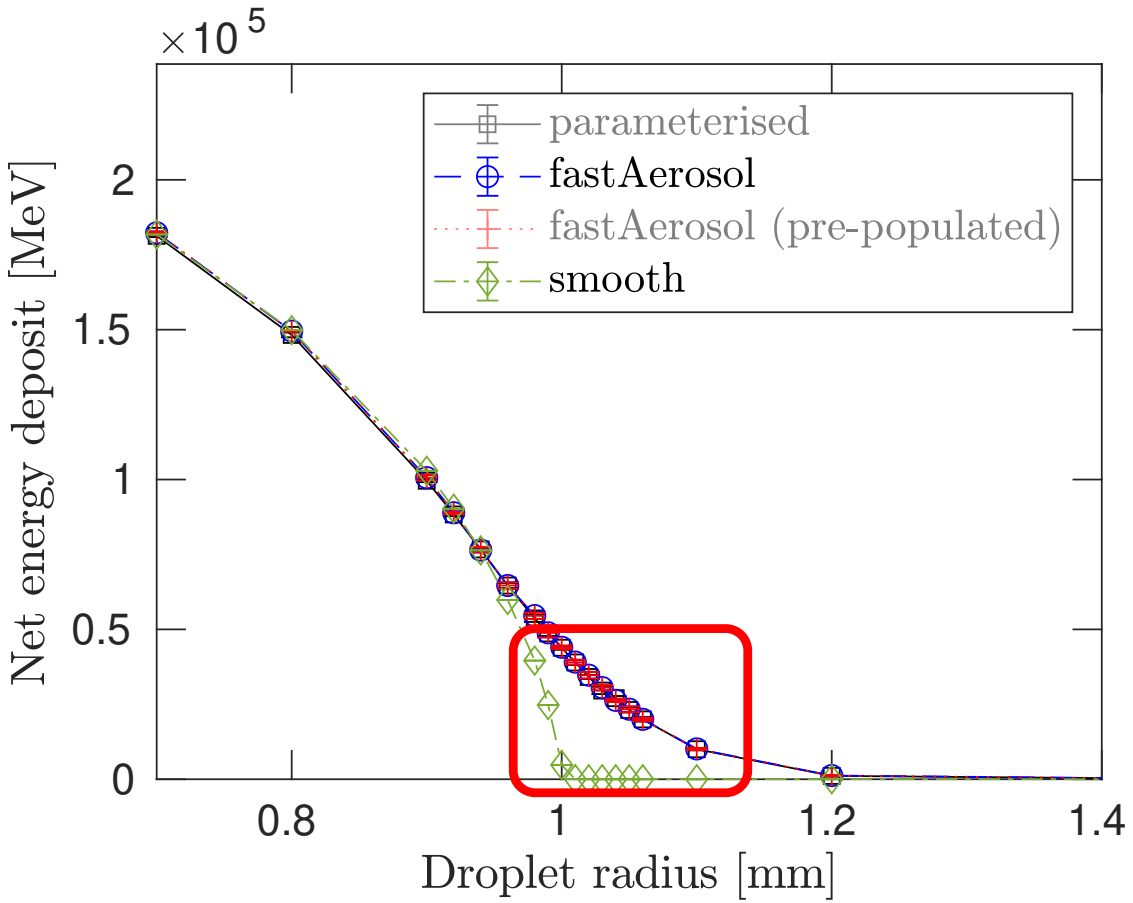


(figure from our associated preprint: arxiv 2008.01236)

Need for Granularity (pt. 2)

fastAerosol is modelled droplet-by-droplet ('granular') - **why**?

Ignoring granularity leads to different transport for some droplet radii:



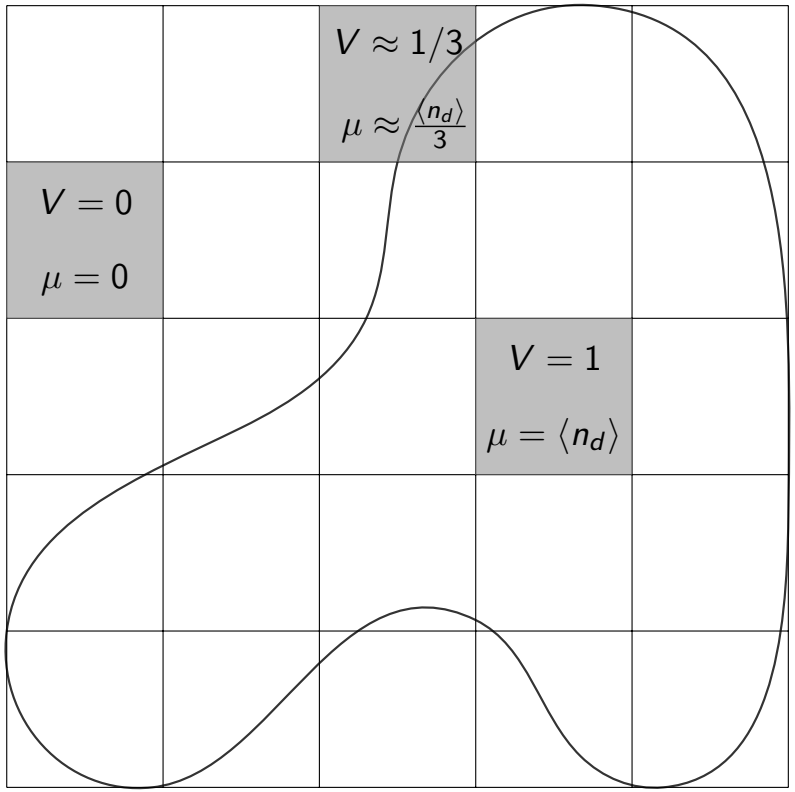
(figure from our associated preprint: arxiv 2008.01236)

Aerosol Generation

"Aerosol" = 3D grid of voxels containing droplets in some bulk shape

Preparation for aerosol with average droplet number density $\langle n_d \rangle$:

- voxelize bounding box with pitch $p_{\text{grid}} = \left(\frac{N_{\text{droplets/voxel}}}{\langle n_d \rangle} \right)^{1/3}$,
- assign expected droplet count per volume as $\mu = \langle n_d \rangle V$,[1]
- assign each voxel a unique seed $s_{\text{vox}} = \text{index}_{\text{vox}} + s_{\text{global}} N_{\text{voxels}}$.



Voxels are then populated **on-demand** by:

- selecting N_{vox} from Poisson dist. with mean μ and seed s_{vox} and then
- placing N_{vox} uniformly in voxel (retrying placement if droplet overlap)

[1] for non-uniform droplet distributions, multiply μ by distribution function $f : \mathbb{R}^3 \to \mathbb{R}$

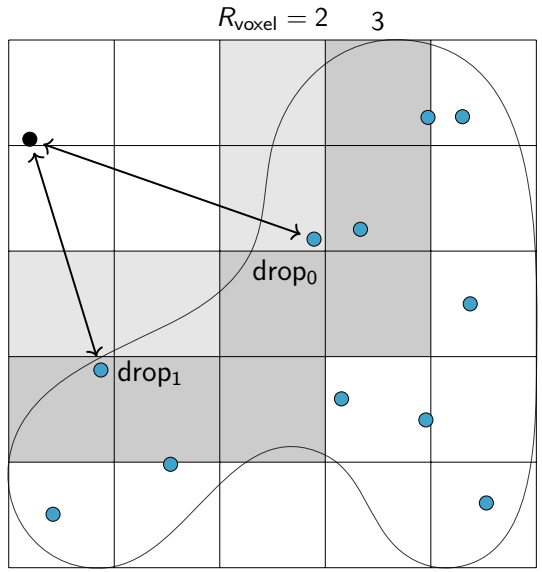
Distance Calculation

Distance functions in fastAerosol operate by

- finding the relevant droplet for the query and then
- delegating the distance calculation to this droplet.

To find the relevant droplet for distance to the inside of the aerosol:

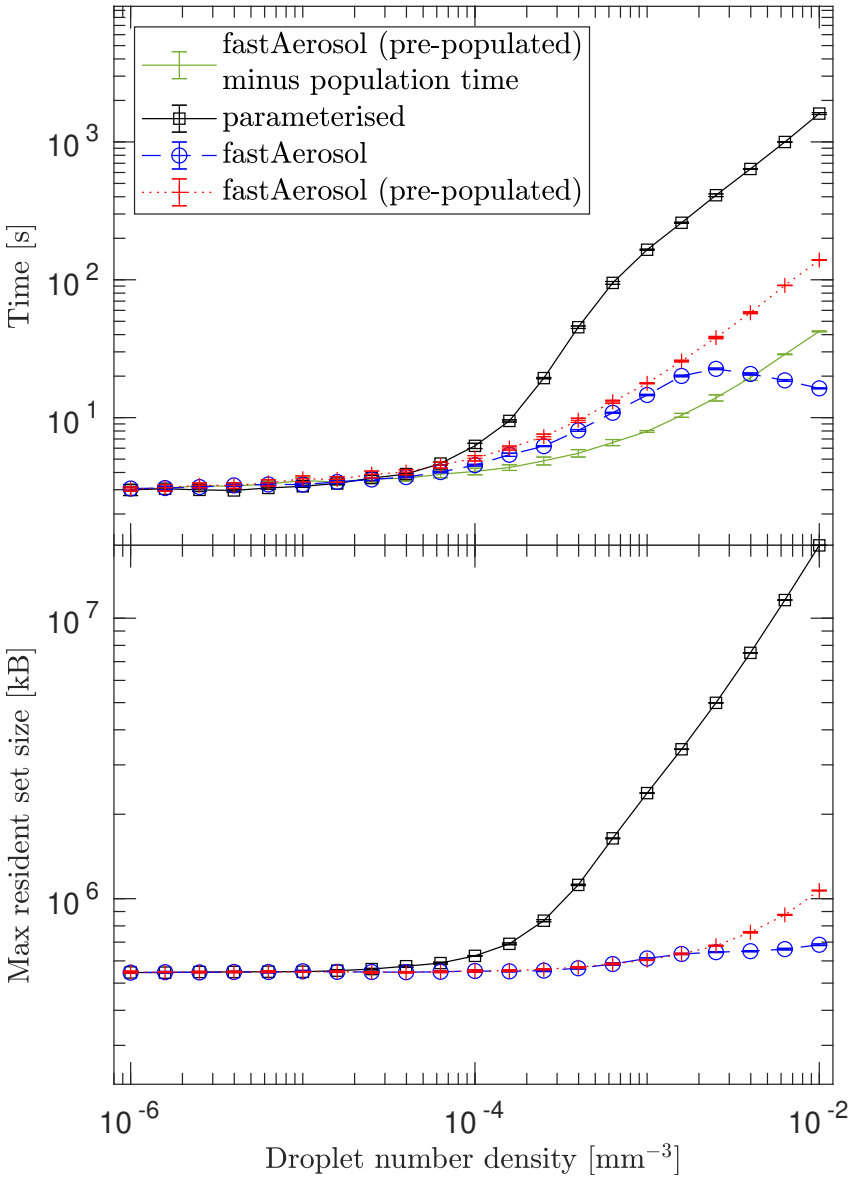
- find distance to bulk in units of grid pitch $R_{\text{voxel}} = \lfloor R_{\text{bulk}} / p_{\text{grid}} \rfloor$,
- collect (in $\mathscr{C}$) all droplets found in voxelized spheres[1] with radius $R_{\text{voxel}} \leq R \leq \lceil \frac{1}{4} + \frac{R_0}{p_{\text{grid}}} \rceil$ for R_0 the closest found droplet center,[2] and then
- return min distance among $\mathscr{C}$.



[1] generated by modified mid-point algorithm defined by (Roget, Sitaraman '13)

[2] non-spherical droplets introduce factor and $\sigma = r_{\text{out}} - r_{\text{in}}$ to upper R limit

fastAerosol is Efficient for Dense Aerosols



(figure from our associated preprint: arxiv 2008.01236)

Conclusion

fastAerosol

- allows speedups in dense granular aerosols
- only depends on macroscopic properties, simplifying implementation:

```
fastAerosol* aerosol =
    new fastAerosol("aerosol", bulkShape, boundingR,
                    minSpacing, avgDropNumDens, sigma,
                    positionDistribution);

fastAerosolSolid* solidAerosol =
    new fastAerosolSolid("aerosolSV", aerosol,
                         dropletShape,
                         rotationDistribution);
```

- allows complicated aerosols with
  - any shape bulk/droplet,
  - any droplet number density distributions, and
  - any droplet rotation distributions.