Computational light scattering (PAP315)

Lecture 12b

Antti Penttilä

Department of Physics, University of Helsinki, Finland

What you need for installing and running SIRIS (multi-particle)

- Linux/Mac/Windows+MSYS2 with GCC gfortran and C++ compilers
  - The GCC version needs to be high enough (v 4 is not enough) so that it can compile with –std=c++14 -option
- GNU make tool
- Boost and CGAL libraries
  - sudo yum install boost boost-devel
  - sudo yum install CGAL CGAL-devel

- …but, if your GCC is too old, see https://forums.centos.org/viewtopic.php?t=71219
- …if CGAL is not found, see https://stackoverflow.com/questions/44037925/how-to-install-cgal-on-centos-7or-centos-6
- …but you would need sudo-rights to your computer, so I hope you have them…

SIRIS, installation

- Go to https://bitbucket.org/planetarysystemresearch/siris4-framework
- Get package by downloading the zip from *Downloads*
  - wget https://bitbucket.org/planetarysystemresearch/siris4-framework/get/94b3eb39c45d.zip
  - unzip 94b3eb39c45d.zip
  - mv planetarysystemresearch-siris4-framework-94b3eb39c45d siris4-framework
- …or by git:
  - git clone https://bitbucket.org/planetarysystemresearch/siris4-framework.git
- cd siris4-framework
- make multiparticle

SIRIS (multi-particle), running

- All parameters are given in input file, and the input file name will be given as option in the command line.

```
nrays 1000000    # Number of rays
max_scattering 200    # Maximum number of scattering events
killswitch_start 70    # Prevent rays splitting to refracted and reflected (only one of these happens) rays after N scattering events
nbins 80    # Number of theta angle bings
nbins_fine_details_start 180    # This is used to print finer details at the backscattering direction
prevent_TR 0    # Prevent total reflection creating refracted rays
check_time_after_nrays 1000    # Check time after N rays
allocated_time_in_hours 9999    # Kill execution after N hours
output_file outputS.out    # Print scattering matrix elements per phase angle
pmatrix_out pmatrix.out    # Print scattering matrix that is readable by the SIRIS (so you can use the output as an input for next round...)
details_out details.out    # Print other details about the finished work
I_cutoff_limit 0.0000001    # Cut off limit. When intensity of the ray goes below this limit, kill it
seed 0    # Generate random seed for the PRNG (0), If nonzero, the given number will be used as a seed.
wavelen 6.283185307179586    # Wavelength. Unit does not matter as long as you keep it consistent with other length parameters
mesh_scale 2000    # Mesh file is scaled with this value. So, if the wavelenght is 6 nm, and if the mesh file has a sphere with radius 1 (dimensionless), SIRIS will compute 2000-nm-sized spehre with mesh_scale 2000.
mesh sphere.off    # Relative path to the shape model (see below)
force_interaction 1    # This is related to the diffuse scattering. Do we force every ray to interact with the diffusely scattering media, or do we let them go through. Without this the observer can see huge spike at the forward scattering direction.
beam_radius 250    # The radius of the incident beam (same units as above). Negative beam size means that the entire medium is covered by the beam.
material1 1.0 0.0 1 1.0 0.9460290562711914 outputS_ave.out 1 ~/dists/cdfconstant04dist.txt    # Define materialX, where the X is the number of the material. Supports 255 materials. It is important to note that material 0 is reserved for the surrounding media. The format is
materialX REF_REAL REF_IMAG DIFFUSE_ON ALBEDO MEAN_FREE_PATH PATH_TO_PHASE_MATRIX EXPERIMENTAL_MFP_ON file_path    # where REF_REAL is the refractive real part and REF_IMAG is the imaginary part. Material can have diffuse inclusions that are enabled by using 1 for DIFFUSE_ON. ALBEDO, MEAN_FREE_PATH and PATH_TO_PHASE_MATRIX are for the diffuse scatterers, whereas EXPERIMENTAL_MFP_ON is about the experimental mean free paths (see SIRIS2019 or SIRIS2020 paper).
media REF_REAL REF_IMAG    # For the surrounding medium, REF_REAL is the refractive real part and REF_IMAG is the imaginary part.
```