Radiative transfer simulations in astrophysics

Master Physics and Astronomy 21 June 2022

Oral exam

Disclaimer: This was not the exact wording.
10 minutes preparation, 50-60 minutes exam (half about theory below, half about project)

1. You are given a snapshot of a hydrodynamical simulation. How do you extract the stellar emission spectrum from this snapshot? *Bonus (unprepared): What determines the SED of a certain particle?*

2. Now that you have the spatial distribution of the snapshot, how do you sample a location from it?

3. If you were to simulate the radiation of a galaxy from UV to sub-mm wavelengths, would you include non-equilibrium (NLTE) dust grain heating or stay confined to LTE conditions?

4. Describe 'continuous absorption'. What is the main benefit of this technique?

5. Why do most modern simulations use adaptive grids? Give an example of a hierarchical grid and an unstructured grid. *Bonus (unprepared): Adaptive grids try to adjust for density differences, but what is it we would really like to track?*

6. What extra properties do you need in a simulation to track the polarization, both for the photon packets and the dust, and why?

7. You are given an observation of a galaxy. How do you construct a model based on these observations?

8. How are dust grains formed and what is their physical function in a galaxy? *Bonus (unprepared, maybe to help me on my way): What are dust grains composed of and where in the evolution of a galaxy are those constituents formed.*

Answers

Disclaimer: No guarantees of being error proof. Syllabus (sub)chapter in bold.

1. **Chapter 6: The SKIRT code; 6.2: User interface and configuration**
   One emission spectrum per particle in the snapshot, smoothed by a smoothing kernel. All particles after smoothing added to get the complete emission spectrum. *Bonus: metallicity and age of the particle; particles in a hydrodynamical simulation usually represent stellar populations (and not just one star)*

2. **Chapter 8: Sampling spatial distributions; 8.1: Geometric building blocks in radiative transfer codes; 8.1.2: Components based on smoothed particles**
   With the composition method: sample a random particle, with particles weighted by luminosity, then sample a position from the distribution of that specific particle/smoothing kernel.

3. **Not sure**
   Include non-equilibrium, LTE grains re-emit in sub-mm range already and NLTE dust peaks show up at lower wavelengths (e.g. PAHs at ≈ 10 μm)

4. **Chapter 5: Weighted Monte Carlo radiative transfer; 5.2: Weighted photon life cycle; 5.2.5: Continuous absorption**
   Absorb energy from photon packets not just in interaction site, but in all cells along its path. Benefit = tracking radiation field with fewer photon packets than without continuous absorption.

5. **Chapter 7: Spatial grids; 7.3-7.5: Uniform grids, Hierarchical grids and Voronoi grids**
   Why: uniform grids use too much cells, some properties need to be calculated for each traversed cell, so less cells is a good thing, which can be reached with hierarchical/unstructured grids. Hierarchical: e.g. octree, k-d tree, explain how it is built. Unstructured: Voronoi (only one seen), explain how it is built. *Bonus: the radiation field, since that will influence dust emission the most, hopefully, density is a good tracker for it*

6. **Chapter 10: Extensions to the radiative transfer equation; 10.3: Polarization due to scattering by dust grains**
   For photon packets: Stokes vector. For dust particles: Müller matrix from which the phase function can be calculated.

7. **Chapter 11: Inverse radiative transfer; 11.2: Applications** Not sure if it was mentioned in the question, but you need to cover both the face-on and the edge-on case.

8. **Chapter 1: Cosmic dust; 1.2: Formation and destruction of dust grains; 1.2.1: Grain formation**
   Dust is condensed out of material. It is built out of carbons/silicons/... that are produced in stars. Sources for the interstellar dust formation are stellar outflows or supernovae or just atoms/ions sticking together.

Project

Below is the description of the project, not sure how useful it will be but whatever.

Radiative Transfer Simulations in Astrophysics
2021-2022 – Project Assignment

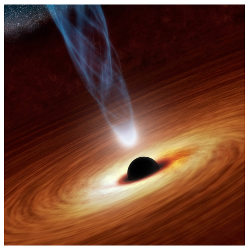
Reverberation mapping with SKIRT

Peter Camps
peter.camps@ugent.be

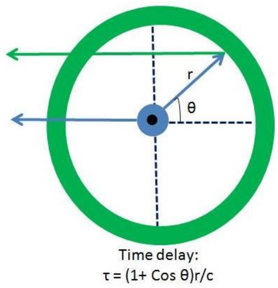
Bert Vander Meulen
bert.vandermeulen@ugent.be

March 2022

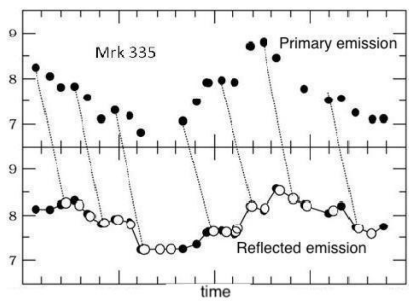
1 Introduction



Active galactic nuclei (AGN) are compact, luminous regions that are observed in the centre of massive galaxies. Their multi-wavelength excessive brightness is understood to be powered by the accretion of gas and dust onto a supermassive black hole, situated in the very centre of the galaxy. AGN are the most luminous persistent sources in the Universe, with a luminosity that can be as high as all stars in the host galaxy combined.



Currently, it is not possible to spatially resolve AGN in observations, due to their compactness and large distances. The most effective way to map out the AGN environment, is to use the echos (or *reverberations*) from light produced by accretion-disk emission (blue) that is reflected on the surrounding material (green), as illustrated in the figure on the left. When the brightness of the central source varies, the reflected emission component will vary in response. However, this response will be delayed with respect to the central source emission, corresponding to the light travel time towards the surrounding media. The *reverberation mapping* technique uses these delays to map out the spatial structure of the spatially-unresolved AGN environment.



Radiative transfer simulations form an important tool to disentangle and interpret the various AGN reverberations that are blended into a single observed time series. As modern simulations are equipped with smart detectors that can access in–simulation data, it is possible to record the photon travel time (or equivalently its travel distance) for each individual photon. This information can then be used to decompose simulated spectra in components of different delay times, and so one can reproduce and interpret observed AGN time series.

2 Overall objectives

For this project, you will work with and adjust the state-of-the-art radiative transfer code SKIRT, version 9. The main objective of this project is to study the geometry and optical properties of a transfer medium in the vicinity of a bright point source, by using the timing information of the observed photons. To this end you will:

1. Install SKIRT 9 and an appropriate C++ software development environment on your computer and perform the SKIRT 9 tutorials as described on `www.skirt.ugent.be`.
2. Create some SKIRT models (.ski files) that can serve as test-beds for your investigations.
3. Extend the SKIRT code to track photon travel times (or distances, which is equivalent) during the simulation.
4. Augment the SKIRT instruments to output photon timing information, and use these data to calculate light curves for your test-bed models.
5. Interpret the light curves for various test-beds and investigate what optical and geometrical medium properties can be derived from these light curves.
6. Summarise your actions and describe your findings and insights in a written project report, to be handed in a few days before the oral exam.
7. Be prepared to respond to questions about the project, your findings and their meaning, during the oral exam.

Take things one step at a time. Ask for help or advice, especially when stuck. Show intermediate results and seek feedback, also when not stuck.

3 SKIRT test-bed models

You will need some simple test-bed models of transfer media surrounding a central point source and/or interposed between the source and the observer. In the context of this project, you can concentrate on extinction effects (absorption and scattering) only, without worrying about secondary emission. Thus, you can set up simulations with a single representative wavelength in the optical range (the oligochromatic regime in SKIRT-speak). Such simulations only have a single stage, i.e. launching photon packages from the primary point source.

Imagine which geometrical setups could produce interesting (interpretable) results for your project. You could consider a 3D version of the slab exercise presented in the syllabus. For example, you can put a single point source on one side of a cuboidal, uniform slab and observe the radiation that penetrates or scatters back from the slab. Also, you could consider a spherical shell surrounding the point source, observed from the outside.

Some more recommendations:

1. Work in SI units, and choose the wavelength output styles (for wavelengths and fluxes) to avoid unnecessary unit conversions.

2. Only include one single point source, in combination with one or more transfer media (SKIRT offers a rich suite of built-in transfer media).

3. You will need some SED for your source, even though this has little meaning for one-wavelength simulations. Select isotropic, unpolarised emission.

4. Use the `MeanListDustMix` composition for your transfer media, which gives you full control over the optical properties inside the .ski configuration file (i.e. the optical depth, the scattering albedo and the Henyey-Greenstein scattering asymmetry parameter). Note that this composition is not available in the SKIRT user experience level 1 "basic".

5. Focus on a distant instrument that outputs the spatially integrated flux density as an SED (but you can check the surface brightness for fun).

6. Include probes such as `SpatialGridConvergenceProbe` to verify the convergence of the spatial grid, and `OpticalMaterialPropertiesProbe` to see the optical properties of the configured material mix.

7. It may be interesting to vary the optical depth through your transfer media. A good starting point is a value of one. There is no need to use optical depths above 5 for this project.

4 SKIRT adjustments

4.1 Tracking photon travel distances

Your next step should be the implementation of some way to keep track of the photon travel times. For photons, this is equivalent to tracking their traversed distance ($D = c \cdot t$), which is easier to implement. Later, these distances can easily be converted to delay times. All of your SKIRT modifications should happen in the `PhotonPacket` class:

1. Add a *distance* data member `_D` to the `PhotonPacket` class, similar to the *wavelength* and *weight* attributes `_lambda` and `_W`.

2. Construct a getter-function for `_D`, similar to `wavelength()` for `_lambda`.

3. Examine the five important functions `launch`, `launchEmissionPeelOff`, `launchScatteringPeelOff`, `propagate` and `scatter`, and properly update the travelled photon-distance `_D` inside these functions (if needed). Never update `_D` outside of these functions.

4. After invoking `launchEmissionPeelOff` or `launchScatteringPeelOff`, the photon packet will be detected. At the end of these two functions, you can correct for the travel distance to the observer. Focusing on distant instruments, all photons will leave the system in the same direction `bfkobs` to the observer. Therefore, it is possible to account for the relative differences in travel distance as:

$$\Delta D = -\vec{\texttt{bfkobs}} \cdot \vec{\texttt{bfr}}, \tag{1}$$

   with `bfr` the photon peel-off position. You do not need to track the absolute travelled distance, as time lags are produced by distance differences.

5. You might want to have a look at the `SpatialGridPath` class, which contains all geometric details of the photon path through the spatial grid.

6. Don't be too concerned with the plethora of other functionality that is present in the `PhotonPacket` class. Especially, don't look into the implementation related to velocities, angular distributions and polarisation.

4.2 Recording timing information

As a next step, you will need to record the timing (here: distance) information when photons are detected, so you can output and use these data for your study. As your simulations consider one wavelength only, you can record the timing data in a 1D array (flux per distance bin), and then write this array to a text file. Your next SKIRT adjustments should happen in the `FluxRecorder` class. Closely follow the implementation for outputting the 1D `_sed` array.

1. Add a private `vector<Array> _lc`, similar to the `_sed` array, which will (eventually) represent the light curve.

2. In the `finalizeConfiguration` function, resize `_lc`. You can hard-code the number of bins and the distance range of `_lc` in the `FluxRecorder` namespace. Make sure these values suit your SKIRT test-bed models. Make an estimation of what should be a reasonable maximum photon travel distance before detection.

3. Don't be too overwhelmed by the vast amount of functionality that you can safely neglect for now. Focus on the `detect` and `calibrateAndWrite` functions, and record light curves in the same way as recording SEDs.

4. In the `detect` function, add the luminosity Lext of detected photons to the correct bin of `_lc`, by following the procedure for adding photons to the `_sed` array. Focus on the `_recordTotalOnly` loop without polarisation, medium emission or statistics. Store all Lext contributions outside the `_lc` range in the last bin of `_lc` to detect distance range problems.

5. In the `calibrateAndWrite` function, skip the calibration part. This is just a global scaling that is not relevant in the context of this project.

Write the `_lc` array to a text file, using the `TextOutFile` class. Once again, follow the procedure for writing out the `_sed` array. Add two text columns (distance and flux).

5 Interpretation

Now, you can run SKIRT simulations and produce timing output for each of your test-bed models. Open the timing data files in Python and plot light curves (i.e. flux as a function of time) by interpreting the travel distances as delay times. Confirm that the direct source emission is observed at $t = 0$. Continue your analysis in units of distance, which has more meaning. Make sure your distance resolution and range are sufficient for a proper evaluation of the results.

Interpret the light curves for some interesting test-bed models. What do you see and why is that expected (i.e. what does it mean)? Can you calculate the distance between source and medium from the light curve? Can you derive information on the optical properties (optical depth, scattering albedo) of the reflecting or intervening media from the light curve?

Compare the light curves of one single model for varying optical depth and scattering albedo. There is no need to make a complete "grid" of all possible parameter combinations; look for values that produce meaningful differences (or remarkable similarities). Interpret the "observational" results in view of the optical properties. Why does a particular change in optical depth or albedo generate a particular effect in the result? In what parameter range can you make valid predictions on the model distances?

Your written report should summarise your actions and describe your findings and insights. Describe the SKIRT models used and your changes to the SKIRT code in addition to your results. You can show a few sections of relevant code changes, but there is no need to include the full C++ listing. Include the relevant plots that justify your interpretations and conclusions. Be selective; avoid including plots that don't show anything new. Every included plot should be discussed in the text.

6 Evaluation

Each student submits a written project report in .pdf format, a few days before the oral exam during the first examination period. The precise deadline will be announced via UFORA. The oral exam includes a discussion (question and answer session) on the project report.

The project report can only be submitted during the first examination period; for students who fail for the course in the first examination period and participate in the second examination period, the project report grade from the first period is automatically transferred.