Hands-on session I

Title: Optical properties and radiative transfer
Lecturers: Ciska Kemper & Alvaro Sanchez-Monge
Context: ICE summer school 2023 – Life Cycle of Dust
Date: 4 July 2023 – 14:00-18:00

Optical properties

In order to calculate the absorption and scattering efficiencies, we will be using a software package called `optool` [1][2][3][4], available from `https://github.com/cdominik/optool` , and developed by Carsten Dominik, based on the routines written by Michiel Min and Ryo Tazaki.
This package allows the conversion from n and k values to absorption and scattering efficiencies (including a full scattering matrix) for different grain shapes, including Mie theory, hollow spheres, fractal grains, and continuous distribution of ellipsoids. It also allows for the combination of different materials in one grain, including vacuum, to represent porosity.
`optool` calculates the mass absorption coefficients κ, rather than Q, as this is more meaningful for the general, e.g. non-spherical case. κ gives the extinction cross-section (in cm^2) per mass (in g) of material. Derive that for the spherical case, κ and Q are related as:

$$\kappa = \frac{3}{4}\frac{1}{\rho_\mathrm{d}}\frac{Q}{a}$$

where a is the radius of the grain, and ρ_d the bulk density.

Installing optool

First, we will install `optool` and check the (existing) installation
Download `optool` into a directory other than your `~/bin/` directory, for instance in a specially created directory for this session:

```
mkdir ~/handsonI
cd ~/handsonI  # or other directory of your choice
git clone https://github.com/cdominik/optool.git
```

enter the code directory and compile optool with multicode support:

```
cd optool
make multi=true
```

copy the executables to your binary path:

```
make install bindir=~/bin/
```

You should now have the following executables in your binary path: `optool` , `optool2tex` , and `optool-complete` . You can now run the program by typing `./optool` in the command line while in the directory where the executable is stored, or simply `optool` , if it is in your binary path.
In order to test the installation run `optool` with the following options in the command line:

```
optool -c pyr 0.87 -c c 0.13 -p 0.25
```

which mixes 87% of pyroxene with 13% of carbon, to which a 25% volume fraction of vacuum is added to simulate porosity. A full description of the options are given in the user guide, which can be downloaded from **https://github.com/cdominik/optool/blob/master/UserGuide.pdf** (https://github.com/cdominik/optool/blob/master/UserGuide.pdf)
Optional: There is also a python module, which can be installed by typing

```
pip install -e .
```

while in the `optool` code directory. It is generally a good idea to activate a python environment, prior to installing and using python module, e.g.

```
source ~/python-virtual-environments/env/bin/activate.csh
```

For more information on python environments, see e.g. **https://realpython.com/python-virtual-environments-a-primer/** (https://realpython.com/python-virtual-environments-a-primer/)

Inspecting the input and output

`optool` comes with the optical properties of a number of different bulk materials built. In order to list the built-in optical properties, type

```
optool -c
```

and study the output. From this we can deduce that our previous command `optool -c pyr 0.87 -c c 0.13 -p 0.25` uses pyroxene of the composition $Mg_{0.7}Fe_{0.3}SiO_3$[5], and amorphous carbon[6].
Now we will print the optical constants for the pyroxene and carbon used, and also the resulting output.
Use your favorite plotting routine, for instance in python, to plot the n and k values of the pyroxene and carbon used, as a function of wavelength, and also plot the resulting $\kappa_{\rm abs}$ and $\kappa_{\rm sca}$ values as a function of wavelength.
The built-in optical constants can be found in `~/handsonI/optool/lnk_data`. The resulting κ values can be found in the working directory, in file `dustkappa.dat`.

Playing with different grain models

Now that we know how to run `optool`, we can play with different grain models. For instance:

- **vary compositions**: change the ratio between carbon and pyroxene in the example above, and see how the resulting κ spectrum changes. Make a plot where you go from pure pyroxene grains to pure carbon grains, with steps of 20%.

- **vary porosity**: now for the default composition above, change the porosity, in steps of 25%.

You can also try some other species from the built-in library. However, the built-in library has its limitations, and you may be interested in trying some other species.

- **alternative optical properties**:
    - Start with the **extended optool library**, which can be found in a separate repository: **https://github.com/cdominik/optool-additional-refind-data** (https://github.com/cdominik/optool-additional-refind-data)

  Inspect the carbonates, and create a dust grain that contains 10% carbonates in addition to 90% pyroxene.
  - You can also read in **your own choice of optical properties**, for instance $Mg_{0.5}Fe_{0.5}O$[7] from **https://www.astro.uni-jena.de/Laboratory/OCDB/mgfeoxides.html** (https://www.astro.uni-jena.de/Laboratory/OCDB/mgfeoxides.html). Be aware that you may need to edit the lnk file to be readable by `optool`, please consult the User's Manual.

- **vary grain sizes**: to study the effect of grain sizes we will use amorphous olivine grains of the $MgFeSiO_4$ composition (abbreviated as `ol` in `optool`). We do not mix this with another material and also do not add any porosity.
  - Use the `-a` option in optool to calculate the opacities for grains of 0.1 μm and 0.2 μm in size (separately), and compare the appearance over the 8-13 μm wavelength range in a plot.
  - Look up the details of the MRN grain size distribution[8], and model the opacity of grains of standard composition (see above) with this grain size distribution.

- **vary grain shapes**: The effect of grain shapes is best studied with a material that has many sharp resonances, such as forsterite (Mg-rich crystalline olivine), which is abbreviated in the built-in dust library as `for`. Do not use any other dust species or vacuum. Use a single grain size of 0.1 micron. Over the 5-50 micron range, plot the resulting opacities for spherical (Mie) grains, hollow spheres, a continuum distribution of ellipsoids (CDE) and fractal particles (MMF) in a single plot. What do you conclude?

Radiative transfer

Radiative transfer is the physical phenomenon of energy transfer in the form of electromagnetic radiation. The propagation of radiation through a medium is affected by absorption, emission and scattering processes. The equation of radiative transfer describes these interactions mathematically. In astrophysics, radiative transfer allows to better understand the medium through which radiation travels from stars to us. In our particular context, by solving the equation of radiative transer we can investigate the effect of dust in the emission that we detect.
During the last decades, several softwares have been developed to solve the radiative transfer in multiple astrophysical environments (galaxies, clouds, disks, stars, ...). Some of them are **RADMC-3D** (https://www.ita.uni-heidelberg.de/~dullemond/software/radmc-3d/), **LIME** (https://lime.readthedocs.io/en/v1.6.1/usermanual.html?highlight=re), **MCFOST** (https://mcfost.readthedocs.io/en/latest/test_suite.html), **hyperion** (https://www.hyperion-rt.org/). In this hands-on session we will use **SKIRT** (https://skirt.ugent.be/root/_home.html) developed at the Astronomical Observatory of the Ghent University (Belgium).

Installing SKIRT

SKIRT can be installed in Linux/Unix and MacOS. The **installation guide** (https://skirt.ugent.be/root/_installation_guide.html) provides additional support and requirements if you encouter problems. We will install SKIRT in the directory that we created before for this session `handsonI` :

```
cd ~/handsonI  # or the directory of your choice
mkdir SKIRT
cd SKIRT
mkdir release run git
```

This will create three directories: `release` , `run` , and `git` , that will be used later. We can now transfer the code from its git repository:

```
cd ~/handsonI/SKIRT
git clone https://github.com/SKIRT/SKIRT9.git git
```

To build SKIRT for the first time, using default build options, enter the following commands:

```
cd ~/handsonI/SKIRT/git
./makeSKIRT.sh
```

You may need to make `configSKIRT.sh` and `makeSKIRT.sh` exectuable using `chmod +rw`
If thins go well, you should see an output similar to this:

```
Using /usr/local/bin/cmake to generate build files

-- The C compiler identification is AppleClang 8.0.0.8000042
-- The CXX compiler identification is AppleClang 8.0.0.8000042
...
[  0%] Built target timestamp
[  0%] Building CXX object SMILE/build/CMakeFiles/build.dir/BuildInfo.cpp.o
[  0%] Linking CXX static library libbuild.a
...
[ 98%] Building CXX object SKIRT/core/CMakeFiles/skirtcore.dir/ZubkoSilicateGrai
[ 99%] Linking CXX static library libskirtcore.a
[ 99%] Built target skirtcore
[ 99%] Building CXX object SKIRT/main/CMakeFiles/skirt.dir/SkirtCommandLineHandl
[ 99%] Building CXX object SKIRT/main/CMakeFiles/skirt.dir/SkirtMain.cpp.o
[100%] Linking CXX executable skirt
[100%] Built target skirt
```

Because of size limitations in GitHub repositories, the resource data files needed by the SKIRT code are hosted elsewhere on the Ghent University and must be downloaded separately. For this, you have to do:

```
cd ~/handsonI/SKIRT/git
./downloadResources.sh
```

The script will ask confirmation before starting the download of a resource package that has not yet been installed.
Always answer `yes` for the “Core” resource package (you can skip the other resource packages, and download them later simply by running the script again)

```
Do you want to download and install SKIRT9_Resources_Core version 6? [y/n] y

Downloading SKIRT9_Resources_Core_v6.zip ...
------------------------------------------------
--2023-06-30 14:46:08--  https://sciences.ugent.be/skirtextdat/SKIRT9/Resources/S
...
inflating: SKIRT9_Resources_Core/SingleGrain/MeanPascucciBenchmarkOpticalProps.s
------------------------------------------------
```

To provide easy access to the executables in the SKIRT code, we recomment that you edit your login script (.profile, .bashrc, .bash_profile or equivalent) to add the appropriate SKIRT executable paths to your system path. For example, add the following line:

```
export PATH="${HOME}/handsonI/SKIRT/release/SKIRT/main:${PATH}"
```

To verify your installation of the SKIRT code, open a new terminal window and execute:

```
skirt
```

If the SKIRT code has been successfully installed, you should see an output similar to this:

```
30/06/2023 14:51:46.743   Welcome to SKIRT v9.0 (git 587faab built on 30/06/2023
30/06/2023 14:51:46.744   Running on dune.local for asanchez
30/06/2023 14:51:46.750   Interactively constructing a simulation...
30/06/2023 14:51:46.750 ? Enter the name of the ski file to be created:
```

First steps with SKIRT

The **User Guide** (https://skirt.ugent.be/root/_user_guide.html) of SKIRT is a comprehensive document describing the main components and usage of SKIRT as well as some hands-on tutorials of different levels.
As first steps with SKIRT, we will follow the recommendations of the SKIRT User Guide and start with the simplest case: “Monochromatic simulation of a dusty galaxy” to get to know better the code and how to generate input files and first results. For this, follow the instructions at **https://skirt.ugent.be/root/_tutorial_basics_mono.html** (https://skirt.ugent.be/root/_tutorial_basics_mono.html)
IMPORTANT.- Note that some of the interactive menus are slightly different between the current version of SKIRT and the one listed in the User Guide (this is because new options have been implemented in the code).
Once you ave defined all your conditions for your model, you should see in the terminal:

```
Successfully created ski file 'MonoDisk.ski'.
To run the simulation use the command: skirt MonoDisk
```

To actually run the simulation, enter the following command (within the same directory where the `MonoDisk.ski` file is located):

```
skirt MonoDisk
```

Once solving the radiative transfer is finished you will have some output files:

```
MonoDisk_log.txt           # containing the progress log as it was written in th
MonoDisk_parameters.xml    # containing a copy of the ski file

MonoDisk_cut_dust_t_xy.fits # t ("theoretical") single-frame FITS data iles provi
MonoDisk_cut_dust_t_xz.fits #

MonoDisk_cut_dust_g_xy.fits # g ("gridded") single-frame FITS data files of the g
MonoDisk_cut_dust_g_xz.fits #

MonoDisk_i88_total.fits    # FITS file containing the data cube with the total s
                           # since the simulation has only one wavelength, there
```

In order to visualize the results, you can open FITS files using softwares such as **SAOImage DS9**

(https://sites.google.com/cfa.harvard.edu/saoimageds9)

Adjusting the ski file

There is no need to go through the lengthy interactive process to re-run a simulation with slightly adjusted parameters. You can manually edit the `ski` file instead. Let's do some tests.
Duplicate the file `MonoDisk.ski` to a new file named `MonoDisk2.ski` and open it with a text editor. This is an xml file with all the parameters that were defined previously. You can make the following changes:

- increase the number of photon packets by a factor of five (replace `1e6` by `5e6`)
- add a second instrument by duplicating the `<FrameInstrument .../>` element
- change the new instrument's name to `i84` and adjust its inclination to 84 degrees

Save the changes and execute the adjusted simulation using the command

```
skirt MonoDisk2
```

Note that the simulation will now be substantially slowed because it is launching more photon packets and because there are more instruments.
Look at the differences between output files

- MonoDisk_i88_total.fits vs MonoDisk2_i88_total.fits
- MonoDisk2_i84_total.fits vs MonoDisk2_i88_total.fits

Playing with SKIRT

The potential of SKIRT is huge, as demonstrated by the multiple **scientific publications** (https://skirt.ugent.be/root/_publications_gallery.html) that make use of it to explain and reproduce their observations. In this hands-on tutorial we can not explore all of it, but we can suggest some directions to play with:

- How does the monochromatic galaxy that we simulated look like at different wavelengths? Explore how the images look like in the ultraviolet, in the near-infrared, far-infrared and even (sub)millimetric wavelengths
- How does the emission at different wavelengths look like if we completely change the angle from close edge-on (90 degrees) to close face-on (0 degrees)?
- We can generate a multi-wavelength (pan-chromatic) simulation following the instructions of the second tutorial: **https://skirt.ugent.be/root/_tutorial_basics_pan.html** (https://skirt.ugent.be/root/_tutorial_basics_pan.html)

References

1. **Dominik, C., Min, M. & Tazaki, R. 2021, Optool, 1.9, Astrophysics Source Code Library, ascl:2104.010** (https://ui.adsabs.harvard.edu/abs/2021ascl.soft04010D) ↩
2. **Min, M. et al. 2005, A&A, 432, 909** (https://ui.adsabs.harvard.edu/abs/2005A%26A...432..909M) ↩
3. **Tazaki, R. & Tanaka,H. 2018, ApJ 860, 79** (https://ui.adsabs.harvard.edu/abs/2018ApJ...860...79T) ↩
4. **Woitke, P. et al. 2016, A&A 586, 103** (https://ui.adsabs.harvard.edu/abs/2016A%26A...586A.103W) ↩
5. **Dorschner, J. et al. 1995, A&A 300, 503** (https://ui.adsabs.harvard.edu/abs/1995A%26A...300..503D) ↩
6. **Zubko, V. G. et al. 1996, MNRAS 282, 1321** (https://ui.adsabs.harvard.edu/abs/1996MNRAS.282.1321Z) ↩

7. **Henning, T. et al. 1995, A&AS 112, 143** (https://ui.adsabs.harvard.edu/abs/1995A%26AS..112..143H) ↩

8. **Mathis, J. S., Rumpl, W., Nordsieck, K. H. 1977, ApJ 217, 425** (https://ui.adsabs.harvard.edu/abs/1977ApJ...217..425M) ↩