OpTool User Guide

C. Dominik

June 19, 2020

Introduction

This tool allows to produce quite complex dust opacities right from the command line, without reference to input files or any surrounding setup structure. It is inspired by and derived from Michiel Min's OpacityTool for the DIANA program, and committed to its simplicity. For a more wide-ranging toolset to produce opacities, you should definitely check out SIGMA, the *Simple Icy Grain Model for Aggregates*[1], by Lefévre et al. (2020), which is also derived from OpacityTool, and from which `optool` has adopted the effective medium mixing tool set.

Capabilities

- Fully command line driven
- Combining materials through mixing into a complex grain
- Built-in materials, but easily used also with external data
- DHS to model shape effects and low-porosity aggregates

Terms of use

The code doing the real work of `optool` was adopted and modified from the publicly available versions of the OpacityTool and SIGMA. I focused on simplicity and introduced my preferred set of conventions. If you are using this tool, you should give credit to the papers describing the methods used to compute the opacities, as well as to the original laboratory papers that published the refractive index measurements.

- Third party software: Toon et al. 1981, Applied Optics 20, 3657
- DIANA standard Opacities: Woitke, Min et al. 2016
- SIGMA: Lefévre, Min et al. 2020, A&A (submitted)
- DHS model for irregular grains: Min et al. 2005, A&A, 432, 909
- References to refractive index data used in your particular application.

[1] Sigma has significant additional capabilities like non-powerlaw size distributions, size-dependent porosities, and adding several different independent grain types in a single run to produce a combined opacity and input files for a variety of radiative transfer codes. It also has the capability to reproduce and easily modify many classical dust models in a computationally extremely efficient way.

Compiling `optool`

On most systems `optool` will compile by simply doing the following steps

```
git clone https://github.com/cdominik/optool.git
cd optool
make
```

This creates the executable `optool`, which you should put on your execution path. The `Makefile` uses the GNU compiler `gfortran`. To use the Intel compiler instead, run

```
make ifort=true
```

To compile with support for writing fits files, install `cfitsio` and use

```
make fitsio=true
```

Examples

A simple grain made only of the default pyroxene

```
optool -c pyr
```

Reproduce the DIANA standard dust model, using the pyroxene with 70% magnesium, and 13% (by mass) carbon, and 25% porosity.

```
optool -c pyr-mg70 0.87   -c c-z 0.13   -p 0.25
```

List the built-in materials

```
optool -c ?
```

Add an ice mantle that is 20% of the core mass

```
optool -c pyr-mg70 0.87   -c c-z 0.13   -m ice-w 0.2 -p 0.25
```

Like the previous example, but use ice refractive index data from a separate file.

```
optool -c pyr-mg70 0.87   -c c-z 0.13   -m ~/lnk/ice_hudgins_2.dat 0.2 -p 0.25
```

Pure ice grains in a narrow size distribution from 1 to 3 microns, with 15 sample sizes following an $f(a) \propto a^{-2.5}$ powerlaw size distribution. Also, restrict the wavelength range to 10-100μm.

```
optool -c ice-w 1.0 -a 1 3 2.5 15 -l 10 100
```

Represent the default dust model (DIANA) in 42 grain sizes, and produce input files with full scattering matrix for RADMC-3D.

```
optool -na 42 -d -s -radmc3d
```

Command line options

`-h, -help`
: Show a compact help message about command line options.

Grain composition and geometry

`-c ?`
: List available built-in materials (the keys for the **-c** and **-m** options).

`-c KEY-or-FILE [Mfrac]`
: Specify a material to include into the grain. `KEYorFILE` can be an abbreviation key for a builtin material, or the path to the correct `lnk` file. `Mfrac` is the *mass* fraction of the material. You can give up to 10 **-c** switches to add the different contributions. All materials will be mixed together using the Bruggeman rule, and vacuum can be included through the porosity.

`-m KEY-or-FILE [Mfrac]`
: Like **-c**, but use this material as a mantle material that will be mixed in using the Maxwell-Garnett rule instead of the Bruggeman rule. Only one **-m** switch is allowed, and it needs to come after all **-c** specifications.

`-p POROSITY [P_MANTLE]`
: Porosity, the *volume* fraction of vacuum, a number smaller than 1. The default is 0. A single value will apply to both core and mantle, but a second value will be specific for the mantle (and may be 0).

`-fmax VHMAX`
: Maximum *volume* fraction of the inner hole for the DHS approach. The default is 0.8. Zero means to use solid spheres, i.e. perfectly regular grains.

Grain size distribution

`-a AMIN AMAX [NA [APOW]]`
: Specify minimum and maximum grain radius, and optionally the number of size bins and the size distribution powerlaw. You can also use the options **-amin**, **-amax**, **-na**, **-apow**, to set individual values. The defaults are 0.05 μm, 3000 μm, 100, and 3.5, respectively.

Wavelength grid

`-l LMIN LMAX [NLAM]`
: Minimum and maximum wavelength and optionally the number of wavelengths points for the construction of the wavelength grid. The default values are 0.05 μm, 10000 μm, and 300, respectively. Use the options **-lmin**, **-lmax**, and **-nlam** to set individual values.

`-l FILE`
: Read the wavelength grid from the first column of FILE. The first number in the file must be the number of grid points. So this could be an `lnk` file.

Controlling the output

The standard output is the file `dustkappa.dat`, with the opacities and the asymmetry parameter g. The following options lead to extended output. See the section Output files for details on the files that can be produced.

`-s` Include the full scattering matrix in the output. Without this switch, the output is limited to just $\kappa_{\rm abs}$, $\kappa_{\rm sca}$, and g.

`-t [TMIN [TMAX [NT]]]`
Compute mean opacities per g of dust mass, $\kappa_{\rm Planck}$ and $\kappa_{\rm Rosseland}$, in the given temperature interval, in `nt` steps, and write them to the file `dustkapmean.dat`. The parameters default to 10K, 10000K, and 200.

`-d [NSUB]`
Divide the computation up into `NA` parts to produce a file for each grain size. Each size will actually be an average over a small range of NSUB grains around the real size, to smear out resonances. The default for NSUB is 5.

`-fits`
Write `dustkappa.fits` with the absorption cross sections and scattering matrix elements, instead of ASCII output. With the `-d` switch, `NA` such files will be written, one for each grain size.

`-radmc3d [LABEL]`
The structure of the ASCII output of `optool` is already compatible with RADMC-3D. However, RADMC-3D uses a different normalization for the scattering matrix, see the appendix for more information. With this switch, the extension of the files will be changed to `.inp`, the normalization will be modified, and if you specify LABEL, it will be used in the file name(s).

Material properties

`optool` needs refractive index data to work. For your convenience, a useful list of materials is compiled into `optool`, but you can also find and use your own data. No matter where the data is from, you should *always* cite the original laboratory papers.

Built-in materials

To access one of the built-in materials, specify the corresponding key string like `pyr-mg70` instead of the path to an `lnk` file. In each material class I have selected a useful default, accessible with an even simpler generic key. The built-in list might be enough for opacity computation. If your goal is detailed mineralogical fits, you will have to use specialized refractive index data.

-c Key generic	**-c Key** full key	**Material**	**State**	ρ [g/cm^3]	$\lambda_{\min}$ [μm]	**Reference**
	pyr-mg100	$MgSiO_3$	amorphous	2.71	0.2	Dorschner+1995
	pyr-mg95	$Mg_{0.95}Fe_{0.05}SiO_3$	amorphous	2.74	0.2	Dorschner+1995
	pyr-mg80	$Mg_{0.8}Fe_{0.2}SiO_3$	amorphous	2.9	0.2	Dorschner+1995
pyr	pyr-mg70	$Mg_{0.7}Fe_{0.3}SiO_3$	amorphous	3.01	0.2	Dorschner+1995
	pyr-mg60	$Mg_{0.6}Fe_{0.4}SiO_3$	amorphous	3.1	0.2	Dorschner+1995
	pyr-mg50	$Mg_{0.5}Fe_{0.5}SiO_3$	amorphous	3.2	0.2	Dorschner+1995
	pyr-mg40	$Mg_{0.4}Fe_{0.6}SiO_3$	amorphous	3.3	0.2	Dorschner+1995
ens	pyr-c-mg96	$Mg_{0.96}Fe_{0.04}SiO3$	crystalline	2.8	**2.0**	Jäger+1998
ol	ol-mg50	$MgFeSiO_4$	amorphous	3.71	0.2	Dorschner+1995
	ol-mg40	$Mg_{0.8}\ Fe_{1.2}\ SiO_4$	amorphous	3.71	0.2	Dorschner+1995
for	ol-c-mg100	$Mg_2\ SiO_4$	crystalline	3.33	**3.0**	Steyer+1974
c	c-z	C	amorphous?	1.8	0.05	Zubko+1996
	c-p	C	amorphous	1.8	0.11	Preibisch+1993
ice	ice-w	Water ice	crystalline	0.92	0.04	Warren+2008
iron	fe-c	Fe	metallic iron	7.87	0.1	Henning+1996
cor	cor-c	Al_2O_3	crystalline	4.0	0.5	Koike+1995

External refractory index files (`lnk` files)

`optool` can also use external refractive index data in files with the following format:

- The file may start with several comment lines (lines starting with `!`, `#`, or `*`).
- The first non-comment line contains two numbers, the number of wavelengths N_λ and the specific weight ρ of the material in g/cm^3
- Starting in the next line, each line has the wavelength in micrometer, and then the real and imaginary values of the refractive index, n and k.

You can use any of the refractory index files provided by Charléne Lefévre's SIGMA package out of the box, or find other files, for example by using the Jena database. Don't forget to add the line with number of grid points and specific density! If for some reason it is not convenient to add that line to the file, `optool` will count the lines for you and you can specify the density (3.42 in this example) after the mass fraction, like this:

```
optool -c path/to/file.lnk 0.7 3.42
```

Output files

dustkappa.dat

This is an ASCII file containing the basic opacity results. It starts with a comment section describing the dust model, followed by the format number (3, currently), followed by the number of wavelengths in the grid, both on lines by themselves. Then follows a block with these columns:

1. wavelength λ [micron]
2. mass absorption cross section $\kappa_{\rm abs}$ [cm^2/g]

3. mass scattering cross section $\kappa_{\rm sca}$ [cm^2/g]
4. asymmetry parameter g

dustkapscatmat.dat

ASCII file with cross sections and full scattering matrix. The comment section at the stars of the file explains the structure. See the appendix for information about the normalization of the scattering matrix. And see the `-radmc` switch which will modify the output to make sure it can be used as an input file for RADMC-3D.

dustkappa.fits

The fits-file (ending in '.fits') is written instead of the ASCII output when using the `-fits` switch. It has two HDU blocks. The first block contains the cross sections per unit mass. This is a $N_\lambda \times 5$ matrix with these columns:

1. wavelengths in [μm]
2. mass extinction cross section $\kappa_{\rm ext}$ in [cm^2/g]
3. mass absorption cross section $\kappa_{\rm abs}$ in [cm^2/g]
4. mass scattering cross section $\kappa_{\rm sca}$ in [cm^2/g]
5. asymmetry factor g

The second HDU block contains the scattering matrix elements. This is a $N_\lambda \times 6 \times 180$ matrix. These are the 6 elements of the scattering matrix for 180 equidistant scattering angles from forward scattering (element 0) to backward scattering (element 179) for each wavelength value. The 6 elements stored are F_{11}, F_{12}, F_{22}, F_{33}, F_{34}, F_{44} respectively.

dustkapmean.dat

This file will only be written with the `-t` switch. It contains 3 columns: (1) T [K], (2) $\kappa_{\rm Planck}$, (3) $\kappa_{\rm Ross}$, both in cm^2 per gram of *dust*. Note that dust evaporation is not considered, and that a wide wavelengths coverage is needed for good results.

Inspecting the computed optical properties

To try out `optool` you could use one of these commands

```
make test              # for a simple size-integrated opacity computation
    ... or ...
make testdiv           # for opacities as a fuction of grain size
```

The commands will run `optool` with the standard DIANA material properties, and then use the python script `optool.py` to plot the computed opacities. You will get two plots

- a plot showing the opacities $\kappa_{\rm abs}$, $\kappa_{\rm sca}$, and $\kappa_{\rm ext}$ as a function of wavelength, along with the asymmetry parameter g. Note that the red g curve does not have its own scale, imagine the y axis going from 0 to 1 for g.
- a plot showing the scattering matrix elements as a function of scattering angle, and with sliders to go through grain sizes and wavelengths. The y axis of the plot is actually $\log_{10}(|F_{ij}|)$, so the downward peaks are actually places where the matrix element goes through zero.

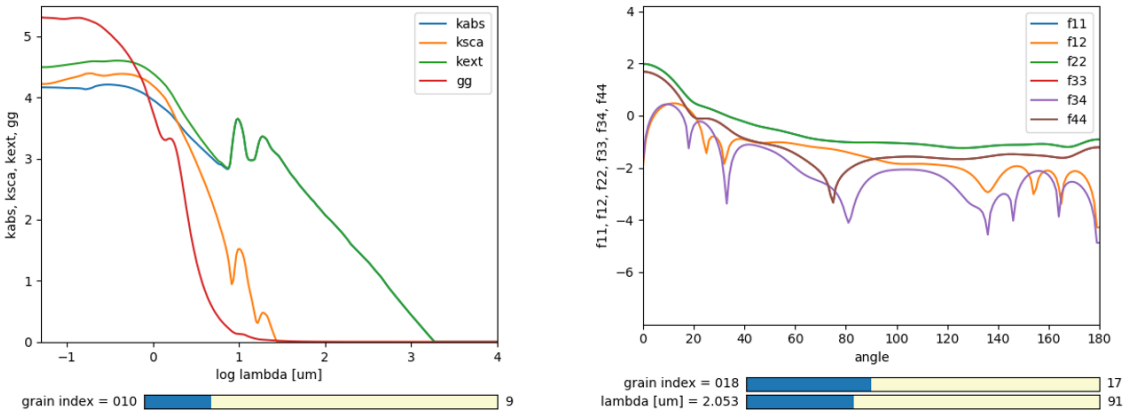

Figure 1: Screenshot of the plots created by running `ipython -i optool.py`. Note that we plot the logarithm of the absolute value of the scattering matrix, in order to deal with range and sign issues.

Acknowledgments

- Michiel Min for the DIANA OpacityTool and all the work that went into it. `optool` is strongly based on it, and this user guide reuses some of his description.
- Charléne Lefévre for SIGMA. `optool` uses the generalized Bruggeman solver and a few other bits from SIGMA's code base.
- Kees Dullemond for his python plotting routine `viewarr` (available on github), and code for computing Planck and Rosseland means.
- Jeroen Bouwman for pointers to refractive index data.

Appendix

How to ingest another material

Additional data tables can be compiled into the code. Here is how to do this:

1. Give your lnk file a name exactly like `pyr-mg70-Dorschner1995.lnk`, where `pyr-mg70` is the key to access the material and `Dorschner1995` is the reference.
2. Put this file into the `lnk_data` directory.
3. Edit `lnk_data/lnk-help.txt`, so that `optool -c ?` will list the new material.
4. Run `make ingest` to update `ref_ind.f90`, now including your new material.
5. Recompile and install the code.

Scattering phase function normalization

A number of different normalizations for the scattering matrix are being used in the literature and in computational tools. The various versions can be quite different and it is important to be aware of the choice. For `optool` we are using a convention in which the average over all directions of the 1-1 element of the scattering matrix equals unity, i.e.

$$\oint_{(4\pi)} F_{11}(\lambda, \Theta) d\Omega = 4\pi \tag{1}$$

See Hovenier (2004) for a discussion of this normalization. `optool` can also produce output for RADMC-3D which uses a different normalization, namely

$$\oint_{(4\pi)} Z_{11}(\lambda, \Theta) d\Omega = \kappa_{\rm sca}(\lambda) \tag{2}$$

You can also find an extensive discussion about the scattering matrix and it's normalization in the RADMC-3D manual.

Bibliography

- Dorschner,J. et al. 1995, A&A 300, 503
- Henning, Th. and Stognienko, R. 1996, A&A 311,291
- Jäger,C. et al. 1998, A&A 339, 904
- Koike,C. et al. 1995, Icarus 114, 203
- Lefèvre,C.; Min,M. et al. 2020, A&A (submitted)
- Min,M. et al, 2005, A&A, 432, 909
- Min,M. et al. 2016, A&A, 585, 13
- Toon,O. & Ackerman,T. 1981, Applied Optics 20, 3657
- Woitke,P.; Min,M. et al. 2016, A&A 586, 103
- Preibisch,Th. et al. 1993, A&A 279, 577
- Steyer,T. 1974, PhD Thesis, The University of Arizona
- Warren,S. and Brandt,R. 2008, JGRD,113, D14220
- Zubko,V. et al. 1996, MNRAS 282,1321
- Hovenier, J, 2004, Report available on ADS.