OpTool User Guide

Carsten Dominik, Michiel Min, and Ryo Tazaki

December 2020, version 1.5

1. Introduction

This tool produces complex dust particle opacities right from the command line. It is derived from Michiel Min's DHS OpacityTool and also implements Ryo Tazaki's MMF theory for highly porous aggregates.

1.1. Capabilities

- stand-alone tool, fully command line driven, no input files need to be edited
- full scattering matrix output in several formats, including for RADMC-3D
- combining materials through mixing into a complex grain with porosity
- a useful collection of built-in materials for standard applications in astronomy
- an easy way to use external refractive index data for more specialized applications
- two computational methods: (i) **DHS (Distribution of Hollow Spheres)** for *irregular grains* and *low-porosity* aggregates. Standard **Mie theory** for *perfect spheres* is included as a limiting case. (ii) **MMF (Modified Mean Field)** theory for *high-porosity/fractal aggregates.*
- `Python` interface module for easy postprocessing and cross-tool use.

1.2. Terms of use

`optool` is distributed under the MIT license and can be used, changed and redistributed freely. The relevant references for this tool are listed below. Depending on which parts of `optool` you are using, please reference the corresponding papers.

- `optool`: `https://github.com/cdominik/optool.git`
- DHS model for irregular grains: Min et al. 2005, A&A, 432, 909
- MMF model for aggregates: Tazaki & Tanaka 2018, ApJ 860, 79
- DIANA standard Opacities: Woitke, Min et al. 2016, A&A 586, 103
- Third party software: Toon et al. 1981, Applied Optics 20, 3657
- References to refractive index data used in your particular application.

2. Examples

A simple grain made only of the default pyroxene, for the default grain size distribution ($a^{-3.5}$ powerlaw from 0.05 to 3000μm), on the default wavelength grid (0.05μm to 1cm).

```
optool pyr
```

Include the scattering matrix in the produced output

```
optool pyr -s
```

Reproduce the DIANA standard dust model, using a specific pyroxene (70% Mg) and carbon, in a mass ratio 0.87/0.13, and with a porosity of 25%.

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

List the built-in materials

```
optool -c
```

Add an ice mantle (built-in data from Warren+08) that is 20% of the core mass

```
optool pyr-mg70 0.87  c 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 pyr-mg70 0.87  c 0.13  -p 0.25  -m data/ice_hudgins.dat 0.2
```

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, and turn off DHS to get perfect spheres.

```
optool ice  -a 1 3 2.5 15  -l 10 100 -fmax 0
```

For silicon carbide, compute the opacity of a single grains size (2.5μm) at λ=8.9μm.

```
optool -a 2.5 -l 8.9 sic
```

Represent the default dust model (DIANA, you also get this when you do not give any materials at all) in 42 grain sizes, and produce input files for RADMC-3D, one for each grain size, with full scattering matrix, chopping 3 degrees from the scattering peak.

```
optool -na 42 -d -s -radmc -chop 3
```

Use MMF to compute the opacities of dust aggregates made of pyroxene monomers. Use a monomer radius of 2.5 microns to construct aggregates with radii of gyration in the range between 1 and 3 cm, and a filling factor of 5% (i.e. a porosity of 0.95)

```
optool pyr  -a 1e4 3e4  -mmf 2.5  -p 0.95
```

3. Compiling `optool`

On most systems, you can download and compile `optool` with these simple steps, using the freely available GNU FORTRAN compiler `gfortran`.

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

The executable is called `optool`, and you should put it on your execution path. To use the Intel fortran compiler, to use multiple cores for speed (highly recommended if your system supports it), or to be able to write FITS files[1], use one or more of the following parameters during compilation:

```
make ifort=true        # use the Intel fortran compiler iFort
make multi=true        # enable multicore support
make  fits=true        # add support for writing FITS files[2]
```

You can also find binaries for Mac and Linux at my homepage.

4. Command line arguments

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

4.1. Grain composition

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

`[-c] KEY-or-FILE [MFRAC]`
Specify a material to include in the grain. `KEYorFILE` can be the key for a builtin material, or the path to the correct `lnk` file. `MFRAC` is the *mass* fraction (default 1.0) of the material. You can give up to 10 materials to build up the grain. Mass fractions do not have to add up to one, they will be renormalized. All materials will be mixed together using the *Bruggeman* rule, and vacuum can be added through the porosity. A **-c** switch before each `KEY-or-FILE` is optional.

`-m KEY-or-FILE [MFRAC]`
Like **-c**, but place this material into the grain mantle. Multiple mantle materials will be mixed using the Bruggeman rule, and than that mix will be added to the core using the *Maxwell-Garnett* rule. The **-m** is *not* optional, it must be present.

4.2. Grain geometry and computational method

`-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).

[1] This requires the `cfitsio` library to be installed on your system.

`-dhs [FMAX]`

Use the *Distribution of Hollow Spheres* (DHS, Min 2005) approach to model deviations from prefect sphericity and low-porosity aggregates. **This is the default method**. Spheres with inner holes with volume fractions between 0 and f_{max} (default 0.8) are averaged to mimic irregularities. f_{max}=0 means to use solid spheres (Mie theory), i.e. perfectly regular grains. For backward compatibility, **-fmax** can be used instead of **-dhs**.

`-mie`

Do a standard Mie calculation. This is short for **-dhs 0** (or **-fmax 0**).

`-mmf [AMONO [DFRAC]]`

Use *Modified Mean Field* theory (MMF, Tazaki 2018) to compute the opacities of highly porous or fractal aggregates. Monomers will have the composition given by the **-c** and **-m** switches. `AMONO` is the monomer radius (default 0.1μm). Particles will be aggregates with a *compact size* given by the **-a** switch, so that the number of monomers is $N_{\rm mono} = a^3/a^3_{\rm mono}$. The *porosity* (**-p**) specifies how the monomers should be spread out to reach a volume filling factor $f = 1-p$. `DFRAC` can optionally specify the fractal dimension. When `DFRAC` is present, **-p** will be ignored, and the effective porosity will vary with size.

4.3. Grain size distribution

`-a AMIN [AMAX [APOW [NA]]]`

Specify (minimum) grain radius, and optionally maximum grain radius, the size distribution powerlaw and the number of size bins. You may also use options to set individual values with **-amin**, **-amax**, **-apow**, **-na**. The defaults are 0.05 μm, 3000 μm, 3.5, and *10 per size decade with a fixed minimum of 5*, respectively. If only a single size is specified with **-a**, then a_{max}=a_{min} and n_a=1 are implied.

4.4. Wavelength grid

`-l LMIN [LMAX [NLAM]]`

Specifiy the (minimum) wavelength, and optionally the maximum wavelength and the number of wavelengths points for the construction of the wavelength grid. The default values are 0.05 μm, 10000 μm, and 300, respectively. You may also use the options **-lmin**, **-lmax**, and **-nlam** (or **-nl**) to set individual values. If only one wavelength is specified with **-l**, then λ_{max}=λ_{min} and n_λ=1 are implied.

`-l FILE`

Read the wavelength grid from `FILE`. The file may start with comment lines, and the first non-comment line needs to contain the number of wavelength values in the data block below it. In the data block, the first column is expected to hold the wavelength values, in μm. For example, an `lnk` file could be used here.

4.5. Controlling the output

The standard output is the file `dustkappa.dat`, with the opacities and the asymmetry parameter g. The following options control and extend the output.

`-o [DIR]`
: Put the output files in directory `DIR` instead of the current working directory. `./output` will be used if `DIR` is not specified.

`-s [NANG]`
: Include the full scattering matrix in the output. `NANG` can optionally specify the number of equally-spaced angular grid points to cover the range of angles between 0 and 180 degrees. The default for `NANG` is 180 and should normally not be changed.

`-chop [NDEG]`
: Cut out the first `NDEG` (2 if unspecified) degrees of the forward scattering peak and compensate by a reduction in the scattering cross section.

`-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` files will be written.

`-radmc [LABEL]`
: RADMC-3D uses a different angular grid and normalization for the scattering matrix, so the output has to be adapted for it. The extension of the files will be changed to `.inp`, and if you specify `LABEL`, it will be used in the file name(s).

5. 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 other data. No matter where the data is from, you should *always* cite the original laboratory papers.

5.1. 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.

-c Key generic	**-c Key** full key	**Material**	**State**	ρ g/cm^3	λ_{min} μm	λ_{max} μm	**Reference**
	pyr-mg100	$MgSiO_3$	amorph	2.71	0.2	500	Dorschner+95
	pyr-mg95	$Mg_{0.95}Fe_{0.05}SiO_3$	amorph	2.74	0.2	500	Dorschner+95
	pyr-mg80	$Mg_{0.8}Fe_{0.2}SiO_3$	amorph	2.9	0.2	500	Dorschner+95
pyr	pyr-mg70	$Mg_{0.7}Fe_{0.3}SiO_3$	amorph	3.01	0.2	500	Dorschner+95
	pyr-mg60	$Mg_{0.6}Fe_{0.4}SiO_3$	amorph	3.1	0.2	500	Dorschner+95
	pyr-mg50	$Mg_{0.5}Fe_{0.5}SiO_3$	amorph	3.2	0.2	500	Dorschner+95
	pyr-mg40	$Mg_{0.4}Fe_{0.6}SiO_3$	amorph	3.3	0.2	500	Dorschner+95
ens	pyr-c-mg96	$Mg_{0.96}Fe_{0.04}$SiO3	cryst	2.8	**2.0**	**99**	Jäger+98
ol	ol-mg50	$MgFeSiO_4$	amorph	3.71	0.2	500	Dorschner+95
	ol-mg40	$Mg_{0.8}\ Fe_{1.2}\ SiO_4$	amorph	3.71	0.2	500	Dorschner+95
for	ol-c-mg100	$Mg_2\ SiO_4$	cryst	3.33	**3.0**	250	Steyer+74
	astrosil	MgFeSiO4	mixed	3.3	6e-5	1e5	Draine+03
c	c-z	C	amorph?	1.8	0.05	1e4	Zubko+96
	c-p	C	amorph	1.8	0.11	800	Preibisch+93
gra	c-gra	C graphite	cryst	2.16?	0.001	1000	Draine+03
org	c-org	CHON organics	amorph	1.4	0.1	1e5	Henning+96
	c-nano	C nano-diamond	cryst	2.3	0.02	**110**	Mutschke+04
ice	ice-w	Water ice	cryst	0.92	0.04	2e6	Warren+08
iron	fe-c	Fe	metal	7.87	0.1	1e5	Henning+96
	fes	FeS	metal	4.83	0.1	1e5	Henning+96
	sic	SiC	cryst	3.22	0.001	1000	Laor93
cor	cor-c	Al_2O_3	cryst	4.0	0.5	**40**	Koike+95

5.2. External refractory index files (`lnk` files)

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

- The file may start with several comment lines (lines starting with `!`, `#`, or `*`).
- The next line contains two numbers, the number of wavelengths n_λ and the specific weight ρ of the material in g/cm^3
- Then follow three columns of data: $\lambda[\mu\mathrm{m}]$, and the real and imaginary parts of the refractive index, n and k.

You can find refractive index data in the Jena database, and associated with original papers in the literature. Don't forget to add the line with n_λ and ρ! If for some reason it is not convenient to add that line to the file, `optool` will count the lines and you can specify the density after the mass fraction, like this: `optool -c path/to/file.lnk 0.7 3.42`. The appendix contains information on how to compile frequently-used external materials into the program.

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

dustkapscatmat.dat

ASCII file with cross sections and full scattering matrix. The comment section at the start 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 an $\rm n_\lambda \times 4$ matrix with these columns:

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

The second HDU block contains the scattering matrix elements. It is a $\rm n_\lambda \times 6 \times n_{ang}$ matrix, containing the 6 elements of the scattering matrix for $\rm n_{ang}$ equidistant scattering angles from forward scattering (element 0) to backward scattering (element $\rm n_{ang}$-1), for each wavelength value. The stored matrix elements are $\rm F_{11}$, $\rm F_{12}$, $\rm F_{22}$, $\rm F_{33}$, $\rm F_{34}$, and $\rm F_{44}$.

7. Python interface

`optool` comes with a `python` module `optool.py` that runs `optool` in the background[2] and puts all computed quantities as `numpy` arrays into a python object. This makes it straight forward to inspect and further process the output, for example to produce custom opacity files for use in an radiative transfer tool. Here is how to use the module:

```
>>> import optool
>>> p = optool.particle('~/bin/optool pyr 0.8 -m ice 0.2 -na 24 -d')
```

The argument to `optool.particle()` must be a valid shell command[3] to run `optool`, if necessary with the full path to the `optool` binary. Depending on the presence of the

[2] The module runs the command as a subprocess, with output to a temporary subdirectory in the current working directory. It then reads the output files and cleans up the temporary directory - unless is called with the `keep` keyword argument: `optool.particle('optool',keep=True)`.

[3] The command may be given as string than can be split on whitespace, or, for example if the path to the binary contains whitespace, in list form `['/path/to my/command','arg1','arg2',...]`.

optool's **-d** switch, the command will produce opacities either for $n_p = 1$ particle, or for $n_p = n_a$ particles. Most of the attributes (with the exception of the global wavelength and angular grids) will therefore be arrays with the first dimension equal to n_p, even if $n_p = 1$. The object returned will have the following attributes:

Attribute	**Type/Shape**	**Quantity**
`cmd`	`string`	The full command given in the particle() call
`radmc`	`boolean`	Output follows RADMC conventions
`scat`	`boolean`	Scattering matrix is available
`nlam`	`int`	Number of wavelength points
`lam`	`float[nlam]`	The wavelength grid
`nang`	`int`	Number of scattering angles
`scatang`	`float[nang]`	The angular grid
`materials`	`[ [...]... ]`	Lists with location,m_{frac},ρ,material
`np`	`int`	Number of particles, either 1 or (with -d) n_a
`fmax`	`float[np]`	Maximum volume fraction of vacuum for DHS
`pcore`, `pmantle`	`float[np]`	Porosity of the core/mantle material
`amin`, `amax`	`float[np]`	min/max grain size used for each particle
`nsub`	`int[np]`	Number of sizes averaged for each particle.
`apow`	`float[np]`	Negative size distribution power law (e.g. 3.5)
`a1`, `a2`, `a3`	`fload[np]`	Mean <a>, <a^2>, and <a^3> of the particle
`kabs,ksca,kext`	`float[np,nlam]`	Absorption,scattering/extinction cross section
`gsca`	`float[np,nlam]`	Asymmetry parameter
`f11`, ..., `f44`	`float[np,nlam,nang]`	Scattering matrix element F_{11}, ... ,F_{44}
`chop`	`float[np]`	Degrees chopped off forward scattering
`plot()`	`method`	Plot the cross sections and matrix elements

The `optool.plot()` method will produce the following plots:

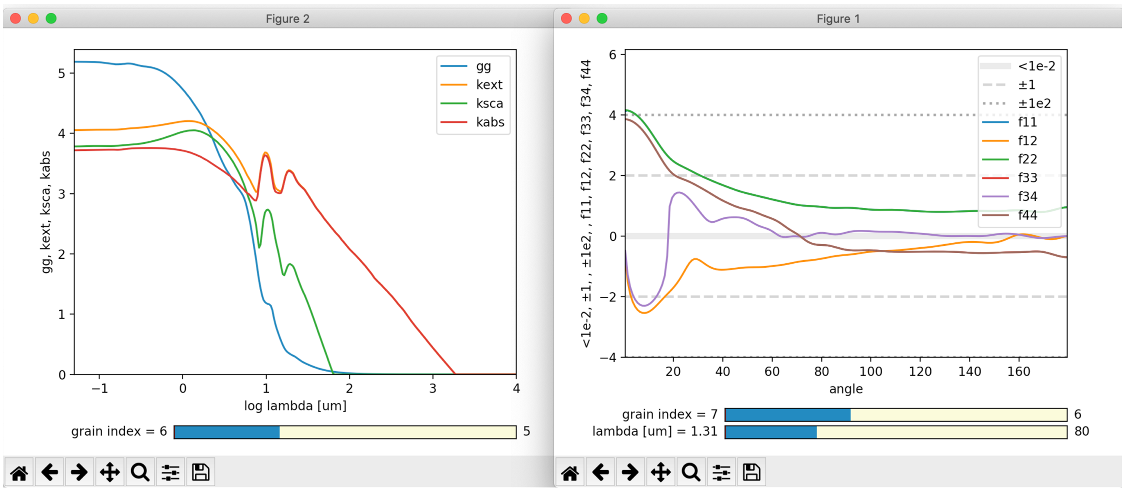

Figure 1: Screenshot of the plots created by running `p.plot()` on an optool particle.

- 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 (on a linear y-scale). Note that the blue g curve does not have its own axis, imagine the full y axis going from 0 to 1 for g.
- a plot showing the scattering matrix elements as a function of scattering angle, with sliders to go through grain sizes and wavelengths. When interpreting the y axis, note that we plot the positive/negative $\log_{10}$ of positive/negative matrix elements, compressing the range from 10^{-2} to 10^2 into a line (use the grey lines as a guide, ignore the y-axis labels).

8. Acknowledgments

- Charléne Lefévre for SIGMA, which triggered me to add a grain mantle using the Maxwell Garnett rule.
- Kees Dullemond for his python plotting routine `viewarr` (available on github).
- Jeroen Bouwman for some pointers to refractive index data.

A. Units

Due to conventions in our field, the input and output of `optool` uses the following units

- **microns** for grain sizes and wavelengths
- **g/cm$^{\mathbf{3}}$** for mass densities of materials
- **cm$^{\mathbf{2}}$ g$^{\mathbf{-1}}$** for opacities $\kappa_{\rm abs}$, $\kappa_{\rm sca}$, and $\kappa_{\rm ext}$
- **sr$^{\mathbf{-1}}$** or **cm$^{\mathbf{2}}$ g$^{\mathbf{-1}}$ sr$^{\mathbf{-1}}$** for the scattering matrix elements, see below.

B. Scattering Matrix: The fine print

B.1. Phase function normalization

A number of different normalizations for the scattering matrix are being used in the literature and in computational tools. The differences are significant, and it is important to be aware of the choice. For `optool` we are using a convention (Hovenier (2004)) in which the average over all directions of the 1-1 element of the scattering matrix equals unity, i.e.

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

`optool` can also produce output for RADMC-3D which uses instead

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

The books by Bohren & Huffman and by Mishchenko use different normalizations again. As described in RADMC-3D's manual, these conventions can be matched by scaling all matrix elements with simple factors involving dust mass and wavenumber $k = 2\pi/\lambda$.

B.2. Forward-scattering peak

Particles that are much larger than the wavelength of the considered radiation can show extreme forward scattering, were much of the *scattered* radiation is sent into just a few degrees around the forward direction. This can be difficult to handle for radiative transfer codes which have limited angular resolution or limited sampling. MCMax3D has the `nspike` keyword to deal with this issue. Other tools (e.g. RADMC-3D) require this to be taken care of by the process that creates the opacity files. The `-chop` switch specifies a number of degrees around the forward scattering direction. Inside that cone, the scattering matrix gets limited to the value at the edge of the cone. To compensate and ensure energy conservation, the scattering cross section will be reduced accordingly. As a result, the radiation that would be *scattered* into this narrow range of angles will be treated as if it did have *no interaction at all* with the grain.

B.3. Angular grid

`optool` uses an angular grid in one degree steps from 0 to 180 degrees. The full degrees are the cell *interfaces* of that grid. `optool` computes the scattering matrix at the cell *midpoints*, i.e. at 0.5°, 1.5° etc to 179.5°, for a total of 180 values. The scattering matrix is normalized in this way, so that a numerical integral gives the correct result.
RADMC-3D requires the values of the scattering matrix on the cell *boundaries*, so at 0°, 1° etc to 180°, for a total of 181 values. For the input files for RADMC-3D, we interpolate and extend the computed values to the cell boundaries.

C. How to ingest refractive index data for another material

Using external refractive index data means that you have to keep track of where those files are. It can be convenient to compile your favorite materials into `optool`, so that accessing them will be as simple as using the built-in materials. Here is how to do that:

1. Give your `lnk` file a name exactly like `pyr-mg70-Dorschner1995.lnk`, where the start of the name (`pyr-mg70`) is the key to access the material and `Dorschner1995` (the text after the final `-`) is the reference.
2. Put this file into the `lnk_data` directory.
3. Optionally edit `lnk_data/lnk-help.txt`, so that `optool -c` will list the new material. Note that, in order to define generic keys, optool looks for pairs that look like `genkey -> fullkey` in this file.
4. Run `make ingest` to update `ref_ind.f90`, now including your new material.
5. Recompile and install the code.

D. Internals

This appendix describes some key aspects of the internal workings of the code.

Refractive Index Data

Measured refractive index data is obtained from data compiled into the code, or read-in from a file. That data is then interpolated and extrapolated onto the wavelengths grid requested for the computation. Extrapolation toward short wavelengths is done keeping the refractive indices constant. Extrapolation toward long wavelength assumes that the last two measured data points define a powerlaw. Interpolation in the measured grid is done using double-logarithmic interpolation.

Mixing

Once the refractive index for all involved materials is available, the core and the mantle mixtures are created independantly, using the Bruggeman rule. Mass fractions are converted into volume fractions, and porosity is implemented using vacuum as an additional material. The subroutine doing the mixing uses an iterative procedure that is very stable, also for a large number of components.
If there is a mantle, the Maxwell Garnett rule is applied with the core being treated an inclusion inside a mantle matrix.

DHS

In order to simulate irregularities in grains (irregular shapes, or the properties of low-porosity aggregates), `optool` averages the opacities of grains with an inner empty region, over a range of volume fractions of this inner region between 0 and f_{max}. The subroutine used to compute the opacities and scattering matrix elements for these structures is `DMiLay` (Toon & Ackerman 1981). However, when the size parameter $x = 2\pi a/\lambda$ exceeds a value of 10^4, then no DHS averaging is used. A standard Mie calculation is performed , using the routine `MeerhoffMie` (reference missing), for a fixed size parameter of 5000, with proper scaling to the actual size of the particle.

MMF

To apply MMF, `optool` needs the number of monomers N, the fractal dimension D_f, and a scaling factor k_f to describe the structure of the aggregate. The size a of the particles as specified by the **`-a`** switch is interpreted as the *compact* size of all material in the aggregate, so that simply $N = a^3/a^3_{\mathrm{mono}}$, where a_{mono} is the radius of the monomer. Porosity is ignored for this consideration. If the user has specified a fractal dimension as the second numerical value of the `-mmf` option, it is used for all particle sizes, resulting in decreasing volume filling factors as a function of aggregate size (unless $D_f = 3$). If no fractal dimension has been specified, the porosity p is used instead to determine a constant filling factor $f = 1 - p$ for all aggregate sizes. D_f is then chosen (different for every size) to achieve that average volume filling fraction. That process also sets the fractal prefactor k_f in a way that the asymptotic density of small aggregates is the monomer material density. The following table summarises the relevant equations.

	`-mmf AMONO DF`	`-mmf AMONO -p p`
N	$= a^3/a^3_{\mathrm{mono}}$	$= a^3/a^3_{\mathrm{mono}}$
D_f	given by user	$= 3\ln N \,/\, \ln(N/(1-p))$
k_f	$= (5/3)^{D_f/2}$	$= (5/3)^{D_f/2}$

With the structure defined, `optool` then applies the formalism from Tazaki & Tanaka (2018) to compute cross sections and the scattering matrix. It also computes the phase shift caused by the aggregate to check the validity of the scattering matrix. If the conditions for accurate scattering matrix results are violated, a warning will be issued.

E. Bibliography

- Bohren, C.F. and Huffman, D.R. 1998, *Absorption and Scattering of Light by Small Particles*, Wiley-VCH
- Draine, B., 2003, ApJ 598, 1017
- Draine, B., 2003, ApJ 598, 1026
- Dorschner, J. et al. 1995, A&A 300, 503
- Henning, Th. and Stognienko, R. 1996, A&A 311,291
- Hovenier, J, 2004, Report available on ADS.
- Jäger, C. et al. 1998, A&A 339, 904
- Koike, C. et al. 1995, Icarus 114, 203
- Laor, A. and Draine, B., ApJ 402, 441
- 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
- Mishchenko, M. et al. 2002, *Scattering, absorption, and emission of light by small particles*, Cambridge University Press
- Mutschke, H. et al. 2004, A&A 423, 983
- Tazaki, R. et al. 2016, ApJ 823, 70
- Tazaki, R. & Tanaka, H. 2018, ApJ 860,79
- 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