DustEM User's Guide

July 6, 2011

DustEM is a numerical tool written in fortran 95 jointly developed by the Institut d'Astrophysique Spatial (IAS, Orsay, France) and the Centre d'Etude Spatiale des Rayonnements (CESR, Toulouse, France) to compute the extinction and the emission of interstellar dust grains heated by photons. The dust emission is calculated in the optically thin limit (no radiative transfer). The code has been designed so that dust properties can easily be changed and mixed and to allow for the inclusion of new grain physics (see **?**). The grain temperature distribution dP/dT is derived using the formalism of **?** and assuming that the grain cooling is fully continuous. To correctly describe the long wavelength emission of dust, the initial algorithm has been revised to cover the low temperature part of dP/dT. The current stable version of DustEM is available at http://www.ias.u-psud.fr/DUSTEM.
After a few words on how to set up DustEM, we detail below the way to control DustEM as well as the files it needs and the one it produces.

1 Getting started

After downloading the tar ball from the DustEM website:

1. unpack the archive:

   ```
   tar xvzf dustemVersion_web.tar.gz
   ```
   or
   ```
   gunzip dustemVersion_web.tar.gz
   tar xvf dustemVersion_web.tar
   ```

   this will generate the dustemVersion_web/ directory that contains the following subdirectories: src/, data/, hcap/, oprop/, out/ and idl_dustem/. The fortran files are in src/, the data files (i. e. input) are in oprop/, hcap/, data/ and output files are in out/. All data files have headers (lines begining with #) that document their content. IDL routines useful to generate or visualize DustEM files can be found in idl_dustem/[1].

2. Go in src/ and open the file DM_constants.f90. Then edit the string data_path as the absolute path to the dustemVersion_web/ dir you just created. If you are running DustEM coupled to the Meudon PDR code, you must also define the string dir_PDR as the absolute path to dustemVersion_web/out/.

3. In src/ again. Open Makefile and set up the standard commands corresponding to your fortran compiler (gfortran, g95 and ifc have been tested), *e.g.*:

   ```
   FC = gfortran
   FFLAGS = -O2 -fno-second-underscore
   ```

4. Type make to compile (possibly preceded by make clean, if a former attempt went wrong). A successful compilation will end with That's it !. The executable is called dustem

[1] Documentation on IDL routines, for instance called rname, is usually obtained by typing rname (tt=rname()) in case of a procedure (function).

5. After running `dustem`, you can check your run by using `show_dustem` in `idl_dustem/`.

2 Input files

The main input file of `DustEM` is `GRAIN.DAT`, located in the directory `data/`. It is the control input file that contains the main parameters and keywords[2] to define a dust model. We describe this file in section **??**. Each dust *population* is defined by its *type* (i. e. material), its total mass abundance and its size distribution to be read in `GRAIN.DAT`. For a required *type*, `tt`, the code will need to read the `oprop/Q_tt.DAT` and `hcap/C_tt.DAT` files that contain the optical properties and heat capacity, respectively. Those files are described in section **??**. `DustEM` can handle an arbitrary number of dust *populations* as soon as `Q_tt.DAT` and `C_tt.DAT` are provided for the corresponding *types*. Other input files non related to the dust properties themselves, like the exciting radiation field are described in section **??**.

2.1 Defining a dust model : `GRAIN.DAT`

We show below a typical example of the `GRAIN.DAT` (as for the diffuse high galactic latitude medium dust of **?**):

```
# DUSTEM: definition of grain populations
#
# run keywords
# G0 scaling factor for radiation field
# grain type, nsize, population keywords, Mdust/MH, rho, amin, amax, alpha/a0 [, at, ac, gamma (ED)] [, au, zeta,
eta (CV)]
# cgs units
#
 none
 1.00
 PAH0_MC10   20 logn-mix   7.80E-04 2.24E+00 3.50E-08 1.20E-07  6.40E-08  1.00E-01
 PAH1_MC10   20 logn-mix   7.80E-04 2.24E+00 3.50E-08 1.20E-07  6.40E-08  1.00E-01
 amCBEx      20 logn       1.65E-04 1.81E+00 6.00E-08 2.00E-06  2.00E-07  3.50E-01
 amCBEx      35 plaw-ed    1.45E-03 1.81E+00 4.00E-07 2.00E-04 -2.80E+00 1.50E-05 1.50E-05 2.00E+00
 aSil        35 plaw-ed    7.80E-03 3.50E+00 4.00E-07 2.00E-04 -3.40E+00 2.00E-05 2.00E-05 2.00E+00
```

After some documentation lines (starting with `#`) comes the *run keywords* line (`none` in the example above) then a line containing a scaling factor (`1.00` in the example above) to be applied to the exciting radiation field (defined in `ISRF.DAT`, see § **??**). The next lines (starting with `PAH0_MC10, PAH1_MC10, amCBEx, amCBEx` and `aSil` in the example above) allow one to define the required dust *populations*. As seen in this example for `amCBEx`, several dust *populations* can make use of the the same grain *type*.
The *run keywords* control the way `DustEM` is executed and the output files it produces. If multiple *run keywords* are used, they must be written on the same line, separated by blanks. The available *run keywords* are

- `none:` if no *run keywords* required,
- `quiet:` sets the verbose mode off at runtime,
- `sdist:` for each size of each dust *population*, `DustEM` will write the size distribution in the file `out/SDIST.RES`,
- `res_a:` the code will produce the spectral energy distribution and extinction cross section output for each size of each dust *population* (see section **??**)

[2] keywords are important to control `DustEM` and can be written in uppercase or lowercase characters.

- `temp:` for each size of each dust *population*, the code will write the temperature distribution in the file `out/TEMP.RES`,

- `pdr:` the code will write a summary of the output in `out/EXTINCTION_DUSTEM.RES` to be used when `DustEM` is coupled to the Meudon PDR code (http://pdr.obspm.fr/PDRcode.html).

The structure for the dust *population* definition lines is the following :

- Column 1: the grain *type* (e. g. `PAH0_MC10`). As already stated, any required *type*, `tt`, must come with `oprop/Q_tt.DAT` and `hcap/C_tt.DAT` files that contain the optical properties and heat capacity, respectively (see section **??**).

- Column 2: `nsize`, the number of size bin for the code to discretize the size distribution.

- Column 3: the *population keywords*. These keywords primarily allow one to load parametrized size distributions (e. g. `logn, plaw, ...`) whose parameters will be given in column 8 and more (see section **??**). Additional *population keywords* (e. g. `mix, beta`) allow one to call optional physical processes included in `DustEM` to be applied to the dust *population* (see section **??**). *Population keywords* can be multiple, then separated by a dash.

- Column 4: the mass abundance of the dust *population* regarding hydrogen nucleus, $M/M_{\rm H}$.

- Column 5: the mass density, ρ, of the *type* required for the dust *population* in $\mathrm{g\,cm^{-3}}$. In any case, the density of the equivalent compact 3D material must be given (i. e. the graphite density for PAHs and the compact material density for porous grains).

- Column 6: the minimum size[3], `a_min` (in cm), to be accounted for in the model for the dust *population* (see section **??**).

- Column 7: the maximum size, `a_max` (in cm), to be accounted for in the model for the dust *population* (see section **??**).

- Column 8 and more : Size distribution parameters (see section **??**).

2.1.1 *Population keywords* : parametrized size distributions

In this section, we describe how to define the size distribution, dn/da (the number of grains whose radius is included in $[a,\, a+da]$), of a given dust *population*. In `DustEM`, the term "size" always refers to the radius a defined as that of an equivalent sphere of same mass $m = \rho\, 4\pi a^3/3$, where ρ is the density of the equivalent compact 3D material.

Parametrized size distributions (e. g. log-normal, power-law) can be used when the corresponding *population keyword* (e. g. `logn, plaw, ...`) is set in `GRAIN.DAT`. All parameters for such distributions are also read from `GRAIN.DAT` (see section **??** above). The parametrized size distribution of the given *population* will be discretized in `nsize` points in the interval [`a_min`, `a_max`] with a constant logarithmic step. Here, we list the different *population keywords* to use for different parametrized size distributions and the parameters that must be provided in `GRAIN.DAT`:

- `logn:` defines a log-normal size distribution with $dn/d\log a \propto \exp(-\log(a/a_0)^2)/\sigma$. The centroid a_0 (in cm) and width σ are given in column 8 and 9.

- `plaw:` defines a power-law size distribution $dn/da \propto a^\alpha$. The index α is given in column 8. Additional keywords can be used when using `plaw`:

[3] In `DustEM`, the term "size" always refers to the radius a defined as that of an equivalent sphere of same mass $m = \rho\, 4\pi a^3/3$.

– `ed:` applies an exponential decay to the power law size distribution multiplying dn/da by the function

$$D(a) = \begin{array}{ll} 1 & a \leq a_t \\ \exp(-\left[(a-a_t)/a_c\right]^{\gamma}) & a > a_t. \end{array} \quad (1)$$

The parameters a_t, a_c (in cm) and γ are then given in column 9, 10, 11, respectively.

– `cv:` applies a curvature term to the power law size distribution multiplying dn/da by the function

$$C(a) = \left[\, 1 + |\zeta| \, (a/a_u)^{\eta} \,\right]^{\mathrm{sgn}(\zeta)} \quad (2)$$

If `ed` is set with `cv` the parameters a_u (in cm), ζ and η are given in column 12, 13, 14, respectively. Otherwise they are given in column 9, 10, 11.

NOTE: The keywords `ed` and `cv` allow to use a size distribution similar to that of Weingartner & Draine (2001).

An other way to define the size distribution of a dust *population* of *type* `tt`, is to give it as tabulated values in an additional input file `data/SIZE_tt.DAT`. In such a case, the *population keyword* `size` must be set. The description of `SIZE_tt.DAT` is given in Appendix **??**.

2.1.2 *Population keywords* : grain physics

Physical processes affecting the properties of grains can be accounted for in `DustEM` through the use of *population keywords* often associated with data files where parameters are passed. Below is the list of such *population keywords*:

- `mix:` apply a size dependent weighting factor when computing the contribution of the dust *population* to the extinction and emission. If the *population* is of *type* `tt`, a file `data/MIX_tt.DAT` must contains a single column of weighting factors, given for each point of the size grid of the *population*. `mix` is very useful when a size dependent mixing of several dust *type* needs to be done.

  In the example `GRAIN.DAT` above, the neutral PAH, `PAH0_MC10`, and charged PAH, `PAH1_MC10`, have the same mass abundance (`7.80E-04`) that is the expected total PAH abundance. Then, they are both weighted through `MIX_PAH0_MC10.DAT` and `MIX_PAH1_MC10.DAT`, according to the PAH ionization fraction that depend on the size. `PAH0_MC10` and `PAH1_MC10` must then share the same size grid and their weighting factors, f_w, must satisfy, $\sum f_w = 1$ for any size.

- `beta:` applies a correction to the $Q_{\rm em}$ (in fact $Q_{\rm abs}$) of a dust *population*, in order to introduce a temperature dependence of the dust emissivity. It is described in Appendix **??**.

2.2 Optical properties and heat capacities files

For a given *type* `tt`, dust properties are provided depending on the size a in data files, namely: `Q_tt.DAT` files for the absorption and scattering efficiencies, $Q(a, \lambda)$, and `C_tt.DAT` files for the heat capacity per unit volume, $C(a, T)$. The Q and C values are provided over the broadest possible size range (usually 0.3 to 10^4 nm). A grain *type* can be used once these Q and C-files exist in the `oprop/` and `hcap/` directories, respectively. For a given model, requested sizes have to fit into those of the above files (i.e. extrapolation is not allowed). When `DustEM` is coupled to radiative transfer codes, $g(a, \lambda)$[4] are also required and must be given in `oprop/G_tt.DAT`. We briefly outline below the format of these files.

[4]The anisotropy factor for scattering defined as $< \cos\theta >$ where θ is the scattering angle.

The Q-files look like:

```
# QABS and QSCA to be used by DUSTEM
#
# nsize (number of sampled sizes in this file)
# sizes (microns)
50
3.0000E-04 3.7104E-04 4.5891E-04 5.6759E-04 7.0200E-04 8.6824E-04 1.0738E-03 1.3281E-03 ...
#
#### QABS ####
1.3461E-03 1.6642E-03 2.0571E-03 2.5419E-03 3.1398E-03 3.8762E-03 4.7819E-03 5.8952E-03 ...
...
#### QSCA ####
5.0179E-07 1.1650E-06 2.6935E-06 6.1884E-06 1.4083E-05 3.1594E-05 6.9380E-05 1.4766E-04 ...
```

The first uncommented line gives the number of sizes defined in the file. The second line gives the sizes (in microns). Then come the $Q_{\rm abs}$ (one column per size) and the $Q_{\rm sca}$ (one column per size). The Q and G-files must all (i. e. for any *type*) be generated with a common wavelength grid which can be found in `oprop/LAMBDA.DAT`. In case of porous grains, the optical efficiencies must be normalized for the size (radius) of the equivalent compact sphere of same mass . Indeed, the sizes are always given as the one of the equivalent compact sphere of same mass (in `GRAIN.DAT`) and the density provided in `GRAIN.DAT` will also be the one of the equivalent compact material. For the G-files, $Q_{\rm abs}$ will be replaced by g (and of course no $Q_{\rm sca}$ will appear).

The C-files look like:

```
# DUSTEM heat capacity
#
# nr of sizes ns
# sizes (microns)
# nr of T-values
# log T(K) log C_1...log C_ns (erg/K/cm3)
50
3.0000E-04 3.7104E-04 4.5891E-04 5.6759E-04 7.0200E-04 8.6824E-04 1.0738E-03 1.3281E-03 ...
30
-1.0000E+00 4.8983E-01 4.8983E-01 4.8983E-01 4.8983E-01 4.8983E-01 4.8983E-01 4.8983E-01 ...
```

The first uncommented line gives the nuber of sizs that are defined. The second line gives the sizes in microns. The third line gives the number of temperatures (in the temperature grid). Then comes the $\log_{10}$ of temperatures (column 1) and the $\log_{10}$ of heat capacities (as many column as defined sizes) in unit of erg/K/cm^3. The same temperature grid (30 points from 0.1 to 5,000 K with a constant log step) is used for all C-files (i. e. for any *type*). This temperature grid is also written in `hcap/TEMP.DAT`.

2.2.1 Pre-defined *types*

We list below the grain *types* for which Q and C files are available in the current `DustEM` distribution:

- **PAH0_MC10, PAH1_MC10:** PAH as defined in **?** for neutral and singly charged PAHs, respectively,
- **PAH0_DL07, PAH1_DL07:** PAH as defined in **?** for neutral and singly charged PAHs, respectively. It takes into account the transition to graphite (Gra_Dr03 *type*) optical prop-

erties for $a > 50\,\text{Å}$,

- **Gra:** Graphite grains. The Q-values have been generated with the Mie method for spherical particles. The refractive index and heat capacity are as described in **??** (optical properties are those of **?**),
- **Gra_Dr03:** Graphite grains. The Q-values have been generated with the Mie method for spherical particles. The refractive index and heat capacity are as in **?** (optical properties are those of **?**),
- **amC-BEx:** Amorphous carbon grains. The Q-values have been generated with the Mie method. The refractive index is the one of **?** and the Q_{abs} have been extrapolated above 0.9 mm (see **?**). The heat capacity is the same as that of graphite.
- **aSil:** Silicate (olivine type) grains. The Q-values have been generated with the Mie method. The refractive index and heat capacity are as in **?** (originally from **?**).
- **PAH0_DBP90:** PAH as defined in **?**
- **VSG_DBP90:** VSG as defined in **?**
- **BG_DBP90:** BG as defined in **?**
- **PAH1_MC08:** PAH described in **?**
- **BG_MC08:** BG as defined in **?**, modified by **?**

3 Template dust models

Several dust models are widely used and constitute reference dust models. Control input files (i. e. `GRAIN.DAT` file) containing parameters to reproduce these models are given in:

- **GRAIN_DBP90.DAT:** for the model of **?**,
- **GRAIN_DL07.DAT:** In this case, the input are as close as possible to the one described in **?**. However, users that wish to get the exact DL07 model should retreive the tabulated values at `http://www.astro.princeton.edu/~draine/` (see Appendix **??** for a comparison),
- **GRAIN_MC08.DAT** for the model of **?**,
- **GRAIN_MC10.DAT:** for the model of **?**,

To be used, these files must be copied (renamed) to `GRAIN.DAT`.

3.1 Astrophysical data

The exciting radiation field used by `DustEM` is read from a file `data/ISRF.DAT`. This file contains 2 columns, the wavelength (microns) and the flux $4\pi I_\nu$ in erg/s/cm^2/Hz. The wavelength range must cover the one of the Q-file. It can be generated with the `create_rfield.pro` routine in `idl_dustem/`. The current distribution includes `ISRF_MATHIS.DAT`, the standard radiation field of **?** for the "solar neighborhood" and `ISRF_CMB.DAT` where the CMB contribution has been added to the Mathis field. These files to be used must be copied (renamed) to `GRAIN.DAT`.

4 Output files

Output files produced by `DustEM` are stored in the directory `out/`. Each file begins with documentation lines (first character is `#`). Any `DustEM` run will produce the 2 following files:

- **SED.RES:** contains the emission per H nucleus $4\pi\nu I_\nu/N_H$ in erg/s/H as a function of wavelength ($N_{\rm H}$ is the H nucleus column density). The first column is the wavelength in μm. Following columns give the SED for each *population* separately, in the same order as read in `GRAIN.DAT`. The last column is the total model emission (i. e. sum over the dust *population*).
- **EXT.RES:** contains the optical cross-section in $cm^2/gram$ for each population. The first column is the wavelength in μm. if `N` *populations* were used, column 2 to `N`+1 are the absorption cross-section for each *population* in the same order as read in `GRAIN.DAT`. Column `N`+2 to `2N`+1 (i. e. the `N` last column) are the scattering cross-section for each *population* in the same order as read in `GRAIN.DAT`.

If the run keywords `temp` or `sdist` are used, `TEMP.DAT` or `SDIST.RES` will be written containing respectively the temperature or the size distribution of the dust populations.

Using the run keyword `res_a` will generate the file `SED_A.RES` which contains the emission per grain in size bin $[a, a+da]$ or $4\pi\nu I_\nu/N_d(a)$ in erg/s/grain ($N_d(a)$ is the column density of grains of size a).

A The SIZE_tt.DAT file use

The data/SIZE_tt.DAT file have the following format (not including the #-lines):

```
# Size distribution of grain species
#
# shape (t-matrix code, 0:spheres, -1:spheroids) eps (b/a, eps>1:oblate, eps<1:  prolate)
# Nbr of bulk materials
# Name of bulks
# Bulk densities in g/cm3
# Mass fractions for each bulk
# Nbr of size bins
# [ a (cm), dloga, a^4*dn/da, rho_eff, fv ]
# fv:  volume fraction of bulks, rho_eff:  volume mean density
#
0 1.00E+00
1
Gra
2.2500E+00
1.0000E+00
46
8.0876E-08 2.0939E-01 7.1858E-02 2.2500E+00 1.0000E+00
....
```

This file can be generated with the IDL routine create_vdist.pro or taken from the dustev ouptut (Guillet *et al.* 2011, in preparation). Size dependent composite grains can be handled this way.

B The $\beta(T)$ correction use

In case the *population keyword* beta is applied to a dust *population* of *type* tt, we have :

$$Q_{em}(a,\nu) = Q_0(a,\nu)\ (\nu/\nu_t)^{\delta(T)\,H(\nu_t/\nu)} \tag{3}$$

where Q_0 is the grain emissivity without the temperature dependence, $\nu_t = c/\lambda_t$ is the frequency threshold and $\delta(T) = \beta(T) - \beta_0$ is the index correction with respect to β_0 some reference value. The threshold function in frequency is $H(x) = 0.5\,(1 + 4\tanh(x)/s)$ where $x = \log(\lambda/\lambda_t)$ and s controls the stiffness of the transition (the width Δx is proportional to s) from 0 to 1: for instance for $s = 1$ $H(x)$ rises from 0 to 1 for x between 0.5 to 1.5 and $\Delta x = 1$. Parameters for this correction are read from a file data/BETA_tt.DAT :

```
# DUSTEM: parameters for BETA(T) behaviour of Qabs
# first DBETA correction to standard index beta0
# then lambda threshold to apply BETA(T)
#
# beta0 a g bmax
# lthresh(microns) lstiff (dlambda=lthresh/lstiff)
# nbetav (nr of beta values if 0 uses formula)
# tbeta betav (if nbetav >= 0)
#
2.11E+00 1.15E+01 -6.60E-01 5.00E+00
3.00E+01 1.00E+00
13
1.00E+00 1.55E+00
...
```