Using the new grain code in Cloudy

Peter van Hoof
Canadian Institute for Theoretical Astrophysics, McLennan Labs, 60 St. George St., Toronto, ON M5S 3H8, Canada
University of Kentucky, Dept. of Physics & Astronomy, 177 CP Building, Lexington, KY 40506–0055, USA

December 6, 2001

1 Introduction

This release of Cloudy contains a new grain model that constitutes a significant upgrade from the original model that was described in Baldwin et al. (1991, ApJ, 374, 580). The new code has been written by Peter van Hoof, except where indicated differently. These are the main new features:

- The code resolves the size distribution of the grains in an arbitrary number of size bins (chosen by the user), and calculates all grain parameters such as temperature, charge, emitted flux, etc., separately for each bin. Since grain opacities depend strongly on the grain radius, they need to be calculated separately for each bin as well.

- For this purpose the code now contains a Mie code (written by P.G. Martin) for spherical grains, which allows the user to calculate grain opacities using either a pre-defined or a user-supplied set of optical constants, an arbitrary size distribution, and an arbitrary number of grain size bins. Note that the code can currently handle no more than `NDUST = 500` grain bins simultaneously. This number can be increased by altering the definition of `NDUST` in `grainvar.h` and recompiling the code.

- The code now fully treats quantum heating of grains using a robust and efficient algorithm (which is a modified version of an original code written by K. Volk), implementing an improved version of the procedure described in Guhathakurta & Draine (1989, ApJ, 345, 230). Combined with resolved size distributions, this will lead to a much more realistic modeling of the grain emission under all circumstances. Quantum heating is turned on automatically for all resolved size distributions[1] and single-sized grains (including PAH's). The user can change the default behavior of the code by including the keyword **qheat** (to enable quantum heating) or **no qheat** (to disable quantum heating) on the **pgrains** command.

- The treatment of the grain physics has been completely overhauled, following the discussion in Weingartner & Draine (2001, ApJS, 134, 263). The new grain model now has an upgraded treatment of the photo-electric effect and collisional processes, and for the first time includes thermionic emissions. The code now uses the new n-charge state model where for each bin the charge distribution is resolved in exactly n charge states, independent of the grain size. The default value for n is 2, but the user can choose any number between 2 and `NCHS = 5` using the **set nchrg** <**n**> command (note that there is no "a" in **nchrg**!). Choosing a higher value will usually give more accurate results at the expense of computing time (which scales roughly linear with n). Using the default $n = 2$ should give sufficient accuracy for all realistic astronomical applications. The maximum value of n can be increased by altering the definition

[1] The actual criterion is that the ratio of the volumes of the largest and smallest grain in each bin is smaller than 100 for quantum heating to be the default. Note that this criterion is very generous and that a much smaller ratio (i.e., a higher number of size bins) will usually be necessary to achieve proper convergence of the emitted spectrum.

of `NCHS` in `grainvar.h` and recompiling the code. A detailed description of the n-charge state model can be found in van Hoof et al. (ASP Conference Series, in press; astro-ph/0107183).

Extensive comparisons in collaboration with Joe Weingartner show that the photo-electric heating rates and collisional cooling rates predicted by Cloudy agree very well with the results from the Weingartner & Draine (2001) model for a wide range of grain sizes (between 5 Å and 0.1 μm), and using various choices for the incident radiation field. A detailed comparison can be found in van Hoof et al. (ASP Conference Series, in press; astro-ph/0107183).

A paper describing the new grain model in detail is currently in preparation (van Hoof et al., to be submitted to ApJ).

2 Using the new grain model

In order to use the new grain model, two steps need to be taken. The first step is to calculate the grain opacities using the **compile grain** command, which is described in detail in Hazy. In order to do this, you start up Cloudy in the data directory containing all the refractive index files (with names ending in "`.rfi`") and size distribution files (with names ending in "`.szd`"). You then type in a single command line, e.g.:

```
compile grain ''silicate.rfi'' ''ism.szd'' 10
```

followed by an extra carriage return. This will instruct the Mie code to calculate opacities using optical constants for astronomical silicate and a Mathis, Rumpl, & Nordsieck standard ISM size distribution. The size distribution will be resolved in 10 bins. This will produce a file `silicate_ism_10.opc` which contains all the opacities. This opacity file is in human readable form and contains many comments to clarify its contents. It also contains a table of the size distribution function $a^4 n(a)$ as a function of the grain radius a (see section 4) for reference. Note that multiple **compile grain** commands can be given in a single Cloudy run. Several pre-compiled opacity files (including the one described here) are already included in the Cloudy distribution.

In the second step, you can use these opacities in a subsequent run of Cloudy with the **pgrains** command described in Hazy. An example could be:

```
set nchrg <n>      (optional, usually not needed)
pgrains ''silicate_ism_10.opc'' +0.100 log      (this will use quantum heating by default)
```

Several **pgrains** commands may be used simultaneously to define mixtures of grains. One can freely mix **set nchrg** and **pgrains** commands, and different grain types may be calculated using a different number of charge states. The **set nchrg** command will only affect **pgrains** commands that come later in the input file. For ease of use, the filenames of certain refractive index, size distribution, and opacity files in the standard Cloudy distribution may be replaced by keywords, as described in Hazy.

Both the refractive index files and the size distribution files may be replaced by user-defined versions, giving the user considerably more freedom to define grain properties compared to the old grain model in Cloudy. The format for each of those files will be defined below.

3 Description of the refractive index files

In order for the Mie code in Cloudy to work, it needs to know the optical properties of the grains under consideration. These have to be defined in a separate file with a name that must end in "`.rfi`". In this section we will describe the format of this file. It is helpful to compare with e.g. the `graphite.rfi` or `silicate.rfi` file in the standard distribution while reading this section.

As is the case with all files connected with the Mie code in Cloudy, the user has the freedom to add comments to the file provided they start with a pound sign (#). These comments may either occupy an entire line (in which case the pound sign has to be in the first column), or be appended to some input value. Comments have been liberally added to the refractive index files that come with the standard Cloudy distribution in an effort to make them self-explanatory. All refractive index files start with a magic number for version control. This number should simply be copied from the files in the standard distribution. Next comes the chemical formula of the grain material, for graphite this would simply be "`C`"; for a certain type of silicate this could be "`Mg0.6Fe0.4SiO3`" indicating $Mg_{0.6}Fe_{0.4}SiO_3$. Note that the formula is case sensitive. For simplicity we will call this elementary building block the grain molecule, even though this term is not always appropriate. The next line gives the molecular weight in amu. This number should normally be set to zero, in which case Cloudy will calculate the atomic weight using the chemical formula and the atomic weights stored in the code. The next two lines in the refractive index file define the default abundance of the grain molecule. The first number gives the maximum number density $A_{\rm max}$ of the grain molecule (relative to hydrogen = 1) that can be formed, assuming it completely depletes at least one of the constituting atoms from the gas phase. Let us assume that the initial abundances in the gas phase (i.e., abundances *before* grains were formed) were $A(X)$. Then , for the silicate example above, $A_{\rm max}$ should be MIN($A({\rm Mg})/0.6$, $A({\rm Fe})/0.4$, $A({\rm Si})$, $A({\rm O})/3$). The second number gives the fraction $A_{\rm eff}$ of the maximum amount that is actually formed (i.e., the efficiency of the process), and should be a number between 0 and 1. The default abundance of the grain molecule is then given by the product of these two numbers: $A_{\rm eff}A_{\rm max}$. The actual grain abundance used in the Cloudy modeling can be set with the **pgrains** and the **metals** command (see Hazy for details). This essentially defines an additional multiplier $A_{\rm rel}$ which may be either smaller or larger than 1, and may depend on position r. The actual grain molecule abundance used in the Cloudy model is then given by $A_{\rm rel}(r)A_{\rm eff}A_{\rm max}$. For the silicate example above, the number density of iron locked up in these grains would be given by $0.4A_{\rm rel}(r)A_{\rm eff}A_{\rm max}$ (relative to hydrogen = 1), or $0.4A_{\rm rel}(r)A_{\rm eff}A_{\rm max}n_{\rm H}(r)$ (in atoms/cm^3, $n_{\rm H}$ is the hydrogen number density as defined by the **hden** command). The next line in the refractive index file gives the specific density of the grain material in g/cm^3. For the quantum heating routine, the code needs to know the enthalpy of the grain as a function of temperature. Currently two choices are hardwired into the code. The next line indicates which of these has to be used. Entering 1 would be appropriate for carbonaceous materials, and 2 for silicates. The next two lines give the work function and the bandgap between the valence and conduction band in Rydberg respectively. For (semi-) conductors the bandgap should be set to zero, for insulators such as silicates, a non-zero value should be used. The next line gives the efficiency of thermionic emissions. This parameter is usually unknown for materials of astrophysical interest, and using 0.5 should be a reasonably safe guess. Next comes the sublimation temperature in Kelvin.

The remaining lines in the refractive index file define the optical constants. First the user has to define how the complex refractive index n is written up; the following choices are supported: 1 – Re(n^2) Im(n^2) (the dielectric function), 2 – Re($n-1$) Im(n), 3 – Re(n) Im(n). The next line gives the number of principal axes N_a for the grain crystal and should be a number between 1 and 3. For amorphous materials one should always choose 1 axis. For crystalline materials the number may be 2 or 3. Next comes a line with N_a numbers giving the relative weights for each of the axes. These numbers will be used to average the opacities over each of the axes in crystalline materials. For materials with only one axis, this number is obviously redundant, and a single 1 should be entered. For graphite it is appropriate to enter "1 2" indicating that the first axis will have relative weight 1/3 and the second 2/3 (i.e., the relative weights are 1:2). Next come N_a chunks of data defining

the optical constants for each axis. Each chunk starts with a line giving the number of data points N_d for that axis, followed by N_d lines containing 3 numbers: the wavelength in micron, and the real and imaginary part of the complex number defined above. Note that the wavelengths may be either monotonically increasing or decreasing, and that the number of data points or the wavelength grid may be different for each axis. Setting N_d to zero will force the code to generate opacities for grey grains for which no refractive index data exist. Note that this format may be changed slightly in the near future to accommodate PAH's in the new grain model.

4 Description of the size distribution files

In order for the Mie code in Cloudy to work, it needs to know the size distribution of the grains under consideration. This distribution has to be defined in a separate file with a name that must end in "`.szd`". In this section we will describe the format of this file. If we denote the number of grains $n_{\rm g}da$ with radii between a and $a+da$ as $n_{\rm g}da = n(a)da$, the purpose of the size distribution file is to define $n(a)$, or alternatively $a^4 n(a)$ which is more commonly used.

As is the case with all files connected with the Mie code in Cloudy, the user has the freedom to add comments to the file provided they start with a pound sign (#). These comments may either occupy an entire line (in which case the pound sign has to be in the first column), or be appended to some input value. Comments have been liberally added to the size distribution files that come with the standard Cloudy distribution in an effort to make them self-explanatory. All size distribution files start with a magic number for version control. This number should simply be copied from the files in the standard distribution. The next line should contain a keyword indicating which type of size distribution will be entered. The following choices are currently supported: **ssize** - a single sized grain, **power** - a simple power law, **exp1, exp2, exp3** - power laws with an exponential cutoff, **normal, lognormal** - a Gaussian distribution in a or $\ln(a)$, **table** - an arbitrary size distribution supplied as a table. This keyword is case insensitive. The rest of the file contains the parameters needed to fully define each of those choices. We will now describe these choices in more detail. It should be noted that at this stage the absolute normalization of the size distribution is irrelevant; that will be defined in the refractive index file. Each parameter mentioned below should entered on a separate line, unless indicated otherwise. All size parameters should be entered in micron.

4.1 ssize

In this case the size distribution is given by a simple delta function:

$$n(a) \propto \delta(a - a_0)$$

The only parameter that needs to be supplied is the radius of the grain a_0.

4.2 power

In this case the size distribution is given by a simple power law:

$$n(a) \propto a^{\alpha} \;\; a_0 \leq a \leq a_1$$

Hence this distribution needs three parameters, which need to be supplied in the order a_0, a_1, α.

4.3 exp1, exp2, exp3

In this case the size distribution is given by a power law with a first-, second-, or third-order exponential cutoff:

$$n(a) \propto a^{\alpha} F(a;\beta) C_l(a;a_l,\sigma_l) C_u(a;a_u,\sigma_u) \;\; a_0 \leq a \leq a_1.$$

The function F is included to give extra curvature in the power-law region, the functions C_l and C_u define the cutoff of the distribution below a_l and above a_u. These functions are defined as follows:

$$F(a;\beta) = \begin{cases} (1-\beta a)^{-1} & \text{if } \beta < 0 \\ 1 & \text{if } \beta = 0 \\ (1+\beta a) & \text{if } \beta > 0 \end{cases}$$

$$C_l(a;a_l,\sigma_l) = \begin{cases} \exp\left(\frac{a-a_l}{\sigma_l}\right)^n & \text{if } a < a_l \\ 1 & \text{if } a \geq a_l \end{cases}$$

$$C_u(a;a_u,\sigma_u) = \begin{cases} 1 & \text{if } a \leq a_u \\ \exp\left(\frac{a_u-a}{\sigma_u}\right)^n & \text{if } a > a_u \end{cases}$$

The values of σ_l or σ_u may be set to zero, in which case a straight cutoff in the size distribution will be used. Note that when β, σ_l, and σ_u are all set to zero, this size distribution degenerates to the simple power law discussed above. The parameters need to be supplied in the following order: a_l, a_u, α, β, σ_l, σ_u, a_0, a_1. The value of n is determined by the keyword used: $n = 1$ for **exp1**, etc. Note that this size distribution extends infinitely beyond a_l and a_u, so additional cutoffs at a_0 and a_1 are needed. Either of these values may be set to zero, in which case Cloudy will calculate a safe default value such that only a negligible amount of mass is contained in the tail beyond that limit.

4.4 normal

In this case the size distribution is given by a Gaussian distribution in a:

$$n(a) \propto \frac{1}{a}\exp\left(-\frac{1}{2}\left[\frac{a-a_c}{\sigma}\right]^2\right) \quad a_0 \leq a \leq a_1$$

The parameters for this distribution need to be supplied in the order: a_c, σ, a_0, a_1. As was discussed in the previous section, the values of a_0 or a_1 may be set to zero in which case Cloudy will calculate a safe default.

4.5 lognormal

This case is completely analogous to the **normal** case discussed above, except that the Gaussian distribution is now given by:

$$n(a) \propto \frac{1}{a}\exp\left(-\frac{1}{2}\left[\frac{\ln\{a/a_c\}}{\sigma}\right]^2\right) \quad a_0 \leq a \leq a_1$$

4.6 table

This option allows the user to define an arbitrary size distribution in the form of a table of $a^4 n(a)$ as a function of a. First values for the lower and upper size limit a_0 and a_1 should be supplied. These values need not coincide with the lower and upper size limit of the table, although the range of the table should of course be at least that large. Next the number of data pairs n in the table should be supplied, followed by n lines each containing two numbers: a (in micron) and $a^4 n(a)$ (in arbitrary units). Note that the values for a in the table must be strictly monotonically increasing.