Using the new grain code in Cloudy

Peter van Hoof
Queen's University of Belfast, APS Division, Physics Dept., Belfast, BT7 1NN, Northern Ireland

January 28, 2003

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 distribution, 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 Cloudy now contains a Mie code for spherical grains (written by P.G. Martin, based on a code by Hansen & Travis 1974, Space Sci. Rev., 16, 527). This code 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 comprehensively upgraded version of a code originally 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, but excluding grey grains for which quantum heating is never used). 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 **grains** command. Note that quantum heating usually only influences the energy distribution of the emitted spectrum, nothing else[2]. So switching off quantum heating can be a good idea to speed up the modeling, provided the correct shape of the emitted spectrum has no influence on the results the user is interested in.

- 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

[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.

[2] Quantum heating can influence the emission line spectrum as well if continuum pumping of the lines by the short wavelength end of the dust spectrum is important.

code now uses the new n-charge state model where for each bin the charge distribution is resolved in exactly n discrete 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 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, Vol. 247, p. 363; 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 discussion of this comparison can be found in van Hoof et al. (ASP Conference Series, Vol. 247, p. 363; 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. In the first step, the grain opacities are calculated using the **compile grain** command. In the second step these opacities can be used with the **grains** command to create the actual photo-ionization model. The Cloudy distribution comes with a number of precompiled opacity files which cover a number of standard combinations of grain materials and size distributions. If these are sufficient for your needs, you can skip the skip the first step, and use these opacities directly in the **grains** command. If you wish to use different grain materials and/or size distributions, you will have to compile the grain opacities first with 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, mixed medium and size distribution files (files with names ending in "`.rfi`", "`.mix`", or "`.szd`", resp.). 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 for reference. The format of the table is such that it could be used directly to define a size distribution table (see section 5.6). Note that only a single **compile grain** command can be given in a single Cloudy run. Note that the opacity file produced in the example above is already included in the Cloudy distribution!

In the previous example a refractive index file was used to define the optical properties of a pure grain material (e.g., astronomical silicate or graphite). A second example could be:

```
compile grain ''fluffy-sil.mix'' ''ism.szd'' 10
```

(note that the mixed medium file `fluffy-sil.mix` is not included in the Cloudy distribution, the text is shown in Table 9). Here the mixed medium file is used instead of a refractive index file to define a mixture of two or more grain materials (possibly including vacuum for defining fluffy grains). In this case the optical properties of the constituting materials are combined with a mixing law (also called effective medium theory or EMT) to create a set of optical constants that represents the mixture and can be fed into the Mie theory. Hence the refractive index and mixed medium files play similar roles in that both define a set of optical constants to be used by the Mie code. They differ in the fact that refractive index files are used to define pure materials, while mixed medium files define impure materials. As a consequence the internal format of both files is very different and will be described in more detail in Sections 3 and 4. The opacities generated from both types of input files are fully compatible though and can be used in exactly the same way, as shown below.

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

```
set nchrg <n>     (optional, usually not needed)
grains ''silicate_ism_10.opc'' +0.100 log    (this will use quantum heating by default)
grains ''fluffy-sil_ism_10.opc'' no qheat    (this will not use quantum heating)
```

Several **grains** commands may be used simultaneously to define mixtures of grains. One can freely mix **set nchrg** and **grains** commands, and different grain types may be calculated using a different number of charge states. The **set nchrg** command will only affect **grains** 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.

The refractive index, mixed medium, and 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 in Sections 3, 4, and 5.

3 Description of the refractive index files[3]

In order for the Mie code in Cloudy to work, it needs to know the optical properties of the grains under consideration. If the grain consists of a single, pure material, 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. This document pertains to refractive index files with magic number `1030103`. Mixtures of grain materials can also be defined using mixed medium files which are described in Section 4.

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 I will call this elementary building block the grain molecule, even though this term is not always appropriate. The molecular weight will be calculated by Cloudy using the chemical formula. The next two lines in the refractive

[3]Note that the term "refractive index file" is used rather loosely here. It also pertains to materials for which no refractive index data in the strict sense exist, such as grey grains and PAH's.

Table 1: The definition of each of the material types hardwired into Cloudy. The code in column 1 needs to be entered in the refractive index file. Each of the entries in columns 3 and higher will be explained in the table indicated in the header of that column.

code	mnemonic	Table 2	Table 3	Table 4	Table 5	Table 6	Table 7	Table 8
1	MAT_CAR	ENTH_CAR	ZMIN_CAR	POT_CAR	IAL_CAR	PE_CAR	STRG_CAR	H2_CAR
2	MAT_SIL	ENTH_SIL	ZMIN_SIL	POT_SIL	IAL_SIL	PE_SIL	STRG_SIL	H2_SIL
3	MAT_PAH	ENTH_PAH	ZMIN_CAR	POT_CAR	IAL_CAR	PE_CAR	STRG_CAR	H2_CAR
4	MAT_CAR2	ENTH_CAR2	ZMIN_CAR	POT_CAR	IAL_CAR	PE_CAR	STRG_CAR	H2_CAR
5	MAT_SIL2	ENTH_SIL2	ZMIN_SIL	POT_SIL	IAL_SIL	PE_SIL	STRG_SIL	H2_SIL
6	MAT_PAH2	ENTH_PAH2	ZMIN_CAR	POT_CAR	IAL_CAR	PE_CAR	STRG_CAR	H2_CAR

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 **grains** 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. The grain code needs to know the material type in order to determine certain grain properties that are currently hardwired into the code. Examples would be the grain enthalpy as a function of temperature and the ro-vibrational distribution of H_2 formed on the grain surface. The next line indicates which material type to use. Currently six choices are available. They are outlined in Table 1. The next two lines give the work function and the bandgap between the valence and conduction band in Rydberg respectively. For 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. Then comes a line with a keyword which identifies what type of refractive index file is being read. It determines what the remainder of the file will look like. Allowed values are **rfi_tbl, opc_tbl, grey, pah1**. In the **grey** and **pah1** cases, no further data is needed and the refractive index file ends there (i.e., the information needed to calculate the opacities is hardwired in the code).

3.1 rfi_tbl

This format is used to define the refractive index as a function of wavelength, which will then be used by the Mie code to generate the opacities when combined with a size distribution file.

The remaining lines in the refractive index file define the optical constants. First the user has to enter a code to define how the complex refractive index n is written up; the following choices are supported:

1. – supply Re(n^2), Im(n^2) (the dielectric function),
2. – supply Re($n-1$), Im(n),

Table 2: The various choices for the enthalpy function hardwired in Cloudy.

mnemonic	type	reference
ENTH_CAR	graphite	Guhathakurta & Draine, 1989, ApJ, 345, 230, Eq. 3.3
ENTH_SIL	silicate	Guhathakurta & Draine, 1989, ApJ, 345, 230, Eq. 3.4
ENTH_PAH	PAH	Dwek et al., 1997, ApJ, 475, 565, Eq. A4
ENTH_CAR2	graphite	Draine & Li, 2001, ApJ, 551, 807, Eq. 9
ENTH_SIL2	silicate	Draine & Li, 2001, ApJ, 551, 807, Eq. 11
ENTH_PAH2	PAH	Draine & Li, 2001, ApJ, 551, 807, Eq. 33

Table 3: The various choices for the minimum charge hardwired in Cloudy.

mnemonic	type	reference
ZMIN_CAR	graphite	Weingartner & Draine, 2001, ApJS, 134, 263, Eq. 23a+24
ZMIN_SIL	silicate	Weingartner & Draine, 2001, ApJS, 134, 263, Eq. 23b+24

Table 4: The various expressions for the grain potential hardwired in Cloudy.

mnemonic	type	reference
POT_CAR	graphite	Weingartner & Draine, 2001, ApJS, 134, 263, Eq. 4
POT_SIL	silicate	Weingartner & Draine, 2001, ApJS, 134, 263, Eq. 5

Table 5: The various choices for the inverse attenuation length. Note that these choices will *only* be used if no refractive index data is contained in the refractive index file.

mnemonic	type	description
IAL_CAR	graphite	use `graphite.rfi` to calculate the inverse attenuation length
IAL_SIL	silicate	use `silicate.rfi` to calculate the inverse attenuation length

Table 6: The various expressions for the photoelectric yield hardwired in Cloudy.

mnemonic	type	reference
PE_CAR	graphite	Weingartner & Draine, 2001, ApJS, 134, 263, Eq. 16
PE_SIL	silicate	Weingartner & Draine, 2001, ApJS, 134, 263, Eq. 17

Table 7: The various choices for splitting up the grain emissions.

mnemonic	type	description
STRG_CAR	graphite	store emitted spectrum as graphitic emissions.
STRG_SIL	silicate	store emitted spectrum as silicate emissions.

Table 8: The expressions for the ro-vibrational distribution of H_2 formed on various grain surfaces.

mnemonic	type	reference
H2_ICE	ice mantle	Takahashi, J., Uehara, H., 2001, ApJ, 561, 843, Table 2
H2_CAR	graphite	Takahashi, J., Uehara, H., 2001, ApJ, 561, 843, Table 2
H2_SIL	silicate	Takahashi, J., Uehara, H., 2001, ApJ, 561, 843, Table 2

3. – supply 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. The grid of wavelengths need not coincide with the Cloudy grid, nor does it need to cover the entire range of energies used in Cloudy. Logarithmic interpolation or extrapolation will be performed where needed. *It is the users responsibility to ensure that each wavelength grid contains sufficient points to make this process meaningful; only minimal checks will be performed by Cloudy.*

3.2 opc_tbl

This format can be used to directly define the opacities as a function of photon energy. It is mainly useful to define alternative prescriptions for PAH's. Note that the refractive index file created this way still needs to be combined with a size distribution file in the usual way with the **compile grains** command. This step will not alter the opacities themselves, but is necessary to compute certain quantities that are related to the size distribution which are not defined in the refractive index file itself. It is not allowed to resolve the size distribution when using an opacity table. If you want to define a resolved size distribution, you will have to supply an opacity table and corresponding size distribution file for each size bin separately.

The remaining lines in the refractive index file define the opacities. The first line contains the number of data values N_v supplied on each line in the table. Allowed values are:

1. – supply $\sigma_{\rm abs}$ only (in cm^2/H),
2. – supply $\sigma_{\rm abs}$, $\sigma_{\rm sct} \times (1-g)$ (in cm^2/H),
3. – supply $\sigma_{\rm abs}$, $\sigma_{\rm sct}$ (in cm^2/H), and $(1-g)$ separately.
4. – supply $\sigma_{\rm abs}$, $\sigma_{\rm sct}$ (in cm^2/H), $(1-g)$, and the inverse attenuation length (in cm^{-1}).

Here $\sigma_{\rm abs}$ and $\sigma_{\rm sct}$ are the absorption and scattering cross sections, and g is called the asymmetry factor or phase function. The latter is needed since the radiative transfer in Cloudy assumes that photons that are scattered in a forward direction (at an angle of less than 90°) are not lost from the beam (i.e., they still make it out of the cloud). The asymmetry factor g gives the average value of the cosine of the scattering angle, so that $(1-g)$ is an approximate correction factor for the anisotropy of the scattering by grains. If no scattering cross sections are supplied, $\sigma_{\rm sct} \times (1-g)$ will default to 10% of the absorption cross section. If no asymmetry factor is supplied, g will default to zero. If no inverse attenuation lengths are supplied, they will be derived from the material type supplied previously in the file. On the next line is a keyword with two allowed values: **log** or **linear**, defining whether the opacity table will contain logarithmic or linear data values. The next line gives the number of data points N_d, followed by N_d lines of actual data. Each line of the table should contain $N_v + 1$ numbers, starting with the photon energy in Rydberg, followed by the absorption cross section, the scattering cross section, the factor $(1-g)$, and the inverse attenuation length (the last three values may be omitted, depending on the value of N_v). The photon energies must either be strictly monotonically increasing or decreasing. The grid of

Table 9: Example of a mixed medium file for fluffy silicate (`fluffy-sil.mix`).

```
# simplistic version of fluffy silicate; for test purposes only!
4030103  # magic number for version control
0.61     # default depletion
2        # number of separate materials in grain
#
  50  "vacuum.rfi"
  50  "silicate.rfi"
#
fa00     # which EMT should be used?
```

photon energies need not coincide with the Cloudy grid, nor does it need to cover the entire range of energies. Logarithmic interpolation or extrapolation will be performed where needed. *It is the users responsibility to ensure that the frequency grid contains sufficient points to make this process meaningful; only minimal checks will be performed by Cloudy.*

4 Description of the mixed medium files

In order to define the optical properties of a grain consisting of a mixture of materials[4], a separate file with a name that must end in "`.mix`" has to be used. In this section we will describe the format of this file. An example of a mixed medium file is shown in Table 9. This document pertains to mixed medium files with magic number `4030103`.

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. All mixed medium files start with a magic number for version control, as shown in Table 9. The chemical formula and the abundance at maximum depletion of the mixed material will be derived by Cloudy using the information supplied in the refractive index files of the constituting materials. However, the default depletion cannot be calculated and needs to be supplied by the user on the next line. This number is defined as a fraction of the maximum depletion, and should be a number between 0 and 1 (see also the discussion in Section 3). When the opacities are calculated, the code will print the chemical formula and the abundances (both at maximum and default depletion), so that the user can check whether the default abundance is correct. The next line gives the number n of separate materials in the grain (n should be at least 2; it is allowed to specify the same material several times), followed by n lines giving information about the materials. Each of these lines should give the relative fraction of the volume occupied by that particular material, followed by the name of the refractive index file between double quotes. Note that the refractive index files have to be of type **rfi_tbl** since mixing laws need optical constants as input and cannot work with opacities. The normalization of the relative volumes can be on an arbitrary scale. The last line in the file should be a keyword identifying which mixing law is to be used. Allowed values are listed in Table 10. In the case of Farafonov (2000), the mixing law assumes that the grain consists of concentric layers of material. It is assumed that the first material supplied in the mixed medium file identifies the innermost layer, and the following lines identify the subsequent layers towards the outer edge of the grain. The outermost layer will determine the material type as defined in Table 1.

[4] In this section the term material can also mean vacuum so that the user can define fluffy grains by mixing one or more materials with vacuum.

Table 10: Allowed choices for the mixing law.

mnemonic	reference
FA00	Voshchinnikov N.V., Mathis J.S., 1999, ApJ, 526, 257 Farafonov V.G., 2000, Optics & Spectroscopy, 88, 441

5 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 I will describe the format of this file. This document pertains to size distribution files with magic number `2010403`.

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 $a^4n(a)$ 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. I 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 by the default grain abundance. Each parameter mentioned below should be entered on a separate line, unless indicated otherwise. All size parameters should be entered in micron.

5.1 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.

5.2 power

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

$$n(a) \propto a^\alpha \;\; a_0 \le a \le a_1$$

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

5.3 exp1, exp2, exp3

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

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

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.

5.4 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.

5.5 lognormal

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

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

5.6 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 be at least that large (no extrapolation will be performed). 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.