

Grains and Stellar Atmospheres in Cloudy

Peter van Hoof
Royal Observatory of Belgium

Cloudy Workshop
Weihai, 2016-06-22

Outline

- The physical effects of grains on their environment
- Interstellar extinction
- Grain size distributions and the photoelectric effect
- Stochastic heating of grains
- Grain formation and destruction
- Spinning grains
- Grains in Cloudy
- Using stellar atmosphere grids in Cloudy
- Running Cloudy in parallel

Physical effects (1)

Grains play an important role in the physics of ionized gas, as well as in the PDR. The main micro-physical processes involving grains are the following:

- Extinction: grains will absorb radiation. Especially in high metallicity environments this can have a substantial influence on the radiative transfer.
- Photoelectric heating: when energetic photons are absorbed, the grain may be ionized (photoelectric effect) and part of the photon energy is carried away by the electron, heating the gas.
- Collisional heating / cooling: when gas particles collide with the grains, they can lose or gain energy, depending on whether the grains are cooler or hotter than the gas.
- Charge exchange: when an ion and a grain collide, the ion may be neutralized by the grain. In rare cases the electron may also move in the opposite direction. This process is very important in fully molecular regions where ions and electrons are rare.

Physical effects (2)

- Grain surface reactions: grains may act as catalysts for molecular reactions (e.g. the formation of H_2). When a hydrogen atom collides with a grain, it can bond to the surface (physisorption, chemisorption) and then hop around from one bonding site to another. That way two bonded H atoms can come together and form H_2 (Langmuir-Hinshelwood mechanism). Alternatively, one bonded atom can react directly with another H atom from the gas (Eley-Rideal mechanism). These mechanisms are much faster than gas phase reactions, but require T_g to be in a favorable range. Other reactions are in the process of being added to Cloudy.
- Molecule freeze-out: grains can act as seeds for molecular freeze-out, thus removing the molecules from the gas phase and impeding reactions with them. The ice mantles can also change the optical properties of the grains, thus altering the radiative transfer in these regions (currently not treated in Cloudy). This only happens in very cold, fully molecular gas ($T_g < 20$ K).

Physical effects (3)

- Thermal emission from grains can pump vibrational transitions in molecules. For most molecules this is currently not implemented in Cloudy since the molecular data for the vibrationally excited levels are missing.
- Grains with a dipole moment will emit anomalous radio emission when they are spinning. The emission will typically peak around 30 GHz and is of great interest for CMB experiments. The emitted power depends very strongly on grain size so that only PAHs are relevant. This is currently in the process of being added to Cloudy.

Interstellar extinction (1)

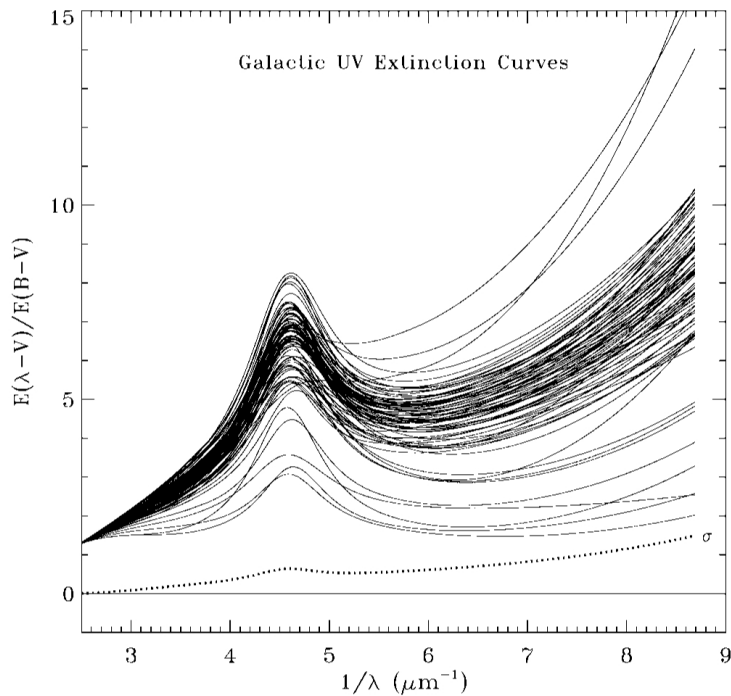


FIG. 2.—Examples of 80 Galactic UV extinction curves derived from *IUE* satellite observations. Analytical fits to the curves are shown, based on the work of Fitzpatrick & Massa (1990). The curves are taken from the Fitzpatrick & Massa catalog, with the addition of the lines of sight toward HD 210121 from Welty & Fowler (1992) and HD 62542 from Cardelli & Savage (1988). This figure demonstrates the enormous range of properties exhibited by UV extinction in the Milky Way. The dotted line, labeled “ σ ,” shows the standard deviation of the sample scaled to the value $\sigma(1500 \text{ \AA}) = 0.74$, as derived from *ANS* satellite data (see § 3.1).

Grains are responsible for interstellar extinction. The properties of interstellar extinction in the UV vary greatly for different sight lines, but are roughly the same in the optical regime (just visible at the left edge of the plot). Interstellar extinction is usually parametrized by a single parameter R_V (the total to selective extinction ratio) ranging from roughly 2.75 to 5.5, but this is clearly already a simplification... These differences are caused by different size distributions and compositions of the grains along the line of sight.

Note that extinction curves in external galaxies can look quite different from the Milky Way (e.g LMC and SMC).

Interstellar extinction (2)

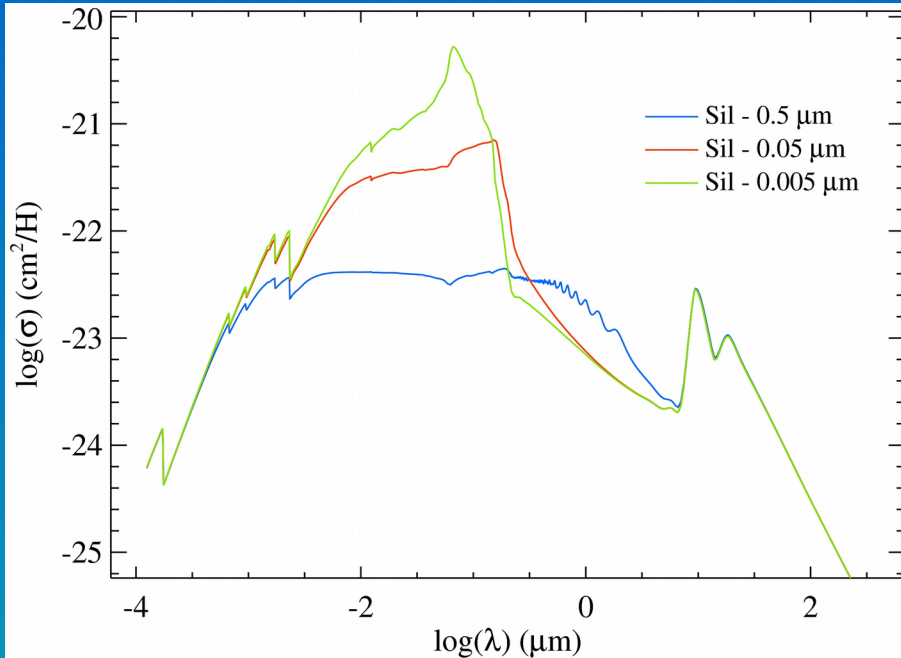


Image credit: P. van Hoof, unpublished work.

Interstellar grains are assumed to consist of a mix of astronomical silicate, graphite and/or amorphous carbon and PAHs (polycyclic aromatic hydrocarbons). The graphite and/or the PAHs are responsible for the strong extinction peak near 220 nm that we saw in the previous plot.

The strong difference in extinction for different R_V values can be understood if we look at this plot of the opacity of astronomical silicate for different grain sizes.

If you want to compare an observed spectrum with a Cloudy prediction, you need to correct for the interstellar extinction first. Disentangling interstellar and circumstellar extinction is usually not possible, so you correct for both. You then compare to the intrinsic spectrum which has no internal extinction effects.

Grain size distributions

Several papers have attempted to determine the size distribution of the grains in the ISM:

Mathis, Rumpl & Nordsieck (1977) ApJ 217, 425 (often called the MRN size distribution): appropriate for the $R_V = 3.1$ extinction curve (called “ism” in Cloudy).

Baldwin et al. (1991) ApJ 374, 580: truncated version of MRN, appropriate for the $R_V = 5.5$ extinction curve (called “orion” in Cloudy).

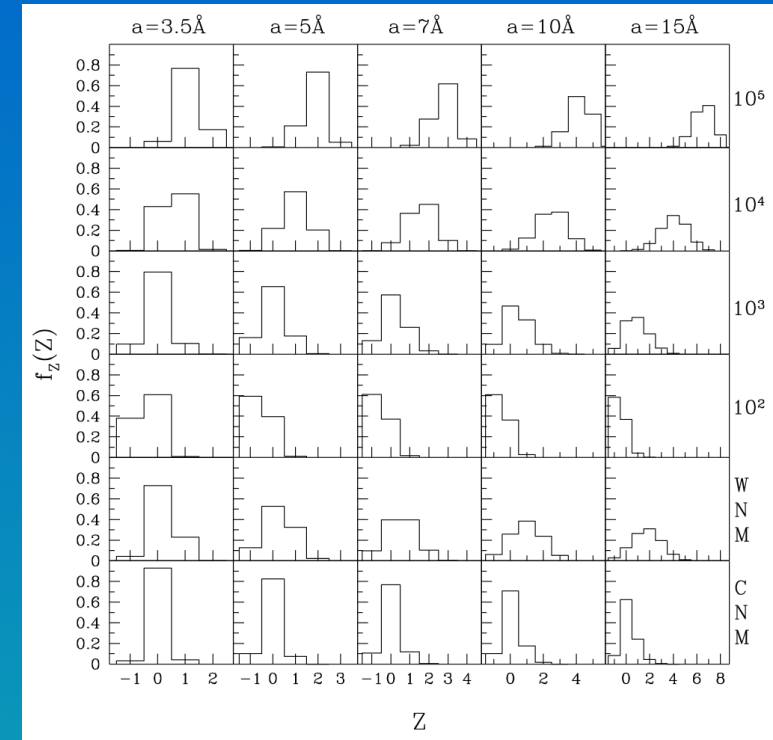
Kim, Martin & Hendry (1994) ApJ 422, 164: give size distributions both for $R_V = 3.1$ and $R_V = 5.3$.

Weingartner & Draine (2001) ApJ 548, 296: give a range of size distributions for $R_V = 3.1$ and $R_V = 5.5$ (with varying PAH content) as well as the LMC and SMC. Updates in Li & Draine (2001) and Draine & Li (2007)

Abel et al. (2008) ApJ, 686, 1125: PAH size distribution (called “ab08” in Cloudy).

Grain Charge

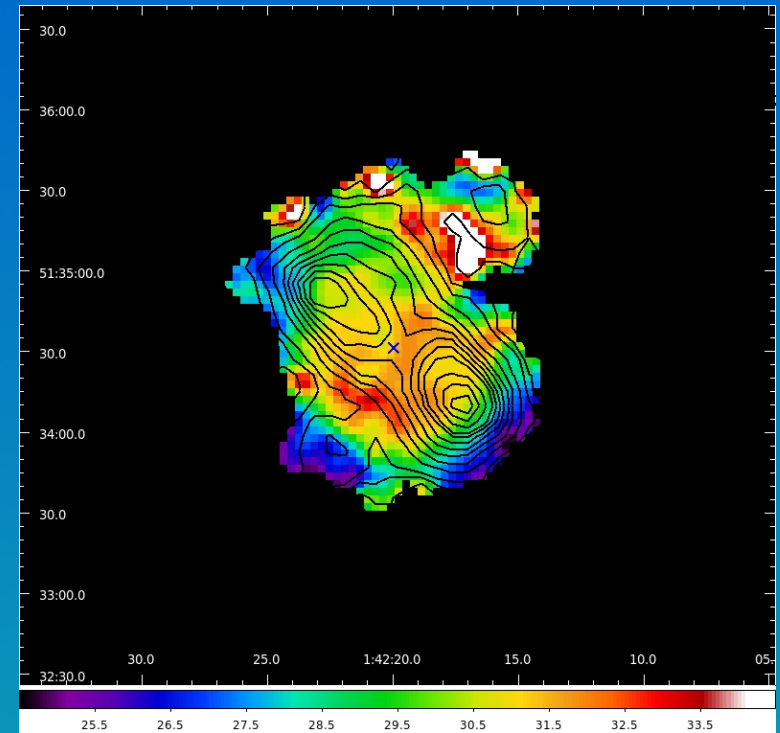
- Solving the grain parameters is iterative. First the grain charge is determined, then the grain temperature.
- The grain charge is determined by a balance of several processes. Most important are the PE effect (incl. Auger cascades), and the capture of free electrons. Charge exchange with ions and thermionic emissions are also taken into account.
- The code uses the hybrid grain charge model (van Hoof+ 2004) which limits the number of charge states to a fixed number (default 2) even for large grains. Use a higher number of charge states for better precision (e.g., set nchrg 4).
- The grain charge is determined self-consistently with the electron density and the ionization balance in the gas.



Weingartner & Draine (2001)

Grain Temperature

- The grain temperature is determined by a balance of heating and cooling processes. The most important heating process is the absorption of ambient photons, the most important cooling process is thermal radiation.
- This thermal emission can pump molecular transitions.
- The photon heating is calculated by integrating over the radiation field taking into account photoelectric losses. Hence the integral depends on grain charge.
- Collisions can also heat or cool the grain depending on the temperature of the gas and the grain. Because of the latter an iterative procedure is needed.



Dust temperature map of NGC 650 observed with Herschel (van Hoof+ 2014)

The PE effect

When UV radiation removes a photon from the grain, part of the photon energy will be carried by the electron as kinetic energy which heats the gas. This is the photoelectric effect. It is an important heating source in the ionized region and the PDR.

The magnitude of the effect depends strongly on the size distribution: small grains have a larger surface area per unit mass and contribute more.

The models on the left all have the same dust mass, only the size distribution is different! The effect is clearly visible.

Unfortunately, the grain size distribution is often poorly known.

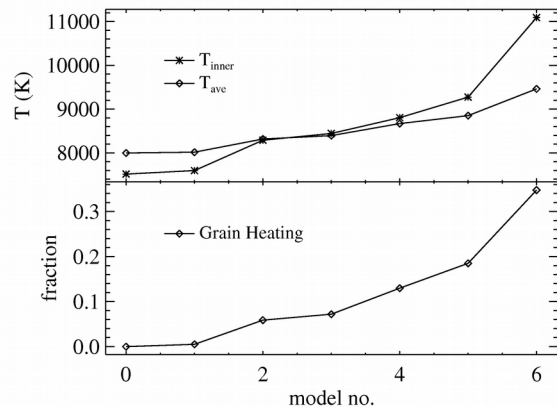
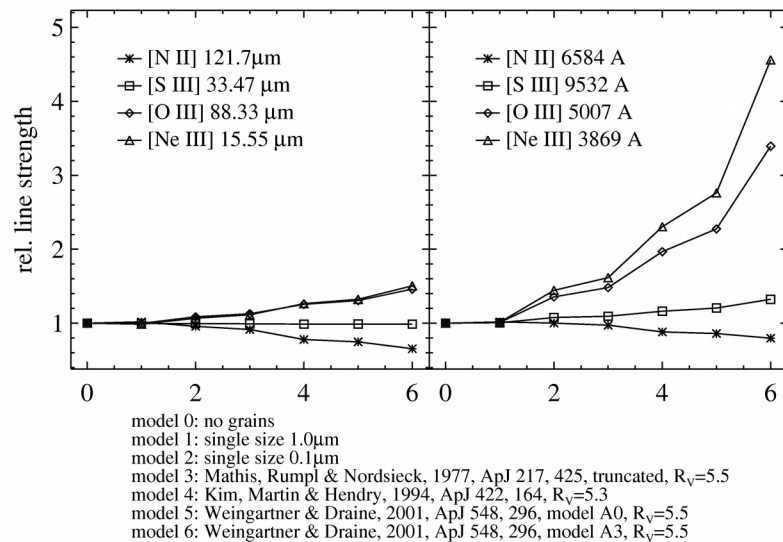


Figure 1. Paris H II region models. In the top left panel we show the relative line strengths for selected infrared fine-structure lines. These are expected to be mostly insensitive to electron temperature and therefore show the difference in the overall ionization structure. The line strengths are normalized to the line strength in the dust-free model. In the top-right panel we show optical/UV forbidden lines of the same species. In the bottom panels we show the electron temperature at the inner edge, as well as averaged over the ionized region, and the fraction of the total gas heating that is due to the photo-electric effect.

Stochastic grain heating (1)

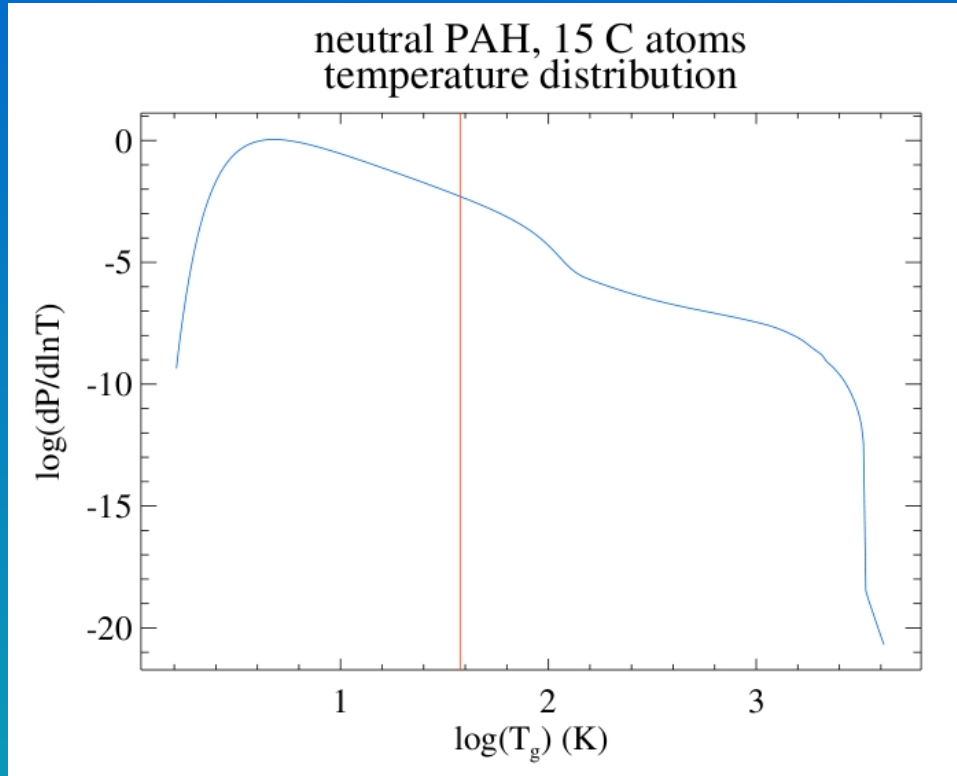


Image credit: P. van Hoof, unpublished work

When a very small grain (VSG, radius less than 100-200 Å) absorbs a single energetic photon, this will raise the temperature of the grain substantially (by up to several thousand kelvin) since the enthalpy of the grain is very small. The grain will then start to radiatively cool until the next photon is absorbed. If that takes a long time, the grain may cool to very low temperatures. Given the grain enthalpy, the grain optical properties, and the incident spectrum, you can calculate what the probability distribution over temperature of the grain will be. The red vertical line indicates the equilibrium temperature.

Stochastic grain heating (2)

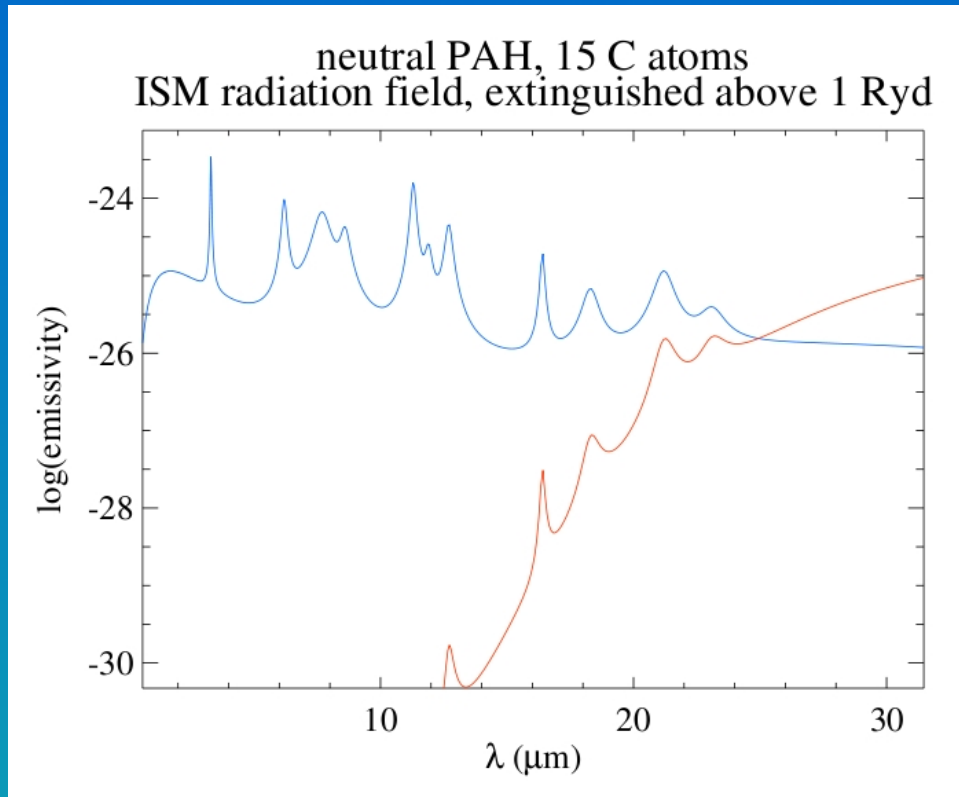


Image credit: P. van Hoof, unpublished work

Since thermal emission is a very non-linear function of temperature (exponential in the Wien tail) having these brief excursions to high temperatures significantly alters the emitted spectrum, even after averaging over time. The short wavelength end of the spectrum (blue curve) will be much stronger compared to the assumption that the grains are constantly at the equilibrium temperature (red curve). This is an important effect in the ISM.

Grain formation (1)

Grains typically form in dense outflows of evolved stars:

- AGB stars
- Late type WR stars
- Supernovas (efficiency still under debate)

These objects typically only form carbon-rich dust or oxygen-rich dust depending on whether $C/O > 1$ or $C/O < 1$. Some mixed-chemistry sources are known.

In the general ISM you will always find a mix of dust from various sources, so using both carbon-rich and oxygen-rich dust is appropriate.

In circumstellar environments the dust can be locally produced. In that case you need to choose between carbon-rich and oxygen-rich dust (e.g. AGB stars, planetary nebulae)

Grain formation (2)

There is a possibility that grains can also form (or grow) in the general ISM.

The most convincing argument for this comes from refractory elements (e.g. calcium or iron). These can be depleted by 2 dex or more in the gas phase, meaning that $> 99\%$ of that element is locked up in grains.

On the other hand, massive stars do not form dust grains in their winds and therefore inject undepleted gas into the ISM. There is also grain destruction in the ISM (e.g. through supernova shocks). This way lots of refractory elements should end up in the gas.

These two facts are hard to reconcile unless you assume that grains can form or grow in the ISM, thereby efficiently locking up at least the refractory elements. The mechanism for that is unclear though.

Grain destruction

Grains can be destroyed by the following processes:

- Sputtering: collisions with energetic particles gradually erode the grain.
- Sublimation: the grain is heated to temperatures where it simply evaporates.
- Coulomb explosions: the grain is charged so high that not enough electrons remain to maintain the molecular bonds (only important for PAHs).
- Shocks: strong shocks can destroy or shatter grains, but non-equilibrium calculations are needed to get reliable destruction rates.

In general it is true that small grains are more easily destroyed than larger grains. The most well-known effect of this is that PAHs cannot survive inside the ionized region. They are abundant in the PDR region, but what happens deep in the molecular region is not yet clear.

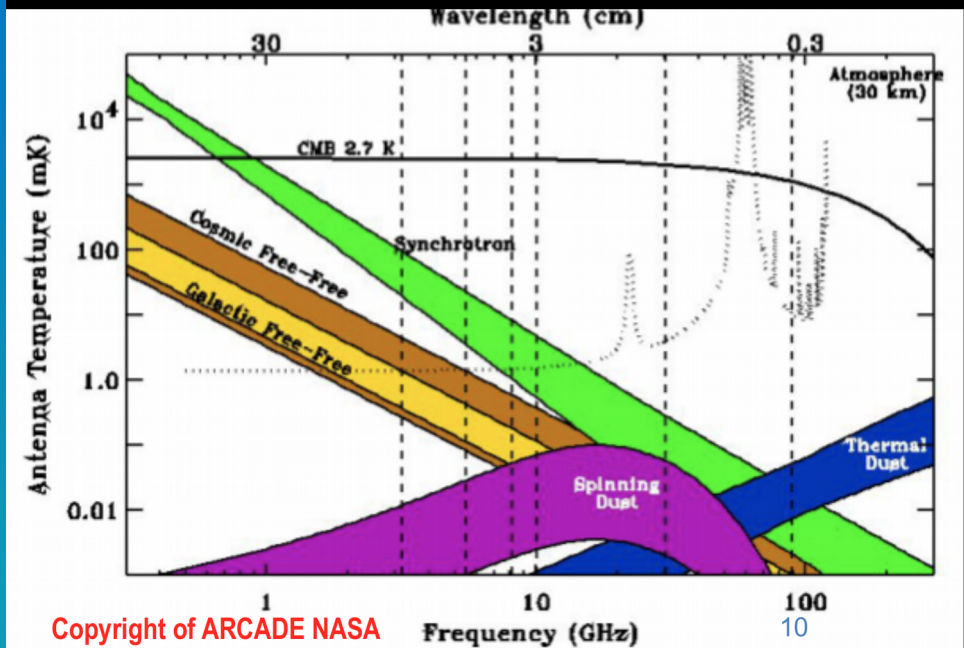
Spinning Grains

- If the grain has a dipole moment, the spinning grain will emit (anomalous) dipole radiation. Peaks typically at 20-30 GHz.
- The efficiency depends strongly on grain size → only PAHs are important.

Small grains are better emitters:

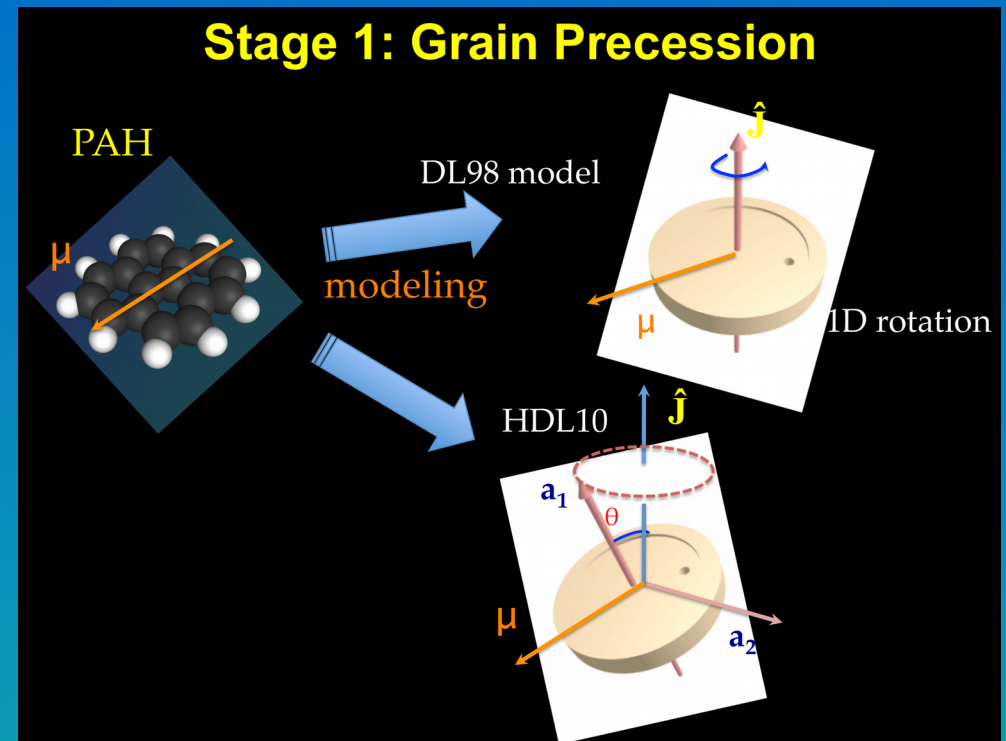
$$P \sim \omega^4 \quad \omega \sim r^{-5/2} \quad P \sim r^{-10}$$

Spinning dust, i.e. spinning tiny PAHs with dipole moments, has become an accepted foreground



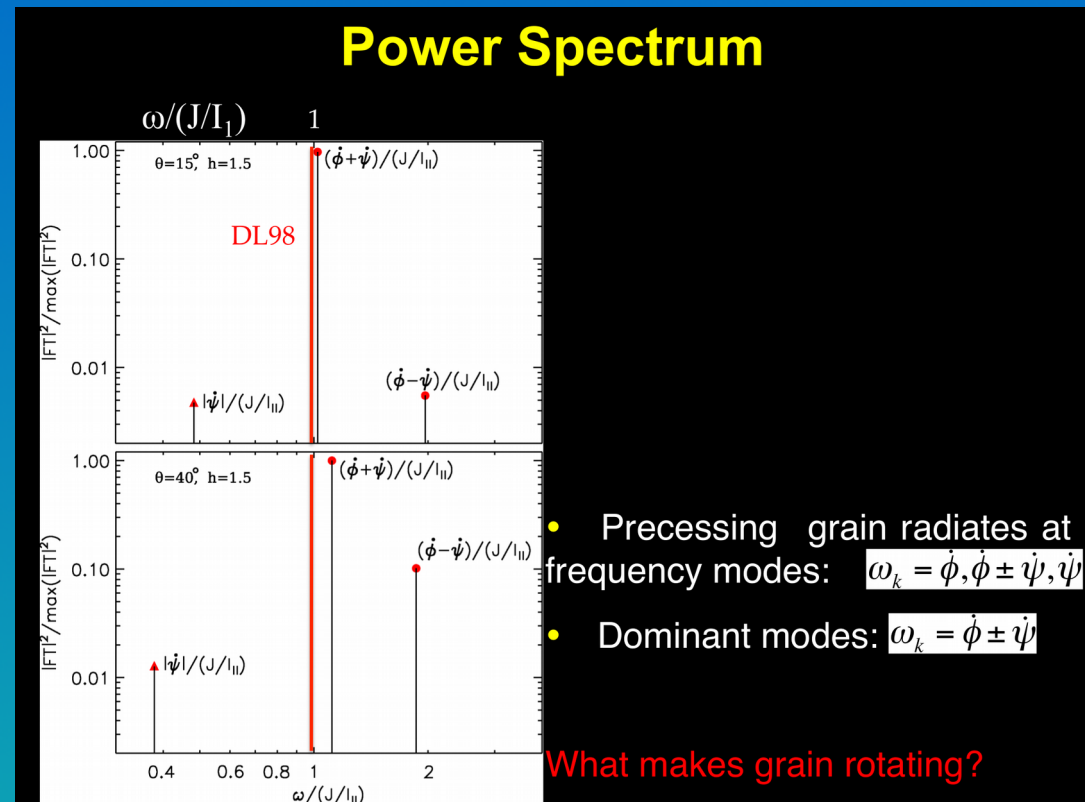
Spinning Grains

- The smallest PAHs ($a < 6 \text{ \AA}$) are assumed to be shaped like a circular disk with a thickness of 3.35 \AA , larger grains are assumed to be spherical.
- Draine & Lazarian (1998) assumed that the rotation axis was aligned with the symmetry axis of the disk.
- In Hoang, Draine & Lazarian (2010) this was relaxed and the two axes could be misaligned (grain wobbling).



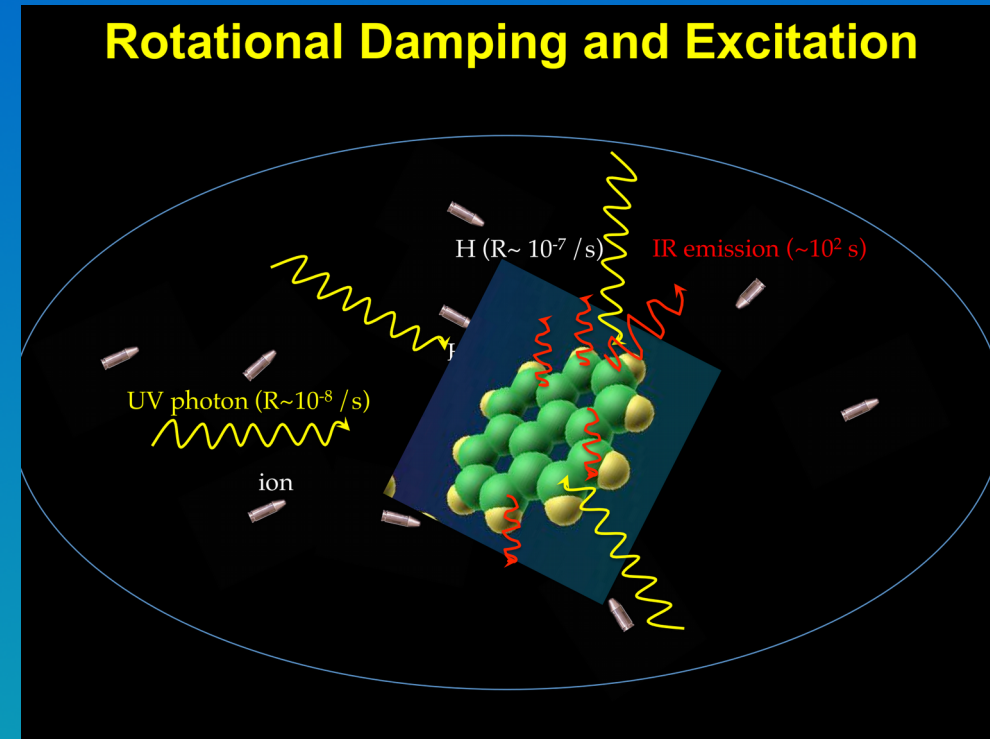
Spinning Grains

- In DL98 the grain will emit at a single frequency (marked in red)
- In HDL10 the grain can emit at three different frequencies, some of which are higher.
- Hence the emissivity at high frequencies will be enhanced!



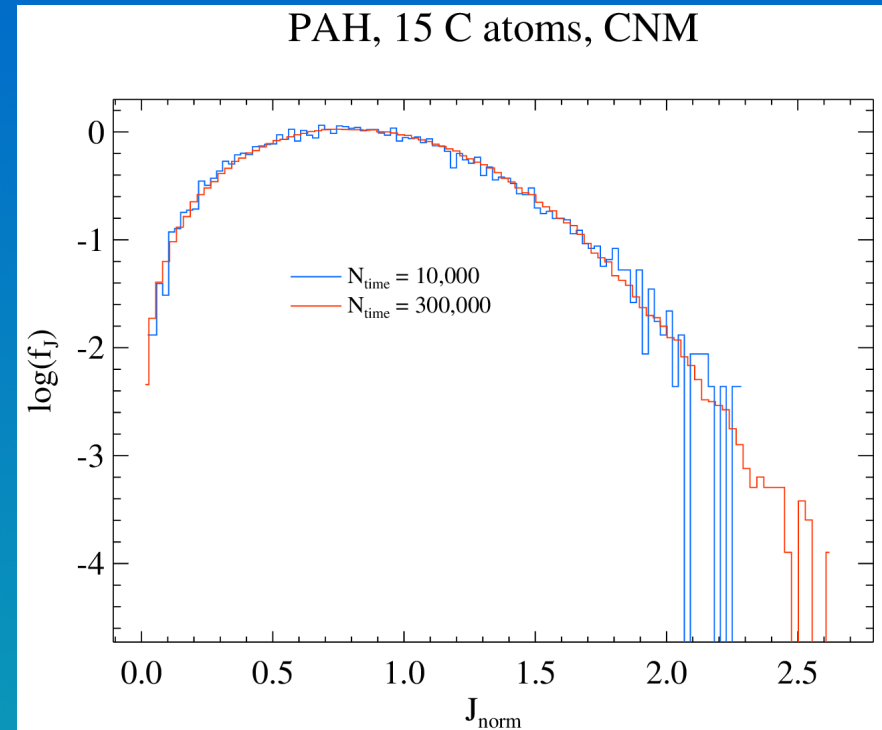
Spinning Grains

- Grains will be spun up by these processes
 - Impact by neutral and charged particles (atoms, ions, H_2 , CO)
 - Coulomb interaction with passing ions
 - Emission of thermal radiation ($2\hbar^2$ angular momentum per photon). This includes stochastic heating effects!
- **Not treated** are these processes
 - Electron emission (PE effect) and recombination (momentum too low)
 - Absorbing photons (ineffective for PAHs, can be important for larger grains)
 - H_2 formation on grains.



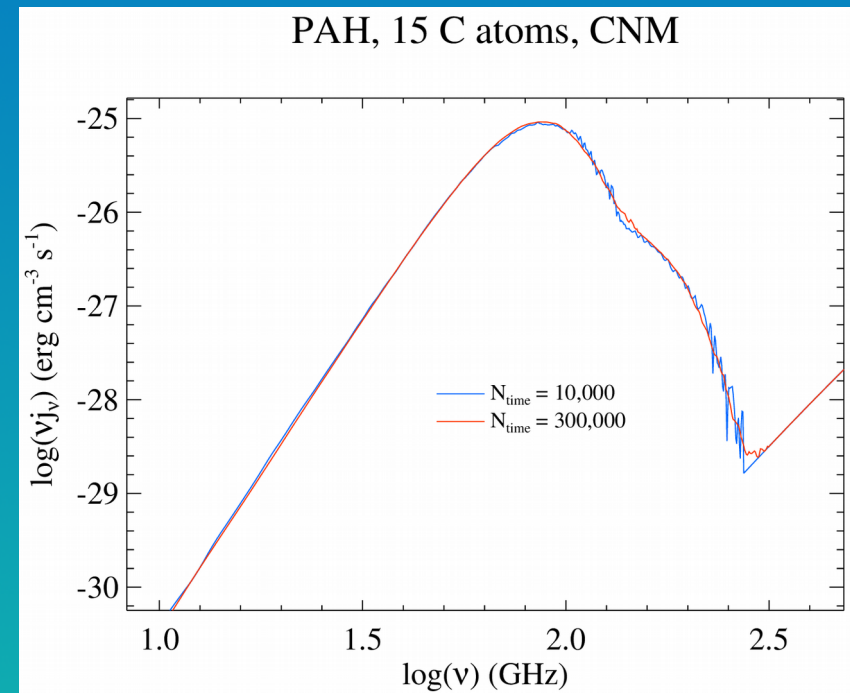
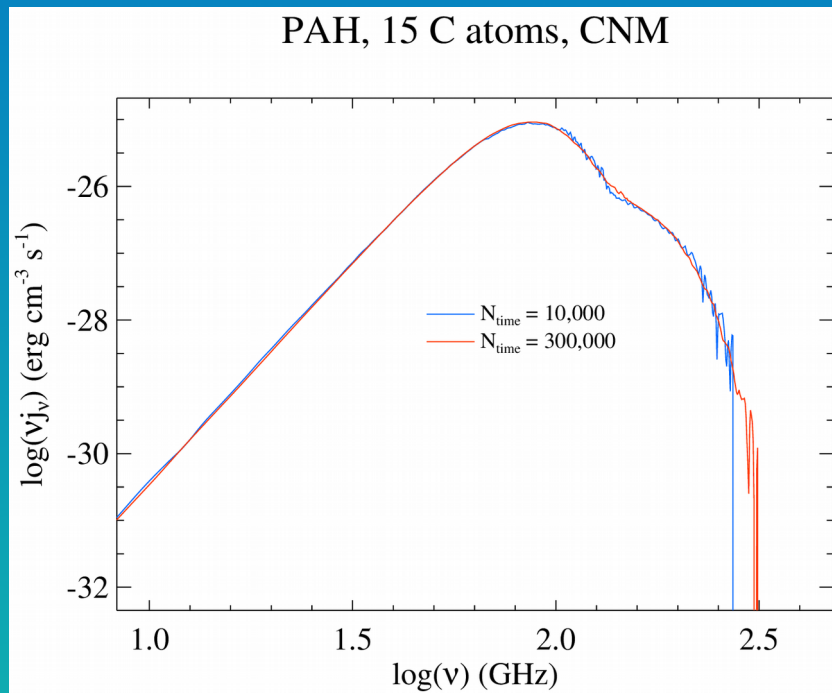
Spinning Grains

- The effects of all physical processes will be combined into two parameters F and G , which are constantly reevaluated for every grain species in every zone.
- Using F and G we solve the Langevin equation for the angular momentum distribution using a Monte Carlo code.
- The quality of the resulting distribution depends on the number of time steps, which is a free parameter.
- Once the angular momentum distribution is known, the dipole emissivity can be calculated (next slide).



Spinning Grains

- Increasing Ntime leads to emission at higher frequencies being included (left panel). This emission is however weaker than the thermal dust emission (right panel).
- The spinning dust code is currently being tested on a development branch. It will however not be ready for the C16 release.



Grains in Cloudy (1)

Cloudy has a built-in Mie code for spherical grains. This allows you to calculate grain opacities for arbitrary grain materials and size distributions. This includes EMTs for approximating mixed grain materials.

Several frequently used grain opacity files are pre-compiled and included in the data directory.

There are also several refractive index and size distribution files ready to use in the data directory.

You can also create your own refractive index and size distribution files. The syntax of the files is described in Appendix A of Hazy 1.

Several abundance commands **automatically** include grains in the simulation (e.g. the **abundances ISM** command). There is a complex interplay between grains set with the **abundances** command and the **grains** command. See Section 7.9.2 of Hazy 1 for details. Check your output to see if you get all the grains you expect! (especially if you use **abundances planetary**)

Use size-resolved opacity files. These give better results. Usually 10 size bins will be a good choice.

Grains in Cloudy (2)

For all grains other than PAHs, Cloudy by default assumes that the grain abundance is constant as a function of radius.

You can set the abundance as a multiplication factor on the **grains** command, or using the **metals grains** command. The resulting absolute abundance can be found in the main output above the output of the first zone, or in the section giving the mean properties (the mean dust-to-gas mass ratio). You can also use the **save grain abundance** or **save grain D/G ratio** commands. If sublimation can occur in the inner regions, use the keywords **sublimation function** on the **grains** command.

For PAHs the grain abundance is assumed to vary as a function of radius. The default law is to scale the abundance as $n(\text{H}^0)/n(\text{H})$. Using the **set PAH "H,H2"** command you can choose a law that scales as $(n(\text{H}^0)+2n(\text{H}_2))/n(\text{H})$. This keeps the PAH abundance constant deep inside the molecular region.

You can code up your own abundance law in the routine `GrnVryDpth()` at the end of `grains.cpp`. Use the keyword **function** on the **grains** command to enable this routine.

Grains in Cloudy (3)

Cloudy treats the grain physics in a self-consistent manner! There are **no free parameters** in the grain physics (other than the choices you make in the Mie code).

- The grain opacities are defined from the X-ray to the radio regime. The grain temperature is calculated from the amount of energy absorbed in the UV / X-ray assuming thermal balance.
- The optical constants are taken from lab experiments (or atomic data in the X-ray regime). We do not use beta laws or other approximations.
- The grain charge distribution is calculated self-consistently from the PE effect (including Auger cascades) as well as electron capture and charge exchange with ions in the gas. Grain charging is done fully self-consistent with the charge state of the gas. This is an important effect in fully molecular regions where grains tend to “soak up” the free electrons which can have a big effect on the chemistry.

Grain opacity files produced by the Mie code in Cloudy have a fixed frequency mesh. If you alter the Cloudy frequency mesh, you need to recompile the grain opacity files as well if you want to use it.

Stellar Atmospheres in Cloudy (1)

Cloudy allows you to incorporate a set of spectral energy distributions (SEDs), interpolate between the models, and use the result as input for your simulations.

- This was originally set up for grids of stellar SEDs, such as the Kurucz ATLAS models.
- The grid can have up to 4 dimensions, typically including T_{eff} and $\log(g)$, but possibly also metallicity, mass loss rate, or other parameters.
- Several popular grids are available for download from the Cloudy web site (see next slide).
- After downloading, the ascii file needs to be uncompressed and compiled into binary form for efficient use.
- The binary SED module is in principle platform dependent, but should in practice work on all modern Intel/AMD compatible hardware.
- The binary SED module is already binned on the Cloudy frequency mesh. So if you alter the Cloudy frequency mesh, you need to recompile the binary SED module if you want to use it.

Stellar Atmospheres in Cloudy (2)

Available at: <http://wiki.nublado.org/wiki/StellarAtmospheres>

Details about individual stellar grids

The following subsections describe details of each of the standard grids now in the code. People who wish to install all the available grids should download all the ascii files in our [stars directory](#) as well as visit the links shown below in the Tlusty, Rauch, and CoStar sections.

Atlas grids

[atlas_fp05k2_odfnew.ascii.gz](#) (log Z = +0.5)
[atlas_fp02k2_odfnew.ascii.gz](#) (log Z = +0.2)
[atlas_fp00k2_odfnew.ascii.gz](#) (log Z = +0.0)
[atlas_fm05k2_odfnew.ascii.gz](#) (log Z = -0.5)
[atlas_fm10k2_odfnew.ascii.gz](#) (log Z = -1.0)
[atlas_fm15k2_odfnew.ascii.gz](#) (log Z = -1.5)
[atlas_fm20k2_odfnew.ascii.gz](#) (log Z = -2.0)
[atlas_fm25k2_odfnew.ascii.gz](#) (log Z = -2.5)

[atlas_3d_odfnew.ascii.gz](#) (all metallicities)

These are the Castelli & Kurucz ([Castelli, F., Kurucz R. L. 2004, astro-ph/0405087](#)) LTE, plane-parallel, hydrostatic model atmospheres using the newest opacity distribution functions. The original versions of the files are from the [Kurucz web site](#). These grids come in various metallicities, ranging from log Z = +0.5 to -2.5, all with a turbulent velocity of 2 km/s. There is also a single file `atlas_3d_odfnew.ascii.gz` which contains all metallicities and which enables interpolation in log Z. If you download the latter file, *you do not need any of the other odfnew files*, unless you want to run the entire test suite. **The ODFNEW models are the most up-to-date and preferred versions of the Atlas models.**

You don't need to download all the files! Simply choose what you need, or skip all of them if you don't need them. Adding files later is also easy.

If you use one of these grids in a publication, cite the original author!

Stellar Atmospheres in Cloudy (3)

- Available are Kurucz grids (LTE, plane-parallel), Tlusty O & B stars grids (non-LTE, plane-parallel), various grids of PN central star SEDs by Thomas Rauch (non-LTE, plane-parallel), and two small grids of non-LTE wind-blanketed models (CoStar and WMbasic).
- You can also include grids of other types of SEDs this way! Most notably this is the case for stellar population synthesis models such as Starburst99 models. Such grids typically have either one dimension (age only) or two dimensions (age and metallicity).
- You can also create your own grids! The format of the ascii file is documented in Hazy.
- This is convenient if you want to interpolate between various SEDs. Once the ascii file is created and compiled, the interpolation will be done automatically.
- The command **table star available** will list all the grids you have installed (this only includes standard grids from the Cloudy site).
- The command **table star <grid> list** will list all the models included in a single grid.

Running Cloudy in parallel (1)

There are two commands in Cloudy that are parallelized:

- The optimizer
- The grid command

Both commands generate lots of individual models and are therefore very easy to parallelize.

There are two methods in use for parallelization:

- Based on the `fork()` system call
- Based on the Message Passing Interface (MPI)

Both methods have advantages and disadvantages which will be discussed on the next slide. It is also possible to run both commands sequentially on a single CPU, but this can take a long time!

All these methods should give essentially identical results. If they don't, you can report that as a bug.

Running Cloudy in parallel (2)

How do you choose which method of parallelization to use?

- The `fork()` method works out of the box. It will be automatically compiled into the code on all systems that support it in a default compilation. This includes Linux and other UNIX systems, Mac OS X, and Cygwin. So this is a hassle-free solution.
- The downside is that you can only fork new processes on a single machine, i.e. you are limited to a single computer or node.

So using the default `fork()`-based method of parallelization is ideal for running small/medium-sized grids (or optimizations with few free parameters) on your laptop or desktop or even a single compute server.

- The big advantage of MPI is that it allows you to use cores on multiple nodes of an HPC machine. This allows you to use a virtually unlimited number of cores (well, as many as the admins allow you to use...)
- The downside of MPI is that it needs external support scripts and libraries. The compilation and startup of MPI codes is not standardized and can differ from one system to another. You may need to load specific modules, or set search paths to find the executables and libraries.

Running Cloudy in parallel (3)

So for very CPU-intensive applications (large grids or optimizations with lots of free parameters) it can be advantageous to use MPI on a suitable HPC cluster.

In a default compilation, Cloudy will run the optimizer and grid commands in parallel on systems that support the `fork()` system call and where the number of cores can be determined. By default it will use all threads (except on Mac OS X, where it will ignore hyperthreads). The number of threads to use can be changed on the **optimize phymir** and **grid** commands.

How do you choose the number of cores?

- Optimizer runs can use no more than $2 \times N$ cores simultaneously, where N is the number of parameters that are varied by the optimizer. So ideally you would use $2 \times N$ cores, but you could also use N cores, or $\text{ceil}((2 \times N)/3)$, or some other small fraction of $2 \times N$. This choice assures that the threads are well-balanced.
- Grid runs will compute $N_1 \times N_2 \times \dots$ independent models, where N_1 is the number of grid points in the first variable, N_2 is the same for the second variable, etc. For good load balancing you can choose an integer divisor of this total number of models, but this is not strictly necessary.

Running Cloudy in parallel (4)

To compile Cloudy for an MPI run requires several steps.

- MPI typically uses a wrapper around the compiler, called something like mpiCC, but alternative names are mpicxx or mpic++. You may need to load a module to find this.
- You need to figure out which compiler the script wraps around. You can do this by typing “mpiCC --version”. Typically this will show that it is either g++ or icc (the Intel compiler).
- You need to compile Cloudy in one of the sys_xxx subdirectories. If mpiCC wraps around g++, use the sys_mpi_gcc directory. If mpiCC wraps around icc, use the sys_mpi_icc directory. In these directories you can compile as you normally would.
- To run the MPI executable, you typically need to use the mpirun executable, i.e. “mpirun /path/to/cloudy.exe -r script”. A batch system will communicate to mpirun how many ranks to start on which nodes. Outside of a batch system, other methods are needed, like using the “-np” flag (for running on the local machine) or using a hostfile (for running on remote machines). See the man page for more info.