Quick-start guide to Cloudy

Spring meeting, Keeneland, Lexington

Version 07.02.00

G.J. Ferland

Department of Physics and Astronomy
University of Kentucky, Lexington
http://www.nublado.org

Quick-start guide to Cloudy

Version 07.02

1. Introduction

Numerical simulations make it possible to understand complex physical environments starting from first principles. Cloudy is designed to do just that. It determines the physical conditions within a non-equilibrium gas, possibly exposed to an external source of radiation, and predicts the resulting spectrum. This makes it possible to predict many observed quantities by specifying only the properties of the cloud and the radiation field striking it.

This is a quick-start guide to Cloudy. It begins with an introduction to the code's web site, outlines how to set up the code, and is followed by a description of HAZY, the code's documentation. This document only gives an outline of the code and the commands that drive it. Sections of HAZY that go into greater detail are referenced and should be consulted to find out more.

Osterbrock & Ferland (2006; hereafter AGN3) gives many more details about photoionized clouds. Ferland (2003) reviews recent advances and current questions in this field.

1.1. The web site, discussion board, and repository

Cloudy is open source. Its source, atomic data files, a large suite of test cases, and its documentation Hazy, are available at the web site www.nublado.org. Begin setting up the code by going to the downloads / current stable version page of the web site. Follow the instructions under "setting up this version".

The web site includes a link to a discussion board. This is the place to ask questions, make suggestions, and report problems.

Cloudy actually lives in a Subversion source-code repository located here. Its ViewVC link is here. Knowledgeable users may prefer to obtain the code directly from the repository rather than from the web site.

1.2. The current stable version

Like all software, Cloudy goes through versions as it is developed. The fidelity of the simulation improves as processors become faster. Atomic and molecular rate coefficients become better as theory and experiment makes progress. Bugs are fixed. As a result many versions of the code exist and the code's predictions improve over time.

The ***current stable version*** should be used for all publications. Papers should say exactly what version of the code was used. All versions of Cloudy that ever have been the current stable version are still on the web site. This makes it possible to reproduce results obtained years ago should a question about results ever arise.

Several development versions are also on the web site and in the subversion repository. These are experimental and should not be used in published work.

New versions of the code, and patches to the current version, are announced on the discussion board.

1.3. Setting up the code

The following is only an outline. The web site goes into much greater detail.

Download the files. A complete download will create six subdirectories. These include the source, atomic data, documentation, and the test suite. Each directory has its own *readme*.htm* document.

Download optional stellar atmosphere files. Cloudy can use the continuum predicted by several stellar atmosphere codes. These are only needed if you want to use these in your calculations. Certain **table star** commands will not work but if you don't need these stellar continua then there is no reason to install them.

Edit *path.cpp*. The code must be able to find its data files. Edit the file *path.cpp* in the source directory and change the string *chDataPath* to give the location of the data directory.

Compile the source. The web page gives the best options for the most popular compilers. The code is written in ISO Standard C++ and it requires a good C++ compiler. It has been extensively tested with the GNU g++ compiler, which I recommend.

Run the smoke test, then the test suite. The next step is to make sure that the executable file you created is valid. First run the smoke test, a simulation with the single command **test**. If this works then run Cloudy's test suite, located in the *tsuite* directory. This is a large body of simulations that exercises the code over its limits. The *tsuite* directory has three subdirectories, *auto*, *slow*, and *programs*. You should run the test cases in the *auto* directory to confirm that you have a valid version of the code.

These tests check for bugs in your compiler and system library. The tests include **assert** commands that tell the code what result to expect. An error is announced if it does not obtain the expected result. Perl scripts in both the *auto* and *slow* directories can run the tests and check their output. The file *doc_tsuite.htm* in the *auto* directory describes individual test cases.

The programs in the *programs* directory show how to run Cloudy as a subroutine of a larger code. The *slow* directory contains sample input scripts which take a fair amount of time to run. The contents of these directories can be used as examples of how to use the code.

View the *hot fixes* and *problems* web pages. Hot fixes are small changes to the source that fix bugs that were discovered after the code was released. These corrections should be made after the test suite has been run since the hot fixes may change results. The problems page lists known problems with that version.

1.4. Hazy

HAZY is Cloudy's documentation. It comes in three volumes. Part 1 gives a complete list of all commands. Part 3 describes how to set up a calculation, defines the output that is generated, shows how to extract observed quantities from the predictions, and explains how to call Cloudy as part of a larger program. Parts 1 & 3 are kept up to date and posted on the web site when new versions of the code are released.

Part 2, which is badly out of date at the time of this writing, describes the physics behind the simulation. The past few years have seen a major expansion of the code's capabilities, especially in infrared emission and molecular physics. Unfortunately, Part 2 of Hazy has not had a high enough priority to be updated with available resources. I have not given up and do intend to update this section eventually. For now, an ADS search on Ferland will find the most recent papers that describe advances in the physics.

1.5. Assumptions

The code computes the non-equilibrium ionization, thermal, and chemical state of a cloud that may (or may not) be exposed to an external source of radiation. The usual assumption is that atomic processes have had time to become time steady. The density of a species or level *i* is given by a balance equation of the form

$$\frac{\partial n_i}{\partial t} = \sum_{j \neq i} n_j R_{ji} + Source - n_i \left(\sum_{j \neq i} R_{ij} + Sink \right) = 0 \ \ [\mathrm{cm}^{-3}\ \mathrm{s}^{-1}] \qquad (1)$$

Here R_{ji} represents the rate [s^{-1}] that a species *j* goes to *i*, *Source* is the rate per unit volume [cm^{-3} s^{-1}] that new atoms appear in *i*, and *Sink* is the rate [s^{-1}] they are lost. This, together with equations representing conservation of energy and charge, fully specify the problem.

In most calculations the cloud is assumed to be static. The geometry is currently one-dimensional. All line and continuum transfer processes are treated with escape probabilities.

1.6. What Cloudy can do

Given these assumptions, the code will determine the ionization, temperature, and chemical state of a cloud, and then predict its spectrum. All of this is done self-consistently with the minimum number of free parameters.

The equations of statistical equilibrium, charge conservation, and conservation of energy are solved. These determine the level of ionization, the particle density, the gas kinetic temperature, the chemical state, populations of levels within atoms, and the full spectrum, which often includes hundreds of thousands of lines. A very large number of observables result from a few free parameters. Often the goal is to determine these free parameters from observations. This approach is outlined in Section 8 of Ferland (2003).

The following sections outline how to set up the boundary conditions for a calculation. You create an input script that specifies the cloud's geometry, composition, density, and thickness. The radiation field striking the cloud, often its only source of heat and ionization, is specified, and other sources of heat can also be included. The code then computes the thermal, ionization, and chemical properties of the cloud, and its observed spectrum. A general introduction to such environments is given in Chapters 1 and 2 of AGN3. The first chapter of Part 1 of HAZY gives a more extensive overview of the code. The format of the input script, the file that specifies the boundary conditions, is described in the chapter *Introduction to commands* of Part 1 of HAZY, and output options are described in the chapter *Controlling output* of Part 1 of HAZY.

1.7. Definitions

There is a fair amount of jargon that goes along with quantitative spectroscopy and numerical simulations of a cloud. These are summarized in the chapter *Definitions* of Part 1 of HAZY and in Chapters 1 through 5 of AGN3. As a minimum you should know the definitions of the types of geometries (open and closed), continua (incident, transmitted, diffuse, and reflected), and the terms covering factor, density, and column density.

1.8. Citing the use of Cloudy

Publications should cite the code by giving the version number and a reference to the last major review of Cloudy's development, Ferland et al. (1998). An example would be "We used version 07.02.01 of Cloudy, last described by Ferland et al. (1998)". Citations are important for sustaining the development of Cloudy. The exact version should be given so that, years from now, it will be possible to find how a particular result was obtained. The command **`print citation`** will print the correct citation in a LaTex-friendly format. Please follow the citation example shown above.

1.9. Collaborations and the future

The web site has a page devoted to intended future developments. Fully time-dependent and advective flows (Henney et al. 2005 is the first paper), exact radiative transfer, and two and three dimensional geometries, are under development.

Some of the most useful additions to the code have been surprises donated by volunteers. There is a great deal of work left to be done. You are welcome to help out!

2. Two very simple models

As a simple introduction let's create two simple models. The first is a planetary nebula (see Chapter 10 of AGN3) and the second is a cloud in the Broad Emission Line Region of a quasar (Chapter 13 of AGN3). As a minimum you must specify a few parameters. These include the shape and luminosity of the continuum striking the cloud, the cloud's density, radius, and composition (if it is not solar). We go over these briefly here.

2.1. Running the program

Cloudy needs to be able to deduce the following information before it can predict conditions within a cloud; the shape and brightness of the radiation field striking the cloud, the total hydrogen density, the composition of the gas and whether grains are present, and the thickness of the cloud. Unless otherwise specified the gas-phase abundances will be close to solar values and grains will not be included.

The code's philosophy is for a reasonable set of conditions to be assumed by default. These are listed in Section 3.3 of Part 1 of HAZY. Commands are added to change these conditions.

2.1.1. Command format

A series of commands tell Cloudy what to do. They are entered, one command per line, into an input file that the code reads. With a few important exceptions, the commands can be entered in any order. Each command is identified by the first 4 or 5 letters on the line. The format for commands is described further in the chapter *Introduction to commands* of Part 1 of HAZY.

Parameters are specified by numbers that appear on the command line. Exponential notation cannot be used – the number 2e3 will be interpreted as a two followed by a three. The code generally avoids exponents by entering numbers as logs. It is important to read HAZY to see what is expected for numerical input. For instance, numbers representing temperatures are interpreted as logs if they are ≤ 10 and as the linear temperature if greater than 10. Many commands have the keywords **`log`** and **`linear`** to change the default interpretation of a number.

2.1.2. Input for a single model

Each line in the input file gives a command. These commands are described further below. Comments can be entered in several ways, as described in the subsection *Comments* in the chapter *Introduction to Commands* in Part 1 of HAZY. In examples below lines beginning with "//" are comments. The file ends with a blank line or the end of file.

If the executable is called *cloudy.exe* then a single simulation could be run with the command

```
cloudy.exe < sim.in > sim.out
```

where the file *sim.in* contains the input commands and the main output will go to the file *sim.out*. This method of running the code is described in the chapter *Running a Single Model* in Part 3 of HAZY.

I also keep a "run" script on my path – in Unix the contents of the *run* file are the following

```
/u/home1/gary/cloudy/trunk/cloudy.exe < $1.in > $1.out
```

and in Windows the following is in *run.bat*:

```
c:\projects\cloudy\trunk\debug\cloudy.exe < %1.in > %1.out
```

In both cases the code could be run with the command "*run sim*". The code would read commands from the file *sim.in* and write its output to the file *sim.out*.

2.2. A simple planetary nebula

A planetary nebula is a cloud of gas and dust that has been ejected by a dying star (see Chapter 10 of AGN3). The temperature and luminosity of the central star are specified since starlight is the energy source. The central star is a cooling white dwarf. We approximate it as a hot blackbody at the Eddington limit for one solar mass. We set the stellar continuum shape and luminosity with separate commands:

```
blackbody, T=100,000K  // the continuum shape
luminosity total 38    // the continuum luminosity
```

The continuum shape is a 10^5 K blackbody. It has a total luminosity of 10^{38} erg s^{-1}. Luminosity and shape commands are described in the section beginning on page 18 below.

Iin general the continuum striking the cloud can be specified either of two ways. In the *luminosity case* the total luminosity emitted by the central object and the inner radius of the cloud are given. In the *intensity case* only the flux of radiation striking the cloud is specified. These are described in Section 3.2 starting on page 11 below. This example specifies the luminosity case. The code must be able to derive the flux of photons striking the cloud's surface so we will need to specify the inner radius of the surrounding nebula.

A typical inner radius for the shell of a planetary nebula is a fraction of a parsec, so let's use 10^{18} cm. A typical hydrogen density, measured from ratios of forbidden lines (AGN3 Chapter 5), is roughly 10^5 cm^{-3}. Images suggest that the gas fully covers the star so we include the **`sphere`** command (discussed in Section 3.3 on page 12 below). Finally, the code would assume a solar chemical composition if we didn't change it. The composition of the ejected gas is actually quite different and it includes dust. A built-in mixture of gas and dust that are specified by the **`abundances`** command (page 16 below). We will use the mixture that is typical of planetary nebulae. The input commands would be the following:

```
radius 18 // the log of the inner radius in cm
hden 5    // the log of the hydrogen density
sphere
abundances planetary nebula
```

We did not set an outer radius so the calculation will continue until the gas kinetic temperature falls below the default value of 4000 K. Generally this will be near the H^+ - H^0 ionization front. Many other stopping criteria could have been used (discussed in Section 3.5 on page 14 below) but in this simple model we are only interested in emission from the H^+ region.

We will have the code create two output files in addition to its standard output. We add **punch** commands which specify what quantity to output and a file name (see page 26 below).

```
punch overview "pn.ovr"
punch continuum "pn.con" units microns
```

The "overview" punch file makes it possible to create plots showing the temperature and ionization structure of the cloud. The "continuum" punch file will show the incident and total continuum. We included the keywords **units microns** to give the spectrum as a function of the wavelength in microns. The default is the photon energy in Rydbergs. Other options are available.

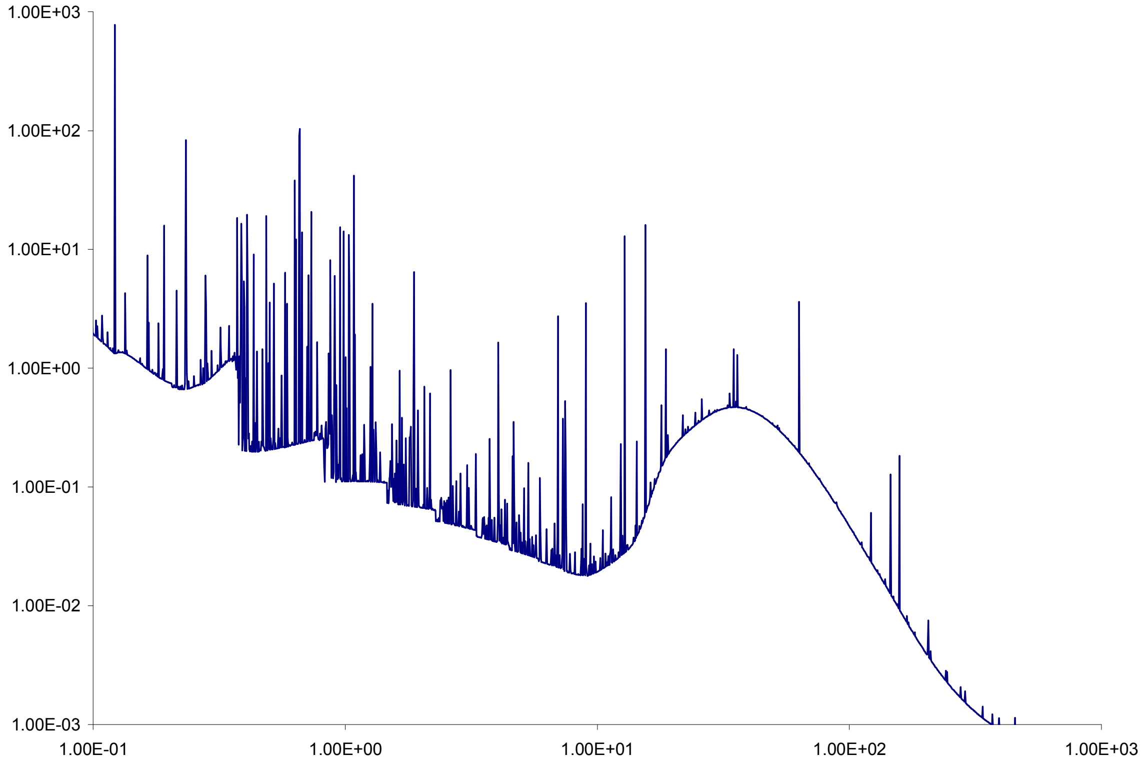
Use a simple editor like vi, emacs, or notepad to create a file with the commands shown in the following sections. Place all of these commands into a single file with the name "pn.in". The commands should be written one after another and there should not be any blank or empty lines before the end of the commands. The entire file will contain the following:

```
blackbody, T=100,000K
luminosity total 38
radius 18
hden 5
sphere
abundances planetary nebula
punch overview "pn.ovr"
punch continuum "pn.con" units microns
```

Run the program as follows:

```
cloudy.exe < pn.in > pn.out
```

You will end up with three output files, the main output **pn.out,** and the two punch files **pn.ovr** and **pn.con**. Have a look at **pn.out**. The output is described further in Section 7 starting on page 25 below. Use the two punch files to create plots. The first line of a **punch** file gives titles for the columns of numbers which are separated by tabs. The first column of the overview file gives the radius in cm. Another column gives the gas temperature. Make a plot showing the temperature as a function of radius. The first column of the continuum file gives the wavelength in microns. Column seven gives the total emission. Make a plot showing the emission as a function of wavelength. Both axes will probably need to be logs due to the large dynamic range. The spectrum is shown below.



The spectrum of a planetary nebula. The x-axis is the wavelength in microns. The y-axis is νf_ν [erg cm^{-2} s^{-1}].

Dust in the nebula, warmed by light from the central star, produces the large emission feature centered at ~30 μm (Chapter 7 of AGN3). Recombination continua (Chapter 4 of AGN3) produce the cliff-like features in the optical and near infrared. The emission lines are produced by both recombination (AGN3 Section 4) and collisional excitation (AGN3 Section 3).

2.3. A quasar cloud

Next compute a simple model of a quasar broad emission-line cloud. The "BLR" clouds are located close to the central engine of an active nucleus (see Chapters 13 and 14 of AGN3) and they probe conditions near the most massive structures that formed in the young universe. The shape of a quasar continuum is often fitted by a set of power laws. Most of the literature in this field describes the continuum intensity in terms of the flux of photons in the hydrogen-ionizing continuum φ(H) (the intensity case) so we will too.

```
table power law  // a built-in power-law continuum
phi(H) 18.5      // the log of the flux of H-ionizing photons [cm-2 s-1]
```

A number of different continuum shapes are stored in the code as look-up tables. This **table** command uses a power-law continuum with a slope $f_\nu \propto \nu^{-1}$ in the visible / UV and with a roll over in the X-rays and infrared. The continuum intensity is given as a flux of photons [cm^{-2} s^{-1}] so this is the intensity case (page 11 below) and a starting radius does not need to be specified.

We still need to specify a hydrogen density and a stopping criterion[1]. A density of ≈10^{10} cm^{-3} is deduced from ratios of emission lines (AGN3 Chapter 13). Most published calculations stop at a certain column density so let's stop at a total hydrogen column density of 10^{22} cm^{-2}. This is large enough for an H^+ - H^0 ionization front to be present within the cloud. Solar abundances are fine so we will not change these. The covering factor, the fraction of 4π sr covered by gas, is small so the **sphere** command is not included. We have:

```
hden 10  // log of hydrogen density cm-3
stop column density 22  // log of hydrogen column density cm-2
```

Many line transfer effects are very important in the BLR. It is necessary to iterate on the solution to converge the optical depths. We do so by including the following command

```
iterate to convergence
```

We will create the same two punch files but with different file names

```
punch overview "blr.ovr" last
punch continuum "blr.con" units microns last
```

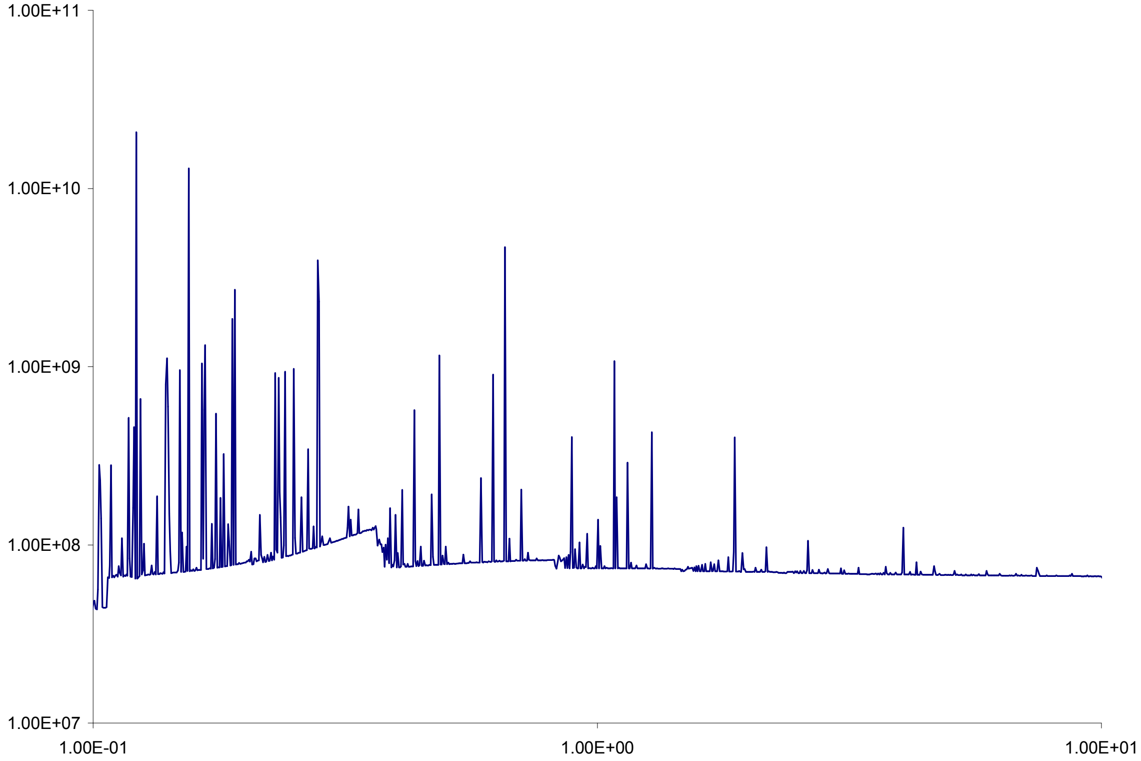
The keyword **last** says to save results for the last iteration. The entire input script, which we will call *blr.in*, is as follows.

```
table power law
phi(H) 18.5
hden 10
stop column density 22
punch overview "blr.ovr"
punch continuum "blr.con" units microns
```

Run the code like we did above. Examine the main output *blr.out* and make the same plots of the temperature structure and emitted continuum. The predicted continuum is shown below.

These are nearly the simplest models that can be calculated. The following sections go over each of the parameters described above and point to sections of HAZY that go into more detail.

[1] A planetary nebula is ionized by a hot star while the continuum striking a BLR cloud is very energetic, extending far into the γ-Rays. The gas temperature falls to ~100 K on the neutral side of the H^+ - H^0 ionization front in a PN since little radiation penetrates through the front. In a BLR cloud a warm partially ionized zone, heated by X-Rays, extends beyond the front so a stopping criterion must be specified. This is one of the major differences between a cloud ionized by starlight and one ionized by a hard non-thermal continuum.



The spectrum of a broad emission-line cloud in an active nucleus. The x-axis is the wavelength in microns. The y-axis is νf_ν [erg cm^{-2} s^{-1}].

2.4. Heads up!

Exponential notation cannot be used to enter numbers in a command. The number 3e4 will be interpreted as two numbers, a 3 followed by a 4. Many commands would accept 30,000 as its log, the number 4.477.

Most commands expect ***numerical parameters*** on the command line to be ***in a particular order, although they can often be omitted from right to left***. Be sure to follow the rules for each command. Default values are assumed when an optional parameter is missing.

Cloudy is designed to be autonomous and self-aware. It continuously audits itself as a calculation progresses. The end of the calculation may have comments on various aspects of the physics. ***If problems occur*** then the code may generate a warning or caution. These are described further on page 25 below.

3. Geometry

3.1. Zones and iterations

The code works by dividing a cloud into a large number of thin layers called zones. There is a default limit of 1400 zones, but this can be changed with the **`set`**

nend command. The code will generate a warning if the calculation stops because the default limit to the number of zones was reached since this probably was not intended. Warnings, cautions, surprises, and notes are described on page 25 below.

By default the code will do one iteration, one complete simulation of a cloud. If line or continuum transfer is important then more than one iteration will be necessary for a valid solution since the optical depths must be known. The number of iterations is controlled with the **iterate** command. The code will complain if too few iterations were performed for the optical depth scale to be converged.

3.1.1. Commands normally used

set nend changes the default limit to the number of zones.

stop zone tells the code to stop at a particular zone. This is mainly used for debugging, or, in the case **stop zone 1**, to only compute the cloud's illuminated face.

iterate sets the number of iterations to be performed. The default is a single iteration and more will be needed when radiative transfer effects are important. There is a special version of the command, **iterate to convergence**, which tells the code to iterate until line and continuum optical depths become stable.

3.1.2. Heads up!

The code will ***generate a warning*** if it stops because it ***reaches the default limit to the number of zones*** since this probably was not intended. Use the **set nend** command to increase the limit to the number of zones.

The code ***will generate warnings or cautions if the optical depth scale changes in the last iteration***. Increase the number of iterations if this occurs with the **iterate** command, or use the **iterate to convergence** command.

3.2. The geometry – intensity & luminosity cases

The brightness of the continuum striking the cloud can be specified as an intensity or luminosity. It must be possible to deduce the energy striking a unit area of the cloud. The subsection *Intensity vs luminosity commands* of Part 1 of HAZY describes the distinction between these two cases.

In the intensity case the energy flux (erg cm^{-2} s^{-1}) or photon flux (cm^{-2} s^{-1}) striking a unit area of cloud is set. The inner radius does not need to be specified. The emission per unit area [erg cm^{-2} s^{-1}] is predicted[2].

[2] The emission per unit area is called the "intensity" in this document, in the code, and in Hazy. For an optically thick slab this is actually the emittance. For an optically thin source this is $4\pi J$ where J is the mean intensity as defined in most radiative transfer texts.

In the luminosity case the total luminosity of the central source of radiation is specified and the inner radius of the cloud must be set. The luminosities of emission lines [erg s^{-1}] are predicted.

The gas covering factor $\Omega/4\pi$ (AGN3 Section 5.9) is the fraction of 4π sr covered by gas as seen from the central object. If the central object has a total luminosity L then the nebula intercepts $L\Omega/4\pi$ of the radiation field. The covering factor will linearly affect line luminosities but have only second-order effects on line intensities.

If the code can determine the separation [cm] between the continuum source and the cloud then it will predict emission-line luminosities. Otherwise it will predict line intensities. This is described further in the section Geometry of Part 1 of HAZY.

3.2.1. Commands normally used

`radius` sets the inner radius of the cloud. Line luminosities can be predicted if this is specified. The command can also specify the cloud thickness.

`covering factor` sets the covering factor of the cloud $\Omega/4\pi$. The predicted emission-line luminosities scale linearly with the covering factor.

3.2.2. Heads up!

None yet.

3.3. Open or closed geometry

The chapter *Definitions* of Part 1 of HAZY defines open and closed geometries and the chapter *Geometry* goes into more details. These considerations affect the transfer of diffuse fields and only change the predicted line intensities at the ~10% level.

An open geometry is the default. This is one where diffuse emission from the illuminated face of the cloud escapes from the system without striking other clouds.

A closed geometry is one where gas covers most of the continuum source. The **`sphere`** command sets this case. Emission from the illuminated face of the cloud passes the continuum source and strikes gas on the far side.

There are two classes of closed geometries, static and expanding. In the expanding case, the default when the **`sphere`** command is included, resonance line photons that cross the central cavity will not be absorbed by gas on the far side due to Doppler shifts. In the static case (the **`sphere static`** command) emission lines cross the central hole and are absorbed on the far side. These considerations have little effect on most lines but do affect emissivity of higher-*n* Lyman lines of hydrogen.

This is described further in the section in Part 1 of HAZY where the **sphere** command is discussed.

3.3.1. Commands normally used

sphere [expanding, static] tells the code to assume a closed geometry. The shell can be either static or expanding. By default the shell is assumed to be expanding rapidly enough that lines escaping from the illuminated face of the shell do not interact with the gas on the far side of the central hole. If the **static** keyword appears on the **sphere** command then lines escaping from one side will be absorbed by gas on the far side. The **expanding** option does not change the intrinsic line width, which is still assumed to be thermal. It only changes whether lines formed in opposite sides of the shell interact.

3.3.2. Heads up!

These considerations affect the transport of the diffuse fields and have only second-order effects on the predicted spectrum or physical conditions in the gas. If you are uncertain about the geometry, try the simulation with and without the **sphere** command, and try both **sphere static** and **sphere expanding**. Is this distinction important? It usually is not.

3.4. Is the gas static or a wind? Is it turbulent?

The cloud is normally assumed to be stationary. Lines are broadened only by thermal motions. A component of microturbulence can be added and a wind can be computed.

3.4.1. Commands normally used

turbulence adds a component of microturbulence to line broadening. This will make line pumping by the incident continuum more important and line trapping less important, so it does affect the spectrum. The parameter is the turbulent velocity u_{turb} in km s^{-1}.

wind simulates an expanding wind. A static cloud is assumed by default. The equations of motion determine the velocity as a function of depth. LVG or Sobolev approximation escape probabilities are used for the line transfer. The parameter is the initial wind velocity in km s^{-1}.

The effects of both commands are discussed further in the sections in Part 1 of HAZY where these commands are described.

3.4.2. Heads up!

If turbulence is included then a turbulent pressure term is added to the gas equation of state, the relationship between density and pressure. This affects the density within constant-pressure clouds. The gas equation of state is discussed in the *Optical Depths and Radiative Transfer* chapter in Part 1 of HAZY.

The velocity entered in the **turbulence** command is the component of turbulence u_{turb} that is added to the thermal width u_{th}

$$u = \sqrt{u_{th}^2 + u_{turb}^2} \ \ [\text{km s}^{-1}]$$

to determine the total line width u.

The velocity that appears on the **turbulence** command is in km s^{-1} because observed velocities are always expressed in these units. Cloudy actually works in cgs units.

3.5. What sets the outer edge to the cloud? Why should the calculation stop?

The code starts at the illuminated face of the cloud and works its way into deeper regions. The integration must stop for some reason. In many cases the outer edge of the simulation is not the outer edge of the cloud but rather the region where the gas has become cold and neutral. In other cases the column density of the cloud may be known from observations. Many different stopping criteria can be specified and the code will stop when the first one is reached.

Cloudy was originally designed to interpret optical / UV emission lines in quasars. These lines are produced in warm ionized gas so the default is to stop the calculation when the gas temperature falls below 4000 K. This will often be near the hydrogen ionization front. This would be a mistake if you want to consider cool atomic or molecular regions.

You should understand what sets the outer edge of the cloud you wish to simulate and then confirm that the code reached that point. The introduction to the chapter *Stopping Criteria* of Part 1 of HAZY goes into this in more detail.

3.5.1. Commands normally used

radius sets the inner and outer radius of the cloud.

stop temperature sets the lowest kinetic temperature to allow. The calculation will stop when the temperature falls below this value. The default is 4000 K. This will usually cause the calculation to stop near the H^+ - H^0 ionization front. You must set a lower temperature if you want the calculation to extend into the PDR or molecular cloud.

stop thickness sets the thickness of the cloud, the distance from the illuminated face to the outer edge.

stop column density sets an upper limit to the hydrogen column density [cm^{-2}]. The default is to include hydrogen in all forms (H^0, H^+, and H_2) although keywords can be used to select only species.

stop efrac stops the calculation when the electron fraction, $n_e/n(\text{H})$, falls below the specified value. This makes it easy to stop near ionization fronts.

double This doubles the computed optical depths at the end of an iteration. This command should be used if the region being simulated is only a layer on a much larger structure. This is the case in a PDR calculation, where an unmodeled molecular cloud is assumed to lie beyond the shielded face of the PDR. Lines will be quite optically thick at this outer edge. Emission from the shielded face will be suppressed if this command is used since we then assume that the layer is the mid plane of the cloud. Were this command not included the code would assume that the outer edge of the model is the outer edge of the cloud and optically thick lines would freely radiate from the shielded face. This is unphysical. The physics is described further in the chapter *Optical Depths and Radiative Transfer* of Part 1 of HAZY.

3.5.2. Heads up!

The stop temperature and stop efrac commands are useful ways to stop a calculation in the PDR or molecular cloud. Use the **stop temperature** command by itself to stop the calculation at a low temperature. Set the stopping temperature to a very low value and use the **stop efrac** command to stop the calculation when the electron fraction falls below a certain value.

The ***calculation will stop when it reaches a depth where any of the stopping criteria are satisfied***. Understand why the calculation stopped. Did it stop for the reason you expected or did it stop prematurely because another criterion was met or because the calculation had problems? Is the calculation a complete simulation of the region you want? The code will explain why it stopped in the first lines after the last zone. A sample printout of the last zone and the explanation for why the calculation stopped follows.

```
####259  Te:2.980E+01 Hden:1.000E+05 Ne:2.441E-01 R:1.000E+30 R-R0:1.408E+17 dR:7.588E+15 NTR: 27 Htot:9.894E-24 T912: 9.14e+04###
 Hydrogen      6.68e-01 2.44e-06 H+o/Hden 6.68e-01 6.63e-14 H-    H2 1.66e-01 7.65e-14 H2+ HeH+ 8.86e-15 Ho+ ColD 9.92e+21 1.01e+17
 Helium        1.00e+00 5.20e-08 0.00e+00 He I2SP3 1.14e-15 Comp H,C 1.78e-30 5.98e-31 Fill Fac 1.00e+00 Gam1/tot 1.58e-02
 He singlet n  1.00e+00 5.81e-22 9.19e-27 8.61e-29 4.08e-28 3.12e-28 He tripl 1.14e-15 1.12e-26 1.53e-28 1.66e-27 8.53e-28
 Pressure      NgasTgas 2.70e+06 P(total) 3.73e-10 P( gas ) 3.73e-10 P(Radtn) 2.14e-17 Rad accl 3.99e-14 ForceMul 8.66e+00
               Texc(La) 2.95e+03 T(contn) 3.00e+01 T(diffs) 6.30e-01 nT (c+d) 7.39e+06 Prad/Gas 5.75e-08 Pmag/Gas 0.00e+00
 Lithium       2.20e-01 7.80e-01 0.00e+00 0.00e+00 Berylliu 6.14e-01 3.86e-01 0.00e+00 0.00e+00 0.00e+00 sec ion: 1.34e-17
 Carbon        0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 H2O+/O   0.00e+00 OH+/Otot 0.00e+00 Hex(tot) 0.00e+00
 Nitrogen      0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 O2/Ototl 0.00e+00 O2+/Otot 0.00e+00
    model of cloud with primordial abundances exposed to background at Z=10
   Calculation stopped because lowest Te reached.    Iteration 2 of 2
   The geometry is plane-parallel.
```

3.6. What about clumping?

Clumping can be included. There are three general considerations.

There are powerful selection effects at work when a range of densities exist. You will tend to observe the highest-density regions because the emission per atom is proportional to density if the line is below its critical density (see AGN3 Section 3.5). Only with a particular mix of densities, where the amount of material at each density exactly compensates for the change in emissivity, will an observer notice emission from a range of densities. So I am always very skeptical at claims that a range of densities contribute to a single spectrum. This would require a coincidence.

But clumps do exist. If the clump size is small compared with the physical thickness of the H^+ region then they can be treated with a filling factor (see the discussion on page 18 below and AGN3 Section 5.9). In this case the gas is modeled as small clumps that are surrounded by vacuum or much lower-density gas. This is done by simply including the **`filling factor`** command.

If the clumps are larger than the physical thickness of the H^+ region then each clump will have its own ionization structure. This is the "LOC" model of quasar emission-line clouds described by Baldwin et al. (1995; the model is developed in several papers by the same team). In this case we compute grids of models and save the results. The spectra are then co-added using distribution functions to describe the range of cloud properties. The final spectrum depends on these distribution functions. The program *mpi.cpp* in the *programs* directory in the code distribution computes a grid of models and extracts the predictions using MPI on a distributed memory machine. The **`grid`** command (see page 23 below) can also be used to compute large numbers of models.

4. Composition and density

What is the chemical composition of the gas? Should grains be included? Should PAHs be included? Commands that set the composition are discussed in the chapter *Chemical Composition* of Part 1 of HAZY. The default composition is close to solar and grains are not included.

The density at the illuminated face of the cloud, and a description of how this density varies with depth, must also be given. By default the code will assume that the density and composition do not change across the cloud.

4.1. Chemical composition

The composition is set by giving the abundances of the lightest 30 elements by number relative to hydrogen.

4.1.1. Commands normally used

`abundances` sets the abundances of all elements to the values given on the line. If no numbers are present but a keyword is given then the composition is set to a standard mixture. Examples include the local ISM or a typical planetary nebula. Grains are included in some abundances sets. Consult the *Chemical Composition* chapter of Part 1 of HAZY to find out more.

`element` sets the abundance of a particular element, removes the element from the calculation, or specifies its ionization state.

`grains` determines the type and abundance of grains. If grains are included then, by default, their abundance will be the same across the cloud and quantum heating will be included when it is important. The **`function`** keyword will make the grain abundance depend on position and the **`no qheat`** keyword will turn off quantum heating. By itself this command specifies classical or large grains but does not include PAHs.

`grains PAH` includes PAHs. By default their abundance will be proportional to the ratio $n(H^0)/n(H_{tot})$ as suggested by observations of the Orion Bar (Section 8.5 of AGN3).

It is easy to create new types of grains by specifying refractive index and size distribution data. This is described in *vanhoof_grain_model.pdf* located in the *docs* directory of the download.

`metals` changes the abundances of the "metals" (all elements heavier than helium) by the scale factor given on the command line. It can also change the gas to dust ratio or set depletion factors for the gas-phase abundances of the elements.

4.1.2. Heads up!

Grain sublimation: A warning will be printed if grains become hotter than their sublimation temperature. They will not be removed from the calculation. The effects of grain sublimation can be mimicked by making the grain abundance vary with depth with the **`function`** option on the **`grains`** command, but this requires writing a new routine. Note also that if grains are destroyed the material contained within them must be returned to the gas phase to be self consistent. This also is not done automatically. Finally, a note will be printed if grains are not present but could exist since they would have been below their sublimation temperature.

Gas-phase abundances and grains: It is possible to leave the gas-phase composition at its solar value, appropriate if all elements are in the gas phase, but also set a population of grains with the **`grains`** command. This is not consistent. When grains are present the elements that comprise them are depleted from the gas phase. If you use solar gas-phase abundances and assume that grains exist then the code will complain but still perform the calculation.

PAH abundances: There is good observational evidence that PAHs only exist in the H^0 region (AGN3 Section 8.5). Observations of the Orion Bar suggest that they are destroyed in the H^+ region and coagulate into larger grains in molecular regions. By default the PAH abundance depends on the ratio $n(H^0)/n(H_{tot})$. Other dependencies on depth can be used by changing the routine called when the **`function`** keyword is used.

4.2. What is the cloud's density? Does it vary with depth?

The density of hydrogen is used to set the cloud density. The default is for the hydrogen density to be the constant value set by the **`hden`** command. Optional commands tell the code to assume constant pressure, include a magnetic field or turbulence in the pressure, or to vary the density with a function specified by the user. Most of these are discussed in Chapter 8 *Density Laws* of Part 1 of HAZY.

4.2.1. Commands normally used

hden specifies the log of the hydrogen density [cm^{-3}]. Constant density is the default. This includes hydrogen in all forms.

constant pressure The cloud will have constant pressure. The equation of state includes pressure terms from thermal gas motions, turbulent motions, a magnetic field, the nearly isotropic radiation pressure produced by trapped emission lines, and the outward push of the incident radiation field on the gas. The keyword **gas** says to keep the gas pressure, rather than the total pressure, constant. This is discussed in the *Optical Depths and Radiative Transfer* chapter in Part 1 of HAZY.

filling factor The gas is normally assumed to fully fill the available space. This command sets a filling factor *f*, the fraction of the volume that contains gas (AGN3 Section 5.9). The remainder of the volume is a vacuum.

magnetic field These are ignored by default. Magnetic fields will contribute to the total pressure, and, optionally, to the turbulent velocity field. The parameters specify the magnetic field at the illuminated face and the field geometry. Cyclotron cooling may become important if the temperature is high enough. All of this is discussed in the chapter *Thermal Solutions* of Part 1 of HAZY. The gas and field are assumed to be well coupled so the magnetic pressure terms are included in constant pressure models.

4.2.2. Heads up!

hden gives the ***total hydrogen density***, defined as

$$n(\mathrm{H}) = n\left(\mathrm{H}^0\right) + n\left(\mathrm{H}^+\right) + 2n\left(\mathrm{H}_2\right) + \sum_{other} n\left(\mathrm{H}_{other}\right) \quad [\mathrm{cm}^{-3}] \qquad (2)$$

where the sum includes H in other molecules and H$^-$. In nearly all cases hydrogen will be in one of the first three forms.

Turbulent and magnetic pressure can be included in the total pressure in a constant pressure cloud. Note that there pressure terms are normally far larger than the gas pressure in the local ISM.

5. The incident radiation field

Often the radiation field striking the cloud is its only energy source. The radiation field is specified by its shape, which describes how it depends on wavelength or frequency, and by its intensity or luminosity. The shape and intensity are usually specified with different commands. The chapter *Defining the Continuum* of Part 1 of HAZY gives an overview of how this is done. More than one continuum source can be included. The following sections describe how to specify both the shape and intensity.

5.1. The luminosity or intensity of the radiation field

The luminosity and intensity cases are described on page 11 above. There are many ways to specify the intensity or luminosity. These are described in the chapter *Continuum Luminosity* of Part 1 of HAZY. By default most luminosity or intensity commands specify the quantity integrated over hydrogen-ionizing energies. Most also have the **`range`** option, which allows this energy range to be changed.

5.1.1. Commands normally used

`Q(H)` specifies the number of hydrogen-ionizing photons emitted by the central object into 4π sr [s^{-1}] (AGN3 Section 2.1).

`phi(H)` is the intensity [cm^{-2} s^{-1}] equivalent of the **`Q(H)`** command. It sets $\varphi(\mathrm{H})$, the flux of hydrogen-ionizing photons striking the face of the cloud.

`luminosity` specifies the luminosity emitted by the central object into 4π sr [erg s^{-1}]. By default this is the luminosity in H^0-ionizing radiation.

`intensity` is the intensity [erg cm^{-2} s^{-1}] equivalent of the **`luminosity`** command. It gives the intensity of radiation striking the cloud face. Note that this "intensity" is 4π times larger than the true mean intensity *J*, which has units erg cm^{-2} s^{-1} sr^{-1}.

`ionization parameter` This sets the dimensionless ratio of densities of ionizing photons to hydrogen, $U \equiv \varphi(\mathrm{H})/c\,n(\mathrm{H}_{tot})$ (AGN3, equation 14.7, page 357). The number is the log of *U*. This can sometimes be useful since clouds with the same ionization parameter have similar levels of ionization and temperature. This is equivalent to an intensity command.

5.1.2. Heads up!

None yet.

5.2. The shape of the radiation field

The continuum shape can be interpolated from a table of points, specified as a fundamental form such as a blackbody or bremsstrahlung, or taken from a previous Cloudy calculation. Methods of setting the shape are described in the chapter *Continuum Shape* of Part 1 of HAZY.

The shape should be specified between the code's energy limits of 10 m (yes, meters) and 100 MeV, if possible. The code will complain but compute the model if the continuum is not specified over the full energy range.

The absolute values of the numbers giving the shape do not matter. The shape is renormalized to have the intensity set with an intensity command.

5.2.1. Commands normally used

`interpolate` sets the continuum shape by interpolating on a table giving pairs of frequency – intensity points. This is the most commonly used method of setting the continuum shape since the results of other calculations, such a stellar atmosphere, can be directly entered.

`CMB` adds the cosmic microwave background for any redshift.

`blackbody` specifies a blackbody. This command has a number of options that allow the intensity of the radiation field to be specified using blackbody relationships.

`background` is a simple estimate of the X-ray – UV background at any redshift. It includes the CMB.

`table` gives one of a set of built-in continua. Some examples include the AGN / starburst cosmic background at any redshift *z*, some stellar continua, and the local ISM galactic background.

`table AGN` enters the Mathews & Ferland (1987) quasar continuum.

`table HM96` employs the Haardt & Madau (1996) background at a range of redshifts.

`table ISM` is the local ISM background.

`extinguish` will extinguish the incident continuum by photoelectric absorption due to a column density of neutral hydrogen. This command is often used to remove hydrogen-ionizing radiation in a PDR calculation, in which is it assumed that an unmodeled H^+ region has extinguished much of the incident radiation field.

`cosmic ray background` will include galactic background cosmic rays. These are important when the calculation extends into molecular gas. The chemistry of the cold ISM is driven by a series of ion-molecule reactions that are initiated by cosmic-ray ionization. The required ions will not exist if no source of ionization is present and the chemistry network may collapse. The code will complain, but try to compute the model, if the calculation extends into cool regions without including background cosmic rays.

5.2.2. Heads up!

Some shape commands also specify an intensity. An example is **`table HM96`** – that command specifies both the shape and intensity of the quasar background at some redshift. Commands that set both are listed in subsection *Keeping shape and intensity commands together* of Part 1 of HAZY, which also describes a ***possible disaster***.

The shape of the ***incident continuum should be specified over the entire energy range*** considered by the code, 10 m to 100 MeV. An easy way to do this is to include, as a minimum, the cosmic microwave background and local ISM diffuse continuum.

The continuum is assumed to be completely reflected for frequencies smaller than the plasma frequency of the cloud. This often occurs for moderate densities due to the very low frequencies that are included in the continuum.

The ***cosmic microwave background should always be included*** with the **`CMB`** command. This is not done by default.

`table ISM` and **`cosmic rays background`** ***should probably be included*** for objects within our galaxy.

The ***quasar background continuum should probably also be included*** using either the **`background`** or **`table HM`** commands.

Pairs of intensity and shape commands should be kept together. They may be misinterpreted if they are not.

6. Other commands

6.1. Radiative transfer

All line-formation processes, including line trapping, collisional deexcitation, continuum pumping, and destruction by background opacities (see AGN3 Section 14.5), are included.

6.1.1. Commands normally used

`Case B` artificially sets the optical depths of hydrogen Lyman lines to very large values. Under some circumstances this will force the hydrogen recombination lines to their Case B intensities, the limit where all Lyman lines scatter often enough to be degraded into Lα and Balmer lines (AGN3, Section 4.2).

`Case B` is only intended for setting up "homework" problems or test cases and should never be used in a simulation of a real object. It can have unexpected effects. If the cloud does not contain dust then the mean intensity of Lα will become very large when this is used. This may result in unphysical photoionization rates for third or fourth-row elements.

This command must be included in homework-problem PDRs, in which the H^+ region is ignored. If the command were not included then a thin layer of ionized gas would be produced on the face of the PDR. This is produced by photoexcitation of H^0 by the continuum in the Lyman lines, followed by decay into the metastable 2*s* level, which is then photoionized by the Balmer continuum.

turbulence and **wind** These commands are discussed on page 13 above. They strongly affect line transfer by changing the line width and resulting opacities. The code assumes that only thermal motions broaden lines unless an additional component of motion is added with one of these commands.

6.1.2. Heads up!

The ***Case B command has many artificial side-effects***, especially when grains are not present. It should not be used except in special test cases.

6.2. The Fe II and H_2 model atoms

Large and complete models of Fe II and H_2 emission can be included. They are slow and are not used by default. Very simple approximations to their results are used when the complete models are not included. These simple approximations may or may not be good representations of the physics of the real atom or molecule.

The large Fe II or H_2 models are used when the **atom FeII** or **atom H2** commands are included (see the chapter *Optical Depths and Radiative Transfer* of Part 1 of HAZY). There are also special punch output options that allow the emission or absorption from these complex species to be saved and analyzed.

The large model of Fe II emission was part of Katya Verner's thesis and is described in Verner et al. (1999). AGN3 describes Fe II emission in Section 14.5. The simulations *feii*.in* in the test suites give examples of its use. The **punch FeII continuum** command is a convenient output option. Application to AGN are given in Baldwin et al. (2004).

The large model of H_2 was part of Gargi Shaw's thesis and is described in Shaw et al. (2005). AGN3 describes some properties of H_2 in Section 8.3 and appendix A6. The simulations *h2*.in* in the test suites give examples of its use. The **punch H2** command gives a number of convenient output options.

6.3. The optimize command

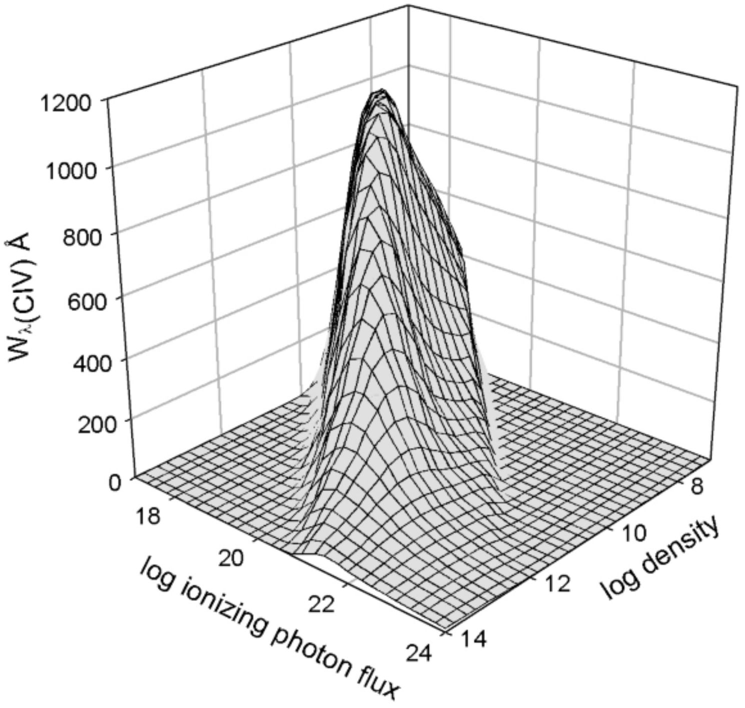
The **optimize** commands make it possible to solve the "inverse problem", probably the most common research area in astrophysics. This is when the outcome, perhaps an observed spectrum, is known and we wish to derive the conditions that created it. We know the "answer" and wish to find the "question".

A set of observed quantities are included in the input stream along with a series of **optimize** commands. The keyword **vary** says which parameters should be varied to try to reproduce the observed quantities. The code will then run a series of calculations, change the input parameters that include the **vary** option, and try to find parameters that match observations. All of this is described in the chapter *The Optimize Command* of Part 1 of HAZY. The simulations *optimize*.in* in the test suites show examples of its use.

It is generally a bad idea to write a paper that simply gives the result of an optimized model. This appears as a computational miracle with little pedagogical value. A better approach is to find the best-fitting parameters then show a series of calculations in which various parameters are changed around the best values. Changes in predicted quantities can then be shown. A discussion of physical cause of these changes will then motivate the final "best" model.

6.4. Grids of models

The greatest physical insight is often obtained from looking at results of grids of calculations to discover physical trends or correlations. An example is shown in this figure giving the equivalent width of one of the strongest emission lines in quasars as a function of two cloud parameters (Hamann & Ferland 1999).



The predicted equivalent width of C IV λ1549 as a function of cloud density and the flux of ionizing photons striking the cloud. See Hamann & Ferland (1999) and Ferland (2003) for further details.

The code is designed to be used as a sub-program of other, larger, codes. The command format is the same when it is used as a subprogram but commands are entered by calling a special routine that contains the command line as an argument. Cloudy is then executed and the predictions are retrieved after the simulation finishes. This method of running the code is described in the chapter *Cloudy as a Subroutine* in Part 3 of HAZY. The test suite directory includes a subdirectory *programs* that includes a set of sample programs that illustrate this use of the code.

The **`grid`** command, introduced by Ryan Porter in C07 (Porter et al. 2006), makes it easy to do such a calculation without writing new code. The predictions in the figure can be produced by modifying the simple BLR script given in Section 2.3 on page 8 above to read as follows:

```
table power law
phi(H) 18.5 vary
grid from 17 to 24 in 0.5 dex steps
hden 10 vary
grid from 7 to 14 in 0.5 dex steps
stop column density 22
punch line list "blr.line" no clobber "LineList.dat"
punch grid "blr.grd" no clobber
```

The **vary** keyword tells the code which parameters to vary. The **grid** commands specify the lower and upper limits to the ranges over which the previous parameter is to be varied and also give the step sizes. The parameters are varied by adding the step size to the initial value until it exceeds the final value. The grid is equally spaced in log space since the values on the **grid** command are logs. The actual value of the parameter given on the command with the **vary** option is not used but must be present to satisfy the command parser.

There are several output option designed for use with the **grid** command. The **no clobber** option tells the **punch** command to write the output from each model into the same punch file rather than overwriting the file (clobbering it) with each new simulation in the grid.

The **punch line list** command makes it possible to read in a list of lines from the file given in the second pair of quotes and save the predicted intensity into the first file. This is how the results shown in the figure were produced. The labels in the file must include both the four character string that is used in the output and the line wavelength. They must match exactly for the line to be recognized.

The **grid** command is described in the chapter *Miscellaneous commands* in Part 1 of HAZY and the **punch line list** command is described in the chapter *Controlling output* of Part 1.

6.5. Miscellaneous commands

6.5.1. Commands normally used

atom changes the treatment of some model atoms. It has many keywords. **atom H-like** and **atom He-like** allow some aspects of the H-like and He-like isoelectronic sequences to be changed. **H2** and **FeII** turns on large (and slow) models of H_2 (Shaw et al. 2005) and Fe II (Verner et al. 1999) emission (see the discussion starting on page 22 above).

init Frequently-used commands can be stored in an initialization file. The **init** command will include the contents of this file in the input stream. The file name appears on the command line within a pair of double quotes.

Comments These can be included within the input stream. The section *Introduction to Commands* of Part 1 of HAZY explains how to enter several types of comments.

6.5.2. Heads up!

Only one init file can occur within an input stream.

7. The code's predictions

7.1. The default printout

When the code is executed on a command line, as in

```
cloudy.exe < test.in > test.out
```

the file *test.in* contains the code's input commands (read from *stdin*) and its output (written to *stdout*) goes to *test.out*. The output is fully described in the chapter *Output* of Part 3 of HAZY. The default output includes a copy of the input commands, a list of the abundances of the chemical elements and grains, the physical conditions in the first and last zone, an explanation why the calculation stopped, the intensity or luminosity of the stronger emission lines, and the mean ionization, temperature, and column density of many species.

7.1.1. Commands normally used

`title` enters a title that is printed at various places.

`print` commands change some aspects of the printout. It can sort the emission lines, change their format, and modify what information is printed.

`print` line **`faint xx`** The predicted intensities of many lines are given in the main printout. Only the brighter lines are predicted since the total number of lines is vast. This command changes the limit to the faintest line to print.

`normalize` The log of the radiated luminosity or intensity of each emission line is printed after the line's label in the main printout. The intensity of the line relative to a normalization line is also given. In optical emission-line spectroscopy the normalization line is usually Hβ and this is the code's default. The **`normalize`** command can specify any other line to normalize the spectrum.

7.1.2. Heads up!

Understand why the calculation stopped. This is given in the first comments after the last zone and is described further in the introduction to the chapter *Stopping Criteria* of Part 1 of HAZY. An example of the last zone printout and the statement of the reason why the calculation stopped is shown in Section 3.5.2 on page 15 above.

7.2. Warnings, cautions, surprises, and notes

Examine the comments after the last zone for any warnings, cautions, or surprises. The code is designed to be autonomous and self-aware. It does many internal sanity checks to make sure that the calculation is valid (Ferland 2001). If there are problems the code will say so. W*arnings* start with "W-" and indicate that something is seriously wrong with the calculation. *Cautions* start with "C-" and indicate that the code is on thin ice. *Surprises* start with "!" and indicate novel or interesting aspects of the results. These are described further in the section *Warnings, Cautions, Surprises, and Notes* of Part 3 of HAZY.

7.3. Observed quantities

The chapter *Observed Quantities* of Part 3 of HAZY explains how to relate quantities predicted by the code to observed properties. The chapter *The Emission Lines* of Part 3 of HAZY gives an overview of the labels used to indicate various emission lines.

7.4. Punch output

A typical calculation generates far too much information for even a small part to be included in the main printout. Instead, a series of **punch**[3] commands are used to create extra files that contain various predictions. These are described in the chapter *Controlling Output* in Part 1 of HAZY. The punch information is written into a file whose name appears within double quotes on the command line.

7.4.1. Commands normally used

punch continuum gives the incident, transmitted, and total spectrum. The photon energy is given in Rydbergs by default but can be changed to keV, microns or Angstroms with the **units** option.

punch overview gives an overview of the calculation. This includes the electron temperature and density, the ionization of several elements, and abundances of some molecules, as a function of depth into the cloud.

punch element gives the abundance of ions and some molecules of a particular element as a function of depth into the cloud.

punch molecules gives the densities of a large number of molecular species as a function of depth into the cloud.

punch XSPEC The code can save its predictions in the FITS format used by the XSPEC X-Ray analysis code. The first application is Porter et al. (2006) and more information is in the section of Part 1 where the **punch XSPEC** command is described.

7.4.2. Heads up!

Emission lines appear in the output produced by the **punch continuum** command. The emission line to continuum contrast ratio depends on the size of

[3] The first generations of computers had large machines that "punched" information onto Hollerith cards. This was an important way to save information since disk drives were so small. These punch commands are in the spirit of those punch machines.

the continuum mesh because the continuum cells are too coarse to resolve the lines. See the section *Line to Continuum Contrast* in the chapter *Observed Quantities* of Part 3 of HAZY.

8. Example calculations

This section describes the code's test suite, a series of simulations that are used to automatically validate the code every time it is changed, and then goes over one model in detail.

8.1.1. The test suite

The test suite is a large group of simulations of various astrophysical environments. They are in the *tsuite* directory in the main download and are designed to exercise the code over its full range of validity. All have file names ending with "*.in*". The first part of the name indicates the type of model – for instance, all BLR models start with "*blr_*", all PDR models start with "*pdr_*", etc. The naming convention should be clear if you do a listing of all the files in the test suite directory (type "**`ls *.in`**" at the command prompt).

The test suite has three parts. ***auto*** contains over two hundred simulations and will run in about half a day on a modern workstation. This is run every night here in Lexington. ***Slow*** includes simulations with the large H_2 and Fe II atoms and takes several days to run. ***Programs*** show how to use the code as a subroutine of larger programs.

Use the Perl script *runall.pl* to run all simulations in the *auto* test suite. This is an important step in setting up the code since it confirms that you have a valid version. A code as large as Cloudy is likely to discover bugs in a compiler, especially when highly optimized code is produced.

Each input file contains **`assert`** commands that checks if the output gives the expected answer. If the predictions are wrong the code will print a string saying that an asserted quantity has been botched. **`Assert`** commands provide an automatic way to validate the code when it is installed and revalidate it every time it is changed. The script *checkall.pl* confirms that all asserted quantities have their expected value.

8.1.2. Use the test suite simulations as examples

The file *doc_tsuite.htm* within each directory contains a list of all the test cases along with a summary of what they do and why they are set up the way they are. You can get an idea of how to set up models by reviewing this file.

8.2. One of the models …

This section considers one of the models in the test suite, *orion_hii_pdr_pp.in*, in detail. A much longer discussion with more details is given in the chapter *Output* in Part 3 of HAZY. This simulates a plane-parallel molecular cloud with an H II region and PDR on its surface. Radiation from a nearby O star and galactic background cosmic rays are the only sources of heat and ionization. The calculation follows a ray of light into the cloud. The illuminated face is an H^+ region, an ionized layer with a temperature of $T \approx 10^4$ K. The PDR, a largely atomic region with a $T < 10^3$ K, occurs at a depth where ionizing radiation has been extinguished. The shielded face is a molecular cloud with $T < 100$ K. This calculation is described further in Ferland (2003).

The gas equation of state is the relationship between temperature and pressure. It includes gas pressure, the outward push caused by absorbed starlight, the magnetic, and turbulent pressure. Turbulence is assumed to be in equipartition with the magnetic field. The field is assumed to be well coupled to the gas.

Comments within the input script explain the purpose of the commands used to set up the simulation. This section discusses the various output files created by the calculation.

8.2.1. *run* orion_hii_pdr_pp.in

Run the *orion_hii_pdr_pp.in* script with the command

```
cloudy.exe < orion_hii_pdr_pp.in > orion_hii_pdr_pp.out
```

You will end up with many punch output files with names *orion_hii_pdr_pp.** and a main output file called *orion_hii_pdr_pp.out*.

8.2.2. *Examine the main output*

Examine the file *orion_hii_pdr_pp.out*. First, confirm that the calculation stopped for the intended reason. The reason is given after the last zone results, and, for this simulation, should be because the outer radius was reached.

The simulation may have had pressure convergence failures where the cloud passed through a *thermal front*. Thermal fronts, and the problems they cause, are described in the chapter *Problems* of Part 3 of HAZY. Convergence problems are announced with lines that begin with the string "PROBLEM".

Next, identify some of the strongest emission lines in the spectrum. These are listed towards the bottom of the output, following the string "`Emission Line Spectrum`". Two iterations were performed to converge the optical depth scale. Make sure that you are looking at the last iteration (similar information is printed for each iteration). The command **`print last`** would tell the code to only print results of the last iteration but is not used in this test. The emission-line intensities are given relative to Hα. If dust were not present the Lα / Hβ ratio would be about 34 (AGN3 Chapter 5). Actually it is predicted to be only about twice as strong as Hβ because Lα is efficiently absorbed by dust. [O III] λ5007 is one of the strongest lines in the spectrum, as expected for an H II region.

Two blocks of emission-line intensities are printed. The first block "`Intrinsic line intensities`" gives the total emission in all directions but does not include the effects of extinction due to the molecular cloud. This spectrum would be observed after correcting for reddening. The second block of lines "`Emergent line intensities`" gives the spectrum that emerges from the illuminated face of the cloud. Some fraction of each line is emitted towards the hemisphere containing the molecular cloud. The grain albedo is used to compute the fraction that is reflected back towards the illuminated face.

Some integrated properties of the cloud are listed towards the end of the file. Column densities of various constituents are given. The line with "`Log10 Column density (cm^-2)`" gives column densities of H^0, H^+, and H_2 – the cloud is predominantly molecular. Mean temperatures are also given following the line "`Log10 Mean Temperature (over volume)`"– the H^+ region has a mean temperature of nearly 10^4 K, the H^0 region has a mean temperature a bit under 10^3 K, and the H_2 region has a mean temperature of around 20 K. The output ends with a list of the asserted quantities – these compare the predictions of your executable with its historical predicted quantities.

The chapter *Output* in Part 3 of HAZY goes over the code's output in detail. Have a look.

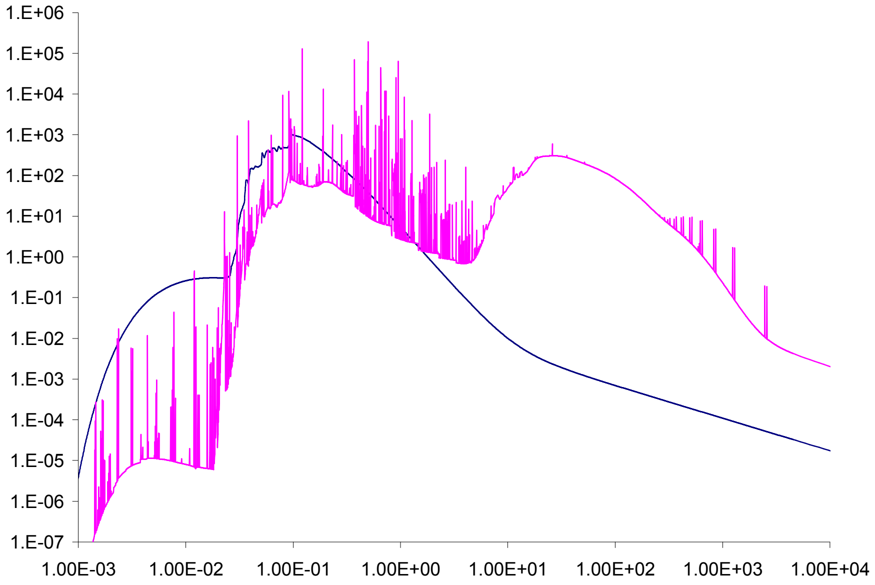
8.2.3. The punch output

The input script contains many **punch** commands. Some are only intended as debugging aids but others contain a wealth of physical information about the results of the simulation. The **punch** commands are described in the Chapter *Controlling Output* of Part 1 of HAZY while the chapter *Observed Quantities* of Part 3 of HAZY explains how to extract some observed quantities from all of this output.

The first line in a **punch** file gives a title for the columns of numbers that follow. In most files each line gives quantities for a single zone. The numbers are tab-delimited to make it easier to enter into a data base or plotting program. These tabs may appear confusing if viewed with an editor that is not aware of the tab settings. The following sections go over individual output files.

orion_hii_pdr_pp.con is produced by the **punch continuum** command and gives the incident, reflected, transmitted, and total continua. These are defined in the chapter *Definitions* of Part 1, and are described in both chapters *Controlling Output* of Part 1 and *Observed Quantities* of Part 3 of HAZY. The **units** option changes the energy scale from Rydbergs (the default) to microns. These wavelengths are the x-axis in the following figure. The plot shows the incident stellar continuum, listed in column two, as the smoother line. The total emitted continuum, contained in column seven, is the line with a great deal of structure. This is mainly the "reflected" spectrum, the continuum emergent from the illuminated face of the H^+

region. Extinction by dust in the molecular cloud prevents much light from emerging from the shielded face.

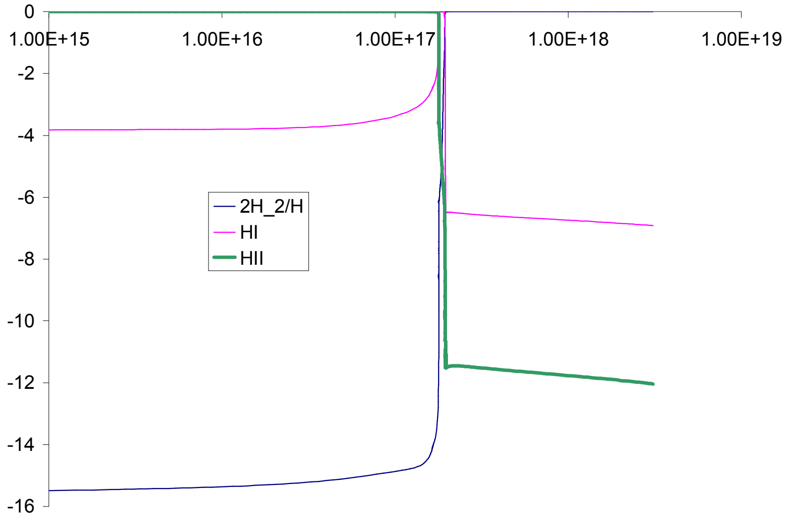


The incident (smooth) and emitted continua contained in the *orion_hii_pdr_pp.con* file. The x-axis is the wavelength in microns and the y-axis gives νf_ν [erg cm^{-2} s^{-1}]. This is produced by the **punch continuum** command

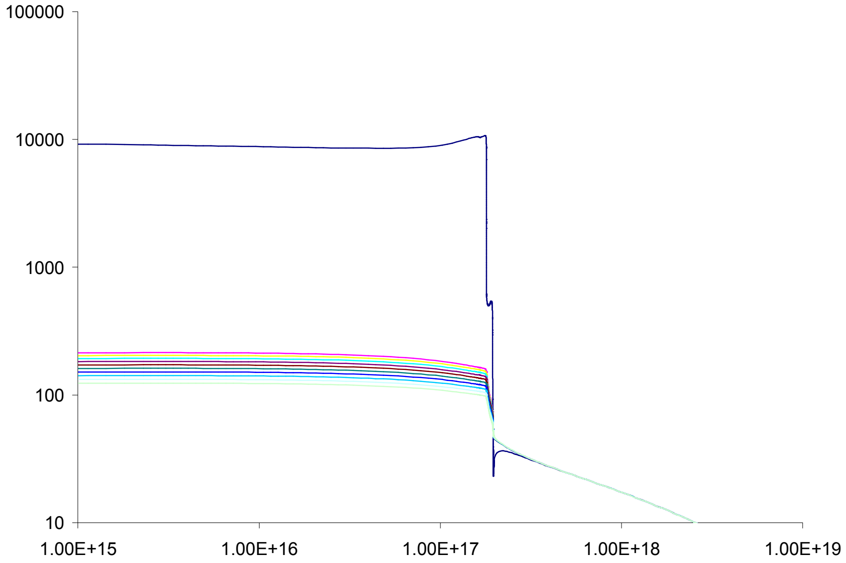
orion_hii_pdr_pp.ovr is produced by the **punch overview** command. It gives the electron temperature and density, the hydrogen density, the heating rate, and the ionization distribution of H, He, C, and O. The following figure shows the hydrogen ionization and chemical structure as given in this file. The x-axis gives the log of the depth into the cloud in cm. The y-axis gives the log of the fraction of hydrogen in the form of H^+, H^0, and H_2. The hydrogen ionization front occurs at a depth of ~ 2×10^{17} cm. There is a small H^0 region and the rest of the cloud is mostly H_2.

orion_hii_pdr_pp.grntem gives temperatures of grains as a function of depth. These, together with the electron temperature contained in the overview file, were used to create the following figure. The highest curve is the electron temperature and is ~10,000 K across the H^+ region, falling to ~500 K in the small H^0 region, then to below 100 K in the H_2 region.

The calculation includes graphitic and silicate size-resolved grains. Their temperatures are shown in the following figure as the lower cluster of curves. They have a range of temperatures ~100 – 200 K across the H^+ region and grow cooler in the molecular region. The gas and dust temperatures approach one another deep in the molecular cloud.



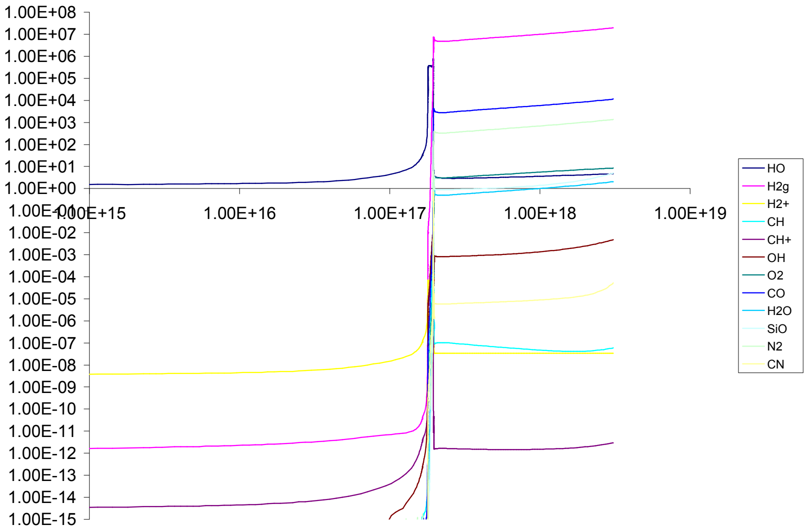
The hydrogen ionization structure. The x-axis is the depth in cm and the y-axis gives the log of the fraction of H in H^+, H^0, and H_2.



Temperatures of grains (the lower cluster of curves) and free electrons (the higher curve). They become nearly equal deep in the molecular cloud. The y-axis is the temperature and the x-axis is the depth into the cloud.

orion_hii_pdr_pp.mol gives molecular densities as a function of depth and was used to produce the following figure. It also includes several measures of extinction due to dust.

These are only some of the predictions that come from this calculation. Feel free to explore by changing parameters and assumptions and by adding other output options.



Densities [cm^{-3}] of some of the molecules included in the calculation are shown as a function of depth [cm] into the cloud.

8.3. Heads ups for classes of objects

Simulations of several types of astronomical objects are in the test suite. They can be grouped together by the first part of their filename. The following subsections describe important considerations for setting up each of these classes of simulations.

8.3.1. BLR of Active Nuclei

The density is high enough for free-free absorption of low-energy radiation to be a significant heating process. As a result the infrared continuum can have a surprising effect on the temperature of these clouds. Extrapolating a reasonable power-law continuum into the infrared may result in runaway free-free heating. This, and other practical aspects of BLR clouds, is discussed in Ferland (1999a) and in the last two chapters of AGN3.

8.3.2. NLR of Active Nuclei

Does dust exist in the ionized gas? Depletion patterns are not clear, as discussed by Ferguson et al. (1997). The NLR is discussed further in the last two chapters of AGN3.

8.3.3. PDRs of star-forming regions

It is critical that background cosmic rays, or a source of X-rays, be included if the simulation is to extend into a molecular cloud. The chemistry of cold interstellar matter is driven by a series of ion-molecule reactions (AGN3 Chapter 8 and Section 11.3). Chemical interactions between atoms and ions have smaller activation barriers than interactions between neutrals so chemistry can occur more quickly when ions are present. Ions will not exist if a source of ionization is not present and the chemistry network may collapse. Always include the **cosmic ray** and **background** commands.

The **double** command should be entered if the calculation stops before the outer edge of the molecular cloud is reached. See the discussion on page 15 above.

The culture in the PDR modeling community is to treat the PDR as an isolated phenomenon rather than an extension of the H II region. Nature does not do this, but you can force the code to do it by removing all hydrogen-ionizing radiation from the incident continuum. This is done with the **extinguish** command (see page 20 above).

If you do extinguish the hydrogen-ionizing radiation and start a PDR at that point you will usually find a thin layer of fairly highly-ionized hydrogen. This is caused by continuum pumping of the Lyman lines which then populates the metastable 2*s* level of hydrogen. That level is then ionized by the Balmer continuum or Lα. Continuum pumping of Lyman lines is called "Case C" in the ionized cloud community (see Ferland 1999b) and is described further in AGN3 Section 11.4. Classical PDR calculations do not consider this physics although it is included in Cloudy. This photoexcitation can be disabled by telling the code to assume that the Lyman lines are very optically thick and so are self-shielded. This is done with the **case B** command, described on page 21 above. The **case B** command should not be included in any realistic simulation, which should start with the H^+ region and extend into the PDR (as done by Abel et al, 2005).

8.3.4. Galactic nebulae

Cosmic rays and the ISM background should be included. Dust is almost certainly present in the ionized gas and should be included along with depleted abundances of the refractory elements. This is important since the grains extinguish the radiation field and their photoelectrons can heat the gas. Grains are included by selecting a mix of gas and solids that includes dust (with one of the **abundances** commands) or by explicitly including these (with a combination of the **grains**, **abundances**, and **depletion** commands).

9. References

Abel, N. P., Ferland, G. J., Shaw, G., & Hoof, van P. A. M., 2005, ApJS, 161, 65, available here

Baldwin, J. A., Ferland, G. J., Korista, K. T., Hamann, F. & LaCluyzé, A., 2004, ApJ, 615, 610 available here

Baldwin, Jack, Ferland, Gary, Korista, Kirk, & Verner, Dima 1995, ApJ, 455, L119 available here

Ferguson, J. W., Korista, K. T., Baldwin, J. A., & Ferland, G. J. 1997, ApJ, 487, 122 available here

Ferland, G. J. 1999a, in *Quasars and Cosmology*, ASP 162, p 147 ed G Ferland & J Baldwin (astro-ph/0307450) available here

Ferland, G.J. 1999b, PASP, 111, 1528 available here

Ferland, G.J. 2001, *Spectroscopic Challenges of Photoionized Plasmas*, ASP Conf 247, astro-ph/0210161, available here

Ferland, G.J. 2003, ARA&A, 41, 517, available here

Ferland, G. J., Korista, K.T., Verner, D.A., Ferguson, J.W., Kingdon, J.B. & Verner, E.M. 1998, PASP, 110, 761 available here

Haardt, Francesco, & Madau, Piero, 1996, ApJ, 461, 20

Hamann, F., & Ferland, G. J. 1999, ARAA, 37, 487 available here

Henney, W.J., Arthur, S.J., Williams, R.J.R., & Ferland, G.J. 2005, ApJ, 621, 328 (astro-ph/0501034) available here

Mathews, W. G., & Ferland, G. J. 1987, ApJ, 323, 456 available here

Osterbrock, D.E., & Ferland, G.J. 2006, *Astrophysics of Gaseous Nebulae and Active Galactic Nuclei*, 2nd edition (Mill Valley: University Science Books) (AGN3) available here

Porter, R.L., Ferland, G.J., Kraemer, S.B., Armentrout, B.K., Arnaud, K.A. & Turner, T.J. 2006, PASP, 118, 920 available here

Shaw, G., Ferland, G.J., Abel, N.P., Stancil, P.C., & van Hoof, P.A.M. 2005, ApJ, 624, 794, (astro-ph/0501485) available here

Verner, E.M., Verner, D.A., Korista, K.T., Ferguson, J.W., Hamann, F. & Ferland, G.J. 1999, ApJS 120, 101 available here