Hazy 1

a brief introduction
to Cloudy

### introduction and commands

version 06.01

G.J. Ferland

Department of Physics and Astronomy
University of Kentucky, Lexington
http://www.nublado.org
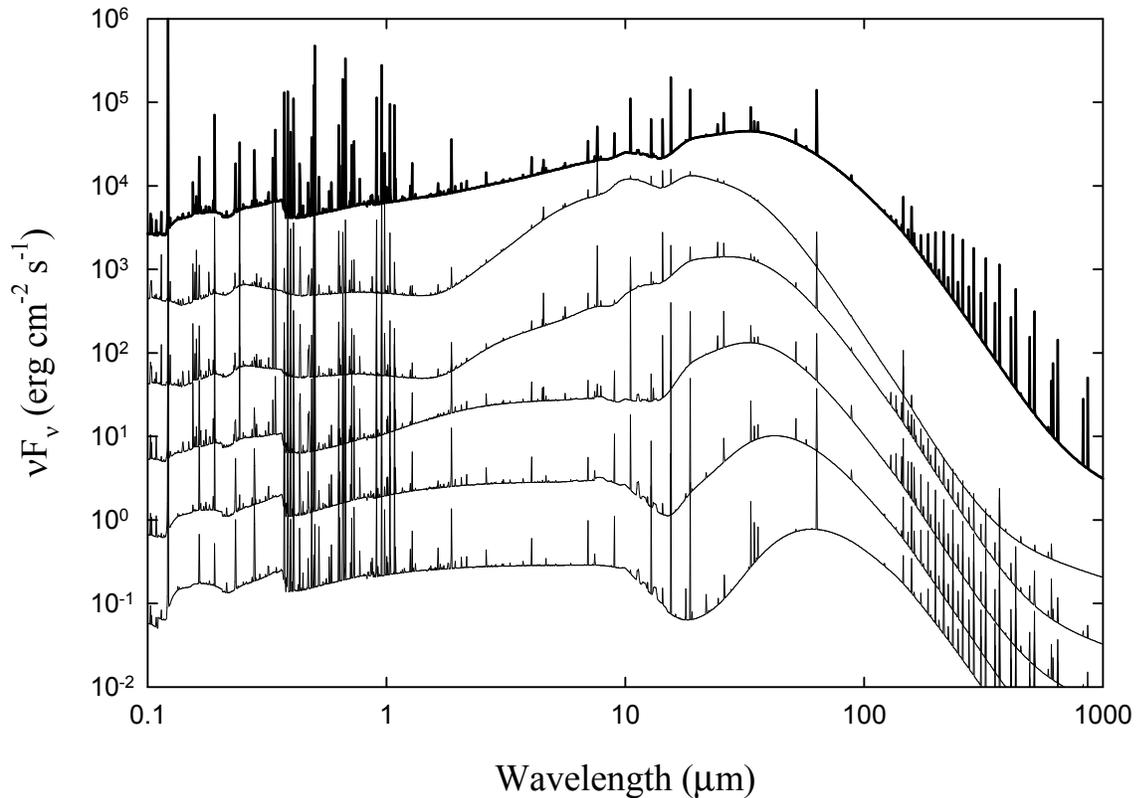
Emission from molecular clouds irradiated by an intense non-thermal continuum. Each continuum corresponds to a cloud located at a different distance from the central object with brighter emission corresponding to a smaller separation.

Use of this program is not restricted provided each use is acknowledged upon publication. The bibliographic reference to this version of Cloudy is "version xx.xx of the code last described by Ferland, G.J., et al 1998, PASP, 110, 761-778." The version number, shown here as "xx.xx", should be given and can be found on the first lines of the code's output.

Portions of this document have been published, and are copyrighted by, the American Astronomical Society, the Astronomical Society of the Pacific, and the Royal Astronomical Society. The remainder of this document, and the code Cloudy, are copyright 1978-2005 by Gary J. Ferland.

Cloudy is an evolving code. Updates are made on a roughly quarterly basis. You should confirm that you have the most recent version of the code by checking the web site http://www.nublado.org.

January 3, 2006.

Cloudy 06.01

G. J. Ferland

Department of Physics and Astronomy
University of Kentucky
Lexington

Table of Contents

List of Figures

List of Tables

1. INTRODUCTION

1.1. Overview

This describes the input, output, and assumptions for the spectral synthesis program Cloudy. It fully defines the commands used to drive the program. The methods, approximations, and assumptions used by Cloudy are outlined in Part 2 of this document, although this part, like Cloudy itself, is still under construction. Part 3 gives details about the program structure, its output, the computational environment, and how to call Cloudy as a sub-program of larger programs. The Quick Start guide summarizes these three volumes and gives a general overview of the commands.

Many environments are encountered in which dilute gas is heated and ionized by the radiation field of a central object. Under these circumstances it is possible to predict the physical conditions (that is, the distribution of ionization, density, and temperature) across the cloud, and its resulting emission-line spectrum, in a unique and self-consistent manner. This is done by simultaneously solving the equations of statistical and thermal equilibrium, equations that balance ionization-neutralization processes, and heating-cooling processes, respectively. Osterbrock & Ferland (2005, hereafter AGN3) describe the physics governing such environments. Davidson and Netzer (1979), Halpern and Grindlay (1980), Kallman and McCray (1982), Kwan and Krolik (1981), Wills, Netzer, and Wills (1985), Ferland and Shields (1985), Netzer (1990), and Ferland (2003) provide additional details of effects of high-energy radiation and line transfer.

1.2. What Cloudy can do

Cloudy is designed to simulate emission-line regions ranging from the intergalactic medium to the high-density LTE limit. The range of temperatures and densities are described in detail in a section in Part 2 of this document. It predicts the thermal, ionization, and chemical structure of a cloud and its spectrum.

1.3. Setting up Cloudy

Several steps are required to set up the code. Obtain the source and data files from the web site http://www.nublado.org Decompress the files, edit *path.c* to tell the system where the data files live, then compile the code. Download and compile the stellar atmosphere models if you want to use them. Finally, verify the behavior of the code by running the test suite. The detailed steps required to do this are described on the web site.

1.4. What must be specified

One powerful asset of photoionization analysis is the large number of observables resulting from only a few input parameters. Intensities of many millions of spectral

lines[1] are predicted by Cloudy. These result from the specification of only a) the shape and intensity of the external radiation field striking a cloud (referred to as the incident continuum), b) the chemical composition and grain content of the gas, and c) the geometry of the gas, including its radial extent and the dependence of density on radius. The following subsections describe the general philosophy of the specification of each.

1.4.1. Incident continuum

Both the shape and intensity of the external continuum striking the cloud must be specified. Often this is the region's only source of heat and ionization. Different commands are usually used to specify the continuum shape and intensity, although a few commands specify both.

1.4.1.1. Continuum shape

The shape of the continuum should be fully specified between an energy of 1.001×10^{-8} Ryd ($\lambda \approx 10$ m) and an energy of 7.354×10^6 Ryd ($h\nu \approx 100$ MeV) if possible. (In much of the following discussion photon energies will be given in Rydbergs. The ionization potential of hydrogen is nearly 1 Rydberg. See the discussion in Part 2 of this document for an exact definition, and how to convert the Rydberg to other units.) A physically motivated continuum spanning the full energy range should be specified, if possible. The continuum can be specified as a fundamental form (such as blackbody emission, optically thin bremsstrahlung emission, or a power law with optional exponential cutoff), interpolated from tables of points, or as the radiation field transmitted through a cloud, as predicted by previous calculations. Additionally, a set of built-in continua (for instance, emergent continua from many model atmospheres, the observed Crab Nebula continuum, or several typical AGN continua) can be specified as look-up tables.

1.4.1.2. Continuum intensity or luminosity

The intensity of the external continuum striking the cloud must be specified. This can be given either as a flux (energy or photon) per unit surface area of cloud or as luminosity (energy or photon) radiated by the central object into 4π sr. These can be set by specifying the flux of photons, a flux density, or luminosity, at arbitrary energies, or by giving the absolute visual or bolometric magnitude of the continuum source.

The code must be able to derive the flux of photons (cm^{-2} s^{-1}) striking the illuminated face of the cloud. If the continuum is specified as a surface flux (i.e., quantity striking a unit area of cloud) then the inner radius of the cloud does not need to be specified. If the inner radius is not specified then a plane parallel geometry will be assumed. A plane parallel geometry is simulated as a sphere with an inner radius of 10^{30} cm. The predicted emission-line spectrum will be given as

---

[1] These is no limit to the number of spectral lines that the code can compute because there is no limit to the number of levels of atoms of the H-like (H^0, He^+, etc) and He-like (He^0, Li^+, etc) isoelectronic sequences. Processor speed and memory limit the simulation to a few million lines in most applications with today's computers, however.

intensities (energy radiated per unit surface area of cloud). If the luminosity of the central source is given (photons or energy radiated into 4π sr) then the inner radius of the cloud *must* be specified, and emission-line luminosities will be predicted.

Finally, beware that the word *intensity* used here does not have the precise definition found in radiative transfer theory. In the standard literature both the specific and mean intensities *I* and *J* are defined per unit solid angle. In the following, the word "intensity" will be used to indicate the total energy arriving from or emitted into 4π sr. In this case the quantity is closest to $4\pi J$. I try to specify units whenever this distinction is important.

1.4.1.3. Combining several continua

Up to 100 continua of any form can be co-added. There must be exactly the same number of shape and luminosity specifications. The code will stop if there are not.

1.4.2. Chemical Composition

The program considers the lightest 30 elements in detail. Grains are not part of the default composition. All stages of ionization are treated, and all published charge exchange, radiative recombination, and dielectronic recombination processes are included as recombination mechanisms. Photoionization from valence and inner shells and many excited states, as well as collisional ionization by both thermal and supra-thermal electrons and charge transfer, are included as ionization mechanisms. The chemistry includes many molecules and will go to the fully molecular limit (H in H_2 and C or O in CO). The default composition is solar, several other standard mixtures can easily be specified, and an arbitrary composition can be entered.

1.4.3. Geometry

The geometry is always 1D spherical, but can be made effectively plane parallel by making the inner radius much larger than the thickness of the cloud. The default is for the gas to have constant density and to fully fill its volume, but other pressure laws and models with only part of the volume filled can be computed as well.

Cloudy normally assumes an open geometry, or one in which the gas has a very small covering factor (these terms are defined in Section 2 beginning on page 6 below). This can be changed with the **sphere** command (page 75 below), which sets the covering factor (defined on page 9 below) to a large enough value for continuous radiation escaping the cloud in the direction towards the central object to always interact with gas on the other side (a closed geometry). Line photons which cross the central hole interact with line-absorbing gas on the other side if **sphere static** is set, but do not interact (because of a Doppler shift due to expansion) if **sphere expanding** is set (this case is the default when **sphere** is specified).

1.4.4. Velocity Structure

Normally, Cloudy assumes only thermal broadening of lines, the absence of any sort of internal velocity structure, and that the gas covering factor is so small that photons escaping the computed ionization structure do not interact with other emitting gas (i.e., an open geometry is assumed).

These assumptions can be changed. A component of microturbulence can be added with the **turbulence** command (page 100 below). A dynamical flow can be computed with the **wind** command (page 81 below).

1.5. What is computed and printed

Cloudy is driven by a set of command lines. These normally reside in a small input file which the code reads. The code creates an output file as the model is computed. The commands are four-letter keywords (either upper or lower case) followed by free-format numbers that may be mixed with letters. When Cloudy is executed as a stand-alone program, the usual case, standard input (*stdin*) is read for input, and standard output (*stdout*) is used for output.

As an example, create a small file (say, called **simple.in**) containing the following lines:

```
title example input
hden 5
blackbody 50,000
ionization parameter -2
```

Suppose that the code has been compiled to create the executable *cloudy.exe*. Then, the model described by the parameters in this input file could be computed by

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

The output will be in the file *simple.out*.

It is also possible for a larger program to drive Cloudy directly by treating it as a subroutine. There are examples of doing this in Part 3 of this document and on the web site.

The program begins by echoing the input commands, except for comment lines beginning with a #, %, //, or c_ (a space after the leading c). Comments are ignored. The input stream ends with either a blank line or the end-of-file. Some properties of the incident radiation field, such as luminosity and number of photons in certain frequency ranges, are then printed.

Cloudy works by dividing a nebula into a set of thin concentric shells, referred to as zones. The zones are chosen to have thicknesses that are small enough for the physical conditions to be nearly constant within. Adaptive logic continuously adjusts the physical thicknesses of these shells to ensure this. Typically ~100 to 200 zones are computed in an optically thick model of an H II region. The physical conditions in the first and last zones are always printed and intermediate zones may be printed if needed (this is governed by the **print every** command described on page 124 below). The output for each zone begins with a line giving the zone number, its electron temperature, the distance from the center of the spherical nebula to the center of the zone, and some other properties of the solution. The next line gives the relative contributions of various emission lines to the radiation pressure, if this amounts to more than 5% of the gas pressure. The remaining lines give the relative populations of ionization stages of the other elements. Many details about the conditions within the zone are intermixed with these relative populations.

After the zone calculations are complete and the model is finished the code will explain why the calculation stopped. It is very important to confirm that the calculation stopped for the intended reason. Next, some warnings, cautions, or notes about the calculation may follow. The code is designed to be autonomous and self aware. This self checking will ensure that its range of validity is not exceeded. It will complain if it goes outside its range of validity, if it feels that some parameter has been incorrectly set, or that something surprising has happened during the calculation. This is an essential core feature of the code since it is now often used to generate grids of thousands of models, making it impossible to validate individual models by hand.

The final print out begins with a recapitulation of the entered commands, followed by the predicted emission-line spectrum. The first two columns of the emission-line spectrum give the ion and wavelength for each spectral line. The third column is the log of the luminosity or intensity of the emission line, and the last column gives its intensity relative to the reference line, which is usually Hβ (other emission lines can be chosen with the **normalize** command, page 120 below). The third column will be either the luminosity or intensity. The luminosity (energy radiated by a shell of gas covering Ω sr of the central object) is predicted if the continuum luminosity is specified as energy radiated into 4π sr. The line intensity $4\pi J$ (the energy emitted per square centimeter of the gas slab) is predicted if the incident continuum is specified as a flux. If the geometry is spherical, but the continuum is specified as a flux (per unit area of cloud), then the line intensities will be expressed per unit area at the inner radius. Only the strongest emission lines are printed; the relative intensity of the weakest line to print is adjusted with the **print faint** command (page 126 below).

Finally, the last page of the output gives some averages of the ionization fractions over the slab, the optical depths in various lines and continua, the intensity of the continuum emerging from the cloud, and other properties of the nebula.

1.6. Acknowledgments

2. DEFINITIONS

2.1. Overview

This section defines many of the quantities used by Cloudy. I try to follow standard notation, such as that used by Mihalas (1978) or AGN3. Part 2 of this document goes into many of these quantities in greater detail.

This document has the following typographic conventions; *filename*, *variable*, `command`, and *routine*.

2.2. Continua

Figure 1 shows several of the continua computed in the calculation.

2.2.1. Incident continuum

The *incident continuum* is the external continuum emitted by the central object, which strikes the illuminated face of the cloud. It is specified in the commands that establish the boundary conditions for the calculation. Often the incident continuum is the only energy source for the cloud. The incident continuum is diminished by extinction within the cloud.

2.2.2. Diffuse continuum

The *diffuse continuum* (often referred to as the diffuse radiation field or diffuse fields) is the radiation field emitted by gas and grains within the nebula. Examples include the Lyman, Balmer, or two-photon continua emitted by hydrogen. These fields are very nearly isotropic and can be significant sources of ionizing radiation under some circumstances.

The main difference between the calculation of a stellar atmosphere and a photoionized nebula is in the treatment of the diffuse fields. The gas albedo, which
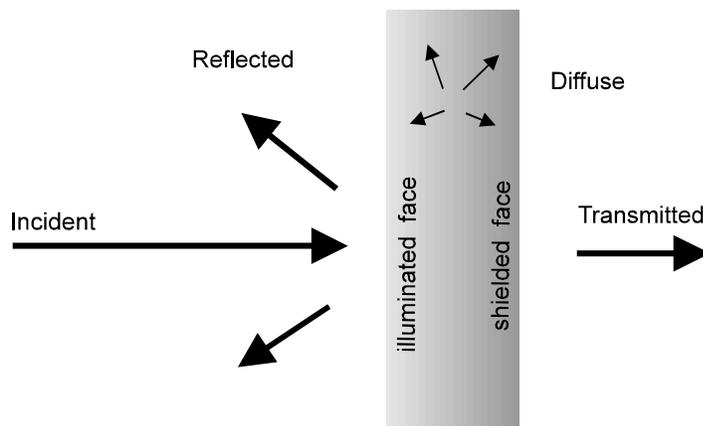
Figure 1 This figure illustrates several of the continua that enter in the calculations. continua

gives the relative probability of a photon being scattered vs absorbed, is generally small in a nebula. As a result, the diffuse fields must be far weaker than the attenuated incident continuum, since the second is usually the cloud's only real energy source. The radiation field is usually dominated by the incident continuum. By contrast in a stellar atmosphere the nearly isotropic diffuse field usually dominates the local intensity. As a result the diffuse fields can be treated by lower order approximations in a nebula than in a stellar atmosphere.

2.2.3. Transmitted continuum

The *transmitted continuum* is the net continuum emergent from the shielded face of the cloud. It includes both the attenuated incident continuum and the transferred diffuse continuum.

2.2.4. Reflected continuum

The *reflected continuum* is the continuum emitted from the illuminated face of the cloud back into the direction towards (i.e., within 2π sr of) the source of ionizing continuum. The reflected continuum is the result of both backscattered incident radiation and diffuse emission emitted from the cloud towards the source of ionizing radiation. This continuum is only computed for an *open geometry* (defined on page 9 below).

Figure 2 shows a plot of the incident and reflected continua for the Compton reflector in an AGN. This is a large column-density cloud, 10^{25} cm^{-2}, illuminated by a
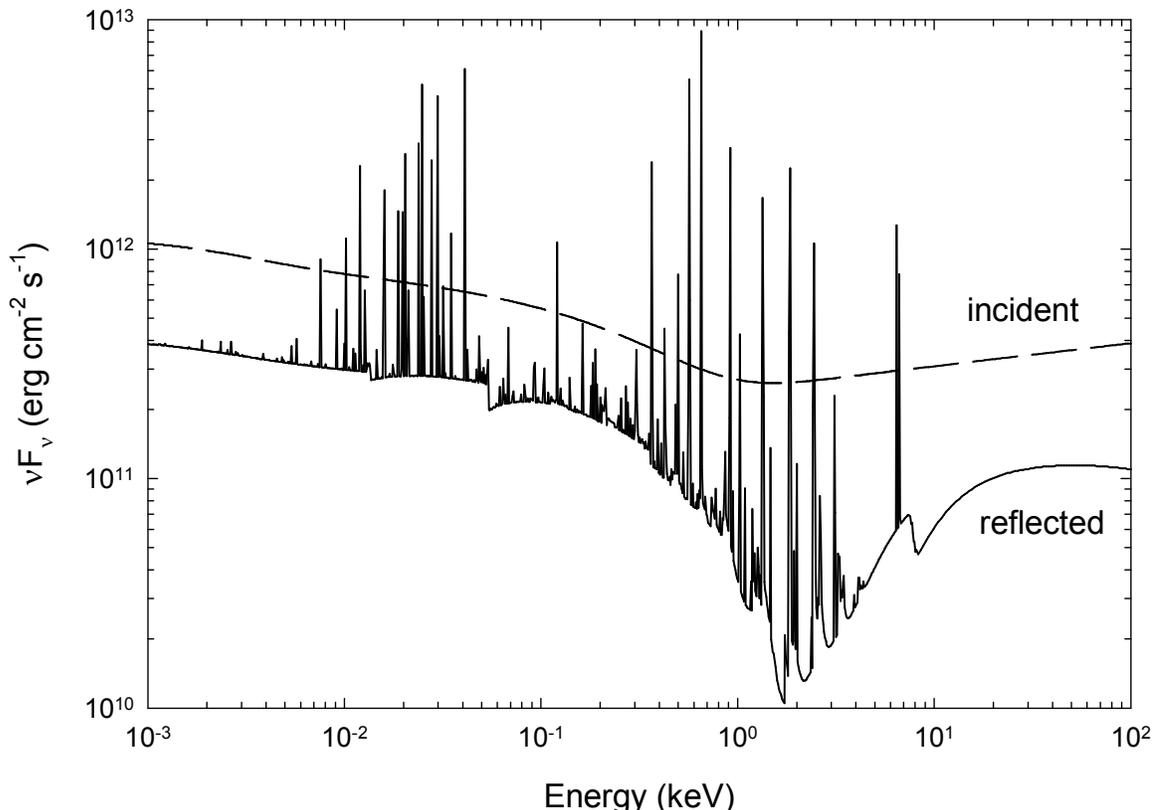
Figure 2 This figure shows the incident (dashed) and reflected (solid) continua, as computed by agn_blr_albedo.in in the test suite. reflected

$f_\nu \propto \nu^{-1}$ power law. The incident continuum is shown as a dashed line and the reflected continuum (obtained from the **punch continuum** command page 131 below). The Compton reflector's peak at X-ray energies is clearly shown.

2.3. Geometry

The geometry is always spherical, but can be changed to effectively plane parallel by making the inner radius much larger than the thickness of the cloud. In addition, it is possible to make the geometry nearly cylindrical, and a wind can be computed. The summary at the end of the calculation will say whether the geometry was plane parallel (the ratio of the thickness to the inner radius, $\Delta r/r_o < 0.1$), a thick shell ($\Delta r/r_o < 3$), or spherical ($\Delta r/r_o \geq 3$).

2.3.1. Illuminated and shielded faces of the cloud

The side of the cloud facing the source of ionizing radiation is the *illuminated face* of the cloud. The opposite side, in shadow, is referred to as the *shielded face* of the cloud. The illuminated face is generally hotter and more highly ionized than the shielded face.

2.3.2. Depth and radius

Figure 3 shows two possible geometries and some terms used to describe them. The *radius* is the distance from the center of symmetry, usually the center of the central object, to a given point. The *depth* is the distance between the illuminated face of the cloud and a point within the cloud. The *inner radius* is referred to as r_o, the *depth* is Δr, and the *current radius* is r. The depth or radius for a zone is the distance to the center of the zone.
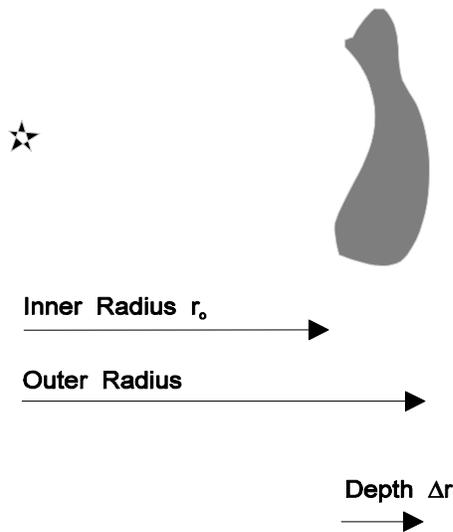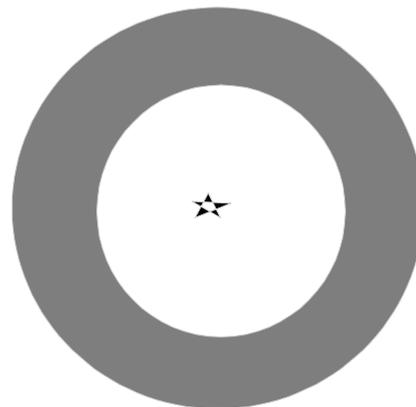
Figure 3 This figure shows the two limiting cases that can be assumed in the calculations. The star is the source of ionizing radiation and the shaded area represents nebular gas. An open geometry is the default, and a closed geometry will be computed if the **sphere** command is entered. geometry

2.3.3. Covering factor

The *covering factor* is the fraction of 4π sr covered by gas, as viewed from the central source of ionizing radiation. It is normally written as $\Omega/4\pi$ (AGN3 Section 5.9), has the limits $0 \leq \Omega/4\pi \leq 1$, and is the fraction of the radiation field emitted by the central object that actually strikes nebular gas. *Line luminosities* give the total energy emitted by a shell covering Ω sr while line *intensities* give the energy emitted per unit area of cloud. Line luminosities scale nearly linearly with increasing covering factor, while line intensities are only weakly dependent on it. A section of Part 2 goes over the two covering factors that actually enter into the calculations.

2.3.4. Filling factor

The *filling factor f(r)* accounts for clumping of emission-line gas. When a filling factor is set the hydrogen density is the density within regions containing gas, while surrounding regions are assumed to be a vacuum. The specific effects of a filling factor are described by Osterbrock and Flather (1959), AGN3 section 5.9, and page 78 below.

2.3.5. Hydrogen density

The *hydrogen density n*(H) is the total hydrogen density given by

$$n(\text{H}) = n(\text{H}^0) + n(\text{H}^+) + 2n(\text{H}_2) + \sum_{other} n(\text{H}_{other}) \; [\text{cm}^{-3}] \tag{1}$$

where $n(\text{H}_{other})$ represents H in all other hydrogen-bearing molecules.

2.3.6. Column densities

The *hydrogen column density N*(H) is given by

$$N(\text{H}) = \int n(\text{H}) f(r) \, dr \; [\text{cm}^{-2}] \tag{2}$$

where *f(r)* is the filling factor. I try to consistently use lower case *n* for a volume density (cm^{-3}) and an upper case *N* for a column density (cm^{-2}).

2.3.7. Open vs. closed geometry

Two limiting cases, referred to as *open* and *closed*, can be identified for the geometry and its influence upon the calculations. Figure 3 shows examples of both. The covering factor determines which is the best approximation. The choice mainly affects the transfer of the diffuse fields and has only second-order effects on predictions.

Open geometry. An *open* geometry is one in which the covering factor of the gas is small. All radiation that escapes from the illuminated face of the cloud, towards the source of continuous radiation, then escapes from the system without further interaction with gas. This is thought to be the case in, for example, the broad-line region of active nuclei or a blister H II region. In this case Lβ and higher hydrogen Lyman lines and ionizing radiation produced by recombinations can escape from the nebula. This geometry is the default and will be assumed if the **sphere** command is not specified.

Closed geometry.   Emission-line gas covers ~4π sr as seen by the central object in a *closed* geometry. If the central object is small relative to the nebula then all diffuse fields which escape from the illuminated face of the cloud in the direction towards the central object will go on to strike the far side of the nebula. This geometry is implicitly assumed in most calculations of planetary nebulae and H II regions. This geometry will be assumed if the `sphere` command is entered (page 80 below).

Static vs. expanding.   The `sphere` command has two optional arguments, `static` and `expanding`. This determines how emission-line photons which cross the central hole and strike the far side of the shell interact with the gas. The `static` option says to assume that the shell is stationary so that all lines interact across the nebula. In this case hydrogen Lyman line interaction should ensure that Case B is reached (AGN3 Section 4.2). If the nebula is expanding then the continuum photons that cross the central hole will interact with gas on the far side, but the expansion velocity of the shell ensures that diffuse line photons do not. In this case the `expanding` option should be set. This second case is the default when `sphere` is specified with no options.

These geometrical considerations (open vs closed, static vs expanding) make only second-order differences in the predicted emission-line spectrum, generally at the ≈10% level, largely because of the different treatments of the radiative transfer of the diffuse fields. If you are concerned about which geometry to use, try both, and compare the results. The differences are usually small.

2.3.8. Matter-bounded and radiation-bounded geometries

Two limiting cases for the hydrogen-ionization structure of a cloud exist.

Matter-bounded geometry. The nebula is said to be matter bounded if the outer limit to the emission-line region is marked by the outer edge of the cloud. In this case the cloud is ionized throughout and is optically thin to the incident continuum. In a matter-bounded cloud the intensity or luminosity of an optically thin recombination line is set by the product of volume and density (called the emission measure n^2V , AGN3, Section 8.1) and is not directly related to the luminosity of the ionizing continuum.

Radiation-bounded geometry. The nebula is said to be radiation bounded if the outer limit to the emission-line region is defined by a hydrogen ionization front, so both warm ionized and cold neutral regions exist. The H^+ region is optically thick to the hydrogen-ionizing continuum and has absorbed nearly all of it. In this case the intensity or luminosity of a recombination line is set by the luminosity of the ionizing continuum, with relatively little dependence on cloud properties.

2.3.9. Intensity & luminosity cases – is a starting radius necessary?

Cloudy must be able to deduce the surface flux of photons [cm^{-2} s^{-1}] at the illuminated face of the cloud. It is possible to specify the incident continuum as either a luminosity (energy or number of photons radiated by the central object into 4π sr), or as "intensity" or flux (the incident energy or photon flux per unit area at the illuminated face of the cloud). In the luminosity case it is also necessary to specify an

inner or starting radius and the emission lines will also be predicted as luminosities. In the intensity case a starting radius need not be specified, although one may be. Line intensities (energy emitted per unit area of cloud) will be predicted if the starting radius is not given. A default starting radius of 10^{30} cm will be assumed if one is not specified. This should result in a nearly plane-parallel geometry.

3. INTRODUCTION TO COMMANDS

3.1. Overview

This section introduces the commands that drive Cloudy. In following chapters they are grouped together by purpose, and individual commands are discussed after examples of their use. The section begins by outlining conditions that are assumed by default, and then goes on to discuss the various classes of commands (i.e., those that set the continuum shape, luminosity, or the geometry).

3.2. Command parsing routines

I try very hard to keep this document parallel with the code. In case of any confusion, please consult the original source. The commands are parsed by the series of routines that have names beginning with "*parse*". The list of routines can be seen by listing the files *parse*.c*. The second half of the name indicates the command that is parsed by that routine.

3.3. Default conditions

Cloudy is designed to be easy to use, so that a minimum number of commands are

Table 1
Default Conditions

Quantity	Value	Command	page
Default inner radius	10^{30} cm	radius	79
Default outer radius	10^{31} cm	radius	79
Highest allowed temperature	10^{10} K		
Stop calculation when T falls below this value	4000 K	stop temperature	118
Relative error in heating-cooling balance	0.005	set temperature convergence	195
Relative error in electron density	0.01	set eden convergence	195
Relative intensity of faintest line to print	10^{-3}	print line faint	126
Low-energy limit to continuum	1.001×10^{-8} Ryd		
High-energy limit to continuum	7.354×10^{6} Ryd		
Limiting number of zones	1400	set nend	190
Total hydrogen column density	10^{30} cm^{-2}	stop column density	112
H$^+$ column density	10^{30} cm^{-2}	stop column density	112
H^0 column density	10^{30} cm^{-2}	stop column density	112
Grains?	no grains	grains	60
Line to continuum contrast	1000 km/s	set PunchLWidth	192
Background cosmic rays	No	cosmic rays	104
Cosmic background	No	background	30

needed to drive it. The general philosophy is for a reasonable set of initial conditions to be assumed by default. These default conditions are summarized in Table 1, which also lists the commands that change each assumption.

The code is also designed to check that its assumptions are not violated. It should complain if problems occur, if its limits are exceeded, or if the input parameters are unphysical. It may print a series of warnings, cautions, or notes if some limit was exceeded or physical assumption violated.

3.4. Command format

Input and Output. When executed as a stand-alone program, Cloudy reads *stdin* for input and produces output on *stdout*. From a command prompt, this would be done as **cloudy.exe < input > output**. The code is also designed to be used as a subroutine of other, much larger, programs, often to generate large grids of simulations. In this case the input stream is entered using the subroutine calls described in a section of Part 3 of this document. In either case, this input stream must contain all the commands needed to drive the program. The command format is the same whether used as a stand-alone program or as a subroutine.

Command-line format. All commands are entered as free-format lines, beginning with a left-aligned four-character key word in columns 1 to 4. This keyword specifies the purpose of the command and is usually followed by one or more numbers or keywords. The keywords can be either lower or upper case. In the following examples the individual command keywords are shown extending beyond column 4. These extra characters are usually ignored.

The end of each command line is marked[2] by the end-of-line, a semi-colon ";", a pair of forward slashes "//", a sharp sign "#", or a percentage sign "%".

The command lines can be in any order and each can be up to 200 characters long[3]. The input stream ends with either a blank line or the end-of-file.

Units. Most commands use cgs units. In a few cases common astronomical nomenclature can be entered (i.e., the luminosity can be specified as erg s^{-1}, in solar units, or even magnitudes). This syntax varies from command to command, so it is important that the units be checked carefully.

Number of commands. Up to 4000 separate commands may be entered.

Output as input. Cloudy can also read its own output as an input stream. As described in the section "Output" in Part 3 of this document, the code echoes the input command lines as a header before the calculation begins. These lines are centered on the page and surrounded by asterisks. Sometimes a particular model will need to be recomputed. You can do this by making a copy of the printed

---

[2] Before version 92 a colon (":") could also mark an end of line. This character is needed to specify a path in the Windows environment and is no longer an end-of-line indicator.

[3] This limit is set by the length of the variable **INPUT_LINE_LENGTH** which occurs in a header file. Increase this variable and recompile the entire code if longer lines are needed.

command lines and using this copy as an input file. The input parser will handle removal of the leading spaces and asterisk. This is mainly a debugging aid.

3.4.1. Syntax used in this document

Sections describing each of the commands are introduced by examples of their use.

Square brackets indicate optional parameters. Optional parameters are shown surrounded by square brackets ("[" and "]"). The examples shown below use the format given in this document.

```
// following needs flux density, but frequency is optional
f(nu) = -12.456 [at .1824 Ryd]
//
// the luminosity command has several optional keywords
luminosity 38.3 [solar, range, linear]
//
// the phi(h) command has the range option
phi(h) = 12.867 [range ...]
```

These square brackets indicate only that the parameters are optional. The brackets should not be placed on the command line. They will be totally ignored if they occur.

Underscores indicate a space. Most commands and keywords require four character matches to be recognized. In some cases the leading or trailing character is a space, which is indicated by an underscore ("_"; an example is the keyword **_lte**). Other examples are shown below. The underscore must not be typed, only the space character. Only one space is needed between words.

The following is an example with the commands written as they are shown in this document:

```
// blackbody with T=50,000, in strict TE
blackbody 50,000 _lte
//
// use ISM radiation field
table _ism
```

The following is how the commands should actually be entered:

```
// blackbody with T=50,000, in strict TE
blackbody 50,000 lte
//
// use ISM radiation field
table ism
```

The space must occur where the underscore is written.

3.4.2. and, because nobody ever reads this document ....

The examples of the commands that appear below show the square brackets and underscores. Many people put these characters into the input stream because nobody ever reads documentation. As a service to the user, the command-line parser will usually replace any square brackets or underscores with the space character when the command lines are initially read in. The exception is any part of a line that is surrounded by double quotes. The string between double quotes is likely to be a file name and an underscore can occur in such a name.

<div align="center">

Table 2

Interpretation of Numerical Input

</div>

Typed Quantity	Interpreted as	Why
1,000	1000	Commas are removed before parsing
1e2	two numbers, 1 and 2	Exponentional notation is ignored
1e2,1e4	three numbers, 1 21 4	"e" ignored, "," is removed
1.2.3	two numbers 1.2 0.3	1.2 is parsed, then .3
100,3.141516	1 number 1003.141516	Comma is removed
.3 3.	0.3 and 3.0	

3.4.3. The continue option

It may not be possible to enter all the required values on a single line for the **interpolate** and **abundances** commands. In these two cases the original command line can be continued on following lines with a series of lines beginning with the keyword **continue**. The format on a **continue** line is unchanged. There is no limit to the number of **continue** lines that can be included, other than the limit of a total of 4000 input lines. The following is an example with the abundances command

```
abundances he =-1 li=-9 be=-11 b=-9 c=-4.3 n=-5 o=-2.3
continue f=-7 ne =-1.2 na =-3 mg =-8
continue al =-8 si =-8 p=-6 s=-8 cl=-9 ar =-8 k=-6
continue  ca =-8 sc=-9 ti=-7 v=-8 cr=-6.3 mn=-6 fe =-8
continue co =-9 ni =-8cu=-7 zn=-7
```

3.4.4. Numerical input

Numerical parameters are entered on the command line as free-format numbers. Exponential notation cannot be used. For instance, the entry "1E20" will be interpreted as the numbers 1 and 20, and no error message will result. Generally, Cloudy avoids exponential notation on input by entering numbers as logs, so 1E20 would usually be entered as 20.0. Commas can be freely embedded in input numbers and they are completely ignored. Numbers may be preceded or followed by characters to increase readability (i.e., T=1,000,000K, 1000000, and usually T=6, are equivalent, but T=1E6 is not. A period or full stop (".") by itself is interpreted as a character, not numeral or number.

Default values are often available. As an example, the **power law** command has three parameters, the last two being optional. The following are all acceptable (but not equivalent) forms of the command;

```
power law, slope-1.4, cutoffs at 9 Ryd and 0.01 Ryd
powe -1.0 5
power law, slope=-1.4 .
```

The last version uses the default cutoffs, i.e., none. If optional parameters are omitted they must be omitted from right to left; numbers must appear in the expected order.

Note that implicit negative signs (for instance, for the slope of the power law) *do not* occur in any of the following commands.

Table 2 lists how various typed inputs will be interpreted. The first column gives the typed quantity, the second its interpretation, and the third an explanation.

3.4.5. Comments

Comments may be entered among the input data in several ways. Comments can be entered at the end of a command line after a semi-colon (";"), double slash ("//"), a sharp sign ("#"), or a percentage sign ("%"). Anything on a line after one of these characters is completely ignored. This can be used to document parameters on a line. Any line beginning with a #, %, //, or a * is totally ignored; it is not even printed. A line beginning with c_ is ignored, but printed (note that there is a space after the c). There is also a **title** command, to enter a title for the simulation, as described on page 155 below.

3.4.6. Hidden commands

A command will be parsed and used by the code, but not printed in the output, if the keyword **hide** occurs somewhere on the command line. This provides a way to not print extensive sets of commands, like the **continue** option on the **continuum** command, or the **print off** command in an initialization file.

3.4.7. Log vs linear quantities

I have tried to keep the input quantities as logical as possible. Most quantities are entered as the log of the number, but some are linear. The following outlines some systematics of how these are entered.

Temperature. Cloudy will interpret a temperature as a log if the number is less than or equal to 10, and the linear temperature if it is greater than 10. Many commands have the optional keyword **linear** to force numbers below 10 to be interpreted as the linear temperature rather than the log.

Other parameters. The pattern for other quantities is not as clear as for the case of temperature. Often quantities are interpreted as logs if negative, but may be linear or logs if positive (depending on the command). Many commands have the keywords **_log** and **linear** to force one or the other interpretation to be used.

3.4.8. An example

Specific commands to set boundary conditions for a simulation are discussed in the following sections. As a minimum, the hydrogen density, continuum shape, continuum luminosity or intensity, and possibly the starting radius, must be specified to compute a model. As an example, a simple model of a planetary nebula could be computed by entering the following input stream.

```
title - this is the input stream for a planetary nebula
//
// set the temperature of the central star
black body, temp = 100,000K
//
// set the total luminosity of the central star
luminosity total 38 // log(L)- ergs/s
radius 17   // log of starting radius in cm
hden 4      // log of hydrogen density - cm^-3
sphere // this is a sphere with large covering factor
```

3.5. Filenames

It is sometimes necessary to read or write external files whose names are specified on a command line. Pairs of double quotes, as in "name.txt", surround file names.

The command parser first checks whether a quote occurs anywhere on the command line. If one does occur then the parser will search for a second pair of quotes and use whatever text lies between as a filename. The code will stop with an error condition if the second of the pair of quotes is not found or if the file cannot be opened for reading or writing.

3.6. The `init` command

This is a special command that tells the code to read a set of commands stored in an ancillary file. This allows frequently used commands to be stored in a single place. When the code is set up, one step is to specify the location of some needed data files by entering a path into the file *path.c*. The `init` command will automatically find initialization files that are located in this data directory, allowing them to be easily accessed from other directories. The `init` command is fully described in the section beginning on page 178 below. The installation instructions, which outline how to edit *path.c* to permanently set the path, are described on page 191 below.

There is no limit to the number of commands that can be in this initialization file, other than the total limit of 4000 command lines that is intrinsic to the code.

Any filename can be specified within a pair of double quotes, as in "ism.ini". The default name for the initialization file is *cloudy.ini*. The code will first search for the file in the local directory and then on the path, as set with the `path` command (described on page 191 below) or by editing *path.c*.

This command provides an easy way to change the default behavior of the code. For instance, many of the elements now included in Cloudy have negligible abundances and the code will run a bit faster if they are turned off with the `element off` command (page 57 below). Also, only about half of these elements were included before version 86 of the code. I normally keep a file called *c84.ini* in the Cloudy data directory, which will make the current version of the code behave more like version 84. My *c84.ini* file contains the following commands:

```
print off hide
elements read
helium
carbon
nitrogen
oxygen
neon
sodium
magnesium
aluminium
silicon
sulphur
argon
```

```
calcium
iron
nickel
end  of elements
element Lithium off
element Beryllium off
element Boron  off
element Fluorine  off
element Phosphor off
element Chlorine off
element Potassium off
element Scandium  off
element Titanium off
element Vanadium off
element Chromium off
element Manganese off
element Cobalt off
element Copper off
element Zinc  off
print on
```

The current version of the code would only include those elements present in version 84 if the command

```
init "c84.ini"
```

were entered.

A series of *.ini* files are included in the data file that comes with the Cloudy distribution. Do an "**ls *.ini**" within the data directory to list the available files. Comments at the start of the files describe their purpose.

4. DEFINING THE CONTINUUM

4.1. Overview

The incident continuum should be defined between the low-energy limit to the code, 1.001×10^{-8} Ryd and the high-energy limit of 7.354×10^6 Ryd. The net continuum striking the illuminated face of the cloud may be a fundamental form like a blackbody, the sum of many individual continua, or interpolated from a table of points.

4.2. Defining a single continuum

Two quantities, the *shape* and *intensity*, specify a single continuum. The intensity can be either the surface flux of ionizing photons striking the illuminated face of the cloud or both the continuum luminosity and the cloud's inner radius. The shape and intensity are specified independently in most cases, although some commands specify both (the command specifying the cosmic background is an example of the latter).

4.2.1. Continuum shape

The continuum shape can be set by interpolating on tables of points, read in from predictions of previous calculations, or by specifying fundamental forms such as blackbody, power law, or bremsstrahlung emission. Individual commands to specify the continuum shape are given in the chapter beginning on page 29 below.

4.2.2. Continuum intensity or luminosity

The intensity of the continuum at the illuminated face of the cloud can be specified as either an intensity (energy or photon flux per unit area) or by specifying *both* the luminosity *and* the inner radius (i.e., separation between the continuum source and the illuminated face) of the cloud. Individual commands to specify the continuum luminosity or intensity are given in the chapter beginning on page 21 below.

4.3. Combining several continua

4.3.1. The sum of several continuum

It is possible to combine up to 100 continua of any shape and intensity.[4] Cloudy will stop if more than 100 continua are entered. This limit is set by the variable *LIMSPC* that occurs in one of the included header files.

When more than one continuum is entered the series of luminosity and shape commands must be in the same order (i.e., map one to one). There must always be exactly the same number of continuum luminosity and shape specifications; Cloudy will stop if there are not.

---

[4]Restrictions on the number of tables that could be entered existed in Cloudy versions 73 and before, but have been lifted. Restrictions on which types of continua could be combined existed in Cloudy versions 67 and before, but have been lifted.

As an example, the following would be a rough approximation of an accretion disk and boundary layer around a white dwarf:

```
// this is the black body associated with the boundary layer
black body, temperature =500,000K
luminosity (total) 37.3
// the following is a rising power law, a simple approximation to the disk
power law, slope = 1.333, cutoff = 0.6 Ryd
luminosity (total) 37.2
```

The 500,000 K blackbody is given a total luminosity of $10^{37.3}$ erg s^{-1}, while the power-law continuum is given a total luminosity of $10^{37.2}$ erg s^{-1}.

4.3.2. *Keeping shape and intensity commands together*

It is not absolutely necessary to keep the ordered pairs of shape and intensity commands together, but this is a good practice since some commands (those given in Table 3) specify *both* the continuum shape and intensity. Problems arise if one of the commands giving both shape and intensity is entered between another pair of shape and intensity commands. For instance, the following will produce unintended results:

```
black body, temp = 500,000K
background, z=2
luminosity (total) 37
```

Table 3
Commands specifying
both shape *and* intensity
background
blackbody, energy density
blackbody, LTE
blackbody, luminosity
blackbody, radius
CMB
table Draine
table HM96
table ISM

because the **background** command enters both the shape and intensity of the cosmic background radiation field. In this example it comes after the **blackbody** command specifies a shape, but before the **luminosity** command specifies the luminosity of the blackbody. As a result the intensity implicitly entered by the **background** command will apply to the hot blackbody continuum rather than the cosmic background and the **luminosity** command will then incorrectly set the intensity of the cosmic background. This problem cannot occur if the shape and intensity commands are always kept together, as in the previous example. The code should produce a warning if shape and intensity commands are mixed together with a command that enters both.

4.4. Check the continuum!

It is important to check that the continuum has been entered correctly. First set up the commands to do a simulation and include the **punch continuum** command as described on page 131 below. Do the simulation and examine the file produced by the **punch continuum** command. The first column gives the photon energy and the second gives incident continuum (as νJ_ν, erg cm^{-2} s^{-1}) at the illuminated face of the cloud. Plot this continuum and check that it is correct.

5. CONTINUUM LUMINOSITY

5.1. Overview

All commands setting the intensity or luminosity of the continuum are defined in this section. The intensity of the incident continuum can be set by specifying a luminosity, the number of photons over an energy range, a flux density f_ν, or the absolute visual or bolometric magnitude.

5.2. Intensity vs luminosity commands

The incident continuum can be specified as an "intensity", the continuum incident onto a unit area of cloud, with units cm^{-2} s^{-1}, or a "luminosity", the continuum radiated by the central object into 4π sr, with units s^{-1}. Each of the following commands is characterized as either an intensity or luminosity command. The word "intensity" appears in quotes since it is not the I or J defined in radiative transfer texts, but rather the photon or energy equivalent of $4\pi J$.

The units of the predicted emission-line spectrum are determined by the units of the continuum. If the continuum is specified as a luminosity then the spectrum will be given as the luminosity radiated by a shell covering Ω sr of the central object (L_{line}, erg s^{-1}). Here Ω is the angular coverage of the nebula so that $\Omega/4\pi$ (with a default value of unity) is the covering factor (AGN3, Section 5.9, and page 77 below). If the continuum is specified as an intensity then the spectrum will be the energy radiated by a unit area of cloud into 4π sr ($4\pi J_{line}$, erg cm^{-2} s^{-1}).

5.2.1. Luminosity commands

If the continuum luminosity is set then the inner radius of the cloud *does* need to be specified, and emission-line luminosities are predicted. A covering factor will linearly change the luminosity of the entire spectrum, but will have only second order effects on relative intensities.

5.2.2. Intensity commands

If the continuum "intensity" is set then a starting radius *does not* need to be specified. If the starting radius is not specified then an inner radius of 10^{30} cm is assumed. A plane-parallel geometry usually results. The predicted emission-line spectrum is then given in intensity units. A starting radius may be specified and, if it is, then the resulting geometry may be spherical, plane parallel, or a thick shell, depending on the ratio of the outer to inner radii. Both absolute and relative intensities of lines have only second-order dependencies on the covering factor.

5.3. The range option

Many of the intensity/luminosity commands specify the number of photons or energy *in hydrogen-ionizing radiation* (1 Ryd $\leq h\nu \leq 7.354\times10^6$ Ryd). Other energy intervals can be specified with the **range** option, an optional keyword that can appear on most luminosity commands.

When the keyword **range** appears there are an additional two parameters, the low- and high-energy limits to the energy range in Rydbergs. These appear as the

second and third numbers on the line (the first number giving the intensity or luminosity). The position of the keyword **range** on the command line does not matter but the order of the numbers on the line does matter. If either of the range parameters is zero then the low (1.001×10^{-8} Ryd) or high (7.354×10^6 Ryd) energy limit of the continuum will be substituted for the zero value. If both energies are specified then the second number must be larger than the first, unless the second is zero, in which case it is the default high-energy limit of the code. If only one parameter appears then only the lower limit of the range will be changed, and the high-energy limit will be left at its default of 7.354×10^6 Ryd. If the first optional number is negative or the keyword **_log** appears then *both* of the extra numbers are interpreted as logs[5].

If **range total**, or simply **total**, appears with no parameters then the full energy range considered by the program, 1.001×10^{-8} Ryd to 7.354×10^6 Ryd, will be used. In this case the number is the total integrated intensity or luminosity.

The following shows examples of the range option for the **luminosity** command. By default, the **luminosity** command has a single parameter, the log of the luminosity [erg s^{-1}] in hydrogen-ionizing radiation (1 Ryd $\leq h\nu < 7.354 \times 10^6$ Ryd). The ";" symbol is used to terminate the line in one case.

```
// this will use the default range, only ionizing radiation
luminosity 38 ;the log of the luminosity in erg s^-1

// either will be the total luminosity
luminosity total 38
luminosity range total 33.4

// this will be the luminosity in visible light
luminosity 37.8 range .15 to .23 Ryd

// the luminosity in radiation more energetic than 0.1 Ryd
luminosity 38.1 range -1

// this will be the luminosity in non-ionizing radiation
luminosity 39.8 range 0 1
```

5.4. absolute [visual, bolometric] magnitude -2.3

It is possible to specify the integrated or monochromatic luminosity in "magnitudes", a quaint unit of historical interest. One of the keywords **bolometric** or **visual** must also appear. The absolute bolometric magnitude M_{bol} is related to the total luminosity by (Allen 1976, page 197)

$$L_{total} = 3.826 \times 10^{33} \times 10^{(4.75 - M_{bol})/2.5} \text{ [erg s}^{-1}] . \tag{3}$$

---

[5] The command parser replaces a zero with one of the energy bounds of the code before it checks for the log option and converts the entered numbers into linear energy bounds. This poses a problem if an energy bound of 1 Ryd is used with the **log** option, since **log range 0** will be interpreted as a energy bound given by dex of the actual energy range. Don't use the **log** option and an energy bound of 1 Ryd together.

The absolute visual magnitude M_V is approximately related to the monochromatic luminosity per octave at 5550 Å by (Allen 1976, page 197)

$$\nu L_\nu\left(5500\text{Å}\right) \approx 2.44\times10^{35}\times10^{-M_V/2.5} \quad [\text{erg s}^{-1}] \; . \tag{4}$$

The conversion between monochromatic luminosity per octave νL_ν and absolute visual magnitude M_V is approximate, with typical errors of roughly a percent. This is because Cloudy assumes that the *V* filter has an isophotal wavelength of 5550Å, and does not actually integrate over the incident continuum using a *V*-filter transmission function.

This is a luminosity command.

5.5. energy density 50,000K [linear]

This specifies the energy density [K] of the incident radiation field. The number is the equivalent energy-density temperature, defined as $T_u = \left(u/a\right)^{1/4}$ where u is the total energy density in all radiation [erg cm^{-3}] and a is the Stefan radiation-density constant. The number is interpreted as the temperature itself if it is greater than 10 and as the log of the number if it is less than or equal to 10. The optional keyword **linear** forces the number to always be interpreted as a linear temperature.

This is an intensity command.

5.6. f(nu) = -12.456 [at .1824 Ryd]

This specifies the monochromatic intensity at an arbitrary energy. The first number is the log of the monochromatic mean intensity at the illuminated face of the cloud, $4\pi J_\nu$ (with units erg s^{-1} Hz^{-1} cm^{-2}), where J_ν is the mean intensity of the incident continuum per unit solid angle.

The optional second number is the frequency in Rydbergs where $4\pi J_\nu$ is specified. The default is 1 Ryd and in the example above the continuum is specified at 0.1824 Ryd = 5000Å. The frequency can be any within the energy band considered by the code, presently 1.001×10^{-8} Ryd to 7.354×10^{6} Ryd. If the energy is less than or equal to zero then it is interpreted as the log of the energy in Rydbergs, and as the linear energy itself if positive.

This is an intensity command.

5.7. intensity 8.3 [range, linear]

This specifies the integrated intensity, $4\pi J$, [erg cm^{-2} s^{-1}] at the illuminated face of the cloud

$$4\pi J = \int_{\nu_1}^{\nu_2} 4\pi J_\nu \, d\nu \quad [\text{erg s}^{-1}\text{ cm}^{-2}] \; . \tag{5}$$

This is the per unit area equivalent of the **luminosity** command (see page 25 below). The number is the log of the intensity unless the optional keyword **linear** appears. Unlike the majority of the commands, the first five characters of the line must be entered.

The default range is over hydrogen-ionizing energies ($1 \text{ Ryd} \leq h\nu \leq 7.354 \times 10^6$ Ryd). The **range** option can be used to adjust the values of ν_1 and ν_2.

Some of the interstellar medium and photo-dissociation region (PDR) literature specifies the incident continuum in units of the Habing (1968) field (see, for instance, Tielens and Hollenbach 1985a, 1985b). This radiation field has an integrated intensity of 1.6×10^{-3} erg s^{-1} cm^{-2} between the limits of 6 and 13.6 eV (Tielens and Hollenbach 1985a; Hollenbach, Takahashi & Tielens 1991). This integrated intensity is sometimes referred to as G_o. The continuum described by Tielens and Hollenbach, but with an intensity of 1 G_o, could be generated with the commands:

```
// a B star with an intensity roughly the Habing 1968 radiation field
blackbody 30,000K
intensity -2.8, range 0.44 to 1 Ryd
extinguish by 24, leakage = 0
```

This set of commands sets the shape of the Balmer continuum to that of a hot blackbody, sets the intensity to the Habing value, and then extinguishes all hydrogen-ionizing radiation, as is assumed in the PDR literature.

This is an intensity command.

5.8. ionization parameter = -1.984

The ionization parameter is the dimensionless ratio of hydrogen-ionizing photon to total-hydrogen densities. It is defined as

$$U \equiv \frac{Q(\text{H})}{4\pi r_o^2 \, n(\text{H}) c} \equiv \frac{\Phi(\text{H})}{n(\text{H}) c} \ . \tag{6}$$

Here r_o is the separation [cm] between the center of the source of ionizing radiation and the illuminated face of the cloud, $n(\text{H})$ [cm^{-3}] is the total[6] hydrogen density (ionized, neutral, and molecular), c is the speed of light, $Q(\text{H})$ [s^{-1}] is the number of hydrogen-ionizing photons emitted by the central object, and $\Phi(\text{H})$ [cm^{-2} s^{-1}] is the surface flux of ionizing photons. The number is the log of U. The ionization parameter is a useful quantity in plane-parallel, low-density, constant-density, models, because of homology relations between models with different photon and gas densities but the same ionization parameter (see Davidson 1977).

This is an intensity command.

5.9. L(nu) = 38.456 [at .1824 Ryd]

This command sets the monochromatic luminosity L_ν [erg s^{-1} Hz^{-1}] of the central object. The first number is the log of L_ν. The optional second number is the frequency in Rydbergs where L_ν is specified. The default is 1 Ryd. In the example above the continuum is specified at 0.1824 Ryd = 5000 Å. The frequency can be any within the energy band considered by the code, presently 1.001×10^{-8} Ryd to 7.354×10^6

---

[6]Before version 65 of the code the electron density was used rather than the hydrogen density. Before version 75 $n(\text{H})$ was the atomic/ionic hydrogen density, and did not include molecules.

Ryd. If the energy is less than or equal to zero then it is interpreted as the log of the energy in Rydbergs, and the linear energy itself if positive.

This is a luminosity command.

5.10. luminosity 38.3 [solar, range, linear]

The number is the log of the integrated luminosity[7] emitted by the central object into 4π sr,

$$L = 4\pi\, R_{star}^2 \int_{\nu_1}^{\nu_2} \pi\, F_\nu\, d\nu \ \ [\text{erg s}^{-1}]. \tag{7}$$

The default range is over hydrogen-ionizing energies (1 Ryd $\leq h\nu \leq 7.354\times10^6$ Ryd). The **range** option can be used to adjust the values of ν_1 and ν_2.

The number is interpreted as the log of the *total* luminosity relative to the luminosity of the sun if the optional keyword **solar** appears. It is the log of the total luminosity if the keyword **total** appears. If the **linear** keyword is also used then the quantity will be the luminosity itself and not the log. The **range** option cannot be used if the luminosity is the total luminosity, or it is specified in solar units. It will be ignored if it appears.

The following are examples of the luminosity command.

```
// log of luminosity (erg/s) in ionizing radiation
luminosity 36

// roughly the Eddington limit for one solar mass
luminosity total 38

// both are a total luminosity 1000 times solar
luminosity solar 3
luminosity linear solar 1000

// this will be the luminosity in visible light
luminosity 37.8 range .15 to .23 Ryd
```

This is a luminosity command.

5.11. nuF(nu) = 13.456 [at .1824 Ryd]

This command specifies the log of the monochromatic mean intensity per octave $4\pi\nu J_\nu$ [erg s^{-1} cm^{-2}] at the illuminated face of the cloud, where J_ν is the mean intensity of the incident continuum.

The optional second number is the frequency in Rydbergs where $4\pi\nu J_\nu$ is specified. The default is 1 Ryd. In the example above the continuum is specified at 0.1824 Ryd = 5000Å. The frequency can be any within the energy band considered by the code, presently 1.001×10^{-8} Ryd to 7.354×10^6 Ryd. If the energy is less than or

---

[7]Before version 83 of the code, the luminosity command was used to enter either luminosity or intensity. The code decided between the two by checking on the resulting ionization parameter. There are now separate intensity and luminosity commands.

equal to zero, it is interpreted as the log of the energy in Rydbergs, and the linear energy if positive.

This is an intensity command.

5.12. nuL(nu) = 43.456 [at .1824 Ryd]

This command specifies the monochromatic luminosity per octave νL_ν [erg s^{-1}]. The first number is the log of the luminosity radiated by the central object into 4π sr. It can be expressed at an arbitrary photon energy but the default is 1 Ryd.

The optional second number is the frequency in Rydbergs where L_ν is specified. In the example above the continuum is specified at 0.1824 Ryd = 5000Å. The frequency can be any within the energy band considered by the code, presently 1.001×10^{-8} Ryd to 7.354×10^{6} Ryd. If the energy is less than or equal to zero, it is interpreted as the log of the energy in Rydbergs, and the linear energy if positive.

This is a luminosity command.

5.13. phi(H) = 12.867 [range ...]

This command specifies the log of $\Phi(H)$, the surface flux of hydrogen-ionizing photons [cm^{-2} s^{-1}] striking the illuminated face of the cloud. It is defined as

$$\Phi(H) \equiv \frac{Q(H)}{4\pi r_o^2} \equiv \frac{R_{star}^2}{r_o^2} \int_{\nu_1}^{\nu_2} \frac{\pi F_\nu}{h\nu} \, d\nu \ \ [\text{s}^{-1} \, \text{cm}^{-2}] \tag{8}$$

and is proportional to the optical depth in excited lines, such as the Balmer lines. The **range** option can be used to change the default energy range, given by the values of ν_1 and ν_2.

This is an intensity command.

5.14. Q(H) = 56.789 [range ...]

This is the log of the total number of ionizing photons emitted by the central object [s^{-1}]

$$Q(H) = 4\pi R_{star}^2 \int_{\nu_1}^{\nu_2} \frac{\pi F_\nu}{h\nu} \, d\nu \ \ . \tag{9}$$

The default energy range is 1 Ryd to 7.354×10^{6} Ryd and the **range** option can be used to change the energy bounds ν_1 and ν_2. The photon flux (the number of photons per unit area of cloud surface) can be specified with the **phi(h)** command, described on page 26 above[8].

This is a luminosity command.

---

[8]Before version 83 of the code the phi(H) and Q(H) commands were the same. The code decided which was specified by checking the order of magnitude of the resulting ionization parameter. These are now two different commands.

5.15. ratio -3.4 0.3645 Ryd to 147 Ryd [alphaox, _log]

This specifies the intensity of a second continuum source (referred to as the *current* continuum source) relative to the intensity of the *previous* continuum source. The ratio of the intensities $4\pi J_\nu$ (erg cm^{-2} s^{-1} Hz^{-1}) of the current to the previous continuum source is given by the first number on the command line. It is assumed to be the linear ratio unless it is less than or equal to zero, in which case it is interpreted as a log. If the keyword **_log** appears then the positive number is interpreted as the log of the ratio.

The second parameter is the energy in Rydbergs where the previous continuum source is evaluated, and the optional third parameter is the energy where the current continuum is evaluated. If the second energy is not entered then the same energy is used for both. The following is an example of using the **ratio** command to simulate the continuum of a typical quasar.

```
blackbody 50,000    ;the big blue bump
ionization parameter -2; its ionization parameter
table power law     ;an alpha =-1 power law
c now set intensity of power law relative to bump at 1 Ryd
ratio 0.001 at 1 Ryd;
```

This command was introduced to provide a mechanism to specify the optical to X-ray spectral index α_{ox}. This is defined here as in Zamorani et al. (1981), except for a difference in sign convention. Here α_{ox} is the spectral index which would describe the continuum between 2 keV (147 Ryd) and 2500Å (0.3645 Ryd) if the continuum could be described as a single power-law, that is,

$$\frac{f_\nu(2\,\mathrm{keV})}{f_\nu(2500\,\text{Å})} = \left(\frac{\nu_{2\,\mathrm{keV}}}{\nu_{2500\,\text{Å}}}\right)^{\alpha_{ox}} = 403.3^{\alpha_{ox}} \quad . \tag{10}$$

The definition of α_{ox} used here is slightly different from that of Zamorani et al. since implicit negative signs are *never* used by Cloudy. Typical AGN have $\alpha_{ox} \sim$ -1.4 . If X-rays are not present then $\alpha_{ox} = 0$.

The **ratio** command has an optional keyword, **alphaox**, which allows α_{ox} to be specified directly. If the keyword appears then only one parameter is read, the value of α_{ox}. A generic AGN continuum could be produced with the following,

```
blackbody 50,000    ;the big blue bump
ionization parameter -2
table power law     ;an alpha =-1 power law
ratio alphaox -1.4  ; set alpha_ox of the X-ray and UV continua
```

Note that α_{ox} may (or may not) depend on the luminosity of the quasar, as described by Avni and Tananbaum (1986). The solid line in their Figure 8 corresponds to

$$\alpha_{ox} = -1.32 - 0.088 \times \log\left(\frac{L_o}{10^{28}\,\mathrm{erg\,s^{-1}\,Hz^{-1}}}\right) \tag{11}$$

where they define L_o as the monochromatic optical luminosity at 2500Å in the source rest frame, and assume $H_0 = 50$ and $q_0 = 0$. Other fits are given by Worral et al. (1987):

$$\alpha_{ox} = -1.11 - 0.111 \times \log\left(\frac{L_o}{10^{27} \text{ erg s}^{-1} \text{ Hz}^{-1}}\right) \tag{12}$$

and by Wilkes et al (1994):

$$\alpha_{ox} = -1.53 - 0.11 \times \log\left(\frac{L_o}{10^{30.5} \text{ erg s}^{-1} \text{ Hz}^{-1}}\right) \quad . \tag{13}$$

However, LaFranca et al (1995) find no dependence of α_{ox} on luminosity. Avni, Worrall, and Morgan (1995) find a complicated luminosity dependence. Clearly this is an area of active research.

N.B. The net continuum may have a smaller than specified ratio of current to total continuum, since the command specifies the ratio of the current to the previous continuum, not the ratio of current to total continuum. The ionization parameter will be slightly larger than specified for the same reason.

In general, it is probably better to use the **AGN** command (described on page 29 below), rather than this command.

This is neither a luminosity nor intensity command — the units of the previous continuum carry over to this command.

5.16. xi –0.1

Tarter, Tucker, & Salpeter (1969) defined an ionization parameter ξ given by

$$\xi = \frac{L_{ion}}{n(\text{H})r^2} = (4\pi)^2 \int_{1R}^{1000R} J_\nu \, d\nu \ / n(\text{H}) \ [\text{erg cm s}^{-1}] \tag{14}$$

where $n(\text{H})$ is the hydrogen density at the illuminated face of the cloud and r is the source – cloud separation. L_{ion} is the luminosity between 1 and 10^3 Ryd. The number is the log of ξ. Only the first two characters on the line are checked to match **xi**, unlike the majority of commands where the first four characters on the line must match.

6. CONTINUUM SHAPE

6.1. Overview

The continuum shape should be specified between the energies of $1.001{\times}10^{-8}$ Ryd ($\lambda \approx 10$ m) and 100 MeV $\approx 7.354{\times}10^{6}$ Ryd. The low-energy continuum is important for Compton cooling, photoionization from excited states of the elements, free-free heating, H^- heating, and grain heating. The high-energy continuum is important for Auger and secondary ionization, Compton heating, and pair production. Energies greater than 100 MeV are not generally important since the Klein - Nishina electron-scattering cross section is small. Cloudy will complain, but compute the model if possible, if the continuum is not specified over the full energy range. An intensity of zero will be assumed for missing portions of the continuum.

The plasma frequency, given by

$$\nu_{pl} = \left(\frac{n_e q_e^2}{\pi\, m_e}\right)^{1/2} = 8.978{\times}10^3\, n_e^{1/2}\ s^{-1} = 2.729{\times}10^{-12}\, n_e^{1/2}\ \text{Ryd} \qquad (15)$$

will move into the energy range considered by the code if the electron density is higher than $\sim 10^7$ cm^{-3}. The incident continuum is set to zero below the plasma frequency and is totally reflected. The code will generate a comment if the plasma frequency occurs within the energy grid.

It is a good idea to plot the continuum using the output of the **punch continuum** command (page 131 below).

6.2. AGN T =150,000k, a(ox) = -1.4, a(uv)=-0.5 a(x)=-1

This command will produce a multi-component continuum similar to that observed in typical Active Galactic Nucleus (AGN). An example is shown in Figure 4. The "Big Bump" component is a rising power law with a high-energy exponential cutoff. It is parameterized by the temperature of the bump, the first argument on the command line. It is interpreted as the log of the temperature if it is less than or equal to 10 and the linear temperature otherwise. The second parameter is the X-ray to UV ratio α_{ox} (see the discussion of α_{ox} beginning on page 27 above). Note that there is no implicit negative sign in this
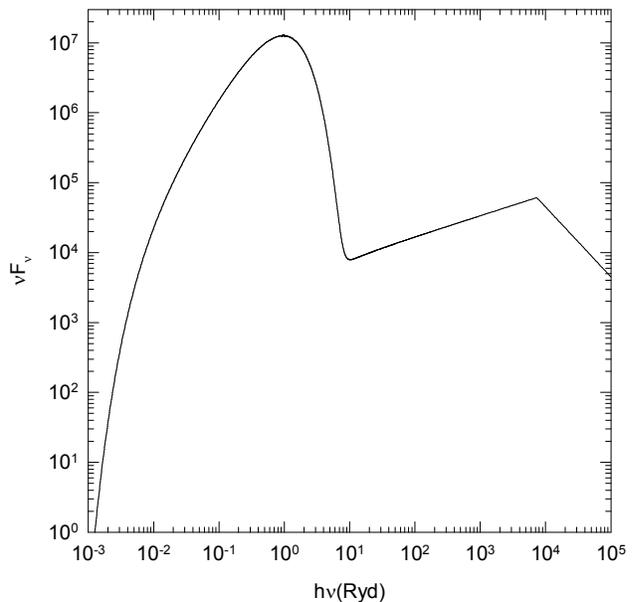
Figure 4 The continua produced by the AGN continuum command. The Big Bump peaks at 1 Ryd, while the X-ray power law dominates high energies. The two are normalized by the second parameter, the value of α_{ox}. agncon

exponent; typical AGN have $\alpha_{ox} \sim$ -1.4, (Zamorani et al. 1981). The third (optional) argument is the low-energy slope of the Big Bump continuum, with the default $\alpha_{uv} =$ -0.5 (Elvis et al. 1994; Francis 1993). The last argument is the slope of the X-ray component with the default $\alpha_x = $ -1. Optional parameters can be omitted from right to left.

The full continuum is the sum of two components, as in equation 16:

$$f_\nu = \nu^{\alpha_{uv}} \exp\left(-h\nu / kT_{BB}\right)\exp\left(-kT_{IR} / h\nu\right) + a\nu^{\alpha_x} \ . \tag{16}$$

The coefficient a is adjusted to produce the correct α_{ox} for the case where the Big Bump does not contribute to the emission at 2 keV. If the bump is very hot then it may contribute to the X-rays as well, and the resulting continuum will have a more negative α_{ox} than specified. The X-ray power law is only added for energies greater than 0.1 Ryd to prevent it from extending into the infrared, where a power law of this slope would produce *very* strong free-free heating. The Big Bump component is assumed to have an infrared exponential cutoff at $kT_{IR} = 0.01$ Ryd. Because of this cutoff the incident continuum will have zero intensity at very long wavelengths. This will cause the code to complain since a zero incident continuum is unphysical, but the model will be computed. The problem of a non-positive FIR continuum can be overcome by including the cosmic background with the **background** command (described on page 30 below), or with the **CMB** command (described on page 33 below).

The last term in equation 16 is not extrapolated below 1.36 eV or above 100 keV. Below 1.36 eV the last term is simply set to zero (the bump dominates for these energies). Above 100 keV the continuum is assumed to fall off as ν^3.

We used this command to generate the continuum used in a large atlas of BLR line intensities (Korista et al. 1997). The specific parameters needed to reproduce that continuum are **AGN 6.00 -1.40 -0.50 -1.0**. Only Kirk Korista can remember all these numbers, and the parameters were not explicitly given in the original paper in the format used by the code, so the **kirk** option on the **AGN** command will generate that continuum. It is fairly similar to the Mathews & Ferland (1987) continuum, but there is much greater flexibility in changing its details.

6.3. background, z=1.825, [f=100; no CMB]

This command will specify a continuum shape and intensity chosen to mimic the cosmic radio to X-ray background, as described by Ostriker and Ikeuchi (1983), Ikeuchi and Ostriker (1986), and Vedel, Hessten, & Sommer-Larsen (1994). Their ultraviolet continuum shape is an $\alpha =$ -1 power-law, with a mean intensity J_ν at 912Å given by

$$4\pi J_\nu\left(912\text{Å}\right) = 4\pi \times 10^{-21}\left(\frac{1+z}{3.5}\right)^4 \ f \ \left[\text{erg Hz}^{-1} \text{ cm}^{-2} \text{ s}^{-1}\right] \tag{17}$$

where z is the redshift and f an optional scale factor entered as the second parameter. Its default value is $f = 1$, and $z = 0$ (i.e., now) is assumed if no redshift is entered. Judging from Bechtold et al. (1987), Bajtlik, Duncan, and Ostriker (1988), and Vedel,

Hessten, & Sommer-Larsen (1994), *f* is confidently known to be within an order of magnitude of unity.

This command specifies *both* the shape and intensity of the continuum. It is important that any previously occurring ordered pairs of shape and intensity commands be complete before this command is given.

Cosmic microwave background radiation is included in the generated background. This radiation field is assumed to be a blackbody radiation field, in strict thermodynamic equilibrium, with temperature given by

$$T_{CMB} = T_o(1+z) \quad [\text{K}] \tag{18}$$

where the redshift dependence is from Peebles (1971) and the present temperature of the background is assumed to be T_o = 2.725 ± 0.002K (Mather et al. 1999; Wilkinson 1987). This background can be an important source of Compton cooling for low-density clouds. If the optional keyword **no CMB** appears on the line then the background due to the CMB will not be included.

Thermal background radiation can also be specified independently with the **CMB** command, described on page 33 below.

A starting radius of 10^{30} cm will be assumed if one is not specified. Some objects, such as Lα forest clouds, may be very large. Note that for the geometry to be plane parallel it is necessary to make sure that $\delta r/r \ll 1$ since, if $\delta r/r \geq 1$, the incident continuum will be attenuated by the r^2 geometric factor. It may be necessary to specify a larger starting radius if a plane parallel slab with thickness greater than 10^{30} cm is desired.

The **table HM96** command (see page 39 below) uses a more sophisticated form of the energetic continuum, but only at a redshift of z = 2.

6.4. blackbody t=100,000 [linear, _log, luminosity]

The continuum will be a blackbody with temperature (K) given by the number. The temperature may be entered directly or as a log. The number is assumed to be a log if it is less than or equal to 10 and linear if greater than 10. The keywords **_log** and **linear** will over ride this default and force the interpretation of the numbers to be either a log or linear. Embedded commas can improve readability, such as

```
black body, Temp = 1,000,000K
```

which is equivalent to

```
black 1000000
```

or

```
black body t=6   .
```

6.4.1. Peter Martin's blackbody luminosity options

The luminosity of the black body can be specified with command-line options. (P.G. Martin added these options.) If the luminosity is specified with any of these options then it must not also be specified with another luminosity command for this

continuum source. The keywords that can appear on the line are given in the following subsections.

6.4.2. blackbody 5, luminosity = 38.

If the keyword **luminosity** appears then the second number is the log of the *total* luminosity (erg s^{-1}) of the black body, $4\pi R_{star}^2 \sigma T_{eff}^4$. This example would be a 10^5 K planetary nebula nucleus at the Eddington limit.

This is a luminosity command.

6.4.3. blackbody 5, radius = 10.

The log of the radius (in cm) of the blackbody R_{star} is used to set the total luminosity when the keyword **radius** appears. The total luminosity is $4\pi R_{star}^2 \sigma T_{eff}^4$. This example is also typical of a planetary nebula nucleus.

This is a luminosity command.

6.4.4. blackbody 50,000K, energy density = 500K.

The energy density of the blackbody radiation field, expressed as the equivalent blackbody temperature T_u in degrees Kelvin, is used to set the luminosity when the **energy density** keyword appears anywhere on the line. The energy density temperature is defined from Stefan's law and the actual energy density of the radiation field u (erg cm^{-3}):

$$T_u \equiv \left(\frac{u}{a}\right)^{1/4} \; [\text{K}] \tag{19}$$

where a is the Stefan's radiation density constant.

The second number is assumed to be a log if it is less than or equal to 10 and linear otherwise. Numbers smaller than 10 K will be interpreted as the linear temperature rather than as a log if the keyword **linear** appears. (Note that if the **linear** option is used, then the blackbody temperature must also be linear since the key triggers both.) Note also that cosmic background radiation should also be included if $T_u \leq$ 2.725 (1+z) K. Cloudy will complain, but compute the model, if the energy density of the incident continuum corresponds to a temperature less than the present energy density temperature of the universe.

This is an intensity command.

6.4.5. blackbody, t = 50,000K, _STE

The keyword **_STE**[9] (note the leading space) with no second number is equivalent to the **energy density** option with T_u set to the color temperature of the radiation field. This is a quick way to check that ionization and level populations go to strict thermodynamic equilibrium in the high radiation density limit.

---

[9] The keyword was **LTE** before version 96 of the code. It was changed to **STE** to better reflect the fact that this condition corresponds to strict thermodynamic equilibrium. The code will continue accept the **LTE** keyword for the foreseeable future.

This is an intensity command.

6.4.6. blackbody, t = 100,000K, dilution factor = -14

Here the second parameter is the dilution factor W, defined as

$$W \equiv \frac{J_\nu}{B_\nu} \approx \frac{\pi R_{star}^2}{4\pi r_o^2} \quad , \tag{20}$$

where R_{star} is the radius of the star and r_o is the separation between the illuminated face of the cloud and the center of the star. The approximation on the RHS assumes that $R_{star} \ll r_o$. The dilution factor can be entered either directly or as a log (if the latter, then it will be negative). The example above is a rough approximation of the radiation field within a typical planetary nebula.

This is an intensity command.

6.5. bremsstrahlung, temp = 8

The continuum will be optically thin pure hydrogen bremsstrahlung emission. The form is given by

$$f_\nu \propto \nu^{-0.2} \exp(-h\nu / kT) \quad \text{[erg cm}^{-2}\text{ s}^{-1}\text{ Hz}^{-1}\text{]}. \tag{21}$$

The argument is assumed to be the log of the temperature if it is less than or equal to 10, and linear otherwise. The form of the continuum is approximate since a simple power-law gaunt factor is assumed, and the emission from an optically thin gas with cosmic abundances is actually characterized by hundreds of overlapping emission lines (see, for example, Kato 1976).

A more realistic continuum could be produced by combining the **coronal equilibrium** command (page 104 below) with the **punch transmitted continuum** command (page 136 below) to generate a continuum which can be read in with the **table read** command (page 42 below).

6.6. CMB [redshift 1000]

This command generates a blackbody radiation field in strict thermodynamic equilibrium (i.e., $T_{color} = T_u$, where T_u is the energy-density temperature). The optional argument is the redshift z. If it is not entered then $z = 0$ is assumed. The temperature of the blackbody is given by

$$T_{CMB} = T_o(1+z) \quad \text{K} \tag{22}$$

where the redshift dependence is from Peebles (1971) and the present temperature of the background is assumed to be $T_o = 2.725 \pm 0.002$K (Mather et al. 1999; Wilkinson 1987). This command specifies *both* the shape and intensity of the radiation field. A starting radius of 10^{30} cm will be assumed if no starting radius is specified.

6.7. extinguish column = 23, leak = 0.05, low =4 Ryd

This command will modify a continuum's shape by extinction due to photoelectric absorption by a cold neutral slab with column density given by the first argument

(entered as a log). This occurs *after* the continuum has been fully generated and normalized to the correct intensity. The form of the extinction is a simple power-law fit to the absorption curves calculated by Cruddace et al. (1974). The extinguished continuum f_ν' is related to the initial continuum f_ν by

$$f_\nu'(\nu \geq 1\,\mathrm{Ryd}) = f_\nu \left\{ \eta + (1-\eta)\exp\left(-6.22\times 10^{-18}\, \nu_{Ryd}^{-2.43}\, N(\mathrm{H})\right)\right\} \qquad (23)$$

where $N(\mathrm{H})$ is the total hydrogen column density (cm^{-2}), ν_{Ryd} is the frequency in Rydbergs, and η is the leakage.

The optional second number is the fractional leakage η through the absorber (see Ferland and Mushotzky 1982). This has a default value of 0, i.e., no leakage. The leakage is interpreted as a log if it is negative and linear otherwise. If unexpected or unphysical results occur when the **extinguish** command is given then it is likely that nearly all ionizing radiation has been attenuated. A plot of the generated continuum (with the **plot continuum** command) may prove interesting. The code will stop if the leakage is greater than 1.0 (100%).

The optional third number is the lowest energy for the absorption to occur. The default is 1 Ryd. The number is interpreted as linear Rydbergs if positive and the log of the energy if less than or equal to zero. The continuum with energies below this cutoff energy will be unaffected by the absorption. The non-ionizing ($h\nu < 1$ Ryd) continuum can be extinguished by this command, but extrapolating the power law to these energies is nonsense.

The second two arguments are optional and may be omitted from right to left. The cutoff energy can only be changed if the leakage is specified.

The command acts by first generating the continuum shape, neglecting extinction. The continuum is then normalized using any of the luminosity or intensity commands (i.e., **Q(H)**, **ionization parameter**, **luminosity**, etc.). Only *then* is the continuum extinguished. The continuum that actually strikes the illuminated face of the cloud *does not* have the ionization parameter or luminosity actually entered. (These values would be correct were the extinction not present.) Physically, the luminosity of the central object is not changed by the presence of an absorbing cloud along the line of sight.

This command should not be used except as a quick test. A more physical way to extinguish the continuum would be to actually transmit it through a model of the absorbing slab, save that continuum with the **punch transmitted continuum** command (page 136 below), then use this with the **table read** command (described on page 42 below).

6.7.1. *extinguish optical depth 1.2, [options]*

If **optical depth** appears rather than **column** then the first number is the log of the optical depth. All other options are the same as the **extinguish column** command. *N.B.* it really is the log of the optical depth, not the linear optical depth.

6.8. fireball [redshift = 2000]

This command is replaced with the CMB command, described on page 33 above.

6.9. interpolate [ν(Ryd) or log ν(Hz)], log(f$_\nu$)

Under most circumstances the continuum will actually be entered as a table of points using this command. Cloudy interpolates upon this table using straight lines in log-log space. Up to 130000 ordered pairs of points can be entered, with **continue** lines used to continue entering values after the initial **interpolate** line is filled. (This limit is set by the variable *NCELL*, which also sets the size of continuum arrays.)

Unlike the majority of the commands, the first five characters of the command must be entered.

The first number within each ordered pair of points is *either* the energy in Rydbergs (linear or as a log) *or* the log of the frequency (in Hertz). Cloudy assumes that the log of the energy in Rydbergs was entered if the first number is negative; that the log of the frequency (Hz) was entered if the first number is greater than 5; and linear Rydbergs otherwise. Any of the three styles can be chosen, but must be used consistently within the command. If the first energy is entered as zero then it is interpreted as the low-energy limit of the code. In this case the remaining energies will be interpreted as linear Rydbergs if the second number is positive, and the log of the energies in Rydbergs if negative. An energy of zero Rydberg is not allowed (except for the first), and the energies must be in increasing order.

The second number in each ordered pair is the log of the relative flux density per unit energy interval [$\log_{10}(J_\nu)$+constant] at that energy. These numbers are only used to set the shape of the continuum. The constant in the equation is set by one of the intensity or luminosity commands.

The **interpolate** command can be freely mixed with other continuum shape commands, and a total of up to 100 **interpolate** and **table** (see 37 below) commands can be entered.[10] Note that **table** and **interpolate** are actually two forms of the same command (they store information in the same arrays). The total number of **table** and **interpolate** commands entered together cannot exceed 100.

As an example, the following approximates a metal-poor 45,000 K stellar atmosphere. The energies are entered in Rydbergs:

---

[10]Limits to the use of the **interpolate** command existed in versions 73 and before, but have been lifted.

```
// following is 45000 K atmosphere from Shields and Searle
interpolate (0.00001 -11.106) (.58 -1.5792) (.99 -1.44)
continue (1.01 -1.7018) (1.8 -1.905) (1.81 -1.939)
continue (2.57 -2.208) (2.59 -2.247) (3 -2.3994)
continue (3.02 -2.8193) (3.49 -2.9342) (3.51 -4.143)
continue (3.99 -5.582) (4.01 -6.3213) (6 -9.9) (10 -17.3)
continue (20 -30) (10,000,000 -30)
q(h) = 52.778151
```

Note that the continuum should be specified between 1.001×10^{-8} Ryd and 7.354×10^6 Ryd even if the intensity is small. If it is not fully specified then a warning will be issued and a model computed with the unspecified continuum set to zero intensity, if this is possible. As a further note, it is important that the continuum be physically correct. For instance, stellar model atmospheres emit almost no X-rays, while real OB stars *are* X-ray sources (although neglecting X-rays for these stars is generally a safe approximation). See page 44 below for a further discussion.

Cloudy will stop if more than 130000 frequency points are entered. The maximum number of frequency points allowed is set by the variable *NCELL*.

6.10. laser, frequency = 3.5 Ryd [rel width 0.02]

The intensity of the continuum will be very small, except within ±5% of the specified energy, where it will be very large. The energy is specified in Rydbergs, and it is interpreted as a log if it is negative. This is provided as a way to check on the computation of the photoionization rate integrals.

The optional second number on the command line can change the relative width of the laser. The relative width is the ratio *dE/E* where *dE* is the half width of the laser. The laser will only be active within ± *dE* of *E*. The code would return an error condition if *dE* is too small, since the laser may not happen to be evaluated within ± *dE* of *E*. The fractional width probably should not be made smaller than roughly 0.01 but the code does not protect against too small a value of *dE*.

Another way to make a laser is to punch out a transmitted continuum, edit this file, and, by hand, increase the intensity of the continuum at particular cells. This is described where the **table read** command is defined (page 42 below).

6.11. power law, slope =-1.4 [hi cut =6 Ryd low cut =.1, Kelvin]

N.B. IT IS VERY DANGEROUS TO USE THIS COMMAND. The continuum will be a power law with slope given by the first parameter. It has optional low-energy and high-energy exponential cutoffs $\nu_{high\ cut}$ and $\nu_{low\ cut}$ in Rydbergs. The form of the continuum is

$$f_\nu = \nu^{+\alpha} \exp\left(-\nu / \nu_{high\ cut}\right) \exp\left(-\nu_{low\ cut} / \nu\right) \ [\text{erg cm}^{-2} \text{ s}^{-1} \text{ Hz}^{-1}]. \tag{24}$$

The first number on the command line is the slope α. Note that there is no implicit negative sign in this exponent; typical AGN have $\alpha_{ox} \sim -1.4$, (Zamorani et al. 1981). The second (optional) number is the high-energy cutoff $\nu_{high\ cut}$. The third optional number is the low-energy cutoff $\nu_{low\ cut}$. Both are expressed in Rydbergs, and can be omitted from right to left. The default values are 10^4 and 10^{-4} Ryd.

If the keyword **Kelvin** appears then both cutoff energies are interpreted as temperatures in Kelvin, rather than energies in Rydbergs. The temperature is a log if it is less than or equal to 10, and the linear temperature itself if greater than 10.

It is generally a *very bad* idea to use this command. Cloudy treats the entire continuum between *very* low and *very* high energies. Extrapolating reasonable continua past the optical-ultraviolet region into radio or γ-ray energies will have unexpected effects. Power-law continua with slopes smaller than -1 will have unphysically large photon occupation numbers and brightness temperatures at very long wavelengths. This will probably produce catastrophic Compton cooling and/or free-free heating. Continua with slopes greater than -1 will be dominated by the radiation field at energies of many MeV, resulting in large Compton heating and pair production rates. The exponential cutoffs can prevent this, but they also drive the continuum to zero intensity when either argument in the exponential becomes large. This is unphysical, and can cause numerical problems.

It is *much* better to use the **interpolate** command (page 35 above), and enter physically reasonable low-energy and high-energy continua. There is a special version of the command, **table power law** (see page 41 below) for entering a well-behaved power-law continuum.

6.12. table command

6.12.1. Overview

Any of several continuum shapes that are stored as a permanent part of the code can be entered with this command. This is a special version of the **interpolate** command, described on page 35 above. The same interpolation on a table of input frequencies and fluxes described there is done when this command is entered. The **table** command can be freely mixed with other shape commands, and a total of up to 100 **table** and **interpolate** commands can be entered.

6.12.2. table _agn

If the keyword **_agn** appears (note the presence of a leading space), then a continuum similar to that deduced by Mathews and Ferland (1987) will be used. The continuum is meant to be similar to typical radio quiet active galaxies. The points used to describe this continuum are given in Table 4.

This continuum differs from the Mathews and Ferland (1987) continuum only in that the continuum is assumed to have a sub-millimeter break at 10 microns. For wavelengths longer than 10 microns the continuum

Table 4
AGN Continuum

ν(Ryd)	log (F_ν)	slope
1.00(-5)	-3.388	+2.50
9.12(-3)	4.0115	-1.00
0.206	2.6576	-0.50
1.743	2.194	-1.00
4.130	1.819	-3.00
26.84	-0.6192	-0.70
7.35(+3)	-2.326	-1.67
7.40(+6)	-7.34	-

is assumed to have a slope $f_\nu \propto \nu^{+2.5}$, appropriate for a self-absorbed synchrotron continuum (Rybicki and Lightman 1979). Note that this represents a typical observed continuum, and may not be directly related to the continuum actually striking BLR gas (Korista, Baldwin, and Ferland 1997).

The energy of the sub-millimeter break is not well determined observationally, but has a major impact on high density, high ionization parameter models, as discussed by Ferland and Persson (1989), Ferland et al. (1992), and Ferland (1999a). The energy of the infrared break can be adjusted with the **break** keyword. The break can be adjusted between the limits of 0.2 Rydberg and 1.001×10^{-8} Ryd by entering the keyword **break** followed by a number specifying the energy of the break. The number is interpreted as the log of the energy in Rydbergs if it is negative and as linear Rydbergs if positive. It is interpreted as the linear wavelength of the break in microns if the keyword **microns** also appears. If no number appears, but the keywords **no break** does, then a break at the low-energy limit of the code (1.001×10^{-8} Ryd) is assumed. The following shows equivalent ways of generating a continuum with a break at 10 microns;

Table 5	
Akn 120 Continuum	
ν(Ryd)	f_ν
1.0(-5)	1.5(-26)
1.9(-5)	1.6(-26)
3.0(-4)	1.4(-23)
2.4(-2)	8.0(-25)
0.15	1.6(-25)
0.30	1.8(-25)
0.76	7.1(-26)
2.0	7.9(-27)
76.	1.1(-28)
7.6(+2)	7.1(-30)
7.4(+6)	1.3(-34)

```
table AGN break .00912   ; energy in Ryd
table AGN break -2.04    ; log of energy in Ryd
table AGN break 10 microns ; wavelength in microns
table AGN no break       ; no sub-millimeter break
```

Note that the nature of the continuum in AGN is still an open question. The continuum given here is very simplistic, and quite uncertain in the ionizing ultraviolet. Moreover, it would not be surprising if the BLR sees a far different continuum than we do. This continuum may not be correct for low redshift Seyfert galaxies (Binette et al. 1989; Clavel and Santos-Lleo 1990) and direct observations of high-redshift quasars suggest a far softer continuum than this (Zheng et al. 1997; Korista, Ferland, & Baldwin 1997). It is probably best to only use this continuum in exploratory situations, and to generate a specific AGN continuum using the **ratio** command, as described on page 27 above, or by using the **agn** command described on page 29 above.

6.12.3. table akn120

If the keyword **akn120** appears then the continuum summarized by Peterson et al. (in preparation) is used. The continuum is described by the observed flux at Earth [erg cm^{-2} s^{-1} Hz^{-1}] and is given in Table 5.

The monochromatic luminosity at 1320 Å is $\nu L_\nu = 1.84 \times 10^{44}$ h^{-2} erg s^{-1}, where $h \equiv H_o/100$ km s^{-1} mpc^{-1}, so, setting $h = 0.75$, the **akn120** continuum could be generated by the commands

```
nul(nu) = 44.514 at 0.6906 Ryd
table akn120
```

6.12.4. table cooling flow

The keyword **cool** generates the continuum described by Johnstone et al. (1992) and used in Ferland, Fabian, & Johnstone (1994; 2002). It is a co-added series of

Raymond-Smith collisional equilibrium continua, chosen to represent the spectrum at a point within a typical cooling flow.

6.12.5. table crab

If the keyword **crab** appears then the continuum summarized by Davidson and Fesen (1985) is generated. This is the net observed continuum, originating in both the pulsar and nebula, and not the pulsar continuum alone. The continuum is given in Table 6, which gives the observed flux at Earth (erg cm^{-2} s^{-1} Hz^{-1}).

According to Davidson and Fesen, the total luminosity of the Crab is L_{tot} = $10^{38.14}$ erg s^{-1}, so the Crab continuum could be generated by combining the commands

```
luminosity (total) 38.14
table Crab
```

6.12.6. table Draine [factor=1.7]

This enters the galactic background radiation field given by equation 23 of Draine & Bertoldi (1996). The continuum is only defined over a very narrow wavelength range so it is only appropriate for certain simple PDR calculations. It is shown in Figure 5 along with the continuum produced by the **table ism** command. This command specifies both the shape and intensity of the continuum. The optional scale factor changes the intensity of the continuum.

Table 6
Crab Continuum

ν(Ryd)	f_ν
1.0(-5)	3.77E-21
5.2(-4)	1.38E-21
1.5(-3)	2.10E-21
0.11	4.92E-23
0.73	1.90E-23
7.3	2.24E-24
73.	6.42E-26
7.3(+3)	4.02E-28
1.5(+6)	2.08E-31
7.4(+6)	1.66E-32

This continuum is defined up to an energy very close to 1 Rydberg. It may spill over into the hydrogen-ionizing continuum for certain choices of the continuum bin resolution. Add the **extinguish** command, described on page 33 above, to insure that H-ionizing radiation is extinguished if this is needed.

This continuum should only be used for restricted tests since the vast majority of the electromagnetic spectrum is undefined. It was created to compute the Leiden PDR test cases (the *pdr_leiden** simulations in the test suite, and the paper by Röllig et al. 2005).

6.12.7. table HM96 [factor=-1]

This enters the Haardt & Madau (1996) background continuum for a redshift of *z* = 2.16. This is the original form of the command which is maintained for reference. The cosmic microwave background is not included – use the **CMB** command described on page 33 above to add this component. Note that this table specifies both the shape and intensity of the continuum. There is an optional multiplicative scale factor to change the intensity. If the scale factor is less than or equal to zero then it is interpreted as the log of the scale factor.

The **background** command (see page 30 above) uses a simplistic form of the energetic continuum, but allows any redshift to be specified.

6.12.8. table HM05 redshift 2.21 [quasar; factor=-1]

This enters the Haardt & Madau (2005 in preparation) continuum at any redshift 0 $\leq z \leq$ 8.9. It interpolates on tables kindly provided by Francesco Haardt. There are two forms of the continua – the default includes both quasars and galaxies. If the keyword **quasar** occurs then a continuum considering only quasars is used. The cosmic microwave background is not included – use the **CMB** command described on page 33 above to add this component. Note that this table specifies both the shape and intensity of the continuum. There is an optional multiplicative scale factor to change the intensity. If the scale factor is less than or equal to zero then it is interpreted as the log of the scale factor.

6.12.9. table _ism [factor = 0.7]

The local interstellar radiation field is generated with the keyword **_ism**. This uses Figure 2 of Black (1987) to represent the *unextinguished* local interstellar radiation field (see Figure 5). This command specifies *both* the shape and luminosity of the radiation field. The continuum generated by Cloudy is exactly that given by Black, except that the radiation field between 1 and 4 Ryd is interpolated from the observed or inferred values. Actually, it is thought that this part of the radiation
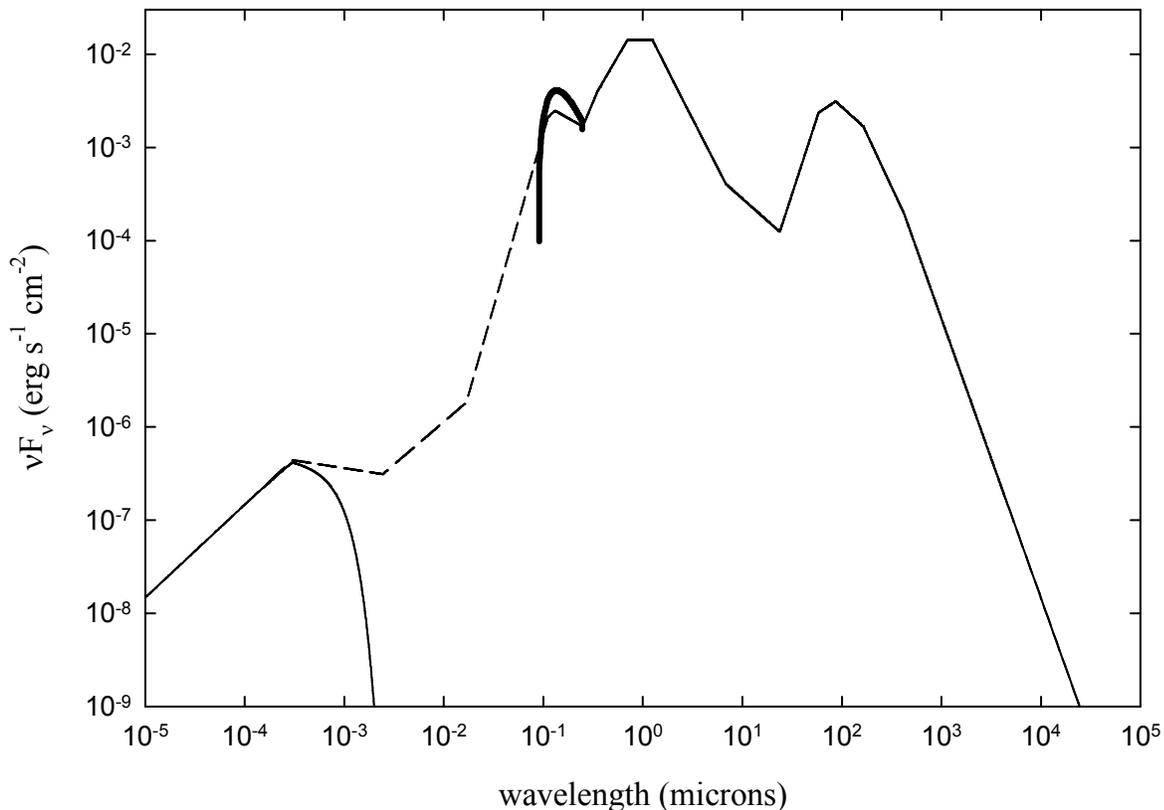
Figure 5 The continuum produced by the table ISM command. The infrared cirrus is the peak at $\lambda \sim$ 100 μm and starlight dominates at shorter wavelengths. The points just shortward of the Lyman limit (0.0912 μm) are interpolated – actually it is thought that interstellar extinction removes most of this continuum. The dashed line shows the interpolated continuum and the solid line shows the effects of absorption introduced by the **extinguish** command. The heavy line is the **table Draine** continuum. Ism_background

field is heavily absorbed by gas in the ISM, so that little 1 to 4 Ryd radiation exists, at least in the galactic plane. Such absorption can be introduced with the **extinguish** command, described on page 33 above.

The **table ism** command also specifies the intensity of the incident radiation field, since this is directly observed. There is an optional parameter that specifies a scale factor for the intensity of the entire radiation field. It is the log of the scale factor if less than or equal to zero, and the scale factor itself if positive. The default is unity (i.e., Black's radiation field). The actual numbers used by Cloudy to interpolate on Black's table are given in Table 7. The frequencies are in Hz, and the product νf_ν in erg cm^{-2} s^{-1}.

Table 7 ISM Radiation Field			
log(ν)	log νf_ν	log ν	log νf_ν
9.00	-7.93	14.14	-2.30
10.72	-2.96	14.38	-1.79
11.00	-2.47	14.63	-1.79
11.23	-2.09	14.93	-2.34
11.47	-2.11	15.08	-2.72
11.55	-2.34	15.36	-2.55
11.85	-3.66	15.54	-2.62
12.26	-2.72	16.25	-5.68
12.54	-2.45	17.09	-6.45
12.71	-2.57	18.00	-6.30
13.10	-3.85	23.00	-11.30
13.64	-3.34		

Note that this radiation field does not include the cosmic microwave background. Use the **CMB** command described on page 33 above to add this component.

The actual ISM radiation field incident on a typical region in the galactic plane could be generated by:

```
table ISM
extinguish column = 22 leak=0 .
```

6.12.10. table power law [spectral index -1.4, low =.01, hi =20]

This command produces a power-law continuum that is well behaved at both the high and low energy ends. The default shape, assumed when no numbers occur on the command line, is the form $f_\nu \propto \nu^\alpha$. Here $\alpha = -1$ for the spectral midrange between 10 microns and 50 keV, and the continuum has slopes $f_\nu \propto \nu^{5/2}$ at lower energies (appropriate for self-absorbed synchrotron, eq 6.54, p.190, Rybicki & Lightman 1979) and $f_\nu \propto \nu^{-2}$ at higher energies. Table 8 summarizes the default continuum.

Three optional numbers may appear on the command line. The first number sets the slope of the mid-range spectral component (infrared to X-ray) and has a default of -1 ($f_\nu \propto \nu^{-1}$).

The next two numbers adjust the energy limits of the mid-range spectral component. Their default units are Rydbergs but the keyword **microns** will change the units to microns *for the first energy only*. The second number is the energy (in Rydbergs) of the infrared break. The default is 0.009115 Ryd (10 microns). If this second number is zero then the low energy limit to the continuum (1.001×10^{-8} Ryd) will be used. The number is interpreted as the log of the energy in Rydbergs if it is negative and linear otherwise. Note that, with no infrared break, free-free heating will

Table 8 Power Law Continuum	
ν(Ryd)	slope
1.00(-8)	+2.50
9.115(-3)	-1.00
3676.	-2.
7.40(+6)	-

probably be significant for denser clouds. A power law continuum with a low energy break at 1 micron would minimize this heating, and could be generated with the command

```
// a power-law with index -1 and 1 micron break
table power law slope -1, 1 micron break
// a power-law with index -1 and 10 micron break, the default
table power law slope -1
```

The third optional number is the energy (in Rydbergs) of the break in the X-ray continuum. The default is 50 keV, and if it is zero then the high-energy limit of the continuum (7.354×10^6 Ryd) is used. The number is interpreted as a log if the energy of the infrared break is entered as a log and linear otherwise. The numbers may be omitted from right to left.

6.12.11. table read "contin.txt"

This command is used to read in the continuum predicted from a previous Cloudy calculation. The first calculation saves the continuum transmitted through a cloud with the **punch transmitted continuum** command. Subsequent calculations use the **table read** command to include this continuum.

The **punch transmitted continuum** command is described on page 136 below. It produces a file containing the frequency in Rydbergs and the transmitted continuum νf_ν (erg cm^{-2} s^{-1}). This continuum is the sum of the attenuated incident continuum and the fraction of the diffuse emission from the cloud that is transmitted in the outward direction. The first two lines of the input file contain header information and are skipped. They should not be deleted.

The **table read** command can be freely mixed with all of the other continuum shape commands. Any number of **table read** commands can be entered[11]. The continuum file must have been produced by the same version of Cloudy. The code will stop if otherwise.

The name of the file containing the previous continuum must be enclosed in a pair of double quotes.

The following gives an example of first creating a file containing the transmitted continuum, then using this file as one of the continua in a later calculation.

```
title this finds transmitted continuum due to warm absorber
hden 9
ionization parameter 1
stop effective column density 21
table AGN
punch transmitted continuum file = "absorber.txt"
```

Now use this continuum in a second calculation:

```
table read file = "absorber.txt"
luminosity 45
radius 18
hden 9
```

---

[11]Only one table read command could be entered in versions 90 and before.

6.12.12. table Rubin

Nearly all attempts at modeling the Orion Nebula have found that theoretical stellar atmospheres do not produce enough flux near 4 Ryd (see, for example, Mathis 1982, 1985; Rubin et al. 1991; Sellmaier et al. 1996).

Bob Rubin has modified the emergent continuum from one of the Kurucz (1979) models to better account for the presence of high ionization lines in the Orion Nebula. This modified continuum can be accessed with the **table Rubin** command. The continuum started life as a log g = 4, T_{eff} = 37,000 K Kurucz model, but the flux between 41 eV and 54 eV was raised by a factor of 11 to reproduce the [Ne III] optical and IR lines.

6.13. table starburst age=6.7 "data.txt"

This command reads in predictions from Starburst 99 (Leitherer et al. 1999). A filename containing the stellar continua as a function of starburst age must appear within the double quotes. The age of the starburst in years must also appear. This is assumed to be linear years unless the keyword **log** forces the interpretation as a log. The age must be between the first and last ages in the data file. The code will not extrapolate. The code does linear interpolation in log age – log φ_ν space, where φ_ν is the photon flux.

The test case *sb99.in* in the test suite demonstrates the operation of this command. The sample Starburst 99 data file included for that test shows the form of the expected input.

Only one **table starburst** can be used in a calculation. If more than one command is entered only the last will be honored.

It is critical that the original Starburst 99 continuum not be changed in any way at all. Doing so will trick the code into generating bogus results. In any case, it is a good idea to make a plot of the generated continuum, using column 1 and 2 of the output from the **punch continuum** command described on page 131 below, to confirm that it is physical.

6.14. table stars overview

Several sets of emergent continua from stellar atmosphere calculations are accessible. These are used when the keyword **star**, followed by a sub-keyword (**Mihalas**, **Kurucz**, **Atlas**, **CoStar**, **Rauch**, or **Werner**) indicate which set of atmospheres to use.

Figure 6 compares predictions for the five 50,000 K continua now included. These include a blackbody and atmospheres computed by Mihalas (1972), Kurucz (1979), Kurucz (1991) and Rauch (1997). All were normalized to have the same total luminosity (10^{38} erg s^{-1}) observed from a distance of 10^{18} cm. Note the order of magnitude dispersion among the continua for energies around 4 Ryd.
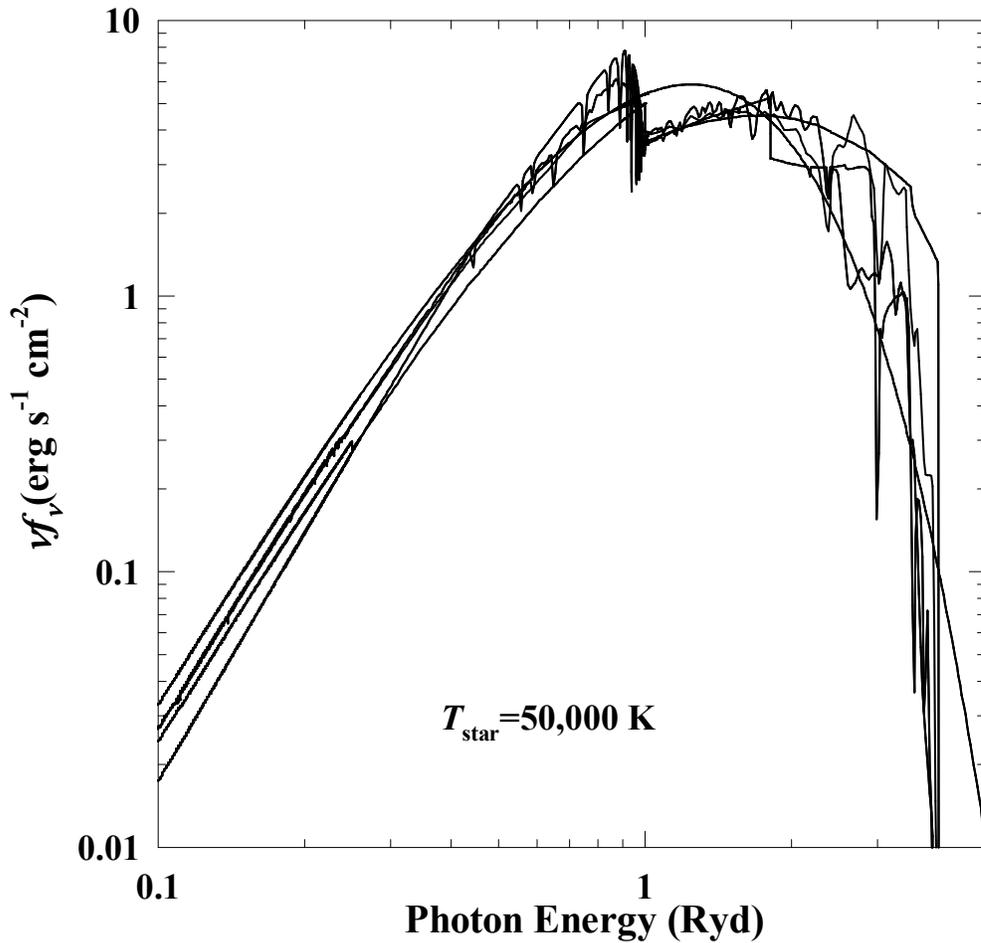
Figure 6 This figure shows the emergent continua predicted by five 50,000 K stars included with the code. The smoothest is the blackbody, and the Kurucz (1991) and Rauch (1997) atmospheres show the most structure. stars

These commands specify only the continuum shape. It is still necessary to specify a luminosity. Tout et al. (1996) provide convenient fitting formulae giving zero age main sequence luminosities as functions of stellar mass and metallicity.

6.14.1. A high-energy component?

Theoretical stellar atmospheres emit little energy above 4 Ryd while real OB stars *are* X-ray sources. Sciortino et al. (1990) find a correlation between the X-ray and bolometric luminosities which can be fitted by

$$\log(L_x) = 1.08(+0.06/-0.22)\log(L_{bol}) - 9.38(+2.32/-0.83) \ . \tag{25}$$

The X-ray luminosity is typically ~6.4 dex fainter than the bolometric luminosity. A source temperature of 0.5 keV is quoted by Sciortino et al. This X-ray continuum must be explicitly added as an independent continuum source. Tests show that the high-energy light has little effect on conditions in the H II region, but *does* affect the ionization in the surrounding PDR.

6.14.2. table star [Kurucz; Mihalas]

Subsets of the Mihalas (1972) non-LTE OB stellar atmospheres and the Kurucz (1979; with supplements) line-blanketed LTE atmospheres are built into the code. Both are static plane-parallel atmospheres. Table 9 summarize the parameters of the models. The temperature and author of the calculation (**Kurucz** or **Mihalas**) must be specified; these can be in any order.

Any temperature between the lowest and highest temperatures listed in the tables can be interpolated, but only the listed gravities can be generated. The Kurucz models are all for solar abundances. If the specified temperature is within a tenth of a percent of one of the temperatures listed in the tables then exactly the published continuum will be used. Otherwise a linear interpolation in temperature-magnitude (Mihalas) or temperature - $\log(f_\nu)$ (Kurucz) space will be done. Extrapolation is not performed; the temperature must be between the lowest and highest values.

Table 9
Mihalas (1972) & Kurucz (1979) Continua

Mihalas		Kurucz		
T_*	$\log(g)$	T_*	$\log(g)$	Reference
30,000	4.0	30,000	4.0	Kurucz (1979)
32,500	4.0	35,000	4.5	private comm
35,000	4.0	40,000	4.5	private comm
37,500	4.0	45,000	4.5	Kurucz (1979)
40,000	4.0	50,000	4.5	Kurucz (1979)
45,000	4.0			
50,000	4.0			
55,000	4.0			

The following would roughly correspond to θ^1 Ori C, the ionizing star in the Orion Nebula;

```
table star Kurucz 39,000
q(h) 49    .
```

6.14.3. table star atlas, temp =40,000 [log(g)=4.5]

Kevin Volk incorporated the Kurucz (1991) grid of Atlas models into Cloudy. The **table star atlas** command asks the code to interpolate on this grid to generate an incident continuum produced by an atmosphere with the specified temperature and gravity. The first number on the command line is the temperature, interpreted as a log if less than or equal to 10, and the second optional number is the log of the surface gravity. A gravity of $\log(g) = 5$ will be used if none is specified. All models are for solar abundances. Table 10 lists the temperatures and surface gravities stored within this set.

The code checks that the temperature is within the bounds of the table so that only interpolation is performed. It does not check that the gravity is within the bounds.

This grid is far too large to actually store within the code. Instead it is stored as an ancillary file, which is generated by compiling some files obtained from the web. This process is described on page 169 below. If the code is executed from directories other than the one containing the compiled star data file then it is also necessary to set the path to the directory containing the files by editing *path.c*, as described on page 191 below.

It may take the machine some time to find the desired atmosphere and interpolate to the correct temperature and gravity. If the same atmosphere is to be used repeatedly then it may be faster to save the interpolated atmosphere with the **punch incident continuum** command, and then read this in for later calculations with the **table read** command, as shown below;

```
// this input stream is just
// to get the right continuum
// still need to set
//parameters
// for the model
hden 0
ionization parameter -1
constant temper 4
set dr 0 # zone thickness of 1
cm
stop zone 1
punch incident continuum
table star atlas, t=33,375 log(g)=4.26
```

This produces a punch file containing the interpolated continuum. The real calculation can then use the interpolated continuum by reading the punch file

Table 10
Atlas (Kurucz 1991) Continua

T/g	0.0	0.5	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
3500	x	x	x	x	x	x	x	x	x	x	x
3750	x	x	x	x	x	x	x	x	x	x	x
4000	x	x	x	x	x	x	x	x	x	x	x
4250	x	x	x	x	x	x	x	x	x	x	x
4500	x	x	x	x	x	x	x	x	x	x	x
4750	x	x	x	x	x	x	x	x	x	x	x
5000	x	x	x	x	x	x	x	x	x	x	x
5250		x	x	x	x	x	x	x	x	x	x
5500	x	x	x	x	x	x	x	x	x	x	x
5750	x	x	x	x	x	x	x	x	x	x	x
6000	x	x	x	x	x	x	x	x	x	x	x
6250		x	x	x	x	x	x	x	x	x	x
6500		x	x	x	x	x	x	x	x	x	x
6750		x	x	x	x	x	x	x	x	x	x
7000		x	x	x	x	x	x	x	x	x	x
7250		x	x	x	x	x	x	x	x	x	x
7500		x	x	x	x	x	x	x	x	x	x
7750			x	x	x	x	x	x	x	x	x
8000			x	x	x	x	x	x	x	x	x
8250			x	x	x	x	x	x	x	x	x
8500			x	x	x	x	x	x	x	x	x
8750				x	x	x	x	x	x	x	x
9000				x	x	x	x	x	x	x	x
9250					x	x	x	x	x	x	x
9500					x	x	x	x	x	x	x
9750					x	x	x	x	x	x	x
10000					x	x	x	x	x	x	x
10500					x	x	x	x	x	x	x
11000						x	x	x	x	x	x
11500						x	x	x	x	x	x
12000						x	x	x	x	x	x
12500						x	x	x	x	x	x
13000						x	x	x	x	x	x
14000					x	x	x	x	x	x	x
15000						x	x	x	x	x	x
16000						x	x	x	x	x	x
17000						x	x	x	x	x	x
18000						x	x	x	x	x	x
19000						x	x	x	x	x	x
20000							x	x	x	x	x
21000							x	x	x	x	x
22000							x	x	x	x	x
23000							x	x	x	x	x
24000							x	x	x	x	x
25000							x	x	x	x	x
25000							x	x	x	x	x
26000							x	x	x	x	x
27000								x	x	x	x
28000								x	x	x	x
29000								x	x	x	x
30000								x	x	x	x
31000								x	x	x	x
32000									x	x	x
33000									x	x	x
34000									x	x	x
35000									x	x	x
37500										x	x
40000										x	x
42500											x
45000											x
47500											x
50000											x

produced by this run with the **table read** command, as described on page 42 above.

6.14.4. table star Rauch, temp=100,000 [halo log(g)=6.5]

Kevin Volk has incorporated both the first generation Rauch (1997) and second generations Rauch (2002) grids of non-LTE model planetary nebula nuclei into Cloudy. This command asks the code to interpolate on one of these grids to generate an atmosphere with the specified temperature and gravity. The first number on the command line is the temperature, interpreted as a log if less than or equal to 10, and the second optional number is the log of the surface gravity. A gravity of $\log(g) = 8$ will be used if none is specified. Table 11 lists the available temperatures and gravities in the first generation set, and Table 12 gives these values in the second generation set.

Two abundance sets are available. The default is solar and the keyword **halo** will invoke the low-metallicity halo abundance set. The default is to use the second generation grids, but the first generation will be used if the keyword **_old** appears on the command line. The Rauch web site http://astro.uni-tuebingen.de/~rauch/flux.html gives further details on the stellar atmospheres.

The code checks that the temperature is within the bounds of the table so that only interpolation is performed. It does not check that the gravity is within the bounds. It will not allow the gravity to exceed the range of the table.

Table 11
"First Generation"
Rauch (1997) Continua

temp\log g	5	6	7	8	9
50,000K	*	*	*	*	
60,000K	*	*	*	*	
70,000K	*	*	*	*	
80,000K	*	*	*	*	
90,000K	*	*	*	*	
100,000K	*	*	*	*	
110,000K		*	*	*	
120,000K		*	*	*	
130,000K		*	*	*	
140,000K		*	*	*	
150,000K		*	*	*	
160,000K		*	*	*	
170,000K		*	*	*	
180,000K		*	*	*	
190,000K		*	*	*	
200,000K			*	*	*
300,000K			*	*	*
400,000K				*	*
500,000K				*	*
600,000K					*
700,000K					*
800,000K					*
900,000K					*
1,000,000K					*

This grid is far too large to actually store within the code, so instead is stored as an ancillary file, which is generated by compiling files obtained from the web. Compiling the star files is described on page 169 below. If the code is executed from directories other than the one containing the compiled file then it is also necessary to set the path to the directory containing the files with the **set path** command or by editing *path.c* as described on page 191 below.

The treatment of these files is entirely analogous to that of the **table star atlas** command (see page 45 above). This also describes a way to save one of these atmospheres for later use.

6.14.5. table star Werner, temp =140,000 [log(g)=7.4]

The Werner and Heber (1991) grid of non-LTE model planetary nebula nuclei atmospheres was incorporated into Cloudy by Kevin Volk. This command asks the

code to interpolate on this grid to generate an atmosphere with the specified temperature and gravity. The first number on the command line is the temperature, interpreted as a log if less than or equal to 10, and the second optional number is the log of the surface gravity. A gravity of $\log(g) = 8$ will be used if none is specified. Table 13 lists the temperatures and surface gravities stored within this set.

This grid is far too large to actually store within the code, so instead is stored as an ancillary file, which is generated by compiling files obtained from the web. Compiling the star files is described on page 169 below. If the code is executed from directories other than the one containing the compiled file then it is also necessary to set the path to the directory containing the files with the **set path** command or by editing *path.c* as described on page 191 below. The treatment of these files is entirely analogous to that of the **table star atlas** command (see page 45).

The discussion of the **table star atlas** command (see page 45 above) describes a way to save one of these atmospheres for later use.

6.14.6. *table star CoStar, temp=34,700K log(g)=4 [index 2, ZAMS, age]*

The CoStar (Schaerer et al. 1996a, b; and Schaerer & de Koter 1997) grid of windy hot stellar atmospheres was incorporated into Cloudy in collaboration with Peter van Hoof. These stars have temperatures ranging from 18,521 K through 53,397 K. The original ASCII file obtained from their web site must be compiled to bring it into the Cloudy energy mesh and convert to binary format for rapid access. There are many possible ways to interpolate on this grid, as described next.

If the keyword **halo** appears then the halo abundance set will be used. If no keyword appears, or the keyword **solar** appears, then the solar set will be used.

table star CoStar, 37,000K, index 2. The CoStar grid consists of 6 evolutionary tracks, each corresponding to a specific Zero Age Main Sequence (ZAMS) mass, and two abundance sets, solar and halo. Each evolutionary track consists of several models (ranging from 3 to 7, depending on the track). The index number of the model corresponds to a specific age of the star as it is followed through its evolution. So the index number is a rough way to indicate how evolved the star is; index 1 indicates a main sequence star, higher numbers indicate more evolved stars. A summary of the effective temperatures of all models can be found in Table 14. If the keyword **index** appears then the index is the second optional number, and the code will interpolate the temperature along this index.

Table 12
"Second Generation"
Rauch (2002) Continua

temp\log g	5	6	7	8
50,000K	*	*	*	*
60,000K	*	*	*	*
70,000K	*	*	*	*
80,000K	*	*	*	*
90,000K	*	*	*	*
100,000K	*	*	*	*
110,000K		*	*	*
120,000K		*	*	*
130,000K		*	*	*
140,000K		*	*	*
150,000K		*	*	*
160,000K		*	*	*
170,000K		*	*	*
180,000K		*	*	*
190,000K		*	*	*

Table 13
Werner and Heber
(1991) Continua

Temp\log g	5	6	7	8
80,000K	*	*	*	*
100,000K	*	*	*	*
120,000K		*	*	*
140,000K		*	*	*
160,000K			*	*
180,000K			*	*
200,000K			*	*

Table 14
CoStar Continua

Index	1	2	3	4	5	6	7
Track							
A (20 M$_\odot$)	35575.4	33350.3	30415.6	25175.5			
B (25 M$_\odot$)	38557.5	36308.4	32628.6	27756.3			
C (40 M$_\odot$)	44248.6	41741.2	37798.4	30289.8	25955.6	18521.0	
D (60 M$_\odot$)	47964.9	46115.8	39538.3	32199.4	26276.2	22082.2	22476.5
E (85 M$_\odot$)	51014.4	48499.7	42563.7				
F (120 M$_\odot$)	53397.7	50319.8	47305.6				

If only one number is on the line and no keyword appears, an index of unity is assumed. It is possible to specify a ZAMS mass, surface gravity, or age with the following commands. The original CoStar file should be consulted to see what range of parameters is available.

table star Costar, 37,000 4. With no keyword the second parameter is the log of the surface gravity.

table star CoStar, ZAMS mass=30, time=1000000. With the **ZAMS** keyword the ZAMS mass and age of the star is specified.

table star CoStar, age=1000000, mass=30. With the **_AGE** keyword the mass and age of the star is specified.

6.15. table tlusty "G55000g450v10.flux"

One of the TLUSTY stellar atmospheres will be used as the incident continuum. The option was introduced by Kevin Volk. The code and resulting grid are described on the home page, http://tlusty.gsfc.nasa.gov/. This grid is *very* large and it is not practical to interpolate upon it at the present time.

To use a TLUSTY stellar atmosphere you must first download one of the spectral energy distribution files on the TLUSTY web site and copy it to the directory in which you are working. Then include the stellar continuum shape with this command, giving the name of the TLUSTY spectral energy distribution file within the double quotes.

This command is actually a version of the **table read** command, see page 42 above. There is currently a limit of one **table read** command per input run, so it is not possible to include other **table read** commands when this is used.

7. CHEMICAL COMPOSITION

7.1. Overview

The default solar composition is summarized in Table 15. C and O abundances come from photospheric abundances of Allende Prieto et al. (2002, 2001), while N, Ne, Mg, Si, and Fe are from Holweger (2001). The helium abundance is a typical value for nebulae with near-solar compositions. The remainder of the first thirty elements comes from Grevesse & Sauval (1998). Meteoritic and photospheric abundances agree for most elements. They differ by significant amounts for P, S, Cl, and Mn. These are fairly volatile elements, so may be deficient in meteorites. For these four the means of the meteoritic and photospheric abundances were used.

Abundances are always specified by *number* relative to *hydrogen*, not by mass or silicon. Abundances are relative to the total hydrogen density, the sum of H in atomic, ionic, and molecular form. These are gas phase abundances, and do not include material locked into grains. The composition will be printed in the header information that starts the printout. You should check this to confirm that the right composition is used.

Table 15
Solar Composition

A			12+log	log	n/n(H)	ref
1	H	Hydrogen	12.00	0.00	1.00E+00	GS98
2	He	Helium	11.00	-1.00	1.00E-01	text
3	Li	Lithium	3.31	-8.69	2.04E-09	GS98
4	Be	Beryllium	1.42	-10.58	2.63E-11	GS98
5	B	Boron	2.79	-9.21	6.17E-10	GS98
6	C	Carbon	8.39	-3.61	2.45E-04	AP02
7	N	Nitrogen	7.93	-4.07	8.51E-05	H01
8	O	Oxygen	8.69	-3.31	4.90E-04	AP01
9	F	Fluorine	4.48	-7.52	3.02E-08	GS98
10	Ne	Neon	8.00	-4.00	1.00E-04	H01
11	Na	Sodium	6.33	-5.67	2.14E-06	GS98
12	Mg	Magnesium	7.54	-4.46	3.47E-05	H01
13	Al	Aluminium	6.47	-5.53	2.95E-06	GS98
14	Si	Silicon	7.54	-4.46	3.47E-05	H01
15	P	Phosphorus	5.51	-6.50	3.20E-07	GS98*
16	S	Sulphur	7.27	-4.74	1.84E-05	GS98*
17	Cl	Chlorine	5.28	-6.72	1.91E-07	GS98
18	Ar	Argon	6.40	-5.60	2.51E-06	GS98
19	K	Potassium	5.12	-6.88	1.32E-07	GS98
20	Ca	Calcium	6.36	-5.64	2.29E-06	GS98
21	Sc	Scandium	3.17	-8.83	1.48E-09	GS98
22	Ti	Titanium	5.02	-6.98	1.05E-07	GS98
23	V	Vanadium	4.00	-8.00	1.00E-08	GS98
24	Cr	Chromium	5.67	-6.33	4.68E-07	GS98
25	Mn	Manganese	5.46	-6.54	2.88E-07	GS98*
26	Fe	Iron	7.45	-4.55	2.82E-05	H01
27	Co	Cobalt	4.92	-7.08	8.32E-08	GS98
28	Ni	Nickel	6.25	-5.75	1.78E-06	GS98
29	Cu	Copper	4.21	-7.79	1.62E-08	GS98
30	Zn	Zinc	4.60	-7.40	3.98E-08	GS98

References: GS98: Grevesse & Sauval (1998), GS98* - mean of photospheric and meteoritic, H01: Holweger (2001), AP01, AP02 Allende Prieto et al. (2001, 2002).

The following sections describe how to modify the chemical composition.

7.2. Absolute abundances and scale factors

Abundances can be specified as either *absolute abundances* or as *scale factors*, relative to some standard abundance. An absolute abundance gives the abundance of an element relative to hydrogen. An example might be $n(O)/n(H) = 4 \times 10^{-4}$. A

relative abundance is given relative to another value. An example might be the abundance relative to the solar value, $n(\mathrm{O})/n(\mathrm{H}) = 4 \times [n(\mathrm{O})/n(\mathrm{H})]_\odot$.

7.3. Precedence

If the absolute abundance is specified with more than one command then the last abundance is used. If the abundance is specified by both its absolute abundance and by a scale factor then both is used. Either of the following will multiply the default H II region nitrogen abundance by a factor of two:

```
abundances H II region
element scale factor nitrogen 2
```

or

```
element scale factor nitrogen 2
abundances H II region
```

The **abundances H II region** command uses one of the absolute abundance mixtures that are stored by the code. These are listed in Table 16. This command enters the gas-phase abundances of all 30 elements and also includes interstellar grains. The **element** command changes some details about a particular element and is described further on page 56 below. The **element scale factor** option applies a scale factor while the **element abundance** option sets an absolute abundance. In the following example both commands set absolute abundances so the first nitrogen **abundance** will have no effect and the final nitrogen abundance will be the default H II region abundance

```
element abundance nitrogen -4.7
abundances H II region
```

In the following only the second nitrogen scale factor has any effect since the second scale factor overwrites the first:

```
element scale factor nitrogen 3
element scale factor nitrogen 2
abundances H II region
```

The result will be HII region abundances with nitrogen having twice its normal value. Similarly, the combination

```
element abundance nitrogen -4
element scale factor nitrogen 2
```

in either order would result in $n(\mathrm{N})/n(\mathrm{H}) = 2 \times 10^{-4}$ since the first command sets an absolute abundance of 10^{-4} and the second command doubles this.

The chemical composition is printed at the start of the calculation. Be sure to check these abundances to confirm that the composition has been entered correctly.

7.4. abundances he c . . .

The chemical composition is entered with a line beginning with the command **abundances**, followed by: a) a complete set of abundances; b) the keyword **_all** and a single number to set all of the abundances, or c) a second keyword to select one of the stored abundance sets.

7.4.1. Arbitrary abundances

The **abundances** command can be used to specify an arbitrary set of abundances. The elements must be in exactly the same order as indicated in Table 15 unless the order is altered with the **elements read** command described on page 58 below. Abundances for all active elements must be specified, but elements can be turned off with the **elements off** command described on page 57 below. The composition can be specified on several lines with **continue** lines following the initial **abundances** line. Abundances of zero are not allowed; Cloudy will stop if they are entered.

N.B. In the following examples the element symbol is written before its abundance. This only indicates which element has which abundance. The code makes no attempt to read the symbols. The numeric abundances *must* appear in the same order expected by the code. This order can be altered with the **elements read** command described on page 58 below.

The best way to enter abundances is as *absolute abundances*, as in the following example

```
abundances he =-1 li =-9   be =-11 b  =-9 c   =-4.3 n  =-5 o  =-2.3
continue    f =-7 ne =-1.2 na =-3  mg =-8
continue   al =-8 si =-8   p  =-6  s   =-8 cl =-9   ar =-8 k  =-6
continue   ca =-8 sc =-9   ti =-7  v   =-8 cr =-6.3 mn =-6 fe =-8
continue   co =-9 ni =-8   cu =-7  zn =-7
```

The abundances can also be entered as a set of scale factors indicating the desired abundances relative to the current set absolute abundance. These will be solar by default. The following example doubles the oxygen abundance and lowers the iron abundance

```
abundances he =1 li =1 be =1 b  =1 c  =1 n  =1 o  =2
continue    f  =1 ne =1 na =1 mg =1
continue   al =1 si =1 p  =1 s  =1 cl =1 ar =1 k  =1
continue   ca =1 sc =1 ti =1 v  =1 cr =1 mn =1 fe =0.0000001 //deplete iron
continue   co =1 ni =1 cu =1 zn =1;
```

It is better to specify absolute abundances since the default solar composition changes from time to time.

The code checks the sign of all of the entered abundances to decide which style was entered. The numbers are interpreted as linear scale factors if *all* are positive, and as logs of the abundance relative to hydrogen if *any* are negative.

7.4.2. Setting all at once

If the keyword **_all** appears and exactly one number is entered then all of the elements heavier than hydrogen are given this absolute abundance. The number is the log of the abundance if it is less than or equal to zero and the abundance itself if it is positive. Either of the following commands will give all elements between and including helium and zinc an absolute abundance of 10^{-10} by number relative to hydrogen:

```
abundances all -10
abundances all 0.000,000,000,1
```

Table 16
Stored Abundance Sets

	Atom	H II Region	Planetary	Nova	Cameron	Primordial	ISM
	key	HII, H II	PLANetary	NOVA	CAMEron	PRIMordial	_ISM
2	He	0.095	0.10	0.098	0.0677	0.072	0.098
3	Li	5.4(-11)	1.0(-20)	2.05(-9)	2.20(-9)	1(-10)	5.4(-11)
4	Be	1.0(-20)	1.0(-20)	2.62(-11)	4.50(-11)	1(-16)	1.0(-20)
5	B	8.9(-11)	1.0(-20)	7.60(-10)	3.40e-10	-	8.9(-11)
6	C	3.0(-4)	7.8(-4)	9.40(-3)	4.22(-4)	-	2.51(-4)
7	N	7.0(-5)	1.8(-4)	9.80(-3)	8.72(-5)	-	7.94(-5)
8	O	4.0(-4)	4.4(-4)	1.70(-2)	6.93(-4)	-	3.19(-4)
9	F	1.0(-20)	3.0(-7)	3.02(-8)	2.90(-8)	-	1.0(-20)
10	Ne	6.0(-5)	1.1(-4)	2.03(-3)	9.77(-5)	-	1.23(-4)
11	Na	3.0(-7)	1.9(-6)	2.06(-6)	2.25(-6)	-	3.16(-7)
12	Mg	3.0(-6)	1.6(-6)	3.80(-5)	3.98(-5)	-	1.26(-5)
13	Al	2.0(-7)	2.7(-7)	2.95(-6)	3.20(-6)	-	7.94(-8)
14	Si	4.0(-6)	1.0(-5)	3.55(-5)	3.76(-5)	-	3.16(-6)
15	P	1.6(-7)	2.0(-7)	3.73(-7)	3.72(-7)	-	1.6(-7)
16	S	1.0(-5)	1.0(-5)	1.62(-5)	1.88(-5)	-	3.24(-5)
17	Cl	1.0(-7)	1.7(-7)	1.88(-7)	1.78(-7)	-	1.0(-7)
18	Ar	3.0(-6)	2.7(-6)	3.63(-6)	3.99(-6)	-	2.82(-6)
19	K	1.1(-8)	1.2(-7)	1.35(-7)	1.35(-7)	-	1.1(-8)
20	Ca	2.0(-8)	1.2(-8)	2.29(-6)	2.35(-6)	-	4.1(-10)
21	Sc	1.0(-20)	1.0(-20)	1.22(-9)	1.16(-9)	-	1(-20)
22	Ti	5.8(-10)	1.0(-20)	8.60(-8)	9.00(-8)	-	5.8(-10)
23	V	1.0(-10)	1.0(-20)	1.05(-8)	9.50(-9)	-	1.0(-10)
24	Cr	1.0(-8)	1.0(-20)	4.84(-7)	4.80(-7)	-	1.0(-8)
25	Mn	2.3(-8)	1.0(-20)	3.42(-7)	3.50(-7)	-	2.3(-8)
26	Fe	3.0(-6)	5.0(-7)	4.68(-5)	3.38(-5)	-	6.31(-7)
27	Co	1.0(-20)	1.0(-20)	2.24(-9)	8.27(-8)	-	1.0(-9)
28	Ni	1.0(-7)	1.8(-8)	1.76(-6)	1.80(-6)	-	1.82(-8)
29	Cu	1.5(-9)	1.0(-20)	1.87(-8)	2.00(-8)	-	1.5(-9)
30	Zn	2.0(-8)	1.0(-20)	4.52(-8)	4.70(-8)	-	2.0(-8)
	grains?	Orion	AGB	no	no	no	ISM

The **metals** command, described on page 66 below, will set abundances of all elements heavier than helium.

7.4.3. Stored abundance sets

Table 16 lists the abundance sets that are stored as a permanent part of the code. These sets are entered if there are no numbers on the **abundances** command, but a keyword occurs, as in the following examples. The four-character part of the keyword that must be matched for the key to be recognized is capitalized in the second line of the table.

```
abundances Cameron
abundances HII region [no grains]
abundances h ii region [no grains]
abundances nova
abundances planetary nebula [no grains]
abundances primordial
```

The assumed abundances are from a variety of sources. Table 16 gives their present values.

Cameron These are from Cameron (1982). Note that the helium abundance is *very* low, either it or the Big Bang is wrong.

nova These are roughly those derived by Ferland & Shields (1978) for the classical nova V1500 Cygni. Abundances close to solar are assumed for those they did not measure.

H II region The H II region abundances are a subjective mean of the Orion Nebula abundances determined by Baldwin et al. (1991), Rubin et al. (1991), and Osterbrock et al. (1992). The grains are the large-R grains described by Baldwin et al. (1991) and updated by van Hoof et al. (2004). The keywords **HII region**, **H II region**, or **Orion** can be used to obtain this abundance set. Abundances of some rare species were taken from the ISM mix of Savage and Sembach (1996). Abundances entered in the table as "1E-20" are not real values, simply values chosen to be small enough to be of no consequence. I would appreciate learning about better numbers.

planetary nebula These abundances are from Aller and Czyzak (1983) and Khromov (1989), with high depletions assumed for elements they do not list. The grains are from unpublished work of Kevin Volk on post-AGB stars. The application of this data to old planetary nebulae is dicey at best — evidence summarized by Clegg and Harrington et al. (1989) suggest that some PNs have dust to gas ratios roughly ten times smaller than ISM. However, Mallik and Peimbert (1988) find dust to gas ratios similar to ISM and Borkowski and Harrington (1991) find one object with a dust-to-gas ratio an order of magnitude above ISM. Abundances entered as "1E-20" are not real values, simply values chosen to be small enough to be of no consequence. I would appreciate learning about better numbers.

_ISM The ISM mixture is an average from the work of Cowie and Songaila (1986) for the warm and cold phases of the interstellar medium, together with numbers from Table 5 for the warm and cool phases towards ξ Oph (Savage and Sembach 1996). The oxygen abundance[12] is from Meyers et al. (1998). The grains are the default interstellar medium grains. The ISM and Orion abundances are not the same, although one can make the case that they should be.

old solar 84 The composition will be the solar used in versions 84 - 94 of the code. These are defined in Table 15 and were taken from the meteoritic abundances of Grevesse and Anders (1989) with extensions by Grevesse and Noels (1993). All three of the keywords **old solar 84** must appear.

---

[12] The ISM O abundance was changed from 5.01 to 3.19×10^{-4} in version 95.

7.4.4. Grains, gas-phase depletions, and quantum heating

Certain elements, especially Si, Ca, Al, Mg, and Fe, are heavily depleted onto grains in the ISM. The abundance sets specified by the **h_ii region**, **_ISM**, or **planetary nebula** keywords will include grains and the gas-phase mixtures given in Table 16. Grains set in this manner will have the properties appropriate for the type of grains indicated (the bottom line of the table, and the section on grains below). Grains can also be specified separately with the **grains** (page 60 below) command.

In some circumstances it is interesting to explore the effects of grain-free mixtures, with the opacity and thermal effects of the grains suppressed, but with the gas-phase abundances left with their depleted values. The optional keyword **no grains** can be placed in the **abundances** line. In this case grains will not be included in the calculation, but the observed (depleted) gas-phase abundances will still be used.[13] This is, of course, not self-consistent.

Quantum heating will be considered for all grain species where it will be important. Quantum heating only affects the Wien tail of the grain thermal-emission spectrum. This process can be turned off with the **no qheat** option on the **abundances** command if this part of the continuum is not of interest.

Table 17 Old Solar Composition

				Solar	
1	H	Hydrogen	1	0.00	12.00
2	He	Helium	0.1	-1.00	11.00
3	Li	Lithium	2.04E-09	-8.69	3.31
4	Be	Beryllium	2.63E-11	-10.58	1.42
5	B	Boron	7.59E-10	-9.12	2.88
6	C	Carbon	3.55E-04	-3.45	8.55
7	N	Nitrogen	9.33E-05	-4.03	7.97
8	O	Oxygen	7.41E-04	-3.13	8.87
9	F	Fluorine	3.02E-08	-7.52	4.48
10	Ne	Neon	1.17E-04	-3.93	8.07
11	Na	Sodium	2.06E-06	-5.69	6.31
12	Mg	Magnesium	3.80E-05	-4.42	7.58
13	Al	Aluminium	2.95E-06	-5.53	6.47
14	Si	Silicon	3.55E-05	-4.45	7.55
15	P	Phosphorus	3.73E-07	-6.43	5.57
16	S	Sulphur	1.62E-05	-4.79	7.21
17	Cl	Chlorine	1.88E-07	-6.73	5.27
18	Ar	Argon	3.98E-06	-5.40	6.60
19	K	Potassium	1.35E-07	-6.87	5.13
20	Ca	Calcium	2.29E-06	-5.64	6.36
21	Sc	Scandium	1.58E-09	-8.80	3.20
22	Ti	Titanium	1.10E-07	-6.96	5.04
23	V	Vanadium	1.05E-08	-7.98	4.02
24	Cr	Chromium	4.84E-07	-6.32	5.68
25	Mn	Manganese	3.42E-07	-6.47	5.53
26	Fe	Iron	3.24E-05	-4.49	7.51
27	Co	Cobalt	8.32E-08	-7.08	4.92
28	Ni	Nickel	1.76E-06	-5.75	6.25
29	Cu	Copper	1.87E-08	-7.73	4.27
30	Zn	Zinc	4.52E-08	-7.34	4.66
		grains?		no	

7.4.5. Interactions between the abundances and grains commands

It is possible to include grain species with the **abundances** command, described here, and with the **grains** command, described on page 60 below. Commands that include more grains do not turn off grains that were previously included. For instance the following would include two sets of Orion grains:

```
abundances Orion
grains Orion
```

The first command would turn on the Orion abundances and grains, and the second would turn on a second set of Orion grains. You should always check that the

---

[13]In versions 77 and before, the abundances of depleted elements were set to solar values when "no grains" was set.

intended grains have been included. The intended grain mixture could have been produced with the following set of commands:

```
abundances Orion no grains
grains Orion
```

Another way to do this would be simply

```
abundances Orion
```

The **abundance** command will only reset grains set by previous **abundance** commands (unless there was a **grain** command between the two, in which case it does nothing). A preceding **grain** command will be overridden by the grains from the **abundance** command. A trailing **grain** command will add to the grains already set by the **abundance** command (however, in the grain case only if they differ from the ones already set). It is best to avoid this confusion by only including a single set of grains.

7.5. abundances starburst, Z=10

This form of the **abundances** command interpolates on Fred Hamann's grid of abundances for an evolving starburst in a massive galactic core. The chemical evolution model is more fully described by Hamann and Ferland (1993). This grid is model M5a of that paper. It uses a star formation rate and infall timescales very close to, but slightly faster than, the "standard" elliptical model (see also Arimoto and Yoshii 1987; Matteucci and Greggio 1986; Matteucci and Tornambe 1987; Bica 1988). Its IMF also has a slope very similar to, but slightly steeper than ($x = 1.0$ instead of 1.1), that of the standard elliptical model. The main difference is that the IMF has a lower mass cutoff at $M = 2.5\ M_\odot$ instead of $\sim$0.1 $M_\odot$ in the standard models. This allows the gas to reach much higher metallicities before the gas is locked up in low-mass stellar remnants.

One number, the metallicity of the gas relative to solar, must appear on the line. It is interpreted as the log of the metallicity if it is less than or equal to zero, and the linear metallicity if positive. The keywords **_log** or **linear** may appear on the line and will force the number to be interpreted appropriately. The limits to the range of possible metallicities are $10^{-3}Z_\odot$ and 36 $Z_\odot$.

The keyword **trace** will result in a printout of the abundances of all elements as a function of metallicity, between these limits. The code will then stop.

7.6. element name [scale, abundance, _off, _log, table]

This command allows the abundance of a particular element to be changed, without specifying the abundances of the other elements.

7.6.1. The name of the element

The "name" must be at least the first four characters of the name of the element as spelled in Table 15. One of the keywords **scale**, **abundance**, **ionization**, or **_off** must appear on the line.

7.6.2. element name scale

If the keyword **scale** appears then the number on the line is interpreted as a scale factor multiplying the current abundance of the element. The scale factor will be interpreted as a linear scale factor if the number is positive or if the **linear** keyword appears, and as the log of the scale factor if the number is negative. If the key **_log** appears (note the leading space) then the scale factor is interpreted as a log irregardless of its sign.

7.6.3. element name abundance

If **abundance** appears then the number is the log of the absolute abundance of the element, by number relative to hydrogen. The number is the log of the abundance unless the **linear** keyword appears.

7.6.4. element name ionization

This allows the ionization distribution of an element to be set. Each number is the ionization fraction, $n(A^{+i})/n(A)$, for successive stages of ionization. The code will scan off up to $A+1$ numbers, where A is the atomic number of the element. If any numbers are negative then all are interpreted as logs of the ionization fraction. If there are fewer than $A+1$ numbers then the missing stages of ionization are assumed to have zero abundance.

This command does not confirm that the sum of the ionization fractions is unity. The abundance of each stage of ionization is set to this ionization fraction times the total gas-phase abundance. If the fractions do not add up to unity the effect will be to change the total abundance of the element.

This only sets the ionization distribution of atoms and ions of a particular element and does not specify its chemical state. The H_2 fraction can be set with the **set H2** command, described on page 188 below. Chemistry will be disabled if the ionization distribution of an element that is part of the chemical network. It will not be possible to obtain a chemical solution if the abundances of the atom and first ion cannot respond to changes in the chemistry.

All of this is unphysical and is only intended as a way to test the code.

7.6.5. element name off

If the keyword **_off** appears (note the leading space) then the element is not included in the calculation. The ionization equilibrium, opacity, and cooling due to the element will not be computed. The abundance must still be specified when the **abundances** command (page 51 above) is used unless the element is excluded with the **elements read** command (page 51 below).

The keyword **_on_** will include an element that was previously turned of in the same input stream

Saving time is the main reason to turn an element off. This is especially true for third- and fourth-row elements. These take longer to compute because of the large number of inner shell electrons. They often have negligible effects on the thermal and ionization structure because of their low abundances.

The code dynamically allocates memory for the species that are present when the core load is initialized. When the code is used to compute a large grid of models it is not possible to turn on an element that was turned off for the first model since the needed memory does not exist. It is fine to turn off an element in later models since the element is simply not computed. In later calculations, the code will ignore any attempt to turn on an element that was initially turned off.

The chemical solution may become destabilized if an element that forms molecules is turned off. Do not turn off elements that are part of the chemistry if the calculation extends into a PDR or molecular cloud. The code will generate a warning if this is done.

7.6.6. *elements read*

Normally Cloudy expects the abundances entered with the **abundances** command to occur in exactly the same order as the atomic numbers of the elements. These begin with helium and include all elements through zinc. The **elements read** command makes it possible to change this order or to leave certain elements out entirely. The command begins with the line **elements read**, follows with a list of elements, and ends with a line beginning with the keyword **end**.

This order will be used for all **abundances** commands that follow this command. If an element does not occur in the list its value cannot be set by an **abundances** command. This does not turn an element off – this is done with the **element name off** command, described on page 57 above.

The following example shows an input stream that changes the **abundances** command so that it behaves more like version 84. The elements included in this list were only those present in that version. The **elements read** command limits the number the elements to be entered, then the missing elements are turned off with the **element off** command.

```
elements read
helium
carbon
nitrogen
oxygen
neon
sodium
magnesium
aluminum
silicon
sulphur
argon
calcium
iron
nickel
end of elements
c
element Lithium off
element Beryllium off
element Boron   off
element Fluorine   off
element Phosphor off
element Chlorine off
element Potassium off
element Scandium   off
element Titanium off
element Vanadium off
element Chromium off
element Manganese off
```

```
element Cobalt off
element Copper off
element Zinc  off
```

In summary: Turning an element off with the **element off** command does not remove that element from the list of abundances entered with the **abundances** command. Leaving an element out of this list with the **elements read** command does not turn off that element.

7.6.7. *element name table*

If the keyword **table** appears on the **elements** command then the code will read in a list of position-dependent abundances for a particular element. This might, for instance, be used to model variable depletions. The following is an example.

```
element carbon table depth
-30 -4
3 -4
5 -3
7 -2
9 -1
end of table
```

The first number on each line is the log of the radius (the default) or depth (if the keyword **depth** also appears on the **element** line). Depth and radius are defined on page 8 above. The second number is the log of the abundance of the element at that point, by number relative to hydrogen. The table ends with a line starting with the keyword **end**. Up to 500 pairs may be entered. This command always specifies the absolute abundance and not the scale factor.

The chemical composition printed at the start of the calculation is always the composition at the illuminated face of the cloud. If the table gives composition as a function of radius, the composition will be evaluated at the inner radius of the cloud. If the table gives the composition as a function of depth, then the composition will be evaluated as a depth of 10^{-30} cm. The table must extend to this depth, as in the example above.

7.7. fluctuations abundances, period, max, min, phase

This makes the metallicity vary with radius as a sine wave. This is designed to investigate the effects of chemical inhomogeneities upon the emission-line spectrum, and was implemented to search for solutions to the t^2 problem (Kingdon & Ferland 1995).

The first number is the log of the period P of the sine wave, in centimeters. The second two numbers are the logs of the largest and smallest metallicities over the sine wave and have the same effect as the metals-scaling factor entered with the **metals** command (page 66 below).

The **fluctuations** command is more fully described in the description of the density version, on page 71 below.

7.8. grains [abundance, options]

7.8.1. Overview

Grains have several effects on a photoionized nebula (see AGN3, Chapter 7). Refractory elements are depleted from the gas phase, thus removing coolants from the gas (Kingdon, Ferland, & Feibelman &1995). Grains absorb the incident continuum, and radiate an infrared continuum (Bottorff et al. 1998). Higher-energy photons ionize the grains, establishing a net charge, and so affecting the charge balance of the gas (Baldwin et al. 1991). Finally several processes partially couple the temperatures of the gas and dust (van Hoof et al. 2004). This treatment of this physics was developed in collaboration with Peter van Hoof, Peter G. Martin, and Joe Weingartner, and Peter van Hoof did the majority of the coding. Reviews are given by Spitzer (1948; 1978), Martin (1979), and Weingartner & Draine (2001b), and described further in a section of Part 2 of HAZY.

The heating and cooling of the grains and gas are done self-consistently. Grains are heated by direct absorption of the incident continuum and the radiation emitted within the cloud, and by gas collisions. The balance between heating and cooling by collisions with the gas, the thermionic effect, and by radiation, establish the temperature for each grain size and type. Grains affect the gas temperature by heating, mainly by the grain photoelectric effect and the thermionic effect, and by cooling, mainly by free-particle capture onto the grain surface.

The grain charge distribution is resolved into the number of discrete integral charge states set by the **set nchrg** command. The default is two charge states. The population of each of charge state is determined self-consistently by solving the grain ionization-recombination balance equations as described in van Hoof et al. (2004).

Grains are not included in the default solar mixture. The effects of grains can be included, either with this **grains** command or by using an abundance mixture that includes grains. The grains set with the **abundances** command (see page 51 above) have properties that are described in that section. These properties cannot be modified. The **abundances** command has a **no grains** option, so that grains will not be included with that mixture. This makes it possible to specify a population of grains with this command, which has many options. It is wise to only specify grains with either the **grains** or the **abundances** command.

Real interstellar grains are known to have a distribution of sizes, sometimes approximated as a power law (the so-called MRN distribution, Mathis et al. 1977). The **grains** command has the ability to resolve this (and other) size distributions into any number of size bins and treat these either classically or with single-photon heating. A grain's temperature, potential, and photoelectric heating, all depend on its diameter. Resolving the size distribution can lead to significant changes in the emitted spectrum in the thermal IR, especially at the shortest wavelengths.

There are several **punch** commands that will output predicted properties of the grains. The **punch grains** command (page 140 below) will give some additional information about the grain properties. The **punch continuum** (page 131 below)

and **punch spectrum** (page 137 below) commands will ouput the infrared continuum emitted by the gas and grains.

7.8.2. Using the "built-in" grain types

The populations of grains summarized in Table 18 are included in the code's data files. The table gives the type of grain material, the size distribution, and the name of the grain opacity file. In most cases these standard grain types will be sufficient – nothing more needs to be done to set up the grains. Two groups are described next.

grains _ISM specifies grains with a size distribution and abundance appropriate for the ISM of our galaxy. They reproduce the observed overall extinction properties for a ratio of extinction per reddening of $R_V = 3.1$. This is the default and will be used if no keywords occur on the **grains** command. If either keyword **graphite** or **silicate** also appears, then only that grain type is included. Both species are included if neither keyword appears.

grains Orion specifies grains with a size distribution and abundance appropriate for those along the line of sight to the Trapezium stars in Orion. The Orion-size distribution is deficient in small particles, and so produces the relatively grey extinction observed in Orion. If either keyword **graphite** or **silicate** appears, then only that grain type is included. Both species are included if neither keyword appears.

Table 18 Standard Opacity files

Keyword	Filename	Type	Size distribution
	graphite_0m010.opc	graphite	Single 0.01 μm
	graphite_0m100.opc	graphite	Single 0.1 μm
	graphite_1m000.opc	graphite	Single 1 μm
ISM single graphite	*graphite_ism_01.opc*	graphite	Unresolved ISM
ISM graphite	*graphite_ism_10.opc*	graphite	ISM, 10 bins
Orion single graphite	*graphite_orion_01.opc*	graphite	Unresolved Orion
Orion graphite	*graphite_orion_10.opc*	graphite	Orion, 10 bins
grey	*grey_ism_01.opc*	grey	Unresolved ISM
grey	*grey_ism_10.opc*	grey	ISM, 10 bins
	pah1_0n341.opc	PAH	0.00034 μm
	pah1_0n682.opc	PAH	0.00068 μm
PAH single	*pah1_bt94_01.opc*	PAH	Unresolved BT94
PAH	*pah1_bt94_10.opc*	PAH	BT94, 10 bins
	silicate_0m010.opc	silicate	Single 0.01 μm
	silicate_0m100.opc	silicate	Single 0.1 μm
	silicate_1m000.opc	silicate	Single 1 μm
ISM single silicate	*silicate_ism_01.opc*	silicate	Unresolved ISM
ISM silicate	*silicate_ism_10.opc*	silicate	ISM, 10 bins
Orion single silicate	*silicate_orion_01.opc*	silicate	Unresolved Orion
Orion silicate	*silicate_orion_10.opc*	silicate	Orion, 10 bins

grains PAH see page 62 below.

The following example sets ISM abundances and uses Orion silicates with abundances twice the ISM value and ISM graphite with default ISM abundances.

```
// use ISM abundances but DO NOT include ISM default grains
abundances ISM no grains
// Orion silicate with twice the normal abundance
grains Orion silicate 2
// ism graphite with size resolution
grains ISM graphite
```

7.8.3. Creating your own grain types

New types of grains may be introduced by first generating data files that specify the refractive indices for the full energy range considered by the code and the size distribution. These are then converted to opacities with the compile grains command. The process is described further where the compile grains command is discussed (page 172 below). The result is a new *opc* file.

The code will read an opacity file whenever a double quote (") occurs anywhere on the command line. If the quote is found the code will then look for the name between a pair of quotes, as in "**special.opc**", and will stop if the file cannot be found or if the second quote is missing. If the file exists then these opacities will be used. So don't place an extra quote on the command line unless there is a pair of quotes surrounding a filename since the code will stop. The contents of that file will determine whether the calculations are size-resolved or not, irrespective of the keywords **distribution** or **single**.

7.8.4. PAHs

PAH's are not included by default but are added with the **grains PAH** command. A detailed treatment of the physics of PAH's is implemented, including photoelectric heating and collisional processes as discussed by Weingartner & Draine (2001), and stochastic heating effects following Guhathakurta & Draine (1989). A power-law distribution of PAH sizes with 10 size bins (Bakes & Tielens 1994, hereafter BT94), and two single-sized PAH's, are included in the grain opacity sets. BT94 is the default and will be used if no further options occur on the command line. If the keyword **single** also appears then an unresolved PAH with their mean properties is used.

The PAH opacity functions were derived by Kevin Volk from a variety of sources. The original opacity function is from Desert, Boulanger, & Puget (1990) and Schutte, Tielens, & Allamandolla (1993). Kevin adapted the opacitis from these papers to agree with the infrared plateaus seen in the Orion Bar (Bergmann et al. 1988). The optical/far-UV opacity values have sharp cut-offs with exponential declines (these are for <912 Angstroms and >8000 Angstroms respectively).

The command **grains PAH C15** will include a single small PAH with 15 carbon atoms per molecule, with an abundance relative to hydrogen of $n(\mathrm{PAH})/n(\mathrm{H}_{tot}) = 1.986 \times 10^{-7}$, corresponding to $n(\mathrm{C})/n(\mathrm{H}_{tot}) = 2.979 \times 10^{-6}$. The command **grains PAH C120** will include a large PAH with 120 carbon atoms per

molecule and an abundance relative to hydrogen of $n(\text{PAH})/n(\text{H}_{tot}) = 2.483 \times 10^{-8}$, also corresponding to $n(\text{C})/n(\text{H}_{tot}) = 2.979 \times 10^{-6}$.

PAHs appear to exist mainly near the surfaces of molecular clouds. Apparently PAHs are destroyed in ionized gas (Sellgren et al. 1990) by ionizing photons and by collisions with ions (mainly H$^+$) and are depleted in molecular regions, possibly due to coagulation into larger grains. The code assumes that the PAH abundance scales with the ratio $n(\text{H}^0)/n(\text{H}_{tot})$. This produces very few grains in ionized and fully molecular gas, but the grain species will have its default abundance when the gas is atomic. This is consistent with Sellgren's observations of the Orion Bar.

The **set PAH** command (page 191 below) makes it possible to specify several other laws describing how PAH abundances depend on physical conditions. Note that if PAHs have a significant abundance in predominantly molecular gas then they will soak up nearly all of the free electrons. This is not observed and may destabilize the chemistry network.

7.8.5. Grain abundances

The first number on the **grain** command gives the abundance relative to its standard abundance. This is interpreted as a linear scale factor if it is positive and as a log if the number is zero or negative. For example, both **grains -2** and **grains .01** would use ISM grains with the constituents having only 1 percent of the standard abundance. The keywords **_log** and **linear** will force the code to interpret the factor as either the log or a linear factor.

H II region grain abundances may be close to the ISM value, the default. Planetary nebula grain abundances are quite uncertain. Clegg and Harrington (1989) find dust-to-gas ratios below the ISM value, while Borkowski and Harrington (1991) find a dust-to-gas ratio an order of magnitude above ISM in a hydrogen-deficient planetary nebula. Mallik and Peimbert (1988) find dust-to-gas ratios roughly equal to the ISM in a sample of PNs. Stasinska & Szczerba (1999) discuss properties of a sample of PNe. In view of this scatter, the grain abundance of PNe should probably be treated as a free parameter.

7.8.6. variable grain abundances

The **function** option on the **grains** command makes it possible to vary the abundance of any grain species across a cloud. This option works by setting the local abundance of a grain species to the product of an intrinsic abundance and the value returned by the function *GrnVryDpth*. That routine can be modified by the user to specify a scale factor that might depend on other physical conditions or location. All the other multipliers for the grain abundance are still in effect when the keyword **function** is used. This means that the abundance multiplier supplied with the **grains** command is also included, as well as the factor supplied in the **metals grains** command. So the abundance of the grain is given by the default abundance multiplied with the two numbers mentioned here times whatever *GrnVryDpth* returns.

If the **grains** command sets the abundance of a single grain species then the **function** option will only apply to that particular species. If the option occurs on a

grains command that specifies more than one species of grains (as in the **ISM** keyword) then all species enabled by that command are affected. This is probably unphysical. However, it is possible to define a different behavior of the grain abundance in routine *GrnVryDpth* for each individual grain size bin. The **abundances** command does not have the **function** option.

The code does not attempt to conserve the mass of the grain constituents. The gas phase abundances are not automatically changed where the grains are changed. This can be done by entering abundances with a depth-dependent table of abundances using the **elements read** command (see page 58 above).

7.8.7. Resolved or averaged grain size distributions

By default the **grains** command will resolve the size distribution into ten size bins. If the keyword **single** appears then the grains will have properties determined by averaging over the entire size range. If the keyword **distribution** appears then the code will use the size-resolved grain. This is the default and will be used if no keyword appears. The **single** option may save some machine time, but will give a less realistic representation of the grain physics since many grain properties are strongly size dependent. In particular, the photoelectric heating of the gas can be underestimated since the smaller grains are the most efficient.

7.8.8. Grain heating and cooling

The keyword **no heating** will turn off heating of the gas by grain photoelectric emissions. The keyword **no cooling** will turn off all forms of cooling of the gas by collisions with the grains. Either violates energy conservation, of course.

7.8.9. Forcing quantum heating on or off

By default, quantum (also called "stochastic" or "single-photon") heating is included for size-resolved species when it is significant. Guhathakurta and Draine (1989) describe the formalism used here. The method was originally implemented by Kevin Volk and subsequently revised and generalized by Peter van Hoof.

Quantum heating is considered for all species except for the unresolved size distributions. To save time, quantum heating is only treated when the grain cooling time is sufficiently short compared with the time between heating events. Quantum heating is considered for a given size bin if the ratio of the volume of the largest and the smallest grain in that particular bin is less than 100. This means that quantum heating is considered for all of our resolved distributions, as well as all single-sized grains, but not for unresolved distributions (except the unresolved BT94 distribution, where it *is* used by default).

In the zone printout an asterisk will appear next to the name of a grain if quantum heating is important.

The keyword **qheat** will force quantum heating to always be considered for species where it is important. Since this is the default for all resolved size distributions, this keyword is not needed. The keyword **no qheat** will disable quantum heating for an individual **grains** command. Grain species with quantum heating enabled and disabled can be mixed.

Quantum heating can be turned off for all species with the **no grain qheat** command described on page 180 below.

7.8.10. Grains no reevaluate

The **no reevaluate** option on the **grains** command will speed up a calculation by not reevaluating grain quantities all the time. Roughly a 30% speedup can be achieved. This is potentially dangerous and can destabilize a solution and/or lead to wrong results. It should be used under certain experimental circumstances and not in a true simulation of a cloud.

7.8.11. Interactions between abundances and grains commands

Be careful when specifying grains with more than one **grains** command, with the **abundances** command, and with a combination of these. The order in which the commands appear in the input stream does make a difference. These commands have the following precedence:

- An **abundances** command will override all grains set with a previous **abundances** command.

- An **abundances** command will NOT override any grains set with a previous **grains** command, instead it will implicitly behave as if the option "**no grains**" was given on the previous command.

- A **grains** command given after an **abundances** command or another **grains** command will add to the grains already included, even if it would cause the same opacity file to be read twice. A warning will be printed if the latter occurs. For instance, the following would add the ISM grains twice

```
abundances ism
grains ism
```

The following would result in the Orion gas-phase composition but ISM grains

```
abundances Orion no grains
grains ism
```

This would result in ISM grains and composition

```
grains ism
abundances Orion
```

The following would result in Orion gas-phase abundances and both ISM and Orion grains

```
abundances Orion
grains ism
```

To avoid confusion it is best to only specify grains with only the default set with the **abundances** command, or by including "**no grains**" on the **abundances** command and then setting them with explicit **grains** commands. In any case, be sure to check the resulting output to verify that things are set properly.

7.8.12. Line intensities with grains.

For a closed geometry, in which the **sphere** option is set, the predicted emission-line spectrum will be the *intrinsic* spectrum of the nebula. Photon destruction by all background opacity sources (including grains) is fully treated (i.e., Hummer 1968),

and the predicted intrinsic intensities include this destruction. The intensities *do not* include the reddening effects of any grains that lie outside the line-forming region, however. Note that, for a true sphere, lines must escape through the hydrogen ionization front and past whatever dust and neutral gas may exist.

For an open geometry this same intrinsic emission-line spectrum is printed as the second block of lines. The first block of lines would be the observed spectrum if the geometry were observed from the illuminated face of the cloud assuming that the ionized gas has a large molecular cloud behind the ionization front. The emergent line intensities include contributions from absorption by grains, and reflection off the molecular cloud.

The effects of grains external to the emission-line region are very difficult to model, since they are very geometry dependent. The best approach is to correct the observed spectrum for reddening to obtain an intrinsic spectrum, and to then compare this intrinsic spectrum with that computed by the code.

This is discussed further in the section *Line intensities in a dusty open geometry*, in the chapter *Observed Quantities*, in Part 3 of HAZY.

7.8.13. Extinction for point and extended sources

Grain extinction is given by the cross section [cm^2] per H nucleon: $\sigma = \kappa / n(H)$, where κ [cm^{-1}] is the opacity due to grains and $n(H)$ [cm^{-3}] is the local density of H in any form. The extinction includes true absorption and scattering. The scattering opacity depends on the geometry of the absorbing cloud, as described next.

Light scattering off grains is not isotropic. The angular dependence is often approximated by the Henyey-Greenstein function. The scattering theory predicts the fraction of scatterings, given by the grain asymmetry factor g, that are "forward scattering", that is, only change the direction of the scattered photon by a small amount. Rather than the total scattering cross section σ_s an effective scattering cross section $\sigma_{scat} = \sigma_s (1-g)$ is used when extended sources are studied. This discounts light scattered nearly into the forward direction. The asymmetry parameter g approaches unity at high and low energies, particularly for larger grains, so that σ_{scat} becomes much less than α_{abs}.

For a point source such as a star even a small deflection of starlight by forward scattering removes light from the ray and so counts as extinction. In this case the total grain opacity is simply $\sigma_{abs} + \sigma_s$. This is referred to as the point source extinction. For an extended source such as a diffuse cloud the loss of photons by small-angle scattering will be compensated by a similar gain of photons from rays that are nearly parallel, so that total opacity is $\sigma_{abs} + \sigma_s(1-g)$. This is referred to as extended source extinction. The code keeps track of both since a beam of light passing through an H II region is attenuated by the extended extinction, while the observational literature will quote a point source extinction.

7.9. metals 0.05 [_log, linear, grains; deplete]

This command multiplies the abundances of the entire mixture of metals (elements heavier than helium) by the scale factor entered on the line. This is useful

when the effects of global enrichments or depletions of the elements are to be investigated. If the number is less than or equal to zero it is assumed to be the log of the scale factor and the linear scale factor if it is positive. The **linear** and **_log** keywords force that interpretation of the number.

Combinations such as

```
abundances planetary nebula
metals 3
```

or

```
metals 3
abundances planetary nebula
```

would multiply the planetary nebula gas-phase abundances by three,[14] while

```
metals -10
```

would multiply the default solar mixture by 10^{-10}.

7.9.1. Scaling grains and metals together

It seems likely that the grain to hydrogen ratio scales with the total gas-phase metallicity. The optional keyword **grains** on the **metals** command causes the grain abundance to also be scaled by the factor on the line. The basic assumption is that the grain to metals ratio does not depend on metallicity while the grain to gas (hydrogen) ratio depends linearly on the metallicity. It is still necessary to include grains with either the **grains** command or by specifying a chemical composition that contains grains (with the **abundances** command). The scale factor that appears on the **metals** command will further multiply the grain abundance specified on the **grains** command. That is, the combination

```
grains .5
metals and grains .5
```

(in any order) will result in a grain abundance that is a quarter of the default and a metallicity that is half of solar.

In the following example the ISM gas phase *and* grain abundances are each increased by a factor of two over their default values;

```
abundances ISM
metals and grains 2
```

7.9.2. Gas-Phase Depletion Factors

It is possible to alter an existing set of abundances by depletion onto grains. In the ISM the observed depletion is a function of the gas density (Spitzer 1985 models this), so there is not really a universal depletion pattern. A set of scale factors that are roughly those appropriate for relatively dense ISM gas ($\sim$1 cm^{-3}) is built into the code. Table 19 lists the depletions that will be assumed if the keyword **deplete** occurs on the **metals** command, but no numbers are on the line. These are loosely

---

[14]Limits to the ordering of the **abundances** and **metals** commands existed before version 72 but have been lifted.

based on the depletions listed by Jenkins (1987) and Cowie and Songaila (1986). This table is obviously incomplete and I would appreciate learning of better references.

This command can be combined with commands that specify abundances, and the **grains** command, to specify an arbitrary set of grain and gas-phase abundances. Specifying grains by themselves (with the **grains** command) does not change the gas-phase abundances, which is not self-consistent. The code will complain if you do this, but still perform the simulation.

The following is an example of using the nova abundance mixture, depleting it with this command, and then setting ISM grains.

```
abundances nova
metals deplete
grains
```

These commands do not attempt to conserve mass. In particular, the grain mass will be less than the mass of depleted heavy elements because the nova mixture has enhanced CNO, while the ISM grains have only a corresponding solar depletion of missing material (see, however, Snow & Witt 1996, who show that even this is not true).

Note that Tables 19, 16, and 15 are not self-consistent since they come from different sources.

	Table 19 Depletions	
	Depl	Reference
He	1.00	noble gas
Li	0.16	White 1986
Be	0.6	York et al 1982
B	0.13	Federman et al 1993
C	0.4	
N	1.	
O	0.6	
F	0.3	Snow and York 1981
Ne	1.0	noble gas
Na	0.2	
Mg	0.2	
Al	0.01	
Si	0.03	
P	0.25	Cardelli et al 1991
S	1.0	
Cl	0.4	
Ar	1.0	noble gas
K	0.3	Chaffee& White 1982
Ca	1(-4)	
Sc	5(-3)	Snow, Dodger 1980
Ti	8(-3)	Crinklaw et al 1994
V	6(-3)	Cardelli 1994
Cr	6(-3)	Cardelli et al 1991
Mn	5(-2)	Cardelli et al 1991
Fe	1(-2)	
Co	1(-2)	
Ni	1(-2)	
Cu	0.1	Cardelli et al 1991
Zn	0.25	Cardelli et al 1991

8. DENSITY LAWS

8.1. Overview

Hydrogen plays a fundamental role in any astrophysical plasma because of its large abundance, and so the hydrogen density [cm^{-3}] is a fundamental parameter. Commands that specify how the hydrogen density changes with radius or depth are described in this section. Constant density is the default. In this case the total hydrogen density (the sum of the protons in atomic, ionic, and molecular form, given by the command **hden** described on page 73 below) is kept constant. Many other density or pressure distributions can also be computed.

8.2. constant density, pressure, gas pressure

This command specifies how the density changes across the cloud. The **hden** command usually specifies the initial hydrogen density. The **constant xxx** command has several optional keywords, depending on what is to be held constant. These are described next.

8.2.1. constant density

This is the default. The hydrogen density, the sum

$$n(\text{H}) = n(\text{H}^0) + n(\text{H}^+) + 2n(\text{H}_2) + \sum_{other} n(\text{H}_{other}) \quad [\text{cm}^{-3}] \tag{26}$$

is kept constant. This is not quite an isochoric density law because the total particle density is not constant — the electron and molecular fractions can vary with depth. I prefer this type of model because the homology relations with the ionization parameter (Davidson 1977) are preserved. The hydrogen nucleon density is set with the **hden** command (page 73 below).

8.2.2. constant gas pressure [index =-1]

An isobaric density law is specified with this command. The gas pressure

$$P_{gas} = n_{tot} k T_e \quad [\text{dyne cm}^{-2}] \tag{27}$$

where n_{tot} is the total particle density [cm^{-3}], is kept constant. The optional index α will force the pressure to change as a power-law of the radius;

$$P_{gas}(r) = P_{\text{o}} \left(\frac{r}{r_{\text{o}}} \right)^{\alpha} \quad [\text{dyne cm}^{-2}] \tag{28}$$

where P_{o} is the pressure at the illuminated face of the cloud.

8.2.3. constant pressure [no continuum, no abort]

This holds the total pressure constant. This includes ram, magnetic, turbulent, particle and radiation pressure. The equation of state is given by

$$\begin{aligned} P_{tot}(r) &= P_{tot}(r_{\text{o}}) + \int a_{rad}\, \rho\, dr \\ &= P_{gas} + P_{lines} + P_{continuum} + P_{ram} + P_{turb} + P_{magnetic} \end{aligned} \quad [\text{dyne cm}^{-2}] \tag{29}$$

where a_{rad} is the radiative acceleration [cm s⁻²] due to the attenuated incident continuum and ρ is the density (gm cm⁻³).

The terms are the following: $P_{gas} = n_{tot}kT_e$ is the thermal gas pressure. The outward force due to the attenuation of the incident radiation field is given by the integral and referred to as $P_{continuum}$. The **no continuum** option will turn this off. The ram pressure is P_{ram}, $= \rho u_{ram}^2 / 2$ where u_{ram} is the flow velocity. This is only present for a wind geometry (see page 81 below). The turbulent pressure is $P_{turb} = P_{turb} = \rho u_{turb}^2 / 2$ where u_{turb} is the turbulent velocity as set with the **turbulence** command (page 100 below). The magnetic pressure is $P_{mag} = B^2 / 8\pi$ where the magnetic field B is set with the **magnetic field** command (page 108 below). Both magnetic and turbulent pressures are zero by default, since turbulence and magnetic fields are not included by default. They must be independently specified. If either turbulent of magnetic pressure dominates then the density will be nearly constant.

P_{lines} is the nearly isotropic radiation pressure due to trapped emission lines (Ferland & Elitzur 1984, and Elitzur & Ferland 1986). Cloudy will stop if the internal line radiation pressure builds up to more than half of the total pressure since such clouds would be unstable unless they are self-gravitating (Elitzur and Ferland 1986). It is necessary to do at least a second iteration when radiation pressure is important since the total line optical depths must be known to compute line widths and level populations reliably. If more than one iteration is done then the radiation pressure will not be allowed to exceed the gas pressure on any except the last iteration. If the option **no abort** appears on the command line the code will never stop because of excessive radiation pressure. The radiation pressure is still computed, and the simulation may become unstable.

8.3. dlaw [options]

An arbitrary density law, specified by the user, will be used. There are two forms of this command. It is possible to either provide a new routine to calculate the density at an arbitrary depth or to interpolate on a table of points.

If the density or density law is specified with both this command and others, such as **hden, constant pressure**, etc, only the last command will be honored.

It is possible to specify a range in densities so extreme that the code will have problems. Cloudy works by linearizing all equations. It is possible to specify a change in the density that is so extreme that conditions change too much for linearization to be possible. The code uses adaptive logic to adjust the zoning and should prevent this from happening. The code will generate a warning if the density does change by too much – if this happens the cure is to not use very large density contrasts, at least if problems occur.

8.3.1. dlaw p1, p2, p3 ...

This is the default form of the command, and it passes the parameters on the command line to a user-provided function. There are up to ten parameters. A new function *fabden* must be written by the user and the version of *fabden* already in Cloudy must be deleted. (The code will stop if the initial version of *fabden* is not

replaced.)   Cloudy will call *fabden* as needed to determine the density as a function of depth. The arguments of the function are the radius and the depth (these are defined on page 8 above). Both are in centimeters and are double precision variables. The function must return the hydrogen density (cm^{-3}) as a double precision variable. The code provided in the function must use the ten or fewer parameters in the structure *dlaw* to compute the density at the current position

The following is an example of a function in which the density is the product of the first number on the command line and the depth.

```
/*fabden called by dlaw command, returns density for any density law */
#include "cddefines.h"
#include "dlaw.h"
#include "fabden.h"

double fabden(double radius, double depth)
{
        return( depth*dlaw.DensityLaw[0] );
}
```

8.3.2.   *dlaw table [depth, radius]*

If the keyword **table** appears on the **dlaw** command then the code will read in a set of ordered pairs of radii and densities. The original form of this option was added by Kevin Volk. There must be two numbers per line, as in the example below. The first number is the log of the radius or depth [cm] and is followed by the log of the hydrogen density [cm^{-3}]. If the keyword **depth** also appears on the command line then the first number is interpreted as the log of the depth from the illuminated face, and the table must begin with a depth smaller than 10^{-30} cm, the first point where the depth is evaluated. The first number is interpreted as the log of the radius if **depth** does not appear. The ordered pairs end with a line with the keyword **end** in columns 1 through 3. Up to 500 pairs may be entered.

Linear interpolation in log-log space is done. The following is an example.

```
dlaw table depth
continue -35 4
continue 12 4
continue 13 5
continue 14 6
continue 15 7
end of dlaw
```

Be sure that the first and last radii or depths extend beyond the computed geometry - this law is only be used for interpolation and the code will stop if extrapolation is necessary. Note that the first depth must be smaller than 10^{-30} cm, and also that there must not be a space in the first column of any lines with the numbers – the code will think that an end of file has been reached. Alphabetic characters can be placed anywhere on the line and will be ignored – I placed the word **continue** in the first four columns for this reason (it is actually totally ignored).

8.4. fluctuations density period . . . . .

This specifies a density that varies as a sine wave. It was introduced to investigate the effects of inhomogeneities upon the emission-line spectrum (see Mihalszki & Ferland 1983; Kingdon & Ferland 1995). The first number is the log of the period *P* of

the sine wave in centimeters. The second two numbers are the logs of the largest and smallest hydrogen densities over the sine wave. Order is important here. The last optional number is a phase shift φ (in radians), which allows the initial zone to occur at any part of the sine wave. If it is omitted the calculation will begin at the maximum value. If the phase is set to π the calculation will start at the minimum density.

The density is scaled according to the relation

$$n(r) = \left(\frac{n_{max} - n_{min}}{2}\right) \times \cos\left(\Delta r \frac{2\pi}{P} + \varphi\right) + \left(\frac{n_{max} + n_{min}}{2}\right) \quad [\text{cm}^{-3}] \tag{30}$$

where n_{max} and n_{min} are the maximum and minimum densities and Δr is the depth into the cloud, measured from the illuminated face.

The simulation may result in a large number of zones since the code must spatially resolve the density fluctuations. To do this, the zone thickness is not allowed to exceed ~0.05 of the period, so that each cycle is divided into at least 20 zones. This may result in very long execution times. The total number of zones (this sets the code's execution time) will be at least 20 times the number of cycles over the nebula.

The keyword **column** will replace the depth variable in equation 30 with the total hydrogen column density. The period should then be the column density of one cycle. This will more evenly weight the final column density over low- and high-density gas.

The **fluctuations abundances** command (page 59 above) proves a mechanism for varying the gas-phase abundances of the elements.

8.5. globule [density =2, depth =16, power =2]

This produces a density law that would be appropriate for a power-law density gradient irradiated from the outside (see, for example, Williams 1992). The total hydrogen density $n(r)$ is given by

$$n(r) = n_o \left(\frac{R_{scale\,depth}}{R_{scale\,depth} - \Delta r}\right)^{\alpha} = n_o \left(1 - \frac{\Delta r}{R_{scale\,depth}}\right)^{-\alpha} \quad [\text{cm}^{-3}] \tag{31}$$

where n_o is the background density outside the cloud, with default value 1 cm^{-3}, and Δr is the depth into the cloud, measured from the illuminated face. The log of n_o is the optional first number on the command line. The variable $R_{scale\,depth}$ is the scale depth for the cloud and has a default of one parsec, $R_{scale\,depth} = 3.086 \times 10^{18}$ cm. Other scale depths are specified by the optional second parameter, which must be entered as a log of the scale depth in cm. The optional third argument is the index α, which has the default[15] $\alpha = 1$. The arguments can be omitted from right to left.

---

[15] The default index was 2 for versions 89 and before.

8.6. hden 5.6, [proportional to R -2, ...]

The first number is the log of the total (ionic, atomic, and molecular) hydrogen density at the illuminated face of the cloud. This is the sum

$$n(\mathrm{H}) = n(\mathrm{H}^0) + n(\mathrm{H}^+) + 2n(\mathrm{H}_2) + \sum_{other} n(\mathrm{H}_{other}) \ [\mathrm{cm^{-3}}] \ . \tag{32}$$

If the optional keyword **linear** appears then the number is the density itself and not its log.

For situations where the hydrogen atom is close to LTE and the gas is hot, there is a problem in defining the neutral hydrogen density because of the well-known divergence of the partition function, as discussed, for instance, by Mihalas (1978). The atomic hydrogen density is defined as the total population in all computed levels. In most circumstances, i.e., $n(\mathrm{H}) \ll 10^{13}$ cm^{-3} and $T \leq 10^4$ K, the ambiguity is much less than 1%.

Several options are available to specify optional power-law dependencies on depth variables. These are described in the next sub-sections.

8.6.1. Power-law dependence on radius

The second (optional) number is the exponent α for a power-law radial density dependence, as in the following example:

```
hden 9, power =-2
```

i.e.,

$$n(r) = n_\mathrm{o}(r_\mathrm{o}) \left(\frac{r}{r_\mathrm{o}} \right)^{\alpha} \ [\mathrm{cm^{-3}}] \ . \tag{33}$$

In this example n_o, the density at the illuminated face of the cloud, will be 10^9 cm^{-3}. The optional power law is relative to radius, the distance to the central object, not the depth into the cloud. If $\alpha = -2$ then the density will be proportional to the inverse square of the radius. Both density and photon flux will fall off as an inverse square and the cloud will tend to have the same ionization parameter (and hence physical conditions) across the ionized zone.

8.6.2. Models extending to infinity

For an inverse-square law density dependence, there is a critical value of the number of ionizing photons emitted by the central object,

$$Q_{crit}(\mathrm{H}^0) = \alpha_B(T_e) n_\mathrm{o}^2 \, 4\pi \, r_\mathrm{o}^3 \ [\mathrm{s^{-1}}] \ . \tag{34}$$

A hydrogen ionization front will not be present and the model will extend to infinite radius when $Q(\mathrm{H}^0) \geq Q_{crit}(\mathrm{H}^0)$. In this expression $\alpha_B(T_e)$ is the hydrogen Case B recombination coefficient and n_o and r_o are the inner density and radius respectively. Generally, a hydrogen ionization front will not be present if the density falls off faster than an inverse square law, but rather the level of ionization will tend to *increase* with increasing radius. In either case, if a reasonable outer radius is not set, the calculation will extend to very large radii, an unphysically small density will result,

and usually the code will crash due to floating point underflow, followed by division by zero. It is usually necessary to set an outer radius when the density falls off with an index $\alpha \leq$ -2, since, for most circumstances, the cloud will remain hot and ionized to infinite radius and zero density.

8.6.3. Power-law dependence on depth

The density will depend on the depth into the cloud rather than the radius if both the optional exponent *and* the keyword **depth** appear:

```
hden 9, power =-2, scale depth = 13
```

The density is given by

$$n(r) = n_{\mathrm{o}}(r_{\mathrm{o}})\left[1 + \frac{\Delta r}{R_{scale}}\right]^{\alpha} \quad [\text{cm}^{-3}] \tag{35}$$

where R_{scale} is the scale depth and Δr is the depth. The log of R_{scale} [cm] is the third number on the line and must be set.

8.6.4. Power-law dependence on column density

The local hydrogen density will depend on the total hydrogen column density if both the optional exponent *and* the keyword **column** appear;

```
hden 9, power =-2, scale column density = 21
```

Here the density is given by

$$n(r) = n_{\mathrm{o}}(r_{\mathrm{o}})\left[1 + \frac{N(\mathrm{H})}{N(\mathrm{H})_{scale}}\right]^{\alpha} \quad [\text{cm}^{-3}] \tag{36}$$

where $N(\mathrm{H})$ is the total hydrogen column density [cm^{-2}] from the illuminated face to the point in question, and $N(\mathrm{H})_{scale}$ [cm^{-2}] is the scale column density. The log of $N(\mathrm{H})_{scale}$ is the third number.

9. GEOMETRY

9.1. Overview

This section describes commands that determine the geometry of the emission-line region. Many of the quantities used below are defined in the section on geometry starting on page 8 above.

The geometry is always spherical but can be made effectively plane parallel by making the radius much greater than the thickness of the nebula. It is also possible to compute a model in which the emission-line region is almost a disk. A covering or filling factor can be specified, and the cloud can be either static or expanding.

9.2. age 44 years [_off]

For a static geometry the code assumes that the cloud is old enough for atomic processes to have become time-steady. The **age** command allows the code to confirm that the micro-physics is indeed time steady. The age of the cloud is the number on the command line. The default units are linear seconds. The keyword **_log** will force the code to interpret the number as a log. The keywords **minutes**, **days**, **weeks**, **fortnights**, **months**, **years**, **centuries**, and **millennia** change the units.

The code keeps track of many equilibrium timescales during a calculation. After the calculation is complete it will check that none of the equilibrium timescales for significant physical processes were longer than the age of the cloud. The code will complain if the age of the cloud is not set, but still compute the model. If a physical process is not significant, for instance, the H_2 formation timescale in a coronal gas, the age is set to a negative number. This retains the value while not including the process as a significant part of the physical simulation.

If the keyword **_off** appears then the age will not be checked.

9.3. aperture [slit, beam]

The **aperture** command simulates observing a part of the computed structure with a spectrometer. It was first incorporated into the code by Peter van Hoof, who wrote the original version of this section.

One of the keywords **slit** or **beam** must appear. The keyword **beam** tells the code to simulate observing a resolved nebula with a small beam centered on the central star. If the keyword **slit** appears then the computed structure is observed with a slit that is longer than the nebula.

The **aperture** command only affects how volume emissivities are added together to form the final spectrum. It has no effect on any aspect of the calculation of conditions in the gas.

In the following a quantity α is defined with the following meaning: $\alpha = 0$ in the pencil beam case (we are observing along a single line of sight passing through the center of the nebula), $\alpha = 1$ in the long-slit case (we are observing through a narrow slit placed over the center of the nebula; the slit is longer than the nebula and the flux

is integrated over the entire slit), and $\alpha = 2$ in the general case (we are observing the flux integrated over the entire nebula). The default index is $\alpha = 2$.

In all cases an observed quantity Q_α can be defined for a line λ as

$$Q_\alpha(\lambda) = C_\alpha D_\alpha \int \left(\frac{r}{r_o}\right)^\alpha \varepsilon(\lambda) \; dr \; , \tag{37}$$

where $\varepsilon(\lambda)$ is the line's volume emissivity (erg cm^{-3} s^{-1}) and

$$\begin{aligned} C_0 &= 2 \\ C_1 &= 2\pi r_o \; , \\ C_2 &= 4\pi r_o^2 \end{aligned} \tag{38}$$

where r_o is the inner radius of the nebula. The covering factor D_α depends on the geometry and is

$$\begin{aligned} D_0 &= 1/2, \; 1 \\ D_1 &= \Theta/2\pi \\ D_2 &= \Omega/4\pi \end{aligned} \quad . \tag{39}$$

For the entire nebula ($\alpha = 2$) this is the familiar definition. In the long slit case ($\alpha = 1$) D_1 is the fraction of a large circle in the plane that is being observed that actually passes through nebular material. In the beam case ($\alpha = 0$) D_0 indicates whether only the front or back side of the line of sight is covered with nebular material ($D_0 = 1/2$) or if both sides are covered ($D_0 = 1$). The covering factor for the entire nebula is set with the **covering factor** command (see page 77 below) – it is not now possible to set the covering factor in the long slit case.

This command only affects the case where the code predicts line luminosities. When intensities are predicted all integrations are for a pencil beam through the slab. In the luminosity case Q_α will have units erg s^{-1} for the entire nebula, erg cm^{-1} s^{-1} for the long slit case, and erg cm^{-2} s^{-1} for the beam. In the intensity case the units are erg cm^{-2} s^{-1}.

In the luminosity case we can also construct relations between observed quantities and the quantities predicted by the code, as follows (neglecting interstellar extinction of course). The flux measured at the Earth will be

$$F_\alpha(\lambda) = \frac{A_\alpha}{4\pi D^2} Q_\alpha(\lambda) \; [\text{erg cm}^{-1} \text{ s}^{-1}], \tag{40}$$

with

$$\begin{aligned} A_0 &= \Omega_b D^2 \\ A_1 &= \Theta_s D \\ A_2 &= 1 \end{aligned} \quad . \tag{41}$$

Here F_α is the observed flux in erg cm^{-2} s^{-1}, D is the distance of the object (cm), as specified with the **distance** command (page 78 below), Θ_s is the slit length in radians, and Ω_b is the surface area of the pencil beam in sr.

In the intensity case (the intensity case implies $\alpha = 0$) the relation between the observed surface brightness $S(\lambda)$ and $Q_o(\lambda)$ is simply:

$$S(\lambda) = \frac{\Omega\left(1\,\text{arcsec}^2\right)}{4\pi} Q_0(\lambda) \approx 1.87043 \times 10^{-12} Q_0(\lambda) \ [\text{erg cm}^{-2}\,\text{s}^{-1}\,\text{arcsec}^{-2}] \ . \qquad (42)$$

For the case where both sides of a spherical shell are observed, $S(\lambda)$ will be twice this.

9.4. covering factor 0.3

This command sets a covering factor $\Omega/4\pi$ for the emission-line region. The argument is interpreted as the log of the covering factor if less than or equal to zero, and the covering factor itself if positive. It is impossible to specify a covering factor of zero.

The covering factor affects both the luminosity of the emitted spectrum and the radiative transfer of lines and continua. If a covering factor is set and the lines or continua are predicted as luminosities, then the luminosities will be for a shell covering Ω sr of the central object. Line luminosities will scale nearly linearly with the covering factor. The covering factor does not strongly affect the line intensities, (the emission per unit area) or the relative emission-line spectrum. It does have second-order effects on the spectrum through changes in the transport of the diffuse fields.

This covering factor is referred to as the geometric covering factor, and is stored as the variable *covgeo*. A second covering factor, *covrt*, affects the transfer of lines and continua. The number on this command line sets both covering factors.

If no covering factor is entered and **sphere** is not set then the default is for a geometric covering factor of unity (the shell fully covers the continuum source) but a radiative covering factor of zero (i.e., an open geometry).

9.5. cylinder log semi height =9.12

The model will be spherical, but truncated so as to simulate a cylinder (see Ferland et al. 1982). Figure 7 gives an example of the assumed geometry.

The inner and outer radii of the cylinder are set by the **radius** command described on page 79 below. The **cylinder** command sets the full height of the cylinder to twice the number entered on the command. The argument is the log of the semi-height in cm.

The effective volume element used to compute the emissivity of the gas is given by
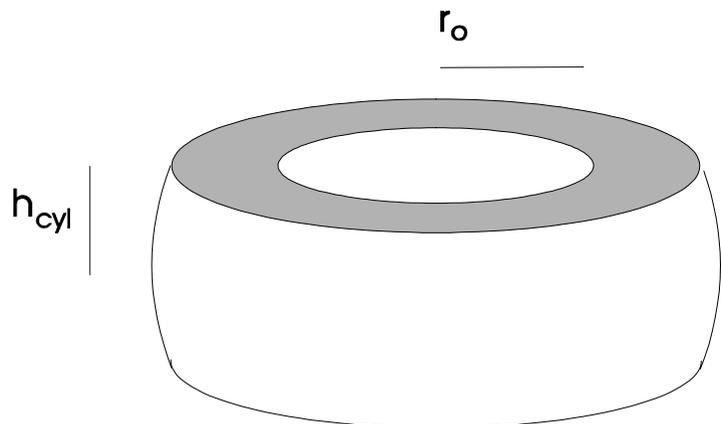
Figure 7 This figure shows the geometry assumed when the cylinder command is used. cylin

$$dV = 4\pi r_o^2 \left(\frac{r}{r_o}\right) \left(\frac{\min(r, h_{cyl})}{r_o}\right) f(r)\,dr \quad [\text{cm}^{-3}] \tag{43}$$

where r_o is the inner radius, h_{cyl} is the cylinder half-height, and $f(r)$ is the filling factor. The default value is $h_{cyl}= 10^{35}$ cm.

Changing the emissivity as described by equation 43 is the only effect of this command. It does not alter the radiative transfer methods, and is only formally correct when the continua and lines are optically thin.

9.6. distance 3.2 linear parsecs

This command sets the distance to the object from the Earth. The number is the log of the distance in centimeters. The **linear** keyword forces the number to be interpreted as the linear distance and the **parsecs** keyword changes the units to parsecs.

If the distance is set then it is possible to predict the emission-line fluxes observed at Earth. The code must have enough information to predict emission-line luminosities (see the discussion on page 21 above) to do this. If the code can predict luminosities and the distance is set, then the observed emission-line fluxes at the Earth will be printed if the **print line flux** command (126 below) is also entered.

This command can be combined with the **aperture** command (page 75 above) to simulate observing parts of a nebula from the Earth.

9.7. filling factor = 0.05 [index =-1]

The first number is the filling factor $f(r)$ for a clumpy model. It can be either the filling factor itself (which is greater than zero and less than or equal to one) or the log of the filling factor (if it is less than or equal to zero). The second optional number is the index α for a power-law variation of the filling factor $f(r)$, i.e.,

$$f(r) = f(r_o) \left(\frac{r}{r_o}\right)^{\alpha} \tag{44}$$

where r_o is the inner radius of the cloud.

The filling factor is used in two ways. The first is to modify the volume emissivity of the gas,

$$dE = \varepsilon f(r)\,dV \frac{\Omega}{4\pi} \quad [\text{erg s}^{-1}] \tag{45}$$

where $\Omega/4\pi$ is the covering factor. The second is to modify the optical depth scale

$$d\tau = \alpha_{l,u} \left(n_l - n_u \frac{g_l}{g_u}\right) f(r)\,dr \tag{46}$$

(see Osterbrock and Flather 1959).

A filling factor greater than unity is not allowed. Cloudy will set a filling factor of unity if a value greater than one is entered. The code will complain (but compute the model) if a filling factor is set in a constant pressure model since this makes no physical sense.

9.8. illuminate 45 deg [radians]

The plane-parallel slab is illuminated by a beam θ away from the normal. The default is $\theta = 0$ (normal illumination). The angle is in degrees unless the keyword **radian** appears.

The only effect of this command is to cause the beam of incident radiation to be attenuated by $\tau_n / \cos(\theta)$ where τ_n is the normal optical depth of the zone. Line and diffuse continua optical depths are not affected.

9.9. radius r(inner) [r(outer), thickness; parsec; linear]

The first number is the log of the inner radius. The second optional number sets a stopping radius and is either the log of the outer radius (if it is larger than the first number) or the log of the thickness of the cloud (if it is less than or equal to the first number).

The numbers are normally interpreted as the log of the radii in cm. If the optional keyword **linear** appears then the numbers are interpreted as linear numbers. The default units are centimeters, but the arguments will be interpreted as the log of the radii in parsecs if the keyword **parsec** appears. Arguments will be interpreted as linear parsecs if both keywords appear. The following gives examples of its use.

```
radius 19.5       ; log of inner radius in cm
radius 19.5 18.5 ; as above, but a thickness of 3x10^18 cm
radius 19.5 20    ; inner radius as above, outer radius 10^20 cm
radius 100 linear; inner radius of 100 cm
radius 0 parsecs ; log of radius in parses, so inner radius 1 pc
radius 1 to 3 linear parsecs ; inner radius 1 pc, outer 3 pc
```

The default outer radius is effectively infinite (actually 10^{31} cm). If the **radius** command is not entered and the surface flux or ionizing radiation field is set in some unambiguous manner (for instance, by setting the ionization parameter or energy density temperature), then a starting radius of 10^{30} cm will be set by default. Under most circumstances this will result in a plane-parallel geometry.

Page 166 below describes a problem that can occur if the second parameter is used with the **vary** option.

The **stop thickness** command (page 118 below) provides another way to set a stopping thickness without specifying a starting radius.

9.10. sphere [options][16, 17]

Cloudy normally assumes an open geometry (defined on page 9 above), i.e., that the gas covering factor is small. The **sphere** command should be included to change this assumption for an closed geometry (defined on page 9 above), one where the covering factor of the gas is large and the model spherical. This command tells Cloudy to take into account ionization by the diffuse continua and lines produced in the far side of the nebula (i.e., from beyond the central object), and not to attenuate the ionizing continuum by pure scattering opacities, such as electron scattering, back scattering by grains, or Rayleigh scattering.

This option should be set when the geometry is spherical and gas nearly fully covers the continuum source. It should not be set when the covering factor is small so that emission from a cloud is unlikely to encounter another cloud. This latter case is the default. In the language of Van Blerkom and Hummer (1967), **sphere** causes Cloudy to assume the symmetric case (their equation 2.14), rather than the default zero case (their equation 2.13) for diffuse continua. Here these are referred to as closed and open geometries, respectively.

Situations can occur where it is not obvious whether or not **sphere** should be used. In this case it would be best to compute models with and without **sphere** and compare results. In most cases this will only make a 10 – 15% difference in predicted quantities.

9.10.1. sphere expanding or static

Two optional keywords, **expanding** and **static,** determine how line transfer is handled. If **expanding** (the default when **sphere** is entered) is set then Cloudy assumes that line photons escaping from the illuminated face of the cloud are Doppler shifted away from lines of absorbing material on the far side of the shell. This will be the case if the expansion velocity exceeds the Doppler width by large amounts. If **static** is set then line photons *do* interact on both sides, so that even line photons produced at the illuminated face of the cloud will be strongly trapped by material on the far side. Lα radiation pressure in the H$^+$ region will probably be significant if **sphere static** is set.

It is necessary to iterate at least one time when the **static** option is used since the total line optical depths are not known on the first iteration. All optical depths are determined self-consistently on second and further iterations.

The specific effects of **sphere** are the following: The total continuous optical depths are assumed to be twice the computed optical depths, and the initial optical depth is half the total. All diffuse reemission (bremsstrahlung, free-bound, etc.) is

---

[16] The **slit** and **beam** options were recognized by the **sphere** command before version 96. These options were moved to the **aperture** command which was introduced in version 96.

[17] Before version 96 the **sphere** command included an option to change the covering factor, which could also be done with the **covering factor** command. The covering factor was removed from the **sphere** command, and now only the **covering factor** command changes the covering factor.

counted in the outward beam rather than only half. Scattering opacities are not considered in the attenuation of the incident radiation field. When **static** is set, the optical depth in Lα in the inner direction is set to 10^5 on the first iteration. Otherwise it is 10^{-20}. The total optical depths of lines are twice their computed depth for the static case. Finally, ionization by lines and continua produced in the other side of the nebula is included. At the end of the iteration all inward optical depths are set to half of the total value computed from the previous iteration. The diffuse continua are transferred using methods described in later sections.

9.11. stop depth …

9.12. stop thickness …

These commands provide methods to set the thickness of a cloud without specifying its radius. They are described on page 118 below.

9.13. wind u=300 km/sec [mass =1.4]

The envelope is a wind. The line widths and escape probabilities are modified in the appropriate manner, i.e., the effective line optical depth is given by

$$\tau_{l,u}(R) = \alpha_{l,u} \min(r, \Delta r) \left(n_l - n_u \frac{g_l}{g_u} \right) \left(\frac{u_{th}}{\max(u_{th}, u_{exp})} \right) \tag{47}$$

where u_{th} and u_{exp} are the thermal and expansion velocities respectively, and the radius used is the smaller of the depth or the radius. This is necessary to keep the effective column density from becoming larger than the total cloud column density when the radius is large and the expansion velocity is small.

9.13.1. *The sign of the velocity*

A *positive velocity* indicates the hypersonic wind solution that has been a part of the code since its beginnings. The equations of motion of the gas are solved. Acceleration due to line and continuous opacity of the gas and deceleration due to the gravity of the central object, are included. The calculation will stop if the gas comes to rest, or if any of the other stopping criteria is met. The initial velocity must be above the sonic point. Further details are presented in a section in Part 2.

The first parameter is the expansion velocity u_o at the illuminated face of the cloud. The approximations used are only correct if the model begins above the sonic point. The initial velocity must be greater than zero, and is entered in km s^{-1}. The density at the illuminated face of the cloud is entered with the **hden** command and the density is varied across the model to conserve mass flux (i.e., the product $\rho(r)r^2 u(r)$ is kept constant). Because of this, a filling factor would not make physical sense and should not be used (although it can be). The optional second parameter is the mass of the central star in solar units; its default value is one solar mass.

A *negative velocity* simulates the flow from a weak-D or D-critical H II region (Henney et al. 2005). This physics is currently under development, in collaboration with Will Henney and Robin Williams. The central object mass is set to zero.

9.13.2. *wind advection, velocity=-5 km/s*

The **advection** keyword turns on the effects of advection on the thermal and ionization equilibria. Currently advection is only treated in the case where the velocity is negative. The argument is the initial gas flow speed in km s^{-1}. This physics is being developed in collaboration with Will Henney and Robin Williams and is still being developed.

The advection case has the following options. In all cases the numerical parameter follows the keyword. There can be many parameters on one command line.

velocity – the number following this keyword is the velocity in km s^{-1}. The following would specify a simple *R*-type front

```
wind advection velocity -100
```

center, **index** – A varying mass flux with depth in the grid is specified by a law $\rho u = \Phi_o (r - c)^i$. This gives the variation in flux with distance *r* into the grid as a function of value *c* given by the **center** parameter and the value *i* given by the **index** parameter. For example,

```
wind advection velocity -20 center 10,000,000,000,000,000 index -2
```

would correspond to a wind diverging spherically from a point 10^{16} cm into the grid with a velocity of 20 km s^{-1} at the illuminated face. By default, the mass flux is constant ($i = 0$). If these are not specified then zero is assumed.

dfdr – This specifies the mass flux gradient, $\Phi' = d\,\rho u\,/\,dr$ [g cm^{-3} s^{-1}], as an alternative to specifying the velocity. This is useful in the case in which the flow reaches stagnation at the surface of the model, in which case the velocity itself does not fully constrain the physical parameters.

```
wind advection dfdr -20.90 index 2
```

The **set dynamics** command (page 186 below) controls many details of the dynamics. The **set dynamics pressure mode** command tells the code whether it should search for globally subsonic or supersonic solution. If this is not specified the code will compare the ram and gas pressure at the illuminated face to try to determine whether a super- or sub-sonic solution is appropriate.

9.13.3. *wind 5 km/s no continuum*

The **no continuum** keyword tells the code to not include continuous absorption in the calculation of the radiative acceleration.

10. OPTICAL DEPTHS AND RADIATIVE TRANSFER

10.1. Overview

Line transfer is relatively unimportant nebulae such as H II regions and planetary nebulae. Radiative transfer can be important in other environments, such as nova shells and the broad-line region of active nuclei, where excited states of hydrogen have significant populations and subordinate lines become optically thick. In other cases grains are present and all lines can be absorbed by background opacity. All radiative transfer effects are included in the treatment, including line thermalization, destruction by background opacities, pumping by the incident continuum, and escape from the cloud. Further details are given in Part 2 of this document.

It is necessary to iterate upon the solution if emission lines are optically thick since total optical depths are not known on the first iteration. The default is for a single pass through the cloud, and this is often adequate for low-density nebulae such as planetary nebulae or H II regions. A second iteration is sometimes enough to establish a fairly accurate line optical depth scale for most resonance transitions. Further iterations are usually needed when subordinate lines are also optically thick. The program has an **iterate to convergence** command (page 99 below) to iterate until the optical depth scale is well defined.

Line-radiation pressure cannot be computed accurately until the total line optical depths are known, so this quantity is only meaningful after the first iteration. Cloudy will stop if the internal radiation pressure exceeds half of the surface gas pressure in a constant-pressure model since such a geometry is unstable unless it is self-gravitating. On the initial iterations of a multi-iteration constant-pressure model, the radiation pressure is constrained to never exceed half the gas pressure. This is to prevent the calculation from stopping when the optical depth scale is not yet well converged.

The following subsections outline various commands that affect the radiative transfer and discuss several of the model atoms that are available.

10.2. atom overview

The set of **atoms** commands provide a mechanism to change details of the physical treatment of several model atoms that are incorporated in the case.

10.3. atom CO [options]

This is another name for the **atom rotor** command, described on page 95 below.

10.4. atom FeII [options]

This command adjusts details of the model Fe^+ ion. The default is to compute the lowest 16 levels, which produce IR emission, as in the real ion, but to do higher levels, which produce optical and UV emission, using the simplified and very fast scheme outlined by Wills, Netzer, and Wills (1985).

When any of the **atom FeII** commands are entered the number of levels is increased to the maximum value and the Wills et al. scheme is turned off. This complete model, described by Verner et al. (1999), is far more accurate but also *much* slower.

N.B. the keyword is **FeII**, not **Fe2**, to avoid scanning the number 2 off the command line. Both **FeII** and **Fe_II** (with a space) will be accepted.

10.4.1. atom FeII levels

This changes the number of levels used by the model atom. The upper limit is 371 levels. The default is to compute the lowest 16 levels and treat the remainder of the levels with the Wills et al. (1985) simple model. Decreasing the number of levels will speed up the execution time [roughly proportional to $n^2 \log(n)$] at the expense of a degraded simulation of the physics.

10.4.2. atom FeII trace

This turns on debugging printout for each call to the model atom.

10.4.3. atom FeII redistribution [options]

The keyword **redistribution** will change the form of the redistribution function for various lines within the model atom.

A keyword to specify which set of lines to change is required. This is either **resonance**, any line decaying to the ground term, or **subordinate**, transitions between excited states. The type of redistribution function to use must also be specified. The options are **_PRD** (partial redistribution), **_CRD** (complete redistribution with Doppler core only), and **CRDW** (complete redistribution with damping wings). (The underscore indicates a space.)

The keywords **redistribution show** tells the code to print the current default redistribution functions.

10.4.4. atom FeII simulate

This will cause results from the atom to be simulated. The large model atom is not actually called. This is very fast and is a debugging aid.

10.4.5. atom FeII slow

The keyword **slow** will cause the atom to always be reevaluated. Normally the code only reevaluates the atom when the local conditions have changed significantly.

10.4.6. atom FeII output options

Fe II emission is an exercise in uncontrolled complexity. Hundreds of thousands of lines contribute to what often appears as a pseudo-continuum. It is not practical to simply list the lines except in a few especially simple situations such as H II regions.

Three options are available for understanding output from the Fe II atom. Most often Fe II emission is seen as a blended continuum rather than individual lines. The general idea is to try to reduce the flood of information to a manageable level by distilling the emission into what an observer would actually see.

Fe II bands in the main output. A series of Fe II "bands" are automatically entered into the main emission-line output when the large Fe II atom is used. Each band represents the total Fe II emission integrated over all lines that lie within a range of wavelengths. This emission appears in the output with the label "Fe2b" and a wavelength close to the center of the band. The bands are chosen to represent features that an observer might be able to measure.

The list of bands is contained in the file *bands_Fe2.dat* which is included in the main Cloudy data distribution. This file is intended to be easily changed by the user. It explains the format of the information that is needed. There is no limit to the number of Fe II bands that can be specified.

punch FeII lines This option punches the intensities of all emission lines predicted by the model atom. The resulting file is very large and mainly useful for debugging the model atom, or understanding where within the atom a particular feature originates. **Punch** commands are described in more detail beginning on page 129 below.

punch FeII continuum This option punches the total Fe II emission as a continuous spectrum. Here a range of wavelengths is broken into a number of intervals and the total Fe II emission within each interval is added together. The result represents what would be observed by a spectrometer with a particular resolution. This command is described further on page 139 below. **Punch** commands are described in more detail beginning on page 129 below.

atom FeII continuum low=1000 high=7000 ncells=1000 The lower and upper bounds and the resolution of the Fe II continuum punched with the **punch FeII continuum** command can be adjusted with the **atom FeII continuum** command. The numbers are the lower and upper limits to the wavelength range in Angstroms and the number of wavelength cell that occur over this range.

10.5. atom H2 options

This controls options for the H_2 molecule. Its implementation is described in Shaw et al. (2005). The large molecule is not included by default. All of the following commands will tell the code to include the large model. The atom is represented by several thousand levels producing roughly half a million lines. This increases the execution time by a very large factor so is not included by default. The three-level model outlined by Tielens & Hollenbach (1985a) and expanded by Elwert et al. (2006) is used if this large model is not.

NB The number "2" appears in the keyword for this command. Any numerical options entered must appear after this two – the code will check that the first number parsed off the command line is the number 2.

Some details of the physical treatment of the H_2 molecule can be changed with the **set H2** command, described on page 188 below.

10.5.1. H₂ output options

Sufficiently strong H_2 lines will appear in the main emission line output. There are many other output options possible with the **punch H2** command (page 141 below). The section in Part 3 of this document describing calling the code as a subroutine describes several routines that will return predictions of the large molecule. The **set line precision** command (page 190 below) will add another significant figure to the wavelengths printed in the main printout. This may help isolate a particular line.

10.5.2. atom H2

With no options, the only effect is to enable the large model of the H_2 molecule.

10.5.3. atom H2 levels

This changes the number of electronic levels within the H_2 molecule. The default is seven and is also the larger number of levels, the ground and first six bound electronic states (all singlets). At least three must be done. This is sufficient to include the Lyman and Werner bands in the UV – this are a necessary minimum number of levels to include the correct photodissociation processes. If no number appears but the keyword **large** does then the code will use the upper limit.

10.5.4. atom H2 trace [options]

This turns on trace information concerning the H_2 molecule. The optional keywords **full**, **iterations**, and **final** will give full information, and overview of iterations during the convergence, and only final results. If the keyword **matrix** occurs the code will print the contents of the matrices that are used for solution of lower levels within the ground electronic state.

10.5.5. atom H2 limit -4

Calculating the level populations and line spectrum of the large H_2 molecule is computationally expensive. The code tries to save time by not computing the populations when the abundance of H_2 is negligible. This command changes the limit for the smallest H_2/H_{tot} ratio – the full model will be computed when the ratio is greater than this. The number is interpreted as the linear ratio if it is greater than zero, and the log of the ratio if it is less than or equal to zero. The keyword **_off** turns off the limit so that the large model of the molecule is always evaluated. The default limit is 10^{-8}, small enough for the large molecule to be computed across the entire Tielens & Hollenbach (1985a) model.

When the H_2 abundance is below this limit the photodissociation, heating, and cooling rates, are evaluated using expressions in Tielens & Hollenbach (1985a). H_2 level populations are set to their LTE value so that self-shielding in the electronic bands is still computed.

10.5.6. atom H2 matrix 43

Populations of the lower ro-vibration states of the ground electronic level are determined by two schemes. The first, and most straightforward, is the solution of a complete set of balance equations. The execution time increases as a power of the

number of levels, so there is some need to keep the number of levels treated this way small. The second is by back substitution, starting from the highest level with X and proceeding downwards. This command sets the total number of levels that are computed within the matrix. Levels higher than this will be treated with back substitution.

If the keyword **_off** (note the leading space) or **none** appears, or if the number of levels is less than 1, the matrix will not be used. It the keyword **_all** appears then all levels within X will be done this way.

10.5.7. atom H2 gbar [off; on]

The g-bar approximation is a rough relationship between the energy of a line and its collision rate coefficient. This can be used to guess a collisional deexcitation rate coefficient when no calculated data exist. This command turns this guess off or on. It is on by default.

10.5.8. atom H2 collisions [options]

This changes various collisional processes within the H_2 molecule. The commands have keywords **_ON** and **_OFF** to turn the process on or off. The default, when neither is present, is to disable the process.

atom H2 ortho para collisions on/off turns off ortho-para changing collisions with gas particles .

atom H2 collisional dissociation on/off turns on or off collisional dissociation. The default is to include it using the estimates given in Shaw et al. (2005). These rates are all only guesses and represent an uncertainty.

atom H2 grain collisions on/off turns off downward transitions induced by collisions with grains.

atom H2 He collisions old/new/off determines which set of He collisional rate coefficients are used. Old is the default and uses data from Le Bourlot et al. (1999). New uses rates now being computed in association with an Oak Ridge-based group led by Phillip Stancil.

Collisional deexcitation for the g-bar transitions are turned off with the **atom H2 gbar** command.

10.5.9. atom H2 chemistry [simple; full]

This changes how the interactions between the H_2 molecule and the rest of the chemical network are treated. By default, or if the keyword **full** appears, then the fully self-consistent formation and destruction rates are used when the large H_2 molecule is enabled. If the keyword **simple** occurs then expressions from Tielens & Hollenbach (1985a) are used instead.

10.5.10. atom H2 thermal [simple; full]

This changes how the heating and cooling by the H_2 molecule are treated. By default, or if the keyword **full** appears, then the fully self-consistent heating and

cooling rates are used when the large H_2 molecule is enabled. If the keyword **simple** occurs then expressions from Tielens & Hollenbach (1985a) are used instead.

10.5.11. *atom H2 noise [mean, standard deviation, seed]*

This multiplies the rates for collisional processes within the H_2 molecule by a Gaussian random number, so that $r' = r\,10^{\text{rand}}$ where r is the correct rate coefficient and *rand* is an Gaussian distributed random number. The first two optional numbers on the command line set the mean and standard deviation for the Gaussian random numbers. The first optional number is the mean, with a default of 0. The second optional number is the standard deviation with a default of 0.5. The last optional number is the seed for the random number generator, which must be an integer greater than 0. If the seed is not specified then the system time is used to generate a random seed.

10.5.12. *atom H2 output options*

Roughly half a million lines are predicted when the large H_2 molecule is included. The main emission-line printout includes all significant lines produced in the ground electronic state but does not include electronic transitions. The **print line H2 electronic** command (page 126 below) will include these lines. The family of **punch H2** commands (page 141 below) provides ways to save information such as column densities, the emission-line spectrum, and details of the effects of H_2 on the conditions in the cloud.

10.6. atom H-like [options]

This is used to change the treatment of atoms of the hydrogenic isoelectronic sequence, those with one electron, H^0, He^+, Li^{+2}, etc.[18] This implementation was initially created as part of Jason Ferguson's PhD thesis, and descriptions of the treatment are in Ferguson & Ferland (1997), Ferguson et al. (2001), and Bottorff et al. (2002). Any number of levels up to n of 400 can be computed. The *l, s* levels in hydrogenic species are nearly energy degenerate so only the $n = 2$ level is resolved into $2s$ and $2p$.

Most of these commands change all atoms and ions of the H-like iso-electronic sequence. A few allow the element to be specified.

10.6.1. *atom H-like collisions . . . .*

Collisional processes, both between levels ionization, are turned off with this command. They are all on by default. This command is mainly used for debugging. Separate collisional processes can be turned off with the following options. Only one option is recognized per command line so multiple commands are needed to turn off several processes. If no sub-options are recognized then all collisional processes are disabled. This command turns off collisions for all elements along the H-like isoelectronic sequence.

---

[18] This was the **hydrogenic** command in versions 90 and before of the code. The **hydrogenic** command still exists for backward compatibility.

atom H-like collisions l-mixing off[19] This turns off *l*-mixing 2*s*-2*p* collisions.

atom H-like collisional ionization off This command turns off collisional ionization by thermal electrons of all levels, except for the very highest level. Collisional ionization from the highest level is not turned off to allow the atom to have some coupling to the continuum. Ionization by cosmic rays is not affected by this command.

atom H-like collisional excitation off This command turns off collisional excitation of all levels, except for 2s-2p.

atom H-like collisions off All three collisional processes will be turned off if none of the three keywords are recognized.

Warning! The code will require a very number of zones if collisions are turned off in an optically thick cloud with a large ($n \gg 15$) hydrogen atom. Collisions will normally hold populations of very highly excited levels to values very near LTE. The FIR and radio lines will have very small line optical depths due to the correction for stimulated emission. When collisions are absent, the normal tendency of departure coefficients to increase with principal quantum number means that FIR and radio lines will strongly mase. The code dynamically adjusts the zoning to prevent these maser optical depths from diverging to minus infinity. A very large number of zones will be required to spatially resolve the masing region. This is a totally artificial, not physical, effect. The solution is to not turn off collisions with a large atom when performing a simulation with more than a trivial thickness.

10.6.2. atom H-like damping off

Rayleigh scattering, continuum scattering due to the extreme damping wings of Lyman lines, can be turned off with the **damping off** option. Rayleigh scattering is a significant opacity source in clouds that have large column densities of neutral material ($N(H^0) > 10^{23}$ cm^{-2}).

10.6.3. atom H-like levels 15 [element iron]

This sets the number of levels. The atom can extend up to any principle quantum number between the limits $4 < n \leq 400$. The size is limited mainly by the available memory and computer time. Note that there will actually be $n+1$ levels in the calculation since the 2*s* and 2*p* states are treated separately. The default highest quantum level is $n = 15$. Increasing the number of levels allows a better representation of the collision physics that occurs within higher levels of the atom at the expense of longer execution times and greater memory requirements.

If no number appears on the **atom H-like levels** command, but the keyword **large** or **small** does, then either 50 or 10 levels will be used. If the keyword **very small** also appears then the smallest possible atom is computed.

---

[19] This was the 2s2p option in versions 94 and before. The 2s2p option still exists for backward compatibility.

If the keyword **limit** appears then the largest possible number of levels, currently 400, will be computed. These keywords provide a version-independent method of insuring that the code uses the largest or smallest possible number of levels.

By default this command changes all elements along the hydrogenic isoelectronic sequence. If the keyword **element** appears together with the name of an element, only the model atom for that particular element will be changed. For example, the following would set the full isoelectronic sequence to a small number of levels, then reset hydrogen, helium, and iron to a large number.

```
atom H-like levels small
atom H-like levels large element hydrogen
atom H-like levels large element helium
atom H-like levels large element iron
```

The number of levels can only be set once at the very start of a calculation when space is allocated for the hydrogenic arrays. If the code is used to run a grid of models then only the first occurrence of **atom H-like levels** is honored and all following occurrences are ignored.

Warning! Note that the command

```
atom H-like levels limit
```

will set all 30 hydrogenic atoms to the limiting large number of levels. This would require roughly half a gigabyte of memory for the H-like sequence along, and would be very slow on today's computers. It is best to set only the most important elements to large levels.

10.6.4. atom H-like Lyman lines

This set of commands changes the treatment of the H I Lyman lines.

atom H-like Lyman pumping off turns off continuum radiative pumping of the Lyman lines of H I. This is meant to take into account the possible presence of Lyman lines in absorption in the continuum source.

atom H-like Lyman pumping scale 3 The continuum radiative pumping rate of the Lyman lines of H I will be multiplied by the scale factor that appears on the line. This is meant to take into account the possible presence of Lyman lines in emission rather than absorption..

atom H-like extra Lyman 1000 Atoms and ions of the H-like and He-like isoelectronic sequences use complete multi-level model atoms. The number of levels included is limited mainly by processor speed and available memory. Higher Lyman lines (used here to mean permitted lines that connect directly to ground) have little impact on emission since they scatter and are degraded into Balmer lines and Lα. However, an absorption spectrum will show them as a series of lines converging onto the photoionization continuum from the ground state. The code includes a large number of "extra" Lyman lines, included as absorbers with optical depths output with the **punch line optical depth** command (page 148 below), but not treated as part of the multi-level atoms. The default number of higher Lyman lines is 100, and this can be changed to any number with this command.

10.6.5. atom H-like matrix [lowt, populations]

This determines which form of the level population solver to use. The options are **lowt** for the simple solver that works for extremely low levels of ionization, and **populations** for the exact matrix solution for the populations themselves. The code normally decides this based on experience, but this command provides an option to override the default choices for testing.

10.6.6. atom H-like redistribution [options]

The keyword **redistribution** will change the form of the redistribution function for various lines within the model atom.

A keyword to specify which set of lines to change is required. This is one of **alpha** (the $2p - 1s$ transition), **resonance** (any higher Lyman line decaying to the ground term), or **subordinate** (all Balmer, Paschen, etc lines). The type of redistribution function to use must also be specified. The options are **_PRD** (partial redistribution), **_CRD** (complete redistribution with Doppler core only), and **CRDW** (complete redistribution with damping wings). (The underscore indicates a space.)

The keyword **show** tells the code to print the current default redistribution functions.

There is at present a fundamental uncertainty in the computation of the line radiation pressure for transitions such as Lα. For a simple two-level atom with incomplete redistribution, it has long been known that the line-width is proportional to $(a\tau)^{1/3}$ (Adams 1972, Harrington 1973; a is the damping constant). It is also easily shown that for complete redistribution and a frequency independent source function that the line width would be determined by inverting the Voigt function, and hence proportional to $(a\tau)^{1/2}$. Line interlocking, whereby scattered Balmer line radiation broadens the upper level of Lα (Hubbard and Puetter 1985), can alter the line width, as can collisional effects when the density is high enough for distant collisions to broaden the line. These effects cause major differences in radiation pressure and emergent flux (factors of several) for Lα, which can easily have an optical depth of $10^7 - 10^9$, when Balmer lines are also optically thick. This command determines which approximation is used. The default condition is incomplete redistribution, which minimizes the line width and radiation pressure. This issue is discussed further in Elitzur and Ferland (1986).

10.6.7. atom H-like TopOff 6 [_add scale]

This sets the lowest level for "topping off" the total radiative recombination coefficient. This is necessary to obtain the correct total radiative recombination rate coefficient with a finite number of levels. Because only a finite number of levels can be computed the sum of the total recombination coefficient will be less than the sum to infinity. This difference must be added somewhere to conserve the total recombination rate. One can choose to either add on the difference (if the keyword **_add** appears) or scale them (if **scale** appears). The default is **atom H-like topoff 11 scale**, which rescales the recombination coefficient from levels $n = 11$ through the highest level, so that the sum is correct.

This "top off" disturbs the model hydrogen atom since those levels with the extra recombination coefficient have unphysically large recombination rates. This often causes artificially strong masers. The command changes the lowest level with the extra recombination and was introduced for testing. The code will not predict any lines that include these disturbed levels.

10.7. atom He-like [options]

The following commands change treatments of some of the physical details of how the models of atoms along the He-like isoelectronic sequence (the species He^0, .. C^{+4}, N^{+5}, etc.). Details are given in Bauman et al. (2005), Porter et al. (2005), and Porter et al. (in preparation).

10.7.1. atom He-like Benjamin

This changes some aspects of the physical treatment of He I to be more like that used in Benjamin et al. (1999). Collisions to higher-n levels are turned off, and their tabulated recombination coefficients are used.

10.7.2. atom He-like collapsed levels x

This sets the number of collapsed levels, the high-n levels that are assumed to be well l mixed and hence can be treated without full consideration of L and S states. This is faster than solving for the resolved states, but the l-mixed approximation is only correct at higher densities. One is the minimum number of collapsed levels, as well as the default.

10.7.3. atom He-like collisions ….

Collisional processes between levels and collisional ionization are turned off with this command. Separate collisional processes can be turned off with the following options. Only one option is recognized per command line so multiple commands are needed to turn off several processes. If no sub-options are recognized then all collisional processes are disabled. This command turns off collisions for all elements along the He-like isoelectronic sequence.

atom He-like collisions l-mixing off This command turns off all collisions within the same n level for all elements in the helium-like isoelectronic sequence. Both L- and S-changing collisions are turned off.

atom He-like collisions l-mixing Pengelly [Vrinceanu] This changes to the Pengelly & Seaton (1964) formalism rather than currently the default, based on Vrinceanu & Flannery (2001) for $L > 2$.

atom He-like collisions l-mixing thermal [averaging, no averaging] The **no thermal averaging** option calculates the Vrinceanu & Flannery (2001) collision strengths at an impact energy equal to kT, rather than averaging over a Maxwellian distribution. This is the default when a small atom is used. Thermally averaged collision strengths are used when the atom is large.

atom He-like collisions excitation This command turns off collisional excitation, for all $n_u \neq n_l$ transitions, for all elements in the helium-like isoelectronic sequence.

10.7.4. atom He-like dielectronic recombination [off]

This option turns dielectronic recombination off for the entire isoelectronic sequence. Dielectronic recombination is included by default. This command has no effect without the **_off** option.

10.7.5. atom He-like error generation X

Monte Carlo error analyses can be performed with this command. Randomly generated Gaussian errors are applied to atomic data. The parameter X is an integer used to set the seed of the random number generator. See Porter et al. (in preparation) for details.

10.7.6. atom He-like FSM

For $L > 2$, levels with the same n and L but different S are strongly mixed due to the spin-other-orbit interaction. This option modifies transition probabilities to account for this fine-structure mixing. The effect on multiplet emissions is negligible. See Bauman et al (2005) for a discussion and calculation of this effect in a zero-density J-resolved helium atom.

10.7.7. atom He-like gbar options

The code employs various forms of the $\bar{g}$ approximation to fill in collision strengths for those transitions with no quantal calculations. This command changes which approximation is used. The options are **Vriens** for the Vriens and Smeets (1980), **_new** for our new approximation, and **_off** to set this to zero. The "new" approximation has two forms, which are specified as a number on the line, either 1 or 2. The default is 1.

10.7.8. atom He-like levels 4 [element iron]

This sets the number of n levels. The argument is the principal quantum number n for the highest level. It must be three or greater.

If no number appears on the command, but the keyword **large** or **small** does, then the number of levels will be large enough to extend to either $n = 40$ or 3. These keywords provide a version-independent method of insuring that the code uses a large or small number of levels. Actually, the atoms are coded so that there is no limit to the number of levels that can be included, other than the memory and computer time requirements.

Another keyword, **huge**, is used solely for collisionless Case B calculations, such as those in Bauman et al (2005). It causes Cloudy to bypass some system memory allocation for a faster, less memory-intensive, calculation of radiative-cascade spectra. The code cannot be used for other purposes since it disables everything not directly related to the radiative cascade of helium. Disabling this physics is the only direct effect of this command. It is still necessary to set the desired number of levels.

The model atom resolves n-levels into l-levels, and the $2\,^3P$ term is split into three J levels. The atom fully resolves the l-levels, and for a given n there will be a total of $n_l = n^2 + n + 1$ l levels. The default number of n levels for He^0 is 7, resulting in 57 l levels, $n = 5$ for 31 l levels for C, O, and Fe, and $n = 3$ for 13 l levels for all remaining

elements. For reference, to include all l -states within the n = 2, 3, 4, 5, and 6 levels, the atom will need 7, 13, 21, 31, and 43 l -states. Increasing the number of levels allows a better representation of the atom's emission, especially the collision physics that occurs within higher levels of the atom, but at the expense of longer execution times and greater memory requirements.

By default this command changes the number of levels for all elements along the He-like isoelectronic sequence. If the keyword **element** appears together with the name of an element, only the model atom for that particular element will be changed. For example, the following would set the full isoelectronic sequence to a small number of levels, then reset helium and iron to a large number.

```
atom He-like levels small
atom He-like levels large element helium
atom He-like levels large element iron
```

The number of levels can only be set once at the very start of a grid of calculations. This is because space is allocated for the needed arrays only one time per core load. If the code is used to run a series of models then only the first occurrence of **atom He-like levels** will be honored and all following occurrences will be ignored.

The model atom does not give a good representation of lines that come from the highest n level, which is always a "collapsed" level. Only lines coming from the first n - 1 levels will be printed at the end of the calculation.

Warning! Note that the command

```
atom He-like levels large
```

will set all 29 He-like atoms to a large number of levels. This would require roughly half a gigabyte of memory and would be very slow on today's computers. It is best to set only the most important elements to large levels. Depending on the application, this may be only He itself (when $\approx 10^4$ K gas is considered) or the more abundant second and third row elements (for hot gas and X-ray applications).

There are also collapsed levels – levels that do not resolve S or L but assume that the states within n are well-mixed. These collapsed levels bring together all of the individual nLS terms as one pseudo-level. The recombination coefficient into this pseudo-level is the sum of recombination coefficients into the individual terms plus the recombination topoff. Transition probabilities from this pseudo-level are calculated as follows

$$A(n_u \to n_l L_l S) = \frac{\sum_{L_u = L_l \pm 1} g_{L_u S} A(n_u L_u S \to n_l L_l S)}{\sum_{L_u = L_l \pm 1} g_{L_u S}}. \tag{48}$$

This causes the collapsed level to behave exactly as if it were a set of resolved terms populated according to statistical weight.

10.7.9. atom He-like Lyman 1000

Atoms and ions of the H-like and He-like isoelectronic sequences use complete multi-level model atoms. The number of levels included is limited mainly by

processor speed and available memory. Higher Lyman lines (used here to mean permitted lines that connect directly to ground) have little impact on the emission, since they scatter and are degraded in Balmer lines and Lα. However, an absorption spectrum will show them as a series of lines converging onto the continuum from the ground state. The code includes a large number of "extra" Lyman lines, included as absorbers with optical depths output with the **punch line optical depth** command (page 148 below), but not treated as part of the multi-level atoms. The default number of higher Lyman lines is 100, and this can be changed to any number with this command.

10.7.10. atom He-like matrix [lowt, populations]

This determines which form of the level population solver to use. The options are **lowt** for the simple solver that works for extremely low levels of ionization, and **populations** for the matrix solutions for the populations. The code normally decides this based on experience, but this command provides an option to override the default choices for testing.

10.7.11. atom He-like no recombination interpolation

The code normally derives recombination coefficients by interpolating on a table that lives in the data directory. This command tells the code to compute recombination coefficients on the fly by integrating over the stored photoionization cross section.

10.7.12. atom He-like redistribution [options]

The keyword **redistribution** will change the form of the redistribution function for various lines within the model atom.

A keyword to specify which set of lines to change is required. This is one of **alpha** (the $2\,^1P - 1\,^1S$ transition), **resonance** (any higher Lyman line decaying to the ground term), or **subordinate** (all Balmer, Paschen, etc lines). The type of redistribution function to use must also be specified. The options are **_PRD** (partial redistribution), **_CRD** (complete redistribution with Doppler core only), and **CRDW** (complete redistribution with damping wings). (The underscore indicates a space.)

The keyword **show** tells the code to print the current default redistribution functions.

10.8. atom rotor [options]

Rotational transitions within the ground electronic – vibration state of some molecules are treated as a rigid rotor. This is a single implementation that currently includes ^{12}CO and ^{13}CO. (The ^{13}CO chemistry is not now done – the $^{12}C/^{13}C$ ratio is set with the **set 12C13C** command described on page 184 below).

10.8.1. atom rotor levels 25

This sets the number of rotation lines (not levels) in the rigid rotor molecules. There will actually be one more level than line, to include the upper level of the

highest rotation transition. There is no limit to the number of levels that can be included. The default number of rotation levels is 20.

A problem can occur if too few levels are used. The lower rotation transitions of CO are often optically thick because of its great abundance. The cooling will shift to higher levels as a result. If the molecule is too small the level populations will pile up into the highest levels, the cooling will be underestimated, and the transition to the highest level may artificially mase. The code will check for this condition and generate a caution if it occurs. If this happens the model should be recomputed with more levels.

The number of levels can only be set once at the very start of a calculation when space is allocated for the rigid rotor line arrays. If the code is used to run a grid of models only the first occurrence of **atom rotor levels** will be honored and all following occurrences ignored.

10.9. Case B [tau ly alpha = 9; options]

This command is used to simulate deep regions of a significantly optically thick cloud, or to check the line emission from the hydrogen and helium atoms in the Case B limit. This command *should not* be used in any model that is supposed to represent a real physical environment, and is intended only to provide an easy way to check predictions of the code against simple, more limited, calculations. In particular, when this is used the ionization of the gas the local intensity of $L\alpha$ and Lyman line optical depths will be artificially large. This may affect the ionization and temperature of the gas.

With no options this command sets the inner optical depth of $L\alpha$ for all atoms and ions of the H- and He-like iso-electronic sequences to 10^5 so that even a one-zone model will be close to Case B[20]. The optional number is the log of the $L\alpha$ optical depth. One-sided escape probabilities are used so the total escape probability is simply that for the inward direction. In keeping with the Case B approximation the **Case B** command suppresses excited states line optical depths.

The atoms include all collisional processes. Case B does not define the population of the ground or first excited state so a true comparison with Case B results should have collisions from these levels turned off. This is done with the Hummer and Storey option (with the key **humm**), to allow comparison with their 1987 and 1995 papers. Collisions from the ground and first excited states *are* included if this second option is not specified. Collisions between levels with $n \geq 3$ *are* included unless the **atom H-like collision off** or **atom He-like collision off** commands are given. Collisions between the $2s$ and $2p$ states are always included unless the **atom H-like collisions 2s2p off** command is given.

---

[20] Before version 96 the default optical depth was 10^9. This caused extreme $L\alpha$ behavior in a grain-free H II region. The lower value is a better estimate of the physics that occurs in an actual H II region.

In the case of the He-like isoelectronic sequence the **Case B** command sets the optical depths in the singlet Lyman lines to a large value. The Hummer & Storey option (with the key **humm**) turns off collisions from $1\,^1S$, $2\,^1S$, and $2\,^1S$.

The **no Pdest** option turns off destruction of Lyman lines by background opacity.

There are several side-effects of this command that may after the spectrum or physical conditions in unexpected ways. The large $L\alpha$ optical depth will often result in an especially strong radiation field within this line. This affects the gas through photoionization of excited metastable states of H and He and of those elements with a small enough ionization potential. The **no photoionization** option on the **Case B** command tells the code not to include photoionization from these excited metastable states. But these strong diffuse fields will also strongly affect the level of ionization of the gas, making the resulting ionization equilibrium a fiction. Optically thin gas is actually described by Case C (Ferland 1999), where continuum pumping enhances Balmer lines. The large Lyman line optical depths that result from the **Case B** command will prevent continuum resonant pumping of the atom. Beware.

This command is used in the test suite to perform classical PDR simulations which ignore the H II region and assume that the PDR is illuminated by a radiation field with no hydrogen-ionizing radiation. Classical calculations do not include a full treatment of H or He so their ions are only produced by cosmic rays. Cloudy does a full treatment and will find a thin layer of highly ionized gas. This is produced by Lyman line pumping into excited states, especially metastable $2s$, which are then photoionized by relatively low-energy light. The process stops when the Lyman line optical depths become large enough for the atom to become self-shielding. The **Case B** command will stop this process from ever becoming important by starting with large Lyman line optical depths.

This command is included in the PDR simulations in the test suite to stop Lyman line pumping from occurring. A better solution would be to include the PDR as a additional layer outside the H II region. The Lyman line optical depths across the ionized gas will be large and prevent continuum pumping from occurring in the PDR. Combining the H II region and PDR also results in a self-consistent calculation (Abel et al. 2005).

10.10. diffuse fields [outward, ots]

This specifies which method is to be used to transfer the diffuse fields, the emission from gas within the computed structure. The options are **outward only** and **_ots**.

The **_ots** option does take into account optical depths in both the inward and outward directions. The **_ots** option has a **SIMPLE** option, which will do a very simple OTS approximation without taking optical depths into account. All diffuse fields with energies capable of ionizing hydrogen are assumed to do so, and those with smaller energies freely escape.

If **outward** is chosen then the code will check for a number. This determines which of the many forms of the outward-only approximation (Tarter 1967) is used. The default[21] is 2. This is intended for testing the code.

This choice does not strongly affect the predicted emission-line spectrum, but it does change the temperature at the illuminated face of the cloud. These approximations are described in Part 2 of this document.

10.11. double optical depths

On second and later iterations the code uses the total optical depths of the computed structure as part of the solution. This command doubles the total optical depth so that the shielded face of the cloud becomes the mid-plane of a larger structure.

This original purpose of this command was to simulate a geometry in which ionizing radiation strikes the plane-parallel cloud from both sides, such as a Lα forest cloud. The total line and continuum optical depths are set to twice the computed optical depth at the end of the iteration. The computed model is then one half of the cloud, and the other half of the cloud is assumed to be a mirror image of the first half. Doubling the total line and continuum optical depths at the end of the iteration is the *only* effect of this command. Physical quantities such as the physical thickness, column densities, or line emission *are not* affected.

This approximation makes sense if the cloud is optically thick in lines but optically thin (or nearly so) in the continua. Lines such as the Lα transitions of He I and He II can be important sources of ionizing radiation. Their transport will be handled correctly in this limit when this command is used. Continuum transport out of the cloud will also be treated correctly, but attenuation of the incident continuum will *not* be if the cloud is optically thick in the continuum.

The second use of this command is the situation when the outer edge of a computed structure is not the other edge of the cloud. A typical PDR calculation is an example. The calculation starts at the illuminated face and continues until the gas becomes cool and molecular. The stopping point often does not correspond to the outer boundary of the molecular cloud, but rather is a point that is "deep enough" for a given study. The optical depths are always computed self-consistently. On second and later iterations the total optical depths are those of the computed structure. Near the shielded face the outward optical depths will be small and radiation will freely escape in the outward direction. The gas temperature may fall dramatically due to the enhanced cooling as a result of the free escape of line photons. Actually considerable neutral or molecular material may extend beyond the stopping point so that line photons do not freely escape. The shielding effects of this unmodeled extra material can be included with the **double optical depths** command. Then, the shielded face of the cloud will correspond to the mid-plane of

---

[21] OTS was the default in version 86 and before.

the overall structure, and lines will not artificially radiate freely into the outer hemisphere.

10.12. helium [options]

This command was replaced with the **atom He-like** command, described on page 92 above, in version 96.

10.13. hydrogenic [options]

This command was replaced with the **atom H-like** command, described on page 88 above, in version 96.

10.14. iterate [2 times]

This specifies the number of iterations to be performed. The default is a single iteration, a single pass through the model. At least a second iteration should be performed in order to establish the correct total optical depth scale when line transfer or radiation pressure is important. Two iterations are sometimes sufficient, and will be done if no numbers are entered on the command line. No more than 200 iterations can be performed because of the present limits to the sizes of several vectors used to store information. A comment will be printed after the last iteration if the total optical depth scale has not converged and further iterations are needed.

10.14.1. Number of iterations

There is a slight inconsistency in how the code counts the number of iterations. The way it functions in practice is what makes the most sense to me.

The word *iterate* is from the Latin for "again". So the true number of "agains" should be one less than the total number of calculations of the cloud structure. When the **iterate** command is not entered there is one calculation of the structure and so formally no iterations. If any one of the following commands is entered:

```
iterate
iterate 0
iterate 1
iterate 2
```

then exactly two calculations of the structure will be done. If the number on the line is two or greater, then the number will be the total number of calculations of the structure.

10.14.2. iterate to convergence [max =7, error =.05]

This is a special form of the **iterate** command in which the code will continue to iterate until the optical depths have converged, or a limit to the number of iterations has been reached. The optional first number on the line is the maximum number of iterations to perform with a default of 10. It is not now possible to specify more than 200 iterations. The second optional number is the convergence criterion. The default is for optical depths to have changed by less than a relative fraction of 0.20 between the last two iterations. The optional numbers may be omitted from right to left. If the transitions are optically thin then only a second iteration is performed.

10.14.3. Convergence problems

The code generally will not converge if it has not done so within ten or so iterations. The most common reason for convergence problems is that the specified column density or thickness causes the simulation to end very near a prominent ionization front. In this case very small changes in the physical conditions result in large changes in the optical depths. This is a physical, not numerical, problem. The code will not have convergence problems if an optical depth is used as a stopping criterion instead.

10.15. no scattering opacity

This turns off several pure scattering opacities. These include scattering by grains, electron scattering, and the extreme damping wings of Lyman lines (Rayleigh scattering). When scattering opacity is included and an open geometry is computed, the scattering opacity is assumed to attenuate the incident radiation field as $(1+0.5\tau_{scat})^{-1}$ rather than $\exp(-\tau)$ (Schuster 1905).

Scattering should be neglected in a spherical geometry with gas fully covering the source of ionizing radiation. Scattered photons are not really lost, but continue to diffuse out with (perhaps) a slight shift in energy. Electron scattering is generally the most important scattering opacity in a grain-free mixture. If $\tau_{scat} \leq 1$ then it is reasonable to consider electron scattering as a heating and cooling process, but not as an absorption mechanism, if the energy shifts are not large (i.e., $h\nu \ll mc^2$) and the geometry spherical (this is not correct for γ-ray energies, of course). Cloudy is not now designed to work in environments that are quite Compton thick, but should work well for clouds where the electron scattering optical depths are less than or of order unity. When this command is entered Compton energy exchange and recoil ionization are still included as heating, cooling, and ionization processes, but not as opacity sources. (Thermal and ionization effects of Compton scattering are turned off with the **no Compton** command.) The **no scattering opacity** command is automatically generated when **sphere** is specified.

10.16. turbulence = 100 km/sec [_log, dissipate]

This enters a microturbulent velocity expressed in kilometers per second. If the optional keyword **_log** appears (note the leading space) then the number is interpreted as the log of the turbulence.

10.16.1. Definitions

The line-of-sight projected Doppler width is given by the quadratic sums of the thermal and turbulent parts,

$$u = \sqrt{u_{th}^2 + u_{turb}^2} = b = \sqrt{2}\sigma \ [\text{cm s}^{-1}] \tag{49}$$

where b is the Doppler parameter used in much of the absorption-line literature and σ is the standard deviation for a normal distribution. The thermal motion of a particle with mass m is given by

$$u_{th} = \sqrt{2kT/m} \ [\text{cm s}^{-1}] \tag{50}$$

and the turbulent line width u_{turb} is normally zero. This command sets u_{turb}. Note that with these definitions the full-width half-maximum (FWHM) of a line is equal to

$$u(FWHM) = u\sqrt{4\ln 2} \ \ [\text{cm s}^{-1}] \,. \tag{51}$$

Fluorescent excitation of lines becomes increasingly important for larger turbulent line widths. Line pumping is included as a general excitation mechanism for all lines using the formalism outlined by Ferland (1992) and described further in a section of Part 2. The line center optical depth varies inversely with u, so the effects of line optical depths and trapping become smaller with increasing u. Larger u inhibits self-shielding so that pumping is more important.

10.16.2. Energy dissipation

The **dissipate** option on the **turbulence** command provides a way in include conversion of wave energy into heat (see Bottorff & Ferland 2002). When the option is used a second number must appear, the log of the scale length for the dissipation in cm. Then the turbulent velocity will have the form

$$u_{turb}(r) = u_{turb}(r_o)\exp(-\Delta r / r_{scale}) \quad [\text{cm s}^{-1}] \tag{52}$$

where $u_{turb}(r_o)$ is the turbulence at the illuminated face and Δr is the depth into the cloud. The wave mechanical energy is assumed to have been converted into heat, and there is a local heating rate given by (Bottorff & Ferland 2002)

$$G(r) = 3.45\times10^{-28}2^{-3/2}u_{turb}(r)^3 \quad [\text{erg cm}^{-3}\text{ s}^{-1}] \tag{53}$$

10.16.3. Turbulent pressure

Turbulence adds a pressure component given by

$$P_{turb} = \frac{1}{2}\rho u_{turb}^2 = 5.8\times10^6 \left(\frac{n_{tot}}{10^5 \text{ cm}^{-3}}\right)\left(\frac{u_{turb}}{1 \text{ km s}^{-1}}\right)^2 \quad [\text{erg cm}^{-3}; \text{ dynes cm}^{-2}] \tag{54}$$

where n_{tot} is the total density, u_{turb} is the turbulent velocity, and solar abundances are assumed. This term is included in the equation of state when the total pressure is kept constant (see page 69 above).

The code will complain but still compute the model if a turbulent velocity is specified in a constant pressure model.

10.16.4. Equipartition turbulent – magnetic pressures

The **equipartition** option on the **turbulence** command sets the turbulent velocity to an equipartition between magnetic and turbulent energy densities. That is,

$$\frac{1}{2}\rho u_{turb}^2 = \frac{B^2}{8\pi} \ [\text{erg cm}^{-3}]. \tag{55}$$

The **magnetic field** command (page 108 below) sets B. The magnetic field and equipartition turbulence are included in the gas pressure law. The turbulent pressure is included by doubling the magnetic pressure. Heiles & Crutcher (2005)

argue that the correlation between turbulent and magnetic pressures, while true on average, does not hold in detail for specific regions of the ISM.

11. THERMAL SOLUTIONS

11.1. Overview

This section describes options that affect the thermal solution and the gas kinetic temperature. These are mainly details concerning processes that affect the temperature or additional sources of heating or cooling.

The accuracy of the thermal solution is set by the error tolerance in the heating – cooling balance. This is set with the **set temperature tolerance** command described on page 194 below. It also can be used to change some details of the thermal solution. The error tolerances in other converged quantities are described on page 183 below.

11.2. cextra -14.231 [temp to the 1.5 power]

This adds an extra source of cooling due to some unspecified physical process. The first number is the log of the cooling rate [erg cm^{-3} s^{-1}]. The second number is an optional exponent to specify a temperature dependence. The cooling will be given by

$$\Lambda = 10^{c_1} \times \left(\frac{T_e}{10^4 \text{ K}} \right)^{c_2} [\text{erg cm}^{-3} \text{ s}^{-1}] \tag{56}$$

where c_1 and c_2 are the two numbers entered with this command. If the second optional argument c_2 is not specified then zero (i.e., constant cooling) is assumed.

The function evaluating cextra is actually coded in routine *coolr*, where the rate is given by the variable *cextxx*. The expression can be easily changed to other forms by editing this routine.

11.3. constant temperature, t=10,000K [linear]

A constant electron temperature calculation will be performed. The number is either the electron temperature itself, or the log of the temperature if ≤ 10. If the optional keyword **linear** appears then the number is always interpreted as the temperature itself, and not its log.

Collisional ionization of all atoms and ions is always included, so this option can be used to produce clouds in coronal or collisional equilibrium.

WARNING! It is also necessary to specify a stopping criterion of some kind when this command is used. Most thermal-equilibrium calculations stop when the electron temperature falls below some lowest value, set with the **stop temperature** command (page 118 below) and with the default value 4000 K. This cannot happen with a constant-temperature model. For instance, a constant temperature model of a planetary nebula will continue until the default limit to the number of zones (now 1400) is reached. The *vast* majority of the cloud will consist of predominantly neutral gas well outside the hydrogen Strömgren sphere. This gas will have a small ambient level of ionization and emission due to collisional ionization. The resulting emission-

line spectrum might be surprising since the neutral gas contributes significant emission.

A more physical model could be done by using the **stop eden** (page 114 below) or **stop efrac** (page 115 below) commands to stop the calculation when the hydrogen ionization front is reached, or **stop zone** (page 119 below) to stop the calculation at a particular zone number.

11.4. constant grain temperature 20K [linear]

Normally the temperature of each grain constituent is determined by balancing heating and cooling. This command sets the grain temperature to the indicated quantity.

If the **linear** keyword appears then the number is always interpreted as the linear temperature. Otherwise numbers ≤ 10 are interpreted as a log of the temperature.

Other aspects of the grain physics are controlled with the **grain** command, described on page 60 above.

11.5. coronal equilibrium, T=10,000,000K [linear]

Coronal equilibrium, in which the gas is mainly collisionally ionized, is assumed. The number is either the temperature or the log of the temperature if the argument is ≤ 10. The optional keyword **linear** will force the interpretation as the linear temperature. This holds the electron temperature at the specified value.

It is necessary[22] to also specify some sort of stopping criteria. The calculation will probably continue until the default limit to the number of zones is reached if another stopping criterion is not specified.

Figure 8 shows the soft X-ray line and continuum emission predicted from the input stream in the test case **ism_hot_brems.in**.

11.6. cosmic rays [background, density, rate]

This adds cosmic ray and their heating and ionization. This physics is described by Ferland & Mushotzky (1984) and in the subsection *Cosmic Ray Interactions* in the section *Other Physical Processes* in Part 2 of this document. Cosmic rays mainly heat ionized gas and both heat and create secondary ionizations in neutral gas. All of this is done self-consistently. Collective effects are not included in the heating and ionization rates, but they may not be important either (Rephaeli 1987).

The **set csupra** command (page 186 below) provides a way to specify a secondary ionization rate. The rate introduce by the **set csupra** command would not be self-consistent with the rest of the calculation but does provide a way to test the code in certain simple limits.

---

[22] In versions 87 and before, the **coronal** command set the zone thickness to 1 cm, and stopped after computing one zone.
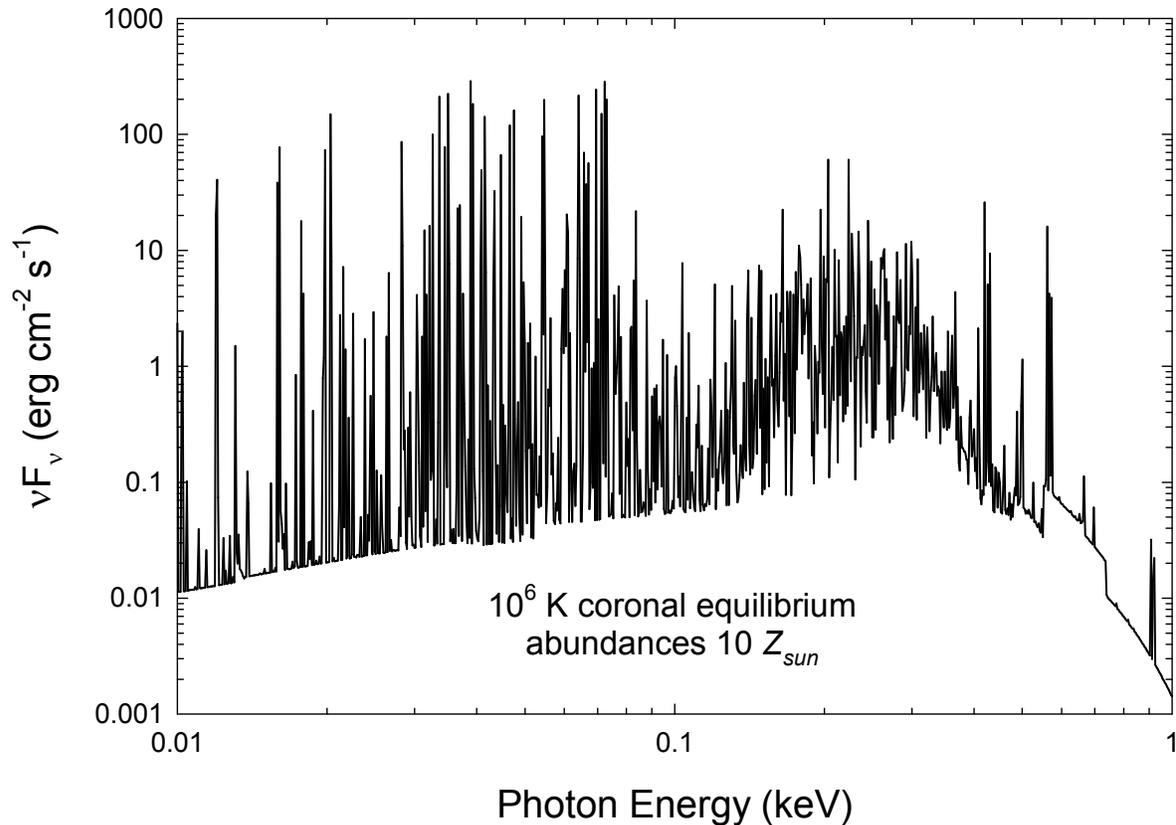
Figure 8 This shows the soft X-ray emission from a simple model of the hot phase of a metal-rich ISM. The input script ism_hot_brems.in was used to create the predictions.

Cosmic rays must be included if the calculation extends into a PDR or molecular cloud. The ion-molecule chemistry that occurs in the cold ISM requires a source of ionization (Dyson & Williams 1997). In fact, most estimates of the amount of galactic background cosmic rays are based on the abundances of chemical ions such as H_3^+ (McCall et al. 2003). The chemistry network will probably collapse if the gas becomes molecular but cosmic rays are not present. The code will complain but try to compute the model if a simulation without cosmic rays extends into cold gas. If cosmic rays are not included in the calculation, but the neutral hydrogen ionization rate falls below 10^{-17} s^{-1}, the code will print a comment stating that the ionization rate fell below the galactic background rate.

11.6.1. cosmic rays [background 1.2]

If no keywords or numbers are on the line then this includes galactic background cosmic rays. They produce an H^0 ionization rate of 2.5×10^{-17} s^{-1} and an H_2 ionization rate of 5.0×10^{-17} s^{-1} in predominantly neutral gas[23] (Williams et al. 1998).

An optional scale factor specifies the cosmic ray ionization rate relative to this background value. The scale factor is assumed to be a log unless the keyword **linear** also appears.

---

[23] Before 2004 the code used a background ionization rate of 7.4×10^{-18} s^{-1} quoted by Tielens and Hollenbach (1985a; Table 10) and McKee (1999).

For reference, the study of the starburst in M82 by Suchkov, Allen, & Heckman (1993) find cosmic ray ionization rates that are several dex larger than the galactic background. The background rate is an open question. McCall et al. (2003) find a background rate 1.2×10^{-15} s^{-1}, about 48 times larger than our default.

11.6.2. cosmic rays rate -17.3

If the keyword **rate** appears then the command specifies the log of the H^0 ionization rate [s^{-1}] in predominantly neutral gas. The default is 2.5×10^{-17} s^{-1} taken from Williams et al. (1998).

11.6.3. cosmic rays density =1.2 [index, etc.]

If the keyword **density** appears then the first number is the log of the cosmic ray density $n(cr)$ [cm^{-3}]. Expressions given in Ferland & Mushotzky (1984) convert this into an ionization rate.

The second optional number is a power-law index α that describes the variation of the cosmic ray density with radius, i.e.,

$$n(cr,r) = n(cr,r_o)\left(\frac{r}{r_o}\right)^{\alpha} \ [\text{cm}^{-3}] \ . \tag{57}$$

The default value of the index is $\alpha = 0$, corresponding to a constant cosmic ray density. The third optional number is the log of the temperature of the fast electrons, if they are not relativistic. If this third number is specified then expressions from Balbus and McKee (1982) will be used to evaluate the electron heating rates. The options can be omitted from right to left.

11.7. failures 100 times [map]

A converge failure occurs when the heating-cooling balance, the electron density, or pressure, is not within a certain tolerance, set by the **set convergence** commands (page 183 below). Normally Cloudy will punt[24] after an excessive number of convergence failures (presently 20) occur. This command increases the number of allowed failures to the value entered as a parameter.

When Cloudy stops because of excessive failures and the **map** option is specified[25] it first produces a map of heating-cooling versus temperature to give an indication of where the equilibrium temperature should have been. A section in Part 3 describes thermal failures in more detail, and describes this output.

Failures occur most often when a thermal front occurs. This is when the solution needs to jump over the peaks in the cooling function that occur near 2000 K and 10^5

---

[24] FAQ: Punt is a technical term from American football. It is something bad that happens when progress in advancing the ball is lacking.

[25] In versions 94 and before, the default was to produce the map, and the **no map** option turned this off. With version 95 the option is not to produce a map, and this must be requested with the **map** option.

K. A warning will be issued at the end of the calculation if there is a discrepancy in the global heating balance.

It should not be necessary to use this command. Please contact me if you find a simulation where this is necessary.

11.8. force temperature to 3400K

This command forces the initial estimate of the temperature of the first zone to the value entered. The temperature is interpreted as a log if it is ≤ 10 and the linear temperature otherwise. The keywords **_log** and **linear** will override this.

This command is useful if more than one initial temperature solution is possible. It forces the first guess of the temperature to the specified value, but *does not* hold the temperature constant; the temperature is determined by energy balance thereafter. (Constant temperature is set with the **constant temperature** command.)

Cloudy may have trouble finding a valid first solution if the initial solution is forced well away from the equilibrium value. This is an inevitable consequence of the complete linearization methods that are intrinsic to the code's solvers. If a large number of thermal failures or warnings result from the use of this command then it is likely that the code has been forced too far away from the solution. This command should not be used in this case.

11.9. hextra -14 [scale r=18, thickness 12]

This includes extra heating due to some unspecified energy source. The first number H_0 is the log of the volume-heating rate [erg cm^{-3} s^{-1}]. The second number is the log of the scale radius r_{scale} [cm]. The extra heating rate varies as[26]

$$H = H_0[\exp(-depth/r_{scale}) + \exp(T - depth)/r_{scale})] \ [\text{erg cm}^{-3} \text{ s}^{-1}]. \tag{58}$$

The default, when r_{scale} is not specified, is constant extra heating. The third optional parameter T is the total thickness of the slab. If this is given then the second exponential term will be added. This will mimic an external heat source that warms the cloud from both the illuminated and shielded faces. If the third parameter is not entered then the rightmost term is not included.

This process is coded in routine *highen* where the variable *TurbHeat* is used. This code can be to changed to other functional forms of extra heating.

11.10. high temperature approach

This command tells the code to search for the first temperature by approaching the thermal solution from the high temperature extreme of 10^6 K. Normally the approach is from low temperatures. This can be useful when more than one thermal solution is possible.

---

[26] In versions through 94.00 the heating rate varied as $exp[-r_{scale}/(r-r_o)]$ and went to infinity as the illuminated face. The radial dependence was changed to its current form in 94.01.

11.11. magnetic field, log(B) = 5 [options]

Magnetic fields are not included by default. This specifies the strength and geometry of the magnetic field. A number, the log of the magnetic field strength in Gauss, is the first number on the line. The physical effects of magnetic fields are discussed in Part 2 of this document.

Ordered and tangled fields can be specified. The field is assumed to be tangled by default.

If the keyword **ordered** appears then the angle between the radiation field of the central object and the magnetic field is also specified. This angle is zero if the field is in the outward radial direction. The angle is given in degrees by default but the **radian** keyword will cause it to be radians instead.

In the case of a tangled field the code will look for a second number, the index for the gamma-law relation between the magnetic field and the gas density. If a second number is not found then an index of $\gamma = 4/3$ will be assumed. This index appears in the relationship

$$B_{\text{tangled}} = B^0_{\text{tangled}} \left(\frac{\rho}{\rho_0} \right)^{\gamma/2} \quad \text{[Gauss]} \tag{59}$$

where the term in parenthesis is the ratio of the current density to the density at the illuminated face of the cloud. For a spherical constant-velocity flow, where $\rho/\rho_0 \propto (r/r_0)^{-2}$, the field will vary as $B/B_0 \propto (r/r_0)^{-\gamma}$. This is in contrast with a dipolar field, in which the radial dependence is $B/B_0 \propto (r/r_0)^{-3}$.

Both ordered and tangled fields can be specified on separate commands. If more than one ordered or tangled field is specified, the second will take precedence over the first.

The major thermal effect of a field is to add cyclotron cooling. This is only important at very high temperatures.

Magnetic pressure and enthalpy terms corresponding to the magnetic energy density are included in the equation of state when a constant-pressure geometry is assumed. The plasma β parameter, defined as the local ratio of magnetic to thermal energy densities, $\beta = P_{gas} / P_{mag}$, is printed in the summary of pressure contributions. The field affects the dynamics of the gas when $\beta \leq 1$.

Supersonic turbulence in rough energy equipartition with the magnetic field is often observed in the cold ISM (Heiles & Crutcher 2005). The **turbulence** command (page 100 above) has an **equipartition** option to set the turbulent velocity to a value in energy equipartition with the magnetic field. Although this correlation is observed in the cold neutral medium in the ISM, it is not thought to be a fundamental relationship (see the Heiles & Crutcher paper).

11.12. map, zone 4 [range 2000, 5000]

This tells the code to compute a heating-cooling map of the specified zone. This is a useful way to check for the existence of more than one thermal solution. If no zone

is specified, or if the zone is less than or equal to 0, then only a thermal map is produced for the illuminated face of the cloud and the calculation then stops. The calculation of the heating and cooling is self-consistent. A section in *Problems* in Part 3 of this document explains how to interpret the map output.

The map produced by this command is not directly comparable to the more typical plot that shows the equilibrium temperature as a function of ionization parameter (Krolik, McKee, and Tarter 1981). That map can be produced by successively calling Cloudy with the same ionizing continuum but different densities. In this second case each deduced temperature is a valid equilibrium temperature. In the map produced by the **map** command only one temperature is a valid equilibrium temperature. The map produced by this command is useful for checking for more than one thermal solution, to check that the heating and cooling curves smoothly flow as the temperature changes, or to investigate why the code had convergence problems (it was originally introduced for only this latter purpose).

The optional keyword **range** specifies the temperature range of the map. If this option is specified then the first number on the line must be the zone for the map, zero if only a map of the first zone, and the next two numbers must be the lower and upper limits to the map. These temperatures will be interpreted as logs if the first number is ≤ 10. Normally about 20 steps occur between the lowest and highest temperature in the map. The number of steps can be reset with the **set nmaps** command, discussed on page 191 below.

The thermal map can be saved as a separate file with the **punch map** command (page 149 below). This produces output that is suitable for processing by other software.

The code stops when the map is complete since it is left in a disturbed state.

11.13. neutrons -2 [efficiency =-2]

This adds energy deposition and ionization by secondaries due to the fast neutrons proposed by Sikora, Begelman, and Rudak (1989). The argument is the luminosity in fast neutrons expressed as a fraction of the *total* photon luminosity of the incident continuum. It is interpreted as a log if ≤ 0, and a linear scale factor if positive.

The second optional argument is the heating – ionization efficiency of the neutrons, unity by default. Both quantities are interpreted as logs if ≤ 0, and linear if positive.

11.14. print coolants, zone 135

See page 123 below.

11.15. print heating

See page 125 below.

11.16. set temperature [floor, solver, convergence]

This is described on page 194 below.

11.17. tlaw [DB96, SN99]

This specifies a temperature law. Although code does exist to allow the range of options enjoyed by the **dlaw** command (page 110 above) only two options currently are implemented.

The **DB96** option tells the code to use the temperature – column density law used by Draine & Bertoldi (1996):

$$T = T_0 / \left[1 + N(\mathrm{H}) \sigma_{d,1000} \right] \ [\mathrm{K}] \tag{60}$$

where T_0 is 500 K and σ is 6×10^{-22} cm².

The **SN99** option tells the code to use the temperature – H_2 fraction relationship assumed by Sternberg & Neufeld (1999);

$$T = \frac{500}{1 + 9 \left[2n(\mathrm{H}_2) / n(\mathrm{H}_{tot}) \right]^4} \ [\mathrm{K}]. \tag{61}$$

11.18. tolerance 0.001

See the **set temperature convergence** command described on page 194 below.

12. STOPPING CRITERIA

12.1. Overview

Cloudy will stop at some depth into the cloud. The physics that sets this depth is important since it can directly affect predicted quantities.

Two geometries, matter bounded and radiation bounded, can be identified. A radiation-bounded cloud is one where the outer edge of the emitting gas is defined by a hydrogen ionization front. In this case the calculation stops because nearly all ionizing radiation has been attenuated and the temperature falls below 4000 K, the default lowest allowed electron temperature. This choice of lowest temperature was made with optical emission lines in mind. Setting another outer limit is not necessary unless molecules or lines with very low ionization and excitation potentials (i.e., the [C I] or [O I] far infrared lines) are of interest. It would then be necessary to lower the stopping temperature with the **stop temperature** command (described on page 118 below).

In a matter-bounded cloud the gas is optically thin to hard radiation and the outer radius of the cloud must be specified. This could be a column density, physical thickness, or optical depth. More than one stopping criterion can be specified, and the calculation will stop when the first one is met. Cloudy will say why it stopped after the results of the last zone calculation are printed.

If no stopping criteria are set the calculation will usually stop because the default lowest temperature (4000 K) or the default greatest number of zones (1400) was reached.

12.2. Danger! Understand why the calculation stopped!

There are circumstances in which the predicted emission-line spectrum will depend strongly on the thickness of the cloud, which is set by the stopping criteria. This happens if the calculation ends within a line's creation region. This is nearly always the case for some lines in an X-ray irradiated gas, and for any radiation field and molecular or low-ionization infrared lines.

There are several checks that should be made to confirm that the spectrum is the one expected and not an artifact of the stopping criteria. The first and most important is to understand *why* the calculation stopped. This is explained in the first comment after the last zone is printed. To find this, first locate the print out for the last zone. The following example, from the *pn_paris* simulation, shows the printout that includes the last zone's results and the start of the calculation's summary. The first line after the ionization distribution of iron gives the title for the model, and the line after that gives the reason that the calculation stopped.

```
####150  Te:3.978E+03 Hden:3.000E+03 Ne:1.276E+02 R:4.062E+17 R-R0:3.062E+17 dR:5.658E+13 NTR:  5 Htot:4.094E-19 T912: 9.97e+07###
Hydrogen    9.70e-01 2.97e-02 H+o/Hden 1.00e+00 3.57e-09 H-    H2 1.57e-07 5.07e-10 H2+ HeH+ 8.59e-08 Ho+ ColD 3.27e19 8.86e+20
Helium      8.77e-01 1.23e-01 1.81e-04 He I2SP3 4.78e-08 5.43e-16 Comp H,C 1.39e-26 3.59e-27 Fill Fac 1.00e+00 Gam1/tot 6.57e-01
Carbon      2.50e-04 9.97e-01 3.12e-03 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
Sulphur   0 7.02e-05 9.64e-01 3.56e-02 2.36e-06 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
Argon     0 7.82e-01 1.96e-01 2.24e-02 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
Iron      0 9.25e-06 9.97e-01 2.80e-03 3.42e-06 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
    parispn.in Meudon Planetary nebula
  Calculation stopped because lowest Te reached.    Iteration  1 of  1
  The geometry is spherical.
  !Some input lines contained [ or ], these were changed to spaces.
  !Non-collisional excitation of [OIII] 4363 reached   2.35% of the total.
  !AGE: Cloud age was not set.  Longest timescale was 3.68e+012 s = 1.17e+005 years.
  Suprathermal collisional ionization of H reached 9.48% of the local H ionization rate.
  Charge transfer ionization of H reached 8.94% of the local H ionization rate.
```

Left on its own, the code will probably stop when the temperature falls below the default lowest temperature of 4000 K. This is what happened in the preceding example. This temperature was chosen for two reasons; a) collisionally excited optical and ultraviolet lines generally form in gas hotter than this (but infrared lines will form at far lower temperatures) and b) more than one thermal solution is possible for temperatures around 3000 K (Williams 1967), so thermal instabilities may result if the gas extends to cooler temperatures.

It is a good idea to check whether the predictions would change if the model were made thicker or thinner. It is safe to assume that a line's luminosity does not depend on the thickness of the cloud if either a) the final temperature is well below the excitation potential of the line, or b) the gas is more neutral than the species of interest.

The description of the output produced by the code, located in Part 3 of this document, lists all the possible reasons for stopping.

12.3. radius inner =18 [thickness =16; parsecs; linear]

The **radius** command is discussed on page 79 above. The optional second number sets the thickness or outer radius of the cloud.

12.4. stop AV 12.1 [point, extended]

This stops a calculation at a specified visual extinction A_V. The value is the extinction in magnitudes at the isophotal wavelength of the V filter (5500Å). The number is the linear extinction in magnitudes unless it is negative, when it is interpreted as the log of the extinction. Note that there must be spaces before and after the key "**AV**".

Properties of grains are described in the section on the **grains** command, starting on page 60 above. The distinction between the extinction for a point versus an extended source is described on page 66 above. By default the A_V specified with this command will be for a point source, which is the quantity measured in extinction studies of stars. The extended-source extinction can be specified instead by including the keyword **extended**.

12.5. stop column density = 23 [neutral; ionized; total; ...]

This stops the calculation at the specified hydrogen column density $N(H)$ [cm^{-2}]. There are several optional keywords, which determine whether the column density is the total (the default), the ionized hydrogen column density, the neutral hydrogen

column density, or several other column densities. The default stopping column density is 10^{30} cm^{-2}.

12.5.1. stop column density 23

The number is the log of the total hydrogen column density (atomic, ionic, and all molecular forms), defined as the integral

$$N(\mathrm{H}) = \int \left\{ n(\mathrm{H^0}) + n(\mathrm{H^+}) + 2n(\mathrm{H_2}) + \sum_{other} n(\mathrm{H_{other}}) \right\} f(r)\,dr \ [\mathrm{cm^{-2}}] \tag{62}$$

where $f(r)$ is the filling factor.

12.5.2. stop neutral column density 23

The number is the log of the atomic hydrogen column density

$$N(\mathrm{H^0}) = \int n(\mathrm{H^0}) f(r)\,dr \ [\mathrm{cm^{-2}}]. \tag{63}$$

12.5.3. stop ionized column density 23

The number is the log of the ionized hydrogen column density

$$N(\mathrm{H^+}) = \int n(\mathrm{H^+}) f(r)\,dr \ [\mathrm{cm^{-2}}]. \tag{64}$$

12.5.4. stop atom column density 21.3

In some PDR literature the atomic hydrogen column density is defined as the sum of $\mathrm{H^0}$ and $\mathrm{H_2}$. This command allows the calculation to stop as this summed column density, defined as

$$N(\mathrm{H^0} + 2\mathrm{H_2}) = \int \left[n(\mathrm{H^0}) + 2n(\mathrm{H_2}) \right] f(r)\,dr \ [\mathrm{cm^{-2}}]. \tag{65}$$

This command was added by Nick Abel. Note that this counts each $\mathrm{H_2}$ as two hydrogen atoms.

12.5.5. stop H/TSpin column density 17.2

This calculation stops at the specified integral

$$N(\mathrm{H^0})/T_{spin} = \int \frac{n(\mathrm{H^0})}{T_{spin}} f(r)\,dr \ [\mathrm{K\ cm^{-2}}]. \tag{66}$$

Note that T_{spin} is computed self-consistently, including the effects of pumping by the background continuum (usually the CMB) and the local $L\alpha$ radiation field.

12.5.6. stop H2 column density 19.2

The calculation stop at the specified column density in molecular hydrogen

$$N(\mathrm{H_2}) = \int n(\mathrm{H_2}) f(r)\,dr \ [\mathrm{cm^{-2}}]. \tag{67}$$

The 2 in $\mathrm{H_2}$ must come before the log of the column density in the command. This is really the $\mathrm{H_2}$ column density, not twice it.

12.5.7. *stop CO column density 19.2*

The calculation stop at the specified column density in CO

$$N(\mathrm{CO}) = \int n(\mathrm{CO}) f(r) dr \; [\mathrm{cm}^{-2}] \; . \tag{68}$$

This command was added by Nick Abel.

12.5.8. *stop effective column density 23*

This is actually a form of the **stop optical depth** command (see page 116 below). Usually, low-energy cutoffs in X-ray spectra are parameterized by the equivalent column density of a cold neutral absorber with cosmic abundances. Actually what is measured is an optical depth at some energy, generally around 1.0 keV. If the gas is ionized then a much larger column density will be needed to produce the observed absorption. The difference can be more than an order of magnitude. This command stops the calculation when the incident continuum has been attenuated by the appropriate absorption at 1.0 keV. The calculation stops when the absorption optical depth at 1.0 keV (neglecting scattering opacities) reaches a value of

$$\tau_{abs}(1.0\,\mathrm{keV}) = N_{effec}\, 2.14 \times 10^{-22} \tag{69}$$

at 73.5 Ryd. The argument of this command is the log of the effective column density N_{effec}. The absorption cross-section per proton for cold neutral gas is taken from Morrison & McCammon (1983). Scattering opacities *are not* included in this optical depth. No attempt is made to use realistic physical conditions or absorption cross sections – this command follows the Morrison & McCammon paper.

If the gas is highly ionized then the actual column density will be greater than the effective column density. If the abundances of the heavy elements are greatly enhanced, then it will be less.

12.6. stop depth . ….

This is another name for the **stop thickness** command, described on page 118 below.

12.7. stop eden 3 [linear]

The calculation stops when the electron density falls below the indicated value. The number is the log of the electron density [cm^{-3}]. The optional keyword **linear** will force the argument to be interpreted as the quantity itself, not its log. This command is one way to stop constant temperature models. For instance, the calculation can be forced to stop at the H^+-H^0 ionization front by setting the stopping electron density to approximately half of the hydrogen density.

The following examples show a case that will stop near the He^{2+}-He^+ ionization front (for solar abundances) and a case that will stop near the H^+-H^0 ionization front.

```
#
# stop at the He++ - He+ ionization front
hden 9
stop eden 9.06 # stop when helium (10% by number) is He+
#
# stop at H+ - H0 ionization front
hden 5
stop eden 4.5 # stop when electron dens falls below H density
```

The default is an electron density of -10^{30} cm^{-3}. (The negative sign is not a typo.)

12.8. stop efrac = 1.05

The model will stop when the electron fraction, defined as the ratio of electron to total hydrogen densities, falls below the indicated value. This is another way to stop calculations at ionization fronts, and is useful if the hydrogen density there is not known beforehand (this occurs in constant pressure calculations, for instance). The argument is the fraction itself if it is greater than zero, and the log of the fraction if it is ≤ 0.

The default is an electron fraction of -10^{37} cm^{-3}. (The negative sign is not a typo.)

12.9. stop line "C 2" 157.6m reaches 0.2 rel to "O 3" 5007

The calculation will stop when the emission line with the label given within the first pair of quotes and the wavelength given by the first number exceeds an intensity given by the second number, relative to an optional second emission line. In this example the calculation will stop when the intensity of [C II] 158 μm reaches 0.2 relative to [O III] λ5007. If a second optional line is not entered it will be Hβ. This can be a useful way to stop matter-bounded models. The results of this command are not exact; the final intensity ratio will be slightly larger than the ratio specified.

The line label and wavelength should be entered exactly as it appears in the emission line output – the normal units are Angstroms, but microns ('m'), centimeters ('c'), and others are also recognized.

The scaling of the line intensities on the final printout can be changed with the **normalize** command, as described on page 120 below. That command can change both the normalization line (usually Hβ) and its relative intensity (usually 1). The **normalize** command does not interact with the **stop line** command. If the third number is not entered with the **stop line** command then Hβ is always used as the line in the denominator in the ratio. The **stop line** command always uses the ratio of the two line intensities, even if the scale intensity of the second line has been reset with the **normalize** command.

Up to 10 different **stop line commands** may be entered. If more than one **stop line command** is entered then the code will stop as soon as any of the limits is reached.

12.10. stop mass 32.98

The calculation will stop when the total mass of the computed structure exceeds the quantity entered. If the inner radius is specified then the entered number is the

log of the mass in gm. If the inner radius is not specified then it is the log of the mass per unit area, gm cm^{-2}.

At the current time no attempt is made to make the computed mass exactly equal to the entered number. The calculation will stop after the zone where the mass is first exceeded.

12.11. stop mfrac = 0.5

The calculation stops when the hydrogen molecular fraction, defined as $2n(H_2)/n(H_{tot})$, exceeds the indicated value. This is a way to stop calculations within a PDR. The argument is interpreted as the molecular fraction itself if it is greater than zero, and the log of the fraction if it is less than or equal to zero.

The default is a molecular fraction of -10^{37} cm^{-3}. (The negative sign is not a typo.)

12.12. stop molecule depletion -2

Nick Abel incorporated the condensation of molecules onto grain surfaces. Currently CO, H_2O, and OH condensation are treated. The density of molecules on grains is given as a species with the molecule's usual label, "CO", "H2O", or "OH", followed by the string "gr". The rates of UV and cosmic ray desorption along with accretion come from Hasegawa, Herbst, & Leung (1992) and Hasegawa & Herbst (1993). The grain surface chemistry mentioned in these papers is not included so our treatment of condensation is closer to the work of Bergin, Langer, & Goldsmith (1995), who also use the Hasegawa rates. Use the **no grain molecules** command (page 180 below) to turn off condensation.

The chemistry network will become unstable when oxygen is highly depleted from the gas phase. The code stops when a certain fraction of oxygen is in the form of molecules frozen out on grain surfaces. By default the code stops when 99% of the oxygen abundance has condensed out of the gas phase. This command changes that default. 116 above

12.13. stop optical depth -1 at 2.3 Ryd

This command stops the calculation at an arbitrary continuum *absorption* optical depth. The first number is the log of the optical depth. The optical depth is interpreted as a log by default, but if the **linear** keyword occurs then the number is interpreted as the linear value. The optical depth does not include scattering opacities. The second number is the energy in Rydbergs. It is interpreted as a log if it is negative, as linear if positive, and must be within the energy bounds considered by the code (presently 1.001×10^{-8} Ryd to 7.354×10^6 Ryd). At present, only one stopping optical depth can be specified. If more than one is entered then only the last is honored.

It is traditional in X-ray astronomy to characterize low-energy cut-offs as the equivalent *completely neutral* column density for *solar* abundances. This is not correct when the gas is ionized (since the high energy absorption opacity is diminished) or when the abundances of the heavy elements are enhanced (the high-energy opacity is

increased). For extreme cases these effects can change the opacity by more than an order of magnitude. The deduced column density is underestimated by the same amount. It is better to convert the deduced column density back into an optical depth at 0.5 or 1 keV (this is actually the observed quantity), and use this optical depth and energy as the stopping criteria, than to use the deduced column density as a stopping criterion. Either this command, or the **stop effective column density** command (which is actually a form of the **stop optical depth** command, see page 112 above) can be used to stop the calculation at an X-ray optical depth corresponding to a certain low-energy absorption.

The optical depth used in this command is the absorption optical depth, and does not include scattering opacities. In general, the effects of scattering opacities are much more geometry dependent than absorption opacities.

12.13.1. stop Balmer optical depth = -.3

This command is a special case of the **stop optical depth** command in which the energy does not need to be specified, but the keyword **Balmer** is given. It will stop the calculation when the log of the absorption optical depth at the Balmer continuum threshold ($\nu = 0.250$ Ryd) reaches the specified value. The default is $\tau_{Bac} = 10^{20}$, and the optical depth is always interpreted as a log. This is the *total absorption* optical depth at the Balmer edge, and includes all computed absorption opacity sources such as grains or free-free absorption, but neglects scattering.

12.13.2. stop Lyman optical depth = 5

This is a special case of the **stop optical depth** command, in which the energy does not need to be specified, but the keyword **Lyman** is given. The number is the log of the Lyman limit optical depth, τ_{912}. The default value is $\tau_{912} = 10^{20}$. The stopping criterion is *really* the *total* 912Å *absorption* optical depth, and *not* the hydrogen Lyman limit optical depth at 912Å. These are not exactly the same, especially when grains are present or the abundances of the heavy elements are enhanced.

12.13.3. stop 21cm optical depth = linear 3

This stops the calculation when the *line center* optical depth of H I λ21 cm reaches the entered value. The populations of the hyperfine levels are determined by solving the full level populations with the effects of pumping by Lα and the external continuum included. The calculation stops when the optical depth from the continuum source to the point exceeds the entered value. The actual optical depth with be greater than or equal to the indicated value.

Note that Cloudy always uses the dimensionless line center optical depth. The integrated optical depth is the standard used in radio astronomy. The integrated optical depth is called the mean optical depth in optical spectroscopy. The dimensionless mean optical depth is $\sqrt{\pi}$ larger than the line center optical depth. In radio astronomy, the integrated optical depth is often given in velocity units, km s^{-1}.

12.14. stop pfrac = 0.23

The calculation will stop when the proton fraction, defined as the ratio of proton (ionized hydrogen) to total hydrogen densities, falls below the indicated value. This is another way to stop calculations at ionization fronts, and is useful if the hydrogen density there is not known beforehand (this occurs in constant pressure calculations, for instance). The argument is interpreted as the fraction itself if it is greater than zero, and the log of the fraction if it is less than or equal to zero.

The default is an proton fraction of -10^{37} cm^{-3}. (The negative sign is not a typo.)

12.15. stop temperature 1,000K [linear, exceeds]

The calculation will stop if the electron temperature drops below T_{low}, the argument of this command. The argument is interpreted as the log of the temperature if ≤ 10, and as the linear temperature if > 10 or if the **linear** keyword appears.

The default value is $T_{low} = 4000$ K. Gas cooler than this produces little optical emission, but may be a strong emitter of infrared lines such as the [C II] 158 μm or the [O I] ^3P lines. The lowest temperature allowed, T_{low}, should be adjusted so that the energy $h\nu$ is $\gg kT_{low}$ for the lowest excitation potential ($h\nu$) transition to be considered. Note that more than one temperature is sometimes possible when $T \approx 10^3$ K (Williams 1967), so thermal stability problems may develop if T_{low} is lowered below a few thousand degrees Kelvin. This issue is discussed further in a section in Part 3 of this document.

If the keyword **exceeds** appears on the line then the specified temperature will be the highest allowed temperature. The calculation stops if the temperature *exceeds* the value on the command. This might be necessary if an entire grid of models is to be computed by calling the code as a subroutine, but those in the high temperature phase (i.e., $T > 10^5$ K) are not of interest. The other rules for the command are unchanged.

If no number appears on the command line, but the keyword **_off** does, then temperature will not be used as a stopping criterion.

12.16. stop thickness 9.3 [parsecs; linear; 23 on sec iter]

This sets an upper limit to the thickness of the model. The argument is interpreted as the log of the thickness unless the keyword **linear** appears. The default units are centimeters, but it will be interpreted as the log of the thickness in parsecs if the keyword **parsec** appears on the line.

The **stop thickness** command has the same effect as the optional second number on the **radius** command (page 79 above). This command makes it possible to set a cloud thickness when the inner radius is not specified, such as when the ionization parameter is given.

Up to 200 thicknesses may be entered on the command line. Each will be the ending thickness for consecutive iterations. The limit to the number of stopping values is set by the limit to the number of iterations that can be performed. If fewer

numbers are entered than iterations performed, then the last number will be used for all further iterations.

The keyword **depth** can be used instead of thickness.

12.17. stop zone 123 [21 on sec iteration, ...]

This sets a limit to the number of zones that will be computed on each iteration. It is not normally used. In this example the calculation will stop after computing 123 zones. The default value is 1400. Up to 200 numbers may be entered, each being the ending zone for consecutive iterations. This limit is set by the limit to the number of iterations that can be performed. If fewer numbers are entered than iterations performed, then the last number will be used for all further iterations.

After the calculation is complete, the code checks that it did not stop because it reached the default number of zones. A warning will be generated if this happens, since it was probably not intended. To extend the default number of zones while keeping this checking active, use the **set nend** command (page 190 below).

13. CONTROLLING OUTPUT

13.1. Overview

Cloudy is capable of keeping a printer going for hours, although its default output is minimal. Several commands vary the printer's mass-loss rate, and are described here. A section of Part 3 of this document describes the meaning of the output.

13.2. normalize to "O 3" 5007 [scale factor = 100]

The strength of an emission line can be given either in energy units, as in erg cm^{-2} s^{-1}, or as its energy relative to another line. The code prints both. In the main printout each line is printed with a label and wavelength, followed by the energy radiated in the line, ending with the intensity relative to a reference line.

Emission-line intensities are usually listed relative to the intensity of Hβ λ4861Å, the default reference line. By default the reference line has an intensity of unity. This command can change the reference line to any of the other predicted lines, and can change the relative intensity of the reference line to another value. The entire emission-line spectrum will have relative intensities normalized to the intensity of the line whose line label is with the double quotes and with wavelength given by the first number. The label must be the four character string that identifies the line in the print out[27], and the wavelength must match the printed wavelength to all four figures. The wavelength units must appear if they are not Angstroms.

The optional second number sets the relative intensity of the reference line. If it is equal to 100, as in this example, then all intensities will be relative to a reference line intensity of 100. The default is for an intensity of unity. The example given above will cause the line intensities to be expressed relative to an [O III] λ5007 intensity of 100. The scale factor must be greater than zero.

The code works by finding the first line in the emission-line stack whose wavelength and label matches the line. There is a possible uniqueness problem since more than one line can have the same wavelength. This is especially true for XUV or soft X-ray lines.

The following shows some examples of the **normalize** command:

```
// normalize to spectrum to Pa
normalize to 1.875m
// normalize spectrum to the [OI] IR line on a scale where it is equal to 100
normalize to [O  1] 63.17m = 100
```

13.3. plot [type, range]

Plots of several predicted quantities can be made. One of the keywords described below must appear on the command line. Up to 10 plots can be generated. The keyword **trace** will turn on a great deal of information concerning the mechanics of generating the plot.

---

[27] The label was optional in versions 94 and before of the code, but now is required due to the large number of lines, making unique wavelengths unusual.

Publication-quality plots can be produced using the **punch** commands (described beginning on page 129 below) to produce a file that can then be post-processed using other plotting software.

13.4. plot continuum [_raw, trace, range][28]

The keyword **continuum** results in a plot of the continuum. This energy range is altered by entering the two optional limits with the **range** key. This is described in detail in section 13.5.1 below.

The default is to plot both the continuum (in units of νf_ν) that is incident upon the cloud (plotted as .'s) and transmitted through the cloud (the o's). If the option **raw** is specified then the continuum in units actually used inside Cloudy (cm^{-2} s^{-1} cell^{-1}) will be plotted. If the keyword **photon** appears then the units of the plotted continuum will be photons cm^{-2} s^{-1} Ryd^{-1}.

13.4.1. plot continuum keywords

It is possible to plot specific components of the continuum with the following series of keywords.

13.4.2. plot diffuse continuum

This plots the diffuse emission per unit volume within the last computed zone. This continuum is the local gas and grains emission in the optically thin limit and unity filling factor.

13.4.3. plot emitted continuum

The net integrated continuum produced by the cloud is plotted. This is the sum of the continua emitted in the inward and outward directions from the computed ionization structure and does not include the incident continuum.

13.4.4. plot outward continuum

The contents of the *outcon* and *flux* arrays, multiplied by the local gas opacity, are plotted to indicate sources of ionization and heating.

13.4.5. plot reflected continuum

The continuum emitted from the illuminated face of the cloud is plotted. This includes the back-scattered portion of the incident continuum along with the diffuse continuum emitted from the cloud in the direction towards the central object. This is possible only for non-spherical (open) geometries.

---

[28] Today most plots are generated by producing punch output, then post-processing that output in other software. The plot commands described here still function but are likely to be removed in a future version of the code.

13.5. plot opacity [type, range]

The opacity (total cross section per hydrogen atom) of the first and last zones is plotted. The continuum between 1.001×10^{-8} Ryd $\leq h\nu < 7.354 \times 10^{6}$ Ryd is usually plotted, unless this is adjusted by using the **range** option.

There are three optional keywords; **absorption**, **scattering**, and **total**, to change which opacity is plotted. If none appear then the total opacity is plotted.

13.5.1. plot range options

There is a further keyword **range** that specifies the energy range of the **opacity** and **continuum** plots. If one number occurs on the line then it is the lowest energy in Rydbergs. If the first number is zero then it is the lowest energy in the continuum, 1.001×10^{-8} Ryd. The optional second number is the highest energy shown on the plot. If it is omitted or zero then it is the high-energy limit of the code, presently 7.354×10^{6} Ryd. If either number is negative then both are interpreted as the logs of the energies, otherwise they are assumed to be the linear energy. If the first number is zero (i.e., interpreted as the lowest energy considered by the code) then the second number is interpreted as the energy of the upper limit to the plot, and not its log.

The following give specific examples of the range option.

```
// plots the absorption opacity between 0.1 to 10 Ryd.
plot absorption opacity, range=.1 to 10 Ryd
//
// plot the opacity between 1 Ryd and
// the high energy limit of the code.
plot scattering opacity, range=1
//
// the range will be the full energy limit of the code
plot opacity
```

13.6. plot _map [Tmin=3,000K, Tmax=20,000K, linear, range]

If the keyword **_map** (note the leading space) appears then a plot of the heating and cooling rates [erg cm^{-3} s^{-1}] as a function of temperature will be made. This is for the attenuated continuum and physical conditions in the last computed zone.

13.6.1. plot map range options

The high and low temperatures on the map can be changed by entering the keyword **range** and one or two optional numbers. If no number appears then a temperature range of 10 K to 10^{9} K is used. If only one number appears then only the lower temperature limit is changed. If two numbers appear then both lower and upper limits are changed.

If the first number is ≤ 10 then both numbers are interpreted as logs of the temperature. If the first number is > 10 then both are interpreted as the temperature itself. If the keyword **linear** appears then both numbers are interpreted as the temperature itself no matter how large or small they may be.

The number of points on the map is set with the **set nmaps** command described on page 191 below.

13.7. print ages

The code normally assumes that the system is old enough for microphysical processes to have become time steady. This tells the code to print all of the timescales tracked by the code and used by the **age** command (page 75 above). Normally only the shortest timescale is printed at the end of the calculation.

If a physical process is not significant, for instance, the H_2 formation timescale in a coronal gas, the age is still computed but is set to a negative number. This retains the value while not including the process when the timescales are determined.

13.8. print arrays

This tells the code to print the ionization balance arrays for all elements. In the output the first line is the vector of ionization rates [s^{-1}]. The second line is the vector of recombination rates [s^{-1}]. If there are too many ionization stages to be printed across the line then the first number will indicate how many ionization stages are "off the page to the left", i.e., if the first number is 11, then the first ionization stage is 12.

13.9. print constants

The physical constants stored in the header file *physconst.h* will be printed.

13.10. print column densities [on; off]

This controls whether the column densities of the various constituents are printed. The keywords are **_ON_** and **_OFF**.

The column densities of several excited states within ground terms of some species are printed as well. The meaning of the labels for the excited states column densities is given in the discussion of *cdColm* in Part 3 of this document.

13.11. print coolants, zone 135

This prints the coolants for the specified zone. If no zone number or 0 appears on the line then the coolants for *all* zones will be printed. The total cooling and the fractional contribution of significant coolants are printed. Only the strongest coolants are printed. For each coolant a four-character label gives an indication of the spectroscopic origin of the coolant and the following integer gives its wavelength, with a 0 to indicate a continuum. The last number of the group is the fraction of the total cooling carried by that agent.

13.12. print continuum [block]

This is actually two very different commands, with different purposes. The second form will eventually be removed.

13.12.1. Adding entries into the emission line stack

This includes portions of the predicted continua in the main emission-line output. If no keywords appear then the code will print the sum of the total emitted and inward reflected continuum as a series of entries in the standard emission line array.

Each has a label *nFnu*. This is the emission produced by the cloud and does not include the attenuated incident continuum. The entry *nInu* is the transmitted plus reflected portion of the continuum. The **set nFnu** command (page 190 below) provides a way to change which contributors are included in this prediction.

If the **print line inward** command (page 127 below) also occurs in the input stream then the total inward emission (the label "InwT") and the reflected incident continuum (label "InwC") will also be printed.

The energies that are printed is stored in the vector *EnrPredCont* that is initialized with *NPREDCONT* points in *zerologic.c*. These can be changed if the continuum at other energies is needed.

13.12.2. *Adding a block of information after the calculation*

If the keyword **block** appears then the code will print the transmitted continuum at the end of the calculation. The information includes the integrated Balmer and Lyman continua, the transmitted X-ray continuum, and frequency-by-frequency continuum intensities.

The **punch continuum** command (page 131 below) provides this information in a far more useful format. This option on the **print continuum** command will be removed in a future version of the code.

13.13. print departure coefficients

LTE departure coefficients for an element along the H-like or He-like isoelectronic sequences will be printed. The **print populations** command (page 125 below) controls printing individual level populations.

If the keyword **He-like** appears some an element on the helium-like isoelectronic sequence will be printed. Otherwise an element of the H-like isoelectronic sequence is chosen. The code will search for the name of an element, and if it finds one, will print that element and isoelectronic sequence. If none are recognized then departure coefficients for H I (the H-like sequence) or He I (the He-like) are printed.

13.14. print errors

The code will always identify problems that occur during a calculation by either printing comments during the calculation or warnings after the calculation is complete. This tells the code to also print these warnings to **stderr**. On many systems this output can be redirected to the screen.

13.15. print every 1000 [5 37 93]

Cloudy will always print the results for the first and last zones. This command varies the number of zones printed between the first and last. If more than one number is entered then each applies to a successive iteration. In the example above, it will print every 1000 zones on the first iteration, every 5 zones on the second iteration, 37 on the next, etc. If there are fewer numbers entered than iterations performed, then the last number entered will be used for all further iterations.

Normally about 100 to 200 zones are computed per model, so printing every five or ten zones on the last iteration may sometimes be useful. The default condition is to print only the first and last zones.

13.16. print heating

The relative heating due to each stage of ionization or physical process is printed. The number is the fraction of the total heating due to this particular stage of ionization, and is printed directly below the relative abundance of that stage.

13.17. print populations [H-like carbon, to level 45]

Level populations are normally not printed for the atoms and ions of the H-like or He-like isoelectronic sequences. This will print them. If no numbers appear on the line then the first 15 levels will be printed. Enter the highest level to print on the line as an integer if more are desired.

If the keyword **He-like** appears then an element on the helium-like isoelectronic sequence will be printed. Otherwise an element of the H-like isoelectronic sequence is chosen. The code will search for the name of an element, and if it finds one, will print that element and isoelectronic sequence. If none are recognized then populations for H I are printed.

The departure coefficients will be printed if the **print departure coefficients** command (page 124 above) also appears in the input stream.

13.18. print last

Normally, results for every iteration are printed as they are computed. If this command is entered then only results for the last iteration will be printed. Only one of these options is recognized on a single command line.

13.19. print line *options*[29]

A large block of emission line intensities is printed after the calculation is complete. This controls details of that printout.

Some options change the layout of this information. These include options to print a single column, to sort the lines by wavelength or intensity, or to print only the strongest lines, or those within a certain wavelength range.

Other options indicate line formation processes. A great deal of information about line formation and beaming is stored within the code but not normally printed to save space. The section of Part 3 of this document *The Emission Lines* gives more information

---

[29] In versions 87 and before, the code printed some relative line intensities for each zone. An extra line could be added with the **print line** command. This command, and that printout, no longer exists. Use the **punch line intensities** command instead.

13.19.1. print line all

All of the contributions to line formation, including collisions, pumping, and heating, will be printed.

13.19.2. print line collisions

Collisions are usually the dominant contributor to formation of an optically thick line. This adds an entry with the label "Coll" followed by the wavelength and the collisional contribution.

13.19.3. print line column [linear]

The main block of emission lines is normally printed with four lines across the page. This command causes the lines to be printed as a single column to make it easier to enter into a spreadsheet. The command also has a keyword **linear**, which will cause the intensities to be printed as the linear flux in exponential format rather than the log.

13.19.4. print line faint -2 [_off]

Cloudy will normally print the intensities of all emission lines with intensities greater than 10^{-3} of the reference line, which is usually Hβ. This command changes the limit to the relative intensity of the weakest line to be printed. The argument is either the log (if ≤ 0) or the linear (if positive) intensity of the weakest line to print, relative to the reference line. The reference line is usually Hβ, and can be changed with the **normalize** command (page 120 above). In the case shown here, only lines with intensities greater than 1% of Hβ will be printed.

If no numbers are entered, but the keyword **_off** appears, then all lines are printed, even those with zero intensity.

13.19.5. print line flux at Earth

If the distance to an object is set with the **distance** command (page 78 above), and the luminosity of the lines can be predicted (see the discussion on page 21 above) then this will print the observed flux at the Earth rather than the line luminosity. The units are ergs cm^{-2} s^{-1}. (No interstellar extinction is included, of course). Both the keywords **flux** and **Earth** must appear. This command can be combined with the **aperture** command (see page 75 above) when only part of a spatially resolved object is viewed.

13.19.6. print line heat

Fluorescent excitation is included as a line formation process. If a line is radiatively excited then collisionally deexcited it will heat rather than cool the gas. This option prints the heating due to line collisional de-excitation. The entry will have the label "Heat" followed by the wavelength.

13.19.7. print line H2 electronic

By default, only ro-vibrational lines within the ground electronic state of H$_2$ are included in the emission-line printout when the large H$_2$ molecule is included with

the **atom H2** command (page 85 above). This **print line** command tells the code to also print electronic transitions.

13.19.8. print line inward

Optically thick emission lines are not emitted isotropically. The "inward" fraction of the line is the part that is emitted from the illuminated face of the cloud into the direction towards the source of ionizing radiation. This will generally be greater than 50% of the total intensity if the line is optically thick. This command prints this inward fraction with the label "Inwd" followed by the wavelength.

13.19.9. print line optical depths [_off, faint]

Line optical depths are not printed by default. The option tells the code to print them at the end of the iteration. There are two optional keywords.

If **_off** appears then line optical depths will not be printed. This is useful if turned on in a previous iteration and no longer needed.

The keyword **faint** sets the smallest line optical depth to print. The default smallest line optical depth to print is 0.1. The log of the limit must be given. Optical depths for all lines that mase are normally printed.

13.19.10. print line pump

All lines include fluorescent excitation by the attenuated incident continuum as a line formation process. Continuum pumping will often be the dominant formation mechanism for optically-thin high-excitation lines. This option prints an estimate of the contribution to the total line intensity from this process. The entry will have the label "Pump" followed by the wavelength.

13.19.11. print line sort wavelength [range 3500A to 1.2m]

The output spectrum to be sorted by wavelength rather than by ion[30]. It was originally added by Peter G. Martin. If the **range** option appears then two more numbers, the lower and upper bounds to the wavelength range, must also appear. Each number is interpreted as the wavelength in Angstroms by default, but is interpreted as the wavelength in microns or centimeters if the wavelength is immediately followed by a "c" or "m". The two wavelengths must be positive and in increasing wavelength order.

13.19.12. print line sort intensity

The emission line predictions will be sorted in order of decreasing intensity.

13.19.13. print line sum

This option prints the sum of the intensities of an arbitrary set of emission lines. This can be useful for applications such as the Stoy (1933) energy balance method of determining stellar temperatures, which rely on the sum of a set of observed line

---

[30] The **print sort** command existed but did not function between 1986 and 2001. It became functional again with version 96 but was moved to become an option on the **print line** command..

intensities relative to a recombination line (see also Kaler & Jacoby 1991). The sum is printed as the last entry in the emission-line array as an entry with the label "Stoy" and a wavelength of 0.

The set of emission lines to be summed is entered one per input line. This list begins on the line after the **print line sum** command and continues until a line with **end** in the first three columns appears. The line label must be the first four characters on each line and the line wavelength must appear as it does in the printout. The default units of the wavelengths are Angstroms and any other units must be specified. The following gives an example of its use.

```
print line sum
o  3 5007
totl 3727
o  1 6300
O  3  51.80m
S  3  18.67m
s  3 9532
end of lines
```

Up to 30 lines can be entered into the sum.

13.19.14. *print line surface brightness [arcsec]*

By default the line intensities that are printed after the calculation is complete is given as L with units erg s^{-1} for the case where a luminosity is specified, or $4\pi J$ [erg cm^{-2} s^{-1}] in the intensity case. This command will change these intensities into surface brightness units. The default is per sr, but if the keyword **arcsec** appears then the surface brightness will be per square arcsec.

13.20. print off

This turns off the print out, as with the **print quiet** command (page 129 below). This is normally paired with a later **print on** command to avoid printing parts of the output.

There is a possible problem. The code can read its own output as input, to make it easy to rerun a model. In many initialization files the following pair of commands appears:

```
print off
commands ….
print on
```

The resulting output will print the first **print off** command, but will not print the commands or the **print on** command. If this output is used as input, no further output will be created for the new model. This problem will not occur if the **print off** command includes the keyword **hide**, described on page 16 above.

13.21. print on

This command turns on printout. This is the opposite of the **print quiet** or **print off** commands.

13.22. print only [header, zones]

The keyword **only** shortens the printout somewhat by stopping the calculation prematurely. If it appears then another keyword, **header** or **zones**, must also appear. The command **print only header** will cause the code to end after printing the header information. The command **print only zones** will cause the code to return after printing the zone results on the first iteration.

13.23. print quiet

This command sets Cloudy's quiet mode, in which nothing is printed at all. Printing can be turned off and then restarted at a particular zone by using the **print starting at** command described below.

13.24. print short

This shortens the detailed final printout. Only the emission lines and a short summary of some thermal properties of the model will be printed.

13.25. print starting at 61

This option turns off *all* printout *until* the specified zone is reached. This should come last in the input stream since command lines appearing after it will not be printed.

13.26. punch commands

13.26.1. Overview

Punch commands save results into a file that can be used later. This is the primary output mechanism for Cloudy. There are many options. For instance, physical quantities as a function of depth into the cloud, including temperature, ionization, and density, can be saved for later plotting. For other cases the continuum or other quantities predicted by the code can be output. In all cases, the general idea is for the file produced by this command to then be post-processed by other plotting or analysis programs to produce final results.

One of the following keywords must appear, and only one keyword per line is recognized. Up to 100 **punch** commands can be entered.

13.26.2. *An output file name must appear inside double quotes*

Each **punch** command must specify a file name[31] for the resulting output. This file name must appear between a pair of double quotes, like "output.txt". It must be a valid file name for the operating system in use. The following is an example.

```
punch overview "model.ovr"
```

The code will stop if a valid file name is not present.

---

[31] In versions 90 and before Fortran default punch units, with names like fort.9, could be used for punch output. The filename must be specified with versions 91 and later.

A prefix can be set for all filenames with the **set punch prefix** command, described on 192 below.

13.26.3. The "last iteration" option

Each **punch** command also has a keyword **last** that will cause the output to only be produced on the last iteration. It this keyword does not appear then output will be produced for every iteration, with results of each iteration separated by a line of hash marks ("**###**").

13.26.4. The "no buffering" option

If the option **no buffering** appears then file buffering will be turned off for that file. This slows down the output considerably but ensures that all output will exist if the code crashes.

13.26.5. The "no clobber" option - punch output in grids of models

When the code is used as a stand-alone program to compute a single model it will open the punch file at the start of the calculation and close it at the end. In a grid of models this will happen for each new model and so will overwrite results of all previous calculations.

The **no_clobber** keyword on the **punch** command will produce one long file containing results of consecutive models. It tells the code to never close the file at the end of any but the last calculation and not try to reopen this file once it is open.

13.26.6. The "no hash" option

If results from more than one iteration are punched, then each iteration's results end with a series of hash marks, "**###**", to make the iteration easy to find in an editor. These hash marks can cause problems if the file is then read in by other programs. The hash marks will not be produced if the **no hash** keyword appears,.

The character string that is printed between iterations can be changed with the **set punch hash** command, described on 192 below.

13.26.7. The "title" option[32]

The title of the model and the version number of the code will be printed on the first line of the punch file.

13.26.8. Depth versus radius

The code and this documentation make a consistent distinction between depth and radius. This is described on page 8 above. The *radius* is the distance from a point in the nebula to the center of symmetry, generally the center of the central object. The *depth* is the distance from a point in the nebula to the illuminated face of the cloud. In both cases the distance is to the center of the last zone.

---

[32] The title was printed by default in versions 95 and before of the code. The title was generally deleted so that the punch file could be used to make plots, so it is now missing by default.

The output from each punch command is described in the following sections. In those cases where quantities are given as a function of position into the cloud, the first column will usually give the depth, not the radius. You need to add the inner radius of the cloud to the depth to get the radius.

13.27. punch abundances

The log of the gas-phase densities [cm^{-3}] of the elements will be punched for each zone. This is the sum of the abundances of a chemical element in atoms, ions, and molecules, but does not include grains. This provides a check for the effects of the **element table** (see page 59 above) and **fluctuations abundances** (see page 59 above) commands.

13.28. punch ages

The timescales for several physical processes will be punched as a function of depth.

13.29. punch agn [options]

This produces output files that were used to create data tables in the 2rd edition of *Astrophysics of Gaseous Nebulae*, referred to as AGN3 here. The options are the following: **charge** transfer, **recombination** coefficients, **recc** for hydrogen recombination cooling, **opacity**, **hemis**, and **hecs** (for He^0 collision strengths).

13.30. punch asserts

The **assert** command is described on page 196 below and provides an automated way to validate the predictions of the code. Normally the results from these checks will be printed on the standard output. If this command appears then the same output will also be sent to a file.

13.31. punch column density

The logs of the column densities [cm^{-2}] of the gas constituents are punched.

The punch FeII column density command is described on page 139 below. The punch H2 column density command is described on page 141 below.

13.32. punch continuum

This command has been the primary mechanism for saving the spectrum predicted by the code. All continua are given as the flux per octave νf_ν [erg cm^{-2} s^{-1}]. They are relative to the inner radius of the cloud so the monochromatic luminosity per octave is the predicted quantity multiplied by $4\pi\, r_o^2$.

This command is being superseded by a new command, **punch spectrum**, described on page 137 below. That command is still under construction at this time.

13.32.1. Emission line – continuum contrast

Emission lines are included in the output for all **punch continuum** commands except **punch transmitted continuum**. In nature the line to continuum contrast

depends on the intrinsic width of the line. By default the lines are added to the continuum assuming that they have an intrinsic width of 1000 km s^{-1}. This can have the effect of making lines appear to be either too strong or too weak relative to the continuum, depending on the actual line width. The assumption also changes the summed total intensity of entries in the punch output file. Other line widths can be set with the **set PunchLWidth** command, described on page 192 below. This issue is discussed further in the section *Observed Quantities* in Part 3 of this document.

13.32.2. Pumped contributions to the lines

Continuum pumping and fluorescence are included as excitation processes for all lines. The contribution is not usually printed as a separate quantity, but will be if the **print line pump** command (page 127 above) is entered. Whether or not the pumped contribution actually adds to the observed line emission depends on the geometry. Continuum pumping increases the line emission if no related continuum absorption is seen by the observer. This will be the case if the continuum source is either not observed or not covered by absorbing gas on the observer's line of sight. If absorbing gas covers an observed continuum source then the situation is like the P Cygni problem, emission is the result of absorption, and pumping may not increase the total intensity of the line at all.

The printed line intensity includes a contribution from fluorescent excitation unless the **no induced processes** command is entered. That command is unphysical since it turns off many induced processes. You can judge how great the contribution of the pumped part of the line is by printing it with the **print line pump** command (described on page 127 above).

In general the treatment of scattering is very geometry dependent. The output produced by the **punch continuum** commands *does not* include the pumped part of the line contribution. This is correct if the continuum source is included in the beam, but is not if only the gas is observed.

13.32.3. Energy units for the punch output

By default, the energy units for the first column, which gives the wavelength or energy for each point in the continuum, are Rydbergs. The units can be changed to any of several energy or wavelength units with the **units** keyword that appears on a **punch continuum** command. The following keywords are recognized: **microns, _keV, _eV_, Angstroms, centimeters** (also **_cm_**), **wavenumbers, MHz**, and **Rydbergs**. Both the keyword **units** and one of these units must appear.

13.32.4. Punch continuum predictions

The **punch continuum** command, with no other keywords, produces a file with the following information. The intensity units for the continuum are the intensity per octave $4\pi \, \nu J_\nu$ [erg cm^{-2} s^{-1}].

Column 1. The first column gives the photon energy in the units set with the **units** option (described on page 132 above). The default units are Rydbergs.

Column 2. This is the incident continuum at the illuminated face of the cloud.

Column 3. This is the transmitted (attenuated) incident continuum and does not include diffuse emission from the cloud.

Column 4. This is the outward portion of the emitted diffuse thermal continuum and line emission. Only the diffuse emission includes a covering factor if one was specified. The total emission from the nebula is this multiplied by the inner area of the cloud. This does not include the attenuated or reflected portions of the incident continuum.

Column 5. This gives the net transmitted continuum, the sum of the attenuated incident (column 3) and diffuse (column 4) continua and lines. This would be the observed continuum if the geometry were viewed through the gas, and includes the covering factor.

Column 6. This is the reflected continuum and is only predicted for an open geometry.

Column 7. This is the sum of the transmitted and reflected continua and lines. The attenuated incident continuum is included.

Column 8 and 9. Line and continuum labels indicate the lines and continuum edges that contribute at that energy. If than one line is included in a continuum cell then the label gives the first one placed there, not the strongest line.

Column 10. This gives the number of emission lines within that continuum bin, divided by the ratio of the energy width of the cell to the cell's central energy, *dE/E*. This is the number of lines per unit relative energy.

13.32.5. *What is observed*

Figure 9 illustrates several possible geometries. Two lines of sight to the central object are shown, and two clouds are shown. Each cloud produces both a reflected and transmitted component of emission.

Figure 9 This figure illustrates several of the continua that enter in the calculations. ObsContin

Three possible geometries, indicated by the letter on the figure, occur for the continuum source: a) we do not directly observe it although we may see it by reflection from a cloud, b) we observe the attenuated continuum transmitted through the emission line region, and c) we observe the unattenuated continuum directly without absorption. Column 2 gives the unattenuated continuum, and column 3 given the attenuated continuum.

There are also three possible situations for the line emission. First, we might only observe clouds that lie on the near side of the continuum source. In this case we see the "outward" line emission. Second, we might only observe clouds that lie on the far side of the continuum source. In this case we only see the "reflected" component.

Lastly, we might observe a symmetric geometry with reflected emission from the far side and outward emission from the near side.

In most cases an observer at large distance from the structure would observe *both* the central object and the nebula and would measure the quantity listed in column 5 (if only transmitted emission is detected) or column 7 (if both reflected and transmitted continua are seen). If the central object is not seen then the quantity in column 4 would be observed.

13.32.6. *punch continuum bins*

This punches the continuum energy bins. The first two numbers are the frequencies of the center of the bin ν. The first column is the center of the bin defined in cells with log increasing with. The second column, which is nearly equal to the first, is the center of the cell, weighted by the continuum shape. This has a slight dependence on the continuum shape. The last number is the cell width $\delta\nu$. The bin extends from $\nu - \delta\nu/2$ to $\nu + \delta\nu/2$.

13.32.7. *punch diffuse continuum*

This punches the locally produced diffuse emission from the gas for the last zone. Optical depth effects are not included and the continuum is the local continuum for the last computed zone. The output gives the photon energy and diffuse continuum [νf_ν erg cm^{-3} s^{-1}].

13.32.8. *punch emitted continuum*

The continuum emitted and reflected from the nebula is punched. The first column is the photon energy. The second column is the reflected spectrum. The third column is the outward diffuse emission. The fourth column is the total emission (the sum of the inward and outward emission). This would be the observed emission from the nebula if the central continuum source were not in the beam but clouds uniformly cover the continuum souce. The last two columns are labels for lines and continua contributing at each energy. The attenuated incident continuum is not included in any of these components. All continua have units νf_ν [erg cm^{-2} s^{-1}] at the inner radius.

13.32.9. *punch fine continuum [range, skip]*

The code transfers the continuum on a coarse mesh, needed for speed in evaluating photo-interaction rates, and on a fine mesh, needed for automatic treatment of line overlap. This multigrid approach is needed to combine precision and speed. This command will output the transmission factor, $1 - \exp(-\tau_\nu)$, for the fine continuum.

If the entire fine continuum is punched the resulting output will be huge. The command accepts a **range** option to make the output file smaller. If the keyword **range** appears then the lower and upper limits to the range of the fine continuum must be entered in Rydbergs. The command also accepts a **units** option to change the units used in the resulting output. (The **range** option does not, at present, recognize the **units** option – the range must be in Rydbergs). The optional third numerical parameter gives the number of fine continuum cells to combine together,

again with the intent of reducing the size of the output file. This can make the output file smaller. The default is to average over 10 cells. If the number of cells to be combined is specified then it must be the third number on the command line, following the lower and upper limit of the range.

13.32.10. *punch grain continuum*

The thermal emission from all grain species included in the calculation is part of the predicted emergent continuum. This command punches only this grain emission. The first column gives the photon energy. The next gives the total emission from graphite. The last column gives all emission from all constituents that are not graphite. In practice this will be mainly the silicates.

The second letter of the grain label determines the separation into graphite and silicates. By default this is either "g" or "s" indicating graphite or silicate. The logic is to put emission into the graphite array if this letter is a "g", and into the silicate array if it is anything else. As a result the two columns will always add up to the total grain emission.

13.32.11. *punch incident continuum*

The incident continuum, that emitted by the central object and striking the illuminated face of the cloud, will be punched. There will be two columns, the photon energy and the continuum with units νf_ν [erg cm^{-2} s^{-1}].

13.32.12. *punch interactive continuum*

This will punch the integral of the product of the internal radiation field times the gas opacity. The results are produced for each zone and give the attenuated incident continuum, the OTS line, the OTS continuum, the outward continuum and the outward lines. The first optional number is the lowest energy to consider in the resulting output. If missing or zero, the lowest energy considered by the code will be used. If the number is less than 100 then it is interpreted as the energy in Rydbergs, and if greater than 100, the cell number.

13.32.13. *punch ionizing continuum [options]*

This command is used to punch the ionizing continuum. If the keyword **every** occurs then this continuum is punched for every zone, otherwise it is only punched for the last zone, at the end of the calculation. The first column is the photon energy. The second is the total number of photons within this frequency bin (*not* per unit frequency). The third column gives this flux multiplied by the gas opacity [s^{-1} cell^{-1}], and is basically a radiation field interaction rate. The next four numbers are the fractions of the total radiation field at that energy due to the attenuated incident continuum, the OTS line, the continuum radiation fields, and the outward only continuum. The 8th number is the ratio of this quantity to the total integrated radiation field interaction rate. The last number is the integrated cumulative interaction. This makes it easy to identify the portions of the radiation field that have the dominant interaction with the gas. The last two labels on the line indicate which lines and continua contribute at that energy.

The first optional number on the command line is the lowest energy to consider in the resulting output. If this is missing or zero then the lowest energy considered by the code will be used. If the number is less than 100 then it is interpreted as the energy in Rydbergs and if greater than 100, as the cell number. The second optional number is the threshold for the faintest interaction to print, with a default of one percent in the units used in the 8th column. Enter zero for this number if you want all interactions to be printed. The optional numbers may be omitted from right to left.

13.32.14. *punch outward continuum*

The photon energy is followed by the attenuated incident continuum, the outwardly directed diffuse continuum, the outward lines, and the sum of the two. If the `local` keyword also appears then only the outward continuum produced in the last computed zone will be punched.

13.32.15. *punch raw continuum*

This punches the "raw" continua. This is exactly the continuum used within the code. The first number is the photon energy. The next columns are the contents of the arrays *flux, otslin, otscon, refcon, outcon, outlin,* and *condif* at this energy. Each gives the number of photons stored in that cell with units photons s^{-1} cm^{-2} $cell^{-1}$. The last number is the number of lines within that cell.

13.32.16. *punch reflected continuum*

The reflected continuum is output if `sphere` is not set. The first column is the photon energy, the second the reflected continuum at that energy $4\pi\nu J_\nu$ [erg cm^{-2} s^{-1}]. The third gives the reflected lines and the fourth is the sum of these two. Someone who could only see the illuminated face of the cloud would observe this. The next column is the effective albedo of the cloud, the ratio of the reflected to incident continuum (the `punch total opacity` command, described on page 150 below, will give the true albedo). The last column gives the label for continuum processes with thresholds at the energy.

13.32.17. *punch transmitted continuum*

This saves the transmitted (attenuated incident and outward component of diffuse) continuum predicted at the end of the calculation.

This punch file can be used as part of the incident continuum in a later calculation, by reading in this file with the `table read` command (see page 42 above). Three cautions apply when reading this file as an input continuum. First, if the keyword `last` does not appear on this command then the continuum from each iteration will be punched and the code will become confused when it tries to read this file. You probably only want results from the last iteration so you should include the `last` option on the `punch continuum` command. Second, punch output should not be created on the same file name as the input file during the second calculation. The file containing the continuum will be overwritten if this occurs. Finally, Cloudy expects the first two lines to contain header information and skips them. They should not be deleted from the input file.

The line-to-continuum contrast factor *PunchLWidth* (see page 192 below) is not used in this command. This is to insure that lines have the correct intensity in the punch file, as needed for energy conservation. This has the effect of setting the line width to the speed of light.

13.32.18. *punch two photon continuum*

This punches the total two-photon continuum and is mainly a debugging aid. The photon energy is followed by the number of photons emitted per Rydberg per second, then by νF_ν.

13.33. punch spectrum

This command is being developed to replace the **punch continuum** command described above. It works together with the **set spectrum** command (see page 193 below), to form a type of punch output that can be modified to suite a particular purpose. The **set spectrum** commands occur before the **punch spectrum** command, and modify the behavior of that punch command.

The units of the continuum are changed with the **units** option described on page 132 above.

This command is still under development.

13.34. punch convergence [reason, error]

These commands produce information about various aspects of the converged solution.

13.34.1. *Punch convergence reason*

This will punch the reason the model was declared "not converged" at the end of each iteration when the **iterate to convergence** command (page 99 above) is used.

13.34.2. *Punch convergence error*

This will produce information concerning the quality of the converged pressure, electron density, and heating-cooling solution. The correct value, converged value, and percentage error, (correct-converged)×100/correct, will be produced for each zone.

13.35. punch cooling

This punches the cooling agents for each zone. The first number is the depth [cm]. The next two numbers are the total heating and cooling rates [erg cm^{-3} s^{-1}]. The following numbers are labels for contributors to the cooling array and the fraction of the total cooling carried by that agent. The faintest coolant punched is normally 0.05 of the total, and can be reset with the **set WeakHeatCool** command (page 196 below).

The line labels may, or may not, correspond to labels used in the main emission-line output. When in question, search over the entire code for the printed string.

13.36. punch charge transfer

Charge transfer recombination and ionization rate coefficients [cm^3 s^{-1}] for hydrogen onto heavier elements will be output. The rates will be evaluated at the temperature of the last computed zone. Rates for recombination $(A^{+x} + H \rightarrow A^{+x+1} + H^+)$ are first, followed by the rates for the opposite ionization process. The first number is the atomic number of the species.

13.37. punch _dr_

The logic behind the choice of zone thickness will be described.

13.38. punch dynamics options

This produces some information concerning the dynamics solutions.

13.38.1. *punch dynamics advection*

This produces information about advection terms.

13.39. punch element *name*

This will punch the ionization structure of any element. The output will have one line per zone giving the ion fraction[33] of each successive stage of ionization. The keyword **element** must be followed by the element name spelled with the first four characters exactly as given in Table 15 (page 50 above).

The first number on the resulting output is the depth [cm] into the cloud. The remaining lines are the relative ionization fraction of the *n+1* possible stages of ionization, where *n* is the atomic number of the element. This may be followed by some of the more abundance molecules.

If the keyword **density** appears on the command line then the density [cm^{-3}] will be given rather than the fraction.

13.40. punch enthalpy

The file will list the depth into the cloud, followed by the total enthalpy, and various contributors to it.

13.41. punch FeII [populations, departure, _all]

This produces some information about the Fe II atom. The atom is slow and is not done by default. It is included with the **atom FeII** command (page 83 above).

Both keywords "**FeII**" with no spaces, and "**Fe_II**" with a space, are accepted. The following examples show the first style, the second is formally correct.

---

[33] Before version 96 the ionization fractions only included atoms and ions. They now include molecules. The sum of the atomic and ionic fractions will not add up to unity if a significant fraction of the element is in molecules.

13.41.1. punch FeII column densities

This gives the excitation energy of the level (wavenumbers), the level's statistical weight, and the column density (cm^{-2}).

13.41.2. punch FeII continuum

The pseudo-continuum of Fe II lines predicted by the large Fe II atom will be punched. See page 85 above for more details. In the resulting output the first number is the wavelength of the center of the band in Angstroms. The second number is the integrated intensity of Fe II emission over that band in intensity units [erg cm^{-2} s^{-1}].

13.41.3. punch FeII departure

The departure coefficients for selected levels of the large Fe II atom will be punched. This is normally only listed for selected levels. All levels will be punched if the keyword **_all** appears.

13.41.4. punch FeII levels

A list of the energy levels in the large Fe II atom will be produced. The first column gives the energy of the level (wavenumbers) above ground, and the second column gives the statistical weight of the level.

13.41.5. punch FeII lines [faint=0.1, range 0.1 to 0.3 Ryd]

This will punch all ~10^5 lines predicted by the large FeII atom at the end of the calculation. The upper and lower level indices are printed in the output file, followed by the log of the intensity or luminosity of the corresponding Fe II line. This is followed by the linear intensity relative to the line set with the **normalize** command, and the optical depth of the transition is given in the last column. If the keyword **short** appears then the relative intensity and the optical depth are not punched.

Three optional numbers can appear on the command line. The first is the intensity of the faintest line to be punched, relative to the normalization line, usually Hβ. The second and third optional numbers are the lower and upper limits to the range of punched lines' energy, in Rydbergs. Both numbers are interpreted as logs if either is negative. These optional numbers can be omitted from right to left.

13.41.6. punch FeII populations

The level populations [cm^{-3}] for selected levels of the large Fe II atom are punched by default. All levels will be punched if the keyword **_all** appears.

13.42. punch gammas

This punches the photoionization rates for all subshells of all ions for the last computed zone. The numbers are the element, ion, and subshell numbers, followed by the photoionization and heating rates from that subshell. The remaining numbers are the fractional electron Auger yields.

13.42.1. punch gammas element oxygen 1

If the **element** keyword appears then the detailed contributors to the photoionization rate for the valence shell of a particular element will be produced by calling *GammaPrt*. The ionization stage must also appear, with 1 the atom, 2 the first ion, etc.

13.43. punch gaunt factors

This produces a table showing the free-free gaunt factors as a function of photon energy and temperature.

13.44. punch grains [options]

These commands show grain properties. There are usually many grain species included in a calculation. Often there are several size bins per grain type. These commands will print a line giving a list of the grain labels, followed by a line giving the grain radius in μm. The following lines then give the individual grain properties (temperature, potential, etc) for each size and type. 140 above

13.44.1. punch grain charge

The charge of each grain species, in number of electrons per grain, is printed for each zone.

13.44.2. punch grain continuum

See page 135 above.

13.44.3. punch grain drift velocity

The drift velocity [km s^{-1}] of each grain species is printed for each zone.

13.44.4. punch grain extinction

The grain extinction at the *V* filter will be punched as a function of depth. This extinction only includes grains, which is nearly the total when grains are present. The first column gives the depth into the cloud [cm], the second is the extinction [mag] at the *V* filter for an extended source, like a PDR, and the third number is the extinction for a point source like a star. The extended source extinction discounts forward scattering by writing the scattering opacity as $\alpha(1-g)$, where α is the total scattering opacity and g is the grain asymmetry factor. The quantity in the last column does not include the $(1-g)$ term.

13.44.5. punch grain H2rate

The grain H_2 formation rate is output for each zone.

13.44.6. punch grain heating

The grain heating [erg cm^{-3} s^{-1}] is output for each zone.

13.44.7. punch grain opacity

The output will list the grain opacity as a function of the photon energy. The first column is the photon energy, the second the total (absorption plus scattering) cross

section, followed by the absorption and scattering cross sections. These are the summed cross section per proton for all grain species in the calculation.

13.44.8. *punch grain potential*

The grain floating potential (eV) is output for each zone.

13.44.9. *punch grain _qs_*

The photon energy is followed by the absorption and scattering *Qs* for each grain species.

13.44.10. *punch grain temperature*

The temperature of each grain species is printed for each zone.

13.45. punch heating

The code will punch the heating agents for each zone. The depth [cm] is first, followed by the total heating and cooling rates [cm^{-3} s^{-1}]. This is followed by a set of labels for various heat sources and the fraction of the total heating carried by that agent. The faintest agent punched is normally 0.05 of the total, and can be reset with the **set WeakHeatCool** command (page 196 below).

The heating labels will probably not correspond to any entries in the emission-line list. If the identification of a heat source is in question the best recourse is to search for the heating label over the entire code.

13.46. punch xxx history

13.46.1. *punch pressure history*

This follows the pressure and density convergence history.

13.46.2. *punch temperature history*

This follows the temperature and heating – cooling convergence history.

13.47. punch H2

Some details of the large H_2 molecule are punched for each zone. One of the following options must appear. The large H_2 atom is not included by default because it is fairly slow. It is turned on with the **atom H2** command, described on page 85 above.

13.47.1. *punch H2 column density*

This punches the column density of ro-vibrational states within the ground electronic state. This command recognizes the same options as the **punch H2 populations** command.

The file begins with the total H_2 column densities in the ortho and para forms, followed by the total H_2 column density. The remainder of the file gives the *v* and *J* quantum indices, followed by the excitation energy of the level in K, the total column density in that level, and finally the column density divided by the statistical weight of the level.

13.47.2. punch H2 cooling

This produces a file containing heating and cooling rates [cm^{-3} s^{-1}] as a function of depth.

13.47.3. punch H2 creation

H$_2$ creation processes are given for each zone.

13.47.4. punch H2 destruction

H$_2$ destruction processes are given for each zone.

13.47.5. punch H2 heating

This produces a file containing the depth, total heating rate [erg cm^{-3} s^{-1}], and H$_2$ heating predicted by an expression in Tielens & Hollenbach (1985a), together with the heating predicted by the large model of the H$_2$ molecule.

13.47.6. punch H2 levels

This creates an energy-ordered list of the levels within X. The first column gives the energy in wavenumbers relative to the lowest level, the second gives the statistical weight, and the last two columns give the vibration and rotation quantum numbers.

13.47.7. punch H2 lines

This produces a file containing the intensities of all significant H$_2$ emission lines that are produced within the ground electronic state. Electronic lines can also be included.

Each line begins with a spectroscopic designation of the line, followed by the upper and lower electronic, vibration, and rotation quantum numbers. This is followed by the wavelength of the line. The line wavelengths are given in air microns. The line wavelength is then printed as it appears in the output. The log of the intensity or luminosity in the line (ergs per unit area or full shell, depending how the continuum was specified), and the intensity of the line relative to the normalization line, follow.

Only lines brighter than 10^{-4} of the reference line are printed by default. The intensity of the faintest line, relative to the normalization line, is set with the first optional number that can appear on the command line. If the number is negative then it is interpreted as the log of the limit.

By default only lines within the ground electronic system are punched, but this can be changed if the keyword **electronic** appears on the command line. If the keyword **_all** also appears then lines within all electronic systems are punched. If the keyword **ground** appears then only lines the ground system are punched. This is the default. If a number appears then it is the number of electronic systems to be punched, 1 for only the ground electronic state. If the number of systems is specified then it must be the second number on the line – the first being the intensity of the faintest line to print.

The following give examples. The first sets the faintest H_2 line to punch. Only lines within the ground electronic system will be produced.

```
punch H2 lines, faintest -4 "filename"
```

This sets the faintest line, and also requests all lines produced by all electronic configurations.

```
punch H2 lines, faintest -4 all "filename"
```

This example requests only lines for the lowest three electronic configurations.

```
punch H2 lines, faintest -4 electronic 3 "filename"
```

13.47.8. *punch H2 populations*

The level populations for the ground electronic state will be punched for the last computed zone. The populations are relative to the total H_2 abundance.

There are several optional parameters. The quantum numbers of the highest vibrational and rotational levels to punch can be specified as consecutive numbers on the command line. These occur in the order *v* then *J*. If no numbers occur, or if a limit that is less then or equal to zero is entered, then all levels populations will be punched.

If the keyword **zone** appears then the populations of the *v*=0 levels will be punched for every zone. Otherwise the populations are only punched at the end of the iteration.

I output is only punched for the last zone then it can be in either a triplet format, with the vibration and rotation quantum numbers followed by the population, or as a matrix, with all populations of a given vibration quantum number lying along a single row. The triplet form is done by default, and the second will occur if the keyword **matrix** occurs on the command line.

13.47.9. *punch H2 PDR*

The output contains useful information regarding conditions within a PDR.

13.47.10. *punch H2 rates*

The output contains useful information regarding H_2 formation and destruction rates.

13.47.11. *punch H2 Solomon*

The output will give the total rate for photo-excitation from the ground to electronic excited states, and then identify those levels which are the dominant contributor to the total rate.

13.47.12. *punch H2 special*

This is intended to provide the infrastructure to punch some debugging information that can be easily changed to suite particular needs.

13.47.13. punch H2 temperatures

The depth, 21 cm spin temperature, gas kinetic temperature, and several temperatures derived from relative populations of *J* levels within the H_2 *v*=0 ground electronic state, are punched for each zone.

13.47.14. punch H2 thermal

A variety of heating – cooling processes involving H_2 are given for each zone.

13.48. punch hydrogen

13.48.1. punch hydrogen 21_cm

This gives some information related to the spin temperature of the 21 cm line. The keyword is either "21 cm" or "21cm". The level populations within 1*s* are determined including radiative excitation by Lα, pumping by the external and diffuse continua, collisions, and radiative decay. Several of the resulting populations and temperatures are included in the output.

13.48.2. punch hydrogen conditions

This gives the physical conditions and hydrogen constituents as a function of depth. The densities of H^0, H^+, H_2, H_2^+, H_3^+, and H^- relative to the total hydrogen density, follow the depth, temperature, hydrogen density, and electron density.

13.48.3. punch hydrogen ionization

This gives rates for processes affecting the hydrogen ionization as a function of depth. The columns are the ground state photoionization rate, the total and Case B recombination coefficients, the predicted ratio of H^+ to H^0, and the theoretical ratio for the simple case. Finally contributors to the ground state photoionization rate are produced with a call to *GammaPrt*.

13.48.4. punch hydrogen _Lya

This produces some information related to the excitation, destruction, and escape, of the Lα line of hydrogen. The optical depth from the illuminated face to the outer edge of the current zone is followed by the excitation temperature, electron kinetic temperature, and the ratio of these.

13.48.5. punch hydrogen populations

This gives the depth [cm], the densities of H^0 and H^+ [cm^{-3}], followed by the level population densities [cm^{-3}] for the levels in the order *n* = 1, 2*s*, 2*p*, 3, 4, 5, … for each zone.

13.48.6. punch hydrogen lines

The upper and lower quantum indices, the line energy, and the optical depth in the line, will be punched.

13.49. punch ionization means

The mean[34] ionization of all elements included in the calculation will be output. The format is exactly the same as the mean ionization printout produced at the end of the standard output.

13.50. punch ionization rates carbon

The total ionization and recombination rates for a specified element will be punched as a function of depth. The name of an element must appear on the command line. Each line of output will have the depth [cm], electron density [cm^{-3}], and the sink timescale for loss of particles due to advection out of the region [s^{-1}]. The remaining numbers give quantities for each possible stage of ionization of the element. For each ionization stage the set of numbers that are printed give the density atoms in that ionization stage [cm^{-3}], the total ionization rate [s^{-1}], the total recombination rate [s^{-1}], and rate new atoms are advected into the region [s^{-1}].

13.51. punch _ip_

This gives the ionization potentials of all shells of all ions and atoms of the 30 elements included in the code. The first row is the spectroscopic designation of the ion. Each additional row gives the subshell and ionization potential of that subshell in eV. The actual key is **IP** with a space to either side.

13.52. punch Leiden

This command produces an output file designed for the comparison calculations presented in the 2004 Leiden meeting on PDR calculations.

13.53. punch lines, options

This set of commands will punch some details about line formation.

13.53.1. *punch lines, array*

This gives the intensities of *all* lines in a form in which they can easily be plotted by other software. Column one lists the line energy in Rydbergs. Column two is the log of the integrated intensity or luminosity of the line. Only lines with non-zero intensity are included. The last field on the line gives the spectroscopic designation of the ion. All lines that appear in the printout will also appear in the resulting output file[35].

The energy units for the line energies are Rydbergs by default. This command recognizes the **units** option, described on page 132 above.

---

[34] Before version 96 the ionization fractions only included atoms and ions. They now also include molecules. The sum of the atomic and ionic fractions will not add up to unity if a significant fraction of the element is in molecular form.

[35] In versions 90 and before of the code, only the level 1 and level 2 lines were output by this command.

13.53.2. *punch lines, cumulative*

This option tells the code to punch the log of the cumulative intensity of up to 100 emission lines as a function of depth into the cloud. The emission lines are specified on the following input lines, and end with a line with the keyword **end** in columns 1-3. The label used by Cloudy to identify each line in the main emission-line output must appear in column 1-4 of the line, and the line wavelength appears as a free-format number in later columns. The line labels and wavelengths are followed by the depth into the cloud, the distance from the illuminated face to the center of the current zone, and the integrated intensities of the lines [erg cm^{-2} s^{-1}], for each zone. This information can be used to follow the build up of emission lines across a computed structure.

The following illustrates its use;

```
punch lines, cumulative, "lines.cum"
totl 4861
12CO  2589m
13CO  2475m
o  3 5007
totl 3727
o  1 6300
end of lines
```

If the optional keyword **relative** is specified then the quantities will given be relative to the normalization line. If not specified, then the intensity $4\pi J$ [erg cm^{-2} s^{-1}] will be punched.

The **punch lines cumulative** and **punch lines structure** commands use the same line array so both commands cannot be used in the same calculation.

13.53.3. *punch lines, data*

This punchs some atomic data for all lines included in the line transfer arrays. It can be used to generate a table listing many lines and their atomic parameters. The code will stop after the data have been punched since it is left in a disturbed state.

The first set of lines consists of recombination lines from Nussbaumer and Storey (1984) and Péquignot, Petitjean, & Boisson (1991). For these the spectroscopic designation and wavelength are given, followed by the log of the recombination rate coefficient.

The remaining sets of lines are those that are treated with full radiative transfer. The first are the "level 1" lines, those with accurate atomic collision data and wavelengths. The next "level 2" lines have many more lines and uses Opacity Project wavelengths and various g-bar approximations to generate approximate collision strengths. These are followed by the hydrogen and helium iso-electronic sequences, then the ^{12}CO and ^{13}CO lines. The H_2 and Fe II lines come last if these large atoms are included.

By default the atomic parameters will be evaluated at a temperature of 10^4 K. Other temperatures can be selected by entering a **constant temperature** command (page 103 above). The number of H-like, He-like, CO, H_2, and Fe II lines that are printed is controlled by the size of the relevant atoms when the **punch line data** command is executed.

13.53.4. punch lines, emissivity

This punches the emissivity of up to 100 emission lines as a function of depth into the cloud. This structure information can then be used by other codes to reconstruct the surface brightness distribution of a resolved emission-line object. The "emissivity" is the net emission $4\pi\bar{J} = n_u A_{ul} P_{ul} h\nu$ [erg cm^{-3} s^{-1}] produced at a point and escaping the cloud. This includes the escape probability P_{ul}.

The emission lines are specified on the input lines that follow the command and end with a line with the keyword **end** in columns 1-3. The label used by Cloudy to identify each line (use the **punch lines labels** command or look at the main emission-line intensity printout to obtain a list of lines) must appear in column 1-4 of the line and the line wavelength appears as a free-format number in later columns. The easiest way to obtain this information is to copy and paste the line identification and wavelength from a Cloudy output.

The punch output begins with emission line labels and wavelengths. The remaining output gives the emission structure. The first column is the depth [cm] into the cloud. The remaining columns give the volume emissivity [erg cm^{-3} s^{-1}] for each line. The intensity is for a fully filled volume so the punched intensity should be multiplied by the filling factor to compare with observations of a clumpy medium.

The following illustrates its use;

```
punch lines, emissivity, "lines.str"
totl 4861
o   3 5007
totl 3727
o   1 6300
12CO  2589m
13CO  2475m
end of lines
```

The **punch lines cumulative, punch lines emissivity,** and **punch lines structure** commands use the same line array, so these commands cannot be used in the same calculation.

13.53.5. punch lines, intensity [every 5 zones]

This option tells the code to punch the intensities of all lines with intensities greater than zero, in the format used for the final printout (line label, wavelength, intensity). The default is for this to be done only after the last zone is computed. Intermediate results can be punched if the additional keyword **every** appears. In this case the first number on the line is the interval between zones to punch, as in the **print every** command.

The resulting punch output will have the line information spread over 6 columns. For some data base applications it would be better to have a single column of results. If the keyword **column** appears then a single column is produced. If no keyword occurs then the wide format is produced.

13.53.6. *punch line labels*

This creates a file listing all emission-line labels and wavelengths in the same format as they appear in the main output's emission-line list. This is a useful way to obtain a list of emission lines to use when asking the code to find specific lines. The file is tab-delimited, with the first column giving the line's index within the large stack of emission lines, the second being the character string that identifies the line in the output, and the last being the line's wavelength in any of several units.

There is a vast number of emission lines predicted by the code, and many lines will have the same wavelength. The line label can usually be used to distinguish between various lines with the same wavelength. This is seldom the case for contributions to the line however. For instance, the label "inwd" gives the inward contribution to all lines, so there are almost certainly many lines with this label and the same wavelength. The line index can be used to resolve this degeneracy in cases where you want to obtain a line's intensity with a call to a routine. Routine *cdLine_ip* (described in the header file *cddrive.h* and also in Part 3 of this document) uses the line index to find the relative intensity and luminosity of a particular line. But note that this index is not necessarily the same in different calculations – it will always be the same for a particular set of input conditions, but it depends on the sizes of various atoms and which chemical elements are used in the calculation.

13.53.7. *punch line optical depths, limit=-2*

This gives the total optical depths for all lines. By default all lines with optical depths greater than 0.1 will be included. The lower limit can be reset with the optional number than appears on the line – it is the log of the smallest optical depth to be printed. This command recognizes the **units** option, described on page 132 above, so the line wavelengths will be given in any of the wavelength or energy units described.

The line identification, element and ion, starts the output line, in the form used in the usual emission line printout. Next follows the line's wavelength or energy. If the **units** option was specified then the line's energy will be given in the appropriate units. If it was not specified then the wavelength will be given in the form found in the standard output. Finally the line's optical depth and damping constant are printed.

Atoms and ions of the H-like and He-like isoelectronic sequences use complete multi-level model atoms. The number of levels included is limited mainly by processor speed and available memory. Higher Lyman lines (used here to mean permitted lines that connect directly to ground) have little impact on the emission, since they scatter and are degraded into Balmer lines and Lα. However, an absorption spectrum will show them as a series of lines converging onto the continuum from the ground state. The code includes a large number of "extra" Lyman lines, included as absorbers with optical depths output with this command, but not treated as part of the multi-level atoms. The default number of higher Lyman lines is 100, and this can be changed with the **atom H-like Lyman** (page 88 above) or **atom he-like Lyman** commands (page 92 above).

13.53.8. *punch line RT*

This produces a file containing information concerning line radiative transfer. The code will read a series of lines that specify which lines to produce. The line label is in columns one through four and is followed by the line wavelength.

13.53.9. *punch line populations, limit=-2*

This will output some information concerning the atomic parameters and level populations for all lines that are transferred. By default all transitions with upper level densities greater than zero will be included. The lower limit to the density threshold can be reset with the optional number than can appear on the line – this is the log of the smallest population density [cm^{-3}] to be printed.

The first block of information gives an index to identify each line. This is followed by a number with the format "26.02", where the leading part is the atomic number of the species, and the fractional part indicates the ionization stage. An atom is zero, first ion is .01, etc. Molecules are identified by their chemical species, as in CO. The lower and upper statistical weights are next, followed by the energy of the line in wavenumbers and the *gf* value for the transition.

The population densities for each zone follow this block of information. Each line begins with the index used for that line in the atomic parameter list. This is followed by the populations of the lower and upper level of the transition [cm^{-3}].

13.53.10. *punch lines, structure*

This is another name for the **punch lines, emissivity** command described on page 147 above.

13.54. punch Lyman alpha

This command was deleted in version 05.05. Use the **punch hydrogen lya** command (see page 144 above) instead.

13.55. punch map, zone 3 [range 3,999 to 4500]

This produces a map of the heating and cooling rates as a function of temperature. The details of the map are described in the description of the **map** command (page 108 above). The first number is the zone for the map, zero if only a map of the first zone

The optional keyword **range** specifies the temperature range of the map. If this option is specified then the zone number must come first, followed by the lower and upper temperature limits to the map. Both temperatures will be interpreted as logs if the first number is ≤ 10. If the temperature range is specified then there must be three numbers on the line – the stopping zone number followed by the two limits to the temperature.

Normally 20 steps occur between the lowest and highest temperature in the map. The number of steps is reset with the **set nmaps** command (page 191 below).

13.56. punch molecules

The densities [cm^{-3}] of all molecules will be punched. The depth and the point and extended visual extinction into the cloud are given in the first three columns. The first line of the output gives header labels for the molecules.

13.57. punch opacities [total, grain, element]

These commands give any of several opacity sources considered by the code. This is a spectrum with the energy in the first column and the opacities in those that follow. This command recognizes the **units** option to change the energy scale (see page 132 above). One of the keywords described in the following sections must appear.

The opacities are only defined over the energy range over which the continuum source is defined, so the resulting opacities will not extend to high energies for softer continua. A hard continuum such as **table agn** (page 37 above) should use used in the input stream to obtain the opacities over the full energy range.

13.57.1. *punch total opacity*

If the keyword **total** appears then the total opacity κ, the absorption cross section multiplied by the density of the absorber, summed over all constituents, will be punched. This has units cm^{-1} – the optical depth would be κ multiplied by a physical depth. This opacity includes all constituents, gas phase and grains, for the last computed zone, with unit filling factor. The first column is the photon energy and the second is the total opacity. The absorption and scattering opacities follow. The fifth column gives the albedo, the ratio $\kappa_s / (\kappa_s + \kappa_\nu)$. The κ's are the scattering and absorption parts of the total continuous opacity. The last column is a label indicating the ionization edge for each species included in the calculation.

13.57.2. *punch grain opacity*

The total grain opacity (all grain types that are included) will be punched after the last zone is calculated. Successive columns give the photon energy, the total (absorption plus scattering) opacity, the absorption opacity, and the scattering opacity. The total and scattering opacity discount forward scattering. The last column gives the scattering opacity with forward scattering included.

13.57.3. *punch fine opacities range 0.7 to 1 ryd, coadd 15 cells*

The code's execution time is partially set by the resolution of the continuum mesh, due to frequent reevaluations of opacities and rates. A very fine continuum mesh, with resolution of 1 km s^{-1} or better, is used to automatically treat line overlap,. The main opacity array cannot use this resolution, because single models would then have execution times of days. Instead, the code uses a multi-grid approach, where a coarse continuum is used for most integrated quantities, but a fine continuum grid is also present to handle the line overlap problem. This command will output the current contents of the fine opacity array. This only includes lines, not the continuum.

Only cells with non-zero opacity will be output. Even then, the file will have many millions of points. If the keyword **range** occurs then the first two numbers on the command line are the lower and upper bounds to the energy range to be output. If neither is specified then the full energy range is output. The last optional number says how many neighboring cells to co-add. If no number appears then 10 is used.

The output will have energies given in Rydbergs unless this is changed with the **units** option. (The range is still given in Rydbergs even if the units are changed).

13.57.4. *punch element opacity [name]*

If neither the **total** or **grains** keywords appear then the name of an element must be specified. The keyword consists of the first four characters of any one of the 30 elements now incorporated in the code. The photon energy (eV) and total photoionization cross section (Megabarns (10^{-18} cm^{-2})) for all stages of ionization of the specified element will be punched. A punch file name must still be specified (to get past the command line parser) but is totally ignored.

The photoionization cross section of each stage of ionization is punched in a series of files. The name of the file will start with the first four characters of the element's name, followed by the stage of ionization (the atom is one), ending with ".opc". Examples are **carb1.opc** or **carb6.opc**. The code stops after producing these files.

13.57.5. *punch opacity figure*

This version of the command creates the file needed to generate one of the figures used in Part 2 of HAZY. The output gives the energy in Rydbergs, then keV, following by the hydrogen, helium, and total gas opacities. The opacities are in units of 10^{24} cm^{-2} and have been multiplied by the cube of the energy in Rydbergs.

13.57.6. *punch opacity shell 26 5 3*

This punches the state-specific photoionization cross section for a subshell of any species. The first number on the command line is the atomic number of the element, the second number the ionization stage, 1 for an atom, and the third number the subshell, between 1 and 7 representing 1s, 2s, 2p, etc. The punch file will contain the incident photon energy in Rydbergs followed by the cross section in cm^2.

13.58. punch optical depths

This creates a file giving the total, absorption, and scattering continuum optical depths for the computed geometry. For a spherical geometry this is the optical depth from the illuminated face to the outer edge of the cloud, and not the total optical depth. The photon energy is followed by the total absorption and scattering optical depths. This command recognizes the **units** option to change the energy scale (see page 132 above).

13.58.1. *punch fine optical depths*

The code's execution time is partially set by the resolution of the continuum mesh, due to frequent reevaluations of opacities and rates. For problems related to line overlap, a very fine continuum mesh, with resolution of 1 km s^{-1} or better, must be

used. The main continuum array cannot use this resolution, because single models would then have execution times of days. Instead, the code uses a multi-scale approach, where a coarse continuum is used for most integrated quantities, but a fine continuum grid is also present to handle the line overlap problem. This command will output the current contents of the fine optical depth array. This only includes lines, not the continuum.

Only cells with non-zero opacity will be output. Even then, the file will have many millions of points. If the keyword **range** occurs then the first two numbers on the command line are the lower and upper bounds to the energy range to be output. If neither is specified then the full energy range is output. The last optional number says how many neighboring cells to co-add. If no number appears then 10 is used.

The output will have energies given in Rydbergs unless this is changed with the **units** option. (The range is given in Rydbergs even if the units are changed).

13.59. punch _OTS

The line and continuum on-the-spot fields will be punched.

13.60. punch overview

This option punches an overview of the thermal and ionization structure of the cloud and is a major output mechanism for the code. The first numbers are the depth [cm], temperature [K], local heating [erg cm^{-3} s^{-1}], total hydrogen density [cm^{-3}], and electron density [cm^{-3}]. All are given as logs except for the depth.

These are followed by various ionization fractions, also given as logs of the quantity. The H$_2$ molecular fraction is expressed as $2\,n(H_2)/n(H)$. Neutral and ionized hydrogen fractions are followed by the ionization fractions for the three stages of ionization of helium, the carbon molecular fraction $n(CO)/n(C)$, the first four stages of ionization of carbon, and the first six stages of oxygen. The last column gives the visual extinction (for point and extended sources) from the illuminated face to the current position.

13.61. punch _PDR

This gives some quantities relevant to a photodissociation region (PDR). The first column gives the depth into the cloud [cm]. The second is the total hydrogen column density [cm^{-2}]. The gas kinetic temperature follows. These are followed by the ratios of densities of H^0, 2H$_2$ and 2H$_2^*$ to total hydrogen, C^0 and CO to total carbon, and water to total oxygen. The next number is the (dimensionless) intensity of the UV continuum relative to the Habing background. The last columns give the total extinction in magnitudes in the V filter measured from the illuminated face of the cloud for a point and extended source.

13.62. punch pointers

This gives the element number, ion stage, and the shell number, for all shells of the elements heavier than helium. This is followed by the energy of the lower and upper ranges of this shell, and the photoionization cross sections at these bounds.

13.63. punch physical conditions

The physical conditions as a function of depth will be punched. The depth into the cloud [cm] is followed by the temperature [K], hydrogen and electron densities [cm^{-3}], heating [erg cm^{-3} s^{-1}] and the radiative acceleration [cm s^{-2}].

13.64. punch pressure

Various contributors to the total pressure in the gas equation of state will be punched. The depth [cm], current total pressure, the pressure at the illuminated face of the cloud, the current gas pressure, the current line radiation pressure, and the current integrated pressure, are included.

13.65. punch qheat

The probability distribution for the grain temperatures is punched. The first column is the grain temperature and the second column gives $dP/d\ln T$. Peter van Hoof added this command.

13.66. punch radius

The zone number is followed by the distance to the central object, the depth to the illuminated face of the cloud, and the zone thickness, all in cm.

13.67. punch recombination [option]

13.67.1. *punch recombination coefficients*

Total recombination coefficients, the sum of radiative, dielectronic and three-body, will be produced for all elements in the code. These rate coefficients [cm^3 s^{-1}] are evaluated at the current electron temperature.

13.67.2. *punch recombination efficiency*

This gives the recombination efficiency for hydrogen, singlet helium and the helium ion.

13.68. punch results

All emission lines with non-zero intensities, and all column densities, can be saved at the end of the calculation by entering the command **punch results last iteration**. This is one way to save the results of a grid of models. The resulting file contains the entire input stream as well.

The resulting punch output will have the line information spread over 6 columns. For some data base applications it would be better to have a single column of results. If the keyword **column** appears then a single column is produced. If no keyword occurs, or the keyword **array** does, then the wide format is produced.

13.69. punch secondaries

The rate secondary ionization of H0, dissociation of H2, and excitation of H I La, are given as a function of depth into the cloud.

13.70. punch source function [depth, spectrum]

13.70.1. *punch source function, spectrum*

The continuum source function for the local diffuse radiation field will be punched. The first column is the photon energy. The second column is the diffuse radiation field at that energy, in units of photons per second per Rydberg. Column three contains the total absorption opacity (cm^{-1}) at that energy. Column 4 contains the source function, the ratio of the diffuse field to opacity (both have the units described above). The last column gives this ratio relative to the Planck function at the local electron temperature.

The last column is a measure of the local source function relative to the local Planck function. This will generally be nearly unity for a thermal plasma close to LTE. Ground states of atoms of hydrogen and helium generally have departure coefficients greater than unity so this ratio will be less than unity at energies when their emission dominates. The helium ion can have departure coefficients much smaller than unity for nebular conditions, so the source function at helium-ionizing energies can be greater than the Planck function.

13.70.2. *punch source function, depth*

The source function of the diffuse fields will be punched for all depths into the cloud at a few energies. The first column gives the integrated optical depth from the illuminated face of the cloud to the current position. The second is the ratio of the diffuse field (photons per Rydberg per second) to the absorption opacity. Ordered pairs of these quantities occur for different energies.

13.71. punch special

If `special` is specified then routine *PunchSpecial* will be called. This routine can be changed to fit the circumstances.

13.72. punch tegrid

The history of the last *NGRID* evaluations of the heating and cooling will be punched. *NGRID* is set in a header file. This is one way to evaluate the stability of the thermal solutions.

13.73. punch temperature

The zone number is followed by the temperature and the first and second spatial derivatives of the temperature with respect to depth, dT/dr and d^2T/dr^2. These derivatives set the rates of conductive heat flow and heat loss respectively.

13.74. punch times

The code will output the zone number, the time required to compute that zone, and the elapsed time since the first zone. This is intended as a mechanism to identify zones that require large amounts of work to converge.

13.75. punch TPredictor

The code tries to estimate the temperature of the next zone from the changes in temperature that have occurred in previous zones. This is only attempted in a constant density geometry. This output gives the old temperature, the estimated new temperature, and the final equilibrium temperature.

13.76. punch wind

The radius and thickness [cm] are followed by the velocity [cm s^{-1}], radiative acceleration [cm s^{-2}], and the dimensionless force multiplier.

13.77. title This is a title

The argument is a title for the calculation, and can be useful for organizing the models in some manner. The title is reprinted several times.

13.78. trace zone 94 [iteration 2; options . .]

This command turns on "trace" information to follow the logical flow within Cloudy. The code uses adaptive logic to control the calculation and this option provides a way to follow these internal decisions.

The trace begins *after* the zone given by the first number on the line. If the zone is zero, or if no numbers occur on the line, then the trace is turned on at the beginning of the calculation. The second (optional with default of 1) number is the iteration on which the trace should be started. It should be set to 2 to turn on the trace for the second iteration. So the command **trace 0 2** would start the trace at the beginning of the second iteration.

Table 21 lists the trace keywords in column 1. The four-character part of the key that must be matched is capitalized. The logical variable in Cloudy that is affected is in column 2. The purpose of each is indicated in column 3.

13.78.1. *trace convergence level*

This is a special form of the **trace** command that will print only an overview of the decisions made during the calculation. The physical state of the gas is determined by nested pressure, temperature, electron density, and ionization solvers. This makes it possible to view the decisions made by any of these solvers.

Table 20
trace convergence keywords and routines

Keyword	Routine
pressure	ConvPresTempEdenIoniz
temperature	ConvTempEdenIoniz
eden	ConvEdenIoniz
ionization	ConvIoniz

The optional keyword on the command line sets the level of information on the output. The code will check for one of the keywords **pressure**, **temperature**, **eden**, or **ionization**, and produce output explaining the decisions made by that solver and all higher ones. With no keyword all levels are printed. Successively deeper layers are obtained with the keywords listed in Table 20.

There are further keywords that create additional printout. The **_OTS** keyword will identify source of OTS fields. The **ESOURCE** option will identify sources of free electrons.

13.78.2. *trace H-like [element name, full]*

This turns on extensive printout describing the physics of one of the model hydrogenic atoms. The same code is used to compute atomic properties for all hydrogenic species. If the keyword **full** appears then the printout will be far more detailed. If no element is specified then this will only be for hydrogen itself. If the name of any other element appears then the printout will be for that element.

13.78.3. *trace He-like [element name, full]*

This turns on extensive printout describing the physics of one of the model helium-like sequence atoms. The same code is used for to simulation all helium-like species. If the keyword **full** appears then the printout will be far more detailed. If no element is specified then this will only be for helium itself. If the name of any other element appears then the printout will describe that element.

Table 21
Trace Keywords and Effects

keyword	Quantity traced
BETA	OI 8446-Lβ problem
CARBon	carbon ionization equilibrium
CALCium	calcium ionization balance
COMPton	Compton heating, cooling, and ionization
CONTinuum	prints out photon arrays, pointers
CONVergence	convergence loop, no other printout
COOLants	cooling
DIFFuse fields	sum of recombination coef in DIFFEM
_DR_	choice of next zone thickness
EDEN	changes in electron density
GAUNt	the free-free gaunt factors
GRAIN	details dealing with grain treatment
HEATing	heating agents
HEAVies	heavy element balance
HELIum	helium ionization equilibrium
HELIum ATOM	Helium singlets ionization equilibrium
HELIum _IONized	helium ion ionization equilibrium
HELIum SINGlet	Helium singlets ionization equilibrium
HELIum TRIPlet	helium ion ionization equilibrium
HYDRogen	Minimal trce of the H ionization
IRON	Fe abundance, K-alpha emission
LINEes	line pointers, opacity. A's, etc
leveln	LevelN n level atom routine
Ly BETA	Lβ - OI 8446 pumping problem
CMOLE HMOLE	Turn on either H_2 or CO network
MOLEcules	rate coefficients for molecules
NEON	recombination, ionization for neon
OPTIcal depths	inner, outer optical depths in STARTR
oPTIMizer	Steps in optimize command driver
_OTS	ots ionization rates
POINters	pointers for element thresholds
THREe body	three-body recombination rates for metals
TWO photon	induced two photon processes

14. THE OPTIMIZE COMMAND

14.1. Overview

The **optimize** command and its keywords tell the code to vary one or more of the initial parameters to try to find a set of parameters to reproduce an observed emission-line spectrum, line flux or luminosity, a set of column densities and/or temperatures. R.F. Carswell wrote the original code and first implemented the

Table 22
Commands with Vary Option

Command	Quantity varied	Min	Max	Inc.
abundances starburst	metallicity	0.001	36	0.2
blackbody	temperature	def	def	0.5
bremsstrahlung	temperature	def	def	0.5
constant temperature	temperature	def	def	0.1
cosmic rays	cosmic ray density	def	def	0.2
dlaw	arbitrary density law	def	def	0.5
element xxx	abundance of an element	def	def	0.2
energy density	energy density temp	def	def	0.1
filling factor	filling factor	def	0	0.5
globule	density	def	def	0.2
grains	grain abundance	def	def	1.0
hden	hydrogen density	def	def	1.0
intensity	intensity of source	def	def	0.5
ionization parameter	ionization parameter	def	def	0.5
luminosity	luminosity of source	def	def	0.5
metals	metallicity	def	def	0.5
phi(H)	photon flux	def	def	0.5
power law	see below	def	def	-
Q(H)	ionizing photons	def	def	0.5
radius	inner radius	def	def	0.5
ratio	alpha ox	def	def	0.2
set H2 Jura scale	scale factor for Jura rate	def	def	0.3
stop column density	column density	def	def	0.5
stop thickness	cloud thickness	def	def	0.5
table Draine	Scale factor	def	def	0.1
table ISM	scale factor	def	def	0.2
table star Atlas	temperature	3,500	50,000	0.1
table star CoStar	temperature	18,521	53,397	0.1
table star Kurucz	temperature	30,000	50,000	0.1
table star Mihalas	temperature	30,000	55,000	0.1
table star Rauch	Temperature	50,000	500,000	0.1
table star Werner	temperature	50,000	500,000	0.1
table starburst	age			0.3
turbulence	turbulent velocity	def	def	0.5
xi	Ionization parameter	def	def	0.5

method in Cloudy. Any of several minimization methods can be used obtain a best fit to a set of observed quantities. The desired emission-line spectrum, line flux or luminosity, column densities, and/or temperatures are specified by a series of **optimize** commands. A keyword **vary** can appear on several of the commands used to specify initial conditions (Table 22) to indicate which parameters are to be varied.

14.2. Commands with vary option

All commands with the **vary** option are listed in Table 23. The section beginning on page 165 below discusses details of some commands.

14.3. What must be specified

At a minimum, a desired emission-line spectrum, a line luminosity, temperature, or a column density must be given, along with a specification of which parameters are to be varied. The parameters to be varied are specified by a keyword **vary** which may appear on any of the commands listed in Table 22. Up to 20 parameters may be varied at a time. The quantities to be varied are actually entered as logs within the code, and increments (the first steps away from the initial guess) are also logarithmic.

Several examples of the **vary** option in action are given in the test suite. A typical input stream follows:

```
// tell the code to vary the ionization parameter
// and hydrogen density
blackbody, 50,000K
hden 4 vary
ionization parameter -2 vary
stop zone 1
//
// the following specifies observed emission lines, order is
// label, wavelength, intensity relative to H-beta, tolerance
optimize lines
O  3 5007 intensity =13.8 error =0.1
totl 3727 < 2.1 (only upper limit)
end of lines
//
end of column densities
```

This example tells the code to vary the density and ionization parameter to reproduce the observed intensities of two emission lines.

Information concerning the optimization process is fed to the code as a series of keywords on the **optimize** command. These are described next. Only one keyword will be recognized per **optimize** command.

14.4. Observed quantities

This section describes how to specify the observed properties we want to match.

14.4.1. *optimize, column density*

This specifies a set of column densities. A series of column densities, ending with a line with the keyword **end** in columns 1 to 3, will be read in from subsequent lines. One column density is entered per line, and up to 100 may be specified. Columns 1 to 4 of the column density lines must contain the first four characters of the name of

the element, spelled as in the output from the zone results. The first number on the line is the ionization stage, 1 indicates Atom I, 3 indicates Atom III, etc. The second number on the line is the log of the column density [cm^{-2}], and the last optional number is the relative uncertainty. It has a default of 0.05 (5 percent). A column density can be specified as an upper limit by entering < anywhere on the line. If "<" appears then the column density is only included in the optimization if the predicted value exceeds the upper limit.

The following gives some examples of its use;

```
optimize column densities
hydrogen 1  < 17 ;make optically thin in Lyman continuum
carbon 4 17.4 error =.001
silicon 3 14.6
end of column densities
```

Molecular column densities can be requested, but this requires a bit of careful formatting. The molecules whose column densities the code can recover are given in a table in Part 3 of this document. The Chapter "Cloudy as a subroutine" includes a section "checking predictions" that describes routine *cdColm*. This optimize command simply passes the label and ion stage to that routine. A series of molecules and excited states of some ions can be recovered by entering the string that is described ("H2__" for H$_2$) where *cdColm* is described. The first four characters of the label given on the input line line must agree with those listed there. The ionization stage given there must also be specified, this is zero for a molecule.

The following shows some examples of optimizing molecular column densities;

```
optimize column densities
H-   0  < 17 ;make optically thin in Lyman continuum
H2   0 17.4 error =.001
H2+  0 14.6
end of column densities
```

Note that extra spaces were placed after the labels for the molecule to pad out the needed four-character label. Zero is always the ion, the first number after the four-character label.

14.4.2. optimize, [intensity, luminosity]=36.425 [error =0.1]

This command specifies the luminosity or intensity of an emission line. The code will try to make the predicted intensity or luminosity of the normalization line (usually Hβ, and set with the **normalize** command) match this value. The sub-keyword is either **intensity** or **luminosity**, and both have exactly the same effect. The number is the log of either the intensity or luminosity of the line, in the same units as found in the third column of the final print out. The second (optional) number is the fractional tolerance allowed for the fit between the observed and computed values. If a tolerance is not specified, then a fractional uncertainty of 0.10 is assumed.

The following gives some examples of its use;

```
// this will request an hbeta intensity of 0.5 erg cm^-2 s^-1
// it applies to Hbeta since this is the default normalization line
optimize intensity -0.3

// the following resets the normalization line to 5007, then
// asks the code to reproduce its luminosity
normalize to "O  3"  5007
// we want a 5007 luminosity of 10^34.8 erg / s
optimize luminosity 34.8
```

14.4.3. *optimize, lines*

This command tells the code to try to reproduce a set of relative emission-line intensities, and to begin reading a list of observed lines. Up to 100 lines may be entered.

```
//
// the following specifies observed emission lines, order is
// label, wavelength, intensity relative to H-beta, tolerance
optimize lines
O  3 5007 intensity =13.8 error =0.1
totl 3727 < 2.1 (only upper limit)
O  3 88.33m
end of lines
//
```

One emission line is specified per line, and the line must contain information in a specific order. Columns 1 to 4 list the label followed by the wavelength used to identify the line. Both must exactly match the identifications used in the emission line output. The code will stop if this is not the case. The third quantity is the observed relative line intensity. This will be in the same units as the relative intensities printed at the end of the calculation. Intensities are normally relative to Hβ, but can be changed to other reference lines with the **normalize** command (described on page 120 above). The last (optional) number is the relative error in the observed line. If an error is not specified, then a fractional uncertainty of 0.05 (5%) is assumed. A line can be specified as an upper limit by entering < anywhere on the line. If "<" appears then the line is only included in the optimization if the predicted value exceeds the upper limit.

The series of emission lines ends with a line that has the keyword **end** in columns 1 to 3. If this end does not appear correctly then the code will continue reading lines until the end-of-file is encountered.

Comments may be entered using any of the special characters in column 1 that were described on page 16 above.

14.4.4. *optimize temperature*

This specifies a set of observed mean temperatures. Each of the lines that follow gives a species, ion stage, mean temperature, an error, and a specification of how to weight the temperature, linear or volume. The following is an example:

```
optimize temperature
H  1  1  36,200  volume
H2   0   150     radius
end of temperatures
```

This behaves as does the previous commands. The "<" option is recognized. Numbers ≤ 10 are interpreted as logs of the temperature. The keywords **volume** and

radius specify how to weight the mean, the default is over radius and will be used if no keyword is recognized. The label for the element must appear in the first four columns, the ion stage follows, with the temperature and an optional error coming next.

This command simply copies the labels and parameters on the each line and passes them to *cdTemp*, a routine that determines the mean temperature and is described in Part 3 of this document. That documentation describes how to request mean temperatures for various ions, molecules, and derived quantities.

14.5. Optimization methods

Several optimization methods have been incorporated into Cloudy. The Subplex (Rowan 1990) method is the default.

14.5.1. *optimize, amoeba*

This tells the code to use the Amoeba optimization algorithm. The method used in the Cloudy distribution is closely based on the logic given in Press et al. (1992) but has been totally rewritten to allow open redistribution.

14.5.2. *optimize phymir [sequential mode] [8 cpu] [continue]*

This tells the code to use Peter van Hoof's *phymir* optimization method (van Hoof 1997). In particular it tries to avoid being upset by the inevitable numerical noise that is present in any simulation, and tries to use global information in a better manner. It has the further option of being able to run on multiple CPUs on multiprocessor UNIX machines.

Phymir by default runs in parallel mode (using more than one CPU) on UNIX systems and in sequential mode on non-UNIX systems. The keyword **sequential** switches *phymir* to sequential mode on UNIX systems. The keyword **cpus** overrides the default number of processors. When **cpus** is specified the command parser searches for a number on the line. This is the maximum number of CPUs that are used simultaneously at any given time when *phymir* runs in parallel mode.

The continue option. The *phymir* method also includes the option to restart the optimization somewhere in the middle of the calculation by saving the code's state after each model. In this case the keyword **continue** must be entered on the **optimize** command. The optimization does not start from scratch, but re-starts using information in the file *continue.pmr* instead.

14.5.3. *optimize, Subplex*

This tells the code to use Rowan's (1990) implementation of the subspace-searching simplex optimization method. This is probably the most robust, and is the default.

14.5.4. *optimize, Powell*

This tells the code to use the Powell routine to search for the best parameters. The logic is closely based on that given in Press et al. (1992) but has been totally recoded to allow redistribution.

14.6. Controlling the optimizer

14.6.1. optimize, file= "best.in"

At the end of the optimization process the optimal input parameters are written into a file for later use[36]. The default filename is "**optimal.in**" but can be changed with this command. The file name must be valid for the system in use and must be enclosed in double quotes.

It is possible to use this file to do later calculations in which various quantities might be punched for plotting. Also, it is generally a good idea to confirm that a single run with Cloudy does reproduce the final results from the many calls of the code made by the optimization method. The two should agree *exactly*, but would not if the code became corrupted during the many calls made during the process.

14.6.2. optimize, increment = 0.5 dex

Increments are the logarithmic quantities that will be added to or subtracted from the initial guess in the first steps away from the initial value. The default increments preset in the code and listed in Table 22 were chosen with typical conditions in mind. It may be necessary to increase these if the process is unable to identify a solution. If a zero is entered as an increment, then the default increment will not be changed.

The increments entered with this command affect *only* the *previously* selected **vary** command. The following gives some examples of changing the increments.

```
hden 4 vary
optimize increments .1 ;this sets .1 dex changes in hden
brems 6 vary          ;increments left at default
radius 13.6 vary
optimize increments .05; this sets changes in radius
```

14.6.3. optimize, iterations =75

This specifies the upper limit to the number of iterations to be performed. The default is 20. This command is quite similar to the **stop zone** command, in that it is a fail-safe method to prevent runaway infinite loops. The optimization process should not stop because of this limit. It may be necessary to increase the limit if the process is still making progress at the end of the calculation.

The number of iterations that is required is a function of the number of parameters that are varied. Each optimizer will spend a certain number of iterations changing each free parameter. Usually only one free parameter is varied during each iteration. To work well, there must be enough iterations to allow each of the free parameters to be varied over a wide range. Make sure that this happens by looking at why the optimization process ended – if it ran out of iterations check whether all of the parameters been varied over a significant range. If they have not, increase the number of iterations or vary fewer parameters.

---

[36] This replaces the **optimize punch** command, that was in versions 90 and before.

14.6.4. optimize, range -2.3 to 3.9

The preset limits to the range over which parameters can be varied are indicated in Table 22. The entry *def* indicates the default limits of -1×10^{37} and 1×10^{37}.

It is sometimes necessary to establish physical limits to parameters. For instance metallicities may be limited to the range $-1 \leq \log(Z) \leq 0$ by observations or physical plausibility. The optimization driver does not know this but can be told a set of bounds with this command. The argument is an ordered pair of limits. These are the log of the lower and upper limits to the allowed range of variation of the *previous* command with **vary** specified. Examples follow.

```
hden 4 vary
// the following sets limits to range of density
optimize range from 3 to 5
// There will be no range for this one
brems 6 vary
radius 13.6 vary
// this sets limits to radius
optimize range from 13 to 14
```

The optimizer does not actually know about the range limit. A residual of 10^{37} is returned if a parameter outside the allowed range is used.

14.6.5. optimize, tolerance = 0.02

The tolerance is a measure of the desired fractional accuracy of the parameters that are varied and is set with this command. The default value of 0.10 should be sufficient for initial trial runs, but the final values should be made lower for more precision.

14.7. Convergence criteria

Observables are placed into one of four categories – "spectrum", "column density", "temperature", and "absolute flux"(see van Hoof 1997) . For the ith observable the error estimate is

$$\chi_i^2 = \left(\frac{F_i^m - F_i^0}{\min\left(F_i^m, F^0\right) \ \sigma} \right)^2 \tag{70}$$

where F^m and F^0 are the model and observed values and σ is the relative error in the observed value. The uncertainty σ is specified when the observed quantities are entered and has a default value of 0.05 (5 percent). The average of this is computed for each category, and the error estimate summed over all categories is minimized.

14.8. Other optimizer commands

14.8.1. no vary

It is sometimes useful to be able to turn off the optimizer for a given input stream, without having to change the (possibly many) occurrences of the **vary** keyword. If the **no vary** command is entered then the **vary** keyword on the other commands will be ignored and a single model will be computed.

14.8.2. optimize, trace start 4

This command turns on trace printout for the nth time the code is called by the optimizer. Specific aspects of the trace are still controlled by the **trace** command, described on page 157 above.

14.8.3. optimize, trace flow

This command turns on trace to follow the logical flow within the optimizer.

14.9. Notes concerning commands with vary option

The keyword **vary** can appear on the commands in Table 22. Notes concerning these follow.

14.9.1. blackbody

It is not possible to specify the luminosity of the blackbody by using the keywords on the **blackbody** command when the **vary** option is used. It is necessary to enter the luminosity using some other command, such as **luminosity** or **ionization parameter**.

14.9.2. dlaw

Up to 5 numbers may be entered on the **dlaw** command, but only the first number will be varied.

14.9.3. elements

Either the absolute abundance of an element relative to hydrogen, or the scale factor multiplying the abundance, can be varied.

14.9.4. filling factor

Only the filling factor itself can be varied. It is possible to specify the optional power law for a radial dependence but it is not possible to vary it.

14.9.5. hden

It is possible to specify the exponent for the optional power-law density dependence upon radius or thickness. It is not possible to vary this additional parameter; only the hydrogen density is varied.

14.9.6. intensity

It is possible to specify all of the options on the **intensity** command, but it is only possible to vary the intensity itself.

14.9.7. luminosity

It is possible to specify all of the options on the **luminosity** command, but it is only possible to vary the luminosity itself.

14.9.8. metals

The **grains** keyword can also be specified.

14.9.9. *phi*(H) *and Q*(H)

It is possible to use the **range** option, but only the log of the photon number will be varied.

14.9.10. *power law*

The **vary** keyword appears in three forms, **vary**, **varyb**, and **varyc**. If **vary** appears then the first parameter, the slope of the power law, is varied. If **varyb** appears then the second parameter, the cutoff temperature in degrees Kelvin, is varied. If **varyc** appears then the last parameter, the low-energy cutoff, is varied. Only one parameter may be varied at a time.

14.9.11. *punch output*

It is not possible to use some of the punch output options while optimizing. Punch output is designed to save predictions so that they can be analyzed after the calculation is complete. It makes little sense to do this during an optimization run with its hundreds of calculations. After the optimization is complete, rerun the optimal simulation with **punch** commands included.

Most **punch** options will work during optimization. Those that read a series of lines to determine what quantities to output will not work. This is because the command parser is not used in its usual mode when commands are read during an optimization run.

14.9.12. *radius*

It is possible to specify the stopping radius or depth on the line, but it is not possible to vary it. Only the starting radius is varied.

There could be a major source of confusion if the second parameter is entered and the two numbers are of the same order of magnitude. The logic used to interpret the second number is described on page 79 above. If the second number is greater than the first then it is interpreted as an outer radius; if less than, then the depth. As a result, the interpretation of the second number can change while the first number is varied. It is safer to set an outer radius with the **stop thickness** command (described on page 118 above) rather than using the second number on this command if there is any danger of this confusion happening.

14.9.13. *stop column density*

All of the optional keywords (neutral, effective, etc.) are recognized.

14.9.14. *stop thickness*

Only one thickness can be specified.

14.9.15. *table stars*

Only the temperature can be varied, not the gravity.

14.9.16. *table star CoStar*

The first parameter on the command line will be varied.

14.9.17. table starburst

The age is varied. The code will stop if the age is does not lie within the range of ages in the starburst data file. You must specify the range of ages to the considered with the **optimize range** command, described on page 164 above. That command must appear after the **table starburst vary** command.

14.10. Notes concerning the optimization process

14.10.1. Use physically motivated initial conditions

The algorithm will not be able to find a solution if one is not physically possible. For instance, an observed He II $\lambda 4686$/Hβ intensity ratio of 0.5 cannot be produced by a 20,000 K blackbody, no matter how many other parameters are varied (it produces no He$^+$-ionizing radiation). It is probably necessary to start with parameters in the general area of the successful model. When far from the solution, it is also a good idea to use a large tolerance (using the **optimize tolerance** command) to stop it from over-optimizing a bad solution.

14.10.2. Change the increment size

The initial increment will be the largest step ever taken during the optimization process for some of the optimization methods. If the initial parameters are far from the solution then it may be wise to increase the increments. Depending on the optimization method used, it may not be possible to find solutions more than one or two increments away from the initial guess. If the increments are too big it may jump over valid solutions.

14.10.3. Set physically motivated limits to the variable quantities

The optimizer driver uses brute force methods, and understands surprisingly little modern astrophysics. For instance, while trying to reproduce an observed He II $\lambda 4686$/Hβ intensity ratio of 0.5 by varying the temperature of a blackbody radiator, the algorithm is likely to examine the consequences of photoionization by a 100 K radiation field. Physically, it is known that He II emission only occurs for stars hotter than $\sim$50,000 K (AGN3 section 2.5), so there is little purpose in examining temperatures lower than this. The process will converge more quickly if reasonable bounds to the range of the varied quantities are set using the **optimize range** command.

This advice is dangerous, of course, since you may limit yourself to solutions close to those you anticipate. Experiments should also be performed far from the anticipated solution.

14.10.4. Don't give up!

My experience is that this process works about a quarter of the time. The problem is that the algorithm can easily home-in on a local minimum which is actually a very bad global solution. When this occurs, the best idea is to restart the optimization process with a different set of initial conditions. Better yet is to start the process with parameters that give answers known to be close to the solution, although there is

Table 24
Optimizer results

Method	Iterations	Density	temperature
Initial condition	-	5	4
Amoeba	42	4.987	4.001
Phymir	46	4.946	4.001
Powell	73	4.999	4.000
Subplex	39	5.002	4.002

some danger of limiting the outcome to be what your expect. Finally, don't be afraid to use CPU time.

14.11. Other optimization methods?

Astrophysics is basically concerned with the inverse problem – knowing an answer (the spectrum) and trying to deduce the question (the conditions that caused it). Optimizing a multi-dimensional function is more an art than a science. A truly robust optimization method would make Cloudy a far more useful research tool. I would be interested in learning about, and possibly adopting, other promising optimization methods. License-free code is necessary since Cloudy is totally Open Source.

14.12. The optimizer test cases

The suite of test cases that comes with the code includes scripts to drive each of the optimizers. These scripts are *optimize_amoeba.in*, *optimize_pahymir.in*, *optimize_paowell.in* , and *optimize_saubplex.in*. These were produced by first running the code at a hydrogen density of 10^5 cm^{-3} and a temperature of 10^4 K. The spectrum of [O II] and [O III] emission lines was taken from this calculation. Each optimizer starts at a density and temperature some distance away from this solution and tries to reproduce the spectrum. Table 24 shows the results of this test.

15. MISCELLANEOUS COMMANDS

15.1. Overview

This section describes commands that are used to disable physical processes, change the code's internal behavior, or to take care of housekeeping activities.

15.2. Introduction to the compile commands

The following subsections describe how to set up and then compile some external data files that are needed by the code. The stellar atmosphere files must be compiled if you wish to use some of the **table star** continua. The He-like and grains files are included in the code's distribution and only need to be compiled if you need to change the code.

The files produced by the compilation process must be accessible to the code when it is executed from other directories. This is most easily done if you set the path in the file **path.c** (described on page 191 below) to point to the directory containing its data files, and then place the compiled files in this directory.

15.3. compile stars

Kevin Volk originally incorporated several large grids of stellar atmosphere continua into Cloudy, and Peter van Hoof made several extensions. The **table star atlas**, **table star Rauch**, **table star CoStar**, and **table star Werner** commands use these atmospheres. The data files are very large and "direct access" is used to read these files quickly. The result is that the final files are not portable, although the code used to read or write them is. The process of converting the stellar atmosphere files from their original format into a form that can be read by Cloudy is referred to as compiling the stellar atmospheres.

These **table star** commands will only function if the atmosphere files are compiled as described here. This is only done once while installing Cloudy, although it will have to be done again if you ever change the continuum energy mesh. This step does not need to be done if you don't want to use these stellar atmospheres.

15.3.1. Preparing the Rauch stellar atmospheres

Obtain the tar files from Thomas Rauch's Web site http://astro.uni-tuebingen.de/~rauch/flux.html.

There are "first generation" and "second generation" grids available. The second generation gives a better representation of the opacity effects of heavy elements, but does not extend to as high a temperature as the first generation. The second generation is used by default, although it is possible to also use the first generation. Each generation also has two abundance sets, halo and solar.

Second generation. The link given above points to the top level for all the stellar atmospheres. You want the "second generation H-Ni" set. Click on the "H-Ni" link. At the bottom of the page you will find two sets of atmospheres. You want the solar

and halo abundances dated July 2002 and June 2002 respectively. The general format for each atmosphere is *temperature_gravity_abundances.bin_0.1*.

First generation. This set is not necessary, but can be used if desired. The link given above points to the top level for all the stellar atmospheres. You want the "first generation H-Ca" set. Click on the "H-Ca" link. At the bottom of the page you will find several sets of atmospheres. You want two sets of files, for solar and halo abundances. These are dated Feb 1998 and May 1998 respectively, and have a note that says that they are for versions 90.05 and later. The general format for each atmosphere is *temperature_gravity_abundances.wf*.

Both generations. Uncompress and explode these tar files into a subdirectory. This will create a large number of stellar atmosphere files with names ending with the extension "*.wf*" (the first generation) and "*bin_0.1*" (the second generation). Next execute Cloudy with the single command

```
compile stars Rauch initialize
```

to compile the second generation files. This will create the files *rauch02_halo.ascii* and *rauch02_solar.ascii* that are needed for the final step. Only these two files need be retained. If you also want to use the first generation grid you will also need to execute the code with the single command

```
compile stars old Rauch initialize
```

This will create the files *rauch_halo.ascii* and *rauch_solar.ascii* that are needed for the final step.

15.3.2. Obtain the CoStar spectra

These are developed by Daniel Schaerer and are described in Schaerer et al. (1996ab) and Schaerer & de Koter (1997). There are two abundance sets, halo and solar. They are available on the web from http://www.stsci.edu/ftp/science/starburst/SdK96.html - you want the files *Sc1_costar_z020_lb.fluxes* (solar abundances) and *Sc1_costar_z004_lb.fluxes* (halo metallicity).

15.3.3. Werner and Kurucz star files

The three files *werner.ascii*, *kurucz.list*, and *kurucz.ascii* files come directly from the Cloudy Web site.

15.3.4. The final set of ASCII files

At this stage the files you have are the following ASCII files:

Sc1_costar_z020_lb.fluxes and *Sc1_costar_z004_lb.fluxes* These are the set of CoStar stellar atmospheres.

werner.ascii This is a plain ASCII version of the Werner and Heber (1991) grid of hot stellar atmospheres. These data extend from 10^{-5} Ryd through 182.25 Ryd.

kurucz.list This is a plain ASCII list of all files in the Kurucz (1991) grid.

kurucz.ascii This is a subset of the Kurucz (1991) atmospheres. The wavelengths of the grid are stored in nanometers as the first record. These data extend from 9.090 nm (10.0 Ryd) through 1.600×10^5 nm (5.7×10^{-4} Ryd).

rauch02_halo.ascii and *rauch02_solar.ascii* These are the sets of second generation Rauch hot stellar atmospheres as created and described in section 15.3.1 above.

rauch_halo.ascii and *rauch_solar.ascii* These are the sets of first generation "old" Rauch hot stellar atmospheres that was described in section 15.3.1 above. These first generation grids are not needed but can be created if you wish.

15.3.5. Compiling the star files

Execute Cloudy with only the single command **compile stars** as input. Examine the resulting output for any comments indicating success or failure. I do this with the output coming to the screen so that I can monitor progress. This step typically takes about 15 minutes on my workstation.

This does not compile the first generation Rauch stars. If you want to use them it is also necessary to execute the code one time with the command **compile stars only old Rauch** as input.

15.3.6. The final files

Six direct access files, *atlas.mod*, *rauch02_halo.mod*, *rauch02_solar.mod*, *costar_sol.mod*, *costar_halo.mod*, and *werner.mod* will be created. Additionally the files *rauch_halo.mod*, *rauch_solar.mod* will be created if the first generation Rauch atmospheres are needed. These are the files that Cloudy must access to use the **table star** continua.

15.3.7. Using the stellar atmospheres

The location of the code's data files was specified by entering the path in the file *path.c* as described on page 191 below before creating the Cloudy executable. The "*.mod*" files should be located in this data directory. The **table stars** command will then function normally.

The code will stop if it cannot locate the *.mod* file when the corresponding **table stars** command is entered. The code also checks that the energy grid in the compiled data file agrees with that in the code itself to confirm that the star files are appropriate for the current version of Cloudy. As a result it is generally necessary to recompile the stars files when the code is updated.

15.3.8. Cleaning up after compilation

Only the *.mod* files are needed to compute simulations. The *.ascii* files and the *.wf* atmosphere files can be deleted or compressed. They will only be needed again if the continuum mesh within the code is changed. If this happens then it will be necessary to recompile the binary files. The Rauch *.wf* files will never be needed again, unless one of the atmospheres is updated and it is necessary to recreate this set.

15.3.9. *If the continuum binning is changed*

The continuum resolution is specified by the contents of the file *continuum_mesh.dat*, which lives in the data directory. That file describes how to change the resolution. If the continuum binning is changed then it will be necessary to recompile the star files, using the **compile stars** command.

15.3.10. *Compiling only one set of stars*

compile stars only xxx. You may not need all of the stellar atmospheres. The keyword **only** followed by one of the keywords **Rauch**, **Atlas**, **CoStar**, or **Werner** will only compile that set of stellar atmospheres.

15.4. compile opacities

N.B. This command does not now function and may be removed.

When the code is initialized it spends some time evaluating numerical fits to the needed opacities. This initialization time can be saved if the opacities are compiled and the resulting file placed on the path. To do this, execute the code and enter only the command **compile opacities**. The code will generate a binary file named **opacity.opc** containing the needed opacities and array indices. This file should be in the directory where the other data files are stored.

N.B. It is not really necessary to compile the opacities – the code will generate them when it starts up if the file does not exist. This may actually slow down the calculation if you are using a fast computer on a slow network.

15.5. compile gaunt factors

The code normally reads in a table of free-free gaunt factors when it initializes. This command regenerates that file. It was introduced by Ryan Porter.

15.6. compile He-like

The code reads in a table of recombination coefficients for helium-like ions when it initializes. This command regenerates that file. It was introduced by Ryan Porter.

15.7. compile grains

This command prepares the grain opacity files that are used by the **grains** command (page 60 above). This does not need to be done if the set of grains included in the data distribution (see Table 18 on page 61 above) are sufficient. If you wish to create other species, with their own size distribution and refractive indices, then these new species must be compiled with this command. The *vanhoof_grain_model.pdf* file in the data distribution subdirectory describes how to create new grain opacity files. It is also necessary to compile the grains if you change the code's continuum energy mesh.

This command uses a spherical Mie code originally developed in collaboration with Peter G. Martin and Peter van Hoof, and implemented into Cloudy by Peter van Hoof, to generate sets of grain opacities from a description of their size distribution

and grain material optical properties (van Hoof et al. 2004). The set of grain opacities created by this **compile** command are then used to compute temperature, charge, drift velocity, and emitted spectrum, for each bin within the grain size distribution.

In most cases refractive index and size distribution files included in the data from the Cloudy web site will be used. In all cases sets of refractive index files (with names ending in "*.rfi*") are combined with size distribution files (names ending in "*.szd*") to produce an opacity file (names ending in "*.opc*").

Specific examples of this step follow.

15.7.1. compile grains

The minimum number of grain types needed for the **grains** command to function will be compiled if the command **compile grains** is entered and no filename or other keywords are recognized. This command must be given from within the directory that contains the data files. This only needs to be done if the code's energy mesh is changed (since precompiled opacity files are included in the data distribution) or if you wish to create a new grain type with your own optical constants or size distributions.

Examine the output to check for problems. This step typically takes about 15 minutes on my workstation.

15.7.2. compile grain 10 bins, [filename, ism graphite]

This version of the command will produce opacities of a single grain species and is used when you wish to create your own grains. Both the optical properties and size distribution must be specified. This can be done by giving keywords (to use built-in types) or a filename (to read properties for a new species).

The number on the command line specifies the number of grain size bins to compute. If no number appears then 10 is set by default.

Keywords for standard data sets One of a set of keywords may be used to specify refractive index and grain size distributions. The keyword is the name of a file included in the Cloudy distribution, without the filename extension.

Table 25
Grain keywords for refractive index information
*.rfi

Keyword	Grain type	Reference
ac1-amcarb	amorphous carbon	Rouleau & Martin 1991
be1-amcarb	amorphous carbon	Rouleau & Martin 1991
gdraine	Graphite	Draine web site
graphite	graphite	Martin & Rouleau 1991
grey, gray	grey grain	
PAH	PAH	
sdraine	silicate	Draine web site
silicate	astronomical silicate	Martin & Rouleau 1991
vacuum	vacuum	

The keywords given in Table 25 may appear to specify a file containing refractive index information. These files have names that end in "*.rfi*".

A grain size distribution may be specified with one of the keywords given in Table 26. The first three are single-sized distribution functions and are mainly used for testing the code. For single-sized grains the supplied number of cells is ignored since one cell is always used. These files have names that end in "*.szd*".

Creating new data files Either a keyword or filename can be entered with the **compile grains** command. The keywords are summarized in the tables. If these do not occur then a used-generated file is used. A pair of double quotes on the command line is assumed to surround the filename, as in "*name.rfi*". If the name ends in "*.rfi*" it is interpreted as a refractive index file while names ending in "*.szd*" are size distribution files. Filenames and keywords can be mixed on a single line, as shown in the examples below.

Table 26
Grain keywords for size distribution
*.szd

Keyword	Size distribution
0m010	0.01 micron
0m100	0.1 micron
1m000	1 micron
0n341	Small PAH, 15 C atoms
0n682	Large PAH, 120 C atoms
_ism	ISM distribution
bt94	Bakes & Tielens 1994
orion	Orion distribution

15.7.3. Examples

```
// compile all grain types
compile grains

// only ism graphite
compile grains ism graphite

// use 20 bins for a silicate
// with Orion size distribution
compile grains 20 Orion silicate

// the grey.rfi file is the one read
// with the key grey, so the following
// is equivalent to grey ism
compile grains "grey.rfi" ism

// explicitly request the graphite.rfi
// (default with graphite) refractive index
// file, and the Orion size distribution
compile grains "graphite.rfi" "orion.szd"
```

15.8. crash [zero, overflow, assert, NaN, …]

This command should cause the code to crash. It is intended as a way to confirm that the machine environment has been correctly set. One of the keywords must appear.

The IEEE standard for floating point arithmetic says to *not* throw an exception when a floating point exception occurs. Instead, the result is set to **+/-inf** or **NaN** (not a number) and the calculation continues. The code will only crash if this is explicitly requested by setting compiler options or enabling traps on the CPU.

Cloudy should be set up to crash on division by zero, overflow, or invalid operation. The **zero**, **overflow**, and **NaN** options on the **crash** command are available to make sure that the correct compiler options have been set to trap these exceptions. The code will crash for each case if it has been properly set up.

crash zero causes the code to divide a positive number by zero. This causes a division by zero exception.

crash overflow causes the code to divide a very large number by a very small number. The result will overflow.

crash long overflow causes the code to try to convert a floating number that is larger than the largest long int into a long int. Some machines may throw an exception in this case (mine does not).

crash NaN will divide zero by zero. This will cause an invalid operation exception.

crash setnan will test the macro SET_DBL_SNAN, which sets double variables to NaN.

crash setnan float will test the macro SET_FLT_SNAN, which sets float variables to NaN.

crash assert causes the code to assert that a positive number is less than zero. This should cause an exception when the code is compiled in debug mode, but will have no effect when the optimization is set to a high level (or the flag -DNDEBUG is included with the compiler options). If the code does indeed crash with a failed assert then it may request that the output be sent back to Lexington. This is normal, and only shows that the asserts are working properly.

crash bounds will cause an array to be evaluated with an index that exceeds the array bounds. The C standard does not require that compilers have options to check that array bounds limits are not exceeded, and few C compilers offer this. This is one area where Fortran is clearly superior to C. Array-bounds checking is available as an add-on for many compilers, however.

The command has an option, the offset to add to the array limit. If it is positive *n* then the code will access **array[ARR_SIZE+n]**. If it is negative then the code will access **array[n]**. The default is zero, that is, to access **array[n]**. If either of two keywords, **_low** and **high**, occur with no number than the code will assume **n = 2** and check only the low or high bounds.

By default this bounds check is for an array that is produced by a *malloc* off the heap. The keyword **static** will use a static array and either keyword **stack** or **auto** will use an array created off the stack.

crash undefined will cause the code to multiply a valid constant by an undefined floats. It has three possibilities. The keyword **static** will multiply by an undefined static variable. The keyword **stack** or **auto** will multiply by a variable created off the stack. If neither keyword occurs than the variable comes off the heap

after a *malloc*. The ideal compiler would produce code that crashed when any of these occur.

15.9. Dielectronic recombination

This was replaced with the **set dielectronic recombination** command, described on page 193 below, in version 06.01.

15.10. drive fread, gaunts, pointers, ...

The **drive** command enters a special debug mode in which the program requests information and responds with deduced quantities. The specific debug mode is selected with the optional keywords, as described next. Parameters for a valid model (density, continuum, and luminosity) must still be specified. The debug mode is entered *after* the last command is specified and the input stream ends.

15.10.1. *drive Case B*

This option is to interpolate on Storey and Hummer (1995)'s calculations of Case B hydrogenic emissivity. The user enters the charge (1 for hydrogen, 2 for helium), temperature, density, and upper and lower quantum numbers, and the code returns the interpolated line emissivity. A null line exits the driver.

15.10.2. *drive cdLine*

The code will loop over all lines entered in the emission-line stack, and confirm that *cdLine* can find each one.

15.10.3. *drive escape probabilities*

This option examines calculations of escape probabilities. The user enters the log of a one-sided optical depth. The code queries three of the escape probability functions and then responds with the one-sided escape probabilities. The three are complete redistribution with damping wings, incomplete redistribution, and complete redistribution with only the Doppler core. A null line exits the driver.

15.10.4. *drive fread*

This command causes the code to enter a debug mode in which the free format input reader reads the input stream and prints the interpreted number. The program will request an input line and print the interpreted number until a line with the number zero is entered.

15.10.5. *drive gaunts*

This command enters a debug mode in which a driver requests a temperature and photon energy, queries the free-free gaunt factor routine, and responds with the returned free-free gaunt factor. The original version of the gaunt factor routine was described in Hummer (1988) and Jason Ferguson extended it to include the full range of energy and temperature Cloudy needs. Ryan Porter recoded and reorganized the routine for version 96.

15.10.6. drive helium

This will exercise code written by Ryan Porter, to produce energy levels and transition probabilities for helium-like ions. The results go to an output file in the local directory.

15.10.7. drive hyas

This command enters a debug mode in which a driver requests a pair of quantum numbers, and responds with the hydrogenic Einstein transition probability. It queries the routine *EinstA*, written by Jason Ferguson.

15.10.8. drive pointers

This command allows the user to interrogate the photon-energy array. After the continuum is generated the driver will ask for a photon energy in Rydbergs (interpreted as a log if negative) and responds with the cell array index, frequency, cell width, boundaries, and the phase space factor $2h\nu^3/c^2$. Once complete, the calculation will continue as usual.

15.10.9. drive pumping

This queries the routine *conpmp*, which evaluates the continuum-line pumping probability at various optical depths (Ferland 1992).

15.10.10. drive starburst

The code will ask the user to enter a metallicity, and will return the abundances of the elements by interpolating on Fred Hamann's grid of starburst abundances (Hamann & Ferland 1993).

15.11. eden -2

This allows an extra component of free electrons to be added to the gas. The argument is the log of the electron density [cm^{-3}]. This command is mainly intended to test the behavior of Cloudy in the limit of very low Compton temperatures. When the color temperature $T_{color} \ll 10^4$ K the gas is almost entirely neutral and free electrons must be artificially added to test the Compton energy exchange problem in the strict TE limit. (Remember, charge conservation is a horrible thing to violate.)

Note: This command adds an extra component of electrons. The **set eden** command (page 187 below) sets the electron density. Both commands violate charge conservation.

15.12. fudge factors 4.59678 [12.3 34.5 958 ...]

The numbers appearing on the line can be communicated to any part of the code that calls routine *fudge*. That routine has a single integer argument that is an index to the array of numbers entered on this command line. A call to *fudge*(0) would return

the first number[37], and in the example given above, a call to *fudge*(1) would return the value 12.3. Up to ten numbers can be entered on the command line.

This command is used to pass numbers to temporary or trial parts of the code. Routine *fudge* is a permanent part of Cloudy and a warning is given at the end of the calculation if this function is evaluated. The code will stop if the index to the array of stored values exceeds the number of values entered. Extra numbers are simply ignored. *fudge* will return the number of parameters entered on the command line if it is called with a negative argument.

15.13. init ["c84.ini", path]

This command tells the code to read a set of commands from an ancillary initialization file. This allows frequently used commands to be stored in a single place and easily accessed. Common uses of this command are described on page 17 above.

There is no limit to the number of commands that can be in this initialization file, other than the total limit of 4000 command lines that is intrinsic to the code. Only one **init** command can appear in a single input stream.

The default name for the initialization file is *cloudy.ini*. This file will be used if no double quotes occur on the command line. Other file names can be specified by including a file name within a pair of double quotes, as in "*special.ini*". The name can be any name allowed by the operating system in use.

The code can search for the file on the path as set in *path.c* (as described on page 191 below). The code will look for the initialization file in the local directory first and then on the path if it is not there. The code will check on the path first if the keyword **path** occurs on the **init** line.

15.14. no . . .

It is possible to disable physical processes as a test of their importance. If a physical process is turned off then a flag is set to indicate that the treatment of physical processes has been disabled.

A warning will be printed at the end of the calculation as a reminder that the results of the calculation are not to be trusted. This warning will not be printed if the four-character keyword **(OK)** appears on the command line. The parenthesis is part of this keyword.

15.14.1. *no advection [H-like, He-like, ...]*

This turns off the ionization, recombination, and energy effects of advection. The keywords make it possible to turn off these terms for only certain parts of the calculation. The **metals** are any species not treated as one of the iso-electronic sequences. **Cooling** will turn off thermal effects of advection.

---

[37] In version 90 and before the array index was on the Fortran scale, so fudge(1) returned the first number. The array index is now interpreted on the C scale, so the first number is fudge(0).

15.14.2. *no Auger effect*

Turn off the Auger effect.

15.14.3. *no charge transfer*

This turns off all charge transfer interactions.

15.14.4. *no collisional ionization*

This turns off collisional ionization from ground states of all atoms and ions.

15.14.5. *no Compton effect*

This command turns off Compton heating and cooling of free electrons, and Compton recoil ionization of bound electrons. Electron scattering opacity *is not* turned off.

15.14.6. *no CTHeat*

This turns off heating due to charge transfer. Kingdon & Ferland (1998) describe the process.

15.14.7. *no diffuse line pumping*

The diffuse continuum produced by gas within the cloud is included as a general line-excitation mechanism. This turns it off and is useful as a check on the importance of the process.

15.14.8. *no FeII pumping*

This turns off H I Lα pumping of Fe II.

15.14.9. *no file buffering*

This turns off file buffering so that all of the standard output goes directly to the file. This is far slower than buffered output but insures that you will get output if the code crashes. Due to the increase in execution time, this should only be used in debugging situations.

15.14.10. *no file opacity*

The code can generate a file of stored opacities with the **compile opacity** command. This file will be used to generate the opacities in later calculations. The **no file opacity** command tells the code to ignore this opacity file even if it exists.

15.14.11. *no fine structure line optical depths*

Fine structure lines, such as the ^3P 52, 88 μm lines of [O III], can become optically thick under certain high-luminosity conditions (see, for example, Rubin 1983). They can absorb the incident continuum and be a significant heating source for photodissociation regions (Tielens and Hollenbach 1985a). Radiative transfer effects, including stimulated and maser emission, are fully treated by Cloudy for all lines. This command turns off optical depths and line transfer for fine structure lines by setting the line opacity to zero.

This command does not turn off line heating - which will then be maximized since the lines will remain optically thin. The **no induced processes** command (described

180 below) turns line pumping off for all lines and should be included to kill heating due to line absorption of a hot continuum.

The line transfer arrays are permanently injured by the **no fine structure** command. Subsequent runs with the same core load in a grid will still have the line optical depths disabled.

15.14.12. *no fine opacities*

The code uses a fine continuum and opacity grid to automatically account for line overlap. Figure 20 of Shaw et al. (2005) shows an example of this fine mesh. Multi-grid techniques are used to propagate this fine mesh back onto the coarse mesh that is used to evaluate photo rates. This fine grid can be turned off using this command.

15.14.13. *no free free*

Free-free heating and cooling are turned off with this command.

15.14.14. *no grain [process]*

no grain electrons This turns off the effects of grain charges, either negative or positive, on the overall charge balance in the gas. This violates charge conservation.

no grain gas collisional energy exchange. Collisional energy exchange between grains and the gas is treated as in van Hoof et al. (2004) and Abel et al. (2005). This tries to bring the grains and gas to the same temperature, which will occur if the process dominates. Often the gas is hotter than the grains and the process heats the grains and cools the gas. This command disables this physics.

no grain molecules Condensation of molecules onto cold grain surfaces was added by Nick Abel and is treated using the formalism outlined by Hasegawa et al. (1992) and Hasegawa & Herbst (1993). This turns off that process.

no grain neutralization Ions can become neutralized or ionized, by several stages of ionization, following a collision with a grain. Van Hoof et al. (2004) describe the treatment of these collisions, which is partially based on Draine & Sutin (1987)'s description of Coulomb interactions between the grains and gas. The effects of grain neutralization upon the ionization balance of the gas are turned off with this command, but the influence on the grain charge is still computed. This is not self consistent.

no grain physics This turns off all grain physics. The only remaining effect of grains is their continuous opacity. The resulting simulation will be much faster, but incorrect. This is only intended to provide a method to do quick and dirty calculations. The spectrum emitted by the grains is not calculated, so this command violates energy conservation.

no grain qheat This turns off quantum heating for all grain species.

15.14.15. *no induced processes*

This turns off induced recombination and stimulated emission for hydrogen and helium, and continuum fluorescent excitation of all lines.

15.14.16. no ionization reevaluation

This tells the code to not constantly reevaluate the ionization. This option can only be set when constant density is assumed.

15.14.17. no line transfer

This turns off line transfer for all lines except the $L\alpha$ transitions of the species treated with the iso-sequences. This is the approximation made in most codes designed to consider classical nebulae, and may be appropriate in these cases. This command also turns off line radiation pressure.

15.14.18. no Lya 21cm pumping

Pumping of the 21 cm line by $L\alpha$ is treated as in Deguchi & Watson (1985). This command turns off the process. All three keywords must match.

15.14.19. no OTS [options]

no Lya OTS This turns off H I $L\alpha$ OTS rates.

no HeII OTS This turns off He II $L\alpha$ and recombination continua OTS rates.

no line OTS This turns of all line OTS rates.

15.14.20. no level2 lines

This turns off the large block of Opacity Project ("level 2") lines. This will cut the execution time by roughly 30% and is appropriate for solar abundances and densities below 10^{10} cm^{-3}. These lines should not be disabled at higher densities since they may carry a large fraction of the cooling as the cloud approaches the black body limit.

15.14.21. no molecules[H2. CO]

Cloudy does a molecule formation network that is close to today's state of the art. It was initially based on Black (1978), Hollenbach and McKee (1979; 1989), and Tielens and Hollenbach 1985a (see the section on molecules in Part 2, and Ferland, Fabian, and Johnstone 1994; 2002; and Abel et al. 2005) but was extensively updated following the 2004 Leiden PDR workshop. It includes the hydrogen molecules H^+, H^0, H^-, H_2, H_2^+, H_3^+, HeH^+, and many heavy element molecules. The **no molecule** command turns off both the H_2 and CO parts of the network. The **no H2 molecules** or **no CO molecules** commands will turn off only the H_2 or CO parts of the network. There must be spaces to either side of the **H2** or **CO** keywords.

15.14.22. no on the spot

This turns on all ground-state recombination coefficients, and turns off ionization by helium resonance lines. Specifically, it sets all hydrogen recombination efficiencies to unity, and sets *otsmin* to 1. This last variable is then used to deduce the ionization efficiency of lines and continua. The effect of this command is to turn off such ionizations.

15.14.23. no opacity reevaluation

Opacities are normally reevaluated every time the cooling function is reevaluated. When this is set opacities are only evaluated one time per zone.

15.14.24. no outward [options]

no outward lines This turns off the outward-only (Tarter 1967) contribution by lines. It is meant as a means to debug or test the code.

no outward continuum This turns off the outward-only contribution by all continua. It is meant as a means to debug or test the code.

15.14.25. no photoionization

This turns off photoionization of the ground states of all elements. It is designed to test the code against collisional-ionization equilibrium simulations.

Several iso-sequences are treated as full multi-level atoms, including all processes that affect internal excitations. The level-populations solver may have difficulty in finding a solution when photoionization is turned off with this command but the gas is very weakly collisionally ionized. Often internal continuum pumping of the Lyman lines will dominate excitations. When these radiative terms overwhelm the collisional-ionization term the ionization rates out of the ground state may underflow. In this case the solver will find negative level populations and the code will stop announcing a catastrophic failure. One solution is to turn off continuum pumping, with the **no induced processes** command, described on page 180 above.

15.14.26. no radiation pressure

This command turns radiation pressure completely off. Radiation pressure due to trapped lines will be counted in the total pressure when the **constant pressure** option is used. The default is for a constant-density model. Radiation pressure is not included if constant gas pressure is specified.

15.14.27. no recoil ionization

This command turns off Compton recoil ionization of hydrogen, helium, and the heavy elements. Compton heating and cooling of free electrons is included, but this is the only thermal effect of electron scattering that remains. Bound-electron scattering opacity is still included when this command is issued.

15.14.28. no scattering escape

This turns of electron scattering as an escape route for line photons. By default this is included for all lines.

15.14.29. no scattering opacity

See page 100 above.

15.14.30. no secondary ionizations

This command will turn off the effects of knock-on supra-thermal electrons. Normally these are treated as in Spitzer and Tomasko (1968), Bergeron and Collin-

Souffrin (1971), Shull (1979), Shull and van Steenberg (1985), Xu and McCray (1991), and Dalgarno, Yan, & Liu (1999). This command will make X-rays 100% effective in generating heat and produce no secondary ionizations or Lα excitations.

15.14.31. no Stark broadening

Stark broadening (important for densities larger than ~10^{10} cm^{-3}) is treated for hydrogen lines using the formalism described by Puetter (1981). This turns Stark broadening off.

15.14.32. no tepredictor

The code tries to predict the temperature of the next zone for constant density calculations. This stops the predictions from being used.

15.14.33. no static opacities

This forces all opacities to be reevaluated constantly within each zone. The default is to only evaluate minor opacities one time per zone.

15.14.34. no three body recombination

This turns off three-body recombination for the heavy elements that are treated as equivalent two-level atoms. Three-body recombination for atoms and ions of the H-like and He-like isoelectronic sequences is turned off by turning off collisional ionization (its time reversal) with the **atom H-like collisional ionization off** command (page 88 above) or the equivalent **atom He-like** command (page 92 above).

15.14.35. no times

This stops the code from printing execution times. Turning this off will allow exact text comparisons of results between models run at different times or on different machines.

15.14.36. no UTA ionization

This turns off ionization by absorption of inner shell lines, which is treated as described by Behar et al. (2001) and Behar & Netzer (2002) .

15.14.37. no vary

This command turns off the **vary** option set on various optimization commands. For a further discussion see page 164 above, where the optimization driver is discussed in more detail.

15.15. set commands

These are a series of commands that change internal variables used by Cloudy. These are not used in most circumstances since the default value should suffice.

15.15.1. Convergence criteria

Several commands affect the convergence criteria. The error in the heating cooling balance is set with the **set temperature convergence** command (page 195 below). The error in the electron density is set with the **set eden convergence**

command (page 187 below). The error in the local pressure is set with the **set pressure convergence** command (page 192 below). In cases where the code is having trouble converging, the **set presioniz** command (page 192 below) will limit the number of ionization evaluations before declaring a failure.

15.15.2. set 12C13C -3.2

The chemistry network does not yet self-consistently determine $^{12}CO/^{13}CO$ ratios, although this is a moderately high priority. This command sets the $^{12}C/^{13}C$ abundance ratio. The default is 30. If the number is negative, or the keyword **_log** appears, then the number is interpreted as the log of the $^{12}C/^{13}C$ ratio, and as the linear ratio otherwise. The code currently predicts the ^{13}CO rotation spectrum and the intensity of ^{13}C III $\lambda 1910$ (Clegg et al. 1997) using this ratio. The code does not now solve for ^{12}C and ^{13}CO abundances and ionic fractions, but rather assumes that the ratio $^{12}CO/^{13}CO$ and the $^{12}C^{+2}/^{13}C^{+2}$ ionization fractions are equal to $^{12}C/^{13}C$.

The $^{12}C/^{13}C$ ratio depends on both location in the galaxy and age. It is ≈ 90 in the solar system (Anders & Grevesse 1989) and reflects the galactic value when the sun formed. The value toward the Galactic center is about 20 increasing to ≈ 77 at the solar radius (Wilson & Rood 1994). The default value is a compromise.

15.15.3. set assert fpe

This will cause the code to thrown an FPE when an assert is thrown. On systems where a modern IDE is not available, and it is not possible to set a breakpoint within the assert routine, the command provides a way to get a core file to trace back with gdb.

15.15.4. set charge transfer -11.5

This command establishes the coefficient in the statistical hydrogen charge transfer rate coefficients used for species more than four times ionized (Ferland et al. 1997). If the number is negative then it is assumed to be the log of the coefficient, if zero then this estimate is turned off, and if positive the number is the coefficient itself. It is stored as the variable *HCTMin* and has the default of 1.92×10^{-9} cm^3 s^{-1}. This is used to set a rate coefficient of *HCTMin* $\times q$ where q is the excess charge of the heavy element.

15.15.5. set chemistry

This changes some aspects of the chemistry network.

set chemistry Federman ON , OFF This determines whether the rate coefficients derived by Federman & Zsargo (2003) are used. The default is to use these rates. If **OFF** is specified then UMIST rates are used instead.

15.15.6. set continuum options

set continuum resolution 0.1 This changes the resolution of the continuum energy mesh by a constant scale factor. The resolution of the continuum mesh as a function of energy is defined by the data file *continuum_mesh.dat*, which is located in the data directory. Permanent changes to the continuum resolution should be made there. This command allows the continuum mesh to be changed by a temporary scale

factor. The number on the command line multiplies the resolution contained in the data file. Factors less then unity make the cells smaller, for higher resolution, while factors greater than unity make the resolution coarser. For instance, an entry of 0.1 would make the resolution ten times finer, or a resolving power ten times greater. If the number is less than zero it is interpreted as the log of the resolution scale factor.

This was originally added so that the entire continuum resolution could be improved by large factors when running some of the fundamental tests of hydrogenic and helium-like emission. The code's execution speed is largely set by the number of continuum cells due to the frequent reevaluations of the opacity and photo-interaction rates. For factors near unity the execution time will scale roughly inversely linearly.

The code can be sped up by making the factor larger than unity. But there is a limit, roughly 5, where the continuum cells become so coarse that the code will no longer pass its internal sanity checks. A factor near 2 will cut the execution time by about 40%. This will have modest affects on the predictions. Factors larger than 5 will probably cause the code to abort with insanity due to the coarse continuum cell size adversely affecting certain sum rules.

set continuum shielding This changes the treatment of shielding of the incident continuum by line optical depths. There are several options. **Pesc** uses the inward looking escape probability. **Federman** is the default and uses the function given in the appendix of Federman et al. (1979). **Ferland** uses the function given by Ferland (1992). **None** turns off self-shielding.

set fine continuum This changes the logic used to determine the resolution of the fine continuum mesh. The code will normally set the resolution so that Fe lines will be resolved at the lowest temperature considered in the calculation. The command has two arguments. The name of an element must appear on the line – the atomic weight of this element will be used to determine the narrowest lines that must be resolved. The number of resolution bins over a HWHM must also appear on the line.

15.15.7. *set convergence ...*

The error in the heating cooling balance is set with the **set temperature convergence** command (page 195 below). The error in the electron density is set with the **set eden convergence** command (page 187 below). The error in the local pressure is set with the **set pressure convergence** command (page 192 below). In cases where the code is having trouble converging, the **set presioniz** command (page 192 below) will limit the number of ionization evaluations before declaring a failure.

15.15.8. *set coverage [fast]*

This limits the number of zones and iterations that will be done so that the code runs more quickly. It allows quick runs of the test suite with tools such as Purify and valgrind. If the **fast** option appears then only one and one iteration will be performed. Otherwise up to ten zones and two iterations will be done. This limit is imposed on top of whatever other limits may exist in the input stream.

15.15.9. set csupra = -12.34

This sets the secondary ionization rate due to supra-thermal electrons to the number on the line. The number is the log of the rate [s⁻¹]. The excitation rate of $L\alpha$ is assumed to be the same. This option is used to test the code in secondary-ionization dominated cases. Normally the secondary ionization rate is determined self-consistently from the X-ray continuum or the **cosmic ray** command (see page 104 above).

15.15.10. set D/H -3.2

This sets the D/H abundance ratio. The default is the primordial abundance of 1.65×10^{-5} as measured by Pettini & Bowen (2001). If the number is negative, or the keyword **_log** appears, then the number is interpreted as the log of the D/H ratio, and as the linear ratio otherwise. This mainly affects HD cooling. In the current version of the code, the molecular abundance ratio HD/H_2 is assumed to be equal to the D/H ratio.

15.15.11. set didz 0.05

The thickness of the first zone is chosen so that the largest continuous optical depth through it is one percent of the entered value. Thereafter, the zone thickness is continuously adjusted by checking that the optical depth at the maximum continuum-gas interaction energy is set to this value. The default is 0.15. If the value is less than or equal to zero, then it is interpreted as the log of the quantity, and linear if greater than zero.

15.15.12. set _dr_ 11.2

This command sets the zone thickness. There are spaces around the " **dr** ". The argument is the log of the thickness in cm, and the default is 10^{31} cm. The number is interpreted as the linear thickness is the keyword **linear** appears.

15.15.13. set drmax 11.2

This command sets the largest allowed zone thickness [cm]. The code will (silently) prevent the thickness from becoming larger than this. The argument is the log of the thickness in cm, and the default is 10^{31} cm.

15.15.14. set drmin 11.2

The number is the log of the minimum zone thickness [cm]. The code will (silently) prevent the thickness from becoming smaller than this. The default value is 10^{-20} cm.

15.15.15. set dynamics [options]

This changes some aspects of the dynamics solutions when negative velocity winds are used.

set dynamics advection length [fraction] This sets the look-back distance used for the advection terms in the initial iterations of the solutions including dynamics. Its optimum value is determined by a compromise between speed of convergence (which is improved for large lengths) and accuracy (which is improved for small

lengths). The code automatically cuts the look-back distance when good convergence is obtained, so a reasonably large value can be assumed initially.

If the keyword **fraction** occurs then the number on the line will be the fraction of the total depth of the model. It is the log of the fraction if it is negative. If **fraction** does not occur then the number is the log of the scale length in cm.

set dynamics antishock depth 16 would put an anti-shock at a depth of 10^{16} cm.

set dynamics antishock Mach 1.01 would put in an anti-shock when the mach number fell below 1.01.

set dynamics pressure mode This changes some aspects of how the pressure solver deals with the sonic point. The options are: **original**, **subsonic**, **supersonic**, and **strongd**. The default chooses subsonic or supersonic based on a comparison of the ram pressure and gas pressure at the face. **Original** mode uses similar criteria to choose in each zone, but is liable to oscillation. **Subsonic** and **supersonic** will force a globally subsonic or supersonic solution. **Original** and **strongd** can pass through a sonic point. **Strongd** does this by following the supersonic branch until there is no valid pressure solution, then following the minimum of the pressure-density relation until it can leave along the subsonic branch. By varying the face conditions, the depth of the internal region of unphysical solution values can be minimized, to converge on a physical strong D type solution.

set dynamics shock depth. The argument is the log of the shock depth in cm.

set dynamics first xx. The argument is the first iteration to include dynamics. The initial pass through the layer is the first iteration. The default is to include dynamics on the second iteration, when the previous structure is known.

15.15.16. *set eden [convergence, solver, value]*

These commands affect the electron density, or how it is determined.

set eden convergence 0.01 This command sets the convergence criterion for the electron density. The number is the largest relative error in the electron density. The number on the line is interpreted as the log of the relative error if it is negative. The default value is 0.01. The old form of the command, **set eden error**, is also recognized.

set eden solver new This command tells the code which solver to use to determine the electron density. There are currently two options, the default simple method, and a new method that is still under development. This command is currently under development and should not be used except under experimental conditions.

set eden 2.3 If no other keywords are recognized then this command sets the electron density. The number is the log of the density [cm^{-3}].

Note: The **eden** command (page 177 above) sets an extra source of electrons. This command sets the electron density itself. Both commands violate charge conservation.

15.15.17. set flxfnt -20

The highest continuum energy that must be considered is lower for relatively soft continua such as a star, than for X-ray sources, such as AGNs. The criterion used to choose the highest energy to be considered ν_{high} is such that $\nu f_\nu(\nu_{high}) / \nu f_\nu(\nu_{peak}) <$ *flxfnt*, where ν_{peak} is the frequency where the continuum reaches its maximum νf_ν. *FluxFaint* is normally 10^{-10}. This command changes the value of *FluxFaint*. The argument is the log of the value.

15.15.18. set H2 ...

This command changes some aspects of the H_2 molecule. Other aspects of this molecule are changed with the **atom H2** command, described on page 85 above.

set H2 fraction This sets the ratio $n(H_2)/n(H)$. The number is interpreted as the log of the ratio if it is less then or equal to zero, and as the linear ratio otherwise. An upper limit to the ratio of 0.5 will be silently imposed. This command turns off the normal H_2 molecular network and forces the H_2 density to the hydrogen density multiplied by this fraction. This is totally unphysical and is only intended to provide an easy way to test the H_2 molecule.

set H2 Jura [TH85; CT02; SN99] This changes "Jura rate", the rate H_2 forms on grains. The default is the Cazaux & Tielens (2002) rates, and this can also be specified with the **CT02** option. **TH85** will use rates from Tielens & Hollenbach (1985a), and **SN99** will use rates for Sternberg & Neufeld (1999).

set H2 Jura scale xx The number on the line is a scale factor that multiplies the formation rate obtained by one of the above methods. If the scale factor is negative or the keyword **_log** appears then the number is interpreted as the log of the scale factor. Habart et al. (2003) suggest that the rate can vary across the galaxy by large factors.

set H2 Jura rate xx The number on the line is the log of the H_2 grain formation rate k [cm³ s⁻¹]. This is used to give a simple grain formation rate $R = k\, n\left(H_{tot}\right) n\left(H^0\right)$ [cm⁻³ s⁻¹].

set H2 Solomon [BD96; TH85; ELWERT] This changes how the Solomon process is treated when the simple H_2 molecule is used. The fits to the dissociation rate derived by Elwert et al. (2006) are the default and will use used if the keyword **ELWERT** appears. The keyword **BD96** will use Bertoldi & Draine (1996) and the keyword **TH85** will use the Tielens & Hollenbach (1985a) rate. These rates differ by ~1 dex for some H_2 column densities. When the large H_2 molecule is used (see the **atom H2** command described on page 85 above) the Solomon process is treated in a self-consistent manner. Abel et al. (2004) show that H_2 column densities are quite sensitive to the correct treatment of this rate.

set H2 grain formation pumping This changes the distribution function for populations of v, J levels of H_2. There are three possible distributions: Takahashi (2001; **taka**); Draine & Bertoldi 1996; **DB96**); and thermal (**Thermal**).

set H2 Tad 420 This command changes the binding energy, T_{ad}, in the formalism described by Le Bourlot (2000) to account for deexcitation of H_2 while on a grain

surface. The number is the binding energy expressed as a temperature. The default value is 800 K, which has the effect of maximizing the rates of H_2 collisional deexcitation and ortho-para conversion. If the number is ≤ 10 it is interpreted as a log of the temperature unless the keyword `linear` appears.

15.15.19. set HCORR xx

The density used in electronic collisions is actually $n_e + 1.7 \times 10^{-4} n\left(H^0\right) f$. The second term is an approximate correction for collisions by neutral hydrogen. This is only a guestamite based on (Drawin 1969, see also Weisheit 1975, Steenbock & Holweger 1984, Lambert 1993, and Kiselman 2000). The number on this command adjusts the scale factor f. You can turn off the process by setting the number to zero.

15.15.20. set HO charge transfer [chemistry; ionization]

This changes the treatment of hydrogen – oxygen charge transfer. Charge transfer rates between atoms and the first ion of H and O is very fast and strongly couples their ionization (AGN3 Section 2.7). The process is normally treated as part of the chemistry network. The `ionization` option will include the rates within the ionization network instead. The chemistry and ionization solvers communicate and the result should not depend on which way the process is included. The strongly disturbed "Leiden hack" PDR models in the test suite require the `ionization` option to function properly.

15.15.21. set ind2 [_on_, _off]

This command provides a way to turn off the affects of induced two-photon emission or absorption. One of the keywords `_on_` or `_off` must appear to turn induced two-photon processes on or off.

15.15.22. set kshell energy 1,000,000

This command is used to change the energy of the highest continuum point considered for photoelectric opacity. The default is 1 MeV, sufficiently high that Compton recoil and/or pair production are the dominant opacity sources, and K-shell opacity may safely be ignored. Setting this limit to smaller values will save some compute time since the evaluation of the photoionization rate integrals will not extend to as high an energy. The argument is the energy in Rydbergs, and it must be greater than 194 Ryd. If zero is entered then the high-energy limit of the continuum will be used.

15.15.23. set Leiden hack

This replaces certain physical processes with the simple prescriptions that were specified for the 2004 Leiden PDR workshop. These commands are only intended for computing these simple tests and must not be used in a real calculation. If any `set Leiden hack` commands are entered the code will complain that a physical process has been disabled. The hacks are:

set Leiden hack H2 off* The chemistry network used in the workshop did not include H_2^* (see Tielens & Hollenbach 1985a for a definition). This turns off reactions

between that species and the CO network (although it is still present in the H-H$_2$ network).

set Leiden hack CR off. This turns off cosmic ray excitation for the complete H$_2$ molecule. Others at the workshop did not include this important excitation process.

set Leiden hack UMIST This forces the code to use parts of the UMIST database that are incorrect.

15.15.24. set line precision [4 or 5]

This command changes how the emission lines are printed in the final pages of output. Emission-line wavelengths are normally printed with four digits, as in "4861". This command can increase the number of digits to 5. The lines are printed in four columns when four digits are chosen. With five digits the lines will no longer fit across the page so the number of columns is automatically reduced to three.

15.15.25. set nchrg 3

The grain physics uses an *n*-charge state model where, for each bin, the charge distribution is resolved in exactly *n* charge states, independent of grain size. This is discussed in van Hoof et al. (2004). The default value of *n* is 2, but can be reset with this command to any value between 2 and 5. Higher values of *n* will give more accurate results at the expense of greater computing time.

15.15.26. set negopc

Negative opacities may occur during a calculation if a level happens to mase (Ferland 1993). The code will generate a comment at the end if this happens. This command tells the code to punch the optical depth array when negative opacities occur. The output will go to the file *negopc.txt*.

15.15.27. set nend 500

This sets the default limit to the number of zones that will be computed. The preset default value is 1400, but more zones may be needed in large column density models, or ones exposed to very intense radiation fields.

The limit to the number of zones that will be computed can be set with either this command or with the `stop zone` command (page 119 above). The only difference is in the level of warning that will be generated if the code stops after reaching the limiting number of zones. If the code stops because it reached the number of zones set by the `stop zone` command then it thinks that this was the intended stopping criterion and no comment is generated. However the code generates a *warning* if it stops because it reaches the default limit to the number of zones since this probably *was not* intended. By using this command the limit can be increased while still retaining the checking and warnings generated if the code stops for an unintended reason.

15.15.28. set nFnu [options]

The continuum emitted by the central object and transmitted through the cloud, and the diffuse continuum produced by the cloud, will be printed within the

emission-line printout if the command **print continuum** (page 123 above) is entered. The diffuse continua will be printed with the label *nFnu* and the transmitted incident continua are printed with the label *nInu*.

The **set nFnu** command changes which continua are included in the entry with the *nFnu* label. There are four possible continua, the transmitted and reflected incident continuum, and the transmitted and reflected diffuse continuum. Any or all of these may be entered, but at least one must occur. All four continua are set when the **set nFnu** command is parsed. Any component left off the command will not be included.

If this command is entered then **print continuum** does not also need to be given – the continuum points will occur in the printout. This command does not change the quantities printed with the *nInu* label – these are always the sum of the reflected plus transmitted incident continuum.

The four keywords, and the component that is included, are the following:

incident reflected or *incident_reflected* includes the incident reflected continuum.

incident transmitted or *incident_transmitted* includes the incident transmitted continuum.

diffuse inward or *diffuse_inward* includes the diffuse inward continuum.

diffuse outward or *diffuse_outward* includes the diffuse outward continuum.

All continua must be specified on the same **set** command, however. The following shows how to only include the outward incident and outward diffuse emission in the *nFnu* entry in the printout:

```
set nFnu diffuse outward, incident transmitted
```

15.15.29. set nmaps 50

This controls the number of steps in the heating-cooling map that results from either the **map** or **punch map** commands. Normally about 20 steps are taken between the lowest and highest temperatures.

15.15.30. set numerical derivatives

This tells the code to use numerical rather than analytic derivatives of the heating and cooling functions. The default is to use the analytic derivatives.

15.15.31. set PAH option

This changes the function that describes how PAH abundances depend on physical conditions. The default, which can also be specified with the keyword **H0**, is for the abundance to scale with $n(H^0) / n(H_{tot})$. The abundance will be constant if the keyword **CONstant** appears. This is not physical and is only for testing.

15.15.32. set path ="/usr/home/cloudy"

This command sets the path Cloudy will use to look for various data files. The current directory will be used if the path is not set either with this command or by setting the path in *path.c*. The path is within a pair of double quotes. If the last

character in the path name is either the "]", "\", or "/" character then the path is used as it is entered to find files (this will work for a VMS or Unix machine). If the last character is anything other than the "/" or "\" characters, one of these will be concatenated.

It is generally more convenient to hardwire the path into the code by editing the path set in the file *path.c*. The path will never need to be set. To do this examine the file *path.c* and follow its instructions.

15.15.33. set phfit [1995, 1996]

The key **1995** tells the code to use photoionization cross sections from Verner & Yakovlev (1995). The key **1996** is the default and tells the code to use Verner et al. (1996), which is partially based on Opacity Project cross sections.

15.15.34. set presioniz 50

This sets a limit to the number of times the bottom ionization solver will be called from the top pressure solver within an individual zone. This is a debugging aid and is only needed to shorten the number of convergence attempts within a zone that is destined to have a convergence failure.

15.15.35. set pressure convergence 0.01

This sets the convergence criterion for the total pressure. The number is the largest relative error in the pressure. The number is interpreted as the linear error if it is positive and the log of this error if negative. The default value is 0.01.

15.15.36. set punch commands

Punch output, described in the section beginning on page 129 above, is the primary output mechanism for the code. This command sets various details about how this output is generated.

set punch hash "newstring" When more than one iteration is punched into a file each ends with a series of hash marks to locate with an editor. This command sets the hash string to something else. A new string must appear within the pair of double quotes.

The special string "**return**" is replaced with a single carriage return, which translates into the pair of carriage returns that *gnuplot* uses to separate plots.

The hash can be completely turned off with the **no hash** option on the punch command (see page 130 above).

set punch prefix "test" This command adds a prefix to all filenames and so provides an easy way to change all file names at one time. The prefix is any string that appears between the pair of double quotes on the command line.

set punch LWidth 10,000 km/sec The observed contrast between emission lines and the continuum depends on the intrinsic line width and, for an unresolved line, the resolution of the spectrometer. Lines are included in the continuum produced by the **punch continuum** commands (see page 131 above). This command adjusts the contrast between the lines and continuum.

Lines and continua are stored separately throughout the code. They are combined only when the output from the **punch continuum** command is produced using the expression

$$\nu F_\nu(\text{total}) = \nu F_\nu(\text{continuum}) + \frac{c}{PunchLWidth} I(\text{line}) \tag{71}$$

where c is the speed of light. Values are entered in km s^{-1} and the default is 1000 km s^{-1}.

If no number appears on the line, but the keyword **_c_** does, the speed of light will be used. Note the space before and after the 'c'.

The intensities of the emission lines in the punch files, defined by subtracting the intensity in the continuum from the sum of the line and continuum, will be correct if the line width is set equal to the speed of light. The lines will have too small a contrast in that case, however, unless the spectrometer has a resolving power of unity. If the line width is smaller than the speed of light then the line to continuum contrast will be greater but the summed intensity of the line plus continuum in the punch output will be greater than the actual radiated power.

The only effect of this command is to change the line to continuum contrast in output from all **punch** commands except for that produced by the **punch transmitted continuum** command. Turbulent velocities are set with the **turbulence** command.

15.15.37. *set dielectronic recombination* **keywords**

This command modifies the treatment of the two contributors of dielectronic recombination to the total rate coefficient. The default condition is for the guestamates of third row and forth row elements to be used, the Burgess process to be suppressed at high densities, but the Nussbaumer and Storey process to be 100% efficient at all densities.

set dielectronic recombination kludge [_on_, _off] The **kludge** option modifies the treatment of the guestamates of dielectronic recombination coefficients presented by Ali et al. (1991).

At present rate coefficients for dielectronic recombination through low-lying autoionizing states have not been computed for most elements on the third row and higher. The code uses the means of the rate coefficients for the four lowest stages of ionization of C, N, and O. These are $3\pm2\times10^{-13}$, $3\pm2\times10^{-12}$, $1.5\pm0.5\times10^{-11}$, and $2.5\pm1.4\times10^{-11}$ cm^3 s^{-1} (see Ali et al 1991). The uncertainty is indicated by the quoted uncertainty, which is the dispersion from the mean of the quoted atoms and ions. These are used for those ions that have no rate coefficients because no better can be done at present.

These rate coefficients can be turned off, or the values changed by a scale factor, with this command. If a number is entered on the line then it is the scale factor to multiply all the above rate coefficients. If no number is entered then zero is assumed and the effect is turned off. If more than one number is entered then each is used as a

scale factor for successive stages of ionization. If fewer than 4 numbers are entered, then the last entered scale factor is used for higher stages of ionization.

This provides a way to check on the importance of this recombination mechanism for specific simulations.

set dielectronic recombination Burgess [_on_, _off] Rate coefficients for recombination at high temperatures are dominated by the Burgess (1965) process, which occurs though a pseudo-continuum of autoionizing levels. This process may be suppressed at high densities (Burgess and Summers 1969, Davidson 1975). Fits to Davidson's results are used if **Burgess on** appears. The process is assumed to be 100% efficient if **Burgess off** appears. The default is for suppression to be included.

set dielectronic recombination Nussbaumer [_on_, _off] Rate coefficients for recombination at nebular temperatures (i.e., where *kT* is much smaller than the ionization potential of the species) are dominated by a few autoionizing levels lying just above the ionization threshold (Nussbaumer and Storey 1983). This process may be suppressed at high densities (Davidson 1975), but probably is not. Fits to Davidson's results are used if **Nussbaumer on** appears. The process is assumed to be 100% efficient if **Nussbaumer off** appears. The default is for collisional suppression *not* to affect this process.

set dielectronic recombination [Badnell; old] This changes which set of dielectronic recombination rate coefficients are used. The **Badnell** option will tell the code to use the rate coefficients calculated by Badnell et al. in a series of papers that appeared in A&A. These supersede the various approximations described above. They will be used if the **_old** option appears.

15.15.37.1. set spectrum [options]

This modifies the behavior of the **punch spectrum** command (see page 137 above). A series of **set spectrum** commands can occur, and these modify any **punch spectrum** commands that follow them. The following shows an example of changing the resolution and then producing a **contin.dat** file.

```
set spectrum resolving power 100
punch spectrum "contin.dat"
```

The following are the options now recognized by the **set spectrum** command.

set spectrum range 1 to 5 Ryd This sets the energy range of the continuum that is predicted. The default, when this is not used, is to produce the full continuum. The energies are given in Rydbergs and (for now) and other units are not recognized.

set spectrum revolving power 1000 This sets the resolving power of the continuum that is predicted. The default, when this is not given, is to use the native resolution of the code.

15.15.37.2. set temperature [floor, solver, convergence]

set temperature floor This sets a lowest temperature to allow in a calculation. When the electron temperature falls below this floor, the code goes over to a constant

temperature solution at the floor temperature. This provides a way to mimic having a minimum temperature that is set by some external and unspecified source of heat.

set temperature solver This tells the code which solver to use for the heating – cooling solution. The options are **brent** and **simple**. (**brent** is currently under development and should be used only for testing).

set temperature convergence The balance between the heating and cooling rates sets the electron temperature. This command changes the error in the convergence tolerance allowed in the heating-cooling match. The number is the largest fractional error allowed, and is interpreted as the error itself if it is positive, and the log of the error if it is less than or equal to zero. The default is a fractional error of 0.005. This will be the error in the heating-cooling balance allowed in each zone. The total error or energy conservation mismatch integrated over the model will be much smaller, usually of order ten times smaller than the tolerance specified.

15.15.37.3. *set test*

This sets the logical variable *lgTestOn* to true. It provides the facility to conditionally run test code somewhere in the main body of Cloudy.

15.15.37.4. *set trimming –9 [upper lower, off]*

The code saves execution time by not computing ionization equilibria for stages of ionization with trivial abundances. The thresholds for excluding an ionization stage are chosen with photoionization equilibrium in mind. These may not be appropriate for some other conditions or it for some reason your definition of trivial is different from mine.

This command changes the limit. The smallest relative abundance to be considered for a stage of ionization higher than the ionization peak is changed with the **upper** keyword. The combination **set trimming upper off** will turn off trimming of the highest stage of ionization. The smallest relative abundance of ions below the peak is changed with the **lower** keyword. The default relative ionization fractions are 10^{-6} and 10^{-10} respectively. If no keyword appears then both are changed to the number entered. The argument is the log of the lowest or highest ionization fraction to consider.

Generally, line excitation energies for atoms with stages of ionization higher than the peak ionization will strongly exceed the ambient temperature, so these ions will have little influence on the calculated temperature or spectrum. This is not true for lines formed from ions below the peak distribution. For convenience the keyword **off** can be used to set the smallest abundance to just above the machine's floating precision limit.

15.15.37.5. *set tsqden 8*

The code performs an analysis of the predicted emission-line spectrum at the end of the calculation. This analysis will find the structural t^2 as well as a t^2 deduced from the [O III] and H I spectrum (Kingdon & Ferland 1995 discuss this at length). Such an analysis only makes sense for densities below the critical density of the [O III] atom, which is $\sim 10^5$ cm^{-3}. The results of this analysis will not be printed if the

density is higher than the value of *tsqden*, currently 10^7 cm^{-3}. The number on the line is the log of the highest hydrogen density for which this analysis will be performed.

15.15.37.6. set WeakHeatCool 0.02

This command sets the threshold for the weakest coolant or heat source included in output from the **punch heating** (page 141 above) or **punch cooling** (page 137 above) commands. The default is 0.05. The number entered is interpreted as a log if it is negative.

15.16. table lines "name.dat"

When the code is used as a subprogram for other, larger, code, it is possible to read in a series of lines whose intensity will later be extracted by first calling *cdGetLineList*, then by calling *cdLine*. The rpocedure is described in Part 3 of this document. A set of lines that occur within any of several data files is then read. This method is used in most of the large grid programs I use.

This command provides a quick way to confirm that the list of lines in the line data files have valid names. The command takes the name of a file as its argument and provides a quick way to check that the list within it is valid. See the discussion of *cdGetLineList* in Part 3 for further information.

15.17. test

This command does a "smoke test" and provides an easy way to do a quick sanity check on the code. It issues a series of commands to run a two-zone constant-temperature model. The **table agn** command is used with the density and temperature constant at 10^4. The ionization parameter is 10^{-2} and the calculation stops after two zones and one iteration. Many **assert** commands are used to verify predictions. This is a debugging aid designed to make quick tests of the code.

15.17.1. test FeII

When the **FeII** keyword appears the large Fe II atom (page 83 above) will be computed and checked as well.

15.17.2. test large

When the **large** keyword appears the model hydrogen atom (page 88 above) will be set to a large number of levels.

15.17.3. test h2

This tests the large H_2 molecule.

15.18. assert commands

These tell the code a set of results that are expected. At the end of the calculation the expected and actual results are compared and an error condition is returned if the two disagree by more than an uncertainty. This set of commands is the foundation for automated testing of the code. In Lexington the full test suite is computed every night and a log is kept of all asserted results. An email warning is sent if any asserts

fail. This insures the reliability of the code and guarantees that bugs are caught almost as soon as they are introduced.

If the characters "<" or ">" appear on the command line the expected result is taken as an upper or lower limit. The default check is for equality.

The relative error is the second optional number on the line. If this is not specified then a relative error of 0.05 will be assumed.

15.18.1. assert column [_CO_, H2 …] 18.9

This checks the predicted column density for CO, H^-, H_2, H_2^+, H_3^+, H_2^g, H_2^*, HeH^+, O_2, SiO, or OH. The labels must be surrounded by spaces. It is important that the log of the column density [cm^{-2}] appear *after* the label because of the 2 in **H2** or **O2**.

The command accepts a **log** option to interpret the error as a log. The column density itself is always interpreted as a log.

For H_2 there is a further option, **levels**, which tells the code to return the column density in a specific v, J level. This only works when the large H_2 molecule is included. The numbers on the line must appear in the following order: a) the 2 in **H2**, b) v, c) J, and d) log $N(H_2)$. The "wavelength" of the asserted quantity will be $100v + J$.

15.18.2. assert csupra –17.09

The secondary ionization rate is given by the variable *csupra*. This command checks the value of the secondary ionization rate. The number is the log of the secondary ionization rate [s^{-1}].

15.18.3. assert departure coefficients mean=1, error=0.04

This confirms that the departure coefficients predicted by one of the large model atoms are correct. One of the following keywords must appear. The computed quantity is the mean departure coefficient for all levels in the model atom and is the first number on the line. The type of error represented by the second number depends on specifics.

assert departure coefficients, H-like helium, error=0.08 The keyword **H-like** tells the code that the departure coefficients predicted by an atom of the hydrogenic isoelectronic sequence should be checked. The name of one of the elements must also appear on the command line.

The error will be the largest single deviation from unity of any departure coefficient, and can be changed with the second parameter on the command line. If the keyword **excited** appears then the ground state will not be included.

assert departure coefficients, He-like helium, error=0.08 This works like the **H-like** option, but for the helium-like isoelectronic sequence.

assert departure coefficients, FeII, error=0.08 The keyword **FeII** tells the code to confirm that departure coefficients predicted by the large Fe II atom are correct. In this case the error is the standard deviation of the departure coefficients.

assert departure coefficient, hminus, error=0.08 The keyword `hminus` tells the code to confirm that the H^- departure coefficient is unity. The error is the deviation of the departure coefficient from unity.

15.18.4. assert depth 13.2

This checks that the cloud's thickness [cm] is equal to the expected value. The argument is always the log of the depth. The thickness is the length between the illuminated and shielded faces of the cloud.

15.18.5. assert eden 9

The number is the log of the electron density [cm^{-3}] of the last zone.

15.18.6. assert H2 grain rate -16.5

The number is the total rate [cm^{-3} s^{-1}] that H_2 forms on grain surfaces for the last computed zone. The number is a log if it is negative.

15.18.7. assert H2 ortho/para ratio 2.02

This checks the ratio of the ortho to para H_2 densities in the last computed zone. This is only meaningful if the large H_2 molecule has been turned on.

15.18.8. assert htot –13.2

This checks that the local heating rate [erg cm^{-3} s^{-1}] of the last zone is equal to the expected value. The argument is the log of the heating rate.

15.18.9. assert hheicf –0.013

This checks the helium – hydrogen ionization correction factor. The number is the linear difference between the atomic fractions of helium and hydrogen.

15.18.10. assert ionization fraction oxygen 3 –3.45, error 0.1, weight = radius

This checks the mean ionization of an element. An element name must appear somewhere on the line. The first number is the ionization stage, 1 for the atom, 2 for the first ion, etc. The second number is the expected ionization fraction. It is interpreted as a log if negative and as the linear ionization fraction if it is positive. The average can be with respect to radius or volume. The default is an average weighted with radius, but if the keyword `volume` appears then the average will be weighted over volume.

15.18.11. assert molecular fraction H2 –3.45, error 0.1

This checks the mean molecular fraction. The label for a molecule, currently only H_2, must appear somewhere on the line. The next number is the expected molecular fraction. If this number is ≤ 0 it is interpreted as a log and as the linear ionization fraction if it is positive. The average can be with respect to radius or volume. The default is an average weighted with radius, but if the keyword `volume` appears then the average will be weighted over volume.

15.18.12. assert itrzn <3.5

This checks the convergence properties of a calculation.   The quantity is the number of iterations required to converge each zone. This is usually an upper limit.

15.18.13. assert line "TOTL" 4861 < 1.01

This checks the relative intensities of a line. The string that gives the line label must appear between two double quotes and it must be four characters long. The line wavelength is the first number after the label. Both label and wavelength must appear exactly as they do in the output produced by the **punch line labels** command or in the large emission-line output.

The sub-keyword is actually "**ine_**" to avoid confusion with the keyword "**linear**" that appears on some commands. The command will not be recognized if the trailing space is missing.

The second number is the expected intensity relative to the normalization line. It is always the linear intensity relative to whatever normalization line is set for the model.

15.18.14. assert line [luminosity intensity] "Q(H)" 4861 38.91

This compares the predicted and expected luminosities or intensities of any line. This command functions exactly as the **assert line** command, except that the quantity is the log of the luminosity or intensity of the transition and not the intensity relative to a normalization line.

15.18.15. assert niter < 4

This checks the number of iterations required in a calculation.   This is usually an upper limit.

15.18.16. assert nzone < 135

This checks how many zones were needed. This is usually an upper limit.

15.18.17. assert pressure error < 0.01

The number is the ratio of the standard deviation of the pressure to its mean value. It is interpreted as a log if it is negative.

15.18.18. assert pradmax 0.34

The asserted quantity is the maximum ratio of radiation to gas pressure.

15.18.19. assert temperature hydrogen 2, 4.02 volume, error 0.01

This checks the mean temperature of any species. The name of one of the elements or the keywords **grains** and **H2** are recognized. When the name of an element appears then the code will compare the computed mean temperature for that stage of ionization with the asserted value. The first number on the line is the stage of ionization, with 1 the atom, 2 the first ion, etc. The second number is the temperature, interpreted as a log if it is ≤ 10. It will be linear if the keyword **linear** appears. The last number is the relative error.

The code computes means weighted over radius and over volume. If the keyword **volume** appears then the temperature will be compared with the volume-weighted mean. The default is weighting over radius.

15.18.20. *assert grain temperature index 2, temperature 234.*

If no element name appears but the keyword **grains** does then the code will compare the computed and asserted grain temperature. The first number on the line is an index giving the grain type. This index is from the order in which the grains were specified in the input stream and is given in the output. The first grain that occurs in the input stream is number 1. The second number is the temperature, interpreted as a log if ≤ 10. The **linear** keyword forces smaller numbers to be interpreted as linear quantities. The optional error is the third number on the line. The temperature is always averaged over radius.

15.18.21. *assert radius 18.2*

This checks that the outer radius [cm] of the computed model is equal to the expected value. The radius is the distance from the center of the central object to the shielded face of the cloud.

15.18.22. *assert thickness 13.2*

This checks the thickness [cm] of the computed structure. The argument is always the log of the thickness. The thickness is the length between the illuminated and shielded faces.

15.18.23. *assert velocity 7.6 km/s*

This checks the final velocity of a wind model. The quantity is the expected final velocity in km/s.

15.18.24. *table lines "LineList_PDR.dat"*

This provides a way to check that the emission-line labels in any of the standard "*LineList*.dat*" files in the data directory are valid. I is actually a form of **assert** command since it will announce a botched assert if it cannot find any of the lines.

15.19. Performance speedups

It is possible to speed up the calculation at the loss of some physical fidelity or stability. This section outlines the commands that make this possible. Each command is described in greater detail in sections earlier in this document.

Many of these options are included in the *fast.ini* initialization file. Using this *ini* file will decrease the execution time by roughly an order of magnitude. This is meant as a way to explore vast ranges of parameters or as a debugging aid.

15.19.1. *Turn off minor elements*

The ionization distribution and the effects of an element on the gas cooling and opacity are all determined self-consistently. This includes the first 30 elements, and many of these have trivial abundances. The calculation can be sped up with no loss of physical fidelity if minor elements are turned off. This is done with the **element**

off command described on page 57 above. The ini files *c84.ini*, *fast.ini*, and *ism.ini*, among others, include a set of **element off** commands to do this.

Turning off an element will affect results if the opacity or cooling due to the element is significant. Minor elements can have major effect in very dense, optically thick, gas, where all coolants are close to the black body limit.

15.19.2. Turn off the chemistry

The chemical network requires a fair amount of time due to the size of the matrix that must be solved. The chemical state is determined even in ionized regions to ensure continuity in the transition between the H^+ – H^0 – H_2 regions. The chemistry can be safely ignored if only ionized regions are present. This is done with the **no molecules** command, described on page 181 above.

15.19.3. Turn off grain physics

The treatment of the grain physics is today's state of the art and can be CPU-intensive. There are several ways to speed up the calculation, although they will degrade the fidelity of the simulation. These may be used for exploratory or coarse calculations, which would then be followed by complete simulations which include all physical processes.

Grains have several major effects. Elements locked up in grains are missing from the gas phase and so their cooling is also missing. Grains provide the dominant opacity across much of the spectrum. This absorbs the incident continuum, affects the radiative acceleration of the gas, and the transfer of optically thick lines. Radiative and collisional interactions between grains and the gas affect the gas temperature. In PDR and molecular regions, grains have an important effect on the charge balance of the gas. This affects both the ionization and chemistry (Abel et al. 2005).

The first effect can be included at no expense by setting the gas-phase abundance of an element to a depleted value. Gas-phase abundances can be decreased by typical ISM depletion factors using the **metals deplete** command, described on page 67 above. This does not turn on grains, and applying a depletion factor has no impact on the execution time.

Turn off quantum heating. Quantum heating is the process where a grain's temperature spikes after absorbing a single photon. This temperature spiking has an effect on the Wien tail of the grain's emitted spectral energy distribution but no other effect. Quantum heating can be disabled (with the **no qheat** option on the **grain** command) if the thermal continuum is not something you need to predict. Quantum heating only affects the emitted spectrum so turning if off will not affect results if the emitted IR continuum does not affect the gas. The IR continuum will affect the gas if it is absorbed by the gas or dust, which can occur with some line, or if the source is optically thick in the near IR continuum.

Use single rather than distributed grains. If the **single** keyword appears on the **grain** command then a single mean grain size will take be used instead of the size–resolved grain distribution. This affects the entire simulation, including the gas

temperature and emission line spectrum, because the grain – gas interactions depend on the grain size.

Turning off all grain physics. All grain-gas interactions, except for their opacity, are turned off with the **no grain physics** command (page 180 above). Diffuse emission of the grains is no longer included, so energy is not conserved. The dynamical and opacity effects of the grains will still be included.

Grains no reevaluate. *Don't revaluate grain rates* The **no reevaluate** option on the **grains** command will speed up a calculation by not reevaluating grain quantities all the time. Roughly a 30% speedup can be achieved. This is potentially dangerous and can destabilize a solution and/or lead to wrong results. It should be used under certain experimental circumstances, and not in a true simulation of a cloud.

15.19.4. Turn off line transfer

Many lines are optically thick. If the density is high then photons can be thermalized as the result of multiple scatterings. When background opacities are present lines can also be absorbed by the continuum. In this case they act as an ionization and heating agent. The **no line transfer** command (page 181 above) turns off line transfer for all lines except the H and He Lα transitions. These assumptions are common in codes designed to treat classical nebulae, but are not appropriate for dense gas.

15.19.5. Turn off level 2 lines

This is done with the **no level2 lines** command, described on page 181 above. The level 2 lines are a large set of transitions that came from the Opacity Project. They are important in dense or very high ionization gas, but can be ignored in most conventional nebulae.

15.19.6. Don't reevaluate the ionization or opacities

The **no ionization reevaluate** command (page 181 above) tells the code not to constantly reevaluate the level of ionization during the thermal solution. This may affect the code's stability for dense or dust-free gas but should not affect results.

The **no opacity reevaluation** command (page 182 above) tells the code not to reevaluate the opacities as it searches for a thermal solution. This may affect the code's stability for dense or dust-free gas but should not affect results.

These commands are safe when grains are present, since their opacity strongly diminishes the diffuse radiation field. When grains are not present the diffuse radiation fields within the Balmer continuum or Lα can be quite intense and dominate the photoionization of some third row atoms. In this case constant reevaluation of the opacity and ionization is needed for a stable solution.

15.19.7. Turn off fine opacities.

The code uses a multi-grid approach to handle continuum interactions quickly, but also resolve lines so that line overlap can be treated in an automatic manner. There is a main continuum mesh, with a relatively coarse resolution and a high-

resolution or fine mesh to resolve all lines. This second mesh is computationally expensive and is turned off with the **no fine opacities** command (page 180 above).

Turning off the fine mesh also turns off the treatment of line overlap and will affect results when overlap is important. There are many situations, like classical nebulae, where line overlap is not important.

15.19.8. Degrade the continuum resolution

The resolution of the coarse continuum mesh affects the execution time since photoionization rates and opacities involve sums over this mesh. The resolution can be changed with the **set continuum resolution** command (page 184 above). This is not included in the *fast.ini* file since it is not possible to use the series of special incident continua that are part of the code distribution. It would be necessary to recompile these continua at the lower resolution (with the **compile stars** command, page 169 above) to use them.

This changes the continuum energy mesh. The versions of the grains ad stellar atmospheres with the code will only work for the standard continuum resolution.

15.19.9. Make the iso-sequence model atoms as small as possible

The code treats several iso-sequences with a unified model atom. The execution time increases when the number of levels within the model atoms is increased. Less time will be used if the smallest possible number of levels are used. The number of levels is adjusted with the **levels** option on the **atoms** command, described on page 83 above. The **atom levels** commands for the H-like and He-like atoms accepts the keywords **very small** to make the atom as compact as possible. The line spectrum predicted by these atoms is likely to be highly inaccurate, and Case B intensities are far better when a compact model is used.

16. REFERENCES

Abbott, D. C. 1982, ApJ, 259, 282

Abel, N. P., Brogan, C. L., Ferland, G. J., O'Dell, C. R., Shaw, G., & Troland, T. H. 2004, ApJ, 609, 247-260 (astro-ph/0404399)

Abel, N.P., Ferland, G.J., Shaw, G. & van Hoof, P.A.M. 2005, ApJS, 161, 65 (astro-ph/0506514)

Adams, T. 1972, ApJ, 174, 439

Aldrovandi, S., & Péquignot, D. 1972, A&A, 17, 88

Aldrovandi, S., & Péquignot, D. 1974, Revista Brasileira de Fisica, 4, 491

Ali, B., Blum, R. D., Bumgardner, T. E., Cranmer, S. R., Ferland, G. J., Haefner, R. I., & Tiede, G. P. 1991, PASP, 103, 1182

Allen, C. W. 1976, *Astrophysical Quantities*, Third Edition (London: Athlone Press)

Allende Prieto, C., Lambert, D.L., & Asplund, M., 2001, ApJ, 556, L63

Allende Prieto, C., Lambert, D.L., & Asplund, M., 2002, ApJ, 573, L137

Aller, L. H. 1984, in *Physics of Thermal Gaseous Nebulae*, (Dordrecht: Reidel)

Aller, L. H., & Czyzak, S. J. 1983, ApJS, 51, 211.

Anders, E., & Grevesse, N. 1989, Geochim. Cosmochim. Acta, 53, 210

Anderson, H., Ballance, C.P., Badnell, N.R., & Summers, H.P., 2000, J Phys B, 33, 1255Arimoto, N., & Yoshii, Y. 1987, A&A, 173, 23

Antonucci, Robert, 1993, ARA&A, 31, 473

Aviv, N., Barlow, T.A., Laor, A., Sargent, W.L.W., & Blandford, R.D. 1998, MNRAS 297, 990

Armour, Mary-Helen, Ballantyne, D.R., Ferland, G.J., Karr, J., & Martin, P.G., 1999, PASP 111, 1251-1257

Arnaud, M., & Raymond, J. 1992, ApJ, 398, 394

Arnaud, M., & Rothenflug, R. 1985, A&AS, 60, 425

Asplund, M., Grevesse, N., & Sauval, A.J. 2005, in *Cosmic Abundances as Records of Stellar Evolution and Nucleosynthesis*, ASP Conf Ser, F.N. Bash & T.G. Barnes, editors (astro-ph 0410214)

Avni, Y., & Tananbaum, H. 1986, ApJ, 305, 83

Avni, Y., Worrall, D. M., & Morgan, W. A. ApJ, 1995, 454, 673

Avrett, E. H., & Loeser, R. 1988, ApJ, 331, 211

Bahcall, J.H., & Kozlovsky, B.-Z. 1969, ApJ, 155, 1077

Bajtlik, S., Duncan, R. C., & Ostriker, J. P. 1988, ApJ, 327, 570

Bakes, E.L.O. & Tielens, A.G.G.M. 1994, ApJ, 427, 822 (BT94)

Balbus, S. A., & McKee, C. F. 1982, ApJ, 252, 529

Baldwin J. A., Ferland, G. J., Korista, K. T., Carswell, R., Hamann, F., Phillips, M., Verner, D., Wilkes, B., & Williams, R. E. 1996, ApJ, 461, 683

Baldwin, J. A., Ferland, G. J., Korista K. T., & Verner, D. 1995, ApJ, 455, L119

Baldwin, J.A., Ferland, G.J., Korista, K.T., Hamann, F., & Dietrich, M. 2003, ApJ, 582, 590

Baldwin, J., Ferland, G. J., Martin, P. G., Corbin, M., Cota, S., Peterson, B. M., & Slettebak, A. 1991, ApJ, 374, 580 (BFM)

Baldwin, J.A., Verner, E.M., Verner, D.A., Ferland, G.J., Martin, P.G., Korista, K.T., & Rubin, R. H., 2000, ApJS, 129, 229-246

Baldwin, J., Wampler, J., & Gaskell, C. M. 1989, ApJ, 338, 630

Balick, B., Gammon, R. H., & Hjellming, R. 1974 PASP, 86, 616

Ballantyne, D.R., Ferland, G.J., & Martin, P.G., 2000, ApJ 536, 773-777

Bässgen, G., Bässgen, M., & Grewing, M. 1988, A&A, 200, 51

Bates, D. R., Kingston, A. E., & McWhirter, R. W. P. 1962, Proc R Soc, A267, 297

Bauman, R.P., Porter, R.L., Ferland, G.J., & MacAdam, K. B. 2005, ApJ, in press (astro-ph/0504083)

Bechtold, J., Weymann, R. J., Lin, Z., & Malkan, M. A. 1987, ApJ, 315, 180

Behar, E., Sako, M, Kahn, S.M., 2001, ApJ, 563, 497-504

Behar, E., & Netzer, H., 2002, ApJ, 570, 165-170

Bell, K. L., Kingston, A. E., & McIlveen, W. A. 1975, J. Phys. B 8, 358

Benjamin, Robert A., Skillman, Evan D., & Smits, Derek P., 1999, ApJ, 514, 307

Bergin, E.A., Langer, W.D., & Goldsmith, P. F. 1995, ApJ, 441, 222

Berger, M. J., & Seltzer, S. M. 1965, NASA SP-3012

Bergeron, J., & Collin-Souffrin, S. 1971, A&A, 14, 167

Bergman, J.D., Allamandola, L. J., Witteborn, F. C., Tielens, A. G. G. M. & Geballe, T. R. 1989, ApJ, 344, 791

Berrington, K. A., & Pelan, A. 1995, A&AS, 114, 367

Berrington, K.A., Nakazaki, S., & Norrington, P.H. 2000, A&AS, 142, 313

Bertoldi, F., & Draine, B. T. 1996, ApJ, 458, 222

Bethe, H. 1930, Ann. Phys. 5, 325

Bica, E. 1988, A&A, 195, 76

Bieniek, R. J., & Dalgarno, A. 1979, ApJ, 228, 635

Binette, L., Prieto, A., Szuszkiewicz, E., & Zheng, W. 1989, ApJ, 343, 135

Black, J. H. 1978, ApJ, 222, 125

Black, J. H. 1987, in *Interstellar Processes*, ed. D.J. Hollenbach & H.A. Thronson, (Dordrecht: Reidel), p 731

Black, J.H. & van Dishoeck, E.F. 1987, ApJ, 322, 412

Bohm, D., & Aller, L. H. 1947, ApJ, 105, 131

Bonihala, J. R. M., Ferch, R., Salpeter, E. E., Slater, G., & Noerdlinger, P. 1979, ApJ, 233, 649

Borkowski, K. J., & Harrington, J. P. 1991, ApJ, 379, 168

Bottorff, M.C., Baldwin, J.A., Ferland, G.J., Ferguson, J.W., & Korista, K.T. 2002, ApJ, 581 932

Bottorff, M. C., Ferland, & Gary J., 2000, MNRAS 316, 103-106

Bottorff, M.C. & Ferland, G.J. 2001a, ApJ, 549, 118-132

Bottorff, M. C., Ferland, & Gary J., 2002, ApJ 568, 581-591

Bottorff, Mark, Ferland, Gary, Baldwin, Jack, & Korista, Kirk, 2000, ApJ, 542, 644-654

Bottorff, Mark C.; Baldwin, Jack A.; Ferland, Gary J.; Ferguson, Jason W.; Korista, Kirk T. 2002, ApJ, 581, 932

Bottorff, M., Lamothe, J., Momjian E., Verner, E., Vinkovic, D. & Ferland, G. 1998 PASP, 110, 1040

Boyd, R., & Ferland, G.J. 1987, ApJ, 318, L21

Bowen, I. S. 1960, ApJ, 132, 1

Bray, I., Burgess, A., Fursa, D.V., & Tully, J.A., 2000, A&AS, 146, 481

Bregman, J. D., Allamandola, L. J., Tielens, A. G. G. M., Geballe, T. R., & Witteborn, F. C. 1989, ApJ, 344, 791

Broad, J. T., & Reinhardt, W. P. 1976, Phys Rev A 14, 2159

Brocklehurst, M., 1970, MNRAS, 148, 417

Brocklehurst, M., 1972, MNRAS, 157, 211

Brooks, Frederick P., 1995, *The Mythical Man-Month, Essays on Software Engineering*, (Reading: Addison-Wesley)

Brown, R. L., & Mathews, W. G. 1970, ApJ, 160, 939

Burgess, A. 1965, ApJ, 141, 1588

Burgess, A., & Summers, H. P. 1969, ApJ, 157, 1007

Burgess, A., & Summers, H. P. 1976, MNRAS, 174, 345

Burgess, A., & Tully, J. A. 1992, A&A, 254, 436

Butler, S. E., Bender, C. F., & Dalgarno, A. 1979, ApJ, 230, L59

Butler, S. E., & Dalgarno, A. 1979, ApJ, 234, 765

Butler, S. E., Heil, T. G., & Dalgarno, A. 1980, ApJ, 241, 442

Butler, S. E., & Dalgarno, A. 1980, ApJ, 241, 838

Callaway, J. 1994, At Dat Nuc Dat Tab 57, 9

Cameron, A.G.W. 1982, in *Essays in Nuclear Astrophysics*, ed CA Barnes, DD Clayton, & DN Schramm,
       (Cambridge: Cambridge University Press)

Canfield, R. C., & Puetter, R. C. 1980, ApJ, 236, L7

Cardelli, J. A. 1994, Science 264, 209

Cardelli, J. A., et al. 1991, ApJ, 377, L57

Castor, J.I., Abbott, D.C., & Klein, R.I., 1975, ApJ, 195, 157-174

Carswell R. F. & Ferland, G. J. 1988, MNRAS, 235, 1121

Castor, J. I. 1970, MNRAS, 149, 111

Cazaux, S., & Tielens, A.G.G.M., 2002, ApJ, 575, L29-L32

Chaffee, F. H., & White, R. E. 1982, ApJS, 50, 169

Chamerlain, J.W., 1956, ApJ, 124, 390

Chan, E. S., Avrett, E. H., & Loeser, R. 1991, A&A, 247, 580

Chapman, R. D., & Henry, R. J. W. 1971, ApJ, 168, 169

Chidichimo, M. C. 1981, J. Phys. B., 14, 4149

Clavel, J., & Santos-Lleo, M. 1990, A&A, 230, 3

Clegg, R. E. S. 1987, MNRAS, 229, 31p

Clegg, R. E. S., & Harrington, J. P. 1989, MNRAS, 239, 869

Clegg, R.E.S., Storey, P.J., Walsh, J.R., & Neale, L. 1997, MNRAS, 284, 348

Cohen, E. R., & Taylor, B. N. 1987, Rev Mod Phys 57, 1121

Cota, S. A. 1987, Ph.D. Thesis, OSU

Cota, S. A., & Ferland, G. J. 1988, ApJ, 326, 889

Cowie, L. L., & Songaila, A. 1986, ARA&A 24, 499

Cowling, T.G., 1976, Magnetohydrodynamics, (Hilger; Bristol)

Craig, I.J.D., & Brown, J.C., 1986, *Inverse Problems in Astronomy* (Adam Hilger: Bristol)

CrinkLaw, G., Federman, S. R., & Joseph, C. L. 1994, ApJ, 424, 748

Crosas, M., & Weisheit, J.C. 1993, MNRAS, 262, 359

Cruddace, R., Paresce, F., Bowyer, S., & Lampton, M. 1974, ApJ, 187, 497

Cunto, W., Mendoza, C., Ochsenbein, F., Zeippen, C. J. 1993, A&A 275, L5

Dalgarno, A., & Kingston, A. E. 1963, Observatory, 83, 39

Dalgarno, A., & McCray, R. A. 1972, ARA&A, 10, 375

Dalgarno, A., & McCray, R. A. 1973, ApJ, 181, 95

Dalgarno, A., & Roberge, W. G. 1979, ApJ, 233, L25

Dalgarno, A., Yan, Min, & Liu, Weihong 1999, ApJS, 125, 237

Davidson, K. 1972, ApJ, 171, 213

Davidson, K. 1975, ApJ, 195, 285

Davidson, K. 1977, ApJ, 218, 20

Davidson, K., & Netzer, H. 1979, Rep. Prog. in Physics 51, 715

Davidson, K., & Fesen, R.A. 1985, ARA&A, 23, 119

de Jong, T., Chu, S-I., & Dalgarno, A. 1975, ApJ, 199, 69

Deguchi, S., & Watson, W.D. 1985, ApJ, 290, 578-586

Desert, F.-X., Boulanger, F., & Puget, J. L. 1990, A&A, 237, 215

Dove, J. E., Rush, A., Cribb, P., & Martin, P. G. 1987, ApJ, 318, 379

Dove, J. E., & Mandy, M. E. 1986, ApJ, 311, L93

Draine, B. T. 1978, ApJS, 36, 595

Draine, B.T., & Bertoldi, Frank, 1996, ApJ, 468, 269-289

Draine, B. T., & Lee, H. M. 1984, ApJ, 285, 89

Draine, B. T., & Salpeter, E. E. 1979, ApJ, 231, 77

Draine, B. T., & Sultin, B. 1987, ApJ, 320, 803

Drake, G., 1993, Chapt 3 in *Long Range Casimir Forces, theory and recent experiments on atomic systems*, edited by Levin & Mihca, Plenum Press

Drake, G. W. F. 1996, in *Atomic, Molecular, and Optical Physics Handbook*, ed. G. W. F. Drake (Woodbury: AIP), 154

Drake, S. A., & Ulrich, R.K. 1980, ApJS, 42, 351

Drawin, H.W. 1969, Z. Physic 225, 483

Dyson, J.E., & Williams, D.A. 1997, *The Physics of the Interstellar Medium* (Bristol; Institute of Physics Publishing)

Elitzur, M. 1982, Rev. Mod. Phys 54, 1125

Elitzur, M. 1984, ApJ, 280, 653

Elitzur, M, 1992, *Astronomical Masers*, (Dordrecht: Kluwer)

Elitzur, M., Ferland, G. J., Mathews, W. G., & Shields, G. 1983, ApJ, 272, L55

Elitzur, M., & Ferland, G. J. 1986, ApJ, 305, 35

Elvis, M. et al. 1994, ApJS, 95, 1

Elwert, T., et al. 2006, in preparation

Emerson, D. 1996, *Interpreting Astronomical Spectra*, (Chichester: John Wiley & Sons)

Fabian, A. C., Pringle, J. E., & Rees M. J. 1976, MNRAS, 175, 43

Federman, S.R., Glassgold, A.E., & Kwan, J. 1979, ApJ, 227, 466

Federman, S. R., et al. 1993, ApJ, 413, L51

Fenley, J.A., Taylor, K.T., & Seaton, M.J. 1987, J. Phys. B. 20, 6457-6476

Ferguson, J. W., Ferland, G. J., & A. K. Pradhan, 1995, ApJ, 438, L55

Ferguson, J. W., & Ferland, G.J. 1997, ApJ, 479, 363

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

Ferguson, J W., Korista, Kirk. T., and Ferland, Gary J., 1997, ApJS 110, 287-297

Ferguson, J W., Korista, Kirk. T., Verner, D.A., & Ferland, Gary J., 2001, ASP Conference Series, Vol 247, *Spectroscopic Challenges of Photoionized Plasmas*, G Ferland & D Savin, editors.

Ferland, G. J. 1977, ApJ, 212, L21

Ferland, G. J. 1979, MNRAS, 188, 669

Ferland, G. J. 1980a, MNRAS, 191, 243

Ferland, G. J. 1980b, BAAS, 12, 853

Ferland, G. J. 1980c, PASP, 92, 596

Ferland, G. J. 1986, PASP, 98, 549

Ferland, G. J. 1986, ApJ, 310, L67

Ferland, G. J. 1992, ApJ, 389, L63 NIII

Ferland, G. J. 1993, ApJS, 88, 49

16 REFERENCES

Ferland, G. J. 1999, PASP, 111, 1524

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

Ferland, G. J., 1999b, ApJ 512 247-249

Ferland, G.J., 2000, RMxAC, 9, 153

Ferland, G.J., 2001a, PASP, 113, 41

Ferland, G.J., 2001b, ASP Conference Series, Vol 247, *Spectroscopic Challenges of Photoionized Plasmas*, G Ferland & D Savin, editors (astro-ph/0210161)

Ferland, G.J., 2003a, in *Star formation through time*, 2003, by Gary J. Ferland, ASP Conf 297, E. Pérez, R.M. González Delgado, & G. Tenorio-Tagle, eds, p 69

Ferland, G.J., 2003b, ARA&A, 41, 517

Ferland, G. J., Baldwin J. A., Korista, K. T., Hamann, F., Carswell, R., Phillips, M., Wilkes, B., & Williams, R. E. 1996, ApJ, 461, 683

Ferland, G., Binette, L., Contini, M., Harrington, J., Kallman, T., Netzer, H., Péquignot, D., Raymond, J., Rubin, R., Shields, G., Sutherland, R., & Viegas, S. 1995, in *The Analysis of Emission Lines*, Space Telescope Science institute Symposium Series, R. Williams & M. Livio, editors (Cambridge: Cambridge University Press)

Ferland, G. J., & Elitzur, M. 1984, ApJ, 285, L11

Ferland, G. J., Fabian, A. C., & Johnstone, R.M. 1994, MNRAS, 266, 399

Ferland, G. J., Fabian, A. C., & Johnstone, R.M. 2002, MNRAS, 333, 876

Ferland, G. J., Henney, W. J., Williams, R. J. R., Arthur, S. J. 2002, RMxAC, 12, 43

Ferland, G. J., Korista, K.T. & Peterson, B.M. 1990, ApJ, 363, L21

Ferland, G. J., Korista, K.T., Verner, D. A., & Dalgarno, A. 1997, ApJ, 481, L115

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

Ferland, G. J., Lambert, D. L., Netzer, H., Hall, D. N. B., & Ridgway, S. T. 1979a, ApJ, 227, 489

Ferland, G. J., Lambert, D. L., Slovak, M., Shields, G. A., & McCall, M. 1982, ApJ, 260, 794

Ferland, G. J., & Mushotzky, R. F. 1982, ApJ, 262, 564

Ferland, G. J., & Mushotzky, R. F. 1984, ApJ, 286, 42

Ferland, G. J., & Netzer, H. 1979, ApJ, 229, 274

Ferland, G. J., & Netzer, H. 1983, ApJ, 264, 105

Ferland, G. J., Netzer, H., & Shields, G. A. 1979, ApJ, 232, 382

Ferland, G. J., Peterson, B. M., Horne, K., Welsh, W. F., & Nahar, S. N. 1992, ApJ, 387, 95

Ferland, G. J., & Persson, S. E. 1989, ApJ, 347, 656

Ferland, G. J., & Rees, M. J. 1988, ApJ, 332, 141

Ferland, G. J., & Shields, G. A. 1978, ApJ, 226, 172

Ferland, G. J., & Shields, G. A. 1985, in *Astrophysics of Active Galaxies & Quasi-stellar Objects*, J.S. Miller, Ed.

Ferland, G. J., & Truran, J. W. 1981, ApJ, 244, 1022

Ferland, G. J., Williams, R. E., Lambert, D. L., Shields, G. A., Slovak, M., Gondhalekar, P. M., & Truran, J. W. 1984, ApJ, 281, 194

Field, G. B. 1965, ApJ, 142, 431

Francis, P. J. 1993, ApJ, 407, 519

Friedrich, H. 1998, Theoretical Atomic Physics, (Berlin: Springer)

Fuhr, J. R., Martin, G. A., & Wiese, W. L. 1988, J. Phys. Chem. Ref. Data, 17, Suppl. 4

Gaetz, T. J., & Salpeter, E. E. 1983, ApJS, 52, 155

Garstang, R.H. 1958, MNRAS, 118, 57

Gavrila, M. 1967, Phys Rev 163, 147, also JILA Report #86, Sept 19, 1966

Ginzburg, V. I., & Syrovatskii, S. I. 1964, *The Origin of Cosmic Rays*, (Oxford: Pergamon)

Gould, R. S. 1978, ApJ, 219, 250

Grandi, S.A., 1975, ApJ, 196, 465

Grandi, S.A., 1975, ApJ, 199, 43

Grandi, S.A., 1976, ApJ, 206, 658

Gredel, R., Lepp, S., & Dalgarno, A. 1987, ApJ, 323, L137

Gredel, R., Lepp, S., Dalgarno, A., & Herbst, E. 1989, ApJ, 347, 289

Greenhouse, M., et al. 1993, ApJS, 88, 23

Grevesse, N., & Anders, E. 1989, *Cosmic Abundances of Matter*, AIP Conference Proceedings 183, p. 1, Ed. C. J. Waddington, (New York: AIP)

Grevesse, N. & Noels, A. 1993 in *Origin & Evolution of the Elements*, ed. N. Prantzos, E. Vangioni-Flam, & M. Casse p. 15 (Cambridge: Cambridge Univ. Press)

Grevesse, N., & Sauval, A.J., 1998, Space Science Review, 85, 161-174

Guhathakurta, P., & Draine, B. T. 1989, ApJ, 345, 230

Guilbert, P. W. 1986, MNRAS, 218, 171

Guilbert, P., & Rees, M. J. 1988, MNRAS, 233, 475

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

Habing, H. J. 1968, Bull. Astr. Inst. Netherlands 19, 421

Habart, E. Boulanger, F. Verstraete, L. Walmsley, C.M., & Pineau des Forets G., 2003, A&A in press, astro-ph/ 0311040

Halpern, J. P., & Grindlay, J. E. 1980, ApJ, 242, 1041

Hamann, F., & Ferland, G. J. 1992, ApJ, 391, L53

Hamann, F., & Ferland, G. J. 1993, ApJ, 418, 11

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

Harrington, J. P. 1969, ApJ, 156, 903

Harrington, J. P. 1973, MNRAS, 162, 43

Hasegawa, T. I. & Herbst, E. 1993, MNRAS, 261, 83

Hasegawa, T. I., Herbst, E., & Leung, C.M. 1992 ApJS, 82, 167

Hauschildt, P.H., & Baron, E., 1999, J. Comp. Appl. Math, 109, 41-63

Heiles, C. & Crutcher, R. 2005, *Magnetic Fields in Diffuse H I and Molecular Clouds*, Chapter in Cosmic Magnetic Fields, astro-ph/0501550

Heitler, W. 1954, *The Quantum Theory of Radiation* (Oxford: Oxford University Press)

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

Hilborn, Robert C., 1982, American Journal of Physics, 50, 982-986, erratum, 51 471

Hjellming, R. M. 1966, ApJ, 143, 420

Hogg, D.W. 1999, astro-ph/9905116

Hollenbach, D., & McKee, C. F. 1979, ApJS, 41, 555

Hollenbach, D., & McKee, C. F. 1989, ApJ, 342, 306

Hollenbach, D.J., Takahashi, T., & Tielens, A.G.G.M., 1991, ApJ, 377, 192-209

Hollenbach, D.J., & Tielens, A. G. G. M. 1997, ARA&A, 35, 179

Hollenbach, D.J., & Tielens, A.G.G.M. 1999, Rev Mod Phys 71, 173

Holweger, H., 2001, Joint SOHO/ACE workshop "Solar and Galactic Composition". Edited by Robert F. Wimmer-Schweingruber. Publisher: American Institute of Physics Conference proceedings vol. 598 location: Bern, Switzerland, March 6 - 9, 2001, p.23

16 REFERENCES

Hubbard, E. N., & Puetter, R. C. 1985, ApJ, 290, 394

Hubeny, I. 2001, Hubeny 2001 Spectroscopic Challenges of Photoionized Plasmas, ASP Conference Series Vol. 247. Edited by Gary Ferland and Daniel Wolf Savin. San Francisco: Astronomical Society of the Pacific, p.197

Hummer, D. G. 1962, MNRAS, 125, 21

Hummer, D. G. 1968, MNRAS, 138, 73

Hummer, D. G. 1988, ApJ, 327, 477

Hummer, D. G, Berrington, K. A., Eissner, W., Pradhan, A. K., Saraph H. E., Tully, J. A. 1993, A&A, 279, 298

Hummer, D. G., & Kunasz, P.B. 1980, ApJ, 236, 609

Hummer, D. G., & Seaton, M. J. 1963, MNRAS, 125, 437

Hummer, D. G., & Storey, P. J. 1987, MNRAS, 224, 801

Hummer, D. G., & Storey, P. J. 1992, MNRAS, 254, 277

Hutchings, J.B. 1976, ApJ, 205, 103

Ikeuchi, S., & Ostriker, J. P. 1986, ApJ, 301, 522

Jackson, J. D. 1975, *Classical Electrodynamics* (New York: Wiley)

Kaler, J., & Jacoby, G. 1991, ApJ, 372, 215

Janev, R. K., Langer, W. D., Post, D. E., & Evans, K. 1987, *Elementary Processes in Hydrogen–Helium Plasmas* (Berlin: Springer--Verlag)

Jenkins, E. B. 1987, in *Interstellar Processes*, D. Hollenbach & H. Thronson, Eds, (Dordrecht: Reidel), p.533

Johnson, L. C. 1972, ApJ, 174, 227

Johnstone, R. M., Fabian, A. C., Edge, A. C., & Thomas, P. A. 1992, MNRAS, 255, 431

Jones, A.P., Tielens, A. G. G. M., & Hollenbach, D.J. 1996, 469, 740-764

Jura, M., 1974, ApJ, 191, 375-379

Jura, M., 1975, ApJ, 197, 575-580

Kaler, J., 1978, ApJ, 220, 887

Kallman, T. R., & McCray, R. 1982, ApJS, 50, 263

Karzas, W. J., & Latter, R. 1961, ApJS, 6, 167

Kaastra, J. S., & Mewe, R. 1993, A&AS, 97, 443

Kato, T. 1976, ApJS, 30, 397

Kellerman, K. I. 1966, ApJ, 146, 621

Khromov, G. S. 1989, Space Science Reviews 51, 339

Kingdon, J. B., & Ferland, G. J. 1991, PASP, 103, 752

Kingdon, J. B., & Ferland, G. J. 1993, ApJ, 403, 211

Kingdon, J. B., & Ferland, G. J. 1995, ApJ, 450, 691

Kingdon, J. B., & Ferland, G. J. 1996, ApJS, 106, 205

Kingdon, J. B., Ferland, G. J., & Feibelman, W.A. 1995, ApJ, 439, 793

Kingdon J.B., & Ferland, G.J., 1998, ApJ 506, 323-328

Kingdon, J. B., & Ferland, G. J. 1998, ApJ, 516, L107-109

Korista, K. T., Baldwin, J. A., & Ferland, G. J. 1998, ApJ, 507, 24

Korista, K. T., & Ferland, G. J. 1989, ApJ, 343, 678

Korista, K. T., & Ferland, G. J. 1998, ApJ, 495, 672

Korista, K. T., Ferland, G. J., & Baldwin, J. 1997, ApJ, 487, 555

Kraemer, S.B., Ferland, G.J., & Gabel, J.R. 2004, ApJ 604, 556

Krolik, J., McKee, C. M., & Tarter, C.B. 1981, ApJ, 249, 422

Kurucz, R. L. 1970, SAO Special Reports 309

Kurucz, R. L. 1979, ApJS, 40, 1

Kurucz, R. L. 1991, in *Proceedings of the Workshop on Precision Photometry: Astrophysics of the Galaxy*, ed. A. C. Davis Philip, A. R. Upgren, & K. A. James, (Schenectady: Davis), 27

Kwan, J., & Krolik, J. 1981, ApJ, 250, 478

Lambert, D. L., & Pagel, B. E. J. 1968, MNRAS, 141, 299

La Franca, Franceshini, A., Cristiani, S., & Vio, R. 1995, A&A, 299, 19

Lambert, D.L. 1993, Phys Scrip T47, 186

Lame N. J., & Ferland, G. J. 1991, ApJ, 367, 208

LaMothe, J., & Ferland, G.J., 2001, PASP, 113, 165

Landini, M., & Monsignori Fossi, B. 1990, A&AS, 82, 229

Landini, M., & Monsignori Fossi, B. 1991, A&AS, 91, 183

Lanzafame, A., Tully, J. A., Berrington, K. A., Dufton, P. L., Byrne, P. B., & Burgess, A. 1993, MNRAS, 264, 402

Laor, A., & Draine, B. T. 1993, ApJ, 402, 441

Latter, W. B., & Black, J. H. 1991, ApJ, 372, 161

Lea, S., & Holman, G. 1978, ApJ, 222, 29

Le Bourlot, J., 2000, A&A, 360, 656-662

Le Bourlot, J., Pineau des Forets, G., & Flower, D.R. 1999, MNRAS, 305, 802

Leitherer, Claus; Schaerer, Daniel; Goldader, Jeffrey D.; Delgado, Rosa M. González; Robert, Carmelle; Kune, Denis Foo; de Mello, Duília F.; Devost, Daniel; Heckman, Timothy M. 1999, ApJS, 123, 3

Lennon, D. J., & Burke, V. M. 1991, MNRAS, 251, 628

Lenzuni, P., Chernoff, D. F., & Salpeter, E. E. 1991, ApJS, 76, 759

Levich, E. V., & Sunyaev, R.A. 1970, Astrophysical Letters 7, 69

Lepp, S., & Shull, J. M. 1983, ApJ, 270, 578

Li, Aigen, & Draine, B.T. 2001, ApJ, 554, 778L

Lightman, A. P., & White, T.R. 1988, ApJ, 335, 57

Liske, J., 2000, MNRAS, 319, 557-561

Lites, B. W., & Mihalas, D. 1984, Solar Physics 93, 23

Liu, X.-W., Storey, P. J., Barlow, M. J., & Clegg, R. E. S. 1995, MNRAS, 272, 369

Longair, M. S. 1981, *High Energy Astrophysics*, (Cambridge: Cambridge University Press)

Lotz, W. 1967, ApJS, 14, 207

Launay, J.R., Le Dourneuf, M., & Zeippen, C.J., 1991, A&A, 252, 842-852

MacAlpine, G. M. 1971, ApJ, 175, 11

Maguire, S. 1993, *Writing Solid Code*, (Redmond: Microsoft Press)

Maguire, S. 1994, *Debugging the Development Process*, (Redmond: Microsoft Press)

Mallik, D. C. V., & Peimbert, M. 1988, Rev Mexicana 16, 111

Maloney, P.R., Hollenbach, D.J., & Tielens, A. G. G. M., 1996, ApJ, 466, 561

Martin, P. G. 1979, *Cosmic Dust* (Oxford: Clarendon Press)

Martin, P. G. 1988, ApJS, 66, 125

Martin, P. G., & Ferland, G. J. 1980, ApJ, 235, L125

Martin, P.G., & Rouleau, F., 1991, in Malina R.F., Bowyer S., eds, Extreme Ultraviolet Astronomy, Pergamon Press, Oxford, p. 341

Martin, P. G., & Whittet, D. C. B. 1990, ApJ, 357, 113

Masters, A. R., Pringle, J. E., Fabian, A. C., & Rees, M. J. 1977, MNRAS, 178, 501

Mather, J.C., Fixsen, D.J., Shafer, R.A., Mosier, C., & Wilkinson, D.T. 1999, ApJ, 512, 511

Mathews, W. G., Blumenthal, G. R., & Grandi, S. A. 1980, ApJ, 235, 971

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

Mathis, J. S. 1982, ApJ, 261, 195

Mathis, J. S. 1985, ApJ, 291, 247

Mathis, J. S., Rumpl, W., & Nordsieck, K. H. 1977, ApJ, 217, 425

Mathis, J. S., & Wallenhorst, S. G. 1981, ApJ, 244, 483

Matteucci, F., & Tornambe, A. 1987, A&A, 185, 51

Matteucci, F., & Greggio, A. 1986, A&A, 154, 279

Mazzotta, P., Mazzitelli, G., Colafrancesco, C., & Vittorio, 1998, A&AS 133, 403-409

McCall, B.J., et al. 2003, Nature, 422, 500

McKee, C. F. 1999, preprint, Astro-ph 9901370

Mendoza, C. 1983, in *Planetary Nebulae*, IAU Sym 103, D. R. Flower, Ed., p 143, (Dordrecht: Reidel)

Meyer, D.M., Jura, M., & Cardelli, J.A. 1998, ApJ, 493, 222-229

Mewe, R. 1972, A&A, 20, 215

Mihalas, D. 1972, *Non-LTE Model Atmospheres for B & O Stars*, NCAR-TN/STR-76

Mihalas, D. 1978, *Stellar Atmospheres*, 2nd Edition (San Francisco: W.H. Freeman)

Mihalszki, J. S., & Ferland, G. J. 1983, PASP, 95, 284

Mohr P.J. & Taylor B.N., 1998 Codata, see Reviews of Modern Physics, Vol. 72, No. 2, 2000

Morrison, R., & McCammon, D. 1983, ApJ, 270, 119

Morton, D. C., York, D. G., & Jenkins, E. B. 1988, ApJS, 68, 449

Nahar, S. N., & Pradhan, A. K. 1992, ApJ, 397, 729

Netzer, H. 1990, in *Active Galactic Nuclei, Saas-Fee Advanced Course 20*, Courvorsier, T.J.-L., & Mayor, M., (Springer-Verlag; Berlin)

Netzer, H., Elitzur, M., & Ferland, G. J. 1985, ApJ, 299, 752

Netzer, H., & Ferland, G. J. 1984, PASP, 96, 593

Neufeld, D. A. 1989, Harvard Research Exam

Neufeld, D.A., 1990, ApJ, 350, 216

Neufeld, D. A., & Dalgarno, A. 1989, Phys Rev A, 35, 3142

Novotny, Eva, 1973, *Introduction to Stellar Atmospheres*, (New York; Oxford University Press)

Nussbaumer, H., & Storey, P. J. 1983, A&A, 126, 75

Nussbaumer, H., & Storey, P. J. 1984, A&AS, 56, 293

Nussbaumer, H., & Storey, P. J. 1986, A&AS, 64, 545

Nussbaumer, H., & Storey, P. J. 1987, A&AS, 69, 123

O'Dell, C.R., 2001, ARAA, 39, 99

Oliveira, S., & Maciel, W. J. 1986, Ap&SS, 126, 211

Oliva, E., Pasquali, A., & Reconditi, M. 1996, A&A, 305, 210

Olive, K.A., Steigman, G., & Walker, T.P., 2000, Physics Reports, 333-334, 389-407

Osterbrock, D. E. 1951, ApJ, 114, 469

Osterbrock, D. E., & Ferland, G.J. 2005, *Astrophysics of Gaseous Nebulae & Active Galactic Nuclei*, 2nd edition, (Mill Valley; University Science Press) (AGN3)

Osterbrock, D. E., & Flather, E. 1959, ApJ, 129, 26

Osterbrock, D. E., Tran, H. D., & Veilleux, S. 1992, ApJ, 389, 305

Ostriker, J. P., & Ikeuchi, S. 1983, ApJ, 268, L63

Pacholczyk, A. G. 1970, *Radio Astrophysics* (San Francisco: Freeman)

Pagel, B. E. J. 1997, *Nucleosynthesis and Chemical Evolution of Galaxies*, (Cambridge: Cambridge University Press)

Palla, F., Salpeter, E. E., & Stahler, S. W. 1983, ApJ, 271, 632

Peebles, P. J. E. 1971, *Physical Cosmology*, (Princeton: Princeton U. Press)

Peimbert, M. 1967, ApJ, 150, 825

Pengelly, R. M. 1964, MNRAS, 127, 145

Pengelly, R.M., & Seaton, M.J., 1964, MNRAS, 127, 165

Péquignot, D. 1986, *Wordshop on Model Nebulae*, (Paris: l'Observatoire de Paris) p363

Péquignot, D., & Aldrovandi, S.M.V. 1986, A&A, 161, 169

Péquignot, D., Ferland, G.J., et al., 2001, in ASP Conference Series, Vol 247, *Spectroscopic Challenges of Photoionized Plasmas*, G Ferland & D Savin, editors

Péquignot, D., Petitjean, P., & Boisson, C. 1991, A&A, 251, 680

Péquignot, D., Stasinska, G., & Aldrovandi, S. M. V. 1978, A&A, 63, 313

Percival, I.C., & Richards, D., 1978, MNRAS, 183, 329

Peterson, B.M. 1993, PASP, 105, 247

Peterson, J. R., Aberth, W., Moseley, J., & Sheridan, J. 1971, Phys Rev A, 3, 1651

Pettini, M., & Bowen, D.V., 2001, ApJ, 560, 41

Poglitsch, A., Krabbe, A., Madden, S.C., Nikola, T., Geis, N., Johansson, L., Stacey, G. & Sternberg, A. 1995, ApJ, 454, 293

Porquet, D., & Dubau, J. 2000, A&AS, 143, 495

Porter, R.L., Bauman, R.P., Ferland, G.J., & MacAdam, K. B. 2005, ApJ, 622, 73L (astro-ph/0502224)

Prasad, S.S., & Huntress, W.T., 1980, ApJS, 43, 1-35

Press W. H., Teukolsky, S.A., Vetterling, W. T., & Flannery, B. P. 1992, *Numerical Recipes*, (Cambridge; Cambridge University Press)

Puetter, R. C. 1981, ApJ, 251, 446

Puy, D., Alecian, G., Le Bourlot, J., Leorat, J., & Pineau des Forets, G. 1993, A&A, 267, 337

Puy, D., Grenacher, L, & Jetzer, P., 1999, A&A, 345, 723

Rauch, T. 1997 A&A, 320, 237

Rauch, T. 2002, H-Ni grid, available at http://astro.uni-tuebingen.de/~rauch

Raymond, J. C., Cox, D. P., & Smith, B. W. 1976, ApJ, 204, 290

Rees, M. J., Netzer, H., & Ferland, G. J. 1989, ApJ, 347, 640

van Regemorter, H. 1962, ApJ, 136, 906

Rephaeli, Y. 1987, MNRAS, 225, 851

Reilman, R. F., & Manson, S. T. 1979, ApJS, 40, 815, errata 46, 115; 62, 939

Roberge, W. G., Jones, D., Lepp, S., & Dalgarno, A. 1991, ApJS, 77, 287

Röllig, M., et al., 2005, Leiden PDR comparison, A&A in press

Rossi, B. 1952, *High-Energy Particles* (New York; Prentice-Hall)

Rouleau, F., & Martin, P.G. 1991, ApJ, 377, 526

Rowan, T. 1990, *Functional Stability Analysis of Numerical Algorithms*, Ph.D. Thesis, Department of Computer Sciences, University of Texas at Austin

Rubin, R. H. 1968, ApJ, 153, 671

Rubin, R. H. 1983, ApJ, 274, 671

Rubin, R. H. Martin, P. G. Dufour, R. J. Ferland, G. J Baldwin, J. A. Hester, J. J. & Walter, D. K. 1998, ApJ, 495, 891

Rubin, R. H., Simpson, J. R., Haas, M. R., & Erickson, E. F. 1991, ApJ, 374, 564

Rutten, Rob, 2002, Radiative transfer in stellar atmospheres, at http://www.fys.ruu.nl/~rutten/node20.html

Rybicki, G. B., & Hummer, D. G. 1991, A&A, 245, 171

Rybicki, G. B., & Hummer, D. G. 1992, A&A, 262, 209

Rybicki, G. B., & Hummer, D. G. 1994, A&A, 290, 553

Rybicki, G. B., & Lightman, A.P. 1979, *Radiative Processes in Astrophysics* (New York: Wiley)

Sanders, D. B., et al. 1989, ApJ, 347, 29

Saraph, H. E. 1970, J.Phys.B., 3, 952

Savage, B. D., & Jenkins, E.B. 1972, 172, 491

Savage, B. D., & Sembach, K. R. 1996, ARA&A, 34, 279

Savin, Daniel Wolf, 2000, ApJ, 533, 106

Savin, D. W.; Kahn, S. M.; Linkemann, J.; Saghiri, A. A.; Schmitt, M.; Grieser, M.; Repnow, R.;
    Schwalm, D.; Wolf, A.; Bartsch, T.; Brandau, C.; Hoffknecht, A.; Müller, A.; Schippers, S.;
    Chen, M. H.; Badnell, N. R., 1999, ApJS, 123, 687

Sciortino, S., et al. 1990, ApJ, 361, 621

Scott, J. S., Holman, G. D., Ionson, J. A., & Papadopoulos, K. 1980, ApJ, 239, 769

Schaerer D., de Koter, A., Schmutz, W., & Maeder, A. 1996ab, A&A, 310, 837, & A&A, 312, 475

Schaerer D., & de Koter A. 1997, A&A, 322, 592

Schuster, A. 1905, ApJ, 21, 1

Schutte, W. A., Tielens, A. G. G. M., & Allamandola, L. J. 1993, ApJ, 415, 397

Schwarzschild, M. 1965, *Structure & Evolution of the Stars*, (New York: Dover)

Seaton, M. J. 1959, MNRAS, 119, 81

Seaton, M. J. 1959, MNRAS, 119, 90

Seaton, M. J. 1987, J.Phys. B, 20, 6363

Sellgren, K., Tokunaga, A. T., & Nakada, Y. 1990, ApJ, 349, 120-125

Sellmaier, F. H., Yamamoto, T., Pauldrach, A. W. A., & Rubin, R. H. 1996, A&A, 305, L37

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

Shields, G. A. 1976, ApJ, 204, 330

Shine, R. A., & Linsky, J. L. 1974, Solar Physics 39, 49

Shull, J. M. 1979, ApJ, 234, 761

Shull, J.M., & Beckwith, S. 1982, ARA&A, 20, 163

Shull, J. M., & Van Steenberg, M. E. 1982, ApJS, 48, 95

Shull, J. M., & Van Steenberg, M. E. 1985, ApJ, 298, 268

Sellgren, K., Tokunaga, A. T., & Nakada, Y. 1990, ApJ, 349, 120

Sellmaier, F.H., Yamamoto, T., Pauldrach, A.W.A., Rubin, R.H 1996, A&A, 305, 37

Sikora, M., Begelman, M. C., & Rudak, B. 1989, ApJ, 341, L33

Simonyi, C. 1977, *Meta-Programming: A Software Production Method*, Thesis, Stanford University

Simpson, J. P. 1975, A&A, 39, 43

Smits, D.P., 1996, MNRAS, 278, 683

Snow, T. P., & Dodger, S. L. 1980, ApJ, 237, 708

Snow, T. P., & York, D. G. 1981, ApJ, 247, L39

Snow, T. P., & Witt, A. 1996, ApJ, 468, L65

Spitzer, L. 1948, ApJ, 107, 6

Spitzer, L. 1962, *Physics of Fully Ionized Gasses*, (New York: Interscience)

Spitzer, L. 1978, *Physical Processes in the Interstellar Medium*, (New York: Wiley)

Spitzer, L. 1985, ApJ, 290, L21

Spitzer, L., & Tomasko, M. G. 1968, ApJ, 152, 971

Stasinska, G. & Szczerba, R 1999, A&A, 352, 297

Stecher, T. P., & Williams, D. A. 1967, ApJ, 149, 29

Steenbock, W. & Holweger, H. 1984, A&A, 130, 319

Sternberg, A., & Dalgarno, A. 1989, ApJ, 338, 197

Sternberg, A., & Neufeld, D.A. 1999, ApJ, 516, 371-380

Stoy, R. H. 1933, MNRAS, 93, 588

Storey, P. J. 1981, MNRAS, 195, 27p

Storey, P. J. 1994, A&A, 282, 999

Storey, P. J., & Hummer, D. G. 1991, Comput. Phys. Commun. 66, 129

Storey, P. J., & Hummer, D. G. 1995, MNRAS, 272, 41 (on the web at http://adc.gsfc.nasa.gov/adc-cgi/cat.pl?/catalogs/6/6064/)

Suchkov, A., Allen, R.J., & Heckman, T. M. 1993, ApJ, 413, 542-547

Swings, P., & Struve, O. 1940, ApJ, 91, 546

Takahashi, Junko, 2001, ApJ, 561, 254-263

Tarter, C. B. 1967, Ph. D. Thesis, Cornell University

Tarter, C. B., & McKee, C. F. 1973, ApJ, 186, L63

Tarter, C. B., Tucker, W.H., & Salpeter, E.E., 1969, ApJ, 156, 943

Telfer, R.C., Zheng, W., Kriss, G.A., & Davidsen, A.F. 2002, ApJ, 565, 773

Tielens, A.G.G.M. 2005, *The Physics and Chemistry of the Interstellar Medium*, (Cambridge: Cambridge University Press)

Tielens, A. G. G. M., & Hollenbach, D. 1985a, ApJ, 291, 722

Tielens, A. G. G. M., & Hollenbach, D. 1985b, ApJ, 291, 746

Tinsley, B. 1979, ApJ, 229, 1046

Tout, C. A., Pols, O. R., Eggleton, P. P. & Han, Z. 1996, MNRAS, 281, 257

Turner, J., & Pounds, K. 1989, MNRAS, 240, 833

Van Blerkom, D., & Hummer, D. G. 1967, MNRAS, 137, 353

Veigele, WM. J. 1973, Atomic Data Tables, 5, 51

van Dishoeck, E.F., & Black, J.H., 1988, ApJ, 334, 771

van Hoof, P. A. M. 1997, PhD Thesis, University of Groningen

van Hoof, P.A.M., Beintema, D.A., Verner D.A., & Ferland, G.J., 2000a, A&A 354, L41-L44

van Hoof, P.A.M., Van de Steene, G.C., Beintema, D.A., Martin, P.G., Pottasch, S.R., Ferland, G. J., 2000b, ApJ 532, 384-399

van Hoof, P.A.M., Weingartner, J.C., Martin, P.G., Volk, K., & Ferland, G.J., 2001, in *Challenges of Photoionized Plasmas*, (G Ferland & D. Savin, eds) ASP Conf Ser 247, 363-378 (astro-ph/0107183)

van Hoof, P.A.M., Weingartner, J.C., Martin, P.G., Volk, K., & Ferland, G.J., 2004, MNRAS, 350, 1330 (astro-ph/0402381)

van Regemorter, H. 1962, ApJ, 136, 906

Vedel, H., Hellsten, U., & Sommer-Larsen, J. 1994, MNRAS, 271, 743

Vernazza, J. E., Avrett, E. H., & Loeser, C. B. 1981, ApJS, 45, 635

Verner, D. A., Yakovlev, D. G., Band, I. M., & Trzhaskovshaya, M. B. 1993, ADNDT, 55, 233

Verner, D. A., & Yakovlev, 1995, A&AS, 109, 125

Verner, D. A., & Ferland, G. J. 1996, ApJS, 103, 467

Verner, D. A., Ferland, G. J., Korista, K., & Yakovlev D. G. 1996, ApJ, 465, 487

Verner, D. A., Verner, K., & Ferland, G. J. 1996, ADNDT, 64, 1

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

Voronov, G. S. 1997, ADNDT, 65, 1

Voit, G. M. 1991, ApJ, 377, 1158

Volk, K., and Kwok, S. 1988, ApJ, 331, 435

Vriens, L., & Smeets, A. H. M. 1980, Phys Rev A, 22, 940

Vrinceanu, D. & Flannery, M. R. 2001, PhysRevA 63, 032701

Watson, W. D. 1972, ApJ, 176, 103

Weingartner, J.C., & Draine, B.T., 2001a, ApJS, 134, 263

Weingartner, J.C., & Draine, B.T., 2001b, ApJ, 548, 296

Weingartner, J.C., & Draine, B.T., 2001c, ApJ, 563, 842

Weisheit, J. C. 1974, ApJ, 190, 735

Weisheit, J. C., & Collins, L. A. 1976, ApJ, 210, 299

Weisheit, J. C., & Dalgarno, A. 1972, Astrophysical Letters, 12, 103

Weisheit, J., Shields, G. A., & Tarter, C. B. 1981, ApJ, 245, 406

Wen, Z., & O'Dell, C.R. 1995, ApJ, 438, 784-793

Werner, K., & Heber, U. 1991, in *Stellar Atmospheres: Beyond Classical Models*, p 341, NATO ASI Series C, eds. L. Crivellari, I. Hubney, & D. G. Hummer, (Dordrect: Kluwer)

White, R. E. 1986, ApJ, 307, 777

Wiese, W.L., Fuhr, J.R., & Deters, T.M., 1996, J Phys Chem Ref Data, Monograph 7

Wiese, W. L., Smith, M. W., & Glennon, B. M. 1966, NSRDS-NBS 4

Wilkes, B. J., Ferland, G. J., Truran, J., & Hanes, D. 1981, MNRAS, 197, 1

Wilkes, et al 1994, ApJS, 92, 53

Wilkinson, D. T. 1987, in *13th Texas Symposium on Relativistic Astrophysics*, M. P. Ulmer, ed., (Singapore: World Scientific), p209

Williams, J.P., Bergin, E.A., Caselli, P., Myers, P.C., & Plume, R. 1998, ApJ, 503, 689

Williams, R. E. 1967, ApJ, 147, 556

Williams, R. E. 1992, ApJ, 392, 99

Wills, B., Netzer, H., & Wills, D. 1985, ApJ, 288, 94

Wilson, T. L., & Rood, R. T. 1994, ARA&A, 32, 191

Wing, W.H., & MacAdam, K.B., 1978, in *Progress in Atomic Spectroscopy*, Part A, W. Hanle & H. Kleinpopper, eds

Winslow, A. M. 1975, Lawrence Livermore Lab. report UCID-16854

Wishart, A. W. 1979, MNRAS, 187, 59p

Wolfire, M. G., Tielens, A., & Hollenbach, D. 1990, ApJ, 358, 116

Worral et al. 1987, ApJ, 313, 596

Wyse, A. B. 1941, PASP, 53, 184

York, D. G., Meneguzzi, M., & Snow, T. 1982, ApJ, 255, 524

Xu, Y., & McCray, R. 1991, ApJ, 375, 190

Zamorani, G., et al. 1981, ApJ, 245, 357

Zheng, W., Kriss, G.A., Telfer, R.C., Grimes, JP. & Davidsen, A.F. 1997, ApJ, 475, 469

Zuckerman, B. 1973, ApJ, 183, 863

Zycki, P. T., Krolik, J. H., Zdziarski, A. A., & Kallman, T. R. 1994, 437, 597

Zygelman, B., & Dalgarno, A. 1990, ApJ, 365, 239

17. INDEX

refer

refer

refer