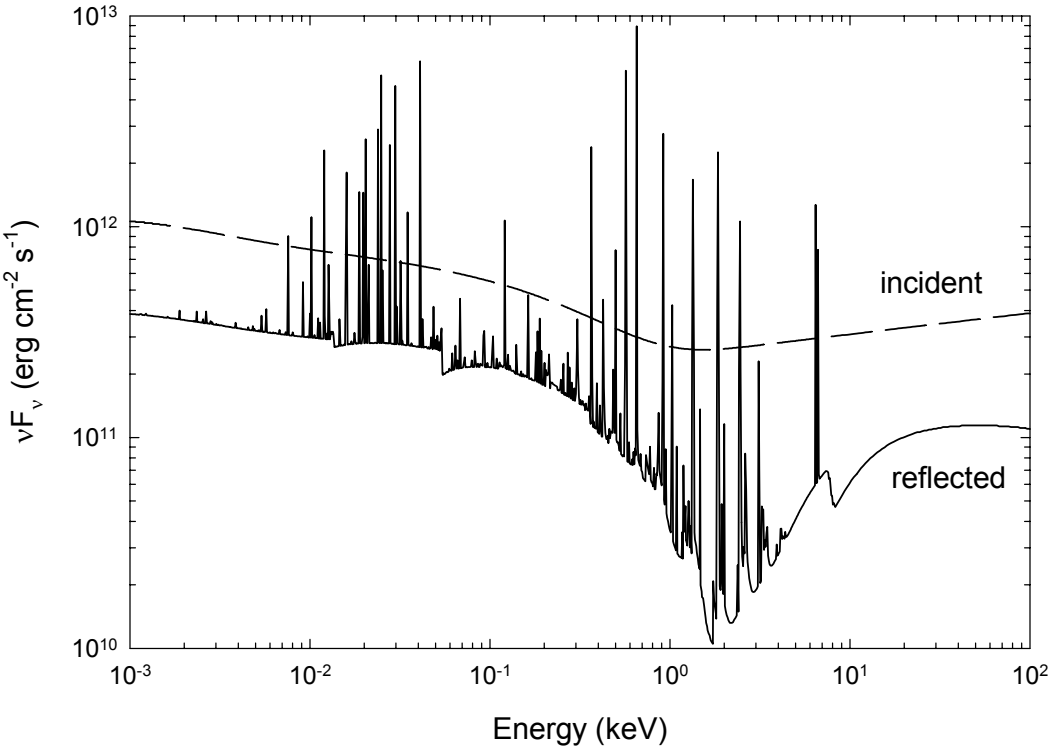
a brief introduction to Cloudy 96

introduction and commands

G.J. Ferland
Department of Physics and Astronomy
University of Kentucky, Lexington
http://www.pa.uky.edu/~gary/cloudy



Use of this program is not restricted provided each use is acknowledged upon publication. The bibliographic reference to this version of **Cloudy** is Ferland, G.J., 2002, *Hazy, a Brief Introduction to Cloudy,* University of Kentucky Department of Physics and Astronomy Internal Report.

Portions of this document have been published, and are copyrighted by, The American Astronomical Society and The Royal Astronomical Society. The remainder of this document, and the code Cloudy, are copyrighted 1978-2002 by Gary J. Ferland.

Cloudy is an evolving code. Updates are made on a roughly quarterly basis, while major revisions occur roughly every three years. You should confirm that you have the most recent version of the code by checking the web site http://www.pa.uky.edu/~gary/cloudy or asking to be placed on the Cloudy mailing list.

February 18, 2002.

Cloudy 96

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 is a brief synopsis of the input, output, and assumptions for the radiative-collisional equilibrium program Cloudy. It fully defines the commands used to drive the program and the output it generates. The methods, approximations, and assumptions used by Cloudy are outlined in Part II of this document, although this part, like Cloudy itself, is still under construction.

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 run of ionization, density, and temperature) of the gas, 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 (1988) and Aller (1984) provide definitive synopses of the basic physics governing such environments, with particular emphasis on low-density nebulae ionized by relatively soft radiation (i.e., starlight). 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), and Netzer (1990) 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 Broad Line Regions of Quasars. The temperature and density ranges of validity are described in detail in the section beginning on page 11 below. It can be used to predict either the structure or the observed spectrum from such regions.

1.3 Setting Up Cloudy

The steps required to set up the code to run on most workstations are described on the web site http://www.pa.uky.edu/~gary/cloudy . First obtain the source and data files from the web site. Next decompress and compile the source code. Compile the stellar atmosphere models and grain opacity files if they are to be used. Finally, verify the behavior of the code by running the test cases shown in Part III.

Cloudy was born at the Institute of Astronomy, Cambridge, in August of 1978. Until 1994 (through version 84) Cloudy was written in strictly ANSI - compliant FORTRAN 77 language. Version 90 was written in a mix of FORTRAN 77 and MILSPEC extensions. The current version is ANSI 89 C, and will move to C++ in the near future. Cloudy is designed to run on a variety of platforms, although it is developed and most extensively tested on Intel processors. It has been tested on a variety of machines including a Sparc, SGI, DEC Alpha, HP, WinTel, and PC Linux. It obtains similar answers on all platforms.

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 roughly 10^6 emission lines are predicted by Cloudy. These result from the specification of only a) the shape and intensity of the incident continuum, b) the chemical composition 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 incident continuum must be specified.

1.4.1.1 Continuum shape

The shape of the continuum should be fully specified between an energy of 1.001×10^{-8} Ryd ($\lambda\sim 10$ m) and an energy of 100 MeV ($\sim 1.001\times 10^{-8}$ Ryd) 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 II of this document for an exact definition.) 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 a transmitted continuum predicted by previous calculations with Cloudy. 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 built-in tables.

1.4.1.2 Continuum intensity or luminosity

The intensity of the continuum 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^{25} cm. The predicted emission-line spectrum will be given as 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* does not have its precise meaning in some of the following discussions. In the standard literature both the specific and mean intensities *I* and *J* are defined per unit solid angle. Often the word "intensity" will be used to indicate the total energy arriving from or emitted into 4π sr. In this case the

quantity is most closely $4\pi J$. I have tried to specify units whenever this distinction is important.

1.4.1.3 Combining several continua

Up to 10 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. 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 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 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 69 below), which sets the covering factor (defined on page 8 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 in several ways. A component of microturbulence can be added with the **`turbulence`** command (page 94 below). A wind model, in which case a Sobolev (large velocity gradient) model is assumed, can be computed with the **`wind`** command (page 75 below).

1.5 What is Computed and Printed

Cloudy is driven by a set of command lines. These are four-letter keywords (either upper or lower case) followed by free-format numbers that may be mixed with letters. Often Cloudy is executed as a stand-alone program. In this 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
stop zone 1 // to make a quick calculation
```

Suppose next that the code has been compiled, and the executable *cloudy.exe* created. Then, the model described by the parameters in the input file above could be computed with the command

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

It is also possible for a larger program to drive Cloudy directly by treating it as a subroutine. There are several examples of this in Part III of this document.

The program begins by echoing the input commands, except for lines beginning with a #, %, //, c_ (a space after the leading c), these lines are treated as comments. 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 spherical nebula into a set of thin concentric shells. The shells 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. Each shell is referred to as a zone, and typically ~100 to 200 zones are computed in an optically thick model. 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). 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, 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 it goes outside its range of validity, if it feels that some parameter has been miss-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 check individual models one by one.

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. 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 (others can be chosen with the **normalize** command, page 109 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 relative to 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 116 below).

Finally, the last page of the print out 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

Cloudy's development has been added by conversations with far too many people to list here. The roles of Peter G. Martin and Hagai Netzer were special, however. Peter added several of the commands that deal with ordering of supplemental line lists and the luminosity option on the blackbody command, insisted that Cloudy run on a VAX, and provided access to the University of Toronto VAX 11/780 during the 1980's, and more recently for the hospitality of CITA. Hagai and I have spent countless hours arguing over methods, assumptions, and just whose code had the bug. These comparisons are the only way to debug codes as large as Cloudy or ION. More recently Peter van Hoof has gone over the code very carefully, finding many problems, and expanding its capabilities.

Comments or suggestions which led to the improvement of Cloudy were made by the many individuals acknowledged on the web site http://www.pa.uky.edu/~gary/cloudy . Sections of the code are taken from public domain software, as acknowledged in this document and in the source. Portions of the code were written by R.F. Carswell, S.A. Cota, J. Ferguson, J. Kingdon, K.T. Korista, P.G. Martin, P. T. O'Brien, P. van Hoof, D. Verner, and K. Volk.

The development of Cloudy would not have been possible without twenty-three years of continuous support by The National Science Foundation. This began with AST 80-2522, and has been continued with grants 83-05094, 85-12414, 87-19607, 90-19692, 93-19034, 96-17083, and most recently AST 00-71180. The support of NASA through its ATP and LTSA programs has been vital. A generous allotment of time on the machines of the University of Kentucky Center for Computational Sciences is also gratefully acknowledged.

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 Osterbrock (1989). Part II 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.

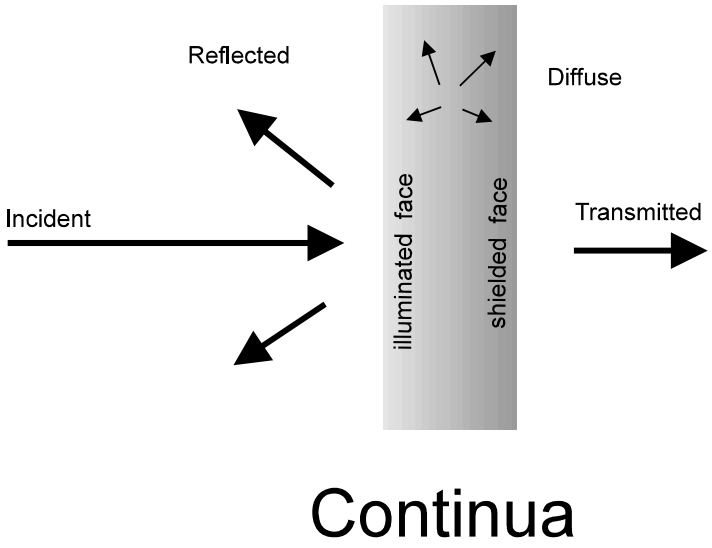


Figure 1 This figure illustrates several of the continua that enter in the calculations. continua

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. Usually, absorption of the incident continuum is the only energy source for the cloud. Within the cloud the incident continuum is diminished by extinction.

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. In a photoionized nebula the diffuse fields must be far weaker than the attenuated incident continuum and the gas albedo is generally small. As a result the radiation field is dominated by the outwardly beamed attenuated 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.

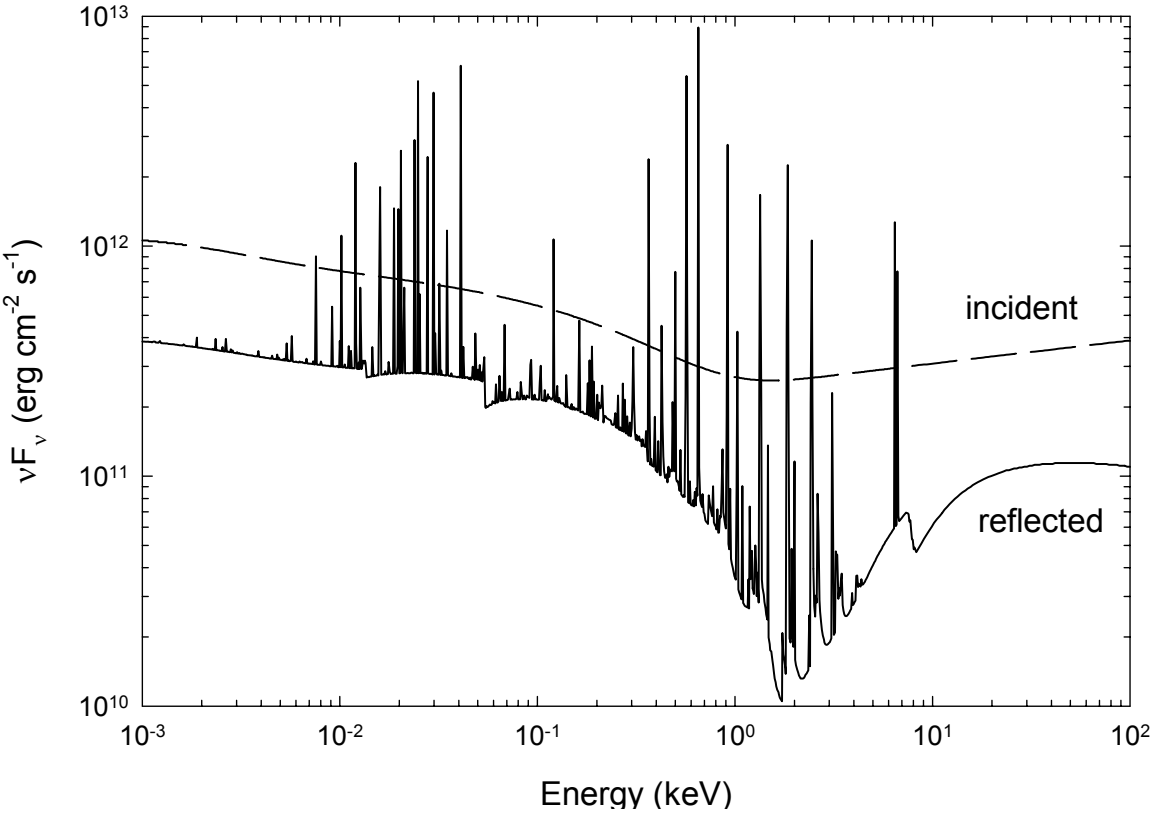


Figure 2 This figure shows the incident (dashed) and reflected (solid) continua. reflected

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 continuum and diffuse emission 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 AGN. This is a constant temperature cloud ($T=10^5$ K) with a column density of 10^{25} cm^{-2} and a density of 10^{11} cm^{-3}. It was illuminated by the ν^{-1} power law shown as a dashed line, and the reflected continuum obtained from the **`punch continuum`** command. The Compton reflector's peak at x-ray energies is clearly shown. The input stream for this model is the test file **`albedo.in`**.

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 simple wind can be computed. The summary at the end of the calculation will say whether the geometry was plane parallel (i.e., 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 in the direction towards the source of ionizing radiation is referred to as the *illuminated face* of the cloud. The opposite side of the cloud is referred to as the *shielded face* of the cloud. The illuminated face is generally hotter and more 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. The *depth* is the distance from the illuminated face of the cloud and a point within the cloud.

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 $\Omega/4\pi$ (Osterbrock 1989), 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. The predicted line *luminosities* are for a shell covering Ω sr, while line *intensities* are 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 III goes over the two covering factors that actually enter the calculations.

2.3.4 Filling factor

The *filling factor* accounts for the presence of small clumps within the emission-line region. 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) and on page

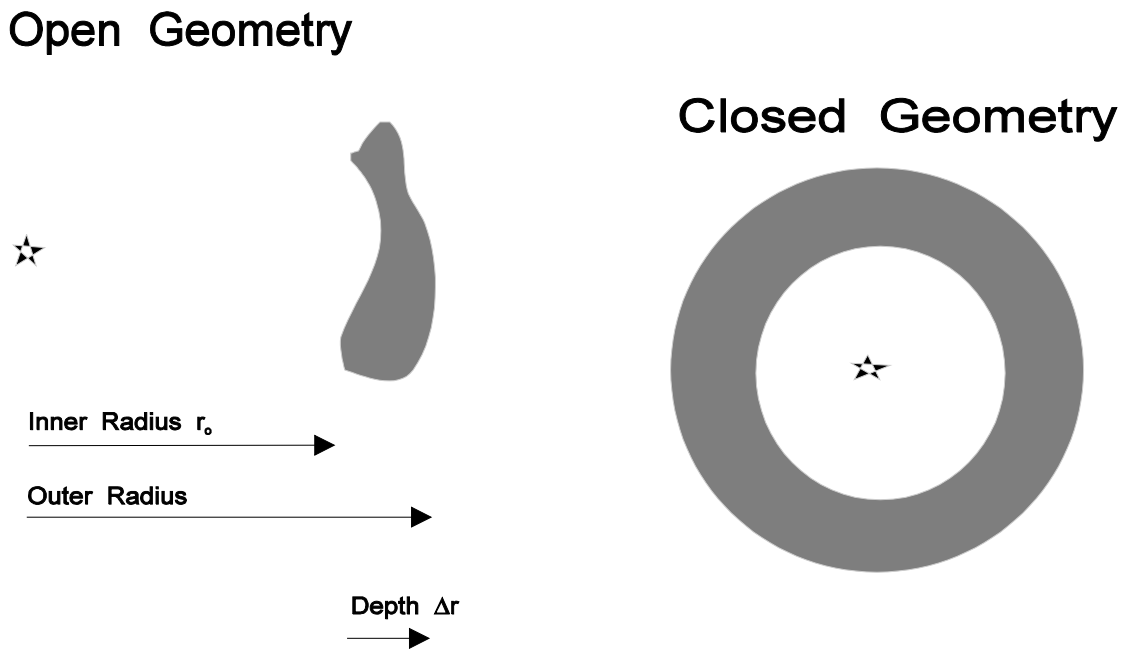


Figure 3 This figure shows the two limiting geometries that can be assumed in the calculations. The shaded area represents nebular gas. An open geometry is the default, and a closed geometry will be computed if the “sphere” command is entered. geometry

69 below.

2.3.5 Radii

The radii used here are illustrated in Figure 3. The *inner radius* is referred to as r_o, the *depth* is Δr, and the *current radius* is r.

2.3.6 Hydrogen density

The *hydrogen density* used here is the total hydrogen density (cm^{-3}), given by

$$n(H) = n(H^o) + n(H^+) + n(H^-) + 2n(H_2) + 2n(H_2^+) + 3n(H_3^+). \quad (1)$$

2.3.7 Column densities

The hydrogen column density (cm^{-2}) is given by

$$N(H_{tot}) = \int \left\{ n(H^o) + n(H^+) + n(H^-) + 2n(H_2) + 2n(H_2^+) + 3n(H_3^+) \right\} f(r) dr \quad (2)$$

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}).

2.3.8 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. Which is the best approximation largely depends on the gas covering factor. The choice mainly affects the calculation 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 the gas. This is thought to be the case in, for example, the broad-line region of active nuclei or the filaments within the Crab Nebula. In this case Lβ and higher hydrogen Lyman lines and H and He ionizing radiation can escape from the nebula. This geometry is the default condition for the code, and will be assumed if the **`sphere`** command is not specified.

Closed geometry. In a *closed* geometry emission-line gas covers ~4π sr as seen by the central object. If itar is small relative to the nebula then all diffuse fields which escape from the illuminated face of the cloud towards the central object 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 74 below).

Static vs. expanding. The **`sphere`** command has two optional arguments, **`static`** and **`expanding`**, which determine how line photons from either side of the shell interact. The **`static`** option tells the code 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 emissivity is reached. If Ω/4π~1 but the nebula is expanding then the diffuse continua interact across the nebula 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 differences in the predicted emission-line spectrum at the ≈10% level, largely because of the different treatments of the diffuse fields and line optical depths.

2.3.9 Matter-bounded and radiation-bounded geometries

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 a recombination line is set by the product of volume and density (called n^2V or the emission measure) 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 cloud 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.

2.3.10 Is a starting radius necessary?

Cloudy must be able to deduce the surface flux of photons 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 an intensity (incident energy or photon flux per unit area at the illuminated face of the cloud). In the first case it is necessary to specify an inner or starting radius, and the emission lines will also be predicted as luminosities. In the second case a starting radius need not be specified, although one may be. The lines will be predicted as intensities (energy per unit area of cloud) if the starting radius is not given. In the second case a default starting radius of 10^{25} cm will be assumed if one is not specified. This should result in a nearly plane-parallel geometry.

3 LIMITS, ASSUMPTIONS, AND RELIABILITY

3.1 Overview

This section outlines some of the assumptions and limits that define the range of validity of Cloudy. The code is designed to check that these limits are not exceeded during a calculation. This self-checking is a central feature of the code since it is designed to be used to compute large grids with thousands of models, where the examination of individual results would not be possible. Cloudy should print a warning after the last zone results if any aspects of the calculation are on thin ice.

3.2 Time Steady

Although it is possible to follow the time-dependent recombination and cooling of an optically thin cell of gas following the rapid extinction of the radiation field with the **time** command, steady-state is generally assumed. The **age** command (page 69 below) should be used to specify the age of the cloud. If the cloud age is set then the code will confirm that the time-steady assumption is valid by comparing the system's age with a host of rates and timescales, and will generate a warning if the environment is not time-steady.

Various time scales characterize the approach to equilibrium of an ionized gas (see Spitzer, 1962, and Ferland 1979 for a specific application). Generally, for an ionized gas with nebular temperatures ($\approx 10^4$ K), the longest is the H^+ recombination time scale,

$$T_{rec} = \frac{1}{\alpha_A(T_e) n_e} = 7.6\, t_4^{0.8}\, n_4^{-1} \text{ years} = 0.66\, t_4^{0.8}\, n_9^{-1} \text{ hours} \qquad (3)$$

where t_4 is the temperature in units of 10^4 K, and n_9 is the electron density in units of 10^9 cm^{-3}, and case A recombination is assumed.

For situations where molecules are important the time scales are usually far more ponderous. Generally among the longer of the time scales is the time to form H^-, an important pacesetter for H_2 formation in grain-free environments. This time scale is roughly given by

$$T_{molecule} = \frac{1}{\alpha_{rad}(T_e) n_e} = 0.3\, t_3^{-0.8}\, n_9^{-1} \text{ years} \qquad (4)$$

where t_3 is the temperature in units of 10^3 K.

Cloudy is not appropriate for the treatment of environments where conditions change more rapidly than the slowest of the equilibrium time scales.

Use the **age** command to be safe.

3.3 Atomic Database

This section outlines some of the atomic and molecular physics issues that affect the reliability of numerical simulations of nebulae. These uncertainties underscore the importance of atomic – molecular theory for the interpretation of astrophysical spectroscopy.

3.3.1 Collisional processes

By its nature, the electron temperature of a photoionized gas is low compared with the ionization temperature of the mixture of atoms and ions, as defined by the Saha equation (if the two were comparable, the gas would be collisionally ionized). Because of this, the rate coefficients describing collisional effects, such as the production of cooling emission lines, are often dominated by the cross sections near threshold. This is where laboratory experiments are difficult and ab initio quantum theory must often be used. As a result, the collision strengths undergo constant revision, towards better and more reliable values.

To cite one extreme example, the collision strength for transitions within the ^{3}P ground term of Ne^{+4} underwent three revisions between 1984 and 1991, each by a factor of 10, because of theoretical uncertainties in positions of autoionizing states with unknown energies (Lennon and Burke 1991). The intensities of all emission lines can be affected by major changes in the atomic data for only one line for some conditions. This is because (in this case) the infrared fine structure lines of Ne^{+4} can be important coolants in low-density high-ionization gasses such as planetary nebulae, and changing their cooling rate alters the thermal structure of the entire nebula. Such changes often give even models of time-steady objects such as planetary nebulae certain time-dependent characteristics.

At present, there are fairly reliable calculations of collision strengths and transition probabilities for the majority of the strong optical and ultraviolet lines in moderate ionization nebulae. A series of papers by Oliva and collaborators (see Oliva et al. 1996 and also van Hoof et al. 2000) outline observational evidence concerning accuracies in collision strengths of moderate ionization far infrared lines. This is clearly an area of uncertainty and concern.

3.3.2 Photoionization cross sections

The photoionization cross-section database has undergone a dramatic improvement with the completion of the Opacity Project (Seaton 1987) and it's fitting with analytic approximations (Verner et al., 1996). These are the photoionization cross sections used by Cloudy and they should be as accurate as 10%. All inner shell multi-electron processes are included (Kaastra and Mewe 1993) using distorted wave cross sections (referenced in Verner et al.). This part of the data base is in fairly good shape, although greater accuracy is always desired.

3.3.3 Recombination rate coefficients

Recombination from closed shell species is accurately known (Verner and Ferland 1996) since these are dominated by radiative recombination. Reliable dielectronic recombination coefficients do not now exist for most other stages of ionization. Currently there is no theory that can reproduce the best experiments (Savin et al. 1999). For these, Cloudy uses the guestamates described on page 152 below. This is clearly the greatest single gap in the atomic data base today. Savin (2000) shows an example where this uncertainty has a direct impact on cosmological studies.

3.3.4 Charge transfer

The rate coefficients for charge transfer are another uncertainty in the atomic and molecular database. This process is sometimes the dominant neutralization mechanism for singly or doubly ionized heavy elements. At present many charge exchange rate coefficients are the result of Landau-Zenner calculations using semi-empirical potential curves (Kingdon and Ferland 1996; 1999). These are thought to be no more accurate than a factor of three. Even the best quantal calculations are not thought to have an accuracy much better than 50 percent. Unpublished tests suggest that these uncertainties affect some line intensities at the ~20% level, and a few by more than this.

3.4 Continuous Opacity

All significant continuous opacity sources are treated for the energy range considered by the code, 1.001×10^{-8} Ryd to 7.354×10^{6} Ryd. These opacity sources include inverse bremsstrahlung, grains (when present), H^- absorption, electron scattering, the damping wings of strong resonance lines (i.e., Rayleigh scattering), pair production, photoelectric absorption by the ground and excited states of all ions of the lightest 30 elements, and photoabsorption by molecules. This treatment should be adequate as long as the optical depths to electron scattering are not large. Cloudy is not now designed to simulate Compton-thick regimes. (A warning will be issued after the last zone calculation if the nebula is very optically thick to electron scattering.)

3.5 Temperature Range

Cloudy assumes that the electrons are non-relativistic, which limits it to temperatures below roughly 10^9 K. Tests presented in Parts II and III show that Cloudy goes to the Compton temperature of the radiation field to great accuracy in the limit of very high levels of ionization for blackbody radiation fields with temperatures between 2.8 K and 10^{10} K. There is no formal lower temperature limit to its validity. Note that very cold gas is rarely in steady state, however.

The present range of validity of the code is approximately from 10 K to 10^9 K. Temperatures outside this range can still be treated, although with greater uncertainty. The code will not permit temperatures below 2.8 K or above 10^{10} K.

3.6 Density Range

There is no formal lower limit to the density that Cloudy can treat. The set of heavy element fine structure lines, which dominate cooling at low densities, is complete for astrophysically abundant elements, and fine structure line optical depths, continuum pumping, and maser effects are fully treated using the escape probability formalism.

There is no formal high-density limit, although the simulation is less complete at high densities. The biggest concerns are the (inexact) treatment of radiative transfer (see Avrett and Loeser 1988) and the approximate treatment of the collisional-radiative ionization processes for excited levels of the heavy elements. All species of hydrogenic and He-like isoelectronic sequences are treated as many-level atoms,

including all of the physical processes that allow the approach to LTE (see, for example, Mihalas 1978). Tests with a hydrogen density of 10^{19} cm^{-3} show that the hydrogen and helium atoms and the hydrogen molecules go to LTE at high densities. The treatment of Stark broadening for hydrogen lines follows Puetter (1981), so radiative transfer is treated correctly (in the context of the escape probability formalism) for densities above ~10^{10} cm^{-3}.

The treatment of the other 28 isoelectronic sequences is presently not as complete as the H and He-like sequences. Three-body recombination is included as a general recombination process, so the treatment of these elements is approximately correct at high densities.

Cloudy has been tested at densities of 10^{-8} cm^{-3} and 10^{19} cm^{-3} on 32-bit machines. The numerical (not physical) limit to the density will actually be set by the limits to the range of the floating point numbers allowed by the machine in use. The physics incorporated in the code imposes no lower limit to the density. The physical high-density limit is roughly 10^{13} cm^{-3}, and is set by the approximate treatment of three-body recombination - collisional ionization for the heavy elements and line transfer. Non-LTE ionization, thermal equilibria, and line transfer at high densities are areas of on-going research.

3.7 Radiative Transfer

Line intensities are predicted with stellar atmosphere conditions in mind. Radiative transfer effects, including continuum pumping and possible maser emission, are treated. Nebular approximations, such as the approximation that all atoms are in the ground state, are not made. Collisional effects, including excitation and de-excitation, continuum fluorescence, recombination, etc, are all included as general line excitation mechanisms. The treatment of level populations is designed to go to LTE in the high particle or photon density cases.

Line and continuum transfer is currently treated using escape probabilities. This is probably the weakest assumption in the present prediction of the spectrum. Work is now underway to begin the conversion to formally correct transport methods. There is no way to judge the error introduced by the escape probability approximation, although it is known to be exact if the conditions do not vary across the line forming region (Elitzur 1982).

3.8 Hydrogenic isoelectronic sequence

The 30 atoms and ions of the hydrogenic iso-electronic sequence (H^o, He^+, through Zn^{+29}) are treated as many-level atoms. The 2s and 2p states are treated separately, so Lα and 2-photon emission are computed properly in low-density nebulae. Higher levels are assumed to be well l-mixed. Up to 400 levels can be included. Tests show that the hydrogen line emissivity predicted by Cloudy agrees with Hummer and Storey's (1987) case B Hβ emissivity predictions to a few percent for all densities and temperatures (Ferguson & Ferland 1997). Details are given in Part II, and in Ferguson and Ferland (1997).

Finally, the hydrogen density used by Cloudy is the *total* hydrogen density given by

$$n(H) = n\left(H^o\right) + n\left(H^+\right) + n\left(H^-\right) + 2n\left(H_2\right) + 2n\left(H_2^+\right) + 3n\left(H_3^+\right) \quad . \qquad (5)$$

3.9 Helium-like isoelectronic sequence

The 29 atoms and ions of the helium-like isoelectronic sequence (He^0, Li^+, through Zn^{+28}) are treated as many-level systems. All *nsl* levels are resolved, and there is no limit to the number of levels that can be considered.

3.10 Helium radiative transfer

The helium line and continuum transfer problems are ones whose importance in determining the intensities of many ultraviolet lines is generally underestimated (see, for example, the discussion by Netzer and Ferland 1983). Recombinations to the He^+ ground and first excited state, and He II Lα, all ionize hydrogen, and the He II Lα line undergoes Bowen fluorescence (Osterbrock 1989; Netzer, Elitzur, and Ferland 1985). Unfortunately these continua, and especially the Bowen lines, can be the main source of photoelectric heating in the He^{++} zone of some high-ionization nebulae. Fundamental uncertainties in the treatment of the Bowen problem introduce substantial uncertainties in the local heating rate, and hence in the intensities of some ultraviolet lines, such as C III] λ1909 and C IV λ1549, because these lines are very temperature sensitive. My experience is that minor changes in the treatment of the Bowen problem typically results in ~20% changes in the intensities of these ultraviolet lines in certain low density nebulae, and in the near ultraviolet [Ne V] lines.

3.11 Ionization decreasing with depth into cloud

The code only computes abundances of ionization stages with non-trivial abundances. This is done to both save machine time and avoid numerical stability problems in the matrix inversion. The smallest fractional abundance that will be considered is specified with the **`set trim`** command (page 164 below).

The level of ionization will usually *decrease with increasing depth* into the cloud as the incident continuum is attenuated. The range of the stages of ionization to be computed is varied accordingly. This logic is not designed to detect the condition where the ionization *increases* with depth. This only happens if the gas density falls off faster than the continuum is attenuated, so that the ionization parameter *increases* with depth. The highest stage of ionization will not be increased and the result will be an artificial cap on the range of ionization.

The solution is to turn off ionization stage trimming for simulations in which the gas density falls off significantly faster than r^{-2}. This is done with the **`set trim small`** option described on page 164.

3.12 Atoms and Ions of the Heavy Elements

Most heavy elements are treated as two level systems (ground term and continuum) although photoionization from excited states is included for those cases where it is sometimes important (O^{++} and N^+ are two examples). Charge transfer,

radiative and dielectronic recombination, collisional ionization, and three-body recombination processes are included in the ionization balance.

The treatment of the heavy element ionization balance should be exact in the nebular limit, but approximate for very high photon or particle densities because of the two-level atom approximation.

3.13 Molecules

At the present time a major effort is being made to complete the treatment of the heavy-element molecular equilibria and cooling in the code. The treatment of the hydrogen molecules/ions H^-, H_2, H_2^+, H_3^+, and HeH^+ are now fairly complete and these go to LTE at high densities. The equilibrium of the heavy-element molecules OH, OH^+, CH, CH^+, O_2, O_2^+, CO, CO^+, H_2O, H_2O^+, H_3O^+, and CH_2^+ is treated following Hollenbach and McKee (1979, 1989). The predictions are thought to be correct for nebular ($n<10^8$ cm^{-3}) conditions, but do not now go to LTE in the high nucleon-photon limits. The code may have convergence problems in the fully molecular limit.

3.14 Reliability

Several issues affect the general question of the reliability of the code. The first is the effects of the bugs that surely must exist in a code the size of Cloudy. I have seldom found bugs in sections of the code older than roughly two to three years. Younger sections of the code sometimes contain bugs that only manifest themselves in exceptional situations. It is my belief that the issue of reliability in the face of complexity will increasingly be the single major problem limiting the development of large-scale numerical simulations. New methods of writing code will have to be developed if we are to take full advantage of the power of future machines. Machines are getting faster more quickly than people are getting smarter.

The second issue is the validity of the numerical methods used to simulate conditions in the plasma. Fundamental uncertainties arise for cases where the density is high ($n \gg 10^{10}$ cm^{-3}). The radiative transfer techniques used by Cloudy are approximate (see the discussion by Avrett and Loeser 1988). Unfortunately, no definitive calculation now exists for the complete non-LTE equilibrium and emission for an intermediate density (~10^{13} cm^{-3}) cloud. For less extreme conditions ($n < 10^{10}$ cm^{-3}) nebular approximations are valid, and the comparisons presented in Part III show good agreement between Cloudy and other codes designed to work in this limit. Test cases that are designed to exercise the code in well-posed limits and for certain standard nebulae are also presented in Part III. The code is well behaved and agrees with predictions of similar codes in these limits. The discussion presented in Ferland et al. (1995) suggests that 10% accuracy can be reached for the intensities of the stronger lines.

Uncertainties in the atomic database are a third concern. A great deal of progress will result over the next few years with the completion of the Opacity Project (Seaton 1987) and its extensions to the "Iron Project" (Hummer et al. 1993). Charge transfer, a collision process normally treated on a molecular basis, remains an uncertainty, and the current status of dielectronic recombination theory is a problem (Savin 2000).

In the end the uncertainties can probably best be judged by looking at both the dispersions among the various photoionization calculations presented in Part III and Ferland et al. (1995), and the changes that have occurred in the predictions made by Cloudy itself over the past few years (see also Part III). Much of the dispersion is due to improvements in the atomic database.

There can be little better way to close a discussion of reliability than to quote the warning included in Kurucz's (1970, page xiii) description of ATLAS5, a code more than an order of magnitude smaller than Cloudy:

WARNING

> "There is no way to guarantee that ATLAS5 does not contain errors. In fact, it is almost certain that it does, since the code is so long. There also may be truncation or underflow problems on computers like an IBM 360, even though all those known at present have been allowed for. We also point out that the computation of a model atmosphere should be considered a physical experiment. The program may not be able to calculate a model for conditions that do not occur in real stars or for conditions that violate the initial assumptions on which the program is based."

The comparisons presented by Ferland et al. (1995) show that predictions by the best photoionization codes agree within 15% of one another. It is not significant to fit a spectrum to better than this.

3.15 The Future

The eventual goal is for Cloudy to give reliable results for all extremes of conditions between and including the intergalactic medium and stellar atmospheres. I estimate that the code is now well over halfway complete.

Current work centers on making the code formally correct in the optically thin limit for all extremes of radiation and matter densities. Much has already been done, and present efforts center on helium, molecules, and the heavy elements.

Line transfer is now treated with escape probabilities, an approximation that is not formally correct when conditions vary across the line-forming region. A major change, to be completed within the next few years, is to transfer Lα correctly, using the proper redistribution function, using an approach similar to that of Hummer and Kunasz (1980). The two major remaining concerns will be the continuum transport (especially in the infrared) and line transfer (complete redistribution is a good approximation for most lines). Both can be treated straightforwardly using standard radiative transfer techniques, especially the accelerated lambda operator (ALO) methods now being developed.

By the time this work is complete, the Opacity and Iron Projects, and their extensions to recombination, should also be finished, and attention will return to the heavy elements. An approach similar to that now used for helium (employing several pseudo-states to allow the model atom to correctly approach LTE) will be used to ensure that the treatment of the heavy elements is correct for all densities and temperatures.

4 INTRODUCTION TO COMMANDS

4.1 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 default conditions, and then goes on to discuss the various classes of commands (i.e., those that set the continuum shape, luminosity, or the geometry).

4.2 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 described in this document are all parsed by the series of routines that have names beginning with "parse". The list of routines can be seen by listing the files "*.c". The second half of the name indicated the command that is parsed by that routine. The associated header file is **`parse.h`**.

4.3 Default Conditions

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

Table 1
Default Conditions

Quantity	Value	Command	page
default inner radius	10^{25} cm	radius	73
default outer radius	10^{30} cm	radius	73
highest allowed temperature	10^{10} K		
stop calculation when temperature falls this low	4000 K	stop temperature	107
error in heating-cooling match	0.02	tolerance	102
relative intensity of faintest line to print	10^{-3}	print faint	116
low energy limit to continuum	1.001×10^{-8} Ryd		
high energy limit to continuum	7.354×10^{6} Ryd		
limiting number of zones	600	set nend	162
total hydrogen column density	10^{30} cm^{-2}	stop column density	104
H^+ column density	10^{30} cm^{-2}	stop column density	104
H^o column density	10^{30} cm^{-2}	stop column density	104
grain mixture	no grains	grains, pgrains	88, 85
Line to continuum contrast	1000 km/s	set PunchLWidth	163
Background cosmic rays	No	cosmic rays	97
Cosmic background	No	background	35

The code is also designed to check that its assumptions are not violated, and to complain if problems occur, if its limits are exceeded, or if the input parameters are misused.

4.4 Command Format

Input and Output. When executed as a stand-alone program, Cloudy reads **stdin** for input and produces output on **stdout**. The code is also designed to be used as a subroutine of other, much larger, programs, or to generate large grids of models. In this case the input stream is entered using the subroutine calls described in a section of Part III of this document. In either case, this input stream must contain all the commands needed to drive the program. The command format rules are the same whether the code is 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. In the following examples the individual command keywords are shown extending beyond column 4, and these extra characters are completely ignored (except for some special commands that use optional keywords). The end of each line is marked either[1] by the end-of-line, a semi-colon ";", a pair of forward slashes "//", or a percentage sign "%". The command lines can be in any order, and each can be up to 80 characters long, in either lower or upper case. The input stream ends with either a blank line or the end-of-file.

Units. Most commands use cgs units. In some cases more common astronomical nomenclature can be entered (i.e., for some cases the luminosity can be specified as erg/s, in solar units, or even magnitudes). This varies from command to command, so it is important that the units be checked carefully.

Number of commands. Up to 1000 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 a later Part 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 command lines and use this copy as an input file. The input parser will handle removal of the leading spaces and asterisk.

4.4.1 Example formats

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

Square brackets indicate optional parameters. In these examples optional parameters are shown surrounded by square brackets ("[" and "]"). Examples are shown below.

```
* following needs flux density, but frequency is optional
```

[1] 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.

```
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 need not be placed on the command line and they will be totally ignored in any case.

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 should not be typed, only the space character. Only one space is needed between words.

The following is the way a command stream will be 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
```

4.4.2 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 1000 input lines.

4.4.3 Numerical input

Numerical parameters are entered on the command line as free-format numbers, and 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 is usually 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 and 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 and the second its interpretation.

4.4.4 Comments

Comments may be entered among the input data in several ways. In-line comments can be entered following a colon, semi-colon, double slash ("//"), or 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 model, as described on page 135 below.

4.4.5 Some systematics

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 linear if greater than 10. Many commands have the optional keyword **`linear`** to force temperatures below 10 K to be interpreted as the linear quantity rather than the log.

Linear vs. log for 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.

4.4.6 An example

Specific commands to describe the continuum (luminosity and shape), and geometrical details 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.

Table 2
Interpretation of Numerical Input

Typed Quantity	Interpreted as
1,000	1000
1e2	two numbers, 1 and 2
1e2,1e4	three numbers, 1 21 4
1.2.3	two numbers 1.2 0.3
100,3.141516	1 number 1003.131516
.3 3.	0.3 and 3.0

```
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
```

4.5 Filenames

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

The command parser 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.

4.6 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 initialization commands to be stored in a single place. When combined with the **`set path`** command (page 163 below) or by entering the path permanently in the file *path.c*, these commands can be easily accessed from other directories. The **`init`** command is fully described in the section beginning on page 155 below. The installation instructions suggest editing *path.c* to permanently set the path as described on page 163 below.

There is no limit to the number of commands that can be in this initialization file, other than the total limit of 1000 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 search for the file in the local directory and then on the path, as set up with the **`path`** command (described on page 163 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 58 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 home 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
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 path "c84.ini"
```

were entered.

5 DEFINING THE CONTINUUM

5.1 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 the sum of many individual continua, or it may be interpolated from a table of points.

5.2 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 at the illuminated face of the cloud or the combination of a continuum luminosity and a starting 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).

5.2.1 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 34 below.

5.2.2 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 26 below.

5.3 Combining several continua

5.3.1 Specifying a summed continuum

It is possible to combine up to 10 continua of any shape and intensity.[2] Cloudy will stop if more than 10 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.

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

[2]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.

```
black body, temperature =500,000K
luminosity (total) 37.3
*the following is a rising power law
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}.

Table 3
Commands specifying both shape *and* intensity

background
blackbody, energy density
blackbody, LTE
blackbody, luminosity
blackbody, radius
fireball
table ISM

5.3.2 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
```

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 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 luminosity 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.

6 CONTINUUM LUMINOSITY

6.1 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. These can be the quantity emitted by the central object into 4π sr (with units s^{-1}) or the surface flux at the illuminated face of the cloud (with units $cm^{-2}\ s^{-1}$).

6.2 Intensity vs luminosity commands

The incident continuum can be specified as an "intensity" (continuum incident onto a unit area of cloud) or "luminosity" (radiated by the central object into 4π sr). Each of the following commands is characterized as either an intensity or luminosity command. The words intensity and luminosity appear in quotes since these are usually not formally the quantities defined in radiative transfer texts.

The "intensity" of the predicted emission lines will be either the luminosity radiated by a shell covering Ω sr (L_{line}, erg s^{-1}) or the intensity ($4\pi J_{line}$, erg $cm^{-2}\ s^{-1}$), the energy radiated by a unit area of cloud into 4π sr. Here Ω is the angular coverage of the nebula so that $\Omega/4\pi$ (with a default value of unity) is the covering factor (page 53 below). Which is predicted depends on whether the incident continuum is specified as a luminosity or intensity.

6.2.1 Luminosity commands

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

6.2.2 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^{25} cm is assumed so that a plane parallel geometry usually results. The predicted emission-line spectrum is also given as intensities. A starting radius may be specified, and if it is, then the resulting geometry may be spherical, plane parallel, or intermediate. Both absolute and relative intensities of lines have only second-order dependence on the covering factor.

6.3 The range option

The default for many of the intensity commands is for the quantity entered to be the number of photons or luminosity *in hydrogen ionizing radiation* (1 Ryd $\le h\nu \le$ 7.354 $\times 10^6$ Ryd). Other energy intervals can be specified with the **`range`** option, an optional keyword on several commands.

The range option appears on the line specifying the luminosity or intensity and is invoked by entering the keyword **`range`**. 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 is 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. If the first optional number is negative or the keyword **_log** appears then *both* of the extra numbers are interpreted as logs. If either parameter 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 keyword **total** (equivalent to **range 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.

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 ($1 \text{ Ryd} \le h\nu < 7.354\times10^{6}$ Ryd) radiation. The ";" symbol is used to terminate the line in one case.

```
* this will use the default range, only ionizing radiation
luminosity 38 ;log of luminosity in erg/sec

* 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
```

6.4 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)

$$L_{total} = 3.826\times10^{33}\times10^{(4.75-M_{bol})/2.5} \quad \text{erg s}^{-1} \quad . \qquad (6)$$

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

$$\nu L_{\nu}\left(5500\text{Å}\right) \approx 2.44\times10^{35}\times10^{-M_V/2.5} \quad \text{erg s}^{-1} \quad . \qquad (7)$$

The conversion between monochromatic luminosity per octave and absolute visual magnitude 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.

6.5 energy density 50,000K [linear]

This specifies the energy density of the incident radiation field. The number is the equivalent energy density temperature, defined as $T_u = (u/a)^{1/4}$ 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.

6.6 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. (The quantity entered is actually $4\pi J_\nu$ since the incident continuum is normal to the illuminated face of the cloud.)

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.

6.7 intensity 8.3 [range, linear]

This specifies the integrated intensity, and is the per unit area equivalent of the **`luminosity`** command (see page 30 below). The quantity referred to as the "intensity" here is $4\pi J$. Unlike the majority of the commands, the first five characters of the line must be entered. The number is the log of the intensity (erg cm^{-2} s^{-1}) at the illuminated face of the cloud

$$4\pi \quad J = \int_{\nu_1}^{\nu_2} 4\pi J_\nu \, d\nu \quad . \tag{8}$$

The number is interpreted as the integrated intensity itself, rather than a log, if the optional keyword **`linear`** appears on the line.

The default range is over hydrogen-ionizing energies (1 Ryd ≤ hν ≤ 7.354×10^{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 roughly generated with the commands:

```
c generate 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 and then extinguishes all hydrogen-ionizing radiation.

The code will print a comment if the incident continuum is less than ten times the intensity of the Habing field.

This is an intensity command.

6.8 ionization parameter = -1.984

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

$$U \equiv \frac{Q(H)}{4\pi r_o^2 n(H) c} \equiv \frac{\Phi(H)}{n(H) c} \quad . \qquad (9)$$

Here r_o is the separation between the center of the source of ionizing radiation and the illuminated face of the cloud, *n(H)* is the total[3] hydrogen density (ionized, neutral, and molecular), c is the speed of light, *Q(H)* is the number of hydrogen-ionizing photons emitted by the central object (s^{-1}), and *Φ(H)* is the surface flux of ionizing photons (cm^{-2} s^{-1}). The number entered is the log of the ionization parameter. 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.

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

This command allows the monochromatic luminosity L_ν to be specified. The first number is the log of the monochromatic luminosity radiated by the central object into 4π sr (erg s^{-1} Hz^{-1}).

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} 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.

---

[3]Before version 65 of the code the electron density was used rather than the hydrogen density. Before version 75 *n(H)* was the atomic/ionic hydrogen density, and did not include molecules.

This is a luminosity command.

6.10 luminosity 38.3 [solar, range, linear]

The number is the log of the integrated luminosity[4] emitted by the central object into 4π sr, (erg s^{-1})

$$L = 4\pi R_{star}^2 \int_{\nu_1}^{\nu_2} \pi F_\nu \, d\nu \ . \qquad (10)$$

The number is interpreted as the linear luminosity rather than a log if the optional keyword **`linear`** appears on the line.

The default range is over hydrogen-ionizing energies (1 Ryd ≤ hν ≤ 7.354×10^6 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. 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 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.

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

This command specifies the log of the monochromatic mean intensity per octave $4\pi \nu J_\nu$ at the illuminated face of the cloud (erg s^{-1} cm^{-2}). It can be given at an arbitrary frequency although the default is 1 Ryd. The number is the log of $4\pi \nu J_\nu$, where J_ν is the mean intensity of the incident continuum.

The optional second number is the frequency in Rydbergs where $4\pi J_\nu$ 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 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.

---

[4]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.

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

This command specifies the monochromatic luminosity per octave νL_ν. The first number is the log of the luminosity radiated by the central object into 4π sr (erg s^{-1}). 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.

6.13 phi(h) = 12.867 [range ...]

This command is used to specify 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

$$\Phi(H) \equiv \frac{Q(H)}{4\pi r_o^2} \equiv \frac{R_{star}^2}{r_o^2} \int_{\nu_1}^{\nu_2} \frac{\pi F_\nu}{h\nu} d\nu \qquad (11)$$

as in Ferland, Netzer, and Shields (1979), 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.

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

The log of the total number of ionizing photons emitted by the central object (with units s^{-1}) can be specified as

$$Q(H) = 4\pi R_{star}^2 \int_{\nu_1}^{\nu_2} \frac{\pi F_\nu}{h\nu} d\nu \quad . \qquad (12)$$

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* (per unit area of cloud surface) can be specified with the **`phi(h)`** command, described on page 31 above[5].

This is a luminosity command.

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

This command allows the intensity of a second continuum source (referred to as the *current* continuum source) to be defined relative to the intensity of the *previous* continuum source. The ratio of the intensities J_ν (energy per unit frequency, time, and area) of the current to the previous continuum source is given by the first number. It is assumed to be the linear ratio unless it is less than or equal to zero, in

[5]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.

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
ratio 0.001 at 1 Ryd; power law relative to bump 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,

$$\frac{f_\nu(2\,\mathrm{keV})}{f_\nu(2500\,\text{Å})} = \left(\frac{\nu_{2\,\mathrm{keV}}}{\nu_{2500\,\text{Å}}}\right)^{\alpha} = 403.3^{\alpha} \quad . \tag{13}$$

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 α_{ox} ~ -1.4 . If X-Rays are not present then α_{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
```

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

$$\alpha_{ox} = -1.32 - 0.088 \times \log\left(\frac{L_o}{10^{28}\,\mathrm{erg\,s^{-1}\,Hz^{-1}}}\right) \tag{14}$$

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

$$\alpha_{ox} = -1.11 - 0.111 \times \log\left(\frac{L_o}{10^{27}\,\mathrm{erg\,s^{-1}\,Hz^{-1}}}\right) \tag{15}$$

and by Wilkes et al (1994):

$$\alpha_{ox} = -1.53 - 0.11 \times \log\left(\frac{L_o}{10^{30.5}\ \mathrm{erg\,s^{-1}\,Hz^{-1}}} \right) \quad . \qquad (16)$$

However, LaFranca et al (1995) find no dependence of α_{ox} on luminosity. Avni, Worrall, and Morgan (1995) find a more 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, not the ratio of current to total. 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 34 below), rather than this command.

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

6.16 xi –0.1

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

$$\xi = L / nr^2 = (4\pi)^2 \int_{1R}^{1000R} J_\nu d\nu \ \ / n \,[\mathrm{erg\ cm\ s^{-1}}] \qquad (17)$$

where n_H is the hydrogen density at the illuminated face of the cloud and r is the source – cloud separation. L is the luminosity between 1 and 10^3 Ryd. ξ is still sometimes encountered in the X-ray literature. 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 that must match the first four characters on the line.

7 CONTINUUM SHAPE

7.1 Overview

The continuum shape should be specified between the energies of 1.001×10^{-8} Ryd ($\lambda \approx 10$ m) and 100 MeV $\approx 7.354\times10^{6}$ Ryd. The low-energy continuum is important for Compton cooling, photoionization from excited states of hydrogen and helium, 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 will move into the energy range considered by the code if the electron density is higher than ~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.

7.2 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). 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 31 above). Note that there is no implicit negative sign in this exponent; typical AGN have α_{ox}~ -1.4, (Zamorani et al. 1981). The third (optional) argument is the low-energy slope of the Big Bump continuum, with the default α_{uv} = -0.5 (Elvis et al. 1994; Francis 1993). The last argument is the slope of the X-Ray component with the default α_x=-1. Optional parameters can be omitted from right to left.

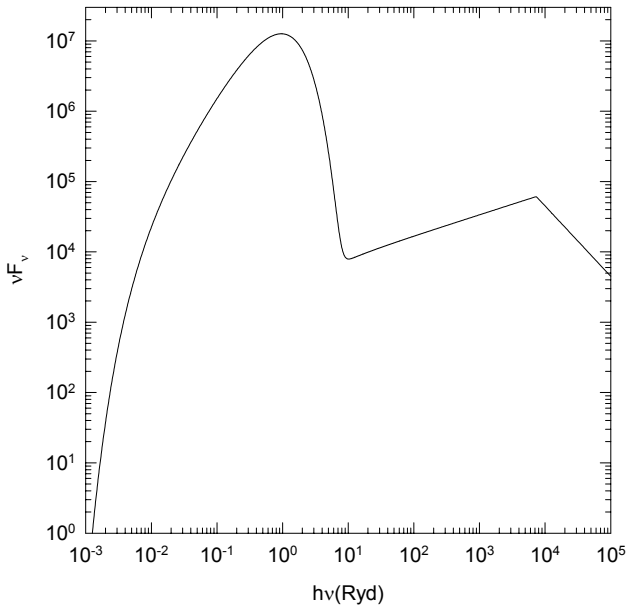


Figure 4 The continua produced by the AGN continuum command. The Big Bump peaks at 1 Ryd, while the X-Ray power law dominates high energies. The two are normalized by the second parameter, the value of α_{ox}. agncon

The full continuum is the sum of two components, as in equation 18:

$$f_\nu = \nu^{\alpha_{uv}} \exp\left(-h\nu / kT_{BB}\right)\exp\left(-kT_{IR} / h\nu\right) + a\nu^{\alpha_x} \quad . \qquad (18)$$

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 non-positive FIR intensities can be overcome by including the cosmic background with the **`background`** command (described on page 35 below), or with the **`fireball`** command (described on page 39 below).

The last term in equation 18 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}.

7.3 background, z=1.825, [f=100; no fireball]

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 α=-1 power-law, with a mean intensity J_ν at 912Å given by

$$4\pi J_\nu\left(912\text{Å}\right) = 4\pi\times10^{-21}\left(\frac{1+z}{3.5}\right)^4 \quad f \quad \text{erg Hz}^{-1}\ \text{cm}^{-2}\ \text{s}^{-1} \qquad (19)$$

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.

Primordial fireball 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

$$T_{fireball} = T_o\left(1+z\right) \quad \text{K} \qquad (20)$$

where the redshift dependence is from Peebles (1971) and the present temperature of the background is assumed to be T_o = 2.756 ± 0.016K (Wilkinson 1987). This background can be an important source of Compton cooling for low-density clouds. If the optional keyword **`no fireball`** appears on the line then the background due to the primordial fireball radiation will not be included.

Thermal background radiation can also be specified independently with the **fireball** command, described on page 39 below.

A starting radius of 10^{25} cm will be assumed is 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^{25} cm is desired.

7.4 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  .
```

7.4.1 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.

7.4.2 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.

7.4.3 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.

7.4.4 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}):

$$T_u \equiv \left(\frac{u}{a}\right)^{1/4} \quad \text{°K K} \tag{21}$$

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 $T_u \leq 2.756(1+z)$ 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.

7.4.5 blackbody, t = 50,000K, _lte

The keyword **`_lte`** (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 LTE in the high radiation density limit. (This corresponds to strict thermodynamic equilibrium, not LTE, of course.)

This is an intensity command.

7.4.6 blackbody, t = 100,000K, dilution factor = -14

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

$$W \equiv \frac{J_\nu}{B_\nu} \approx \frac{\pi R_{star}^2}{4\pi r_o^2} \tag{22}$$

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.

7.5 bremsstrahlung, temp = 8

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

$$f_\nu \propto \nu^{-0.2} \exp\left(-h\nu / kT\right) \quad . \tag{23}$$

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 97 below) with the **punch transmitted continuum** command (page 124 below) to generate a continuum which can be read in with the **table read** command (page 45 below).

7.6 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

$$f_\nu'(\nu \geq 1\,\mathrm{Ryd}) = f_\nu \left\{ \eta + (1-\eta) \exp\left(-6.22 \times 10^{-18}\, \nu_{Ryd}^{-2.43}\, N\,(H) \right) \right\} \qquad (24)$$

where *N(H)* is the total hydrogen column density, 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 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 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** command (page 124 below), then use this with the **table read** command (described on page 45 below).

7.7 extinguish optical depth 1.2, [options]

This behaves as did the previous command, except that the first number is the log of the optical depth. The linear option will force the value to the linear optical depth. If the low energy of the absorption is changed then the optical depth is at this new low energy, rather than 1 Ryd, the default.

7.8 fireball [redshift = 2000]

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

$$T_{fireball} = T_o (1+z) \quad \text{K} \tag{25}$$

where the redshift dependence is from Peebles (1971) and the present temperature of the background is assumed to be $T_o = 2.756 \pm 0.016$K (Wilkinson 1987). This command specifies *both* the shape and intensity of the radiation field. A starting radius of 10^{25} cm will be assumed if no starting radius is specified.

7.9 interpolate [ν(Ryd) or log ν(Hz)], log(f_ν)

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 2200 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 10 **interpolate** and **table** (see 41 below) commands can be entered.[6] 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 10.

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

```
* 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 46 below for a further discussion.

Cloudy will stop if more than 2200 frequency points are entered. The maximum number of frequency points allowed is set by the variable ***NCELL***.

7.10 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 45 below).

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

7.11 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 $\nu_{\text{high cut}}$ and $\nu_{\text{low cut}}$ in Rydbergs. The form of the continuum is

$$f_\nu = \nu^{+\alpha} \exp\left(-\nu / \nu_{high\ \ cut}\right) \exp\left(-\nu_{low\ \ cut} / \nu\right) \ . \qquad (26)$$

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 α_{ox}~ -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 39 above), and enter physically reasonable low-energy and high-energy continua. There is a special version of the command, **`table power law`** (see page 44 below) for entering a well-behaved power-law continuum.

7.12 table command

7.12.1 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 39 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 10 **`table`** and **`interpolate`** commands can be entered.

*7.12.2 table agn_*

If the keyword **agn_** appears (note the presence of a trailing 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.

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	-

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 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 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 31 above, or by using the **agn** command described on page 34 above.

7.12.3 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.

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)

The monochromatic luminosity at 1320 Å is νL_ν=1.84 ×10^{44} h^{-2} erg s^{-1}, where $h \equiv H_0/100$ 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
```

7.12.4 table cooling flow

The keyword **cool** generates the continuum described by Johnstone et al. (1992). 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.

7.12.5 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}).

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

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
```

7.12.6 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. 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 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 38 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 ν f_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		

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 .
```

7.12.7 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 α=-1 for the spectral midrange between 10 microns and 50 keV, and the continuum has slopes α=ν$^{5/2}$ at lower energies (appropriate for self-absorbed synchrotron, eq 6.54, p.190, Rybicki & Lightman 1979) and α=ν^{-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 ($f_\nu \propto \nu^{-1}$).

Table 8
Power Law Continuum

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

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 this infrared break free-free heating may still 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. Be aware that the **`microns`** keyword will change the meaning of both the low and high energy limits. The numbers may be omitted from right to left.

7.12.8 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 in the subsection starting on page 119 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[7]. 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
```

7.12.9 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 [NeIII] optical and IR lines.

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

7.13 table stars overview

Several sets of emergent continua from stellar atmosphere calculations are accessible to the code. 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 5 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.

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.

7.13.1 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

$$\log(L_x) = 1.08(+0.06/-0.22)\log(L_{bol}) - 9.38(+2.32/-0.83) \quad . \qquad (27)$$

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

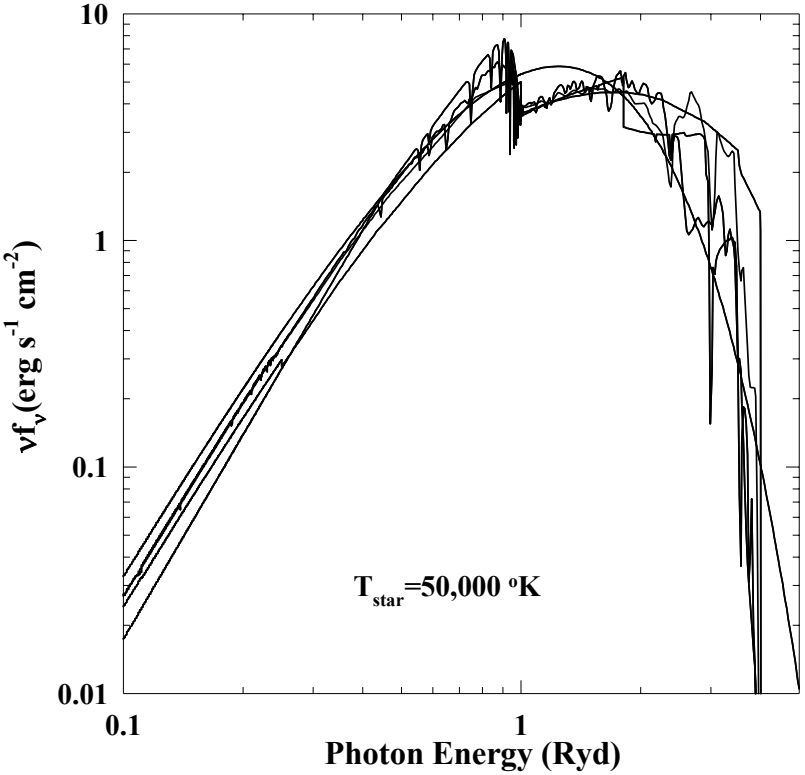


Figure 5 This figure shows the emergent continua predicted by five 50,000 K stars included with the code. The smoothest is the blackbody, and the Kurucz (1991) and Rauch (1997) atmospheres show the most structure. stars

high-energy light has little effect on conditions in the HII region, but *does* affect the ionization in the surrounding PDR.

Table 9
Mihalas (1972) Continua

T*	log(g)
30,000	4.0
32,500	4.0
35,000	4.0
37,500	4.0
40,000	4.0
45,000	4.0
50,000	4.0
55,000	4.0

7.13.2 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. Tables 9 and 10 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_ν) (Kurucz) space will be done. Extrapolation is not performed; the temperature must be between the lowest and highest values.

Table 10
Kurucz (1979) Continua

T*	log(g)	Reference
30,000	4.0	Kurucz (1979)
35,000	4.5	private comm
40,000	4.5	private comm
45,000	4.5	Kurucz (1979)
50,000	4.5	Kurucz (1979)

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

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

7.13.3 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 11 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. It will try to do something "reasonable" if extrapolation is needed.

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 147 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 with the **set path** command or by editing **path.c**, as described on page 163 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;

Table 11
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

```
* 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 will
be 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 produced by this run with the **table read** command, as described on page 45 above.

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

Kevin Volk incorporated the Rauch (1997) grid of non-LTE model planetary nebula nuclei into Cloudy. 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 12 lists the available temperatures and gravities.

Table 12
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					*

Two abundance sets are available. The default is solar and the keyword **`halo`** will invoke the low-metallicity halo abundance set. 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 try to do something "reasonable" if extrapolation on the gravity is requested.

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 147 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 163 below.

The treatment of these files is entirely analogous to that of the **`table star atlas`** command (see page 47 above). This also describes a way to save one of these atmospheres for later use.

7.13.5 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.

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			*	*

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 147 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 163 below. The treatment of these files is entirely analogous to that of the **`table star atlas`** command (see page 47).

The discussion of the **`table star atlas`** command (see page 47 above) describes a way to save one of these atmospheres for later use.

7.13.6 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 18,521K through 53,397K. 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.

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.

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				

8 CHEMICAL COMPOSITION

8.1 Overview

The composition will be solar (defined in Table 15) unless a different mixture is specified. These are taken from the meteoritic abundances of Grevesse and Anders (1989) with extensions by Grevesse and Noels (1993).

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 atomic, ionic, and molecular species. These are gas phase abundances, and do not include the amount locked into grains.

The following sections describe how to modify the chemical composition. Abundances can be specified as either *absolute abundances* by number relative to hydrogen, or as *scale factors*, relative to some standard abundance.

Table 15
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

8.2 Precedence

If the absolute abundance (by number relative to hydrogen) is specified with more than one command, then the abundance specified by the last command is used. If the abundance is specified by both its absolute abundance relative to hydrogen and by a scale factor then both will take effect. Either of the following will multiply the HII region nitrogen abundance by a factor of two:

```
abundances HII region
element nitrogen scale 2
```

or

```
element nitrogen scale 2
abundances HII region
```

since the **`abundances HII`** command sets an absolute abundance (Table 16) and the **`element`** command applies a scale factor. In the following example the first nitrogen **`abundance`** will have no effect, and the final nitrogen abundance will be the default HII region abundance

```
element nitrogen abundance -4.7
abundances HII region
```

since both specify absolute abundances. In the following only the second nitrogen scale factor has any effect since the second scale factor overwrites the first:

```
element nitrogen scale 3
element nitrogen scale 2
abundances HII region
```

and the result will be the HII region abundance set with nitrogen twice its normal value. Similarly, the combination

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

in either order would result in a nitrogen abundance of 2×10^{-4} relative to hydrogen since the first command sets an abundance of 10^{-4} and the second command doubles this. It is important to confirm that the various abundance commands interact in the expected manner by checking the composition printed in the header.

8.3 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.

8.3.1 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 58 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 I have written the element symbol before its abundance. This is only shown to indicate 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*, the log of the abundance by number relative to hydrogen. This is shown in the following examples:

```
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
```

The abundances can also be entered as a set of scale factors indicating the desired abundances relative to the current absolute abundance, usually solar;

```
abundances he =1 li=1 be=1 b=1 c=1 n=1 o=1
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
continue co =1 ni =1 cu=1 zn=1; (deplete iron)
```

It is better to use the first style since the default solar composition changes from time to time. The code checks the sign of all numbers entered to decide which style was entered. The numbers are interpreted as linear scale factors relative to solar if *all* are positive, and as logs of the abundance relative to hydrogen if *any* are negative. Be sure to check the abundances listed in the printed header to confirm that the composition has been entered correctly.

8.3.2 Setting all at once

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

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

The **metals** command, described on page 60 below, is useful for changing abundances of all elements heavier than helium.

8.3.3 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, and 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 and Shields (1978) for the classical nova V1500 Cygni. Abundances close to solar are assumed for those they did not measure.

HII 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). The keywords **HII region**, **h ii region**, or **Orion** can be used to

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(-20)	2.05(-9)	2.20(-9)	1(-10)	5.4(-11)
4	Be	1(-20)	1(-20)	2.62(-11)	4.50(-11)	1(-16)	1(-20)
5	B	8.9(-11)	1(-20)	7.60(-10)	3.40e-10	-	8.9(-11)
6	C	3(-4)	7.8(-4)	9.40(-3)	4.22(-4)	-	2.51(-4)
7	N	7(-5)	1.8(-4)	9.80(-3)	8.72(-5)	-	7.94(-5)
8	O	4(-4)	4.4(-4)	1.70(-2)	6.93(-4)	-	3.19(-4)
9	F	1(-20)	3.0(-7)	3.02(-8)	2.90(-8)	-	1(-20)
10	Ne	6(-5)	1.1(-4)	2.03(-3)	9.77(-5)	-	1.23(-4)
11	Na	3(-7)	1.9(-6)	2.06(-6)	2.25(-6)	-	3.16(-7)
12	Mg	3(-6)	1.6(-6)	3.80(-5)	3.98(-5)	-	1.26(-5)
13	Al	2(-7)	2.7(-7)	2.95(-6)	3.20(-6)	-	7.94(-8)
14	Si	4(-6)	1.0(-5)	3.55(-5)	3.76(-5)	-	3.16(-6)
15	P	1.6(-7)	2(-7)	3.73(-7)	-6.429	-	1.6(-7)
16	S	1(-5)	1.0(-5)	1.62(-5)	1.88(-5)	-	3.24(-5)
17	Cl	1(-7)	1.7(-7)	1.88(-7)	1.78(-7)	-	1.0(-7)
18	Ar	3(-6)	2.7(-6)	3.63(-6)	3.99(-6)	-	2.82(-6)
19	K	1.1(-8)	1.2(-7)	1.35(-7)	-6.869	-	1.1(-8)
20	Ca	2(-8)	1.2(-8)	2.29(-6)	2.35(-6)	-	4.1(-10)
21	Sc	1(-20)	1(-20)	1.22(-9)	1.16(-9)	-	1(-20)
22	Ti	5.8(-10)	1(-20)	8.60(-8)	9.00(-8)	-	5.8(-10)
23	V	1.0(-10)	1(-20)	1.05(-8)	9.50(-9)	-	1.0(-10)
24	Cr	1.0(-8)	1(-20)	4.84(-7)	4.80(-7)	-	1.0(-8)
25	Mn	2.3(-8)	1(-20)	3.42(-7)	3.50(-7)	-	2.3(-8)
26	Fe	3(-6)	5.0(-7)	4.68(-5)	3.38(-5)	-	6.31(-7)
27	Co	1(-20)	1.0(-20)	2.24(-9)	8.27(-8)	-	1.0(-9)
28	Ni	1(-7)	1.8(-8)	1.76(-6)	1.80(-6)	-	1.82(-8)
29	Cu	1.5(-9)	1(-20)	1.87(-8)	2.00(-8)	-	1.5(-9)
30	Zn	2.0(-8)	1(-20)	4.52(-8)	4.70(-8)	-	2.0(-8)
	grains?	Orion	AGB	no	no	no	ISM

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[8] is from Meyers et al. (1998). The grains are the default interstellar medium grains.

8.3.4 Grains and gas-phase depletions

Certain elements are heavily depleted onto grains in the ISM. This is especially true of Si, Ca, Al, Mg, and Fe in the general ISM, H II regions, and planetary nebulae. The abundance sets specified by the **h ii region**, **_ISM**, or **planetary nebula** keywords will invoke 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 88 **below**) or **pgrains** (page 85 **below**) commands.

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 (depleted) gas-phase abundances unchanged. 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.[9] This is, of course, not self-consistent.

There are two parallel grain treatments in the current version of Cloudy. These are invoked with the **grains** and **pgrains** commands, described on pages 88 below and 85 **below**. The new grains are set with the **abundances** command unless the keywords **old grains** appears. There is also a "**new grains**" option on the **abundances** command that tells the code to use the new grain treatment.

8.3.5 Dangerous interactions between the abundances and grains commands

It is possible to turn on grain species with the **abundances** command, described here, and with the **grains** (page 88 **below**) and **pgrains** (page 85 **below**) commands. Commands that turn on more grains do not turn off grains that were previously turned on. For instance the following would turn on two sets of Orion grains:

```
abundances orion
pgrains orion
```

The first command would turn on the old default Orion grains, and the second would turn on a set of size-resolved Orion grains with quantum heating. You should always check that the intended grains have been turned on. The intended grain mixture could have been produced with the following set of commands:

```
abundances orion no grains
pgrains orion
```

Another way to do this would be simply

```
abundances orion new grains
```

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

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

The **abundance** command will only wipe out grains set by previous **abundance** commands (unless there was a **grain** or **pgrain** command inbetween the two, in which case it does nothing). A preceding **grain** or **pgrain** command will override the grains from the **abundance** command (i.e., the **abundance** command will neither add nor delete grains in that case). A trailing **grain** or **pgrain** 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).

8.4 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 ~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.

8.5 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.

8.5.1 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**, or **_off** must appear on the line.

8.5.2 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 its log no matter its sign.

8.5.3 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 may be positive or negative, but is interpreted as the log of the abundance unless the **linear** keyword appears.

8.5.4 element name ionization

This allows the ionization distribution of an element to be set. Each number is the ionization fraction, $n(A^{+i})/n(A)$, of successive stages of ionization. The code will scan off up to N+1 numbers, where N 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 N+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.

8.5.5 element name off

If the keyword **_off** appears (note the leading space) then the element is turned off. 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 53 above) is used unless the element is excluded with the **elements read** command (page 53 below).

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 their large number of inner shell electrons, but tend to have small abundances and negligible effects on the thermal and ionization structure because of their low abundances.

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 code dynamically allocates memory for the species that are present when the core load is initialized. 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 at turning on an element that was initially turned off.

The keyword **_on_** turns on an element that was previously set off, in the same input stream

8.5.6 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**. The code will then read a list of elements. The list ends with a line beginning with the keyword **end**.

The ordered set of numbers entered on all **abundances** commands that follow this command will be interpreted as the abundances of these elements in the order specified. 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. An element is turned off with the **element name off** command, described on page 58 above.

The following example shows an input stream that will cause the **abundances** command in the current version of the code to behave more like version 84. The elements included in this list are only those present in that version. The **elements read** command limits the number the elements to be entered, then remaining 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.

8.5.7 element name table

If the keyword **table** appears on the **elements read** command then the code will read in a list of position-dependent abundances for a particular element. This might be used for modeling variable depletions, for instance. 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 in the list is the log of the radius (the default) or depth (if the keyword **depth** also appears). 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 the keyword **end** for the last line. Up to 500 pairs may be entered. This command always specifies the absolute abundance and not the scale factor.

When this command is used, the chemical composition printed when the code initializes is 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.

8.6 fluctuations abundances, period, max, min, phase

This command specifies a model in which the metallicity varies as a sine wave over the radius. 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 60 below).

The **fluctuations** command is more fully described in the description of the density version, on page 65 below.

8.7 grains

See page 88 below.

8.8 metals 0.05 [_log, linear, grains; deplete]

This command multiplies 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 are to be investigated. If the number is zero or negative then it is assumed to be the log of the number. If it is positive then it is interpreted as a linear scale factor. If the **linear** keyword appears then the number is interpreted as linear (unless negative). If the **_log** keyword appears then the number is interpreted as the log of the metal abundance, no matter what sign the number has.

Combinations such as

```
abundances planetary nebula
metals 3
```

or

```
metals 3
abundances planetary nebula
```

would multiply the planetary nebula gas-phase abundances by three,[10] while

```
metals -10
```

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

8.8.1 Scaling grains and metals together

It seems likely that the grain to hydrogen ratio somehow scales with the total gas-phase metallicity. The optional keyword **grains** on the **metals** command (page 60 above) 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 turn on 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 solar 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  .
```

Table 17
Depletions

	Factor	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

8.8.2 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 (~1 cm^{-3}) is built into the code. Table 17 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 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** or **pgrains** commands, to specify an arbitrary set of grain and gas-phase

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

abundances. Specifying grains by themselves (with the **grains** or **pgrains** commands) does not change the gas-phase abundances, which is not self-consistent. The code will complain if you do this, but still compute the model.

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 17, 16, and 15 are not self-consistent since they come from different sources. Physically, there is a growing suspicion that the total ISM metallicity (gas and grains) does not amount to a solar metallicity (Snow and Witt 1996) so that depletion factors have been systematically overestimated. Clearly this is an area of active research.

8.9 pgrains

See page 85 below.

9 DENSITY LAWS

9.1 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 (atomic, ionic, and molecular, given by the command **hden** described on page 66 below) is kept constant. Many other density or pressure distributions can also be computed.

9.2 constant density, pressure, gas pressure

The **constant xxx** command specifies what quantity is to be kept constant across the computed structure. 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.

9.2.1 constant density

This is the default. The hydrogen density, the sum

$$n(H) = n\left(H^o\right) + n\left(H^+\right) + n\left(H^-\right) + 2n\left(H_2\right) + 2n\left(H_2^+\right) + 3n\left(H_3^+\right) \qquad (28)$$

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 66 below).

9.2.2 constant gas pressure [index =-1]

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

$$P_{gas} = n_{tot} k T_e \qquad (29)$$

where n_{tot} is the total particle density, is kept constant. This type of model is trendy today in active nuclei, but not really warranted since the ionizing continuum is seldom constant over the sound travel time across a typical cloud. (The latter is several months for standard BLR clouds, and constant pressure is really only approached after conditions have been stable for several sound travel times.) The optional index α will force the pressure to change as a power-law of the radius;

$$P_{gas}(r) = P_o \left(\frac{r}{r_o} \right)^{\alpha} \qquad (30)$$

where P_o is the pressure at the illuminated face of the cloud.

The results of this command are not exact. The actual gas pressure will vary from zone to zone with an rms scatter of typically 0.7 percent.

9.2.3 constant pressure [no continuum, no abort]

If you are holding the pressure constant, you really should hold the total pressure, particle and radiation, constant. This option turns on the physics discussed by Ferland and Elitzur (1984) and Elitzur and Ferland (1986).

The pressure is the *total* pressure, the sum of the gas and radiation pressure,

$$P_{tot}(r) = P_{tot}(r_o) + \int a_{rad}\,\rho\,dr = P_{gas} + P_{lines} + P_{continuum} \qquad (31)$$

where a_{rad} is the radiative acceleration [cm s^{-2}] due to the incident continuum and ρ is the density (gm cm^{-3}). This pressure law includes thermal gas pressure (P_{gas}), the nearly isotropic pressure due to trapped emission lines (P_{lines}), and the outward force due to the attenuation of the incident radiation field. The latter is the integral, referred to as $P_{continuum}$. Specifying the **`no continuum`** option on the command line can turn this off.

Turbulent and magnetic pressures are not included in the equation of state since they will either be negligible or dominate the pressure. In the former case these pressure terms would be trivial, and in the latter it would be impossible to determine the particle density. These extra pressures would add terms $\rho v^2/2$ and $B^2/8\pi$ to equation 31.

Cloudy will normally 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, escape probabilities, and level populations, reliably. If more than one iteration is to be 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 results of this command are not exact. The actual total pressure will vary from zone to zone with an rms scatter of typically 0.7 percent. It is not possible to specify a power-law index for this pressure law.

9.3 dlaw [options]

An arbitrary density law, specified by the user, will be used. There are two forms of this command. The default calls a routine provided by the user and the second interpolates 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 entered command will be honored.

9.3.1 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 evaluate this function as needed to determine the density as a function of depth. The arguments of the function are the radius (distance from the current location to the center of symmetry), and the depth (distance from the current location to the illuminated face of the cloud). 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.

```
/*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] );
}
```

9.3.2 *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. The first is the log of the radius or depth (in cm) and is followed by the log of the hydrogen density (cm^{-3}). If the keyword **`depth`** appears on the **`dlaw table`** command 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 otherwise. 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 option will 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 leading space on any lines with the numbers – the code will think that an end of file has been read. 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).

9.4 fluctuations period, max den, min den, phase

This command specifies a model in which the density varies as a sine wave. This is designed to investigate the effects of inhomogeneities upon the emission-line spectrum (see Mihalszki and Ferland 1983; Kingdon and 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

$$n(r) = \left(\frac{n_{\max} - n_{\min}}{2}\right) \times \cos\left(depth \times \frac{2\pi}{P} + \varphi\right) + \left(\frac{n_{\max} + n_{\min}}{2}\right) \qquad (32)$$

where n_{max} and n_{min} are the maximum and minimum densities and ***depth*** is the depth into the cloud.

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

9.5 globule [density =2, depth =16, power =2]

The density law resulting from this command 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

$$n(r) = n_o \left(\frac{R_{scale\,depth}}{R_{scale\,depth} - \Delta r}\right)^{\alpha} = n_o \left(1 - \frac{\Delta r}{R_{scale\,depth}}\right)^{-\alpha} \qquad (33)$$

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, $R_{scale\ depth} = 3.086\times10^{18}$ 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[11] $\alpha = 1$. The arguments can be omitted from right to left.

9.6 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

$$n(H) = n\left(H^o\right) + n\left(H^+\right) + n\left(H^-\right) + 2n\left(H_2\right) + 2n\left(H_2^+\right) + 3n\left(H_3^+\right) \quad \mathrm{cm}^{-3} \quad . \qquad (34)$$

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

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(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.

9.6.1 Power-law radial dependence

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

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

i.e.,

$$n(r) = n_o(r_o)\left(\frac{r}{r_o}\right)^{\alpha} \quad \text{cm}^{-3} \quad . \tag{35}$$

In this example n_o will be 10^9 cm^{-3}. It is the density at the illuminated face of the cloud. The optional power law is relative to the distance to the central object, not the depth into the cloud. If α= -2 (i.e., a power law with index alpha =-2 is entered as in the example above), then the density will be proportional to the inverse square of the distance to the central object. Spherical models will tend to have the same ionization parameter (and hence physical conditions) across the ionized zone.

9.6.2 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,

$$Q_{crit}(H) = \alpha_B(T_e) n_o^2 \, 4\pi \, r_o^2 \quad \text{s}^{-1} \tag{36}$$

where a hydrogen ionization front will not be present and the model will extend to infinite radius. 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. The runaway occurs when $Q(H) \geq Q_{crit}(H)$. 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 radius. Cloudy is not now designed to treat this case. 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 α ≤ -2, since, for most circumstances, the cloud will remain hot and ionized to infinite radius and zero density.

9.6.3 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 depth is the distance (in cm) between the current position and the illuminated face of the cloud. The radius is the distance between the current position and the center of symmetry of the system. With this command the density is given by

$$n(r) = n_o(r_o)\left(1+\frac{\Delta r}{R_{scale}}\right)^{\alpha} \quad \mathrm{cm}^{-3} \tag{37}$$

where R_{scale} is the scale depth and Δr is the depth. The scale depth is entered as the third number on the line, and is the log of the scale depth in centimeters.

9.6.4 Power-law dependence on column density

The local hydrogen density will depend on the 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

$$n(r) = n_o(r_o)\left(1+\frac{N(H)}{N(H)_{scale}}\right)^{\alpha} \quad \mathrm{cm}^{-3} \tag{38}$$

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

10 GEOMETRY

10.1 Overview

This section describes commands that determine the geometry of the emission-line region.

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.

10.2 age 44 years [_off]

The code assumes that the cloud is old enough for atomic processes to have reached steady state. The **age** command allows the code to confirm that the computed cloud is indeed time steady. The number on the command line is the age of the cloud. The default units are linear seconds. The keyword **_log** will force the code to interpret the number as a log. The default units are seconds, but keywords **minutes**, **days**, **weeks**, **fortnights**, **months**, **years**, **centuries**, and **millennia** are also recognized.

During a calculation the code keeps track of many equilibrium timescales. 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 the keyword **_off** appears then the age will not be checked.

10.3 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 the nebular structure and assumes that the lines are optically thin.

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

$$Q_\alpha(\lambda) = C_\alpha D_\alpha \int \left(\frac{r}{r_o}\right)^\alpha \varepsilon(\lambda)\ \ dr\ \ , \tag{39}$$

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

$$\begin{aligned} C_0 &= 2 \\ C_1 &= 2\pi r_o \\ C_2 &= 4\pi r_o^2 \end{aligned} \quad . \tag{40}$$

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

$$\begin{aligned} D_0 &= 1/2, 1 \\ D_1 &= \Theta/2\pi \\ D_2 &= \Omega/4\pi \end{aligned} \quad . \tag{41}$$

For α = 2 this is the familiar definition. In the long slit case 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 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).

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 cm$^{\alpha-2}$ s^{-1}. 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).

$$F_\alpha(\lambda) = \frac{A_\alpha}{4\pi D^2} Q_\alpha(\lambda)\ \ , \tag{42}$$

with

$$\begin{aligned} A_0 &= \Omega_b D^2 \\ A_1 &= \Theta_s D \\ A_2 &= 1 \end{aligned} \quad . \tag{43}$$

Here F_α is the observed flux in erg cm^{-2} s^{-1}, D is the distance of the object in cm, 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 relation between the observed surface brightness $S(\lambda)$ in erg cm^{-2} s^{-1} arcsec^{-2} and $Q_o(\lambda)$ is simply (the intensity case implies α = 0):

$$S(\lambda) = \frac{\Omega\left(1 \text{ arcsec}^2\right)}{4\pi} Q_0(\lambda) \approx 1.87043 \times 10^{-12} Q_0(\lambda) \quad . \tag{44}$$

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

10.4 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 can also be set as an optional argument on the **sphere** command (see page 74 below).

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. 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). The **covering factor** and **sphere** commands set the same variables. If both are used in the same input stream only the second will be honored.

10.5 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 6 gives an example of the assumed geometry.

The inner and outer radii of the cylinder are set by the **radius** command described on page 73 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 is given by

$$dV = 4\pi r_o^2 \left(\frac{r}{r_o}\right)\left(\frac{\min\left(r, h_{cyl}\right)}{r_o}\right) f(r)\, dr \tag{45}$$

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 45 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.

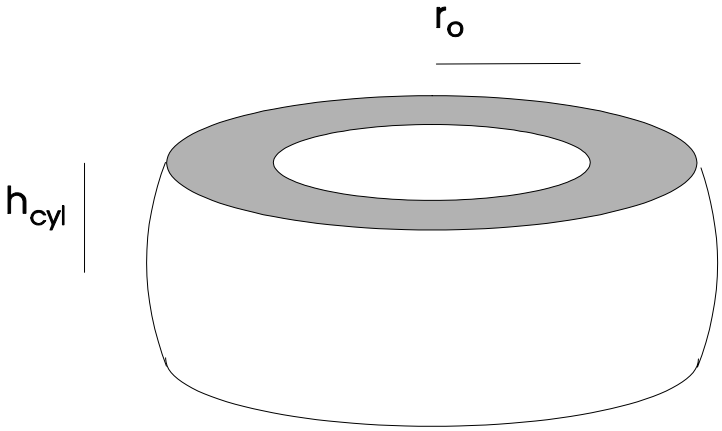


Figure 6 This figure shows the geometry assumed when the cylinder command is used. cylin

10.6 distance 3.2 linear parsecs

This command sets the distance to the object from the Earth. The number is the log of the distance is 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 from the Earth. The code must have enough information to predict emission line luminosities (see the discussion on page 26 above) to do this. If the code can predict luminosities and the distance is set, then the emission line fluxes will be printed if the **`print flux`** command (113 below) is also entered.

10.7 filling factor = 0.05 [index =-1]

The first number is the filling factor 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 (in which case it will be less than or equal to zero). The second number is optional, and is the index α for a power-law variation of the filling factor $f(r)$, i.e.,

$$f(r) = f(r_o)\left(\frac{r}{r_o}\right)^{\alpha} \tag{46}$$

where $f(r_o)$ and r_o are the filling factor and inner radius of the cloud.

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

$$dJ = 4\pi\, j\, f(r)\, dV \frac{\Omega}{4\pi} \tag{47}$$

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

$$d\tau = \alpha_{l,u}\left(n_l - n_u \frac{g_l}{g_u}\right) f(r)\, dr \tag{48}$$

(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 if a filling factor is set in a constant pressure model since this makes no physical sense.

10.8 illuminate 45 deg [radians]

This will simulate a plane parallel slab 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.

10.9 radius log r(inner) [outer radius, thickness; parsec; linear]

The first number is the log of the inner radius. The second number sets a stopping radius and is optional. The second number 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 on the line then the numbers are interpreted as the linear numbers rather than a log. The default units are centimeters, but the arguments will be interpreted as the log of the radii in parsecs if the keyword **parsec** appears anywhere on the line. 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 value for the outer radius is effectively infinite (actually 10^{30} cm). If the **radius** command is not entered and the surface flux or ionizing radiation field is set in some unambiguous manner (for instance, with the ionization parameter or energy density temperature), then a radius of 10^{25} cm will be set by default. Under most circumstances this radius will result in an effectively plane parallel geometry. Page 144 below describes a problem that can occur if the second parameter is used with the **vary** option.

The stop thickness command (page 107 below) provides another way to set a stopping thickness, but without needing to specify a starting radius.

10.10 sphere [options][12,13]

Cloudy normally assumes that the gas covering factor is small, as is the case in the BLR of AGNs. The **sphere** command should be included to change this assumption if 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 to use **sphere** or not. In this case it would be best to compute models with and without **sphere** set and compare results. In most cases this will only make a 10 – 15% difference in predicted quantities.

10.10.1 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. First, 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 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.

---

[12] 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.

[13] 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 only the **covering factor** command changes the covering factor.

Otherwise it is 10^{-20}. The total optical depths of lines are twice their computed depth. Finally, ionization by lines 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.

10.11 stop depth …

10.12 stop thickness …

These commands provide methods to set the thickness of a cloud without specifying its radius. They are described on page 107 below.

10.13 wind v=300 km/sec [mass =1.4]

The model will be a large velocity gradient (v ~ R Sobolev approximation) wind. The line widths and escape probabilities are modified in the appropriate manner, i.e., the effective line optical depth is given by

$$\tau_{l,u}(R) = \alpha_{l,u} \min(r, \Delta r) \left(n_l - n_u \frac{g_l}{g_u} \right) \left(\frac{\nu_{th}}{\max(\nu_{th}, \nu_{\exp})} \right) \qquad (49)$$

where v_{th} and v_{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.

The first parameter on the command line is the expansion velocity v_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/sec. 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 $\rho(r)r^2v(r)$ is kept constant). Because of this, a filling factor would not make physical sense and should not be used. The optional second parameter is the mass of the central star in solar units; its default value is one solar mass.

10.13.1 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 II.

A *negative velocity* simulates the flow from a weak-D H II region. This physics is currently under development and should be used only for experiments. In particular this solver cannot go through a sonic point, and is best used for weak-D flows. The central object mass is set to zero.

10.13.2 wind advection, v=-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 gas flow speed in km/s and the. This physics is being developed in collaboration with Robin Williams, Will Henney, and Jane Arthur, and is not now fully functional.

10.13.3 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.

11 OPTICAL DEPTHS AND RADIATIVE TRANSFER

11.1 Overview

In some classes of nebulae, such as H II regions and planetary nebulae, line transfer is relatively unimportant. In other objects, such as nova shells and the broad-line region of active nuclei, 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. Cloudy presently treats line radiative transfer with escape probabilities. Further details are given in Part II 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. Cloudy is fairly fast, so there is no reason not to iterate at least one time when line transfer is important. 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, so that the proper escape probabilities can be computed. More iterations are generally needed when subordinate lines are also optically thick. The program has an **`iterate to convergence`** command (page 93 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 meaningful only 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 transfer solution.

11.2 atom options . . ..

The **`atom`** command is used to change the treatment of one of the model atoms. Currently the keywords, **`FeII`** for the large FeII atom, **`H-like`** for the hydrogenic isoelectronic sequence, **`He-like`** for the helium-like isoelectronic sequence, and **`rotor`** for molecules that are treated as simple rotors, are recognized.

11.3 atom rotor [options]

The code has a model rigid rotor that is used to treat molecular rotational transitions. This is a single implementation that includes ^{12}CO and ^{13}CO. Yes, I know that CO is not an atom. But here it is.

11.3.1 atom rotor levels 25

This command sets the number of rotation lines (not levels) in the model 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 atom can extend up to any line with a lower level J >1. The number of levels is limited only by the available heap memory and compute time. The default number of rotation levels is 20. Increasing the number of levels allows a better representation of the collision physics and radiative transfer at the expense of longer execution times and greater memory requirements.

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. Higher levels will carry the cooling 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.

The number of levels can only be set once at the very start of a calculation. This is because space is allocated for the rigid rotor line arrays only one time per core load. 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.

11.4 atom H-like [options]

This is used to change the treatment of atoms of the hydrogenic isoelectronic sequence.[14]

11.4.1 atom h-like collisions . . . .

Collisional processes between levels of the hydrogenic atoms and collisional ionization can be turned off with this command. This is mainly used for debugging the hydrogenic model atoms. 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.

atom h-like collisions l-mixing[15] This command turns off *l*-mixing 2s-2p collisions in the hydrogenic sequence.

atom h-like collisional ionization off This command turns off collisional ionization of all levels in the hydrogenic sequence, except for the very highest level. Collisional ionization from the highest level is not turned off to allow the atom some coupling to the continuum.

atom h-like collisional excitation off This command turns off collisional excitation of all levels in the hydrogenic sequence, 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.

---

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

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

Warning! The code will require a very number of zones if collisions are turned off in an optically thick cloud with a very large ($n \gg 15$) hydrogen atom. Collisions will normally hold populations of very highly excited levels to values very near LTE. As a result the FIR and radio lines will have very small line optical depths. 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. The effect is that 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 very large atom.

11.4.2 atom h-like damping off

Continuum scattering due to the extreme damping wings of Lyman lines (i.e., Rayleigh scattering) 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 ($N(H^o) > 10^{23}$ cm^{-2}).

11.4.3 atom h-like levels 15 [element iron]

The number of levels in the model hydrogenic atoms is set with this command. The atom can extend up to any principle quantum number $4 < n \leq 400$. The size is limited mainly by the available heap memory and compute time. Note that there will actually be $n+1$ levels in the calculation since the $2s$ and $2p$ states are treated separately. The default highest quantum level is 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 **`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.

The default behavior is for this command to change the behavior for 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. This is because space is allocated for the hydrogenic arrays only one time per core load. If the code is used to run a grid of models only the first occurrence of **`atom h-like levels`** will be honored and all following occurrences 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 and would be very slow on today's computers. It is best to set only the most important elements to large levels.

11.4.4 atom h-like lowest temp 200

Normally the level populations are determined by solving the equations of statistical equilibrium using departure coefficients. These diverge at low temperatures when hydrogen is ionized, in which case either level populations or fits to Martin's (1988) results are used. The lowest temperature considered by the hydrogen atom is machine dependent, and can be altered with this command. The default value of the lowest temperature is T_{low} = Z 1000 K where Z is the atomic number. The number is interpreted as the log of the lowest temperature if it is less than or equal to 10 and the linear temperature otherwise. Tests show that numerical instabilities in the matrix inversion routine limit the lowest temperature to only slightly below the default value, even with 64 bit words.

11.4.5 atom h-like redistribution [alpha, resonance, subordinate] [[_PRD, _CRD, CRDW]

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 once 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).

11.4.6 atom h-like TopOff 6 [_add scale]

This sets the lowest level within the hydrogen atom 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**.

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 maser effects. The command changes the lowest level with the extra recombination. The code will not predict any lines that include these disturbed levels.

11.4.7 atom h-like matrix [lowt, departure, populations]

This tells the code which form of the level populations to use. The options are **lowt** for the low temperature solution and **populations** for the populations themselves. The code normally decides this based on experience, but this command provides an option to override the default choices.

11.5 atom he-like [options]

11.5.1 atom he-like levels 4 [element iron]

This command sets the number of levels in atoms and ions along the helium isoelectronic sequence (He^0 through Zn^{28+}). The argument is the highest principal quantum number n, and must be three or greater.

If no number appears on the command, but the keyword **large** or **small** does, then either n = 20 or 3 levels will be used. These keywords provide a version-independent method of insuring that the code uses a large or small number of levels. The atoms are coded so that there is no limit to the number of levels that can be included, other than the memory and compute time requirements.

The model atom resolves the atom 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 $n_l = n^2 + n + 1$ l levels included. The default number of n levels for He^0 is 6, resulting in 43 l levels, n = 4 for 21 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, you need 7, 13, 21, 31, and 43 l -levels. 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.

The default behavior is for this command to change the behavior for all elements along the helium-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 calculation. This is because space is allocated for the line 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. Only lines coming from the first n - 1 levels will be printed at the end of the calculation.

11.5.2 atom he-like collisions ….

Collisional processes between levels of the helium-like ions and collisional ionization can be 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[16] This command turns off collisions within the same n level for all elements in the helium-like isoelectronic sequence. For $n > 2$ this is mainly l-mixing collisions, while for transitions within $n = 2$ electron exchange collisions are disabled as well.

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.

atom he-like collisions fake Quantal calculations of collision strengths only exist for lower quantum levels. The code uses "reasonable" estimates of the collisions strength for the remaining transitions. This command will set these "fake" values to zero, as a means of testing their effects on the spectrum.

11.6 atom feii [options]

This command determines which model FeII atom is used. The default is the simplified and very fast scheme outlined by Wills, Netzer, and Wills (1985). When the **`atom feii`** command is entered the code employs the large FeII atom developed by Katya and Dima Verner and described in Verner et al. (1999). This atom is far more accurate but also much slower.

N.B. – there is no space between the element symbol and the spectroscopic designation. **`Fe_II`** will not work. Note that the keyword is **`FeII`** and not Fe2, to avoid scanning the number 2 off the command line.

11.6.1 atom feii levels

The optional keyword **`levels`** will change the number of levels used by the model atom. The upper limit (and default) is 371 levels. As few as 16 levels can be computed. 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.

---

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

11.6.2 atom feii print

The keyword **print** will turn on debugging printout for each call to the model atom.

11.6.3 atom feii redistribution [resonance, subordinate] [_PRD, _CRD, CRDW]

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**. 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.

11.6.4 atom feii simulate

The keyword **simulate** will cause results from the atom to be simulated. The large model atom is not actually called. This is very fast and intended only for debugging.

11.6.5 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.

11.6.6 atom feii output options

FeII 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 now available for understanding output from the Fe II atom. Most often FeII 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.

FeII bands in the main output. A series of FeII "bands" are automatically entered into the main emission line output when the large FeII atom is used. Each band represents the total FeII emission integrated over all lines that lie within a band of wavelengths. The band 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 *Fe2Bands.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 FeII bands that can be specified.

11.6.7 punch Verner

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.

11.6.8 punch FeII continuum

This option punches the total FeII emission as a continuous spectrum. Here a wavelength interval is broken into a number of intervals and the total FeII 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 122 below.

11.6.9 atom Feii continuum low=1000 high=7000 ncells=1000

The lower and upper bounds and the resolution of the FeII continuum punched with the **`punch FeII continuum`** command can be adjusted with the **`atom feii continuum`** command described here. The numbers are the lower and upper limits to the wavelength range in Ångstroms and the desired number of intervals.

11.7 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 behavior of the hydrogen and helium atoms in the case B limit.

With no options this command sets the inner optical depth for hydrogen and helium Lα to 10^5 so that even a one-zone model will be close to case B[17]. The optional number is the log of the Lα optical depth, so it is possible to change this assumption. 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 **`caseb`** command suppresses excited states line optical depths.

Normally the treatment of the hydrogenic sequence includes all collisions between the levels considered for each atom or ion. 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 n ≥ 3 levels *are* always included unless the **`hydrogen collision off command is`** given. Collisions between the 2s and 2p states are always included unless the **`no 2s2p`** command is given.

The large Lyα optical depth will often result in an especially strong radiation field with the line. Photoionization from excited states of H and He can become very important as a result. The **`no photoionization`** option on the **`caseb`** command tells the code not to include photoionization from excited states.

In the case of the helium-like isoelectronic sequence the **`caseb`** command sets the optical depths in the singlet Lyman lines to a large value. The Hummer & Storey option turns off collisions from $1\ ^1S$, $2\ ^1S$, and $2\ ^1S$.

There are several side-effects of this command that may after the spectrum or physical conditions in unexpected ways. Optically thin gas is actually described by Case C (Ferland 1999). In Case C continuum pumping enhances Balmer lines. The

---

[17] Before version 96 the default optical depth was 10^9. This caused extreme Lyα behavior in the H II region. The new version is a better estimate of the actual physics that occurs in the H II region.

large Lyman line optical depths that result from the **caseb** command will prevent continuum resonant pumping of the atom. The large optical depths also result in a large $\bar{J}$ within Lα even though the Lα emergent intensity is small, unless the **no photoionization** option is used. This can have a large effect on the level of ionization of atoms and first ions of many third-row species, where Lyα can photoionize atoms. Beware.

11.8 diffuse fields [outward, ots]

This command 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**. If **outward** is chosen then the code will check for a number. This determines which of the many outward only approximations is used. The default[18] is 2.

This choice does not strongly affect the predicted emission-line spectrum, but it does change the temperature at the illuminated face of the cloud. Which method is used is controlled by the three character variable **chDffTrns**. It can have any of the values "OTS", "OU1", or "OU2", etc, depending on which approximation is used. These approximations are described in Part II of this document.

11.9 double optical depths

This command simulates 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 dimension, column densities, or line emission *are not* affected.

These approximations only make 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.

11.10 pgrains abundance, [qheat]

11.10.1 Overview

There are now two independent **grains** commands. The **pgrains** command described here was introduced in version 96 and gives a far better treatment of grain physics at the expense of longer compute times. The older **grains** command is described in the following subsection beginning on page 88 below. In a future

[18] OTS was the default in version 86 and before.

version of Cloudy the **pgrains** will become the default **grains** command, the older **grains** command will be renamed **ograins**, and it will eventually be removed.

The **abundances** command takes precedence over the default grains set with the **pgrains** command. The grains set with an **abundances new grains** command will completely override all parameters set with previous **pgrains** commands.

This physics invoked by this command was developed in collaboration with Peter G. Martin and Peter van Hoof, and Peter van Hoof did the majority of the coding. 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 **pgrains** command has the ability to resolve this (and other) size distribution into several size bins and treat these either classically or with single-photon (quantum) heating. The older command uses quantities that are averaged over the size distribution and ignores quantum heating for the majority of the species. This was needed since it was computationally impractical to do the complete physics at the time the command was implemented. The new command is far more realistic since 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.

The section beginning on page 92 below describes predictions of line intensities when grains are present in an open geometry.

Dangerous interactions between the **abundances** command and with the **pgrains** commands are described on page 56 above.

11.10.2 Compiling the opacity files

All of the data files needed to use the standard grain types given in Table 18 are included in the Cloudy distribution. Nothing more need be done to use these grain types unless the continuum resolution is changed, as described below.

New grain types, or size distributions can easily be created, as described in the files *vanhoof_grain_model.** that are included in the data distribution. The current implementation has a built-in spherical Mie code that uses refractive index data and an arbitrary size distribution to compute realistic grain opacities. Any newly defined grains must be compiled with the **compile grains** command, described on page 149 below, before they can be used. That command creates opacity data files that have names end with ".opc".

The continuum energy mesh within the *.opc files must exactly match the continuum mesh within Cloudy. The code checks, and will stop if they do not match. If you change the continuum mesh used by the code it will be necessary to also recompile any grains you wish to use..

11.10.3 Grain abundance

The first number on the command line is the abundance of the grain type relative to its standard abundance. This is interpreted as a linear scale factor, or as a log if the number is zero or negative, or it the keyword **_log** appears.

11.10.4 Resolved or averaged grain size distributions

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. If the keyword **single** appears then the grains will have properties determined by averaging over a size distribution. This is similar to the physics considered by the older **grains** command.

11.10.5 Specifying "built-in" opacity types

There are a number of specific grain opacity files that are built into the code. These are listed in Table 24 and Table 25 on pages 150 and 151 below.

If either keyword **Orion** or **_ISM** appears then grain size distribution functions appropriate for these environments will be used. The ISM type reproduces the observed ratio of total to selective extinction, R = 3.1. The Orion size distribution is deficient in small particles, producing the large R observed in Orion. If either keyword **graphite** or **silicate** appears, then only that grain type is turned on. Both species are turned on if neither keyword appears.

A **gray** (or **grey**) distribution with ISM size distribution is also available.

If the keyword **silicate** appears together with a second number a single sized silicate will be used. Currently the diameters available are 0.01, 0.1, and 1 micron. The second number must match one of these.

The following example uses an Orion silicate with enhanced abundances (twice the default), and ISM graphite with default ISM abundances. Both use the size resolved distributions.

```
// Orion silicate with twice the normal abundance
pgrains distribution Orion silicate 2
// ism graphite
pgrains distribution ISM graphite
```

11.10.6 Specifying arbitrary opacity files

In most cases the built-in opacity types, described in the previous section, will be used. But there is an option to ask the code to read in an arbitrary opacity file that was created by the user. The process is described in the section starting on page 149 below.

The code tries to read in this file whenever a double quote (") occurs anywhere on the command line. If the quote is found the code then tries to read a file of stored opacities. It will then look for the name between a pair of quotes, as in "**special.opc**", and will stop if the file cannot be found. If the file exists then the opacities stored there will be used rather than the standard files produced with the **compile grains** command (see page 149 below). 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.

If **special.opc** was read in, the contents of that file will determine whether the calculations are size-resolved or not, irrespective of the keyword **distribution**.

11.10.7 grain heating and cooling

The keyword **no heating** will turn off photoelectric heating of the gas by grain photoionization. The keyword **no cooling** will turn off free particle

recombination cooling of the gas by grain collisions. Either would violate energy conservation, of course.

11.10.8 qheat - quantum heating

The keyword **qheat** turns on quantum heating, while **no qheat** turns it off. (There must be a single space between the **no** and **qheat**.) Quantum heating is turned on by default, except for unresolved size distributions.

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.

To save compute time quantum heating, when enabled, is only used when the grain cooling time is sufficiently short compared with the time between heating events.

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

In the following examples quantum heating will be included in all cases except the second. All use distributed sizes.

```
// Orion a single size-averaged silicate grain with Orion-like optical properties
// and with quantum heating
pgrains Orion silicate
// ism a single size-averaged graphite with quantum heating disables
pgrains ISM graphite no qheat
// this is equivalent but with quantum heating on by default
pgrains ISM graphite
// the default, size-resolved ism grains
pgrains
```

Quantum heating for all species can be turned off with the **no grain qheat** command described on page 157 below.

11.11 grains [-2; planetary; Orion; no heating; type...]

The effects of grains can be included, either with this command, or by using an abundance mixture that includes grains by default. The **abundances** command takes precedence over the default grains set with the **grains** command. The grains set with an **abundances old grains** command will completely override all parameters set with previous **grains** commands.

The **grains** command has two optional arguments, the abundance of the grains, and a index giving the type of grain to be used. These are described below.

This command will be replaced with physical treatment enabled by the **pgrains** command in a future version of Cloudy. That command is described on page 85 above.

The treatment of grain physics in Cloudy was developed in close collaboration with P.G. Martin. Details are provided in a section of Part II, and in Baldwin et al. (1991). The populations of grains summarized in Table 18 are presently incorporated in the code, and others can be easily added. The first column of the table gives a pointer to the grain type, the second is the type of grain, the third column summarizes the grain property, and the last column gives the area, $\int 4\pi a^2 \, n(a) \, da$, of the grain (cm^2) per hydrogen nucleon.

The temperature, potential, and drift velocity of the grains are determined using standard assumptions, as described, for instance, by Martin (1979) or Spitzer (1948; 1978), and in a section of Part II of HAZY. Heating by direct absorption of the continuum, Lα, and all other lines and continua included in the OTS fields, and gas collisions, are included as heating mechanisms in the calculation of the grain temperature. The balance between this heating process and cooling by collisions with the gas and by radiative cooling is used to establish the grain temperature. Gas heating by grain photoionization, and cooling by free particle capture onto the grain surface, are also included. The grain potential is determined by solving the photoionization-recombination balance equation. The heating and cooling of the gas by grain photoionization-recombination is determined self-consistently.

The default condition for the code is to not include grains, and when grains are enabled the default grain mixture has ISM properties. However, the **`abundance`** command can switch on various grains by default.

11.11.1 grain abundances

The abundances of the grain populations can be changed with the first optional number on the command line. This is a scale factor used to multiply the stored grain opacities. The scale factor is the log of the opacity relative to the standard value if it is less than or equal to zero, and the scale factor itself if it is positive. For example, both **`grains -2`** and **`grains .01`** would use ISM grains with each of the two 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.

11.11.2 grain types

It is possible to turn on each of the species listed in Table 18 independently. If two numbers occur on the line *and* the grain type is not specified with one of the keywords described below, then the first number is interpreted as the abundance, and the second is an integer pointer to the grain type (listed as the first column in Table 18). The

Table 18
Grain Populations

I	Type	Property	area per H
1	graphite	ISM	2.097×10^{-21}
2	silicate	ISM	2.397×10^{-21}
3	graphite	Orion	8.562×10^{-22}
4	silicate	Orion	9.787×10^{-22}
5	silicate	0.01 μm	8.476×10^{-21}
6	silicate	0.1 μm	8.476×10^{-22}
7	silicate	Volk AGB	2.397×10^{-21}
8	gray	Volk 0.1 μm gray	2.397×10^{-21}
9	PAH	Volk 0.001 μm	3.461×10^{-21}
10	PAH	Volk 0.00035 μm	2.447×10^{-21}
11	Silicate	1.0 μm	8.476×10^{-22}

other grain types are not turned off by this option, so it is possible to turn on several grain types with successive applications of the **grains** commands.

11.11.3 grain heating and cooling

The optional keyword **no heating** turns off photoelectric heating of the gas by grain photoionization. The optional keyword **no cooling** turns off free particle recombination cooling of the gas by grain collisions. Both options will violate energy conservation.

11.11.4 grains Orion

The optional keyword **Orion** makes the grains more similar to the large-R grains in the Orion Nebula, which have a fairly gray ultraviolet extinction. The two grain populations marked Orion in Table 18 are used in this case. This is the grain type used in Baldwin et al. (1991).

11.11.5 grains gray

This turns on the Volk 0.1 μm gray grains. This species provides an important test that grain heating and processes balance in the thermodynamic limit. The grain temperature should equilibrate at the energy density temperature when exposed to a black body radiation field. The alternative spelling **grey** is also accepted.

11.11.6 grains pah [qheat]

This turns on the two PAH grain species. These were added by Kevin Volk using PAH opacity functions given by Bregman et al. (1989) and Desert, Boulanger, and Puget (1990). Two sizes, 0.001 μm and 0.00035 μm, are in the mix. The cross section per hydrogen is the ISM abundance given by Schutte, Tielens, and Allamandola (1993). The temperatures of the PAH species are normally calculated using quantum, rather then classical, heating.

11.11.7 grains _agb [planetary nebula]

This makes the grains more similar to those observed in post AGB stars or proto planetary nebulae. The opacity for the silicate population is taken from unpublished work by Kevin Volk. This population is referred to as "AGB" in the printout. ISM graphite grains are used. The resulting mass in grains is a bit less than the ISM case, but may be an overestimate for classical planetary nebulae if grains are destroyed as the nebula ages. The AGB grains are also turned on with the **abundances planetary nebula** command. To be compatible with this abundances command, the program will also accept the command **grains planetary nebula** to turn on the post-AGB star grains.

The optical properties of the AGN grains are empirical, with no proper information about the refractive indices of the materials, and so no **pgrains** version of this mix exists.

The grain abundances of the population of planetary nebulae are quite uncertain. The dust-to-gas ratio resulting from the **grains planetary nebula** command is just below the ISM value of 0.007. Clegg and Harrington (1989) find dust-to-gas ratios below the ISM value, while Borkowski and Harrington (1991) find one object with a dust-to-gas ratio an order of magnitude above ISM. Mallik and Peimbert

(1988) find a dust-to-gas ratio in a sample of PNs roughly equal to the ISM. In view of this scatter the grain abundance should probably be treated as a free parameter.

11.11.8 grain abundance 0 "grey1.opac"

If a filename occurs between a pair of double quotes then the code will read in the parameters for this species from an ancillary file. Kevin Volk added this option. The file is actually read in by routine ***rdfile***, and will have the default file name **`dust.val`** if no name is specified. If a file name is specified then it must appear as the only character string after the grain species number.

Line 1: The dust grain label, the grain work function in eV, an integer flag for whether the grain is graphitic (value 0) or silicate (value 1), the total number of atoms in the grain, the number of H atoms in the gas per grain (the last 3 values matter only for the quantum heating calculation, if this is to be done), and the grain sublimation temperature in K. In Fortran the format would be `(1x,a10,2x,f7.3,1x,i1,1x,e12.4,1x,e12.4,1x,f7.1).`

Line 2: The dust density (g cm^{-3}), the molecular weight of the basic molecule, the normalizing abundance of this molecule by number with respect to hydrogen, a depletion factor for these molecules, the number of cross section values, the surface area (cm^2) per hydrogen atom, and the effective radius in cm. The format must be `(4(e9.3,1x),i3,2(1x,e9.3)).`

Line 3: This line has up to 4 optional elemental depletion values, giving the atomic number and then the depleted abundance relative to hydrogen. The format must be `(4(f5.1,1x,e12.4)).`

Subsequent lines: Each subsequent line gives an energy value in Rydbergs, the associated absorption cross-section per hydrogen atom, and the associated scattering cross-section per hydrogen atom. These values are given in common log form, with the cross-sections in cm^2 units. Each line has the values for one photon energy. The format must be `(3f11.5).` The values have to be given in order of increasing energy.

An example of such a file is the grey1.opac file included in the distribution.

11.11.9 examples of grains

The following are some examples of the use of the **`grains`** command.

```
*ism grains with ISM abundance
grains

* Orion grains with half their standard abundance
grains Orion .5

* turn on ISM graphite and Orion silicate
grains abundance =1, type =1
grains abundance =1, type =4

* only include opacity effects of ISM grains
grains no heating, no cooling
```

11.11.10 grains Orion function

There is good evidence that small grains are underabundant within ionized regions of the Orion Nebula (Sellgren et al. 1990). The **`function`** option on the

grains command makes it possible for the abundance of any species to vary across a cloud.

If the **grains** command sets the abundance of a single grain species then the **function** option will only apply to that particular species. If it occurs on a command that specifies more than one species of grains (as in the **Orion** keyword) then all species enabled by that command are affected.

This option works by setting the local abundance of a grain species to the product of an intrinsic abundance and the value of the function ***GrnVryDpth***. This is a function that is included with the distributed source. The user can modify it to produce any desired behavior. The function should return a scale factor for the abundance, unity if grains are to have their "standard" abundance. This scale factor could depend on the local physical conditions or the depth into the cloud.

It is also possible to specify certain default sets of grains with keywords on the **abundances** command. That 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 enhances where grains are destroyed. The user can do this by entering abundances with a depth-dependent table (see page 58).

11.11.11 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 using escape probabilities (i.e., Hummer 1968), and the predicted intrinsic intensities include this physics. The intensities *do not* include the reddening effects of any external grains that lie outside the line-forming region, however.

For an open geometry this same intrinsic emission-line spectrum is printed as the second block of lines. The first set of lines would be the observed spectrum if the geometry were observed from the illuminated face of the cloud, if the modeled region has a large molecular cloud behind it. This includes absorption by the grains, and back-scattering by grains beyond the shielded face of the cloud. The back-scattering assumes that large amounts of neutral material extend beyond the computed geometry.

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.

11.12 helium [options]

11.12.1 helium collisions off

This turns off collisional ionization and n-changing collisions (but not 2s-2p) within the helium singlets and helium ion. It was replaced with the **atom he-like** command, described on page 81 above, in version 94.

11.13 hydrogenic [options]

This command allowed some details of the treatment of the hydrogenic atoms to be changed. It was replaced with the **atom h-like** command, described on page 78 above, in version 94.

11.14 iterate [2 times]

This command 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 20 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 another iteration is needed.

11.14.1 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 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 number of calculations of the structure.

11.14.2 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 line optical depth scale has 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, and the default is 10. It is not now possible to specify more than 20 iterations. The second optional number is the convergence criterion. The default criteria are that the hydrogen and helium line and continuum optical depths have not changed by more than a relative fraction of 0.20 on the next-to-last iteration. The optional numbers may be omitted from right to left. The calculation stops when the relative changes in optical depths of several HI, HeI, and HeII lines are less than the second number. If the transitions are optically thin then only a second iteration is performed.

11.14.3 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 model 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.

11.15 no scattering opacity

This command 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 to be computed, the scattering opacity is assumed to attenuate the incident radiation field as $\left(1+0.5\tau_{scat}\right)^{-1}$ rather than $\exp(-\tau)$ (Schuster 1905).

Scattering should be neglected in a spherical geometry with gas fully covering the source of ionizing radiation. Photons absorbed by a pure scattering process 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., $h\nu << mc^2$) 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. If this command is given then 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.

11.16 turbulence = 100 km/sec [_log, dissipate]

The input number is the microturbulent velocity expressed in kilometers per second. This velocity field affects the line width and optical depth scale through the Doppler width

$$v = \sqrt{v_{th}^2 + v_{turb}^2} \ \text{[cm s}^{-1}\text{]}, \tag{50}$$

where

$$v_{th} = \sqrt{2kT/m} \ \text{[cm s}^{-1}\text{]} \tag{51}$$

is the projected line width due to thermal motions of particles of mass m, and the turbulent line width v_{turb} is normally zero. If the optional keyword **`_log`** (note the leading space) appears then the number is interpreted as the log of the turbulence.

Turbulence should add a turbulent or ram pressure component given by

$$P_{turb}(r_o) = \frac{1}{2}\rho v_{turb}^2 = 5.8\times10^6 \left(\frac{n_{tot}}{10^5 \text{ cm}^{-3}}\right)\left(\frac{v_{turb}}{1 \text{ km s}^{-1}}\right)^2 \quad \text{[cm}^{-3}\text{ K]} \tag{52}$$

where n_{tot} is the total density, v_{turb} is the turbulent velocity, and solar abundances are assumed. Turbulent pressure is not now included in the pressure law since it would either be negligible or totally dominate the pressure. The code will complain if a turbulent velocity is specified in a constant pressure model.

Line fluorescent excitation by the continuum will be increasingly important for larger turbulent line widths. Continuum pumping is included as a general excitation mechanism for all lines, using the formalism outlined by Ferland (1992), and described in further in a section of Part II.

The turbulence dissipate option provides a way in include conversion of wave energy into heat. 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

$$V_{turb}(r) = V_{turb}(r_o)\exp(-d / r_{scale}) \quad [\text{cm s}^{-1}] \tag{53}$$

where $V_{turb}(r_o)$ is the turbulence at the illuminated face and d is the depth into the cloud. The mechanical energy in the wave is assumed to have been converted into heat, and there is a local heating rate given by

$$G(r) = 3.45\times10^{-28}2^{-3/2}V_{turb}(r)^3 \quad [\text{erg cm}^{-3}\text{ s}^{-1}] \tag{54}$$

12 THERMAL SOLUTIONS

12.1 Overview

This section describes options that affect the thermal solution, the determination of the electron temperature. These deal with the accuracy of the solution, constant temperature models, or with additional sources of heating, such as cosmic rays or turbulence.

12.2 cextra -14.231 [temp to the 1.5 power]

This command adds an extra source of cooling due to some unspecified physical process. The first number is the log of the cooling rate in erg cm^{-3} s^{-1}. The second number is an optional exponent to specify a temperature dependence. The cooling will be given by

$$\Lambda = 10^{c_1} \times \left(\frac{T_e}{10^4 \text{ K}} \right)^{c_2} \quad \text{erg cm}^{-3} \text{ s}^{-1} \tag{55}$$

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.

12.3 constant temperature, t=10,000K [linear]

A constant electron temperature calculation will be performed. The number can be either the electron temperature itself, or the log of the temperature (the latter is assumed if the argument is less than or equal to 10). If the optional keyword **`linear`** appears on the line 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 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 600) is reached. The *vast* majority of the model will consist of predominantly neutral gas well outside the hydrogen Strömgren sphere, and this gas will have a small ambient level of ionization and emission due to collisional ionization. The resulting emission-line spectrum would be surprising since the neutral gas contributes significant emission. To get a more physical model it would be necessary to use the **`stop eden`** (page 105 **below**) or **`stop efrac`** (page 105 **below**) commands to stop the calculation when the hydrogen ionization front is reached, or **`stop zone`** (page 108 **below**) to stop the calculation at a particular zone number.

12.4 coronal equilibrium, T=10,000,000K [linear]

A model in coronal equilibrium, in which the gas is mainly collisionally ionized, will be computed. This calculation is very similar to those presented by, for instance, Raymond, Cox, and Smith (1976) or Gaetz and Salpeter (1983). The number is either the temperature or the log of the temperature (the argument is interpreted as a log if it is less than or equal to 10). If the optional keyword **`linear`** is specified then the number is always interpreted as the linear temperature. The command works by holding the electron temperature constant at the specified value, and adding a very weak radiation field with small intensity.

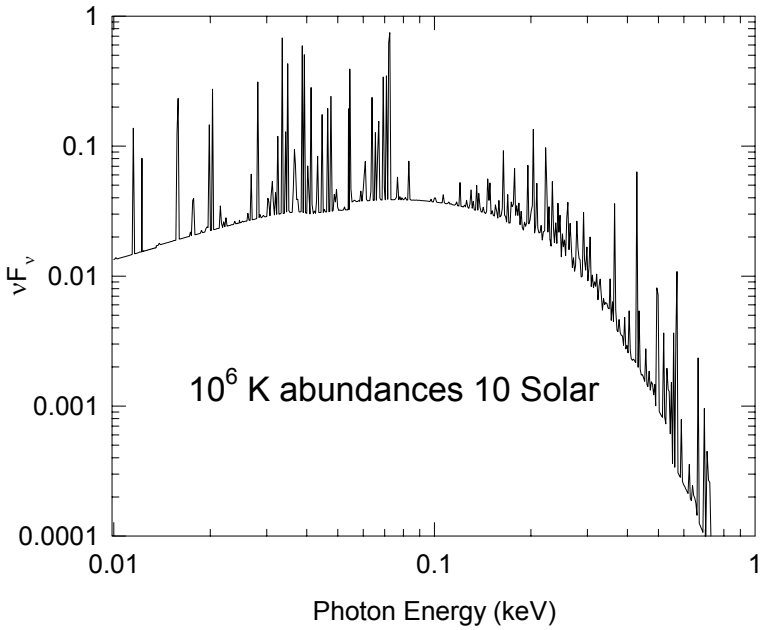


Figure 7 This figure shows the soft X-Ray emission from a simple model of the hot phase of the ISM. The input script brems.in was assumed.. coronal

It is necessary[19] 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 7 shows the soft X-Ray line and continuum emission predicted from the input stream in the test case **`brems.in`**.

12.5 cosmic rays, background, density =1.2 [index, etc.]

This command turns on energy deposition and ionization due to relativistic particles, as described by Ferland and Mushotzky (1984) and Part II. The first number is the log of the cosmic ray density [n(cr), cm^{-3}]. The second optional number is a power-law index α that describes the variation of the cosmic ray density with radius, i.e.,

$$n(cr,r) = n(cr,r_o)\left(\frac{r}{r_o}\right)^{\alpha} \quad \mathrm{cm}^{-3} \quad . \tag{56}$$

The default value of the index is α =0, or constant 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

[19] In versions 87 and before, the **`coronal`** command set the zone thickness to 1 cm, and stopped after computing one zone.

be used to evaluate the electron heating rates. The options can be omitted from right to left.

Collective effects are not included in the heating and ionization rates, but they may not be important either (Rephaeli 1987).

If no numbers appear on the line, but the keyword **background** does, then a constant cosmic ray density of $n(cr) = 2.6\times10^{-9}$ cm^{-3} will be used. This density will produce a neutral hydrogen ionization rate of ~7.4×10^{-18} s^{-1}, the value quoted by Tielens and Hollenbach (1985a; Table 10) and McKee (1999) for the galactic cosmic ray ionization rate. 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.

12.6 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 **tolerance** command (page 102 below). Normally Cloudy will punt[20] 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 it first produces a map of heating-cooling versus temperature to give an indication of where the equilibrium temperature should have been, if the **map** option is specified[21]. A section in Part III describes thermal failures in more detail, and describes the output produced before the program stops.

Failures occur most often when the code needs to jump over the peaks in the cooling function that occur near 2000 K and 10^5 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.

12.7 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 less than or equal to 10 and the linear temperature if greater than 10. 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.)

---

[20] FAQ: Punt is a technical term from American football. It is something bad that happens when progress in advancing the ball is lacking.

[21] 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.

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. If a large number of thermal failures or warnings result from the use of this command then it is likely that the code is too far away from the solution to converge. It should not be used in this case.

12.8 hextra -14 [scale r=18, thickness 12]

This command turns on extra heating due to some unspecified energy source. The first number H_o 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},. The extra heating rate varies as[22]

$$H = H_o[\exp(-depth / r_{scale}) + \exp(T - depth) / r_{scale})] \text{ [erg cm}^{-3}\text{ s}^{-1}\text{]}. \quad (57)$$

The default, when r_{scale} is not specified, is constant extra heating. If the third optional parameter T is the total thickness of the slab, and, if it is entered, 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 can be recoded to implement other functional forms of the extra heating.

12.9 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.

12.10 magnetic field, log(B) = 5

This command turns on the effects of magnetic fields. The argument is the log of the magnetic field strength in Gauss. Magnetic effects are not normally considered by the code. When a magnetic field is specified by this command, cooling due to electron cyclotron emission, using equations from Fabian, Pringle, and Rees (1976; these assume optically thin emission) are included. The volume-cooling rate is given by

$$\Lambda_{cyclotron} = n_e \frac{B^2}{8\pi}\frac{4}{3}\sigma_{\text{Thom}} c \left(\frac{v_e}{c}\right)^2 = 4.5433\times10^{-25} n_e B^2 T_e \quad \text{erg cm}^{-3}\text{ s}^{-1} \quad (58)$$

where σ_T is the Thomson cross-section and

$$v_e = \left(\frac{8kT_e}{\pi m_e}\right)^{1/2} = 6.2124\times10^5\, T_e^{1/2} \text{ cm s}^{-1} \quad (59)$$

[22] 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.

is the mean electron speed. See, however, Masters, Pringle, Fabian, and Rees (1977). They show that this emission process is likely to be optically thick under some circumstances. Cyclotron optical depth effects are not now treated, so this cooling rate is likely to be an overestimate.

Cosmic rays should not be included when a magnetic field is specified, since the effects of a field on cosmic ray transport are not now treated. A warning will be printed if both are included.

Magnetic pressure terms are not now included in the gas equation of state. This will contribute a pressure term of order $B^2/8\pi$, but which will depend on the magnetic field geometry at higher levels of precision. The magnetic energy density will be of order the thermal energy density of a gas with density n and temperature T when

$$P_{mag}/k = B^2/8\pi \quad k = B^2 2.882\times10^{14} \approx nT \quad [\mathrm{cm^{-3}\ K}]. \tag{60}$$

In the ISM this magnetic pressure is often roughly equal to the ram or turbulent pressure

$$P_{ram}/k = \rho v^2/2 \quad k = 60.14 \quad nv_{kms}^2 \approx nT \quad [\mathrm{cm^{-3}\ K}]. \tag{61}$$

where the velocity is in km/s and n is the nucleon density. For comparison, the Alfvén velocity, the speed at which magnetic fields convey information, is

$$v_A = \frac{B}{(4\pi\rho)^{1/2}} \approx 2.19\times10^6 B \quad n^{-1/2} \ \ \mathrm{km\ \ s^{-1}}\ . \tag{62}$$

The magnetic pressure term is neglected for the same reason that turbulent pressure is neglected - it would either be negligible or would so dominate the pressure that the density would be degenerate.

12.11 Map, zone 4 [range 2000, 5000]

This command 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 no zone calculations are performed. The calculation of the heating and cooling is self-consistent. A section in ***Problems*** in Part III 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 described here 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 less than or equal to 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 163 below.

The thermal map can be punched with the **`punch map`** command (page 130 below). This will produce a form of the output that is suitable for later processing by other software.

The code stops when the map is complete since it is left in a disturbed state.

12.12 neutrons -2 [efficiency =-2]

This command 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 less than or equal to zero, and a linear scale factor if positive.

The second argument is optional, and is the heating – ionization efficiency of the neutrons. Its default is unity. Both quantities are interpreted as logs if less than or equal to zero, and linear if greater than zero.

12.13 print coolants, zone 135

See page 112 below.

12.14 print heating

See page 114 below.

12.15 set temperature [solver, tolerance]

This command changes the temperature solver and its error tolerance. It is described on page 101 above.

12.16 time dependent model of zone 3

A time-dependent model of the specified zone is performed, as discussed by Ferland and Truran (1981). The model follows the recombination and cooling after the ionizing radiation is instantaneously cut off. The initial conditions are those appropriate for the zone specified as the argument. Collisional ionization is included, so this calculation is somewhat like a shock.

This command does not now work, and will not function again until development work on helium is complete.

12.17 tolerance 0.001

The balance between the heating and cooling rates sets the equilibrium electron temperature. This command is used to change the error tolerance allowed in the heating-cooling match. The number is the largest fractional error allowed, and is interpreted as the tolerance itself if it is positive, and the log of the tolerance if it is less than or equal to zero. The default tolerance is a fractional error of 0.02. 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.

This command has been replaced with the set temperature tolerance command (page 164 below).

The balance between the heating and cooling rates sets the equilibrium electron temperature. This command is used to change the error tolerance allowed in the heating-cooling match. The number is interpreted as the tolerance itself if it is positive, and the log of the tolerance if it is less than or equal to zero. The default tolerance is a fractional error of 0.02. This will be the error in the heating-cooling balance allowed in each zone. The total error or energy conservation mismatch over a model will be much smaller, usually of order ten times smaller than the tolerance specified.

13 STOPPING CRITERIA

13.1 Overview

Cloudy will stop at some depth into the cloud. The physics that sets this limit to the radial integration is important since this 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 an outer limit is not necessary in this case, unless lines with very low ionization and excitation potentials (i.e., the [C I] or [O I] far infrared lines) are of interest. It would be necessary to lower the stopping temperature with the **`stop temperature`** command if atomic FIR lines are to be considered.

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 criteria 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 (600) was reached.

13.2 Danger! Understand why the calculation stopped!

There are circumstances in which the predicted emission-line spectrum will depend strongly on 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 low-ionization infrared lines.

There are several checks that should be made to make sure 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. Left to its own devices the code will probably stop when the temperature falls below the default lowest temperature of 4000 K. 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), and thermal instabilities may result. It is also possible that the calculation will stop because of an internal error. The code will explain if this is the case. If an internal error occurs then all results are suspect, the code will say so, and it will ask that you send me the input stream and version number.

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.

13.3 radius inner =18 [thickness =16; parsecs; linear]

The **`radius`** command is discussed on page 73 above. The optional second number can set the thickness of the cloud.

13.4 stop column density = 23 [neutral; ionized; total; ...]

This command causes the calculation to stop when the specified hydrogen column density [*N(H)*, cm^{-2}] is reached. 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 the effective column density (defined in this subsection). For all cases the default stopping column density is 10^{30} cm^{-2}.

13.4.1 stop column density 23

The number is the log of the total hydrogen column density (atomic, ionic, and molecular hydrogen), defined as the integral

$$N(H) = \int \left\{ n\left(H^o\right) + n\left(H^+\right) + n\left(H^-\right) + 2n\left(H_2\right) + 2n\left(H_2^+\right) + 3n\left(H_3^+\right) \right\} f(r)\, dr \qquad (63)$$

where *f(r)* is the filling factor.

13.4.2 stop neutral column density 23

The number is the log of the neutral (atomic) hydrogen column density

$$N\left(H^o\right) = \int n\left(H^o\right) f(r)\, dr \ . \qquad (64)$$

13.4.3 stop ionized column density 23

The number is the log of the ionized hydrogen (H^+) column density

$$N\left(H^+\right) = \int n\left(H^+\right) f(r)\, dr \ . \qquad (65)$$

13.4.4 stop effective column density 23

This command is actually a form of the **`stop optical depth`** command. 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. Using this command, it is possible to stop the calculation when the incident continuum has been attenuated by the appropriate absorption at 1.0 keV. The calculation will stop when the absorption optical depth at 1.0 keV (neglecting scattering opacities) reaches a value of

$$\tau_{abs}\left(1.0\ \mathrm{keV}\right) = N_{effec}\, 2.14 \times 10^{-22} \qquad (66)$$

at 73.5 Ryd. N_{effec} is the effective column density. The absorption cross-section per proton for cold neutral gas is taken from Morrison and McCammon (1983). Scattering opacities *are not* included in this optical depth.

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.

13.5 stop eden 3 [linear]

The model will stop if the electron density falls below the indicated value. The number is entered as a log. In this case the model will stop if $n_e < 10^3$ cm^{-3}. There is an optional keyword **`linear`** that will force the argument to be interpreted as the quantity itself, not its log. This command is a useful way to stop constant temperature models. For instance, the calculation can be forced to stop at the H$^+$-H^o 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^o 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+ - Ho 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.)

13.6 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 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 electron fraction of -10^{37} cm^{-3}. (The negative sign is not a typo.)

13.7 stop line 6300 reaches 0.1 relative to 5007

The model will stop when the emission line with the wavelength given by the first number exceeds an intensity given by the second number, relative to a second emission line with wavelength given by the optional third number, in this example [O III] λ5007. If a third number is not entered, this second emission line 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.

This command only uses the wavelength of the line to identify the line. There are cases where more than one line can have the same wavelength. In these cases the first line that has the correct wavelength will be used.

The scaling of the line intensities on the final printout can be change with the **normalize** command, as described on page 109 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 one of the limits is reached.

13.8 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 and the second number is the energy in Rydbergs. 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 interpreted as a log if it is negative, as linear Rydbergs 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 104 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.

13.8.1 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 cause Cloudy to stop when the log of the absorption optical depth at the Balmer edge (ν = 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 opacity sources such as grains or free-free absorption.

13.8.2 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 entered 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.

13.9 stop temperature =1,000K [linear, exceeds]

The model will stop if the electron temperature drops below T_{low}, the argument of this command. The temperature is interpreted as a log if the argument is less than or equal to 10, and as the linear temperature if greater than 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] ^{3}P 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~10^3 K (Williams 1967), so thermal stability problems may develop if T_{low} is lowered below a few thousand degrees Kelvin. If stability problems occur then it may be necessary to increase the number of thermal failures allowed, with the **`failures`** command. This issue is discussed further in a section in Part III.

It is possible to use a form of this command to stop a calculation if the temperature *exceeds* the input value. 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_e>10^5 K) are not of interest. If the keyword **`exceeds`** appears on the line then the specified temperature will be the highest allowed temperature. The other rules for the command are unchanged.

13.10 stop thickness 9.3 [parsecs; linear; 23 on sec iter] stop depth . ….

This command 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 73 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 20 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.

13.11 stop zone 123 [21 on sec iteration, ...]

In this example the calculation will stop after computing 123 zones. The default value is 600. Up to 20 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 162 below).

14 CONTROLLING OUTPUT

14.1 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 a later part of this document describes the meaning of the output.

14.2 normalize to "o 3" 5007 "o 3" [scale factor = 100]

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 printout[23], and the wavelength must match the wavelength in the printout to all four figures. The wavelength units must appear if they are not Ångstroms.

The optional second number gives the 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 matches the first numeric parameter on the command to four significant figures. 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. Specifying an optional line label can lift this degeneracy. If a double quote (") is present on the line the code will try to form a line label by looking at the string that occurs between the pair of quotes. This label is not needed if the line wavelength is unique. The label must be within a pair of double quotes and specify the four-character line label used within the code (see the section "Lines" in Part IV of this document). The label can be either upper or lower case.

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
```

14.3 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

[23] 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.

keyword **trace** will turn on a great deal of information concerning the mechanics of generating the plot.

Publication-quality plots can be produced using the **punch** commands (described beginning on page 117 below) to produce a file that can then be post-processed using other plotting software.

14.4 plot continuum [_raw, trace, range]

If the keyword **continuum** is entered then the continuum (usually 1.001×10^{-8} Ryd $\leq h\nu \leq 7.354\times10^{6}$ Ryd) is plotted. This energy range is altered by entering the two optional limits with the **range** key. This is described in detail in section 14.5.1 below.

The default is to plot both the incident continuum (in units of νf_ν) entering the cloud (plotted as .'s) and that 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}.

14.4.1 plot continuum keywords

It is possible to plot specific components of the continuum with the following series of keywords.

14.4.2 plot diffuse continuum

If the keyword **diffuse** appears then the diffuse emission per unit volume will be plotted. This will show emission within the last computed zone. This continuum is the local gas and grains emission in the optically thin limit and unity filling factor.

14.4.3 plot emitted continuum

If the keyword **emitted** appears then 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.

14.4.4 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.

14.4.5 plot reflected continuum

If the keyword **reflected** appears then only 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.

14.5 plot opacity [type, range]

If the keyword **opacity** is entered then the opacity (total cross section per hydrogen atom) of the first and last zones is plotted. The continuum between $1.001\times$

10^{-8} Ryd ≤ $h\nu$ < 7.354×10^{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.

14.5.1 plot range options

For the **opacity** and **continuum** options, there is a further keyword **range** to specify the energy range of the plot. If one number occurs on the command line then it is interpreted as the lowest energy (in Rydbergs) on the plot. If the first number is zero then it is interpreted as the lowest energy in the continuum, 1.001×10^{-8} Ryd. The optional second number is interpreted as the highest energy shown on the plot. If the second number is omitted or zero then it is interpreted as 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 in Rydbergs. 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
```

14.6 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 as a function of temperature will be made. This will follow the last zone calculated, and will be appropriate for the attenuated continuum and physical conditions in that zone.

14.6.1 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 less than or equal to ten then both numbers are interpreted as logs of the temperature. If the first number is greater than ten then both numbers 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 163 below.

14.7 print ages

This command tells the code to print all of the timescales associated with the **age** command (page 69 above). Normally only the shortest timescale is printed.

14.8 print arrays

This option tells the code to print the ionization balance arrays for all elements heavier than helium. The first line is the vector of ionization rates [s^{-1}]. The second line is the vector of recombination rates [s^{-1}]. These lines are the two diagonals of the bi-diagonal matrix used in the solution of the ionization balance equation and include *all* ionization and recombination processes. 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.

14.9 print constants

The physical constants stored in the header file physconst.h will be printed.

14.10 print column densities

The column densities of the various ions are not printed unless explicitly requested with this command.

14.11 print coolants, zone 135

This turns on an option to print the emission-line cooling arrays for the specified zone. If no zone number or 0 appears on the line then the coolants for *all* zones will be printed. The numbers printed are the log of the cooling per unit volume. Only the strongest coolants are printed. These are indicated in the header of the printed array. 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 following number is the fraction of the total cooling carried by that agent.

14.12 print continuum [block]

This is actually two very different commands, with different purposes. The second form will eventually be removed.

14.12.1 Adding entries into the emission line stack

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 162 below) provides a way to change which contributors are included in this prediction.

If the **print line inward** command (page 115 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 set of energies that specify where these continuum points is stored in the vector ***EnrPredCont*** that is set in *zerologic.c*. There are ***NPREDCONT*** points in this vector. Both live in the *predcont.h* header file.

14.12.2 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 119 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.

14.13 print departure coefficients

This command tells the code to print LTE departure coefficients for many species. The **`print populations`** command (page 114 below) controls printing individual level populations.

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 departure coefficients for H I are printed.

14.14 print errors

The code will always identify problems that occur by printing comments during the calculation, or warnings after the calculation is complete. This command will cause the code to also print these warnings to **`stderr`**. On many systems this output can be redirected to the screen.

14.15 print every 1000 [5 37 93]

Cloudy will always print the results for the first and last zones. This command can be used to vary 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.

14.16 print flux at Earth

If the distance to an object is set with the **`distance`** command (page 72 above), and the luminosity of the lines can be predicted (see the discussion on page 26 above) then this command tells the code to 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.

14.17 print heating

This tells the code to print the relative heating due to each stage of ionization or physical process. 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.

14.18 print populations H-like levels [to level 45]

Level populations are normally not printed for the atoms of the H-like or He-like isoelectronic sequences.. This option will allow the level populations to be printed. If no numbers appear on the line only the levels up to 15 will be printed to same room. Enter the highest level on the line as an integer to print 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 poulations for H I are printed.

The departure coefficients will be printed if the **print departure coefficients** command (page 113 above) also appears in the input stream.

14.19 print he-like levels

Not yet . ..

14.20 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.

14.21 print line *options*

A great deal of information about line formation and beaming is stored within the code but not normally printed to save space. The following **print line** commands tell the code to display this information. A section of a later Part of this document gives more information

Any of a series of options can appear on the **print line** command[24]. These tell the code to indicate various processes that contribute to line formation. The code does not normally break out these contributions to lines to save space. Only one of these options is recognized on a single command line.

14.21.1 print line collisions

Collisions are usually the dominant contributor to formation of an optically thick line. The entry will have the label "Coll" followed by the wavelength.

---

[24] 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.

14.21.2 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. The **`print line pump`** 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.

14.21.3 print line heat

Lines include fluorescent excitation as a line formation process. If a line is radiatively excited then collisionally deexcited it will heat rather than cool the gas. The heating due to line collisional de-excitation will be printed when this option is enabled. The entry will have the label "Heat" followed by the wavelength.

14.21.4 print line all

Contributions from collisions, pumping, and heating will be punched.

14.21.5 print line inward

Optically thick emission lines are not isotropically radiated. The "inward" fraction of the line is the part that is emitted in the inward 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.

14.21.6 print line sort wavelength [range 3500A to 1.2m]

This command causes the output spectrum to be sorted by wavelength rather than by ion[25]. 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 microns or centimeters if it is immediately followed by a "c" or "m". They must be positive and in increasing wavelength order.

14.21.7 print line sort intensity

The emission line predictions will be sorted in order of decreasing intensity.

14.21.8 print line optical depths [_off, faint]

Line optical depths will be printed at the end of the iteration. There are two optional keywords. If **`_off`** appears then printing line optical depths will be turned off (useful if turned on in a previous iteration and no longer needed). If the keyword **`faint`** appears then a number will be scanned off the input line, the log of the smallest line optical depth to print. The default smallest line optical depth to print is 0.1. Optical depths for all lines that mase are normally printed.

14.21.9 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

[25] The **`print sort`** command did not function between 1986 and 2001. It became functional again with version 96 but was moved to be an option on the **`print line`** command..

determining stellar temperatures, which rely on the sum of a set of observed line 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 Ångstroms 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.

14.21.10 print line column [linear]

The main block of emission lines is normally printed as a four-line wide array. With this command the line will be printed as a single long column. 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.

14.21.11 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 less than or equal to zero) or the linear value of the intensity of the weakest line to print (if positive), relative to the reference line. The reference line is usually Hβ, and can be changed with the **normalize** command (page 109 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.

14.22 print off

This turns print out off, as with the **print quiet** command (page 117 below).

14.23 print on

This command turns on printout. This is the opposite of the **print quiet** or **print off** commands.

14.24 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 return 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.

14.25 print optical depths [_on_, _off, faint]

Line optical depths are not printed by default. The option **_on_** will tell the code to print them. The default is to not print these, and can be set with the **_off** option. The smallest line optical depth to be printed is 0.1 by default. This is changed with the **faint** option. If it appears then the log of the smallest line optical depth to be printed must appear on the line.

14.26 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.

14.27 print short

The detailed final printout is shortened when the **short** keyword appears. Only the emission lines and a short summary of some thermal properties of the model will be printed.

14.28 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.

14.29 punch commands

14.29.1 Overview

Punch commands save results into a file that can be used later. Punch output is the primary output mechanism for Cloudy. There are many options. For instance, physical quantities such as 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 20 **punch** commands can be entered.

14.29.2 An output file name must appear inside double quotes

Each **punch** command must specify a file name[26] for the resulting output. This file name must appear between a pair of double quotes, like "output.txt". This must be a valid file name for the operating system in use. The following is an example.

```
punch overview "model.ovr"
```

The code will complain and stop if a valid file name is not present.

14.29.3 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 punch output will be produced for every iteration, with results of each iteration separated by a line of sharp signs.

14.29.4 The "noclobber" option

By default the code will open the punch file for each model 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 clobber results of all previous calculations.

The **noclobber** option should be used in a grid of models to produce output in 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.

14.29.5 The " no hash" option

If results from more than one iteration are punched, then each iteration will be ended 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 spreadsheets. If the **no hash** keyword appears on any punch command, the hash marks will not be produced for any punch files.

14.29.6 The " no title" option

The title of the model and the version number of the code are normally printed on the first line of the punch file. This introduces an extra step when copying and pasting the punch file into spreadsheets, since this line is not needed. The **no title** option will stop this first line from being produced.

14.30 punch abundances

The log of the gas-phase abundances of the elements will be punched for each zone. This provides a check for the effects of the **element table** command (see page 59 above).

14.31 punch advection

Produces information about advection terms.

---

[26] 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.

14.32 punch ages

The timescales for several physical processes will be punched as a function of depth.

14.33 punch agn [options]

This command produces output files that were used to create data tables in the 3rd edition of *Astrophysics of Gaseous Nebulae*. The options are the following: **charge** transfer, **recombination** coefficients, **opacity**, **hemis**, and **hecs** (for He^0 collision strengths).

14.34 punch asserts

The **assert** command is described on page 166 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.

14.35 punch column density

This produces a file containing the logs of the column densities of the atoms and ions.

14.36 punch continuum

This command has been the primary mechanism for saving the continuum predicted by the code. All continua are given as the flux per octave νf_ν (with units 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 $4\pi r_o^2$.

Details in the following discussion describe the **punch continuum** command with no additional keywords. Many aspects of this discussion also pertain to all of the **punch continuum** subcommands, which are described starting in section after the discussion of the observed quantities.

This command is being superceded by a new command, **punch spectrum**, described on page 124 below. That command is still under construction at this time.

14.36.1 Emission line – continuum contrast

Emission lines are included in the resulting punch 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 the lines 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 163 below. This issue is discussed further in the section *Observed Quantities* in Part IV of this document.

14.36.2 Pumped contributions to the lines

Continuum pumping and fluorescence is included for all lines. The contribution is not usually printed as a separate quantity, but will be if the **`print line pump`** command (page 115 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. If absorbing gas covers an observed continuum source then the situation is like the P Cygni problem, and pumping does not increase the total intensity of the line at all.

The printed line intensity includes the pumped contribution unless the **`no induced processes`** command is entered. That command is unphysical since it turns off continuum pumping as a line excitation process. 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.

14.36.3 Energy units for the punch output

By default the energy unit for the independent axis of a spectrum is Rydbergs. The unit can be changed to any of several energy or wavelength units with the **`units`** keyword that appears on a **`punch continuum`** command. The keywords recognized are the following: **`microns`**, **`_keV`**, **`_eV_`**, **`Angstroms`**, and **`Rydbergs`**. Both the keyword **`units`** and one of these units must appear.

14.36.4 Punch continuum predictions

Column 1. The first column gives the photon energy in the units set with the **`units`** option (described on page 120 above). The default units are Rydbergs.

Column 2. The second column is of the incident continuum at the illuminated face of the cloud. The intensity unit for all elements of the continuum is the intensity per octave ($4\pi\ \nu J_\nu$).

Column 3. The third column is the transmitted (attenuated) portion of the incident continuum, and does not included 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 (so that the total emission from the nebula is this multiplied by the inner area of the cloud). This column does not include the attenuated or reflected portions of the incident continuum.

Column 5. The 5th column 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 if the geometry is not spherical.

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. More than one line may be included in a continuum cell.

14.36.5 What is observed

Figure 8 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.

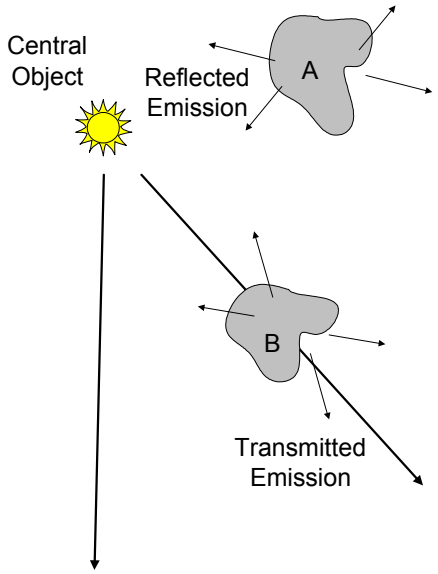


Figure 8 This figure illustrates several of the continua that enter in the calculations. ObsContin

Three possible geometries 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 (the line through cloud B), 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" 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 computed 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 (both reflected and transmitted seen). If the central object is not in the beam then the quantity in column 4 would be observed.

The following sections describe various ways to produce output files with portions of these continua included. The **`print diffuse continuum`** command (page 112 above) tells the code to include some contributors to the continuum in the emission lines in the main output. The lines with the label *nFnu* include the diffuse continuum, and those with the label *nInu* include only the reflected plus attenuated incident continuum. The **`set nFnu`** command (page 162 below) provides a way to change which continuum contributions are included in the *nFnu* entry.

14.36.6 punch continuum bins "cont.txt"

This command is used to punch the continuum energy array. The first number is the frequency of the center of the bin ν, and the second number is its width $\delta\nu$. The bin extends from $\nu-\delta\nu/2$ to $\nu+\delta\nu/2$.

14.36.7 punch diffuse continuum "cont.txt"

This command is used to punch the local diffuse continuum per octave (νf_ν erg cm^{-3} s^{-1}) at the end of the calculation. This is the locally produced diffuse emission from the gas, per unit volume with unit filling factor, for the last zone. Optical depth effects are not included and the continuum is the local continuum for the last computed zone. The output has three columns, the photon energy, interactive diffuse continuum, and non-interactive continuum.

14.36.8 punch emitted continuum "cont.txt"

The continuum emitted and reflected from the nebula is punched. The first column is the photon energy. The second column is the reflected continuum plus lines. The third column is the outward diffuse emission from the computed structure. 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. 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}) and are relative to the inner radius.

14.36.9 punch FeII continuum "feii.con"

The pseudo-continuum of FeII lines predicted by the large FeII atom will be punched. See page 84 above for more details. In the resulting output the first number is the wavelength of the center of the band in Ångstroms. The second number is the integrated intensity of FeII emission over that band. The file *Fe2Bands.dat* contains the definitions of the bands and explains how to change them.

14.36.10 punch grain continuum "grains.con"

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.

14.36.11 punch incident continuum "cont.txt"

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}).

14.36.12 punch interactive continuum "cont.txt"

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 are 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.

14.36.13 punch ionizing continuum [options]

This command is used to punch the ionizing continuum 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 number is this photon flux multiplied by the gas opacity. This quantity has units 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 8^{th} 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 8^{th} column. Enter zero for this number if you want all interactions to be printed. The optional numbers may be omitted from right to left.

14.36.14 punch outward continuum "out.txt"

In the output file the photon energy is followed by the attenuated incident continuum, the outwardly directed 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.

14.36.15 punch raw continuum "raw.txt"

This command is used to punch the "raw" continua at the end of the calculation. 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 s^{-1} cm^{-2} $cell^{-1}$.

14.36.16 punch reflected continuum "reflec.txt"

This command is used to punch the reflected continuum at the end of the calculation. This is only done if **`sphere`** is not set. The first column is the photon energy, the second the reflected continuum at that energy $4\pi \nu J_\nu$ (units 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 albedo of the cloud, the ratio of the reflected to incident continuum.

The last column gives the label for continuum processes with thresholds at the energy.

14.36.17 punch transmitted continuum "trans.txt"

This command is used to save the transmitted (attenuated incident and outward component of diffuse) continuum predicted at the end of the calculation.

This punch file can then be used as part of the incident continuum in a later calculation, by reading in this file with the **`table read`** command (see page 45 above). Three cautions apply when reading this file as an input continuum. First, if the keyword **`last`** does not appear on the line 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 either the **`last`** option should the included on the punch **`continuum command`** or the punch file must be edited to leave only the last computed continuum. 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, the program 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 163 below) is not used in this command. This is so 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.

14.37 punch spectum

This command is being developed to replace the punch continuum command described above. It works together with the **`set spectrum`** command (see page 159 below), to form a type of punch output that can be modified to suite a particular purpose. The **`set cpunch`** 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 120 above.

14.38 punch convergence [reason, error]

These commands produce information about various aspects of the converged solution.

14.38.1 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 93 above) is used.

14.38.2 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.

14.39 punch cooling

The code will punch the cooling agents for each zone. The first number is the zone number. The next two numbers are the heating and cooling rates (cm^{-3} s^{-1}). The following numbers are labels for members of the cooling array and the fraction of the total cooling 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 165 below).

14.40 punch charge transfer

Charge transfer recombination and ionization rate coefficients for hydrogen onto heavier elements will be produced. The rates will be evaluated at the current temperature, which can be specified with the **`constant temperature`** command (page 96 above) for 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.

14.41 punch _dr_

The logic behind the choice of zone thickness will be described.

14.42 punch element *name*

This command will punch the ionization structure of any element. The resulting punch output will have one line per zone and give the relative abundance of each successive stage of ionization. The keyword for this command is **`element`** and this must be followed by the element name spelled with the first four characters exactly as given in Table 15 (page 52 above).

The first number on the resulting output is the physical depth (in cm) from the current position to the illuminated face of 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.

14.43 punch FeII [populations, departure, _all]

This will produce some information about the FeII atom. The atom is turned on with the **`atom FeII`** command (page 82 above).

punch FeII departure The departure coefficients for selected levels of the large FeII atom will be punched. This is normally only listed for selected levels. All levels will be punched if the keyword **`_all`** appears.

punch FeII populations The level populations for selected levels of the large FeII atom will be punched. It is normally only for selected levels. All levels will be punched if the keyword **`_all`** appears.

punch feii continuum See page 122 above (under **`punch continuum`**).

14.44 punch gaunt factors

This produces a table showing the free-free gaunt factors as a function of photon energy and temperature.

14.45 punch grains [options]

These commands show predictions of the grain models.

14.45.1 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.

14.45.2 punch grain physical conditions

The grain temperature, floating potential, drift velocity, fraction of total heating, and fraction of total cooling is output for each zone.

*14.45.3 punch grain _qs_*

The photon energy is followed by the absorption and scattering *Q*s for each grain species.

14.45.4 punch grain continuum

See page 122 above.

14.46 punch heating

The code will punch the heating agents for each zone. The zone number is first, followed by the heating and cooling. This is followed by a set of labels for members of the heating array 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 165 below).

14.47 punch helium

One of the keywords **`triplets`**, **`singlets`**, or **`_ion`** must appear. The ionization and recombination rates affecting the model atom for the triplet, singlet, or ion will be punched.

14.48 punch htwo

A variety of H_2 formation and destruction mechanisms are punched for each zone.

14.49 punch hydrogen

14.49.1 punch hydrogen conditions

This will output the physical conditions and hydrogen constituents as a function of depth. The densities of H_o, H^+, H^2, H_2^+, H_3^+, and H^- relative to the total hydrogen density follow the depth, temperature, hydrogen density, and electron density.

14.49.2 punch hydrogen ionization

This will output 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, and predicted ratio of H^+ to H^o, and the theoretical ratio for the simple case.

14.49.3 punch hydrogen populations

This will punch the depth, the ionization fractions H^0 and H^+, the level populations for the lowest 6 levels, followed by the populations of 2s and 2p.

14.49.4 punch hydrogen 21cm

This will punch the depth, the optical depth in 21 cm, and the temperature at that point.

14.49.5 punch hydrogen recc

This produces a table of hydrogen radiative recombination cooling coefficients, and is used to produce a table for AGN.

14.50 punch ionization

The mean 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.

14.51 punch ip

The code will output the valence shell ionization potentials of all ions and atoms of the 30 elements included in the code. The first column is the energy in Rydbergs, and is followed by the spectroscopic designation of the ion.

14.52 punch gammas

The code will punch 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.

14.53 punch lines, options

This set of commands will punch some details about line formation.

14.53.1 punch lines, array

The code will punch the array of total line intensities in a form in which the line spectrum 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 punched. 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[27].

The energy units for the lines are Rydbergs by default. This command recognizes the **`units`** option, described on page 120 above.

[27] In versions 90 and before of the code, only the level 1 and level 2 lines were output by this command.

14.53.2 punch lines, cumulative

This option on the **punch lines** command 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 (see the section "Lines" in Part IV) 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 then punched. The depth into the cloud and the integrated intensities of the lines (erg cm^{-2} s^{-1}) are then punched for each zone. This information can then 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 punched quantities will be the intensity 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.

14.53.3 punch lines, data

This option on the **punch lines** command tells the code to punch some atomic data for all lines included in the line transfer arrays. It can be used to generate a table listing many lines. 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 Pequignot, Petitjean, and Boisson (1991). For these the spectroscopic designation and wavelength are given, followed by the log of the recombination coefficient.

The remaining sets of lines are those that are treated with full radiative transfer. The first set of lines is the "level 1" lines, those with accurate atomic collision data and wavelengths. The next set of "level 2" lines is much larger 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 FeII lines come last if the large atom is turned on.

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 96 above). The number of H-like, He-like, CO, and FeII lines that are printed is controlled by the size of the atom when the **punch line data** command is executed.

14.53.4 punch lines, intensity [every 5 zones]

This option on the **punch lines** command 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, or if the keyword **array** does, then the wide format is produced.

14.53.5 punch lines, labels

This tells the code to create a punch file that lists all emission line labels and wavelengths in the same format as they appear in the emission line list. This is a useful way to obtain a list of emission lines to then use to obtain predictions using routine ***cdLine***.

14.53.6 punch line optical depths, limit=-2

This will output the total optical depths for all lines. This is only meaningful after at least a second iteration. By default all lines with optical depths greater than 0.1 will be output. The lower limit can be reset with the optional number than can appear on the line – this is the log of the smallest optical depth to be printed.

The line identification and wavelength, in the form used in the usual emission line printout, and log of optical depth, are printed, one per line. The wavelength units for the lines are Ångstroms, microns, or centimeters, as in the printout, by default. This command recognizes the **units** option, described on page 120 above, so any of the wavelength or energy units described there can be used instead.

14.53.7 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 lines with upper level populations greater than zero will be output. The lower limit 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 that is produced 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 of output 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}).

14.53.8 punch lines, structure

The **structure** option tells the code to punch 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 escaping the cloud, produced at that point, $n_u A_{ul} P_{ul} h\nu$, which 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 (see the section "Lines" in Part IV of this document) 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 cut and paste the line identification and wavelength from a Cloudy output.

The punch output begins with line labels and wavelengths. The remaining lines give 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 to be 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, structure, "lines.str"
totl 4861
o  3 5007
totl 3727
o  1 6300
12CO  2589m
13CO  2475m
end of lines
```

The **punch lines cumulative** and **punch lines structure** commands use the same line array, so both commands cannot be used in the same calculation.

14.54 punch Lyman alpha

The Lα optical depth to the illuminated face to the current zone is punched, together with the excitation temperature, electron temperature, and the ratio of these.

14.55 punch map, zone 3 [range 3,999 to 4500]

This command is used to produce 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 100 above).

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 temperature limits to the map. Both will be interpreted as logs if the first number is less than or equal to 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 163 below).

14.56 punch molecules

The abundance ratios $2H_2/H$, $2H_2^*/H$, CO/C, H_2O/O, OH/O, CH/C, and $2O_2/O$ will be punched. The depth into the cloud is given in the first column.

14.57 punch opacities [total, grain, element]

These are options to output any of several opacity sources considered by the code. The opacities are only defined over the energy band of the continuum source defined for the current model. So, for softer continua, the resulting opacities will not extend to high energies. A hard non-thermal continuum such as **table agn** (page 42 above) should use used in the input stream to obtain the opacities over the full energy range. This command recognizes the **units** option to change the energy scale (see page 120 above).

One of the following keywords must appear.

14.57.1 punch total opacity

If the keyword **total** appears then he total gas opacity (absorption cross section per atom, with units cm^{-1}) will be punched. This will be the total opacity 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 local albedo, the ratio $\kappa_s/(\kappa_s+\kappa_\nu)$, at that energy. 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.

14.57.2 punch grain opacity

If the keyword **grain** appears then only the total grain opacity (all species enabled for the current calculation) will be punched. The output will only be produced after the last zone is calculated.

Columns in the output file will contain the photon energy, the total (absorption plus scattering) opacity, the absorption opacity, and the scattering opacity, followed by the albedo of the gas at that energy.

14.57.3 punch [helium] opacity

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 total photoionization cross section for all stages of ionization of the specified element will be punched. The photon energy is given in eV and the cross section in megabarns (10^{-18} cm^{-2}). The file name must still be specified (to get past the command line parser) but is totally ignored.

The opacity 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.

14.57.4 punch opacity figure

This version of the command creates the punch file needed to generate one of the figures used in Part II 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.

14.57.5 punch opacity shell 26 5 3

This option will punch the state-specific photoionization cross section for a subshell of any species. The first number on the command line is the atomic weight of the element, the second number the ionization stage, 1 for an atom, and the third number the subshell, between 1 and 7 representing 1*s*, 2*s*, 2*p*, etc. The punch file will contain the incident photon energy in Rydbergs followed by the cross section in cm^2.

14.58 punch optical depths

This will create a file giving the total, absorption, and scattering continuum optical depths for the computed geometry. For a spherical geometry this is the optical depth to the illuminated face, and not the total. 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 120 above).

14.59 punch _OTS

The line and continuum on-the-spot fields will be punched.

14.60 punch overview

This option punches an overview of the model and is a major output mechanism for the code. This command is useful for obtaining information concerning the structure of the cloud. 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 given as logs.

These are followed by various ionization fractions, which are given as negative 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.

14.61 punch _PDR

This command will output the quantities relevant to photodissociation region (PDR) calculations. The first column gives the depth into the cloud (cm). The second is the total hydrogen column density (cm^{-2}). The third column is the total extinction in magnitudes in the V filter, and the temperature follows. These are followed by the abundance ratios of atomic to total hydrogen, H_2 to total hydrogen, atomic carbon to total carbon, carbon monoxide to total carbon, and water to total oxygen. The total hydrogen density is defined throughout Cloudy as the total number of protons per unit volume, so a fully molecular gas will have $n(H_2)/n(H) = 1/2$.

14.62 punch pointers

The code will punch 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 as these bounds.

14.63 punch physical conditions

The physical conditions as a function of depth will be punched. The zone number is followed by the depth into the cloud (cm), the temperature (K), hydrogen and electron densities (cm^{-3}), and the radiative acceleration ($cm\ s^{-2}$).

14.64 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 output.

14.65 punch qheat

The probability distribution for the grain temperatures is punched. The first column is the grain temperature, the second column is the probability that the grain is in that particular bin, and the third column gives $dP / d\ln T$. Peter van Hoof added this command.

14.66 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.

14.67 punch recombination [option]

14.67.1 punch recombination coefficients

Total recombination coefficients, the sum of radiative, dielectronic and three-body, will be produced for all elements in the code. The rate coefficients ($cm^3\ s^{-1}$) are evaluated at the current electron temperature.

14.67.2 punch recombination efficiency

This punches the recombination efficiency for hydrogen, singlet helium and the helium ion.

14.68 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 input stream, and the predicted emission lines and column densities, can then be read at a later time, without recomputing a model, by calling the subroutine ***cdGett***. The general strategy behind

calling Cloudy as a subroutine, generating large grids of output files, and then reading this output with ***cdGett***, is described in the section on calling the code as a subroutine, in Part III of this document.

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. Only output from the second default format will be recognized by ***cdGett***.

14.69 punch source function [depth, spectrum]

14.69.1 punch source function, spectrum

The continuum source function for the local diffuse continuum will be punched. The first column is the continuum 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 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 can be greater than the Planck function.

14.69.2 punch source function, depth

The source function of the diffuse fields will be punched for all depths in 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.

14.70 punch special

If **`special`** is specified then routine ***PunSpec*** will be called. This routine can be changed to fit the circumstances.

14.71 punch tegrid

The history of the last ***nGrid*** evaluations of the heating and cooling will be punched. This is the best way to evaluate the stability of the thermal solutions.

14.72 punch temperature

The zone number is followed by the temperature and the first spatial derivatives of the temperature with respect to depth, dT/dr.

14.73 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 find zone that require surprising amounts of time to converge.

14.74 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 punch option allows the predictor's correction to be examined. The output gives the old temperature, the estimated new temperature, and the final equilibrium temperature.

14.75 punch Verner "feii.txt" [faint=0.1, range 0.1 to 0.3 Ryd]

The **`atom feii`** command (page 82 above) turns on the very large model FeII atom (Verner et al. 1999). The **`punch Verner`** command will punch all ~10^5 lines predicted by the large FeII atom at the end of the calculation. In the output of this command the upper and lower level indices are printed, followed by the log of the intensity or luminosity of the corresponding FeII 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.

14.76 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.

14.77 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.

14.78 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 many choices and this option provides a useful way to follow the internal decisions the code makes.

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 20 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. These are variables in the common block of the same name. The purpose of each is indicated in column 3.

14.78.1 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. It is not possible to start the trace on a certain zone or iteration.

Table 19
trace convergence levels and routines

Level	Routine
1	ConvPresTempEdenIoniz
2	ConvTempEdenIoniz
3	ConvIoniz
4	ConvIonizeOpacityDo

The optional number on the command line sets the level of information on the output. With no number the level is set to a high enough level to request full output. The smallest amount of output is set with 1 on the command line, which will only trace as far down as the top pressure loop. Successively deeper layers are obtained with larger numbers, 2 for the temperature solver, ***ConvTempEdenIoniz***, 3 for the electron density solver, ***ConvEdenIoniz***, 4 for the ionization solver, ***ConvIoniz***, 5 for a single evaluation of the ionization in ***ConvIonizeOpacityDo***, and larger for still deeper layers.

14.78.2 trace hydrogen

This produces a minimal trace of the hydrogen ionization distribution.

14.78.3 trace h-like [element name] [full]

This turns on extensive printout describing the physics of one of the model hydrogenic atoms. The same atom is used 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 describe that element.

14.78.4 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 atom is used for 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.

14.78.5 trace he-like [element name] [full]

This turns on extensive printout describing the physics of one of the model helium-like atoms. The same atom is used for 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 hydrogen itself. If the name of any other element appears then the printout will describe that element.

Table 20
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
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
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

15 THE OPTIMIZE COMMAND

15.1 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 fit a specified emission-line spectrum, line flux or luminosity, and/or a set of column densities. R.F. Carswell wrote much of the code for the present version and first implemented the method in Cloudy. At present the code can use any of several minimization methods to obtain a best fit to a set of observed quantities. The desired emission-line spectrum, line flux or luminosity, and/or column densities, are specified by a series of **optimize** commands. A keyword **vary** can appear on several of the commands used to specify initial conditions (Table 21) to indicate which parameters are to be varied.

Table 21
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
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
pgrains	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
stop column density	column density	def	def	0.5
stop thickness	cloud thickness	def	def	0.5
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
turbulence	turbulent velocity	def	def	0.5
xi	Ionization parameter	def	def	0.5

15.2 Commands with Vary Option

All commands with the **vary** option are listed in Table 22. The section beginning on page 143 below discusses details of some commands.

15.3 What must be specified

At a minimum, a desired emission-line spectrum, a line luminosity, or a column density must be given, along with a specification of which parameters are to be varied. The parameters to be varied during the optimization are specified by a keyword **vary** which may appear on any of the commands listed in Table 21. 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 a sample input stream in Part III of HAZY. 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.

15.4 Observed quantities

This section describes how to tell the code what observed properties to predict.

15.4.1 optimize, column density

This tells the code to try to reproduce 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
```

15.4.2 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 5007
// we want a 5007 luminosity of 10^34.8 erg / s
optimize luminosity 34.8
```

15.4.3 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 of the line lists the label Cloudy uses to identify the line. This is followed by the wavelength. Both must exactly match the identifications used inside Cloudy and printed as line labels in the output. The code will stop if this is not the case. The section, called "Lines" in Part III, gives the emission line or continuum labels and wavelengths. The third quantity is the desired 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 109 above). The last (optional) number is the *fractional* error allowed for the fit between the observed and computed values. 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 21 above.

15.5 Optimization methods

Several optimization methods have been incorporated into Cloudy. The default is to use the Subplex (Rowan 1990) method.

15.5.1 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.

15.5.2 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.

15.5.3 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.

15.5.4 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.

15.6 Controlling the optimizer

15.6.1 optimize, file= "best.in"

At the end of the optimization process the optimal input parameters are written to into a file for later use[28]. 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. This could happen if a pointer went astray or an internal variable was not properly reset. Please let me know if this happens.

15.6.2 optimize, increment = 0.5 dex

Increments are the amounts by which each variable is changed in the first step away from the initial value. The default increments preset in the code and listed in Table 21 were chosen with typical conditions in mind. The increments are logarithmic quantities that will be added to or subtracted from the initial guess. 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
```

15.6.3 optimize, iterations =75

The upper limit to the number of iterations to be performed is specified with this command. 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 normally stop on this limit. It may be necessary to increase the limit if the process is still making progress at the end of the calculation.

15.6.4 optimize, range -2.3 to 3.9

The preset limits to the range over which parameters can be varied are indicated in Table 21. 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 $-1 \le \log(Z) \le 0$ 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.

[28] This replaces the optimize punch command, that was in versions 90 and before.

```
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.

15.6.5 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.

15.7 Convergence criteria

Observables are placed into one of three categories – "spectrum", "column density", and "absolute flux"(see van Hoof 1997) . For the i^{th} observable the error estimate is

$$\chi_i^2 = \left(\frac{F_i^m - F_i^o}{\min\left(F_i^m, F_i^o\right) \ \sigma} \right)^2 \qquad (67)$$

where F^m and F^o 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 read in 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.

15.8 Other optimizer commands

15.8.1 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. This can be done with the **no vary** command. If this command is entered then the **vary** keyword on the other commands will be ignored and a single model will be computed.

15.8.2 optimize, trace start 4

This command turns on trace printout for the n^{th} time the code is called by the optimizer. Specific aspects of the trace are still controlled by the **trace** command, described on page 137 above.

15.8.3 optimize, trace flow

This command turns on trace to follow the logical flow within the optimizer.

15.9 Notes concerning commands with vary option

The keyword **vary** can appear on the commands in Table 21. Notes concerning these follow.

15.9.1 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**.

15.9.2 dlaw

Up to 5 numbers may be entered on the **dlaw** command, but only the first number will be varied.

15.9.3 elements

Either the absolute abundance of an element relative to hydrogen, or the scale factor multiplying the abundance, can be varied.

15.9.4 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.

15.9.5 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 initial hydrogen density is varied.

15.9.6 intensity

It is possible to specify all of the options on the intensity command, but it is only possible to vary the intensity itself.

15.9.7 luminosity

It is possible to specify all of the options on the luminosity command, but it is only possible to vary the luminosity itself.

15.9.8 metals

The **grains** keyword can also be specified.

15.9.9 phi(h) and Q(H)

It is possible to use the **range** option, but only the log of the photon number will be varied.

15.9.10 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.

15.9.11 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 73 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 rather than using the second number on this command if there is any danger of this confusion happening.

15.9.12 stop column density

All of the optional keywords (neutral, effective, etc.) are recognized.

15.9.13 Stop thickness

Only one thickness can be specified.

15.9.14 table stars

Only the temperature can be varied, not the gravity.

15.9.15 table star CoStar

The first parameter on the command line will be varied.

15.10 Notes concerning the optimization process

15.10.1 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 HeII λ4686/Hβ 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.

15.10.2 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 able 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.

15.10.3 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 λ 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 HeII emission only occurs for stars hotter than ~50,000 K (Osterbrock 1989), 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.

15.10.4 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 some danger of limiting the outcome to be what your expect. Finally, don't be afraid to use CPU time.

15.11 Other optimization methods?

Astrophysics is basically concerned with the inverse problem – observing 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.

15.12 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 *amoeba.in*, *phymir.in*, *powell.in* , and *subplex.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 23 shows the results of this test.

Table 23
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

16 MISCELLANEOUS COMMANDS

16.1 Overview

This section describes commands that are used to disable physical processes within the code, change its internal behavior, or to take care of housekeeping activities.

16.2 Introduction to the compile commands

The **`table stars`** command which uses incident continua from the Atlas, CoStar Rauch, and Werner stellar atmospheres, must be compiled for certain commands to work.

The following subsections describe how to set up and then compile this data. The files produced by the process must be accessible to the code when it is executed from other directories. This is most easily done by editing the path in the file **`path.c`** as described on page 163 below to point to the directory containing the compiled files.

16.3 compile stars

16.3.1 Kevin Volk's Stellar Atmospheres

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. These commands will only function if the atmosphere files are compiled as described here.

There are two steps involved in preparing the star files for use by Cloudy. First remap the original stellar atmospheres onto the Cloudy grid with the **`compile stars`** command. Next use these files by including the **`table star`** command in the input stream. The first step is only done once while installing Cloudy.

16.3.2 Preparing the Rauch stellar atmospheres

There are two abundance sets, halo and solar.

First obtain the tar files from Thomas Rauch's Web site http://astro.uni-tuebingen.de/~rauch/flux.html. Obtain the two tar files with the complete data sets. Uncompress and explode these files in their own subdirectory. This will create a large number stellar atmosphere files with names ending with the extension ".wf". Next execute Cloudy with the single command

```
compile stars Rauch initialize
```

This will create the files *rauch_halo.ascii* and *rauch_solar.ascii* that are needed for the final step. Only these two files need be retained.

16.3.3 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).

16.3.4 The final set of ASCII files

Four files are needed to compile the stellar atmospheres so that they can be used in model calculations. The *rauch_halo.ascii* and *rauch_solar.ascii* files were created above, the *Sc1_costar_z020_lb.fluxes* and *Sc1_costar_z004_lb.fluxes* files were obtained from the STScI website, and the *werner.ascii*, *kurucz.list*, and *kurucz.ascii* files come directly from the Cloudy Web site. At this stage the files you have are the following:

Sc1_costar_z020_lb.fluxes and *Sc1_costar_z004_lb.fluxes* The set of CoStar stellar atmospheres from STScI.

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 the 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).

rauch_halo.ascii and *rauch_solar.ascii* These are the sets of Rauch hot stellar atmospheres as created and described in section 16.3.2 above.

When the stars are compiled, as described next, a set of files with names ending in **.mod* are created and the files discussed above will no longer be needed.

16.3.5 Compiling the star files

Several of the stellar atmosphere files are very large and reading them using sequential access would be very slow. Direct access is used to read these files quickly. Unfortunately the format of a direct access file is machine specific. The result is that the final files are not portable although the code used to read or write them is.

Follow these steps to compile the star files:

- Move the files *kurucz.ascii, kurucz.list*, *rauch_halo.ascii*, *rauch_solar.ascii*, and *Sc1_costar_z020_lb.fluxes, Sc1_costar_z004_lb.fluxes*, and *werner.ascii* (ASCII files obtained from the web site or created above) into the directory where the **.mod* files (the files Cloudy will use to compute photoionization simulations) are going to live.
- 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.
- Six direct access files, *atlas.mod*, *rauch_halo.mod*, *rauch_solar.mod*, *costar_sol.mod*, *costar_halo.mod*, and *werner.mod* will be created. These are the files that Cloudy must access to use the **`table star atlas`**, **`table star Rauch`**, **`table star CoStar`**, or **`table star Werner`** continua.
- To execute the code from other directories it is necessary to either edit the path in the file *path.c* as described on page 163 below before compiling the source

(best) or to set the path to the directory containing these files with the **set path** command described on page 163 below.

In later photoionization calculations the code will stop before computing a model 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 star 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.

16.3.6 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 so that it is 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.

16.3.7 If the continuum binning is changed

Arguments in the calls to routine ***fill*** from ***CreatePoint*** can be changed to change the continuum binning or resolution. This is described in a section of Part II of this document. If the continuum binning is changed then it will be necessary to recompile the star files, using the **compile stars** command.

16.3.8 Compiling only one set of stars

compile only xxx. Some projects may not need all of the stellar atmospheres. The keyword **only** on the command line tells the code to search for one of the keywords **Rauch**, **Atlas**, **CoStar**, or **Werner**. Only that set of stellar atmospheres will be compiled.

16.4 compile opacities

N.B. This command does not function in the current version 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 will be located in the directory where the code is executed and must be moved to the directory where the other data files are stored.

NB 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.

16.5 compile he-like

This regenerates the table of recombination coefficients for helium-like ions that is read in by the code when it initializes.

16.6 Grains with distributed sizes

This command prepares grain opacity files that are used by the **pgrains** command (page 85 above) to simulate a multi-sized grain distribution. This compile step does not need to be done if the default size-distributed grains included in the data distribution 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 code. The *new_grain_model* file in the data distribution sub directory describes how to create new grain opacity files. It is also necessary to compile the standard 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 the size distribution and grain optical constants. The set of grain opacities are then used by Cloudy to compute temperature, charge, drift velocity, and emitted spectrum, for particular sizes within a large distribution of grain sizes.

In most cases refractive index and size distribution files included in the data from the Cloudy web site will be used. In this case sets of refractive index files (ending in ".rfi") are combined with size distribution files (ending in ".szd") to produce an opacity file (ending in ".opc"). This last file can be a single average over the entire grain size distribution, which would correspond to the old-style grain treatment. The main improvement in the **pgrains** command is the option to resolve the size distribution into a series of size bins, which results in a better treatment of the grain physics and predicted spectrum.

16.6.1 compile grains

The minimum number of grain types needed for the **pgrains** 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 will be the only step needed for nearly all applications. The following commands are only needed 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.

16.6.2 compile grain 10 bins, [filename, ism graphite]

This 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

Table 24
Grain keywords for refractive index information

Keyword	Grain type	Reference
ac1-amcarb	amorphous carbon	Rouleau & Martin 1991
be1-amcarb	amorphous carbon	Rouleau & Martin 1991
graphite	graphite	Martin & Rouleau 1991
silicate	astronomical silicate	Martin & Rouleau 1991
Grey, gray	grey grain	

properties of 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, less the extension.

The keywords given in Table 24 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 25. 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. The last two distributions specify size distributions that produce ratios of total to selective extinction that reproduce ISM and Orion observations. These files have names that end in "*.szd*".

Creating new data files A pair of double quotes on the command line is assumed to surround a 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 25
Grain keywords for size distribution

Keyword	Size distribution
0m010	0.01 micron
0m100	0.1 micron
1m000	1 micron
_ISM	ISM R = 3.1
Orion	R = 5.5 Orion distribution

16.6.3 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"
```

16.7 crash [zero, overflow, assert, NaN, bounds, undefined]

This 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 is to *not* throw an exception when division by zero or overflow occurs. Instead, the result is set to **`NaN`** (not a number) and the calculation continues. The code will only crash if this is explicitly requested by setting compiler options or masking signals on the CPU. Cloudy should be set up to crash on division by zero, overflow, or evaluation of zero divided by zero. 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.

crash overflow causes the code to divide a very large number by a very small number. The result will overflow on 32-bit machines.

crash NaN will divide zero by zero.

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.

crash bounds will cause an array to be evaluated first with an index that is less then zero and second with an index that is beyond the array end. 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.

With no further keyword the code will first use an index that is too low then one that is too high. Two keywords, **_low** and **high**, will cause only the low or high bounds to be checked.

crash undefined will cause the code to multiply a valid constant by two undefined floats. The first was taken off the heap with malloc, and the second off the stack upon entry to the routine. The ideal compiler would produce code that crashed when this occurs.

16.8 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.

16.8.1 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.

16.8.2 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.

16.8.3 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.

16.9 drive fread, gaunts, pointers, ...

The **`drive`** command causes Cloudy to enter 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.

16.9.1 drive case b

This option is to interpolate on standard 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 Case B hydrogenic emissivity. The data set is that given by Storey and Hummer (1995) although the interpolation routine is independent of theirs. A null line exits the driver.

16.9.2 drive cdLine

The code will loop over all line entered in the emission line stack, and confirm that cdLine can find each one.

16.9.3 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.

16.9.4 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.

16.9.5 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 gaunt factor routine was described in Hummer (1988) and was extended to include the full range of energy and temperature Cloudy needs by J. Ferguson.

16.9.6 drive helium

This will exercise code written by Ryan Porter, to produce energy levels and transition probabilities for helium-like ion. The results go to an output file in the local directory.

16.9.7 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.

16.9.8 drive molecules

Here it is possible to change individual molecular abundances within the iteration loop.

16.9.9 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.

16.9.10 drive pumping

This queries the routine ***conpmp***, which evaluates the continuum-line pumping probability at various optical depths (Ferland 1992).

16.9.11 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).

16.10 eden -2

This command 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} is much less than 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: The **set eden** command (page 160 below) sets the electron density. This command adds an extra component of electrons. Both commands violate charge conservation.

16.11 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 the routine ***fudge***. This routine has a single integer argument that is an index to the array of numbers entered on the command line. A call to ***fudge***(0) would return the first number[29], 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 not normally used but can be a useful way to pass numbers to temporary or trial parts of the code. All elements of ***fudge*** are initially zeroed when the code is initialized. Routine ***fudge*** is a permanent part of Cloudy and a warning is given at the end of the calculation if this function is ever evaluated. Also, the function checks that the index to the array of stored values is not larger than the number of values entered in the command line. The code will stop if too few values are entered. Extra numbers are simply ignored.

16.12 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 22 above.

There is no limit to the number of commands that can be in this initialization file, other than the total limit of 1000 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. The code will search for *cloudy.ini* in the local directory, the directory where the code is executed.

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 name does not need to be specified if the default name (*cloudy.ini*) is used.

The code can search for the file on any path, as set up with the **path** command (see page 163 below) or set in ***path.c*** (as described on page 163 below). The path must be set before the **init** command is given for the path to be used. The path does not need to be set if the current directory contains the initialization file. If the path is set then the code will look for the initialization file in the local directory first

[29] 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).

and if it is not there it will then check on the path. The code will check on the path first if the keyword **path** occurs on the **init** line.

16.13 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.

16.13.1 no Auger effect

This command turns off the Auger effect.

16.13.2 no file buffering

This command turns off file buffering so that all of the standard output goes directly to the file. This is slower than buffered output but insures that you will get output if the code crashes.

16.13.3 no charge transfer

This command turns off heavy element ionization - neutralization of hydrogen by charge transfer with the heavy elements. It *does not* turn off the effects of hydrogen on the heavies.

16.13.4 no CTHeat

This turns off heating due to charge transfer. Kingdon and Ferland (1998) describe the process.

16.13.5 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.

16.13.6 no diffuse line pumping

The diffuse continuum produced by gas within the cloud is included as a general line excitation mechanism. This command turns it off and is useful as a check on the importance of the process.

16.13.7 no FeII pumping

This turns off H I Lα pumping of Fe II.

16.13.8 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 opacity table in later calculations. The **no file opacity** command tells the code to ignore this opacity file even if it exists.

16.13.9 no fine structure line optical depths

Fine structure lines, such as the ^3P 52, 88 μm lines of O^{+2}, 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 using escape probabilities. This command turns off the treatment of 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 157 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.

16.13.10 no free free

Free-free heating and cooling are turned off with this command.

16.13.11 no grain qheat

This turns off quantum heating for all grain species.

16.13.12 no grain neutralization

Ionic recombination through grain surfaces is turned off with this command.

16.13.13 no induced processes

This command turns off induced recombination and stimulated emission for hydrogen and helium, and continuum fluorescent excitation of all lines.

16.13.14 no level2 lines

This turns off the large block of Opacity Project lines, referred to as level 2 lines within the code. 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.

16.13.15 no molecules [_H2_]

Cloudy does a molecule formation network based on Black (1978), Hollenbach and McKee (1979; 1989), and Tielens and Hollenbach 1985a (see the section on molecules in Part II, and Ferland, Fabian, and Johnstone 1994). It includes the hydrogen molecules H^+, H^o, H^-, H_2, H_2^+, H_3^+, HeH^+, and many heavy element molecules. The **`no molecule`** command turns the entire network off.

The H_2 part of the network takes an astonishingly long time to reach equilibrium. This is because there is no efficient formation mechanism for this homonuclear molecule (Bertoldi & Draine 1996). If the keyword "**`_H2_`**" (note that the "**`H2`**" is surrounded by spaces) only the heating and cooling effects of H_2 will be turned off, although the H_2 population can still be substantial. Additionally, the H_2 population

is counted in the HI column density and the weighting for the 21 cm mean temperature.

16.13.16 no on the spot

This command 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.

16.13.17 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.

16.13.18 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.

16.13.19 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 electron scattering thermal effect remaining. Bound electron scattering opacity is still included when this command is issued.

16.13.20 no scattering opacity

See page 94 above.

16.13.21 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.

16.13.22 no Stark broadening

Stark broadening (important for densities larger than ~10^{10} cm^{-3}) is treated for hydrogen lines using escape probabilities from Puetter (1981). This turns Stark broadening off.

16.13.23 no tepredictor

The code tries to predict the temperature of the next zone for constant density calculations. This stops the predictions from being used.

16.13.24 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.

16.13.25 no three body recombination

This turns off three-body recombination for the heavy elements. It is not possible to turn off three-body recombination for hydrogen or helium.

16.13.26 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.

16.13.27 no vary

This command turns off the **vary** option set on various optimization commands. For a further discussion see page 143 above, where the optimization driver is discussed in more detail.

16.14 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.

16.14.1 Convergence criteria

Three commands affect the convergence criteria. The error in the heating cooling balance is set with the **set temperature error** command (page 164 below). The error in the electron density is set with the **set eden error** command (page 160 below). The error in the local pressure is set with the **set pressure error** command (page 163 below).

16.14.2 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.

16.14.3 set colimt=0.3

This command sets the limit to the ratio CO/C_{tot}, the ratio of the carbon monoxide abundance to the total carbon abundance. The default is 0.80. The code stops when the CO abundance exceeds this limit. Instabilities in the heavy element molecular equilibrium system of equations prevent higher CO abundances from being successfully treated.

16.14.4 set spectrum [options]

This is used to modify the behavior of the **punch spectrum** command (see page 124 above). A series of **set spectrum** commands occur, and these modify the first **punch spectrum** command that follows 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 cpunch command.

set spectrum range 1 to 5 Ryd

This sets the energy or wavelength range of the continuum that is predicted. The default, when this is not used, is to produce the full continuum.

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.

16.14.5 set csupra = -12.34

This command 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 Lα is assumed to be the same. This option is used to test the code in secondary-ionization dominated cases.

16.14.6 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.

16.14.7 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^{30} cm. The number is interpreted as the linear thickness is the keyword **linear** appears.

16.14.8 set drmax 11.2

This command sets the largest allowed zone thickness. The argument is the log of the thickness in cm, and the default is 10^{30} cm.

16.14.9 set drmin 11.2

The number is the log of the minimum zone thickness (in cm). The default value is 10^{-20} cm.

16.14.10 set eden [error, solver, value]

These commands affect the electron density, or how it is determined.

set eden error 0.01 This command sets the convergence criterion for the electron density. The number is the largest relative change in the electron density that can have occurred during an iteration with that iteration considered to be converged. The number on the line is interpreted as the relative change in the electron density if it is positive and the log of this change if negative. The default value is 0.01.

set eden solver new This command tells the code which solver to use for the electron density convergence. There are currently two options, the default simple method, and the new method.

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.

Note: The **eden** command (page 154 above) sets an extra source of electrons. This command sets the electron density itself. Both commands violate charge conservation.

16.14.11 set flxfnt -20

The highest continuum energy that needs to be considered in equilibrium calculations is lower for relatively soft continua, such as H II regions, than for X-Ray sources, such as AGNs. The criterion used to choose the highest energy to be considered ν_{high} is that $\nu f_\nu(\nu_{high})/\nu f_\nu(\nu_{peak}) <$ ***flxfnt***, where ν_{peak} is the frequency where the continuum reaches its maximum $\nu f\nu$. ***FluxFaint*** is normally 10^{-10}. This command changes the value of ***FluxFaint***. The argument is the log of the value.

16.14.12 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.

16.14.13 set matrix [linpack, matin1, veclib]

This command specifies which matrix inversion routine will be used to solve for the populations of multi-level atoms. The keywords recognized are **bevington, matin1, veclib**, and **linpack**. The default is **linpack**. Code exists to call one of the routines from the ***bevington, linpack*** or ***veclib*** set of software. The source for the ***linpack*** routine is included in the distribution, but the ***veclib*** source is not. Two small dummy routines called ***dgeco*** and ***dgesl*** are included in the distribution, and must be deleted if the ***veclib*** routines are to be used. It will then be necessary to link the code into your system's version of ***veclib***.

16.14.14 set nchrg 3

In version C96 the grain physics was updated to the processes described in Weingartner & Draine (2001), except that the code 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 model is discussed in van Hoof, Weingartner, et al. (2001). 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.

16.14.15 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***.

16.14.16 set nend 500

This command sets the default limit to the number of zones that will be computed. The preset default value is 600, 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 108 above). The only difference between these two commands 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. The **set nend** command makes it possible to increase the default limit to the number of zones when computing very large grids of models. Some of these may require more than the current default limit to the number of zones. 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.

This command has no effect if the **stop zone** command if also entered.

16.14.17 set nFnu [incident_reflected] [incident_transmitted] [diffuse_inward] [diffuse_outward]

The continuum emitted by the central object and transmitted through the cloud, and the diffuse continuum produced by the cloud, will be printed if the command **print continuum** (page 112 above) is entered. If that command is entered then the diffuse continua will be printed with the label *nFnu* and the transmitted incident continua are is 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 the **print continuum** does not also need to be given. 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*** Include the incident reflected continuum.

incident transmitted or ***incident_transmitted*** Include the incident transmitted continuum.

diffuse inward or ***diffuse_inward*** Include the diffuse inward continuum.

diffuse outward or ***diffuse_outward*** Include the diffuse outward continuum.

All desired continua must appear 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
```

16.14.18 set nmaps 50

This is used to control 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. This number can be reset with this command.

16.14.19 set numerical derivatives

This tells the code to use numerical rather than analytic derivatives for changes in the heating and cooling functions. The default is to use the analytic derivatives.

16.14.20 set path ="/usr/home/cloudy"

This command sets the path Cloudy will use to look for various ancillary files. If the path is not set then the current directory will be used. The path begins with either a double quote, and must end with a double quote. 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**. Then, the path will never need to be set. To do this examine the file and follow its instructions.

16.14.21 set phfit [1995, 1996]

The key **1995** tells the code to use photoionization cross sections from Verner and 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.

16.14.22 set pressure error 0.01

This command sets the convergence criterion for the total pressure. The number is the largest relative error in the pressure. The number on the line is interpreted as the linear error if it is positive and the log of this error if negative. The default value is 0.01.

16.14.23 set PunchLWidth 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 119 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

$$\upsilon F_\upsilon(\text{total}) = \upsilon F_\upsilon(\text{continuum}) + \frac{c}{PunchLWidth} I(\text{line}) \qquad (68)$$

where c is the speed of light. The **set PunchLWidth** command sets the value of ***PunchLWidth***. 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 entered. Note the space after the 'c', and two spaces before.

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 the **punch** commands. Turbulent velocities are set with the **turbulence** command.

16.14.24 set resolution 0.1

This changes the resolution of the continuum energy mesh by a constant scale factor. The resolution of the continuum mesh is contained in the data file *continuum_mesh.dat*, which is part of the data directory. Permanent changes to the continuum 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. 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.

16.14.25 set temperature [floor, solver, tolerance]

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 agent.

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 for testing).

set temperature error The balance between the heating and cooling rates sets the equilibrium electron temperature. This command is used to change the error tolerance allowed in the heating-cooling match. The number is the largest fractional error allowed, and is interpreted as the tolerance itself if it is positive, and the log of the tolerance if it is less than or equal to zero. The default tolerance is a fractional error of 0.02. 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.

16.14.26 set test

This command sets the logical variable ***lgTestOn*** to true. It provides the facility to conditionally run test code somewhere in the main body of Cloudy.

16.14.27 set trim –9 [upper lower, small]

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 smallest relative abundance of ions below the peak is changed with the **lower** keyword. The default relative abundances 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 fractional abundance of the lowest or highest abundance to consider.

Generally, line excitation energies involved with stages of ionization higher than the peak ionization strongly exceed the ambient temperature, so these will have little influence on the calculated temperature or spectrum. This is not true for lines formed from ions below the peak distribution. Ionization stage trimming can also cause problems if the ionization of the gas *increases* with depth into the cloud. This occurs if the gas density falls off much faster than r^{-2}. In this case it will be necessary to tell the code to consider a far wider than usual range of ionization. Specifying a very small number for the range can do this. For convenience the keyword **small** can be used to set the smallest abundance to just above the machine's floating precision limit.

16.14.28 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 the one deduced from the [O IIII] 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 ~ 10^5 cm^{-3}. The code will not print the results of this analysis if the density is higher than the value of ***tsqden***, currently 10^7 cm^{-3}. This upper limit is changed with this command. The number on the line is the log of the highest hydrogen density for which this analysis will be performed.

16.14.29 set WeakHeatCool 0.02

This command resets the threshold for the weakest coolant or heat source output with the **punch heating** (page 126 above) or **punch cooling** (page 125 above) commands. The default is 0.05. The number entered is normally linear but interpreted as a log if it is negative.

16.15 test

This single command issues a series of commands to run a single-zone constant-temperature model. This is an aid to make it easy to generate a quick model while within a debugger or to make quick tests of the code. 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 one zone and one iteration. Many **assert** commands are used to verify predictions.

16.15.1 test feii

When the **feii** keyword appears the large FeII atom (page 82 above) will be computed and checked as well.

16.15.2 test large

When the **large** keyword appears the model hydrogen atom (page 78 above) will be set to its largest number of levels.

16.16 assert commands

These are a series of commands that tell the code what results are expected. At the end of the calculation the code will then compare the expected and actual results and return an error condition 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 of models are computed every night and a log is kept of all asserted results. An email warning is set if any asserts fail. This insures both 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.

16.16.1 assert column [_CO_, H2] 18.9

This checks the predicted column density for either CO or H_2. The molecule label must be either **_CO_** or **_H2_** , the labels CO or H2 surrounded by spaces. It is important that log of the column density (cm^{-2}) appear *after* the label because of the 2 in H2.

16.16.2 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 entered is the log of the secondary ionization rate.

16.16.3 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 to confirm that departure coefficients predicted by one of the hydrogenic isoelectronic sequence atoms 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.

assert departure coefficients, FeII, error=0.08 The keyword **FeII** tells the code to confirm that departure coefficients predicted by the large FeII 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 difference deviation of the departure coefficient from unity.

16.16.4 assert depth 13.2

This checks that the thickness (cm) of the computed model is equal to the expected value. The argument is always the log of the depth in centimeters. The thickness is the length between the illuminated and shielded faces.

16.16.5 assert eden 9

The number is the log of the electron density.

16.16.6 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.

16.16.7 assert hheicf –0.013

This provides a way to confirm the helium – hydrogen ionization correction factor. The number is the linear difference between the atomic fractions of helium and hydrogen.

16.16.8 assert ionization fraction oxygen 3 –3.45, error 0.1, weight = radius

This checks that the computed mean ionization of an element agrees with the expected value. 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. If this number is less than or equal to 0 it is interpreted as a log, and as the linear ionization fraction if it is positive. The linear relative error is optional and is the third number on the line. The default of 0.05 will be used if no number appears. 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.

16.16.9 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.

16.16.10 assert line "q(h)" 4861 < 1.01

This will compare the predicted and expected relative intensities of any line. The string that gives the line label must appear between two double quotes and 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 standard output at the end of a calculation.

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. The optional last number is the relative error that is expected. The code will

report failure if the absolute magnitude of the computed and expected value divided by the expected value is larger than this relative error. The default is 0.05.

16.16.11 assert line [luminosity intensity] "q(h)" 4861 38.91

This will compare 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.

16.16.12 assert niter < 4

This checks the number of iterations required in a calculation. This is an upper limit.

16.16.13 assert nzone < 135

This checks how many zones were needed. This is usually an upper limit.

16.16.14 assert pressure error < 0.01

The number is the relative standard deviation of the total pressure across the model. This should be a small number in a constant pressure model. In most cases the number will be an upper limit. The number is the ratio of the standard deviation of the pressure to its mean value. The number is interpreted as a log if it is negative.

16.16.15 assert temperature hydrogen 2, over volume, 4.02 error 0.01

This provides a method of verifying the computed mean temperature of any species. The name of one of the elements or the keyword **`grains`** must appear on the line.

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 less than or equal to 10. It will be linear if the keyword **`linear`** appears. The last number is the relative error.

The code computes two means. These are averages 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.

16.16.16 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. In this case the first number on the line is an index giving the grain type. This index is the order in which the grains were specified in the input stream. The first grain that occurs in the input stream is number 1. The second number on the command is the temperature, interpreted as a log if less than or equal to 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.

16.16.17 assert radius 18.2

This checks that the outer radius (cm) of the computed model is equal to the expected value. The argument is always the log of the outer radius in centimeters.

The radius is the distance from the center of the central object to the shielded face of the cloud.

16.16.18 assert thickness 13.2

This checks that the thickness (cm) of the computed model is equal to the expected value. The argument is always the log of the thickness in centimeters. The thickness is the length between the illuminated and shielded faces.

16.16.19 assert velocity 7.6 km/s

This checks that the final velocity of a non-static model is equal to the expected value. The quantity is the expected final velocity in km/s.

17 REFERENCES

Abbott, D. C. 1982, ApJ, 259, 282
Adams, T. 1972, ApJ, 174, 439
Aldrovandi, S., & Pequignot, D. 1972, A&A, 17, 88
Aldrovandi, S., & Pequignot, 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).
Aller, L. H. 1984, in *Physics of Thermal Gaseous Nebulae*, (Dordrecht: Reidel).
Aller, L. H., & Czyzak, S. J. 1983, ApJS, 51, 211.
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
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
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
Balbus, S. A., & McKee, C. F. 1982, ApJ, 252, 529
Baldwin, J., Ferland, G. J., Martin, P. G., Corbin, M., Cota, S., Peterson, B. M., & Slettebak, A. 1991, ApJ, 374, 580
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., 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
Bechtold, J., Weymann, R. J., Lin, Z., & Malkan, M. A. 1987, ApJ, 315, 180
Bell, K. L., Kingston, A. E., & McIlveen, W. A. 1975, J. Phys. B 8, 358
Berger, M. J., & Seltzer, S. M. 1965, NASA SP-3012
Bergeron, J., & Collin-Souffrin, S. 1971, A&A, 14, 167
Berrington, K., & Pelan, A. 1995, A&AS, 114, 367
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
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., Lamothe, J., Momjian E., Verner, E., Vinkovic, D. & Ferland, G. 1998 PASP, 110, 1040
Bottorff, M. C., Ferland, & Gary J., 2000, MNRAS 316, 103-106
Bottorff, Mark, Ferland, Gary, Baldwin, Jack, & Korista, Kirk, 2000, ApJ, 542, 644-654

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 Univ 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
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
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
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. 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

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., & 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., 1999, *Atomic Molecular Physics Handbook*, ed by Gordon Drake (AIP Press,)
Drake, S. A., & Ulrich, R.K. 1980, ApJS, 42, 351
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
Fabian, A. C., Pringle, J. E., & Rees M. J. 1976, MNRAS, 175, 43
Federman, S. R., et al. 1993, ApJ, 413, L51
Ferguson, J., Ferland, G. J., & A. K. Pradhan, 1995, ApJ, 438, L55
Ferguson, J., & Ferland, G.J. 1997, ApJ, 479, 363
Ferguson, J. W., Korista, K. T., Baldwin, J. A., & Ferland, G. J. 1997, ApJ, 487, 122
Ferguson, Jason W., Korista, Kirk. T., and Ferland, Gary J., 1997, ApJS 110, 287-297.
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
Ferland, G. J. 1993, ApJS, 88, 49
Ferland, G. J. 1999, PASP, 111, 1524
Ferland, G. J. 1999a, in ASP 162, Quasars and Cosmology, ed G Ferland & J Baldwin
Ferland, G. J., 1999b, ApJ 512 247-249
Ferland, G.J., 2000, RMxAC, 9, 153
Ferland, G.J., 2001, PASP, 113, 41
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., Pequignot, 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., 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
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)
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
Habing, H. J. 1968, Bull. Astr. Inst. Netherlands 19, 421
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
Heitler, W. 1954, *The Quantum Theory of Radiation* (Oxford: Oxford University Press)
Hjellming, R. M. 1966, ApJ, 143, 420
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. 1999 Rev Mod Phys 71, 173
Hubbard, E. N., & Puetter, R. C. 1985, ApJ, 290, 394
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, 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
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
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
Lame N. J., & Ferland, G. J. 1991, ApJ, 367, 208
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
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
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
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
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
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
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
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., & 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
Oliveira, S., & Maciel, W. J. 1986, Ap&SS, 126, 211
Oliva, E., Pasquali, A., & Reconditi, M. 1996, A&A, 305, 210
Osterbrock, D. E. 1951, ApJ, 114, 469
Osterbrock, D. E. 1989, *Astrophysics of Gaseous Nebulae & Active Galactic Nuclei*, (Mill Valley; University Science Press)
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
Pequignot, D. 1986, *Wordshop on Model Nebulae*, (Paris: l'Observatoire de Paris) p363

Pequignot, D., & Aldrovandi, S.M.V. 1986, A&A, 161, 169
Pequignot, D., Stasinska, G., & Aldrovandi, S. M. V. 1978, A&A, 63, 313
Pequignot, D., Petitjean, P., & Boisson, C. 1991, A&A, 251, 680
Peterson, J. R., Aberth, W., Moseley, J., & Sheridan, J. 1971, Phys Rev A, 3, 1651
Porquet, D., & Dubau, J. 2000, A&AS, 143, 495
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
Rauch, T. 1997 A&A, 320, 237
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
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
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., & 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
Sellmaier, F. H., Yamamoto, T., Pauldrach, A. W. A., & Rubin, R. H. 1996, A&A, 305, L37
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., & 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
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
Stecher, T. P., & Williams, D. A. 1967, ApJ, 149, 29
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/)
Swings, P., & Struve, O. 1940, ApJ, 91, 546
Tarter, C. B., & McKee, C. F. 1973, ApJ, 186, L63
Tarter, C.B., Tucker, W.H., & Salpeter, E.E., 1969, ApJ, 156, 943
Tielens, A. G. G. M., & Hollenbach, D. 1985a, ApJ, 291, 722
Tielens, A. G. G. M., & Hollenbach, D. 1985b, ApJ, 291, 746
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 in press (astro-ph/0107183)
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
Watson, W. D. 1972, ApJ, 176, 103
Weingartner, J.C., & Draine, B.T., 2001, ApJS, 134, 263
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
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, R. E. 1967, ApJ, 147, 556
Williams, R. E. 1992, ApJ, 392, 99
Wills, B., Netzer, H., & Wills, D. 1985, ApJ, 288, 94
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

18 INDEX

—V—

—W—

—X—

—Z—