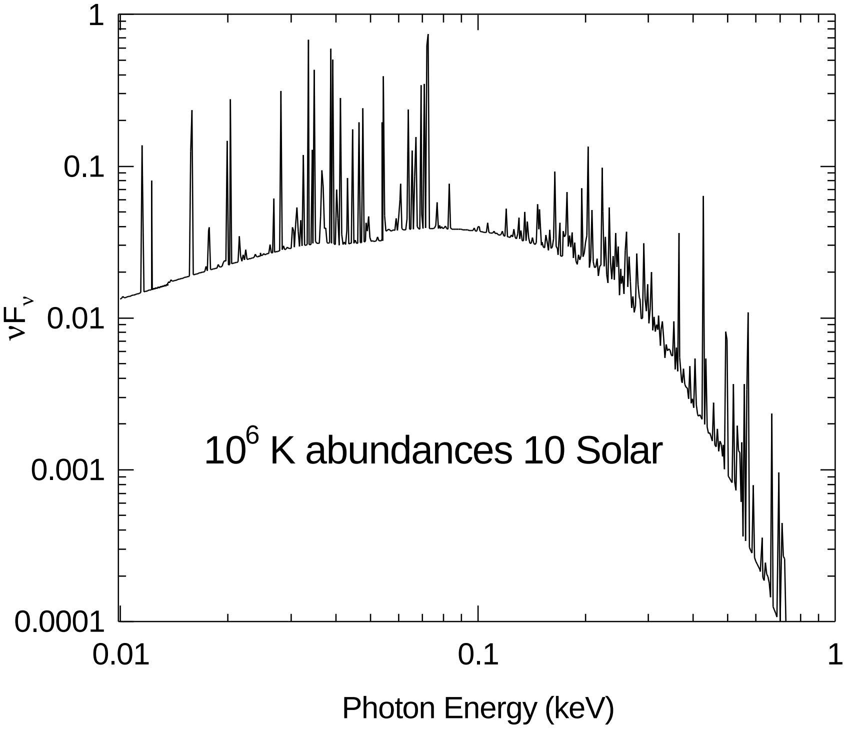
Hazy I

a brief introduction to Cloudy 90

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., 1996, *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 1978-1997 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.

October 21, 1997.

Cloudy 90

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, 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 in **readme.htm** file that is included in the source distribution tar file. First obtain the files from the web site. Next decompress and either view or print the documents, then decompress and compile the Fortran source. Compile the stellar atmosphere models, if these are to be used. Finally, verify the behavior of the code by running the test cases shown in Part III.

CLOUDY is designed to run on a variety of platforms. Until 1994 (through version 84) CLOUDY was written in strictly ANSI - compliant FORTRAN 77 language. The current version is written in a hybrid language using popular Fortran 90 extensions. The code is designed to run on IEEE 32-bit CPUs. It has been tested on a variety of machines including a Cray, Sparc, SGI Indy, DEC Alpha, HP, and a PC-compatible. 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 10,000 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^{-5} Ryd (λ~1 cm) and an energy of 100 MeV ($\sim 7.354\times 10^{6}$ Ryd) if possible. (In much of the following discussion 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, some of the Mihalas and Kurucz 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 a 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 a 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 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 the emission line luminosities will be predicted.

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, which sets the covering factor 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 (defined on page 8) 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. A wind model, in which case a Sobolev (large velocity gradient) model is assumed, can be computed with the **`wind`** command.

1.5. What is Computed and Printed

CLOUDY is driven by a set of command lines that 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 Fortran unit 5 is read for input, and unit 6 is used for output. It is also possible for a larger program to drive CLOUDY directly by treating it as a subroutine.

The program begins by echoing the input commands (except for lines beginning with an #, %, or *; these lines are treated as comments and are ignored). The input stream ends with either a blank line or the end-of-file. Some properties of the incident radiation field, such as luminosity and number of photons in certain frequency ranges, are then printed.

CLOUDY works by dividing a 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. The physical thicknesses of the shells are continuously adjusted to ensure this. Each of these shells 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 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 following line give the ionization fraction and level populations of hydrogen excited states (2s and 2p are printed separately). 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 fully self checking, to ensure that its range of validity is not exceeded. It will complain if this occurs, 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 often used today to generate grids of thousands of models, making it impossible to check individual models one by one. Next, optional plots of the incident and emergent continua, gas opacities, or heating-cooling curves, etc., may follow.

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). 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 (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.

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 at a crucial time. 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.

My collaborators, J. Baldwin, R. Boyd, E. Capriotti, R. Carswell, S. Cota, M. Elitzur, A. Fabian, C. Gaskell, F. Hamann, R. Johnstone, K. Korista, D. Lambert, W. Mathews, J. Mihalszki, R. Mushotzky, S. Persson, B. Peterson, M. Rees, G. Shields, J. Shields, and J. Truran contributed to the development of various parts of CLOUDY. 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, D. Verner, and K. Volk. Sections of the code are taken from public domain software, as acknowledged in this document and in the source. Comments or suggestions which led to the improvement of CLOUDY were made by N. Arav, D. Balsa, S. Daines, M. Diaz, F. Elizalde, B. Espey, C. Everall, A. Fabian, M. Gaskell, P. van Hoof, R. Johnstone, K. Kwitter, H. Lee, S. Morris, T. Oliva, G. Perola, T. Plewa, L. Puchnarewicz, D. Rigopoulou, S. Schaefer, C. Stevenson, M. Voit, K. Volk, T. Woods T. Yamamoto and W. Zheng.

Version 90 of CLOUDY is the result of a team effort, involving Kirk Korista, Dima Verner, Jason Ferguson, Jim Kingdon, and Katya Verner. The bugs are mine, but this version would not have been possible without their help. Peter van Hoof very carefully went over every dark corner of version 84, and uncovered many problems.

The development of CLOUDY would not have been possible without twenty 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, and most recently AST 96-17083. 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.

2.2. Continua

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

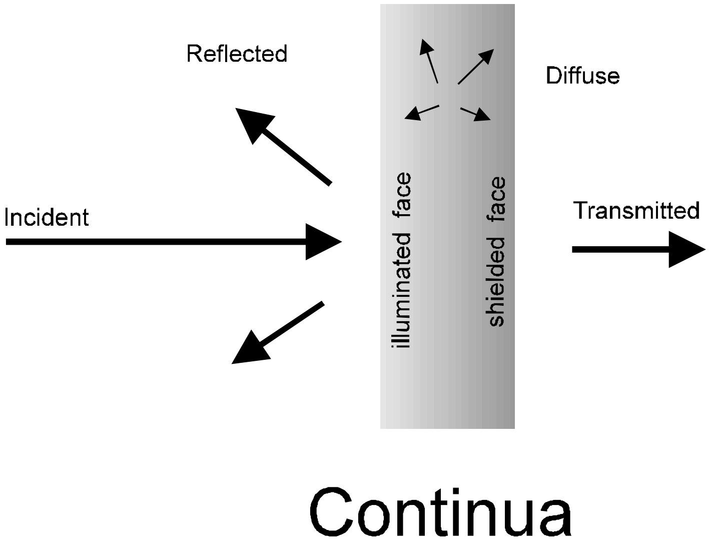


Figure 1 This figure illustrates several of the continua that enter in the calculations.. continua

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 beginning 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 field) 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 nebula photoionized by an external continuum, 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 local intensity is often dominated by the nearly isotropic diffuse field. As a result the diffuse fields can be treated by lower order approximations in a nebula than in a stellar atmosphere.

2.2.3. Transmitted continuum

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

2.2.4. Reflected continuum

The reflected continuum is the continuum emitted from the illuminated face of the cloud in the direction towards (i.e., within 2π sr of) the source of the incident continuum. This continuum is only computed for an open geometry (defined below).

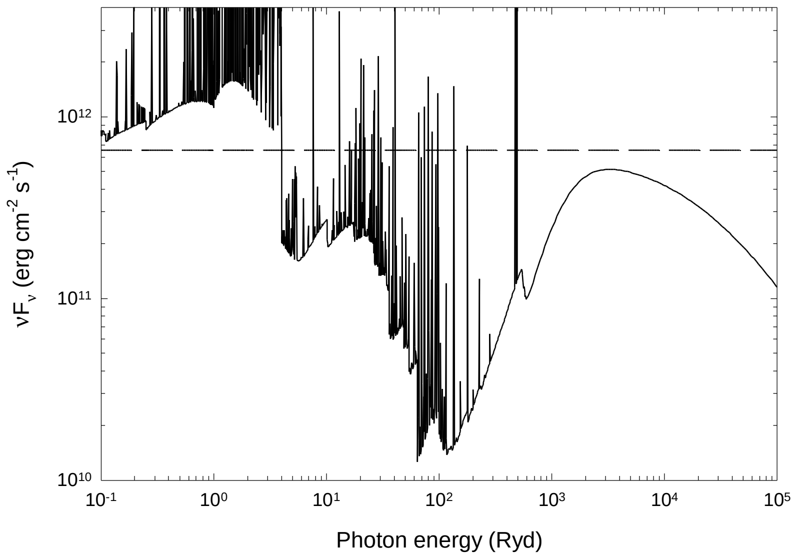


Figure 2 This figure shows the incident (dashed) and reflected (solid) continua. reflected

Figure 2 shows a plot of the incident and reflected continua for a situation similar to 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 is clearly shown. The input stream for this model is the test file **`reflector.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 thickness over 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. Radius is the distance from the center of symmetry, usually the center of the central object. 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 $0 \le \Omega / 4\pi \le 1$, 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 which actually enter the calculations, and describes the code variables in greater detail.

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 refers to the regions containing gas, while surrounding regions are assumed to be in vacuum. The specific effects of a filling factor are described by Osterbrock and Flather (1959) and section 10.5 on page 66 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 referred to as Δ*r*, and the current radius as *r*.

2.3.6. Hydrogen density

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

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

2.3.7. Column densities

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

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

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 can be identified for the geometry and its influence upon the calculations. Here, they are referred to as "open" and "closed" geometries. 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.

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 in the Crab Nebula. In this case Ly β and higher hydrogen 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 (described below) is not specified.

Closed geometry. Here, emission-line gas covers ~4π sr as seen by the central star. If the star is small relative to the nebula, then all diffuse fields, which escape from the illuminated face of the cloud towards the star, go on to strike the other 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.

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 causes 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 $\Omega/4\pi$~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 and the cloud is ionized throughout. The cloud is optically thin to the incident continuum. In this case the intensity of a recombination line is set by the volume and density of the cloud and is not directly related to the luminosity of the ionizing continuum.

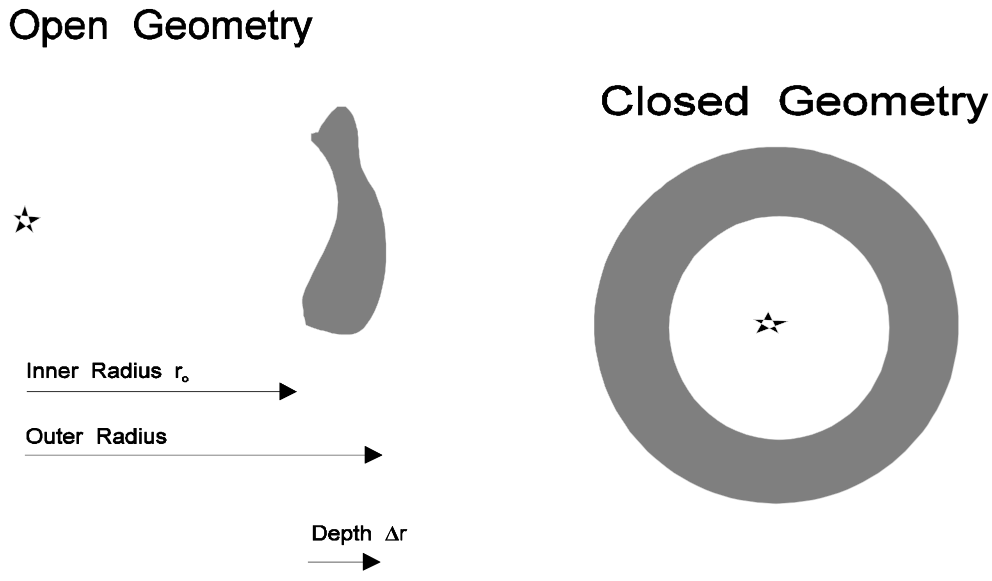


Figure 3 This figure shows the two limiting geometries that can be assumed in the calculations. The hatched area represents nebular gas. An open geometry is the default, and a closed geometry will be computed if the “sphere” command is entered. geometry

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. Nearly all of the incident continuum has been absorbed by the cloud, which is optically thick to hydrogen-ionizing portions of the incident continuum. In this case the intensity 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 surface 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 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, and will generate a warning if it is not.

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 (≈10^4 K), the longest is the H^+ recombination time scale,

$$T_{rec} = \frac{1}{a\left(T_e\right)n_e} = 1.15\, t_4^{0.8}\ n_9^{-1} \text{ hours} \tag{3}$$

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}.

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 pace-setter for H_2 formation in grain-free environments. This time scale is roughly given by

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

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

CLOUDY is not appropriate for the treatment of situations where conditions change more rapidly than these two 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 section 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} has undergone three revisions between 1984 and 1991, each by a factor of 10, because of theoretical uncertainties in positions of autoionizing states that have not been observed (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 optical and ultraviolet lines in moderate ionization nebulae. A series of papers by Oliva and collaborators (see Oliva et al. 1996) presents observational evidence that the current collision strengths of moderate ionization far infrared lines are seriously in error. 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). The photoionization cross sections used by CLOUDY should be as accurate as 10%. All inner shell multi-electron processes are included (Kaastra and Mewe 1993).

3.3.3. Recombination rate coefficients

Reliable recombination coefficients do not now exist for all stages of ionization of astrophysically abundant elements. Recombination from closed shell species is accurately known (Verner and Ferland 1996). Unfortunately low-temperature (through low-lying autoionizing states) dielectronic recombination rate coefficients have not been computed for most third row and higher elements. For these, CLOUDY uses the guestamates described on page 130.

3.3.4. Charge transfer

The rate coefficients for charge transfer are another uncertainty in the atomic and molecular data base. 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 semiempirical potential curves (Kingdon and Ferland 1996). These are thought to be no more accurate than a factor of three. Even the best quantal calculations of charge transfer rate coefficients are not thought to have an accuracy much better than 50 percent. Unpublished tests suggest that these uncertainties affect many line intensities at the ~20% level, and some 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^{-5} 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 the 30 elements included in the calculation, 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 low and moderate stages of ionization, and fine structure optical depth, 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. Hydrogen and atoms and ions of helium 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 CLOUDY's hydrogen and helium atoms and the hydrogen molecules do 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 elements is presently not as complete as hydrogen and helium. 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^{-4} 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 (densities of 10^{-6} cm^{-3} and 10^{20} cm^{-3} do not work on IEEE 32-bit machines for this reason). 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 the approximate treatment of 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 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. Hydrogen

Hydrogen is treated as a many-level atom. The 2s and 2p states are treated separately, so Lyα and 2-photon emission are computed properly in low-density nebulae. 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. Details are given in Part II, and in Ferguson and Ferland (1997).

The full set of hydrogen level balance equations is solved by working with LTE departure coefficients. As a result, the departure coefficient of the ground state diverges for very low temperatures when the gas is ionized yet cold, so it is not possible to treat hydrogen line transfer in detail for temperatures below 1000 K on 32-bit machines. For these low temperatures CLOUDY uses a special set of approximations (fitted to the results of Martin 1988) to solve for the hydrogen ionization balance and predict emissivities. The intensities of these lines are reliable for temperatures above ~10^2 K as long as radiative transfer and optical depth effects are not important. (It is unlikely that significant H II emission will occur for gas colder than this 10^2 K limit.) If such low temperatures occur, and the matrix solution is not used, then a comment is printed after the last zone calculation. None of these issues require user intervention.

Finally, the hydrogen density used by CLOUDY is the *total* hydrogen density, usually referred to by the label ***hden***, and is 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

3.9.1. Model atoms

Helium is treated as three separate atoms or ions; ionized helium (a ten level atom), the singlets (a ten level atom), and the triplets (presently a five level atom). The treatment of the three ions is entirely analogous to that of hydrogen; it goes to LTE in the limits of large photon or particle densities. The low-temperature limit to the matrix treatment of the He^o singlets and He^+ are 1000 K and 1500 K respectively. Below these temperatures approximations to case B results are used.

These model singlet and He^+ atoms assume complete *l*-mixing for levels greater than 2. This is not a good approximation for low densities. For instance, the intensity of He II λ4686 is exact at high densities, but differs by nearly 40 percent from low density case B predictions. For the time being, it is probably better to use the case B predictions, indicated by the line label “Ca B”, rather than the results of the multi-level atom calculations, for He II line emission at low densities, when line transfer and collisional excitation are not important. The predictions of the ten-level atom are better when either process is important, or the density is high enough for complete l-mixing to be a good approximation.

3.9.2. 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 Lyα, all ionize hydrogen, and the He II Lyα 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.

3.10. 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.

3.11. 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.

3.12. Reliability

Several issues affect the general question of 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 nebulae. 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, will remain an uncertainty.

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 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.”

3.13. 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 Lyα 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. 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.

Table 1 Default Conditions

Variable	Value	Quantity
rdfalt	10^{25} cm	default inner radius
router	10^{30} cm	default outer radius
telow	2.8 K	lowest initial temperature
tehigh	10^{10} K	highest allowed temperature
tend	4000 K	lowest temperature to allow in model
toler	0.02	tolerance in heating-cooling match
faint	10^{-3}	relative intensity of faintest line to print
emm	1.001×10^{-5} Ryd	low energy limit to radiation field
egamry	7.354×10^{6} Ryd	high energy limit to radiation field
nend	600	limiting number of zones
colend	10^{30} cm^{-2}	limiting total hydrogen column density
colpls	10^{30} cm^{-2}	limiting ionized hydrogen column density
colnut	10^{30} cm^{-2}	limiting neutral hydrogen column density
grains?	no grains	grain mixture

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.

4.3. Command Format

When executed as a stand-alone program, CLOUDY reads Fortran unit 5 for input and produces output on unit 6. 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 the commands used 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.

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.

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 by column 80, the end-of-line, a colon ":", a semi-colon ";", 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.

Up to 1000 separate commands may be entered. This limit is established by the variable ***nkrd*** that appears in several parameter statements throughout the code. The input stream ends with either a blank line or the end-of-file.

CLOUDY can also read its own output as an input stream. As described in the section "Output" of Part IV, 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. This can be done by copying the printed command lines, and using this as an input file. The input parser will handle removal of the leading spaces and asterisk.

4.3.1. 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 of examples are shown below.

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

These square brackets indicate only that the parameters are optional, and the brackets need not be placed on the command line (they are totally ignored and have no effect 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 blank, 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.

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

4.3.2. The continue option

It is sometimes not 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 set by the variable ***nkrd*** (see page 19).

4.3.3. 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 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

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

4.3.4. Comments

Comments may be entered among the input data in several ways. Anything on a line that occurs after a colon, semi-colon, or percentage sign 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 115.

4.3.5. 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 sub-keywords **_log** and **linear** to force one or the other interpretation to be used.

4.3.6. 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.

```
title - this is the input stream for a planetary nebula
* this is the temp and total luminosity of central star
black body, temp = 100,000K, luminosity=38;[log(L)- ergs/s]
radius 17   ;log of starting radius in cm
hden 4      ;log of hydrogen density - cm^-3
filling factor 0.3 ;set a filling factor of 30 percent
sphere ; this is a sphere with large covering factor
```

4.4. 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 138) these commands can be easily accessed from other directories. The **init** command is fully described in the section beginning on page 132.

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.

The default name for the initialization file is **cloudy.ini**. This file will be used if no name is specified on the command line. The code will search for **cloudy.ini** in the local directory when the **init** command is encountered. Other file names can be specified with the **file** option. The code can search for the file on any path, as set up with the **path** command (described on page 138).

This command is a very useful 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. 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 file=“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^{-5} 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, its shape and its intensity (surface flux at the illuminated face of the cloud) specify a single continuum. The two are specified independently in most cases, although both can sometimes be specified by the same command (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 33 of this document.

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 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 25 of this document.

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.[1] CLOUDY will stop if more than 10 continua are entered. This limit is set by the variable ***limspc*** that occurs in several parameter statements throughout the code. Increase ***limspc*** everywhere if more than 10 continua are required.

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.

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

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

```
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}.

5.3.2. Keeping shape and intensity commands together

It is not absolutely necessary to keep the ordered pairs of shapes 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 set the luminosity of the cosmic background. This problem cannot occur if the shape and intensity commands are kept together, as in the previous example.

Table 3

Commands Specifying Shape *and* Intensity
background
blackbody, energy density
blackbody, LTE
blackbody, luminosity
blackbody, radius
fireball
table ISM

Table 3 is a list of all continuum shape commands that *also* enter a luminosity or intensity.

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}).

The intensity of the predicted emission lines will be either the luminosity radiated by a shell covering Ω sr or the intensity (energy radiated per unit area of cloud). Here Ω is the angular coverage of the nebula and $\Omega / 4\pi$ (with a default value of unity) is the covering factor. Which is predicted depends on whether the incident continuum is specified as a luminosity or intensity.

Internally, CLOUDY actually works with continuum fluxes in units similar to photons cm^{-2} s^{-1} Ryd^{-1}, to avoid single-precision floating-point exponent limits on 32-bit IEEE machines.

6.2. Intensity vs luminosity commands

The incident continuum can be specified as an intensity (quantity per unit area of cloud) or luminosity (quantity radiated by the central object into 4π sr). Each of the following commands is characterized as either an intensity or luminosity command.

6.2.1. Luminosity commands

If the continuum is set as a luminosity then a starting radius *does* need to be specified, and the predicted emission-line spectrum will also be 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 is set as an intensity 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 results. The predicted emission-line spectrum is also given as intensities. A starting radius may be specified, and if so then the resulting geometry may be spherical, plane parallel, or intermediate. Both absolute and relative intensities of lines have little dependence on the covering factor.

6.3. The range option

For most of the intensity commands the default is for the quantity entered to be the number of photons or luminosity *in hydrogen ionizing radiation* (13.6 eV $\le h\nu \le$ 100 MeV). 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 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^{-5} 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 will be used.

The following are some examples of the use of the range option in modifying the range on the **luminosity** command. The default condition on the **luminosity** command is for the single parameter to be the log of the luminosity (erg s^{-1}) in hydrogen ionizing ($1 \text{ Ryd} \le h\nu < 7.354\times10^{6}$ Ryd) radiation.

```
* 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 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 specific luminosity at 5550Å by (Allen 1976, page 197)

$$n\,L_n(5500\text{Å}) \approx 2.44\times10^{35}\times10^{-M_V/2.5} \quad \text{erg s}^{-1} \quad . \qquad (7)$$

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

This is an intensity command.

6.6. F(nu) = -12.456 [at .1824 Ryd]

This command allows the flux density F_ν to be specified at an arbitrary energy. The first number is the log of the surface flux density 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 $4\pi J_\nu$.)

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^{-5} 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 command specifies the integrated surface energy flux, and is the per unit area equivalent of the **`luminosity`** command (see page 29). Unlike the majority of the commands, the first five characters of the line must be entered. The number is the log of the surface energy flux (erg cm^{-2} s^{-1}) at the illuminated face of the cloud

$$I = \int_{n_1}^{n_2} 4p\ J_n\ dn \quad . \tag{8}$$

The number is interpreted as the intensity itself, rather than a log, if the optional keyword **`linear`** appears on the line. Note that the quantity referred to as the intensity here is actually 4π times larger than the intensity defined in most radiative transfer texts (there, the intensity I is per unit solid angle).

The default range is over hydrogen-ionizing energies (1 Ryd $\le h\nu \le 7.354\times10^{6}$ 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} (Tielens and Hollenbach), between the limits of roughly 1000Å and 2400Å. This 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 0.1 to 0.38 Ryd
extinguish by 24, leakage = 0
```

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, and is defined as

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

where 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[2] 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 specific luminosity L_ν to be specified. The first number is the log of the specific 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, 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^{-5} 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.

This is a luminosity command.

[2] 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.

6.10. luminosity 38.3 [solar, range, linear]

The number is the log of the luminosity[3] emitted by the central object into 4π sr, (erg s^{-1})

$$L = 4p\, R_{star}^2 \int_{n_1}^{n_2} p\, F_n \, dn \qquad (10)$$

or the log of the total luminosity in solar units, if the **solar** keyword is specified. The number is interpreted as the luminosity 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.

If the optional keyword **solar** appears, the number is interpreted as the log of the *total* luminosity, relative to the luminosity of the sun (unless the **linear** keyword is also used, in which case the quantity will be the relative luminosity itself). 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 flux density νF_ν. The first number is the log of the surface flux density at the illuminated face of the cloud (erg s^{-1} cm^{-2}). It can be given at an arbitrary frequency and the default is 1 Ryd. The number is the log of 4π νJ_ν, where J_ν is the mean intensity of the incident continuum.

The optional second number is the frequency in Rydbergs where f_ν 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^{-5} 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.

[3] 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.

This is an intensity command.

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

This command specifies the specific luminosity νL_ν. The first number is the log of the specific luminosity radiated by the central object into 4π sr (erg s^{-1}). It can be expressed at an arbitrary frequency 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^{-5} 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 Φ(H), the surface flux of hydrogen-ionizing photons (cm^{-2} s^{-1}) striking the illuminated face of the cloud. It is defined as

$$\Phi(H) \equiv \frac{Q(H)}{4p\, r_o^2} \equiv \frac{R_{star}^2}{r_o^2} \int_{n_1}^{n_2} \frac{p\, F_n}{hn}\, dn \qquad (11)$$

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 in equation 11.

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

$$Q(H) = 4p\, R_{star}^2 \int_{n_1}^{n_2} \frac{p\, F_n}{hn}\, dn \qquad (12)$$

where the default value for ν_1 is 1 Ryd, and the default value for ν_2 is the high energy limit to the code, presently 7.354×10^{6} Ryd. 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 above[4].

This is a luminosity command.

[4] 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.

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 flux densities f_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 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. 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,

$$\frac{f_n(2\ \mathrm{keV})}{f_n(2500\ \mathrm{Å})} = \left(\frac{n_{2\ \mathrm{keV}}}{n_{2500\ \mathrm{Å}}}\right)^{a} = 403.3^{a} \quad . \qquad (13)$$

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 $\alpha_{ox} \sim -1.4$. If no X-Rays are present then $\alpha_{ox} = 0$.

There is an optional keyword on the **`ratio`** command that allows α_{ox} to be specified directly. If the keyword **`alphaox`** 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

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

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):

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

and by Wilkes et al (1994):

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

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 33), rather than this command.

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

7. CONTINUUM SHAPE

7.1. Overview

The continuum shape should be specified between an energy of 1.001×10^{-5} Ryd ($\lambda \approx 1$ cm) and 100 MeV ≈ 7.354×10^{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.

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

This will produce a multi-component continuum similar to that of an Active Galactic Nucleus (AGN). The “Big Bump” component is a rising power law with a high energy exponential cutoff, parameterized as the temperature of the bump. This temperature is the first argument on the command line. If it is less than or equal to 10 it is interpreted as the log of the temperature. The second parameter is the X-Ray to UV ratio α_{ox} (see the discussion of α_{ox} beginning on page 31). 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 slope of the low energy 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. This is also optional, and has the default α_x=-1. Optional parameters can be omitted from right to left.

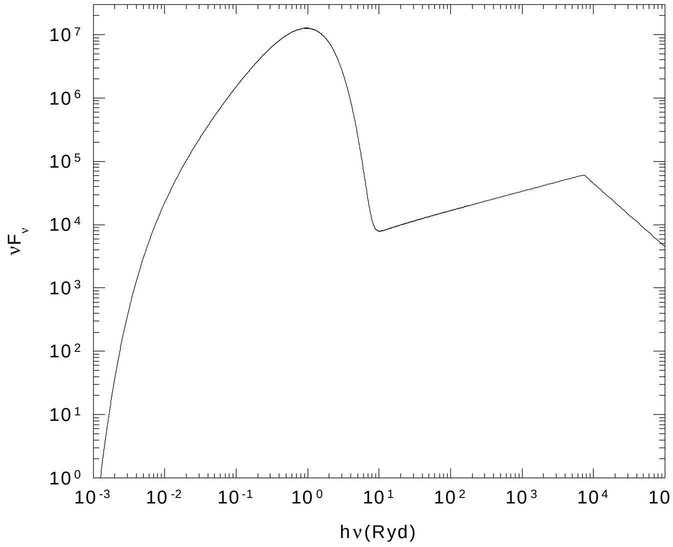


Figure 4 The continua produced by the AGN continuum command. The Big Bump peaks at 1 Ryd, while the X-Ray power law dominates high energies. The two are normalized by the second parameter, the value of α_{ox}. agncon

The full continuum is the sum of two components, as in equation **17**:

$$f_n = n^{a_{uv}} \exp\left(-hn / kT_{BB}\right)\exp\left(-kT_{IR} / hn\right) + an^{a_x} \quad . \tag{17}$$

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. This problem can be overcome by including the cosmic background, as described in the following **`background`** command.

The last term in equation **17** 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

$$4p\ J_n(912\text{Å}) = 4p \times 10^{-21}\left[1+\left(\frac{5}{1+z}\right)^4\right]^{-1} \quad f \quad \text{erg Hz}^{-1}\ \text{cm}^{-2}\ \text{s}^{-1} \qquad (18)$$

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 a factor of 10 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

$$T_{fireball} = 2.756(1+z)\ ^\circ\text{K} \qquad (19)$$

where the redshift dependence is from Peebles (1971) and the present temperature of the background is assumed to be T_{fireball} = 2.756 ± 0.016 K (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 37.

If a starting radius is not specified, then a value of 10^{25} cm will be assumed. Some objects, such as Lyα forest clouds, may be very large. Note that for the geometry to be plane parallel it is necessary to make sure that $\delta r / r$ « 1 (if $\delta r / r$ » 1 then 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 given by the number, the temperature in degrees Kelvin. 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 body t=6  .
```

7.4.1. blackbody luminosity options

The luminosity of the black body can also be specified with options that may occur on this command line. (These options were added by P.G. Martin.) 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 of the blackbody R_{star} (in cm) 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}):

$$T_u \equiv \left(\frac{u}{a}\right)^{1/4} \text{ K} \qquad (20)$$

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 $T_u \le 2.8(1+z)$ 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

$$W \equiv \frac{J_n}{B_n} \approx \frac{p R_{star}^2}{4 p r_o^2} \qquad (21)$$

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} « 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

$$f_n \propto n^{-0.2} \exp\left(-h n / kT\right) \quad . \qquad (22)$$

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 83) with the **punch transmitted continuum** command (page 107) to generate a continuum which can be read in with the **table read** command (page 43).

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

After the continuum has been fully generated and normalized to the correct intensity, this command will modify its shape by extinction due to photoelectric absorption by a cold neutral slab with column density given by the first argument (entered as a log). 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

$$f_n'\left(n \geq 1\mathrm{Ryd}\right) = f_n \left\{h + \left(1-h\right)\exp\left(6.22\times10^{-18}\, n_{Ryd}^{-2.43}\; N\left(H\right)\right)\right\} \qquad (23)$$

where N(H) is the total hydrogen column density, ν_{Ryd} is the frequency in Rydbergs, and η is the leakage.

The second (optional) number is the fractional leakage η through the absorber (see Ferland and Mushotzky 1982), which has a default value of 0. This number 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 third (optional) 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 extending 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.), *then* the continuum is 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.

A more physical way to do this would be to actually transmit a continuum through an absorbing slab, save that continuum with the **punch transmitted** command, then use this with the **table read** command (see the discussion on page 43).

7.7. fireball [redshift = 2000]

This command generates a blackbody radiation field in strict thermodynamic equilibrium (i.e., $T_{color} = T_u$, where u is the energy density). 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

$$T_{fireball} = 2.756\left(1+z\right)\,\mathrm{K} \qquad (24)$$

where the redshift dependence is from Peebles (1971) and the present temperature of the background is assumed to be $T_{fireball} = 2.756\pm0.016$ 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.

7.8. interpolate [n(Ryd) or log n(Hz)], log(f_n)

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 3000 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 dimension of the continuum arrays.) The first of each pair of points is *either* the energy in Rydbergs (either linear or as a log) *or* the log of the frequency (in Hertz). The second number of each pair is the log of the relative flux density per unit energy interval (f_ν) at that energy. 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. Any one of the **`intensity`** or **`luminosity`** commands then sets the luminosity of the continuum.

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

The **`interpolate`** command can be freely mixed with other continuum shape commands, and a total of up to 10 **`interpolate`** and **`table`** (see below) commands can be entered.[5] Note that **`table`** and **`interpolate`** are actually two forms of the same command. The total number of **`table`** and **`interpolate`** commands entered together cannot exceed 10, the current value of the variable ***limspc***, which occurs in parameter statements throughout the code.

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^{-5} 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

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

intensity, if this is possible. As a further note, it is important that the continuum be physically correct. For instance, stellar model atmospheres emit almost no X-Rays, while real OB stars *are* X-Ray sources (although neglecting X-Rays for these stars is generally a safe approximation). See page 44 for a further discussion.

CLOUDY will stop if more than 3000 frequency points are entered. The maximum number of frequency points allowed is set by the variable ***ncell*** that occurs in several parameter statements throughout the code. Presently ***ncell*** is 3000. If more points are needed then ***ncell*** should be increased.

7.9. laser, frequency = 3.5 Ryd

The intensity of the continuum will be very small, except within ±5% of the specified energy, where it will be very large.[6] The energy is specified in Rydbergs, and it is interpreted as a log if it is negative. This is a useful way to check on the computation of the photoionization rate integrals.

7.10. 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 optional low and high energy exponential cutoffs, parameterized by the cut-off frequencies $\nu_{high\ cut}$ and $\nu_{low\ cut}$, expressed in Rydbergs. The form of the continuum is

$$f_n = n^{+a} \exp\left(-n / n_{high \quad cut}\right) \exp\left(-n_{low \quad cut} / n\right) \quad . \qquad (25)$$

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 degrees Kelvin, rather than 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 microwave 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 low energies, probably producing 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.

[6] The luminosity normalization for this command was not exact before version 74.

It is *much* better to use the **interpolate** command (page 38), and enter physically reasonable low and high energy continua. There is a special version of the command, **table power law** (see page 42) for entering a well behaved power-law continuum.

7.11. table command

7.11.1. Overview

Any of several continuum shapes that are stored as a permanent part of the code in ***block data scalar*** can be entered with this command. This is a special version of the **interpolate** command, described 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.

*7.11.2. 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) and Ferland et al. (1992). The energy of the break can be adjusted with the **break** keyword. The break can be adjusted between the limits of 0.2 Rydberg and 1.001×10^{-5} 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, as linear Rydbergs if positive, and as the (linear) wavelength of the break in microns if a second keyword **microns** appears. If no number appears, but the keywords **no break** does, then a break at an energy of 1.001×10^{-5} 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). 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, or by using the **agn** command described on page 33.

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)

7.11.3. table akn120

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

The specific luminosity at 1320Å is νL_ν=1.84 ×10^{44} h^{-2} erg s^{-1}, where h≡H_o / 100 km s^{-1} mpc^{-1}, so, setting h = 0.75, the **akn120** continuum could be generated by the commands

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

7.11.4. 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.

7.11.5. 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 entered in the block data as the observed flux at Earth (erg cm^{-2} s^{-1} Hz^{-1}) and is given in Table 6.

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
```

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

7.11.6. 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 elsewhere.

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 zero (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 .
```

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

This option produces a power law continuum that is well behaved at both the high and low energy ends. The default condition (when no numbers occur on the input line) is for the form $f_\nu \propto \nu^\alpha$ with α=-1 for the spectral midrange, between 10 microns and 50 keV, and has slopes $\alpha=\nu^{5/2}$ at lower energy, and $\alpha=\nu^{-2}$ at higher energies. Table 8 summarizes the default continuum.

Three optional numbers may appear on the command line. The first number sets the slope of the mid-range spectral component (infrared to X-Ray) and has a default of -1. The next two numbers adjust the energy limits of the mid-range spectral component. The second number is the energy (in Rydbergs) of the infrared break. The default is 10 microns (0.009115 Ryd). If this second number is zero then the low energy limit to the continuum (1.001×10^{-5} Ryd) will be used. The number is interpreted as the log of the energy in Rydbergs if it is negative and linear otherwise. The third optional number is the energy (in Rydbergs) of the break in the X-Ray continuum. The default is 50 keV, and if it is zero then the high-energy limit of the continuum (7.354×10^{6} Ryd) is used. The number is interpreted as a log if the energy of the infrared break is entered as a log and linear otherwise. The numbers may be omitted from right to left.

Table 8
Power Law Continuum

ν(Ryd)	log (F_ν)	slope
1.00(-5)	-5.	+2.50
9.115(-3)	-0.5604	-1.00
3676.	-6.166	-2.
7.40(+6)	-12.77	-

7.11.8. table read [Fortran unit 8]

This command is used to read in the continuum predicted from previous calculations using CLOUDY. The first calculation is used to predict the continuum transmitted through a cloud. The **table read** command is then used in subsequent calculations, in which the transmitted continuum from the first calculation is to be included in the incident radiation field of the second calculation. The file containing the previously computed continuum is produced by including the **punch transmitted continuum** command in the initial calculation. This **punch** command is described in the sub-section starting on page 102. It produces a file containing the frequency (in Rydbergs) and the transmitted flux νf_ν. 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 **table read** command can be freely mixed with all of the other continuum shape commands. Only one **table read** command can be entered and the input file must have been produced by the current version of CLOUDY. The code will stop if these are not the case. The default is for the file containing the continuum points to be Fortran I/O unit 7 (i.e., named **fort.7** on many Unix machines, or assigned with the **assign name for.007** statement on VMS machines). The I/O unit number can be changed with the optional argument on the line.

When this command is used, punch output (using the **punch** command) must not also be produced on the same Fortran I/O unit number; the input file will be overwritten if this occurs. It is up to the user to make sure that this does not happen.

The first two lines of the input file contain header information, and are skipped. They should not be deleted.

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 io=7
```

Now use this continuum in a second calculation:

```
table read
luminosity 45
radius 18
hden 9
```

7.11.9. 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.

7.12. table stars overview

Several sets of emergent continua from stellar atmosphere calculations are available as built-in blocks within CLOUDY. These are accessed with the keyword **star**, followed by a sub-keyword (**Mihalas**, **Kurucz**, **Atlas**, **Rauch**, or **Werner**) to indicate which set of atmospheres is to be used.

Figure 5 compares predictions for five 50,000 K continua; a blackbody, 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 that there is an order of magnitude dispersion among the continua for energies approaching 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.

7.12.1. 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

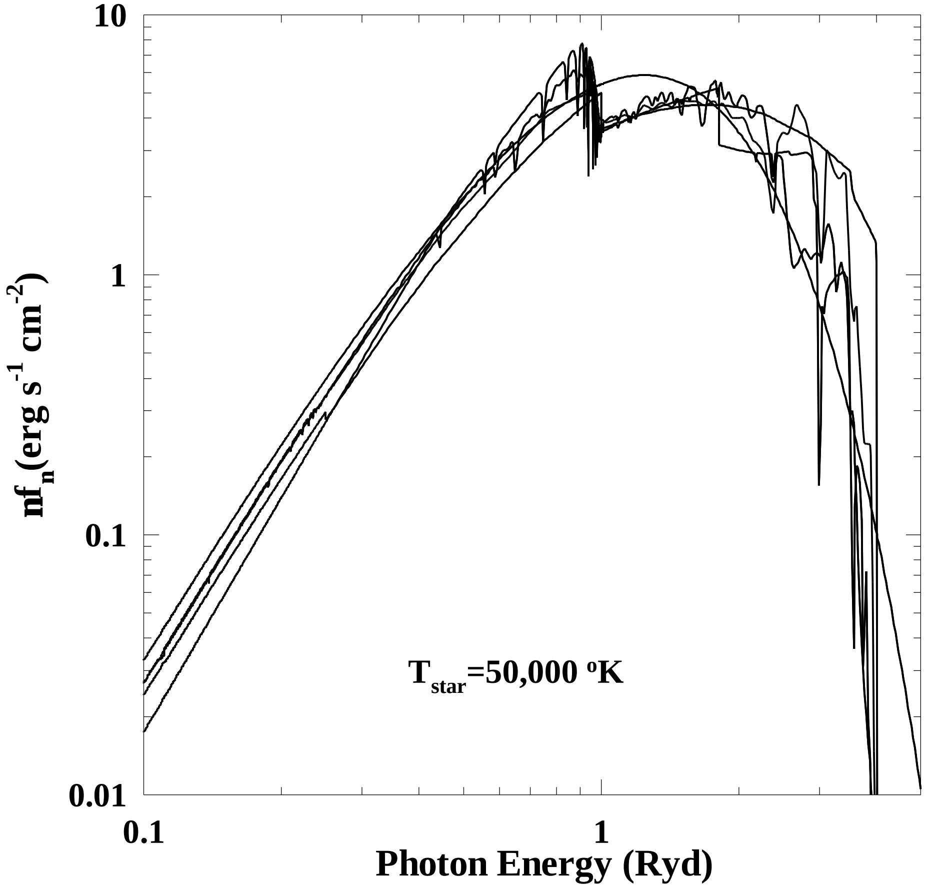


Figure 5 This figure shows the emergent continua predicted by five 50,000 K stars included with the code. The smoothest is the blackbody, and the Kurucz (1991) and Rauch (1997) atmospheres show the most structure. stars

bolometric luminosities which can be fitted by

$$\log\left(L_x\right) = 1.08\left(+0.06 / -0.22\right)\log\left(L_{bol}\right) - 9.38\left(+2.32 / -0.83\right) \quad . \qquad (26)$$

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

7.12.2. 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 as block datas. Both are static plane-parallel atmospheres. The parameters of the models included are summarized in Tables 9 and 10. The temperature and author of the calculation (**Kurucz** or **Mihalas**) must be specified; these can be in any order.

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)

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.

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   .
```

7.12.3. 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.

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

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 files obtained from the CLOUDY web site. This process is described on page 128 of this document. 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, as described on page 138 of this document.

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 would save a lot of time if you save the interpolated atmosphere with the **punch incident continuum** command, and then read this in for later calculations with the **table read** command, as shown below;

```
* this input stream is just to get the right continuum
hden 0   ;the first 4 lines get the code to run at all
ionization parameter -1
constant temper 4
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 43 of this document.

Table 11 Atlas (Kurucz 1991) Continua

Teff\log(g)	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

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

Kevin Volk has 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
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				*

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, so instead is stored as an ancillary file, which is generated by compiling files obtained from the CLOUDY web site. Compiling the star files is described on page 128 of this document. 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. The treatment of these files is entirely analogous to that of the **`table star atlas`** command (see page 45).

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

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

The Klaus Werner (Werner and Heber 1991) grid of non-LTE model planetary nebula nuclei atmospheres was incorporated into CLOUDY by Kevin Volk. This command asks the code to interpolate on this grid to generate an atmosphere with the specified temperature and gravity. The first number on the command line is the temperature, interpreted as a log if less than or equal to 10, and the second optional number is the log of the surface gravity. A gravity of log(g)=8 will be used if none is specified. Table 13 lists the temperatures and surface gravities stored within this set.

This grid is far too large to actually store within the code, so instead is stored as an ancillary file, which is generated by compiling files obtained from the CLOUDY web site. Compiling the star files is described on page 128 of this document. 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. The treatment of these files is entirely analogous to that of the **table star atlas** command (see page 45).

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

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			*	*

8. CHEMICAL COMPOSITION

8.1. Overview

The composition will be solar (defined in Table 14) 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).

Table 14 Solar Composition

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

Abundances are always specified by *number* relative to *hydrogen*, not by mass or silicon. Abundances are always by number relative to the total hydrogen density (the sum of atomic, ionic, and molecular). These are gas phase abundances, and do not include the fraction 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.

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, 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 15). In the following example the first nitrogen 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. It is important to confirm that 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

An arbitrary abundance set can be specified by entering all abundances with the **abundances** command. The elements must be in exactly the same order as indicated in Table 14 unless the order is altered with the **elements read** command described on page 55. Abundances for all active elements must be specified, but elements can be turned off with the **elements off** command described on page 55. 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.

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 55.

The best way to enter abundances is as the log of the abundance by number relative to hydrogen, referred to as absolute abundances:

```
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 decides which style was entered by checking the sign of each number. 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 below, is useful for changing abundances of all elements heavier than helium.

8.3.3. Stored abundance sets

Table 15 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 line, 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 their present values are given in Table 15, by number relative to hydrogen.

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 obtain this abundance set. Abundances of some rare species were taken from the ISM mix of Savage and Sembach (1996). 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.

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

Table 15 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)	-	5.01(-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

application of this data to old planetary nebulae is dicey at best – evidence summarized by Clegg and Harrington et al. (1989) suggest that these 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 grains are the default interstellar medium grains.

Some mixtures have certain elements, especially Si, Ca, Al, Mg, and Fe in H II regions and planetary nebulae, with gas-phase compositions reduced by depletion onto grains. Specifying **h ii region**, **_ISM**, or **planetary nebula** will invoke grains and the gas phase mixtures given in Table 15. 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** command. If grains are set by both the **grains** and **abundance** commands then the parameters set by the **grains** command are used.

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.[7] This is, of course, not self-consistent.

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 an evolution model 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 $M = 2.5M_{\odot}$ instead of $\sim 0.1M_{\odot}$ 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. The "name" must be at least the first four characters of the name of the element as spelled in Table 14. One of the keywords **scale**, **abundance**, or **_off** must appear on the line.

8.5.1. element name scale

If the keyword **scale** appears then the number on the line is interpreted as a scale factor multiplying the abundance of the element. The scale factor will be interpreted as a linear scale factor if the number is positive, 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.

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

8.5.2. 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 always interpreted as the log of the abundance.

8.5.3. element name off

If **_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. This may save some time. The abundance must still be specified when the **abundances** command is used unless the element is excluded with the **elements read** command described next.

This command sets the logical variable ***lgElmtOn(n)*** to false, where n is the atomic number of the element.

8.5.4. elements read

Normally CLOUDY expects the numbers entered with the **abundances** command to occur in exactly the same order as the atomic numbers of the elements, beginning with helium, and for all elements to be specified. 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 begin to read in a list elements. The list ends with a line beginning with the keyword **end**.

On all **abundances** commands following this, the ordered set of numbers entered on the command will be interpreted as the abundances of these elements. If an element does not occur in the list its value is unchanged by an **abundances** command. (An element is turned off with the **element name off** command, described on page 55.)

The following example shows an input stream that will cause 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.

```
elements read
helium
carbon
nitrogen
oxygen
neon
sodium
magnesium
aluminum
silicon
sulphur
argon
calcium
iron
nickel
end of elements
```

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.

8.5.5. element name table

If the keyword **table** appears 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
```

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 7. 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, as set by the code variable ***limTabD***. 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, this will be the inner radius of the cloud. If the table gives the composition as a function of depth, then this 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. 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.

The **fluctuations** command is more fully described in the description of the density version, on page 61.

8.7. grains

See page 72.

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,[8] while

```
metals -10
```

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

The metal depletion factor is stored as the variable ***dmetal***, in the common block of the same name. The grains depletion factor is stored as the variable ***grmetl***. Both have default values of unity.

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. There is an optional keyword on the **metals** command, **grains**, which causes the grain abundance to also be scaled by the factor on the line. The basic assumption here 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 the default, and a metallicity of half 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  .
```

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 16 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).

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

This command can be combined with others that specify abundances, and the **grains** command, to specify a mixture of grains and depleted gas. Specifying grains by themselves (with the **grains** command) does not deplete the assumed gas-phase abundances, which is, of course, not self-consistent. (The code will complain if you try 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 result in any 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.

Note that Tables 16, 15, and 14 are not self-consistent since they come from different sources. Physically, there is a growing suspecion 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.

Table 16 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

9. DENSITY LAWS

9.1. Overview

Commands that specify the run of hydrogen density with radius are described in this section. The default condition is constant density, with the total hydrogen density (atomic, ionic, and molecular, given by the command **hden**) held constant. Power law and sinusoidal density distributions, as well as constant gas pressure, constant total (gas and radiation) pressure models, or an arbitrary density law, can also be computed.

9.2. constant density, pressure, gas pressure

This 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

$$n\left(H\right) = n\left(H^{o}\right) + n\left(H^{+}\right) + n\left(H^{-}\right) + 2n\left(H_{2}\right) + 2n\left(H_{2}^{+}\right) + 3n\left(H_{3}^{+}\right) \qquad (27)$$

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, which has as an option a power law dependence on radius.

9.2.2. constant gas pressure [index =-1]

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

$$P_{gas} = n_{tot} k T_{e} \qquad (28)$$

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;

$$P_{gas}\left(r\right) = P_{o}\left(\frac{r}{r_{o}}\right)^{a} \qquad (29)$$

where P_o is the pressure at the inner radius.

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]

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;

$$P_{tot}(r) = P_{tot}(r_o) + \int a_{rad}\ r\,dr = P_{gas} + P_{lines} + P_{continuum} \qquad (30)$$

where a_{rad} is the radiative acceleration 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 integral, referred to as $P_{continuum}$; this can be turned off by specifying the **no continuum** option on the command line).

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 $rv^2/2$ and $B^2/8p$ to equation 30.

CLOUDY will stop if the internal line radiation pressure builds up to more than half of the total pressure, since such clouds would be unstable unless they are self-gravitating (Elitzur and Ferland 1986). It is necessary to do at least a second iteration when radiation pressure is important since the total line optical depths must be known to compute line widths, 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.

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 subroutine provided by the user, and a second form which 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, the last entered command will be honored.

9.3.1. dlaw p1, p2, p3 ...

This is the default form of the command, and is passes the parameters on the command line to a user provided function. There are up to ten parameters, which are stored as a vector in the common block ***dlaw***. 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 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 single precision variable. The code provided in the function must use the ten or fewer parameters in

common/dlaw/ dlaw(10) to compute the density at the current position. Both the vector ***dlaw*** and arguments to the function ***fabden*** are double precision, although the function value is single precision.

The following is an example of a function.

```
     real function fabden( radius , depth )
     implicit none
     common/dlaw/ dlaw(10)
     double precision radius , dlaw , depth
*
*    vector dlaw and radius are double precision,
*    density (fabden) is not
     fabden = dlaw(1) * depth + dlaw(2)
*
     end
```

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 per line, with the log of the radius or depth (in cm) 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, controlled by the variable ***limTabD*** which appears in parameter statements throughout the code. Linear interpolation in log-log space is done. The following is an example.

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

Be sure that the first and last radii or depths extend beyond the computed geometry - this option can only be used for interpolation, and the code will stop if extrapolation is necessary.

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 omitted, the calculation will begin at the maximum value. If the phase is set to π the calculation will start at the minimum density.

A variable ***flong*** is set to the value 2π / P in routine ***flcsub*** when the calculation is initiated. The density is then scaled (in routine ***presur***) using the equation

$$n(r) = \left(\frac{n_{\max} - n_{\min}}{2} \right) \times \cos\left(depth \times flong + j \right) + \left(\frac{n_{\max} + n_{\min}}{2} \right) \qquad (31)$$

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 very long execution times. The code must spatially resolve the density fluctuations to obtain a true simulation. To do this, the zone thickness (set in routine ***NextDR***) is not allowed to exceed ~0.05 of the period, so that each cycle is divided into roughly 20 zones. 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 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

$$n(r) = n_o \left(\frac{R_{scale\, depth}}{R_{scale\, depth} - \Delta r} \right)^{a} = n_o \left(1 - \frac{\Delta r}{R_{scale\, depth}} \right)^{-a} \qquad (32)$$

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 radii are specified by the optional second parameter, which must be entered as a log, in cm. The optional third argument is the index α, which has the default[9] α = 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

$$n(H) = n\left(H^o\right) + n\left(H^+\right) + n\left(H^-\right) + 2n\left(H_2\right) + 2n\left(H_2^+\right) + 3n\left(H_3^+\right) \quad \mathrm{cm}^{-3} \quad . \qquad (33)$$

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 $T\le10^4$ K) the ambiguity is much less than 1%.

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

Several options are available to specify optional power-law dependencies on depth variables. These are described further 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.,

$$n(r) = n_o(r_o)\left(\frac{r}{r_o}\right)^{\alpha} \quad \mathrm{cm}^{-3} \quad . \tag{34}$$

In this example n_o is 10^9 cm^{-3} and is the density at the inner radius or illuminated face. 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,

$$Q_{crit}(H) = \alpha_B(T_e) n_o^2 \, 4\pi \, r_o^2 \quad \mathrm{s}^{-1} \tag{35}$$

where $\alpha_B(T_e)$ is the hydrogen case B recombination coefficient and n_o and r_o are the inner density and radius respectively. A hydrogen ionization front will not be present and the model will extend to infinite radius when $Q(H) \geq Q_{crit}(H)$. Generally, an 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 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 $\alpha \leq -2$, 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`** appears:

```
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

$$n(r) = n_o(r_o)\left(1 + \frac{\Delta r}{R_{scale}}\right)^{\alpha} \quad \mathrm{cm}^{-3} \tag{36}$$

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`** appears;

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

Here the density is given by

$$n(r) = n_o(r_o)\left(1 + \frac{N(H)}{N(H)_{scale}}\right)^a \quad \mathrm{cm}^{-3} \qquad (37$$

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 this 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. The covering factor affects the transport of diffuse fields and has a second order effect on the predicted emission line spectrum. Finally, the cloud can be either static or expanding.

10.2. age 44 years [_off]

The **`age`** command allows the code to check 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. 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.

The covering factor affects both the luminosity 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 directly affect the line intensities, (the emission per unit area) if these are predicted rather than luminosity. It does have a second order effect 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*** in the common block ***cover***. A second covering factor, ***covrt***, affects the transfer of lines and continua. Both covering factors are set by the number on this command line.

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). Both the **`covering factor`** and **`sphere`** commands should not be used in the same input stream. If they are then only the second will be honored.

10.4. 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 67. 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

$$dV = 4p\, r_o^2 \left(\frac{r}{r_o} \right) \left(\frac{\min\left(r, h_{cyl}\right)}{r_o} \right) f\left(r\right)\, dr \qquad (38)$$

where r_o is the inner radius, h_{cyl} is the cylinder half-height, and f (r) is the filling factor. The half-height h_{cyl} is stored as ***cylind***, the sole element of the common block with the same name. The default value is $h_{cyl} = 10^{35}$ cm.

Changing the emissivity as described by equation 38 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.

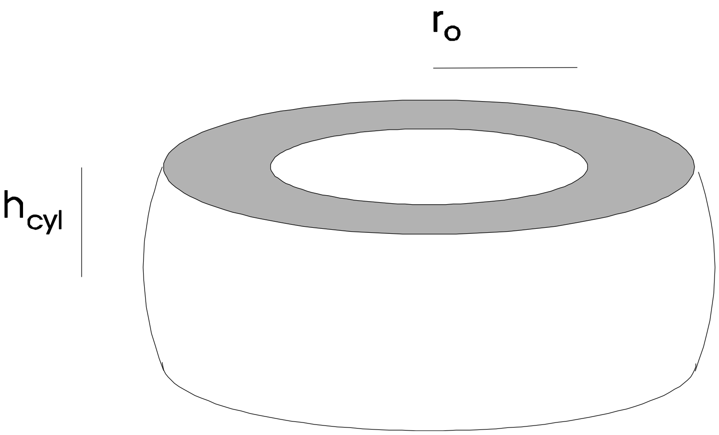


Figure 6 This figure shows the geometry assumed when the cylinder command is used. cylin

10.5. 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.,

$$f\left(r\right) = f\left(r_o\right) \left(\frac{r}{r_o} \right)^{a} \qquad (39)$$

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 cloud,

$$dI = 4p\ j\ f\left(r\right) dV \frac{\Omega}{4p} \qquad (40)$$

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

$$dt = a_{l,u}\left(n_l - n_u \frac{g_l}{g_u}\right) f\left(r\right) dr \qquad (41)$$

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

10.6. illuminate 45 deg [radians]

This will simulate a plane parallel slab illuminated by a beam θ away from the normal. The default is θ = 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 τ_n / cos(θ) where τ_n is the normal optical depth of the zone. Line and diffuse continua optical depths are not affected. The number 1 / cos(θ) is stored as the variable ***AngleIllum***.

10.7. 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 numbers themselves and not their 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 125 describes a problem that can occur if the second parameter is used with the **vary** option.

10.8. sphere [expanding ; static; covering factor =.4]

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, and 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.

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 other side. This is 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 the escape probability at the illuminated face of the cloud is small for optically thick lines. 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. The optical depths for all lines are determined self-consistently on second and further iterations. Lyα radiation pressure in the H^+ region will probably be significant if **sphere static** is set.

The optional number on the line is the covering factor for the emission-line region. It is interpreted as the log of the covering factor if it is less than or equal to zero, and the covering factor itself if positive. The covering factor can also be set using the **covering factor** command. The effects of the covering factor are described further on page 65. The **covering factor** and **sphere** commands should not both be used in the same input stream. If they are then only the last will be honored.

When a covering factor of unity is set the specific effects of **sphere** are; a) the total continuous optical depths are assumed to be twice the computed optical depths, and the initial optical depth is half the total, b) all diffuse reemission (bremsstrahlung, free-bound, etc.) is counted in the outward beam rather than only half, c) scattering opacities are not considered in the attenuation of the incident radiation field, d) when **static** is set, the optical depth in Lyα in the inner direction is set to 10^5 on the first iteration; otherwise it is 10^{-20} and all lines are given twice their computed depth, and finally e) include ionization by lines produced in the other side of the nebula. 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.

10.9. 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

$$t_{l,u}(R) = a_{l,u} \min(r,\Delta r)\left(n_l - n_u \frac{g_l}{g_u}\right)\left(\frac{n_{th}}{\max(n_{th}, n_{\exp})}\right) \tag{42}$$

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 the cloud when the radius is very large and the expansion velocity 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 $r(r)r^2 v(r)$ 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.

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. Further details are presented in a section in Part II.

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 (see Ferland and Netzer 1979; Weisheit et al 1981; Kwan and Krolik 1981; Canfield and Puetter 1980). In other cases grains are present and all lines can be absorbed by background opacity. The present version of CLOUDY treats line radiative transfer in the escape probability formalism (Hummer 1968; Hummer and Kunasz 1980; Elitzur 1982; Netzer, Elitzur, and Ferland 1985). 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 transitions, so that the proper escape probabilities can be computed, when line transfer is important. The program also has an **`iterate to convergence`** command to continue iterating until the optical depths are well defined. If the Lyα or Hα optical depths change by more than ~20% on the last iteration then a warning that the model has not converged will be printed at the end of the last iteration.

Line radiation pressure cannot be computed accurately until the total line optical depths are known, so this quantity is meaningful only after at least one 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.

11.2. case b [tau ly alpha = 9; Hummer and Storey]

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 Lyα to 10^9, so that even a one-zone model will be close to case B. The optional number is $\log(\tau_{Ly\,\alpha})$, so it is possible to change this assumption. One-sided escape probabilities are used in this case, so the total escape probability is simply that for the inward direction. The **`caseb`** command also suppresses optical depths in excited states, in keeping with the case B approximation.

Normally, the treatment of the hydrogen and helium atoms 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 collisions off** command is given. Collisions between the 2s and 2p states are always included unless the **no 2s2p** command is given.

11.3. diffuse fields [outward, ots]

This command specifies which method is to be used to transfer the diffuse fields. 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[10] 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 has the values either "OTS", "OU1", or "OU2", etc, depending on which approximation is used. These approximations are described in Part II of this document.

11.4. double optical depths

This command simulates a geometry in which ionizing radiation strikes the plane parallel cloud from both sides, such as a Lyα forest cloud. At the end of the iteration, the total line and continuum optical depths are set to twice the computed optical depth. 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 Lyα 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.

11.5. escape __K2; incomplete; simple; [emit; destroy]

This command specifies how line escape probabilities are to be handled. One of the keywords **__k2, simple,** or **incomplete** must appear. If no other keywords appear then both the emission and destruction probabilities are set to either of the specified cases. If either keyword **emit** or **destroy** appear, then only the emission or destruction probability is changed.

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

The keyword **simple** sets all escape probabilities to a very simple form, $1/(1+\tau)$. This is intended for debugging the code.

11.6. FeII [Netzer; Verner]

N.B. – there is no space between the element symbol and the spectroscopic designation. **Fe_II** will not work.

11.6.1. feii Netzer

This command determines which model FeII atom is used. The default is the **Netzer** atom, the simplified scheme outlined by Wills, Netzer, and Wills (1985). This is very fast.

11.6.2. feii Verner [options]

The **Verner** option employs the large FeII atom developed by Katya and Dima Verner, which is far more accurate, but also much larger, and far, far slower. The Verner model atom is part of Katya Verner's Ph.D. thesis, and is not now included in the distribution version of CLOUDY since development is not complete. There are additional parameters when the **Verner** option is used. These set options for their model FeII atom.

FeII Verner simulate This tricks the code into computing a model with no FeII cooling. The FeII data does not need to be linked in if this command is entered.

FeII Verner range 0.1 to 0.4 Ryd The punch output file created with the **punch Verner** command can be very large. This command allows only the fraction of the total lines that lie within an energy range of interest to be punched. This sets the energy range for those punched. There must be two numbers, the lowest and highest energies (in Rydbergs) of the lines to be punched. If either number is negative then both are interpreted as logs. The default is for all lines to be punched.

FeII Verner threshold 0.001 This is the intensity of the weakest FeII line to be punched, relative to the normalization line (usually Hβ). The default is for all lines to be punched. If the number is negative it is interpreted as a log.

11.7. 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 **grains** command takes precedence over the default grains set with the **abundances** command. The **grains** command has two optional arguments, the abundance of the grains, and a pointer to the type of grain to be used. These are described below.

The treatment of grains was developed in close collaboration with P.G. Martin. Details are provided in a section of Part II, and in Baldwin et al. (1991). Ten populations of grains, summarized in Table 17, 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 of the grain (cm^{-2}) per hydrogen nucleon.

Table 17 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}

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, Lyα, 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

11.7.1. 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.

11.7.2. grain types

It is possible to turn on each of the species listed in Table 17 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 17). The 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.

11.7.3. 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.

11.7.4. 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 17 are used in this case. This is the grain type used in Baldwin et al. (1991).

11.7.5. 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.

11.7.6. 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 classical heating – cooling balance. If the keyword **`qheat`** appears on the command line then quantum heating will be used instead (see page 80).

11.7.7. 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 for the AGB case. 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 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-gar 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.

11.7.8. grain abundance 0 id=11 grey1.opac

If the grain species is specified by a number, and this number is greater than the number of species listed in Table 18, 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. This species must be between 11 and 20. The grain abundance must be the first number on the line.

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. The format must 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.

11.7.9. examples

The following are some examples;

```
*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
```

11.7.10. 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 within the distributed source that can be modified by the user to produce any desired behavior. The function should return unity if grains are to have their expected abundance, and the user can modify it to return any scale factor. This scale factor should depend on the local physical conditions for any of the grain species in the calculation.

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. Note that 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 55).

11.7.11. line intensities with grains.

For a closed geometry the abundances of, 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. This includes absorption and back-scattering by grains beyond the shielded face of the cloud, assuming that large amounts of neutral material extends 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 de-redden the observed spectrum to obtain an intrinsic spectrum, and to then compare this intrinsic spectrum with that computed by the code.

11.8. helium [options]

11.8.1. helium collisions off

This turns off collisional ionization and n-changing collisions (but not 2s-2p) within the helium singlets and helium ion.

11.8.2. helium radiation pressure on

This turns on radiation pressure due to trapped He II Lyα. This line undergoes Bowen fluorescence and strong destruction by the background continuum, and as a result its line radiation pressure is very difficult to compute. Normally it is neglected. The approximations used when the option is turned on are correct when the gas is highly ionized and oxygen is depleted (as in a Lyα forest cloud).

11.9. hydrogen [options]

This allows some details of the treatment of the hydrogen atom to be changed. Unlike the vast majority of the commands, the entire keyword **hydrogen** must appear for this command to be recognized.

11.9.1. hydrogen collisions off

Collisions between principal quantum levels of the hydrogen atom and collisional ionization can be turned off with this command. All collisions *except* 2s-2p are turned off by this command. (These are turned off for H and He with the **`no 2s2p`** command.)

11.9.2. hydrogen 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 > 10^{23} cm^{-2}).

11.9.3. hydrogen levels 15

The model hydrogen atom can extend up to any principle quantum number between 4 and 50, the current value of ***lmhlvl***. That variable appears in parameter statements throughout the code. The number of levels to be used in the current calculation is altered with this command, and is saved as the variable ***nhlvl***. 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, but at the expense of longer execution times.

If no number appears on the **`hydrogen levels`** command, but the keyword **`maximum`** or **`limit`** does, then the maximum number of levels, the value of ***lmhlvl***, will be used. This provides a version-independent method of insuring that the code uses the largest possible number of levels.

11.9.4. hydrogen lowest temp 200

Normally the hydrogen (and helium) 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 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 1000 K. There is a single argument, and the number is interpreted as the log of the temperature if it is less than or equal to 10. 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.

11.9.5. hydrogen matrix [linpack, matin1, veclib]

This command specifies which matrix inversion routine to use to solve for the populations of the multi-level hydrogen atom. The three keywords recognized by this command are **`matin1, veclib`**, and **`linpack`**. The default is **`matin1`**. Code exists to call one of the routines from the ***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***.

11.9.6. hydrogen redistribution

Lyα transfer can be treated with either complete or incomplete redistribution (see, for example, Hummer 1962); the default is incomplete redistribution, and can be changed by entering

```
hydrogen redistribution complete
hydrogen redistribution incomplete  .
```

There is at present a fundamental uncertainty in the computation of the line radiation pressure for transitions such as Lyα. 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 Lyα (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 Lyα, 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).

11.9.7. hydrogen 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. One can choose to either add on the difference in recombination coefficients (if the keyword **_add** appears) or scale them (if **scale** appears). This extra part can either be added on (if the keyword **_add** appears) or a scale factor (if **scale** appears). The defaults are **hydrogen topoff 11 scale**, and **hydrogen levels 15**.

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 within these disturbed levels.

11.10. iterate [2 times]

This command specifies the number of iterations to be performed. The default is one 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.

11.10.1. 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 have been computed. The optional first number on the line is the maximum number of iterations to perform, and the default is 10. It is not possible to specify more than ***ItrDim*** (currently 20) iterations. The second optional number is the convergence criterion. Normally, it is that the hydrogen and helium line 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 changes in the Hα, H, HeI and HeII Lyα, and the HeII 3-2 and 4-3 optical depths are less than the second number, unless the transitions are optically thin, in which case only a second iteration is performed.

11.10.2. 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 is very near a prominent ionization front. In this case very small changes in the physical conditions results in large changes in the optical depths. The code will not have convergence problems if an optical depth is used as a stopping criterion instead.

11.10.3. Code variables

The upper limit to the number of iterations that can possibly be performed is set by the value of ***ItrDim*** (currently 20) by parameter statements within the code. This is used to set the dimension of several vectors. Increase ***ItrDim*** everywhere it appears in the code to increase this limit. The actual limit to the number of iterations set by the user is stored as the code variable ***itermx***, the second variable in the common block ***iter***. Because of the position of the check of the relative values of ***itermx*** and ***iter*** within the code, ***itermx*** is one less than the limit, so the default is 0. The first variable of this common block is ***iter***, the counter for the current iteration – ***iter*** is equal to 1 during the first iteration. The relative fractional change in the optical depth scale is stored as the variable ***autocv***. The logical flag ***lgAutoIt*** is set to true when **iterate to convergence** is specified. The test for convergence is performed in routine ***update***.

11.10.4. 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 iteration 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 no iterations. If any one of the following commands is entered:

```
iterate
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 it will be the number of calculations of the structure.

11.11. 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 $\left(1+0.5t_{scat}\right)^{-1}$ rather than $\exp(-t)$ (Schuster 1905).

Scattering should be neglected in a spherical geometry with gas fully covering the source of ionizing radiation, since 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 $t_{scat} \le 1$ 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., $hn << mc^2$) 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 nebulae 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.

11.12. qheat [options]

The command turns on Kevin Volk's grain quantum heating treatment. This will only affect heating and cooling of the two PAH species (see page 74). Options on the command line specify how this physics is treated. Quantum heating can also be enabled with the keyword **qheat** that can appear on the **grains pah** command, but it is not possible to set these options in that case.

Guhathakurta and Draine (1989) describe the formalism for the calculation. The temperature probabilities are calculated for a finite series of temperature bins. This returns a T^4 weighted average in place of an equilibrium temperature. For anything except a gray body grain, this temperature will be different from the equilibrium temperature, because the emission does not actually go as T^4.

There are two possible problems due to numerical discreteness. If the maximum temperature of the true distribution is greater than that allowed in these bins then the computed temperature will be too low. This greatest temperature is set by the second parameter on this command line, as described below. If the radiation field is too intense then the temperature distribution will have a narrow peak but the mean temperature will be too low.

In the zone printout, an asterisk will appear to the right of any grain species in which quantum heating is enabled.

This command does not turn on grain species – this must be done separately with the **grains** command (page 72).

The first number is the number of temperature bins. This can be between 10 and **nqgrid**. **nqgrid** is currently 200. If this number is not specified then it is set to the default of 50. The execution time goes as the square of the number of temperature grid points.

The second parameter is the maximum temperature for the distribution. This must lie between 50 K and 5000 K and it has a default of 3000 K.

Two keywords, **all** and **print**, are recognized. The first, **all**, will turn on quantum heating for all grains specified. Physically this only makes sense for the two PAH species or small radius (a < 0.01 μm) grains. If **all** is not specified then only the two PAH species are automatically chosen. The **print** keyword will generate a file called **qheat.out**, which contains the distribution of temperatures for the quantum-heated grains, for each zone printed with the **print every** command.

11.13. turbulence = 100 km/sec [_log]

The input number is the turbulent velocity (assumed to be microturbulence) expressed in kilometers per second. This velocity field affects the line width and optical depth scale through the Doppler width $v = \sqrt{v_{th}^2 + v_{turb}^2}$, where $v_{th} = \sqrt{2kT/m}$ 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 component to the total pressure, of

$$P_{turb}(r_o) = \frac{1}{2} r v_{turb}^2 = 5.8\times10^6 \left(\frac{n_{tot}}{10^5\,\mathrm{cm}^{-3}}\right)\left(\frac{v_{turb}}{1\mathrm{kms}^{-1}}\right)^2 \mathrm{cm}^{-3}\mathrm{K} \qquad (43)$$

where n_{tot} is the total density and v_{turb} is the turbulent velocity. Turbulent pressure is not now included in the pressure law since it would either be negligible or totally dominate the pressure. In the latter case the density could not be determined. 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.

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]

It is possible to add an extra source of cooling (due to some unspecified physical process) with this command. 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

$$\Lambda = 10^{c_1} \times \left(\frac{T_e}{10^4 \ {}^\circ\mathrm{K}} \right)^{c_2} \quad \mathrm{erg\ cm^{-3}\ s^{-1}} \qquad (44)$$

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 extra cooling is stored as the code variable ***cextra*** and the optional exponential is ***cextpw***. These variables are used to generate a total extra cooling, in routine ***coolr***. Other expressions for extra cooling can be obtained by modifying the code there.

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

A constant 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 included, so this option can produce clouds in coronal or collisional equilibrium. For technical reasons, the photon array must be defined for all energies that contribute to photoionization of all stages of ionization with significant abundances. For instance, to do models of the solar corona it is necessary to include both the cool blackbody from the solar photosphere. This is important for excited state induced recombination, photoionization, and Compton cooling. It is also necessary to add a weak ~10^8 K bremsstrahlung continuum to define the photon array at all possible coronal energies (see the **`coronal equilibrium`** command, described below).

WARNING! When this command is used, it is also necessary to specify stopping criteria of some kind. Many 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 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`** or **`stop efrac`** commands to stop the calculation when the hydrogen ionization front is reached, or **`stop zone`** to stop the calculation at a particular zone number.

When constant temperature is specified the logical variable ***lgTSetOn*** in common block tseton is set true.

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.

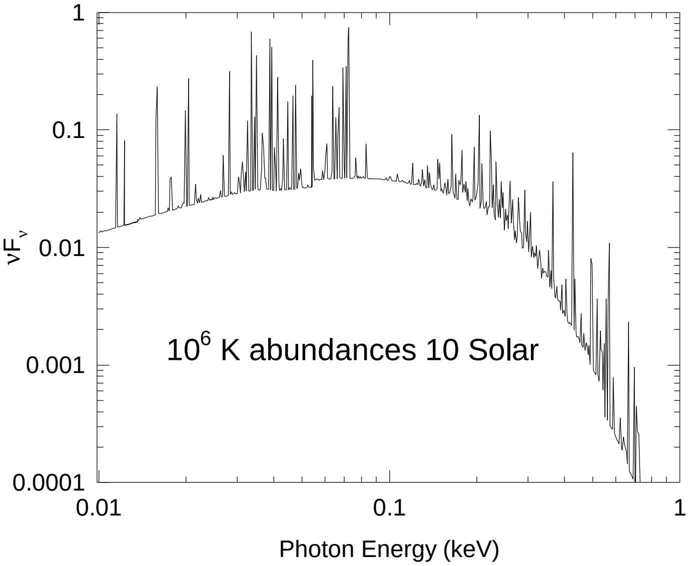


Figure 7 This figure shows the soft X-Ray emission from a simple model of the hot phase of the ISM. The input script brems.in was assumed.. coronal

It is necessary[11] 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`** in Part III of this document.

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

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

number is a power-law index α, describing the variation of the cosmic ray density with radius, i.e.,

$$n(cr,r) = n(cr,r_o)\left(\frac{r}{r_o}\right)^a \quad \mathrm{cm}^{-3} \quad . \qquad (45)$$

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 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) = 2×10^{-9} cm^{-3} will be used. This density will produce a neutral hydrogen ionization rate of ~2×10^{-17} s^{-1}, the value quoted by Tielens and Hollenbach (1985a) for the galactic cosmic ray ionization rate. *This value is quite uncertain.* 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 [no map]

A temperature failure occurs when the heating-cooling balance is not within a certain tolerance, set by the **tolerance** command, after 20 tries. Normally CLOUDY will punt[12] after an excessive number of temperature failures (presently 20) occur. The limit to the number of failures is 20, which is stored as the variable ***limfal*** in the common block of the same name. 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. The **no map** option will turn off this map. 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.

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

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

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.)

12.8. hextra -14 [scale r=18]

This command turns on extra heating due to some unspecified energy source. The first number is the log of the volume heating rate (erg cm^{-3} s^{-1}). The second number is the log of the scale radius r$_{scale}$, such that the extra heating rate varies as $\exp\left(-r_{scale} / \left(r - r_o\right)\right)$, where r$_o$ is the inner radius. The default, when r$_{scale}$ is not specified, is constant extra heating.

The heating set with this command is stored as the code variable ***turhet*** and the scale radius is stored as ***turrad***. The total extra heating due to all sources is stored as the variable ***hxtot*** and is evaluated in routine ***highen***. Other expressions for extra heating can be obtained by changing the source in ***highen***. Look for the statement where ***hxtot*** is incremented by an expression containing ***turhet***.

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

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

$$\Lambda_{cyclotron} = n_e \frac{B^2}{8p} \frac{4}{3} s_{\mathrm{Thom}} c \left(\frac{v_e}{c} \right)^2 = 4.5433 \times 10^{-25} n_e B^2 T_e \quad \mathrm{erg\ cm^{-3}\ s^{-1}} \qquad (46)$$

where σ_T is the Thomson cross-section and

$$v_e = \left(\frac{8kT_e}{p\, m_e} \right)^{1/2} = 6.2124 \times 10^5\, T_e^{1/2}\ \mathrm{cm\ s^{-1}} \qquad (47)$$

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.

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 term of order $B^2 / 8p$, 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

$$B^2 / 8p\, k = B^2\, 2.882\times10^{14} \approx nT\,\mathrm{cm}^{-3} K \ . \qquad (48)$$

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

$$rv^2 / 2\,k = 60.14\, nv_{kms}^2 \approx nT\,\mathrm{cm}^{-3} K \ . \qquad (49)$$

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

$$v_A = \frac{B}{(4pr)^{1/2}} \approx 2.19\times10^6 B\, n^{-1/2}\ \mathrm{km\ s}^{-1} \ . \qquad (50)$$

The magnetic 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 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 generally 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 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 is controlled by the variable ***nmaps***, and can be reset with the **`set nmaps`** command, discussed on page 138.

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

The code is left in a disturbed state after a map is computed, so it stops when complete.

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 98.

12.14. print heating

See page 99.

12.15. 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.

12.16. tolerance 0.001

The equilibrium electron temperature is set by balancing the heating and cooling rates. 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.

The tolerance is given by the variable ***toler***, which is the sole variable in the common block ***toler***.

13. STOPPING CRITERIA

13.1. Overview

In some nebulae ionized by starlight, such as certain planetary nebulae or H II regions, the outer limit of the emitting gas is well defined by a hydrogen ionization front (the nebula is said to be radiation bounded), and setting an outer limit is not necessary. In these cases 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. If lines with very low ionization and excitation potentials (i.e., the [C II] or [O I] far infrared lines) are of interest then it is necessary to lower this stopping temperature with the **`stop temperature`** command.

In other circumstances, particularly X-Ray ionized nebulae, the gas is optically thin to hard radiation and an outer limit must be specified. In other situations, optically thin models, or ones in which only part of the hydrogen Strömgren sphere is present, must be computed (in this case the nebula is said to be matter bounded). In these cases stopping criteria must be specified. 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, then 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 usually 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 ask that you send me the input stream and version number.

It is a good idea to check whether, if the model were made thicker or thinner, the line intensities would change. It is safe to assume that a line no longer depends 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 67. 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

$$N(H) = \int \left\{ n\left(H^o\right) + n\left(H^+\right) + n\left(H^-\right) + 2n\left(H_2\right) + 2n\left(H_2^+\right) + 3n\left(H_3^+\right) \right\} f(r)\, dr \qquad (51)$$

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

$$N\left(H^o\right) = \int n\left(H^o\right) f(r)\, dr \quad . \qquad (52)$$

13.4.3. stop ionized column density 23

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

$$N\left(H^+\right) = \int n\left(H^+\right) f(r)\, dr \quad . \qquad (53)$$

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, and 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 all scattering) reaches a value of

$$t_{abs}(1.0 \text{ keV}) = N_{effec}\, 2.14 \times 10^{-22} \qquad (54)$$

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.) It is stored as the variable ***endedn***, the sole variable in common block ***endedn***.

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 (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.) It is stored as the variable ***efrend***, the sole variable in common block ***efrend***.

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 command is useful for stopping matter-bounded models. The results of this command are not exact; the final intensity ratio will be slightly larger than the ratio specified.

It is possible to change the scaling of the line intensities on the final printout by using the **normalize** command, as described on page 95. 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. The limit to the possible number of **stop line commands** is set by the variable ***mxstpl*** that appears in parameter statements throughout the code.

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 always a log, and 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^{-5} 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 be 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) 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] ^3P lines. The lowest temperature allowed, T_{low}, should be adjusted so that $h\nu \gg kT_{low}$ for the lowest excitation potential ($h\nu$) transition to be considered. Note that more than one temperature is sometimes possible when $T\sim10^3$ 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 temperature specified by the command will be the highest allowed temperature. The other rules for the command are unchanged.

The lowest temperature allowed is stored as the variable ***telow*** and the highest allowed temperature is stored as ***tehigh***. Both are variables in common block ***telow***.

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

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 **`parsecs`** appears on the line. The **`stop thickness`** command has the same effect as the optional second number on the **`radius`** command. 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. This limit is set by the value of ***ItrDim***, 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.

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 value of ***ItrDim***, 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 137).

14. CONTROLLING OUTPUT

14.1. Overview

CLOUDY is capable of keeping a printer going for hours. Several commands vary the printer's mass-loss rate, and are described here. A description of the meaning of the output follows in a section of Part IV.

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

Emission-line intensities are usually listed relative to the intensity of Hβ. These intensities are usually listed relative to a reference line intensity of 1.00. This command changes the reference line to any of the other predicted lines, and can change the relative intensity of the reference line.

The entire emission-line spectrum will have its relative intensity normalized to the intensity of the line whose wavelength is given by the argument.

The optional second number is a scale factor for the relative intensity array. 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 the reference line to have 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 searches for the first line in the emission-line stack whose wavelength matches the first numeric parameter on the command. There is a possible uniqueness problem since there can be more than one line with the same wavelength, especially for XUV or soft X-Ray lines. This degeneracy can be lifted by specifying the optional label. It may appear after the scale factor and is used to specify the four-character line label used by the code (see the section "Lines" in Part IV of this document). The label can be either upper or lower case and must be enclosed in single or double quotes.

The scale factor should appear before the label since a number is probably part of the label. If a label like "O 3" appeared before the scale factor, the code would interpret the scale factor as 3.

14.3. plot [type, range]

A plot of any of several properties of the calculation can be made. One of the keywords described below must appear on the command line. Up to ten plots can be generated. This limit is set by the variable ***ndplot*** that appears in several parameter statements throughout the code. The 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 102) 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^{-5} Ryd $\leq h\nu\leq 7.354\times10^{6}$ Ryd) is plotted. The range is altered by entering the two optional numbers with the **range** key. The default is for both the incident continuum (in units of νf_{ν}) entering the cloud (plotted as .'s) and that transmitted through the cloud (the o's) to be plotted. 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, for the last computed zone, will be plotted. This continuum is only that locally emitted by the gas and grains in the optically thin limit and unity filling factor, within the last zone.

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 continuum 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 diffuse emission from the cloud. This is meaningful only for non-spherical (open) geometries.

14.5. plot opacity [type, range]

If the keyword **opacity** is entered then the opacity of the first and last zones (per hydrogen atom) is plotted. The continuum between 1.001×10^{-5} Ryd $\leq h\nu < 7.354 \times10^{6}$ Ryd is usually plotted, unless this is adjusted by entering the optional energy range.

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^{-5} 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 138.

14.7. print _all

This command tells the code to print many emission lines whose intensities are simply scaled from lines usually printed, such as the higher Balmer lines, and many helium lines. These assume case B emissivities, and are not reliable at high densities, or when lines are optically thick. This option was added by P.G Martin.

14.8. print ages

This command tells the code to print all of the timescales associated with the **age** command (page 65). Normally only the shortest timescale is printed.

14.9. 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, in units s^{-1}. The second line is the vector of recombination rates, also 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 too many ionization stages need to be printed 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).

14.10. 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, as 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 carried by that agent.

This command sets the variable ***nzdump***, the sole element of block ***dump***, to the entered number. If set, then routine ***dmpary*** is called.

14.11. print continuum [diffuse]

If no keywords appear, this command tells the code to 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. Generally it is more useful to “punch” the continuum (see the discussion following page 102) and use other software to post-process this file.

If the keyword **diffuse** appears then the code will print the sum of the total emitted and inward reflected continuum as a series of points, with the label “nFnu”. This is the emission produced by the cloud, and does not include the attenuated incident continuum. If the **print line inward** command also appears then the total inward emission (the label “InwT”) and the reflected incident continuum (label “InwC”) will also be printed.

14.12. print departure coefficients

This command tells the code to print departure coefficients in addition to the relative populations for the lowest levels of hydrogen, H^-, helium, and some molecules.

14.13. print errors [io=9]

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 the Fortran i/o unit that appears as the optional integer argument. The default is 2. On many systems this output can be redirected to the screen.

14.14. 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. In the example above, it will print every 1000 zones on the first iteration, every 5 zones on the second iteration, etc. 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. If there are fewer numbers entered than iterations performed, then the last number entered will be used for all further iterations. The default condition is to print only the first and last zones.

14.15. print 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. 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.

14.16. 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.

14.17. 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.

14.18. 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 Part IV of this document gives more information

Any of a series of options can appear on the **print line** command[13]. Only one of these options is recognized on a single command line.

14.18.1. print line collisions

The code indicates various processes that contribute to formation of each line. Collisions are usually the dominant contributor to an optically thick line. To save space, the code does not normally break out the collisional contribution to lines, but this can be turned on with this command. The entry will have the label "Coll" followed by the wavelength.

14.18.2. print line pump

Most lines include fluorescent excitation as a line formation process. Pumping by the incident continuum will often be the dominant formation mechanism for optically thin high excitation lines. The **print line pump** option prints the contribution to the total line intensity from this process. This increases the size of the print out somewhat, so is not done by default. The entry will have the label "Pump" followed by the wavelength.

14.18.3. print line heat

Most lines include fluorescent excitation as a line formation process, and as a result they can heat rather than cool the gas. With this option, the heating due to line collisional de-excitation will be printed. This increases the size of the print out somewhat, so is not done by default. The entry will have the label "Heat" followed by the wavelength.

14.18.4. print line all

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

14.18.5. print line inward

Most lines calculate the fraction of the line which is emitted in the inward direction, towards the source of ionizing radiation. This will not be 50% if the line is optically thick. This command prints this fraction. This increases the size of the print out somewhat, so is not done by default. The entry will have the label "Inwd" followed by the wavelength.

14.18.6. print line optical depths [_off, faint]

This turns on the large print out of line optical depths 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 faintest line optical depth to print. The default smallest line optical depth to print is 0.1. All negative optical depths are normally printed.

[13] 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.

14.18.7. print line sum

This option adds an extra entry, the sum of the intensities of an arbitrary set of emission lines, to the final integrated emission-line spectrum. This can be useful for applications such as the Stoy (1933) energy balance method of determining stellar temperatures, which rely on the sum of a set of observed line intensities. 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, beginning on the line after the **`print line sum`** command appears. It 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 is the following integer. The following gives an example of its use.

```
print line sum
o  3 5007
totl 3727
o  1 6300
s  3 9532
end of lines
```

Up to 30 lines can be entered into the sum. This limit is set by the variable ***nrdsum*** that appears in parameter statements throughout the code.

14.19. print off

Turns off print out, as with the **`print quiet`** command (page 102).

14.20. print on

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

14.21. print only [header, zones]

The keyword **`only`** shortens the printout somewhat by stopping the calculation prematurely. If it appears, then another keyword, presently **`header`** or **`zones`**, must also appear. The command **`print only header`** will cause the code to stop after printing the header information. The command **`print only zones`** will cause the code to stop after printing the zone results on the first iteration.

14.22. 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.

14.23. print quiet

This command sets CLOUDY's quiet mode, in which nothing is printed at all. Printing can be turned off and then restarted at a particular zone by using the **`print starting at`** command described below.

14.24. 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.

14.25. print sort

This option causes the output spectrum to be sorted by wavelength rather than by ion. It was added by Peter G. Martin. It does not work.

14.26. 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.

14.27. punch commands

14.27.1. Overview

Punch (output on an arbitrary Fortran logical output unit) any of several possible pieces of information. The options are many. Physical quantities such as temperature, ionization, density, for each zone are produced for some options. 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 (called fort.n in Unix, where n is the integer on the command line) to then be post-processed by other plotting programs to produce final plots.

Punch output is a primary output mechanism for CLOUDY. Several of the most important options are described in following subsections. Other options that are not described in great detail are summarized in Table 19. One of the keywords must appear, and only one keyword per line is recognized. The four character key that must be matched is capitalized in Table 19.

This command causes a subroutine called ***DoPunch*** to be called after every zone calculation. The punch command is parsed by routine ***GetPunch***. Up to 20 **punch** commands can be entered. This limit is stored as the variable **`limpun`** that appears in parameter statements throughout the code.

14.27.2. The punch i/o unit

Each **punch** command has an optional Fortran I/O unit number to be used for the output. Unit 11 will be used[14] for the first file if no number appears on the line. If no I/O unit number appears on subsequent **punch** commands the unit number will be incremented by one on each call (i.e., the second file will be fort.12). This is to ensure that no more than one **punch** command writes to a given file. The default i/o unit number can be changed with the **set iPunDef** command described on page 137.

DANGER! It is up to the user to make sure that punch output on the same unit is not requested by more than one command – if it is then files will overwrite one another. For instance, the commands

```
punch element helium
punch element carbon 11
```

will result in the output from both commands going to unit 11 (because the first command used the default unit, 11), but the following will not:

```
punch element helium
punch element carbon
```

Note that CLOUDY uses Fortran logical unit 6 for output, and unit 5 for input when it is used as a stand-alone code. These units should not also be used for punch output, and the same logical output unit should not be used for more than one punch option. The code also performs other I/O at various times, and punch units greater than 90 are reserved for this purpose. These units cannot be used for punch output.

14.27.3. punch file name

A file name can be specified for the punch output, and some systems require that this be done. If the keyword **file** appears anywhere on the command line then the code will search for a file name between either single or double quotes (" or '). This must be a valid file name for the system in use. The following is an example.

```
punch overview, file="model.ovr"
```

Warning! The punch i/o unit number will still be controlled by all the rules described here, but the output associated with this unit will go to the specified file name. Care must be taken to not accidentally send more than one punch to the same output unit.

14.27.4. punch last iteration

Each punch command also has a keyword **last**, which 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 each iteration.

[14] The initial I/O unit number was 7 in versions 84 and before, and unit 10 in unreleased versions 85-87.

14.28. 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 56).

14.29. punch continuum

This command is the primary mechanism for saving the continuum predicted by the code. All continua are expressed as the flux νf_ν (units of νf_ν are erg cm^{-2} s^{-1}). They are relative to the inner radius of the cloud so the specific luminosity is the predicted quantity multiplied by $4\pi\ r_o^2$.

14.29.1. Emission line – continuum contrast

Lines are included in the resulting punch output for all command 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 too strong or weak relative to the continuum, depending on the actual line width. It 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 138. This issue is discussed further in the section Observed Quantities in part IV of this document.

14.29.2. Energy units for the punch output

By default the energy units are Rydbergs. If the keyword **units** appears on a **punch continuum** command line, the energy units can be changed to any of several energy or wavelength units. The keywords recognized are the following: microns, _keV, _eV_, Angstroms, and Rydbergs. Both the keyword **units** and one of these units must appear.

14.29.3. What is predicted

Column 1. The first column is the energy in Rydbergs.

Column 2. The second column is the intensity of the incident continuum at the illuminated face of the cloud.

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 diffuse thermal continuum and line emission emitted by the computed structure in the outward direction. 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

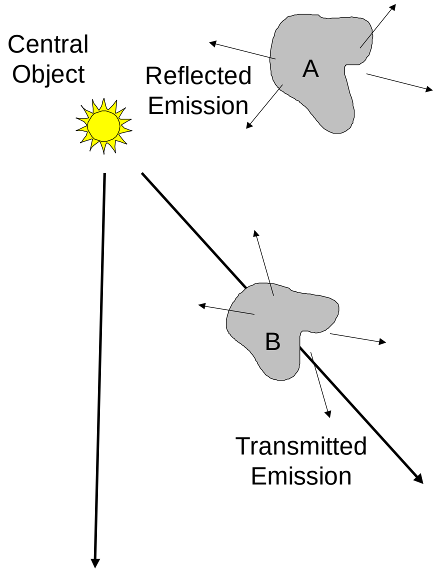


Figure 8 This figure illustrates several of the continua that enter in the calculations. ObsContin

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.

14.29.4. What is observed

Figure 8 illustrates a possible geometry. 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.

Three possible situations occur for the continuum source: a) we do not observe it, b) we observe the continuum transmitted through the emission line region gas (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.

For the line emission there are also three possible situations. First, we might only observe near side clouds (we see the outward directed emission). Second, we could only observe far side clouds, (only see the reflected component). Lastly, we might observe a symmetric geometry, reflected emission from the far side and transmitted 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 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.

14.29.5. punch continuum bins

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 $\nu+\delta\nu/2$.

14.29.6. punch diffuse continuum

This command is used to punch the local diffuse continuum (νf_ν erg cm^{-2} 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. The results are the contents of the array ***diffus***. Optical depth effects are not included and the continuum is the local continuum for the last computed zone.

14.29.7. punch emitted continuum

The continuum emitted and reflected from the nebula is punched. The first column is the energy in Rydbergs. 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 are in units νf_ν (erg cm^{-2} s^{-1}) and are relative to the inner radius.

14.29.8. punch incident continuum

The incident continuum, that emitted by the central object and striking the illuminated face of the cloud, will be punched. There will be two columns, the energy in Rydbergs and the continuum in νf_ν with units erg cm^{-2} s^{-1}.

14.29.9. punch interactive continuum

This will punch the integral of the product of the internal radiation fields 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 on the command line is the punch unit number. The second 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 this second number is less than 100 then it is interpreted as the energy in Rydbergs, and if greater than 100, the cell number.

14.29.10. punch ionizing continuum [options]

This command is used to punch the ionizing continuum at the end of the calculation. The first column is the energy in Rydbergs. 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, then continuum radiation fields, and the outward only continuum. The 8th number is the ratio of this quantity to the total integrated radiation field interaction rate, and the last number is the cumulative interaction. This allows the portions of the radiation field that have the dominant interaction with the gas to be identified. 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 punch unit number. The second 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 this second number is less than 100 then it is interpreted as the energy in Rydbergs, and if greater than 100, the cell number. The third optional number is the threshold for the faintest interaction to print, with a default of zero (all interactions printed), in the units used in the 8th column. The optional numbers may be omitted from right to left.

14.29.11. punch outward continuum

The photon energy is followed by the attenuated incident continuum, the outwardly directed continuum (the array ***outcon***), lines (the array ***outlin***), 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.

14.29.12. punch _raw continuum

This command is used to punch the "raw" continua, exactly that used inside the code, at the end of the calculation. The first number is the frequency. The next columns are the contents of the arrays ***flux***, ***otslin***, ***otscon***, ***refcon, outcon, outlin,*** and ***condif*** at this energy. Each is the number of photons stored in that cell, with units s^{-1} cm^{-2} $cell^{-1}$.

14.29.13. punch reflected continuum

This command is used to punch the reflected continuum at the end of the calculation if **`sphere`** is not set. The first column is the frequency in Rydbergs, the second the reflected continuum at that energy (νf_ν; erg cm^{-2} s^{-1}). The third is 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. Diffuse emission from the cloud is not counted in computing the albedo. The last column gives the label for continuum processes with thresholds at the energy.

14.29.14. punch transmitted continuum

This command is used to save the transmitted (attenuated incident and outward component of diffuse) continuum 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 43). Three cautions 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. Probably results from only the last iteration are needed, so either the **`last`** option should the used, or the file must be edited after the initial calculation to leave only the last computed continuum. Second, punch output should not be created on the same Fortran unit number 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 138) is not used in this command. This is so that lines have the correct intensity in the punch file, as needed for energy conservation, but has the effect of setting the line width to the speed of light.

14.30. punch convergence

This will punch the reason the model was declared "not converged" at the end of each iteration when the **`iterate to convergence`** command is used.

14.31. punch hydrogen

14.31.1. punch hydrogen conditions

This will punch the physical conditions and hydrogen constituents as a function of depth. The depth, temperature, hydrogen density, and electron density are followed by the densities of H^o, H^+, H_2, H_2^+, H_3^+, and H^- relative to the total hydrogen density.

14.31.2. punch hydrogen ionization

This will punch 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.

14.31.3. punch hydrogen populations

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

14.32. 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 ionization of each successive stage. 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 14 (page 50).

The first number on the resulting output is the depth 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. Each stage of ionization is represented as an integer, which is -1000 log10(n(i) / n_{tot}).

14.33. punch helium

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

14.34. punch cooling

The code will punch the cooling agents for each zone. The actual output is generated by routine ***PunCool***. The first two numbers after the zone number are the heating and cooling. The following numbers are labels for members of the array ***cooling*** 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 140).

14.35. punch heating

The code will punch the heating agents for each zone. The actual output is generated by routine ***PunHeat***. The first two numbers after the zone number are the heating and cooling. The following numbers are labels for members of the array ***heating*** 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 140).

14.36. punch ip

The code will punch 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.

14.37. 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 yields.

14.38. punch lines, options

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

14.38.1. 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. Only lines transferred in the uniform array formalism are included in this list. In particular, most forbidden lines will not be included in this list.

14.38.2. punch lines, cumulative

This option on the **`punch lines`** command tells the code to punch the log of the cumulative intensity of up to 15 emission lines as a function of depth into the cloud. This limit is controlled by the variable ***npunlm*** that appears in parameter statements. 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 integer used to identify 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
totl 4861
o  3 5007
totl 3727
o  1 6300
end of lines
```

There is an optional keyword **relative**. If specified then the punched quantities will be the intensity relative to the normalization line. If not specified, then the absolute intensity 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.

14.38.3. 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 was used to generate the second half of the chapter "Lines" in Part IV of this document. The code will stop after the data have been punched since it is left in a disturbed state.

The first set of lines are 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 next set of lines are the "level 1" lines, those with accurate atomic collision data. The last set of "level 2" lines is much larger and uses various g-bar approximations to generate approximate collision strengths.

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.

14.38.4. 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 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 must be the IO unit number, and the second number 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 the keyword **array** does, then the wide format is produced.

14.38.5. punch lines, labels

This tells the code to create a punch file with all emission line labels and wavelengths, in the same format as appears 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***.

14.38.6. punch lines, structure

This option on the **punch lines** command tells the code to punch the emissivity of up to 15 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 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 of this document) must appear in column 1-4 of the line, and the integer used to identify the line wavelength appears as a free-format number in later columns. The easiest way to do this is to cut and past the line identification and wavelength from a CLOUDY output.

The punch output begins with line labels and wavelengths. The remaining lines give the structure. The depth into the cloud (cm) is the first column. The remaining columns give the volume emissivity (erg cm^{-3} s^{-1}, for unit filling factor) for each line. The intensity is for unit filling factor, so the punched intensity should be multiplied by this to compare with observations of a clumpy medium.

The following illustrates its use;

```
punch lines, structure
totl 4861
o  3 5007
totl 3727
o  1 6300
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 run. The limit to the number of emission lines is controlled by the variable ***npunlm***, which appears in parameter statements throughout the code.

14.39. punch map

This command is used to produce a map of the heating and cooling rates as a function of temperature. The details of this map are described above, in the description of the **map** command. The range option is not implemented in the **punch map** command.

Normally 20 steps occur between the lowest and highest temperature in the map. The number of steps is controlled by the variable **nmaps**, and can be reset with the **set nmaps** command (page 138).

14.40. punch opacities [total, grain, element]

One of the keywords must appear.

14.40.1. punch total opacity

If the keyword **total** appears then the 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 and unit filling factor. The first column is the energy in Rydbergs and the second is the total opacity. The absorption and scattering opacities follow. The fifth column gives the local albedo, the ratio $k_s/(k_s+k_n)$, 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.

14.40.2. 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.

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

14.40.3. punch element opacity

If neither the **total** or **grains** keywords appear then an element name must be specified. The keyword is 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}). Each stage of ionization is punched in successive fort.n files beginning with the atom in fort.11, so the io unit number is ignored on this command. The code stops after producing these files.

14.40.4. 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.

14.40.5. punch opacity shell 11 26 5 3

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

14.41. punch Verner

If Katya Verner's FeII atom is linked into the code, this command will punch all ~10^4 lines at the end of the calculation. 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.

14.42. punch _OTS

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

14.43. punch overview

This option punches an overview of the model. This command is the most useful for obtaining information concerning the structure of the cloud. The first numbers are the depth, temperature, local heating, total hydrogen density, and electron density, all given as linear quantities.

These are followed by various ionization fractions, which are given as negative logs of the quantity. Neutral and ionized hydrogen fractions are followed by the ionization fractions for the three stages of ionization of helium, the first four stages of ionization of carbon, and the first six stages of oxygen.

14.44. punch _PDR

This command is used to output a set of quantities relevant to photodissociation region (PDR) calculations. The first column gives the depth into the cloud in centimeters. The second is the total hydrogen column density (cm^{-2}). The third column is the total extinction in magnitudes in the V filter, then 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 $n(H_2)/n(H)=1/2$.

14.45. punch pointers

The code will punch the element number, ion stage, 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.

14.46. 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 the best 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***. There is a common block, ***gett***, associated with this subroutine. This common block contains some information specified in the input stream, such as the ionization parameter, hydrogen density, stopping column density, flux of ionizing photons, etc. 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 the software described above.

Table 19 Miscellaneous Punch Options

Keyword	Output
GRAIN PHYSical	te, potential, drift velocity, frac heat
GAUNt factors	ff gaunt factor vs photon energy and temperature
HTWO	H2 formation, destruction, processes
LYMAn alpha	tau (Lyα), n(2p)/n(1s), T(excitation)
MOLEcules	H2/HDEN, CO/H, H2O/H, OH/H, CH/H
OPTIcal depths	nu, total, absorption, scattering optical depths
_OTS	nu, flux, otscon, otslin, outcon
_QS_	nu, qabs, qscat
RADIus	NZONE, R, R-Ro, dr
SPECial	Specially defined
TEMPerature	NZONE, T, dT/dr
WIND	Radius, thickness, velocity, accel, force multiplier

14.47. punch _dr_

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

14.48. punch charge transfer

Charge transfer recombination and ionization rate coefficients will be produced.

14.49. punch physical conditions

Physical conditions will be punched. The zone number is followed by the depth into the cloud, the temperature, hydrogen and electron densities, the radiative acceleration.

14.50. punch pressure

Various contributors to the total pressure in the gas equation of state will be punched. The depth, 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.

14.51. punch recombination [option]

14.51.1. punch recombination coefficients

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

14.51.2. punch recombination efficiency

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

14.52. punch source function [depth, spectrum]

14.52.1. punch source function, spectrum

The full spectrum of the source function for the local diffuse continuum will be punched. The first column is the continuum energy in Rydbergs. 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 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 Planck function is evaluated by routine ***plankf***.

The quantity in the last column is a measure of the local source function relative to the local Planck function. This will generally be nearly unity for a thermal plasma close to LTE. Ground states of atoms of hydrogen and helium generally have departure coefficients greater than unity, so the ratio will be less than one 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.

14.52.2. punch source function, depth

With this keyword the diffuse fields source function at a few energies will be punched for all depths in the cloud. The first column is the integrated optical depth from the illuminated face of the cloud to the current position. The second is the ratio of the diffuse field at an energy (photons per Rydberg per second) to the absorption opacity. Ordered pairs of these quantities occur for different energies.

14.53. punch special

If **`special`** is specified then a portion of the subroutine will be used, which can be changed to fit the circumstances. The user must provide this.

14.54. 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.

14.55. punch TPreditor

The code tries to estimate the temperature of the next zone from the changes in temperature that have occurred in previous zones of constant density models. This punch option allows the predictor correction to be examined. The output is the old temperature, the estimated new temperature, and the final equilibrium temperature.

14.56. 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.

14.57. trace zone 94 [iteration 2; options . .]

This command turns on "trace" information to follow the logical flow within CLOUDY. CLOUDY uses adaptive logic to control many choices, and this is a useful way to follow the internal decisions CLOUDY 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. 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.

Table 20 Trace Keywords and Effects

keyword	Quantity traced
BETA	OI 8446-Lyβ problem
CARBon	carbon ionization equilibrium
CALCium	calcium ionization balance
COMPton	Compton heating, cooling, and ionization
CONTinuum	prints out photon arrays, pointers
CONVergence	convergence loop, no other printout
COOLants	cooling
DIFFuse fields	sum of recombination coef in DIFFEM
_DR_	choice of next zone thickness
EDEN	changes in electron density
GAUNt	the free-free gaunt factors
GRAIN	details dealing with grain treatment
HEATing	heating agents
HEAVies	heavy element balance
HELIum	helium ionization equilibrium
HELIum ATOM	helium singlets ionization equilibrium
HELIum _IONized	helium ion ionization equilibrium
HELIum SINGlet	helium singlets ionization equilibrium
HELIum TRIPlet	helium ion ionization equilibrium
HYDROGEN FULL	all aspects of hydrogen solution
HYDRogen	hydrogen ionization equilibrium
IRON	Fe abundance, K-alpha emission
LINEes	line pointers, opacity. A's, etc
leveln	LevelN n level atom routine
Ly BETA	Lyβ - OI 8446 pumping problem
MOLEcules	rate coefficients for molecules
NEON	recombination, ionization for neon
OPACities	continuous opacities, zone by zone
OPTIcal depths	inner, outer optical depths in STARTR
oPTIMizer	steps in optimize command driver
_OTS	ots ionization rates
POINters	pointers for element thresholds

THREe body	three-body recombination rates for metals
TWO photon	induced two photon processes

14.57.1. trace convergence

This is a special form of the trace command that will print only an overview of how the calculation is going. It is not possible to start the trace on a certain zone or iteration.

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 an optimal set of parameters to fit a specified emission-line spectrum, line flux or luminosity, and/or a set of column densities. R.F. Carswell, who wrote much of the code for the present version, first implemented the method in CLOUDY. It uses 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.

15.2. Commands with Vary Option

All commands with the vary option are listed in Table 22. Section 15.21 on page 124 discusses details of some commands.

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
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
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 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

15.3. The Subplex Method

The Subplex method as coded by Rowan (1990) is the preferred optimization method. The numerical technique is similar to Press et al.'s Amoeba, but Rowan's implementation and formalism are far more robust. This is the default optimization algorithm, and was incorporated into CLOUDY by J. Ferguson.

15.4. The Press et al. Codes

Two methods from Press et al. (1992), the "downhill simplex" (Press et al.'s "Amoeba") and Powell's method, are incorporated into CLOUDY. Code for these methods is not included in the CLOUDY distribution files since Press et al. do not permit their source to be freely redistributed. To use one of these methods it is necessary to obtain first your own copy of the Press et al. routines, and then make a few modifications. These modifications are described in detail in a file in the Web page.

15.5. What must be specified

At a minimum, a desired emission-line spectrum, line flux, or column density, and a specification of which parameters are to be varied, must be given. 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. This limit is set by the variable ***limpar*** that appears in several parameter statements throughout the code. The quantities being varied are actually entered as logs within the code, and increments (the first steps away from the initial guess) are also logarithmic.

An example of the **vary** option in action is 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
*
* following are elements, stage of ioniz, log col density
optimize column densities
carbon 4 11.4
silicon 3 11.6
end of column densities
```

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.

15.6. no vary

It is sometimes useful to be able to turn off the optimizer for a given input stream, without having to change the (many) occurrences of the **vary** keyword. This can be done with a command line that has the key **no vary** in the first seven columns. If this command is entered then the **vary** keyword on the other commands will be ignored, and a single model will be computed.

15.7. optimize, amoeba

This tells the code to use the Amoeba routine (Press et al. 1992) to search for the best parameters. The code for this method is not included in the CLOUDY distribution since Press et al. do not permit redistribution of their source. The method is stored as the character variable ***oprout*** in the common block of the same name.

15.8. 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
```

15.9. optimize, increment = 0.5 dex

The increments are the amounts by which each variable is changed in the first step away from the initial parameter. 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
```

The increments are stored in the vector variable ***vincr***.

15.10. optimize, (intensity, luminosity)=36.425 [error =0.1]

The code will try to make the predicted intensity or luminosity of the normalization line (usually Hβ, and set with the **normalize** command) match the entered 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.

15.11. optimize, iterations =75

The upper limit to the number of iterations to be performed is specified with this command. The maximum number of iterations is stored as the variable ***itoptm***, which has the default of 20. It is probably a good idea to reduce the number of iterations if the initial guess is far from the solution. It may be necessary to increase the limit if the process is still making progress at the end of the calculation. The limit to the number of iterations should really be some function of the number of parameters being varied.

15.12. 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 limit is stored as the variable ***nobslm*** that appears in several parameter statements throughout the code.

```
*
* 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
*
```

One emission line is specified per input line, and the line must contain information in the following order. Columns 1 to 4 of the line must list the label CLOUDY uses to identify the line. This must then be followed by the wavelength, a free format integer. 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 IV, gives the emission line or continuum labels and wavelengths. The third quantity on the line is the desired relative intensity of the line. This will be in the same units as the relative intensities normally 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 95). If the scale factor for the reference line is changed with the **normalize** command then the intensities entered with this command are interpreted in terms of the rescaled, altered, units. 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 is assumed. An error of 0.05 would be 5%. 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 which 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 20.

15.13. optimize, Powell

This tells the code to use the Powell routine to search for the best parameters (Press et al. 1992). The code for this method is not included in the CLOUDY distribution since Press et al. do not permit redistribution of their source. The default is to use the Subplex (Rowan 1990) method. The method is stored as the character variable ***oprout*** in the common block of the same name.

15.14. optimize, punch =36

At the end of the optimization process the optimal input parameters are entered into a file for later use. Normally[15] this is Fortran punch unit 9 (i.e., fort.9 in Unix). The unit number can be changed with this command.

Using this file, it is possible to make later calculations in which various quantities are also 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.

Be careful that the unit specified with this command is not the same as one of those used by the **punch** command. The files will overlap if the same unit is used.

15.15. optimize, range -2.3 to 3.9

The preset limits to the allowed range over which parameters are to 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 that are varied. 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 arguments are ordered pairs 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. In this way the command is similar to the **optimize increments** command described above. Examples follow.

[15] This was unit 7 in versions 84 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. The limits to the range are stored in the vectors ***varang(1,n)*** (lower limit) and ***varang(2,n)*** (upper limit).

15.16. optimize, Subplex

This tells the code to use Rowan's (1990) implementation of the subspace-searching simplex optimization method. This is the default method, and is similar to Press et al's Amoeba, but Rowan's implementation is far more robust. The Subplex algorithm was incorporated into CLOUDY by Jason Ferguson. This is the only optimization algorithm included in the ftp distribution files.

15.17. optimize, tolerance = 0.2

The tolerance, a measure of the *overall* fractional agreement between all the observed and specified spectra, intensities, and/or column densities, is set with this command. The tolerance is stored as the variable ***vtoler***, with the default value 0.10. Individual uncertainties are set with the second number on the line that specifies observed values.

15.18. optimize, trace start 4

This command turns on trace printout for the nth time the code is called by the optimizer. Specific aspects of the trace are still controlled by the trace command, described on page 116.

15.19. optimize, trace flow

This command turns on trace for the flow within the optimizer. This command sets the logical variable ***lgOptimFlow*** to true.

15.20. Convergence criteria

The sum of residuals

$$\sum \max\left[\left(\frac{\text{observed} - \text{model}}{\text{observed} \times d}\right)^2, \left(\frac{\text{observed} - \text{model}}{\text{model} \times d}\right)^2\right] \qquad (55)$$

is accumulated over all relative intensities, line intensities or luminosities, and column densities, and minimized. In this expression *observed* and *model* are the observed and computed quantities. If the model prediction is zero then a residual of 10^{37} is returned. The uncertainty δ is specified when the observed quantities are read in and has a default value of 0.05 (5 percent). For column densities and relative line fluxes the linear quantity is used to form the residuals. To avoid floating point problems on 32-bit IEEE machines, the log of the line flux or luminosity is used. The process stops when the sum of residuals is smaller than the variable ***vtoler*** entered with the **`optimize tolerance`** command, which has the default of 0.10.

15.21. Notes concerning commands with vary option

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

15.21.1. 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`**.

15.21.2. elements

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

15.21.3. 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.

15.21.4. 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.

15.21.5. intensity

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

15.21.6. luminosity

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

15.21.7. metals

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

15.21.8. phi(h) and Q(H)

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

15.21.9. 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.

15.21.10. 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 67. 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.

15.21.11. stop column density

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

15.21.12. Stop thickness

Only one thickness can be specified.

15.21.13. table stars

Only the temperature can be varied, not the gravity.

15.22. Notes concerning the optimization process

15.22.1. 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 $\lambda4686/H\beta$ 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 limit the number of iterations the optimizer can perform (using the **`optimize iterations`** command) to stop it from over-optimizing a bad solution.

15.22.2. Change the increment size

The initial increment will be the largest step ever taken during the optimization process. 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.

15.22.3. Set physically motivated limits to the variable quantities

The optimizer driver uses a pure brute force method, and understands surprisingly little modern astrophysics. For instance, while trying to reproduce an observed He II $\lambda4686/H\beta$ 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.

15.22.4. 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. In order to avoid wasting time, it is also a good idea to limit the number of iterations with the **`optimize iterations`** command.

15.22.5. Additional Comments, by K. Korista

With the Powell, Amoeba, and Subplex optimization methods, it is usually a bad idea to vary too many parameters, especially those that affect strongly the structure of the cloud. Three is generally a reasonable limit. Often times, an optimization problem will have a χ^2 manifold which looks like the weakly dimpled surface of a golf ball (in multiple dimensions). In this limit, the solution found by the optimizer may depend heavily upon the starting conditions, as the optimizer falls into many local minima, finding that they all look about the same, and may then stop searching before finding the global minimum. Another way of looking at it is that the manifold becomes a densely laid minefield, and the optimizer never gets to its intended destination as it tries to make its way across. Limiting the number of variable parameters is one way to avoid this limit. The list of parameters to be optimized should be checked to ascertain whether, when varied, they will effectively impact the constraints (and whether there are many more of them than the number of constraints).

The number of parameters to be optimized may also be reduced in the following fashion. Optimizing parameters on a two or more dimensional grid of other parameters held fixed during the optimization is one way to search parameter space while minimizing the number of minima within the χ^2 manifold. For example, if two fundamental parameters (e.g., blackbody temperature and density) can be identified from the list of parameters to be varied, then for each coordinate position on the “fundamental plane”, an optimization may be done for which the remaining parameters (e.g., elemental abundances and ionizing flux) may be varied. I refer to a “fundamental plane” as that upon which good optimization solutions may be found over much of the plane to be explored. Of course, the fundamental plane could, in principle, be a cube. This procedure can be thought of as a “grid of optimizations”. The results may then be presented as contour plots, with the optimized parameters contoured on the fundamental plane. This is a much more realistic outlook to the problem of minimization, as there are unlikely to be unique solutions to problems such as those encountered here.

The user should also be aware that many times there exist regions on the χ^2 manifold where significant changes in parameter values result in minuscule effects in the χ^2. This has been observed to happen even near global minima. This is to be expected from the non-linear nature of photoionization and thermal equilibria. A related problem develops when the optimizer increments a parameter value into a regime where its effect has minimal impact upon the constraints. The optimizer can then get lost "on the vast plains" of a flat χ^2 manifold. This can be avoided by using the **optimize range** command.

Finally, the user can experiment with alternately freeing up and freezing certain parameters. This works especially well when the starting place is known to lie near a minimum. Increment sizes may be decreased from one optimization to the next, as one gets closer to the solution.

15.23. Code variables

The following is a partial list of the variables used to control the interaction between CLOUDY and the optimizer codes.

lgVarOn This logical variable says whether or not the current line image contains a **vary** option. It is used while parsing the input stream.

nparm This sets which variable is being varied. It is a pointer to the number in the stack of varied variables, (1 for the first, 2 for the second, etc.).

vformt This is a character variable. ***vformt(nparm***) is the Fortran format statement specifying how to write the control line for the ***nparm*** variable to be varied.

vparm ***vparm(nparm,1***) is the value of the first parameter on the ***nparm***th line to be varied, ***vparm(nparm,2***) is the second value on the ***nparm***th line to be varied, etc.

nvarxt The number of parameters on the ***nparm***th line to be varied is ***nvarxt(nparm***).

varang This specifies the range of allowed variation. ***varang(1,nparm***) is the lower limit, and ***varang(2,nparm***) is the upper limit.

nvfpnt is the pointer to the place in the input stream held by the current line.

vincr ***vincr(nparm***) sets the initial increment.

nvary This is the number of parameters that will be varied.

chOptRtn This is a four character variable which is the name of the optimization routine to be used. Valid values are 'AMOE', 'POWE', and 'SUBP'.

15.24. Other optimization methods?

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 preferable since it is usually necessary to make changes to the optimizer to make it truly CLOUDY-friendly.

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. compile stars

16.2.1. Kevin Volk's Stellar Atmospheres

Kevin Volk incorporated three large grids of stellar atmosphere continua into CLOUDY. The **`table star atlas, table star Rauch,`** and **`table star Werner`** commands use these. 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, then use these files with the **`table star`** commands. The first step is only done once while installing CLOUDY.

16.2.2. Preparing the Rauch stellar atmospheres

First obtain the tar file from Thomas Rauch's Web site, http://astro.uni-tuebingen.de/~rauch/flux.html and obtain the tar file with the complete data set. Uncompress and explode this file in its own subdirectory. This will create 57 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 file **`rauch.ascii`** that is needed for the next step. Only this file need be retained.

16.2.3. The Kurucz, Rauch, and Werner star files

Four files are needed to prepare to compile the stellar atmospheres, so that they can be used in model calculations. The **`rauch.ascii`** file was created above, and the **`werner.ascii`**, **`kurucz.list`**, and **`kurucz.ascii`** files come directly from the CLOUDY Web site.

`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 set of 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.ascii` This is the set of Rauch hot stellar atmospheres, and was created as described in section 16.2.2 above.

When the stars are compiled, as described next, three files, **`atlas.mod`**, **`rauch.mod`**, and **`werner.mod`**, are created.

16.2.4. Compiling the star files

The stellar atmosphere files are very large, and reading them using Fortran sequential access would be very slow. Direct access is used to read these files in a timely manner. Direct access reads are a part of the ANSII Standard Fortran. Unfortunately the form of the file is machine specific, so that these files are not portable, although the code used to read or write them is portable.

It is necessary to have CLOUDY compile the stellar atmosphere files before they can be used. This only needs to be done when the code is first installed, or if the energy mesh is changed. Follow these steps to compile the star files:

- Move the files **`kurucz.ascii, kurucz.list`** , **`rauch.ascii`**, and **`werner.ascii`** (ASCII files obtained from the web site or created above) into the directory where CLOUDY lives.

- 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 workstations.

- Three direct access files, **`atlas.mod`**, **`rauch.mod`**, and **`werner.mod`** will be created. These are the files which CLOUDY must access to use the **`table star atlas`**, **`table star rauch`**, or **`table star Werner`** continua.

- To execute the code from other directories it is necessary to set the path to the directory containing these two files with the **`set path`** command described on page 138 of this document.

When doing later photoionization calculations, the code will stop before computing a model if it cannot locate the **`atlas.mod`**, **`rauch.mod`**, or **`werner.mod`** files when the corresponding **`table stars`** command is entered. The code also checks that the energy grids in the star files and the code itself agree, to confirm that the star files are appropriate for the current version of CLOUDY.

16.2.5. Cleaning up after compilation

Only the **`atlas.mod, rauch.mod`**, and **`werner.mod`** are needed by CLOUDY, and they only need to be created one time. The files **`werner.ascii`**, **`kurucz.list`**, **`rauch.ascii`**, **`kurucz.ascii,`** and the *.wf atmosphere files can be deleted or compressed after the compilation is complete. They will only be needed again if the continuum binning within the code is changed. The Rauch *.wf files will never be needed again, unless one of the atmospheres is updated and it is necessary to recreate this set.

16.2.6. If the continuum binning is changed

The call to routine ***fill*** 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, and then change the values of three related variables within the code. During the compilation of the star files the code will note the values of the variables ***nWerner***, ***nRauch***, and ***nAtlas*** that must be entered. ***Block data scalar*** must be edited to enter these new values where they appear in data statements.

16.3. 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.

16.3.1. dielectronic recombination kludge [_on_, _off]

This command with the keyword **kludge** 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. For the four lowest stages of ionization, the code uses the means of the rate coefficients for C, N, and O; these are $3\pm2\times10^{-13}$, $3\pm2 \times10^{-12}$, $1.5\pm0.5\times10^{-11}$, and $2.5\pm1.4 \times10^{-11}$, $cm^3 s^{-1}$, for the successive stages of ionization (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. Because no better can be done at present, these are used for those ions that have no rate coefficients. These rate coefficients can be turned off, or the values changed by a scale factor, by entering 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. This is a useful way to check on the importance of this recombination mechanism for specific calculations.

16.3.2. dielectronic recombination Burgess [_on_, _off]

Rate coefficients for recombination at high temperatures is 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.

16.3.3. 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.

16.4. 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. A flag is set when the commands are entered. Parameters for the entire model (density, continuum, and luminosity) must still be specified. This special mode is entered after the last command is specified and the input stream ends with a blank line or the end-of-file.

16.4.1. drive escape probabilities

The command causes the code to enter a debug mode. The user enters the log of a one-sided optical depth and the code queries three of the escape probability functions and 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.

16.4.2. drive fread

This command causes the code to enter a debug mode in which the free format input reader, subroutine ***fread***, 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.

16.4.3. drive gaunts

This command enters a CLOUDY debug mode in which a driver requests temperature and photon energies, queries the free-free gaunt factor routine, and responds with the returned gaunt factor. The gaunt factor routine was written by David Hummer, and is described in Hummer (1988). It was extended to include the full range of energy and temperature needed by CLOUDY by J. Ferguson.

16.4.4. 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.

16.4.5. drive molecules

Here it is possible to change individual molecular abundances within the iteration loop.

16.4.6. drive pointers

This command allows the user to interrogate the photon energy array. After the continuum is generated, the driver will ask for energies in Rydbergs (interpreted as a log if negative) and respond with the cell pointer, frequency, cell width, boundaries, and the phase space factor $2h\nu^3/c^2$. Once complete, the calculation will continue as usual.

16.4.7. drive pumping

This queries the routine ***conpmp***, which evaluates the continuum-line pumping probability.

16.4.8. drive starburst

The code will ask the user to enter a metallicity, and will return the abundances of the elements, interpolating on Fred Hamann's grid of starburst abundances.

16.5. 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 used 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.)

16.6. fudge factors 4.59678 [12.3 34.5 958 ...]

The numbers appearing on the line can be communicated to any part of the code which calls the function ***fudge***. This function has a single integer argument that points to the numbers entered on the command line. In the example given above, a call to ***fudge***(2) would return the value 12.3. Up to ten numbers can be entered on the command line. Extra numbers are ignored.

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 in the large block data. The function ***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 pointer 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.

16.7. init [file="c84.ini", path]

This is a special command that tells the code to read a set of commands stored in 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 21.

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 name is specified on the command line. The code will search for **cloudy.ini** in the local directory where the code is executed.

Other file names can be specified with the **file** option. If the keyword **file** appears then the name within single or double quotes is used. 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 138). The path must be set before the **`init`** command is given. The path does not need to be set if the current directory contains the initialization file. If the path is set with the **`path`** command then the code will look for the initialization file in the local directory first, and if it is not there it will check the path. The code can be forced to check on the path first if the keyword **`path`** occurs on the **`init`** line.

16.8. no . . .

It is possible to disable physical processes as a test of their importance. If a physical process is turned off, then the logical variable ***physok*** in the common block of the same name is set to false to indicate that the treatment of physical processes is not OK. A warning is then printed at the end of the calculation, as a reminder that the results of the calculation are not to be trusted.

16.8.1. no (2p2s; 2s2p) collisions

This command turns off 2s-2p collisions in the hydrogen atom and helium singlets and ion. It is mainly used for debugging the hydrogen and helium atoms. The keyword can be either 2s2p or 2p2s.

16.8.2. no Auger effect

This command turns off the Auger effect.

16.8.3. 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.

16.8.4. 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.

16.8.5. no FeII pumping

This turns off H I Lyα pumping of Fe II.

16.8.6. no fine structure line optical depths

Fine structure lines, such as the ^{3}P 52, 88 μm lines of O^{+2}, can become optically thick under certain high-luminosity conditions (see, for example, Rubin 1983), and their absorption of the incident continuum can 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, including fine structure 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 be maximized since the lines will remain optically thin. The **`no induced processes`** command described below turns line heating off for all lines.

The line transfer arrays are injured by the `no fine structure` command - subsequent runs with the same core load will still have the line optical depths disabled.

16.8.7. no free free heating

Free-free heating is turned off with this command. It sets the logical variable ***freeon*** in the common block of the same name to false.

16.8.8. no grain neutralization

Ionic recombination through grain surfaces is turned off with this command.

16.8.9. no induced processes

This command turns off induced recombination and stimulated emission for hydrogen and helium, and continuum fluorescent excitation of all lines. It sets the logical variable ***lgFluor*** (the sole contents of common block ***fluors***) to false.

16.8.10. no molecules

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 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.

16.8.11. 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 (in the vector ***hreff(n,***2)) 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.

16.8.12. 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.

16.8.13. 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.

16.8.14. 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.

16.8.15. 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) and Xu and McCray (1991). This command will make X-Rays contribute 100% heat, and produce no secondary ionizations or Lyα excitations.

16.8.16. 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.

16.8.17. 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.

16.8.18. no vary

This command turns off the **vary** option set on various optimization commands. For a further discussion see page 119 above where the optimization driver is discussed in more detail.

16.9. 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.

16.9.1. 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***, with the default 1.92×10^{-9} cm^3 s^{-1}, and is used to set a rate coefficient of ***HCTMin*** × q where q is the excess charge of the heavy element.

16.9.2. 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 and is stored as the variable ***colimt***. The code stops when the CO abundance exceeds ***colimt***. Instabilities in the heavy element molecular equilibrium system of equations prevent higher carbon monoxide molecular from being successfully treated.

16.9.3. set csupra = -12.34

This command sets the secondary ionization rate due to supra-thermal electrons to the number on the line, the log of the rate (s^{-1}). The excitation rate of Lyα is assumed to be identical. This option is used to test the code in secondary-ionization dominated cases. The secondary ionization rate is denoted by the variable ***csupra***, and the rate of secondary excitation of Lyα and higher lines scale from this.

16.9.4. set didz 0.05

This command sets one of the main variables used to adjust the zone thickness, ***didz***. 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. The variable is the sole element of the common block ***didz***.

16.9.5. set _dr_ 11.2

This command sets the zone thickness. The argument is the log of the thickness in cm, and the default is 10^{30} cm.

16.9.6. 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.

16.9.7. set drmin 11.2

The number is the log of the minimum zone thickness (in cm). The default value is 10^{-20} cm.

16.9.8. set DstWght 0.2

The number is the fraction of the new destruction probability to use when evaluating the escape and destruction probabilities. The default value is 0.25.

16.9.9. set EdenError 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, and that iteration still be considered as having a converged electron density. 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 value (default of 0.01) is stored as the code variable ***EdenError***.

16.9.10. set Exemplar

The Fortran compiler on our Exemplar has a bug in its open file statement. As a result punch output will be lost when a job ends. This command should be entered when running on an Exemplar. The open statement will not be issued, and the punch file will not be lost, if this is present.

16.9.11. 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 in routine ***SetCon*** to choose the highest energy 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.

16.9.12. set iPunDef 21

When **punch** commands are given the code will assign successive output to successive fort.n files if no i/o unit is specified. The first unit will be the value of ***iPunDef***, which is set to 11 in routine ***zero***. This initial unit is reset to any integer greater than 6 with this command. This command must appear before any **punch** commands are entered.

16.9.13. 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 result in some compute time savings 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. The energy is stored as the variable ***EnergyKshell***, and a pointer to this energy is stored as ***KshellLimit***.

16.9.14. set negopc 12

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 optional number on the line is the Fortran i/o unit number for this output. The default is unit 10.

16.9.15. set nend 500

This command sets the default limit to the number of zones that will be computed. This limit is stored as the code variable ***nend***. 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. The only difference between these two commands is in the level of warning that will be generated by the code if it 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 was introduced so that it is 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.

16.9.16. 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 is stored as the variable ***nmaps*** and can be reset with this command.

16.9.17. 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.

16.9.18. 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 single or double quote, and ends with a single or double quote, or a space. 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 else then the " / " character will be concatenated, assuming that a Unix system is in use.

16.9.19. 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. The logical variable ***lgPHFIT*** controls whether to use the 1995 (true) or 1996 (false) results.

16.9.20. set PunchLWidth 10,000 km/sec

The observed contrast between 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 104). 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

$$uF_u\,(total) = uF_u\,(continuum) + \frac{c}{PunchLWidth} I(line) \qquad (56)$$

where c is the speed of light. This command set the value of ***PunchLWidth***. Values are entered in km s^{-1} and the default is 1000 km s^{-1}.

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. If the assumed 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.

16.9.21. set telow = 300K [linear][16]

The search for the temperature of the first zone can find temperatures as low as the value of the code variable ***telow***, which has a default value of 2.8 K. This command can be used to change the value of this lowest possible initial temperature. The argument is interpreted as a log if it is less than or equal to 10, otherwise as the temperature itself. There is an optional **linear** keyword to force all values to be linear, rather than logs. The number is the sole element of common block ***telow***.

After the first zone the calculation will stop if the temperature falls to values below ***tend***, which has a default value of 4000 K. ***tend*** is changed with the **stop temperature** command.

16.9.22. 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.

16.9.23. set trace 2

This set the level of detail in the trace output. The variable set by this command is ***LevTrace***, and its default is 0, full output. Larger numbers decrease the level of detail.

16.9.24. set trim –9 [upper lower]

The code saves 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, and may not be appropriate for some other conditions, or it for some reason your definition of trivial is different from mine. The smallest relative abundance for a stage of ionization higher than the ionization peak is changed with the **upper** keyword and the smallest relative abundance of ions below the peak is changed with the **lower** keyword. The defaults are 10^{-6} and 10^{-10} respectively. If no keyword appears then both are changed to the number entered. Generally, energies involved with high stages of ionization strongly exceed the ambient temperature, so these have little influence on calculations. This is not true for ions below the peak distribution. The argument is the log of the fractional abundance of the lowest or highest abundance to consider.

16.9.25. set tsqden 8

The code will perform 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 one deduced from the [OIIII] and HI spectrum, among other things. Such an analysis only makes sense for densities below the critical density of the [OIII] atom, roughly 10^5 cm^{-3}. The code will not print the results of this analysis if the density is higher

[16] This command replaces the **lowest temperature** command, which existed in versions 88 and earlier of the code.

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 lowest ending hydrogen density for which this analysis will be performed.

16.9.26. set WeakHeatCool 0.02

This command resets the threshold for the weakest coolant or heat source output with the **`punch heating`** (page 108) or **`punch cooling`** (page 108) commands. The command sets the variable ***WeakHeatCool***, which has a default of 0.05. The number entered is normally linear but interpreted as a log if it is negative.

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, Publ. A.S.P. 103, 1182

Allen, C.W., 1976, *Astrophysical Quantities*, Third Edition (London: Athlone Press).

Aller, L.H., 1984, in *Physics of Thermal Gaseous Nebulae*, (Reidel: Dordrecht).

Aller, L.H., & Czyzak, S.J. 1983, ApJ Sup. 51, 211.

Arimoto, N., & Yoshii, Y., 1987, A&A 173, 23

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

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., and Verner, D., 1995ApJ 455, L119

Baldwin, J., Wampler, J., and Gaskell, C.M., 1989, ApJ 338, 630

Bässgen, G., Bässgen, M., & Grewing, M., 1988, Ast.Ap. 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., and Pelan, A., 1995, A&AS 114, 367

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*, Hollenbach & Thronson, editors

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

Boyd, R., & Ferland, G.J., 1987, ApJ 318, L21

Bowen, I. S., 1960, ApJ 132, 1

Bregman, J.D., Allamandola, L.J., Tielens, A.G.G.M., Geballe, T.R., and Witteborn, F.C., 1989, ApJ 344, 791

Broad, J.T., & Reinhardt, W. P., 1976, Phys Rev A 14, 2159

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

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 Univ Press; London)
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
Carswell R.F. and Ferland, G.J. 1988, MNRAS 235, 1121
Castor, J.I., 1970, MNRAS 149, 111
Chaffee, F. H., & White, R. E., 1982, ApJ Sup. 50, 169
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., and 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, ARAA 24, 499
CrinkLaw, G., Federman, S. R., & Joseph, C. L., 1994, Ap J 424, 748
Crosas, M., & Weisheit, J.C., 1993, MNRAS 262, 359
Cruddace, R., Paresce, F., Bowyer, S., & Lampton, M., 1974, ApJ 187, 497
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
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, Ann. Rev. A&A 23, 119
Desert, F.-X., Boulanger, F., and 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, 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*, Kluwer, Dordrecht
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
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., 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 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. and Peterson, B.M., 1990, ApJ 363, L21

Ferland, G.J., Korista, K.T., Verner, D. A., and Dalgarno, A., 1997, ApJ 481, L115

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

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

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 (Cambridge Univ. Press, p. 15)

Guhathakurta, P., and 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., 1992, ApJ391, L53

Hamann, F., & Ferland, G.J., 1993, ApJ 418, 11

Harrington, J. P., 1969, Ap J 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

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., 1987, MNRAS 224, 801

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)

Janev, R.K., Langer, W. D., Post, D. E., & Evans, K., 1987, *Elementary Processes in Hydrogen–Helium Plasmas* (Springer--Verlag; Berlin)

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

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., & Feibelman, W.A., 1995, ApJ, 439, 793

Kingdon, J.B., & Ferland, G.J., 1995, ApJ 450, 691

Kingdon, J.B., & Ferland, G.J., 1996, ApJS 106, 205

Korista, K.T., & Ferland, G.J., 1989, ApJ 343, 678

Korista, K.T., Baldwin, J., and Ferland, G.J., 1997, ApJ in press

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*, A.C. Davis Philip, A.R. Upgren, & K.A. James (Davis, Schenectady), p 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., and 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, Astrophy Let 7, 69

Lepp, S., & Shull, J.M., 1983, ApJ 270, 578

Lightman, A. P., & White, T.R., 1988, ApJ 335, 57

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*, 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., & 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
Mendoza, C., 1983, in *Planetary Nebulae*, IAU Sym 103, D. R. Flower, Ed., p 143, (Dordrecht: Reidel)
Mihalas, D., 1972, *Non-LTE Model Atmospheres for B & O Stars*, NCAR-TN / STR-76
Mihalas, D., 1978, *Stellar Atmospheres*, 2nd Edition (W.H. Freeman: San Francisco)
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
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., and 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*, 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)
Palla, F., Salpeter, E. E., & Stahler, S. W., 1983, ApJ 271, 632
Peebles, P.J.E., 1971, *Physical Cosmology*, (Princeton U. Press; Princeton)
Peimbert, M., 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., Petitjean, P., and Boisson, C., 1991, A&A 251, 680
Peterson, J.R., Aberth, W., Moseley, J., & Sheridan, J., 1971, Phys Rev A, 3, 1651
Press W.H., Teukolsky, S.A., Vetterling, W.T., & Flannery, B.P., 1992, *Numerical Recipes*, (Cambridge University Press; Cambridge)
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
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
Rauch, T., 1997 A&A 320, 237
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)

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., 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* (Wiley, New York)
Sanders, D. B., et al. 1989, ApJ 347, 29
Saraph, H. E., 1970, J.Phys.B. 3, 952
Savage, B. D., and Sembach, K. R., 1996, ARAA 34, 279
Sciortino, S., et al., 1990, ApJ 361, 621
Scott, J.S., Holman, G.D., Ionson, J. A., & Papadopoulos, K., 1980, ApJ 239, 769
Schuster, A., 1905, ApJ 21, 1
Schutte, W.A., Tielens, A.G.G.M., and Allamandola, L.J., 1993 ApJ 415, 397
Schwarzschild, M., 1965, *Structure & Evolution of the Stars*, (New York: Dover)
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
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., and Nakada, Y., 1990, ApJ 349, 120
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*, (Interscience: New York)
Spitzer, L., 1978, *Physical Processes in the Interstellar Medium*, (Wiley: New York)
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., 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
Swings, P., & Struve, O., 1940, ApJ 91, 546
Tarter, C.B., & McKee, C.F., 1973, ApJ 186, L63
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. and Han, Z., 1996, MNRAS 281, 257
Turner, J., & Pounds, K., 1989, MNRAS 240, 833
Van Blerkom, D., & Hummer, D. G., 1967, MNRAS 137, 353
van Regemorter, H., 1962, ApJ 136, 906
Vedel, H., Hellsten, U., and 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, Atomic Data Nuc Data Tables 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., Korista, K., & Yakovlev D. G., 1996, ApJ, 465, 487
Verner, D.A., Verner, K., & Ferland, G.J., 1996, Atomic Data Nuc Data Tables, 64, 1
Voronov, G.S., 1997, ADNDT 65, 1

Voit, G. M., 1991, ApJ 377, 1158

Vriens, L., & Smeets, A.H.M., 1980, Phys Rev A 22, 940

Watson, W. D., 1972, ApJ 176, 103

Weisheit, J.C., 1974, ApJ 190, 735

Weisheit, J.C., and Collins, L.A., 1976, ApJ 210, 299

Weisheit, J.C., & Dalgarno, A., 1972, Ap. 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, (Kluwer; Dordrect)

White, R. E., 1986, ApJ 307, 777

Wiese, W.L., Smith, M.W., and 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 Relavistic Astronphysics*, M.P. Ulmer, ed., (World Scientific; Singapore), 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

Zycki, P.T., Krolik, J.H., Zdziarski, A.A., and Kallman, T.R., 1994, 437, 597

Zygelman, B., & Dalgarno, A., 1990, ApJ 365, 239

18. INDEX