Quick Start Guide to CLOUDY *C25*

Cloudy & Associates

Cover image: ESA's Ariane 5 gives JWST a Quick Start. Drawing from ESA: Artist's view of Webb on an Ariane 5 rocket.

CONTENTS

1   Introduction

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

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

Osterbrock & Ferland (2006; hereafter AGN3) give many more details about photoionized clouds. This is an active research field. Ferland (2003) reviews recent advances and current questions.

Table 1: Major CLOUDY release papers

Version	Year	Citation
C33 − 87	1983 − 1998	Ferland (1991, 1993, 1996)+
C90	1998	Ferland et al. (1998)
C13	2013	Ferland et al. (2013)
C17	2017	Ferland et al. (2017)
C23	2023	Chatzikos et al. (2023a) Gunasekera et al. (2023)
C25-rc1	2025	Gunasekera et al. (2024)

1.1   The web site and discussion board

Cloudy is open source. Its source, atomic data files, a large suite of test cases, and its documentation HAZY, are available at the web site www.nublado.org. Begin setting up the code by going to the web site and follow the instructions under *Instructions for downloading and installing the release version* .

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

The following are among the most useful pages on the web site:

- Download and setup the code

- Cloudy's Frequently Asked Question page

- The discussion group

- Just download the code and its documentation

- A list of improvements to the code

- Known problems with the code

- Hot fixes, small changes to the code that fix bugs

1.2   The release version

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

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

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

New versions of the code, and patches to the current version, are announced on cloudyastrophysics.groups.io.

1.3   HAZY, the documentation

Cloudy's documentation HAZY comes in four volumes. This *Quick Start Guide* gives an overview. Part 1 gives a complete list of all commands. Part 2 describes the output that is generated, shows how to extract observed quantities from the predictions, and explains how to call Cloudy as part of a larger program. The *Quick Start Guide*, and Parts 1 and 2, are kept up to date and PDF versions are included in the download in the directory `cloudy/docs`.

Part 3, which is badly out of date at the time of this writing, describes the physics behind the simulation. The past few years have seen many expansions of the code's capabilities, especially in infrared emission, molecular physics, dynamics and advective flows, and time dependent simulations. Unfortunately, Part 3 of HAZY has not had a high enough priority to be updated with available resources. I have not given up and do intend to update this section eventually. For now, an ADS search on Ferland will find the most recent papers that describe advances in the physics.

The LaTeX source needed to build the documentation is included in `cloudy/docs/latex`. The Perl script `CompileAll.pl` will compile all four parts of the documentation, leaving PDF files in the appropriate subdirectories.

1.4   Assumptions

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

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

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

Most calculations assume that the cloud is static. The ability to do dynamical and time-dependent calculations is now in the code although this part is not fully mature. Henney et al. (2005) and Henney et al. (2007) present examples of dynamical calculations.

The ***Introduction*** chapter in HAZY Part 1 gives more details on the assumptions. The section ***The Future*** on www.nublado.org outlines the directions future development will take. ALI line radiative transfer and expansion to two and three dimensional geometries are some of the planned future development.

1.5   What Cloudy can do

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

parameters.

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

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

1.6   Definitions

There is a fair amount of jargon that goes along with quantitative spectroscopy and numerical simulations of a cloud. These are summarized in the chapter ***Definitions*** of Part 1 of HAZY and in Chapters 1 through 5 of AGN3. As a minimum you should know the definitions of the two types of geometries (open and closed), the radiation field (incident, transmitted, diffuse, and reflected), and the terms covering factor, density, and column density. The Rydberg unit of energy is commonly used in this field. Many astronomers do not understand the distinction between H^0 and H_I. Don't be one of them! The difference is significant and is given on the web site.

1.7   Citing the use of CLOUDY

Publications should cite the code by giving the version number and a reference to the last major review of Cloudy's development, Chatzikos et al. (2023c). An example would be "We used version 07.02.01 of Cloudy, last described by Chatzikos et al. (2023c)." Citations are important for sustaining the development of Cloudy. The exact version should be given so that, years from now, it will be possible to find how a particular result was obtained. The command **print citation** will print the correct citation in a LaTeX-friendly format. Please follow the citation example shown above.

1.8   Collaborations

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

2   Two very simple models

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

2.1   What must be specified

Cloudy needs to be able to deduce the following information before it can predict conditions within a cloud;

- The shape and brightness of the radiation field striking the cloud.

- The total hydrogen density.

- The composition of the gas and whether grains are present.

- The thickness of the cloud.

Unless otherwise specified the gas-phase abundances will be close to solar values and grains will not be included. The total hydrogen density will be kept constant across the cloud since this is the default. Constant pressure, hydrostatic equilibrium, wind models and several other geometries could have been done.

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

2.2   Running a model

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

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

```
cloudy.exe -r sim
```

This will read from the input file `sim.in` and produce the output file `sim.out`.

Alternatively you can supply the CLOUDY commands on the command line. This is useful if you quickly want to execute a one- or two-line script. An example is:

```
cloudy.exe -e test
```

which will execute the command **test** (the "smoke test", see HAZY 1 for details).

The www.nublado.org web site has many more details about running this version of CLOUDY in the page RunCode.

It is also possible for a larger program to drive CLOUDY directly by treating it as a subroutine. See RunCode for more details.

2.3   Command format

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

Parameters are specified by numbers that appear on the command line. Many numbers are entered as logs. It is important to read HAZY to see what is expected for numerical input. For instance, numbers representing temperatures are interpreted as logs if they are less than or equal to 10 and as the linear temperature if greater than 10. Many commands have the keywords **log** and **linear** to change the default interpretation of a number.

2.4   A simple planetary nebula

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

```
blackbody, T=1e5 K   # the continuum shape
luminosity total 38    # the luminosity, log erg s-1
```

The spectral energy distribution is a 10^5K blackbody with a total luminosity of 10^{38}ergs^{-1}. The **blackbody** command specifies the shape of the incident radiation field and the **luminosity** command specifies its brightness. Luminosity and shape commands are described in Section 5 on page 22. The brightness of the incident radiation field is specified as either a luminosity or intensity as described on page 22 below.

   This example specifies a luminosity so the inner radius of the cloud must also be specified. This is the *luminosity case* described on page 22. A typical inner radius for the shell of a planetary nebula is a fraction of a parsec, so let's use 10^{18} cm. A typical hydrogen density, measured from ratios of forbidden lines (AGN3 Chapter 5), is roughly 10^5 cm^{-3}. Images suggest that the gas fully covers the star so we include the **sphere** command (discussed in Section 3.3.1 on page 17). A solar chemical composition would be used if we didn't change it. The composition of the ejected gas has been affected by nuclear processing and dust formed during late stages of mass loss. We use a built-in mixture of gas and dust that is specified by the **abundances** command described in Section 4.1.1 starting on page 20. That uses a composition that is typical of planetary nebulae. The input commands would be the following:

```
radius 18 # the log of the inner radius in cm
hden 5    # the log of the hydrogen density cm-3
sphere
abundances planetary nebula
```

We did not set an outer radius so the calculation will continue until the gas kinetic temperature falls below the default lowest temperature of 4000 K. Generally this will be near the H$^+$– H^{0}ionization front. Many other stopping criteria could have been used, see Section 3.5 on page 18, but in this simple model we are only interested in emission from the H$^+$ region.

   We will have the code create two output files in addition to its standard output. We add **save** commands to specify what quantity to output and a file name. **save** commands are described in Section 7.5 starting on page 30.

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

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

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

```
blackbody, T=1e5 K
luminosity total 38
radius 18
hden 5
sphere
abundances planetary nebula
iterate
print last iteration
save overview "pn.ovr" last
save continuum "pn.con" units microns last
```

Run the program as follows:

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

You will end up with three output files, the main output `pn.out`, and the two save files `pn.ovr` and `pn.con`. Have a look at `pn.out`. The output is described further in Section 7 on page 28. Use the two save files to create plots. The first line of a save file gives titles for the columns of numbers which are separated by tabs. The first column of the overview file gives the distance from the center of the central star to a point in the cloud. Another column gives the gas temperature. Figure 1 shows the kinetic temperature as a function of radius. The first column of the continuum file gives the wavelength in micron. Column seven gives the total emission. Make a plot showing the emission as a function of wavelength. Both axes will probably need to be logs due to the large dynamic range. The spectrum is shown in Figure 2.

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

2.5   A quasar cloud

Next compute a simple model of a cloud in the quasar broad emission-line region (BLR). The BLR clouds are located close to the central engine of an active nucleus (see Chapters 13 and 14 of AGN3) and they probe conditions near the most massive structures that formed in the young universe. The shape of a quasar continuum is often fitted by a set of power laws. Most of the literature in this field describes the continuum intensity in terms of the flux of photons in the hydrogen-ionizing continuum $\varphi(\mathrm{H})[\mathrm{cm}^{-2}\ \mathrm{s}^{-1}]$. This is the intensity case described in Section 5.1 on page 22.

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

A number of different spectral shapes are stored in the code as look-up tables. This **table** command uses a power-law continuum with a slope $f_v \propto v^{-1}$ in the visible/UV and with a roll over in the X-rays and infrared. The intensity of the incident radiation field is given as a flux of photons that are capable of ionizing hydrogen $[\mathrm{cm}^{-2}\ \mathrm{s}^{-1}]$. A starting radius does not need to be specified since this is the intensity case.
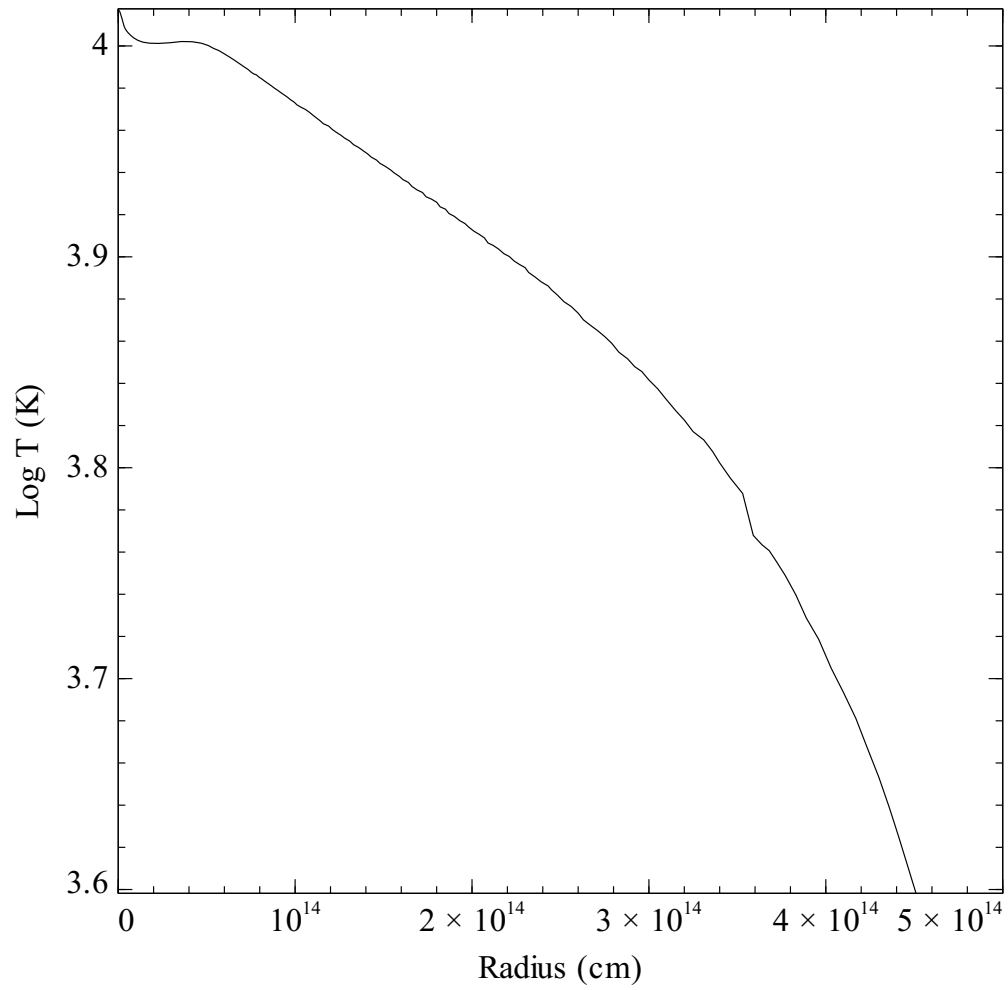
Figure 1: The gas kinetic temperature as a function of radius for a simple planetary nebula.
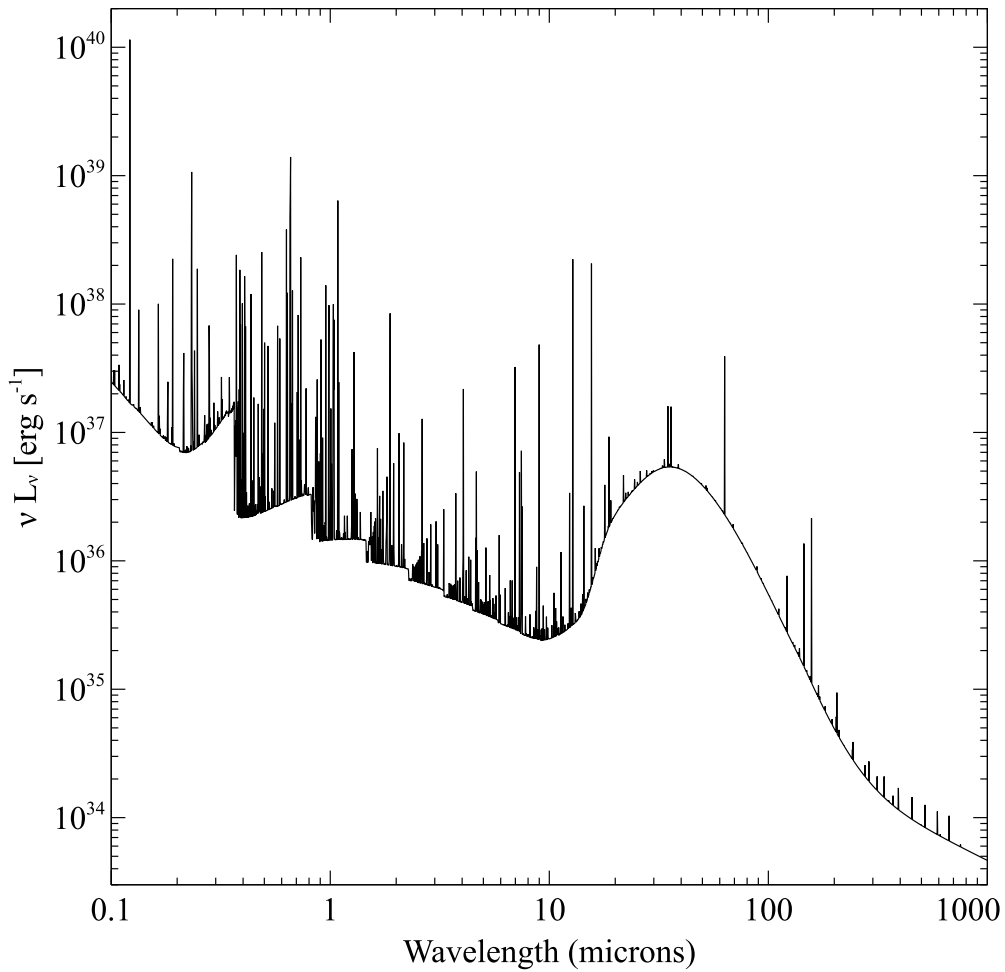
Figure 2: The spectrum of a planetary nebula. The *x*-axis is the wavelength in micron. The *y*-axis is νL_ν [erg s^{-1}]. The large bump peaking at $\sim 30 \mu$m is due to thermal dust emission. Most of the emission lines are hydrogen and helium recombination lines although strong forbidden lines are also present. Several radiative recombination edges are present across the optical – IR spectral region.

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

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

Many line transfer effects are very important in the BLR. It is necessary to iterate on the solution to converge the optical depths. But we only want to see the results from the last iteration. We do this by including the following commands

```
iterate to convergence
print last iteration
```

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

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

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

```
title BLR spectrum in Quick Start Guide
table power law  # a built-in power-law continuum
phi(H) 18.5      # the log of the flux of H-ionizing photons [cm-2 s-1]
#
hden 10          # log of hydrogen density cm-3
stop column density 22  # log of hydrogen column density cm-2
#
iterate to convergence
print last iteration
#
save overview "blr.ovr" last
save continuum "blr.con" units microns last
```

The lines beginning with "#" are comments and are ignored. Run the code like we did before. Examine the main output blr.out and make the same plots of the temperature structure and emitted radiation field. The predicted radiation field is shown in Figure 3.

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

---

[1]A planetary nebula is ionized by a hot star while the radiation field striking a BLR cloud is very energetic, extending far into the γ-Rays. The gas temperature falls to $\sim$ 100K on the neutral side of the H$^+$–H^0 ionization front in a PN since little radiation penetrates through the front. In a BLR cloud a warm partially ionized zone, heated by X-Rays, extends beyond the front so a stopping criterion must be specified. This is one of the major differences between a cloud ionized by starlight and one ionized by a hard non-thermal continuum.
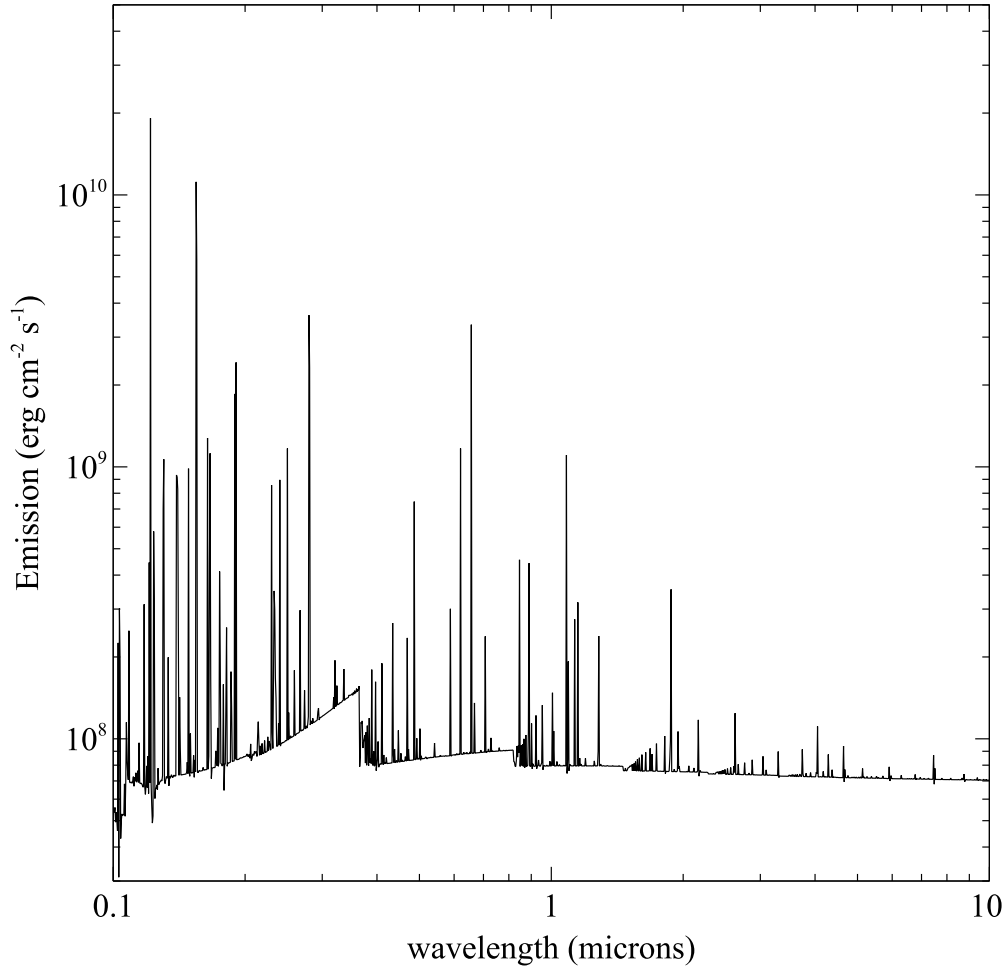
Figure 3: The spectrum of a single broad emission-line cloud in an active nucleus. The x-axis is the wavelength in micron. The y-axis is νf_ν [erg cm^{-2} s^{-1}].

2.5.1 Heads up!

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

Cloudy is designed to be autonomous and self aware. It continuously audits itself as a calculation progresses. The end of the calculation may have comments on various aspects of the physics. If problems occur then the code may generate a warning or caution. These are described further in Section 7.3 on page 29.

3 Geometry

3.1 Zones and iterations

The code works by dividing a cloud into a large number of thin layers called zones. There is a default limit of 1400 zones, but this can be changed with the **set nend** command. The code will generate a warning if the calculation stops because the default limit to the number of zones was reached since this probably was not intended.

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

3.1.1 Commands normally used

Frequently-used commands follow:

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

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

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

3.1.2 Heads up!

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

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

3.2   The geometry—intensity & luminosity cases

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

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

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

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

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

3.2.1   Commands normally used

Frequently-used commands follow:

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

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

3.2.2   Heads up!

None yet.

3.3   Open or closed geometry

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

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

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

There are two classes of closed geometries, static and expanding. In the expanding case, the default when the **sphere** command is included, line photons that cross the central cavity will not

---

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

be absorbed by gas on the far side due to Doppler shifts. In the static case (the **sphere static** command) emission lines cross the central hole and are absorbed on the far side. These considerations have little effect on most lines but do affect emissivity of higher-*n* Lyman lines of hydrogen.

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

3.3.1   Commands normally used

Frequently-used commands follow:

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

3.3.2   Heads up!

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

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

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

3.4.1   Commands normally used

Frequently-used commands follow:

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

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

$$u = \sqrt{u_{th}^2 + u_{turb}^2} \quad [\mathrm{km\,s^{-1}}] \tag{2}$$

to determine the total line width u.

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

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

3.4.2   Heads up!

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

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

The line width used in optical and UV astronomy is not the same as the line width used in radio astronomy. The description of the **turbulence** command in Part 1 of HAZY gives more information.

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

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

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

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

3.5.1   Commands normally used

Frequently-used commands follow. These are only a small fraction of the many ways that a calculation can be stopped.

radius sets the inner and outer radius of the cloud.

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

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

stop column density sets an upper limit to the hydrogen column density [cm^{-2}]. The default is to include hydrogen in all forms (H^0, H$^+$, and H$_2$) in this column density although keywords can be used to select only species like H$^+$ or H^0.

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

stop Av stops the calculation at a certain visual extinction. By default the A_V will be for a point source, which is the quantity measured in extinction studies of stars. The extended-source

extinction, appropriate for extinction across a cloud, is specified with the keyword **extended**.

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

3.5.2   Heads up!

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

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

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

3.6   What about clumping?

Clumping can be included. There are three general considerations.

   There are powerful selection effects at work when a range of densities exist. You will tend to observe the highest-density regions because the emission per atom is proportional to density if the line is below its critical density (see AGN3 Section 3.5 and Figure 4 on page 27). Only with a particular mix of densities, where the amount of material at each density exactly compensates for the change in emissivity, will an observer notice emission from a range of densities. So I am always very skeptical of claims that a range of densities contribute to a single emission line. This would require an amazing coincidence (Ferland, 2011).

   But clumps do exist. If the clump size is small compared with the physical thickness of the H^+ region then they can be treated with a filling factor (see the discussion in Section 4.3 on page 22, AGN3 Section 5.9, and Osterbrock and Flather (1959)). In this case the gas is modeled as small clumps that are surrounded by vacuum or much lower-density gas. This is done by simply including the **filling factor** command to specify the fraction of the volume that is filled by clumps.

If the clumps are larger than the physical thickness of the H^+ region then each clump will have its own ionization structure. This is the "LOC" model of quasar emission-line clouds described by Baldwin et al. (1995). The model is developed in several papers by the same team. In this case we compute grids of models and save the results. The spectra are then co-added using distribution functions to describe the range of cloud properties. The final spectrum depends on these distribution functions. The program `mpi.cpp` in the programs directory in the code distribution computes a grid of models and extracts the predictions using MPI to parallelize the computations. The **grid** command (see Section 6.4 on page 26) can also be used to compute large numbers of models. Giammanco et al. (2004) show CLOUDY calculations where optically thick clumps are present in the ISM.

Complex geometries are done by using CLOUDY to compute volume elements within a much larger cloud. An example is the Cloudy_3D code, available from sites.google.com/site/cloudy3d and described in Morisset (2006) and Morisset and Stasinska (2008). An image is shown in HAZY 1. The RAINY3D code is another example (Moraes and Diaz, 2009).

4   Composition and density

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

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

4.1   Chemical composition

The composition is set by specifying the abundances of the lightest 30 elements. Abundances are specified by number relative to hydrogen. On this scale a typical carbon abundance is $n(C)/n(H) \approx 2 \times 10^{-4}$.

4.1.1   Commands normally used

Frequently-used commands follow:

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

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

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

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

It is easy to create new types of grains by specifying their refractive index and size distribution data. This is described in the appendix *Using the Grain Code in Cloudy* in Part 1 of Hazy.

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

4.1.2   Heads up!

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

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

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

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

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

4.3   Commands normally used

Frequently-used commands follow:

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

constant pressure   The cloud will be isobaric or in hydrostatic equilibrium. The equation of state includes pressure terms from thermal gas motions, turbulent motions, a magnetic field, the nearly isotropic radiation pressure produced by trapped emission lines, and the outward push of the incident radiation field on the gas. The keyword **gas** says to keep the gas pressure, rather than

the total pressure, constant. This is discussed in the *Optical Depths and Radiative Transfer* chapter in Part 1 of HAZY. Self-gravity of the gas can be included with the **gravity** command, described in chapter *Density Laws*.

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

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

4.3.1   Heads up!

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

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

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

Turbulent and magnetic pressure, as well as gravity, can be included in the total pressure of a **constant pressure** cloud.

5   The incident radiation field

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

5.1   Luminosity vs intensity cases

In general the radiation field striking the cloud can be specified either of two ways. In the *luminosity case* the total luminosity emitted by the central object and the inner radius of the cloud are given. In the *intensity case* only the flux of radiation striking the cloud is specified.

5.2   The luminosity or intensity of the incident radiation field

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

5.2.1 Commands normally used

Frequently-used commands follow:

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

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

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

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

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

5.2.2 Heads up!

None yet.

5.3 The shape of the incident radiation field

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

The shape should be specified between the code's limits of 10 MHz (the lowest frequency observable with LOFAR) and 100 MeV, if possible. The code will complain but compute the model if the continuum is not specified over the full energy range.

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

5.3.1 Commands normally used

Frequently-used commands follow:

table SED will interpolate on a table giving pairs of frequency/intensity points. This is the most commonly used method of setting the shape since the results of other calculations, such a stellar atmosphere, can be directly entered.

table STARS is an expanded form of the **table SED** command which makes it possible to interpolate upon grids of SEDs predicted by stellar atmosphere models. The command has its own subsection in HAZY 1 and is further described on the *Stellar atmospheres* section of the *nublado.org* web site.

interpolate is an older method of specifying the SED as a table of points.

CMB adds the cosmic microwave background for any redshift.

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

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

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

The **save transmitted continuum, table read** pair of commands first saves the continuum transmitted through a first cloud, then uses the **save transmitted continuum** command to include that as part of the SED in a second calculation. Examples of doing this are the simulations `func_trans_save.in`, `func_trans_read.in`, `func_trans_read_scale.in` in the test suite.

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

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

table ISM is the local ISM background.

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

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

5.3.2   Heads up!

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

The shape of the incident radiation field should be specified over the entire energy range considered by the code, 3.040×10^{-9} Ryd to 100 MeV. An easy way to do this is to include, as a minimum, the cosmic microwave background and local ISM diffuse continuum.

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

The chemistry convergence will struggle, and likely find bizarre results, if the simulation extends into cold molecular regions but does not include the cosmic ray background. Interstellar chemistry requires a source of ionization to drive the network. Include the **cosmic rays background** command extends into the H^0 or H_2 regions.

The cosmic microwave background should always be included with the **CMB** command. This is not done by default. The command **no isotropic continua report** may be used to *not* include the CMB (and any other isotropic radiation fields) in the output fluxes. This affects *only* the output – CLOUDY always includes them in the solution of the physics problem.

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

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

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

6 Other commands

6.1 Radiative transfer

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

6.1.1 Commands normally used

Frequently-used commands follow:

Case B artificially sets the optical depths of hydrogen Lyman lines to very large values. Under some circumstances this will force the hydrogen recombination lines to their Case B intensities, the limit where all Lyman lines scatter often enough to be degraded into $L\alpha$ and Balmer lines (AGN3, Section 4.2). This command is only intended for setting up "homework" problems or test cases and should never be used in a simulation of a real object. It can have unexpected effects. If the cloud does not contain dust then the mean intensity of $L\alpha$ will become very large when this is used. This may result in unphysical photoionization rates for valence shells of third or fourth-row elements.

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

turbulence and **wind**   These commands are discussed in section 3.4.1 on page 17. They strongly affect line transfer by changing the line width and resulting opacities. The code assumes that only thermal motions broaden lines unless an additional component of motion is added with one of these commands.

6.1.2 Heads up!

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

6.2 The H_2 model

A large and complete model of H_2 emission can be included. It is slow and not used by default. Very simple approximations are used when the complete model is not included. These simple approximations may or may not be good representations of the physics of the real molecule.

The large H_2 models is used when the **database H2** command is included (see the chapter *Optical Depths and Radiative Transfer* of Part 1 of HAZY). There are also special save output options that allow the emission or absorption from this complex species to be saved and analyzed.

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

6.3   The optimize command

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

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

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

6.4   Grids of models

The greatest physical insight is often obtained from looking at results of grids of calculations to discover physical trends or correlations. An example is shown in Figure 4 giving the equivalent width of one of the strongest emission lines in quasars as a function of two cloud parameters (Hamann & Ferland 1999).

The code is designed to be used as a sub-program of other, larger, codes. The command format is the same when it is used as a subprogram but commands are entered by calling a special routine that contains the command line as an argument. Cloudy is then executed and the predictions are retrieved after the simulation finishes. This method of running the code is described in the chapter *Cloudy as a Subroutine* in Part 2 of HAZY. The test suite directory includes a subdirectory `programs` that includes a set of sample programs that illustrate this use of the code.

The **grid** command, introduced by Ryan Porter in C07, see Porter et al. (2006), makes it easy to do such a calculation without writing new code. The predictions in Figure 4 can be produced by modifying the simple BLR script given in Section 2.5 on page 10 to read as follows:

```
table power law
phi(H) 18.5 vary
grid from 17 to 24 in 0.5 dex steps
hden 10 vary
```
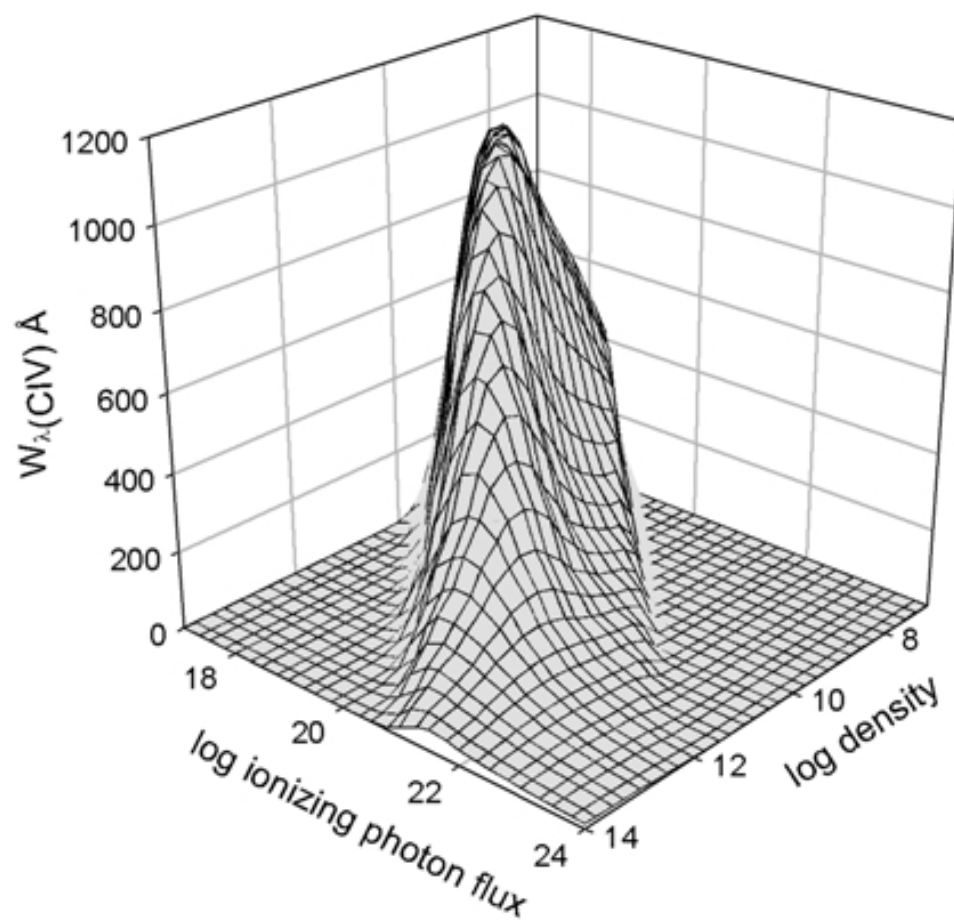
Figure 4: The predicted equivalent width of C IV $\lambda1549$Å as a function of cloud density and the flux of ionizing photons striking the cloud. See Hamann & Ferland (1999) and Ferland (2003) for further details.

```
grid from 7 to 14 in 0.5 dex steps
stop column density 22
save line list "blr.line" no clobber "LineList.dat"
save grid "blr.grd" no clobber
```

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

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

The **save line list** command makes it possible to read in a list of lines from the file given in the second pair of quotes and save the predicted intensity into the first file. This is how the results shown in Figure 4 were produced. The labels in the file must include both the four (or more) character string that is used in the output and the line wavelength. They must match exactly for the line to be recognized.

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

6.5   Miscellaneous commands

6.5.1   Commands normally used

Frequently-used commands follow:

atom changes the treatment of some model atoms. It has many keywords. **atom H-like** and **atom He-like** allow some aspects of the H-like and He-like isoelectronic sequences to be changed. **H2** turns on a large (and slow) model of H_2 (Shaw et al., 2005) emission.

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

Comments These can be included within the input stream. The section *Introduction to Commands* of Part 1 of HAZY explains how to enter several types of comments. Generally, any line that begins with a sharp sign, "#", is taken as a comment and ignored.

coronal time init This command can be used to study how a parcel of gas cools under conditions of constant density (isochoric) or pressure (isobaric).

7   The code's predictions

7.1   The default printout

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

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

the file `test.in` contains the code's input commands (read from `stdin`) and its output (written to `stdout`) goes to `test.out`. The output is fully described in the chapter *Output* of Part 2 of HAZY. The default output includes a copy of the input commands, a list of the abundances of the chemical elements and grains, the physical conditions in the first and last zone, an explanation why the calculation stopped, the intensity or luminosity of the stronger emission lines, and the mean ionization, temperature, and column density of many species.

7.1.1   Commands normally used

Frequently-used commands follow:

title enters a title that is printed at various places.

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

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

normalize The log of the radiated luminosity or intensity of each emission line is printed after the line's label in the main printout. The intensity of each line relative to a normalization line is also given. In optical emission-line spectroscopy the normalization line is usually $H\beta$ and this is the code's default. The **normalize** command specifies another line to use.

7.2   Understand why the calculation stopped

The reason the calculation stopped is given in the first comments after the last zone and is described further in the introduction to the chapter *Stopping Criteria* of Part 1 of HAZY. An example of the last zone printout and the statement of the reason why the calculation stopped is shown on page 19.

There are a number of default stopping criteria although the one most often encountered is **Calculation stopped because lowest Te reached.** This means that the gas kinetic temperature fell below the default lowest temperature of 4000 K. Generally this will be near the $H^+ - H^0$ ionization front. It is also common for very high metallicity gas to have low kinetic temperatures due to the thermostat effect (see 7.3 of Ferland (2003)) to that highly ionized gas can be quite cold.

Setting other stopping criteria do not ensure that they will be reached since the code will stop when the first criterion is reached. For instance, a PDR simulation might need a thickness corresponding to an Av of ten magnitudes. The command **stop Av 10** sets that limiting magnitude, but the simulation will probably stop near the $H^+ - H^0$ ionization front when the kinetic temperature falls below 4000 K. To make sure that the desired thickness is reached you should also set the lowest temperature to a small value, or remove temperature as a stopping criterion with the command **stop temperature off**.

7.3   Warnings, cautions, surprises, and notes

Examine the comments after the last zone for any warnings, cautions, or surprises. The code is designed to be autonomous and self-aware. It does many internal sanity checks to make sure that the calculation is valid (Ferland, 2001). If there are problems the code will say so.

- *Warnings* start with "W-" and indicate that something is seriously wrong with the calculation.

- *Cautions* start with "C-" and indicate that the code is on thin ice.

- *Surprises* start with "!" and indicate novel or interesting aspects of the results.

These are described further in the section *Warnings, Cautions, Surprises, and Notes* of Part 2 of HAZY.

7.4   Observed quantities

The chapter ***Observed Quantities*** of Part 2 of HAZY explains how to relate quantities predicted by the code to observed properties. The chapter ***The Emission Lines*** of Part 2 of HAZY gives an overview of the labels used to indicate various emission lines.

7.4.1   Multiplets and blends

Observed spectra often contain many emission lines. If these lines are too close, they may overlap and form what is known as a blend. Which lines form a blend can be difficult to predict. It depends on the intrinsic broadening of the lines in the source as well as the resolving power of the spectrograph. A low-resolution spectrum may show two closely spaced lines as a blend, while a high-resolution spectrum of the same source may show two separate lines. If you have a blend in your spectrum, you can still model it. CLOUDY allows you to define custom blends using the **set blend** command described in Part 1 of HAZY. CLOUDY also has a set of pre-defined "blends" in the file blends.ini which resides in the data directory. Many of the entries in this file are not blends in the strict sense of the word, but rather multiplets. These can be useful in various theoretical line ratios, such as temperature and density diagnostics. This file is read automatically when the code starts. This can be prevented by using the **no blends** command.

[discussion of line IDs, save options to identify lines, saving linelists, all in Hazy 2]
[emission line profiles]
[line equivalent widths]
[line luminosities]
[continuum specific luminosities]
[continuum band luminosities]
[column densities]

7.5   Save output

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

7.5.1   Commands normally used

Frequently-used commands follow:

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

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

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

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

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

7.5.2   Heads up!

Emission lines appear in the output produced by the **save continuum** command. The emission line to continuum contrast ratio depends on the size of the continuum mesh because the continuum cells are too coarse to resolve the lines. In an observation the ratio is partially set by the spectrometer resolution. The **set save line width / resolution** described in HAZY 1 changes the line to continuum contrast by changing the resolution that is assumed in adding the lines to the continuum.

The emission in the **save continuum** command includes a covering factor when the luminosity case is used. See the section *Units of the save output* in HAZY 1 for more details.

The emission line wavelengths follow the convention that vacuum wavelengths are used for $\lambda < 2000$Å and STP air wavelengths are used for $\lambda \geq 2000$ Å. The **print line vacuum** command tells the code use use vacuum wavelengths throughout. The continuum is always reported in vacuum wavelengths to avoid a discontinuity at 2000 Å.

7.6   Miscellaneous Helper Commands

[point to species, or make discussion of species, discuss lines, databases]

There are a number of commands that can ease the handling CLOUDY's results. These are discussed in the HAZY 1 chapter "Controlling Output."

save species labels all produces a list of all species known to the code.

save lines labels produces a list of all the emission lines handled by the code. Transitions obtained from external databases are listed with information connecting them to the original data.

save species lines and **save lines data** report emission line fundamental data, which include the gf, and Einstein A values of the transitions.

8   Example calculations

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

8.1   The test suite

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

The test suite has three parts:

- `auto` contains several hundred simulations and will run in about half a day on a modern workstation. This is run every night here in Lexington.

- `Slow` includes simulations with the large H_2 and Fe II atoms and takes several days to run.

- `Programs` show how to use the code as a subroutine of larger programs.

Use the Perl script `run_parallel.pl` to run all simulations in the auto test suite. The script contains detailed instructions for using it in various ways. This is an important step in setting up the code since it confirms that you have a valid version. A code as large as Cloudy is likely to discover bugs in a compiler especially when highly optimized code is produced.

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

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

8.2   One of the models...

This section considers one of the models in the test suite, `orion_hii_pdr_pp.in`, in detail. A much longer discussion with more details is given in the chapter *Output* in Part 2 of HAZY. This simulates a plane-parallel molecular cloud with an H II region and PDR on its surface. Radiation from a nearby O star and galactic background cosmic rays are the only sources of heat and ionization. The calculation follows a ray of light into the cloud. The illuminated face is an H^+ region, an ionized layer with a temperature of $T \approx 10^4$K. The PDR, a largely atomic region with $T < 10^3$K, occurs at a depth where hydrogen-ionizing radiation has been extinguished. The shielded face of the PDR is a molecular cloud with $T < 100$K. This calculation is described further in Ferland (2003).

The gas equation of state is the relationship between temperature and pressure. It includes gas pressure, the outward push caused by absorbed starlight, the magnetic, and turbulent pressure. Turbulence is assumed to be in equipartition with the magnetic field. Flux freezing, where field is well coupled to the gas, is assumed. In this model the layer is in hydrostatic equilibrium with gas and magnetic pressures balancing the outward push of the stellar radiation field.

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

8.2.1   run orion_hii_pdr_pp.in

Run the `orion_hii_pdr_pp.in` script with the command

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

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

8.2.2   Examine the main output

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

The simulation may have had pressure convergence failures where the cloud passed through a *thermal front*. Thermal fronts, and the problems they cause, are described in the chapter *Problems* of Part 2 of HAZY. Convergence problems are announced with lines that begin with the string "PROBLEM".

Next, identify some of the strongest emission lines in the spectrum. These are listed towards the bottom of the output following the string `Emission Line Spectrum`. Two iterations were performed to converge the optical depth scale. Make sure that you are looking at the last iteration (similar information is printed for each iteration). The command **print last** would tell the code to print only results of the last iteration but is not used in this test. The emission-line intensities are given relative to Hα. If dust were not present the Lα / Hβ ratio would be about 34 (AGN3 Chapter 5). Actually Lα is predicted to be only about twice as strong as Hβ because Lα is efficiently absorbed by dust. The line [O III] $\lambda 5007$Å is one of the strongest lines in the spectrum, as expected for an H II region.

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

Some integrated properties of the cloud are listed towards the end of the file. Column densities of various species are given. The line with `Log10 Column density (cm^ -2)` gives column densities of H^0, H$^+$, and H$_2$: the cloud is predominantly molecular. Mean temperatures are also given following the line `Log10 Mean Temperature (over volume)`. The H$^+$

region has a mean temperature of nearly 10^4 K. The H^0 region has a mean temperature a bit under 10^3 K, and the H_2 region has a mean temperature of around 20 K. The output ends with a list of the asserted quantities. These compare the predictions of your executable with its historical predicted quantities.

The chapter *Output* in Part 2 of HAZY goes over the code's output in detail. Have a look.

8.2.3   The save output

The input script contains many **save** commands. Some are only intended as debugging aids but others contain a wealth of physical information about the results of the simulation. The **save** commands are described in the chapter *Controlling Output* of Part 1 of HAZY while the chapter *Observed Quantities* of Part 2 of HAZY explains how to extract some observed quantities from all of this output.

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

orion_hii_pdr_pp.con is produced by the **save continuum** command and gives the incident, reflected, transmitted, and total continua. These are defined in the chapter *Definitions* of Part 1, and are described in both chapters *Controlling Output* of Part 1 and *Observed Quantities* of Part 2 of HAZY. The **units** option changes the energy scale from Rydbergs (the default) to microns. These wavelengths are the x-axis in the Figure 5. The plot shows the incident stellar continuum, listed in column two, as the smoother line. The total emitted continuum, contained in column seven, is the line with a great deal of structure. This is mainly the "reflected" spectrum, the continuum emergent from the illuminated face of the H^+ region. Extinction by dust in the molecular cloud prevents much light from emerging from the shielded face.

orion_hii_pdr_pp.ovr is produced by the **save overview** command. It gives the electron temperature and density, the hydrogen density, the heating rate, and the ionization distribution of H, He, C, and O. Figure 6 shows the hydrogen ionization and chemical structure as given in this file. The x-axis gives the log of the depth into the cloud in cm. The y-axis gives the log of the fraction of hydrogen in the form of H^+, H^0, and H_2. The hydrogen ionization front occurs at a depth of $\sim 2 \times 10^{17}$ cm. There is a small H^0 region and the rest of the cloud is mostly H_2.

orion_hii_pdr_pp.grntem gives temperatures of grains as a function of depth. These, together with the electron temperature contained in the overview file, were used to create Figure 7. The highest curve is the electron temperature and is $\sim 10^4$ K across the H^+ region, falling to ~ 500 K in the small H^0 region, then to below 100 K in the H_2 region.

The calculation includes graphitic and silicate size-resolved grains. Their temperatures are shown in Figure 7 as the lower cluster of curves. They have a range of temperatures ~ 100–200 K across the H^+ region and grow cooler in the molecular region. The gas and dust temperatures approach one another deep in the molecular cloud.

orion_hii_pdr_pp.mol gives molecular densities as a function of depth and was used to produce Figure 8. It also includes several measures of extinction due to dust.

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

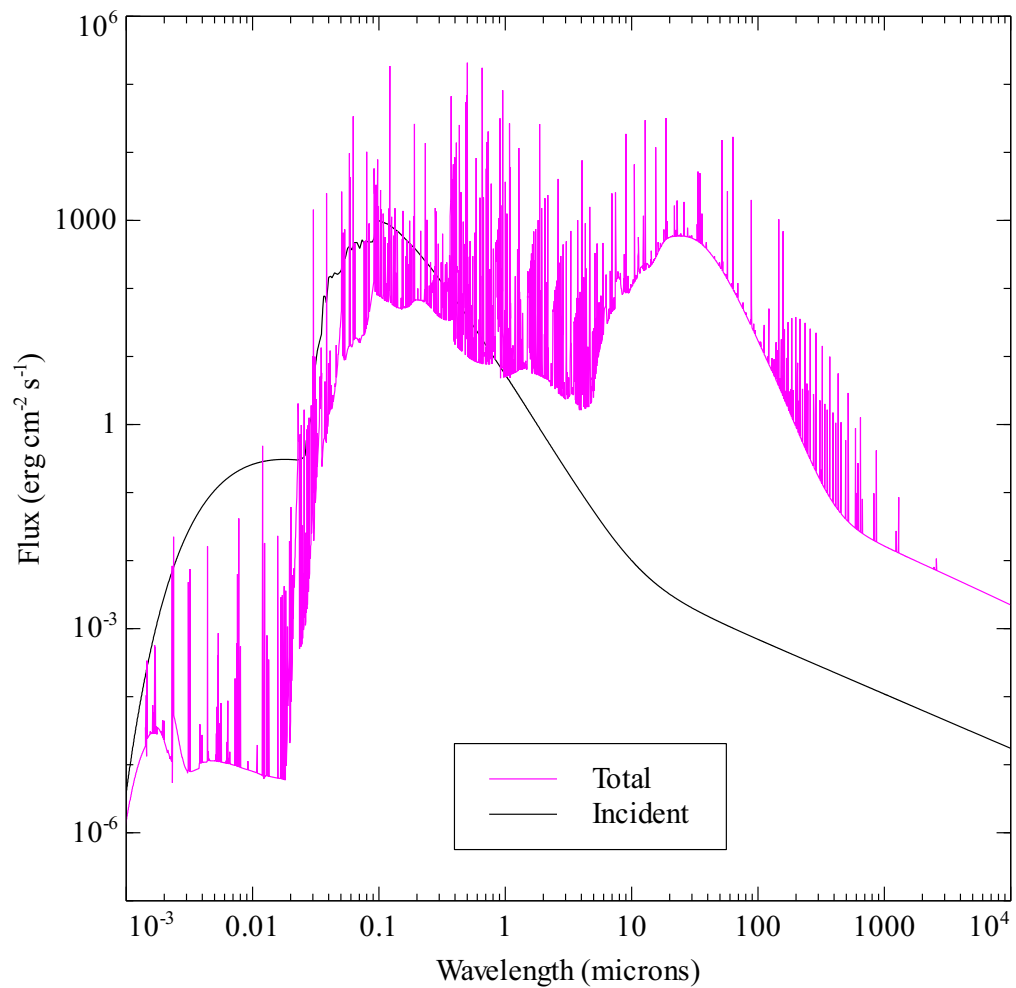Figure 5: The incident (smooth) and emitted spectrum contained in the orion_hii_pdr_pp.con file. The x-axis is the wavelength in micron and the y-axis gives νf_ν [erg cm^{-2} s^{-1}]. This is produced by the **save continuum** command.
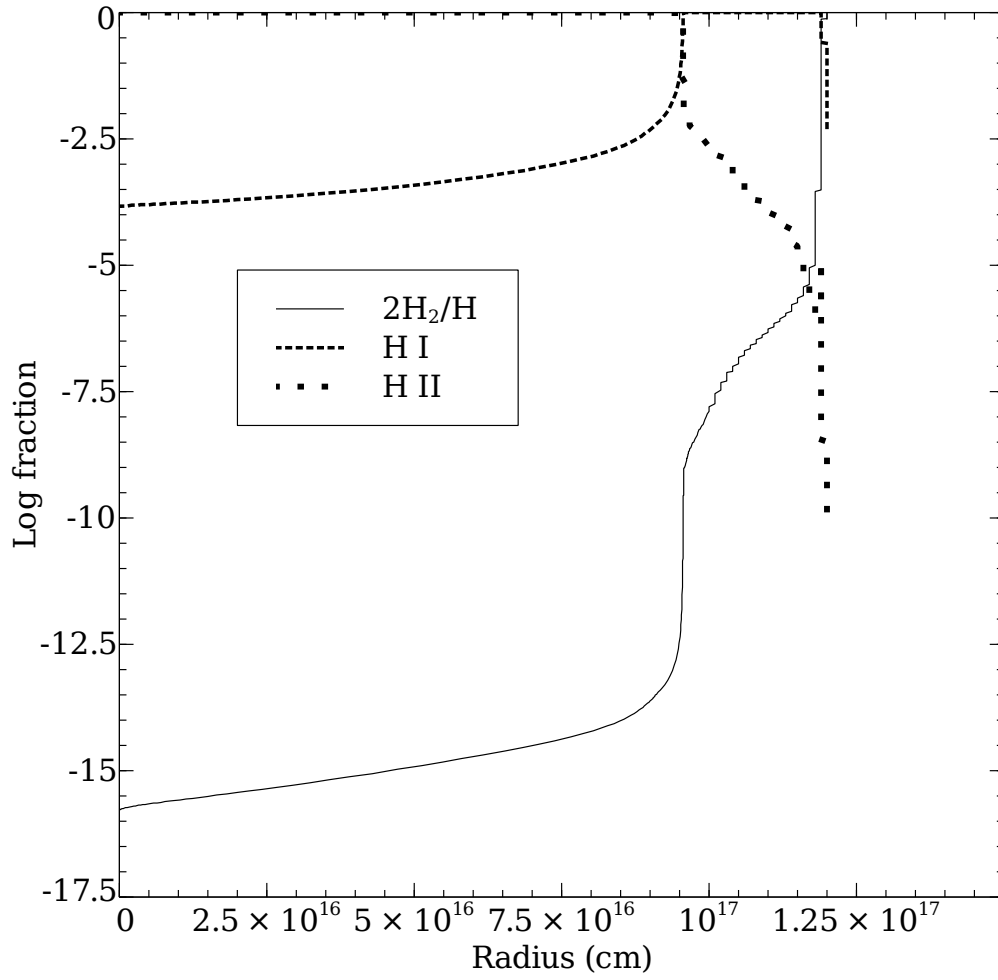
Figure 6: The hydrogen ionization structure. The x-axis is the depth in cm and the y-axis gives the log of the fraction of H in H^+, H^0, and H_2.

Figure 7: Temperatures of grains (the lower cluster of curves) and free electrons (the higher curve). They become nearly equal deep in the molecular cloud. The y-axis is the temperature and the x-axis is the depth into the cloud.

Figure 8: Densities [cm^{-3}] of some of the molecules included in the calculation are shown as a function of depth [cm] into the cloud.

8.3 Heads ups for classes of objects

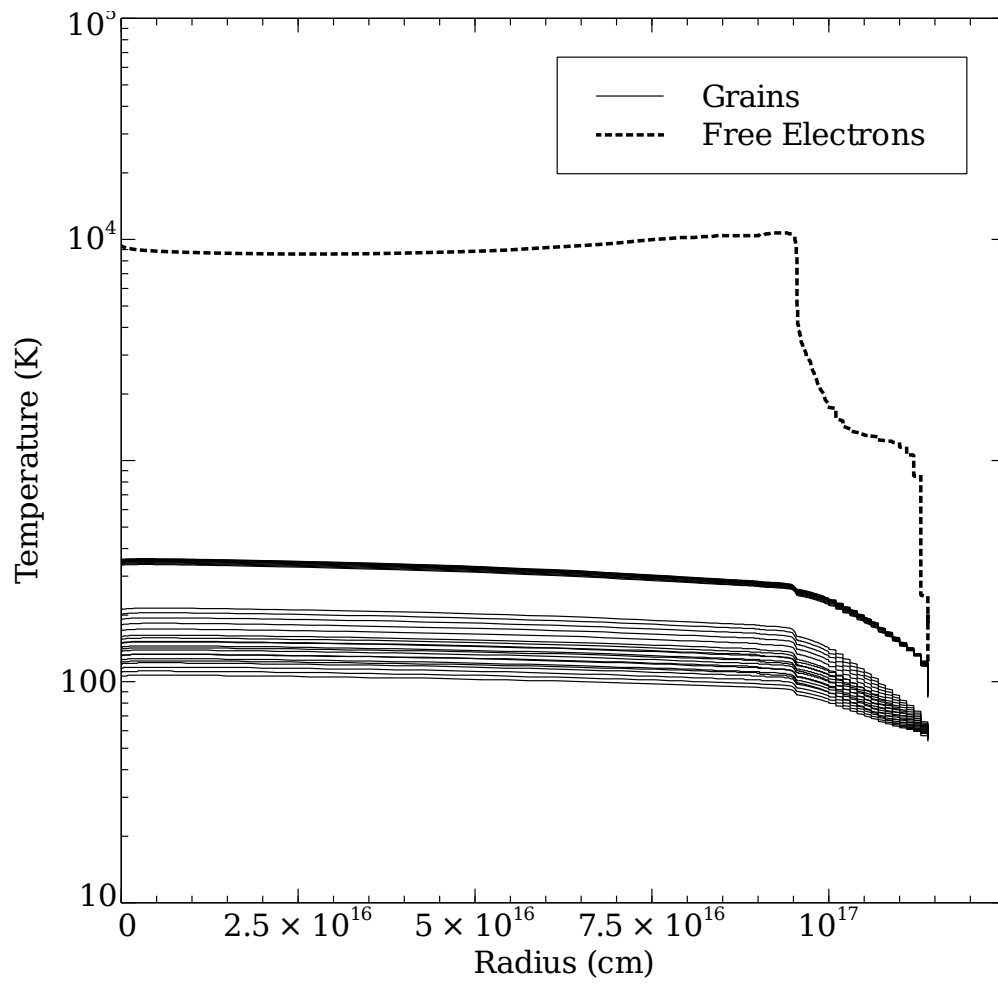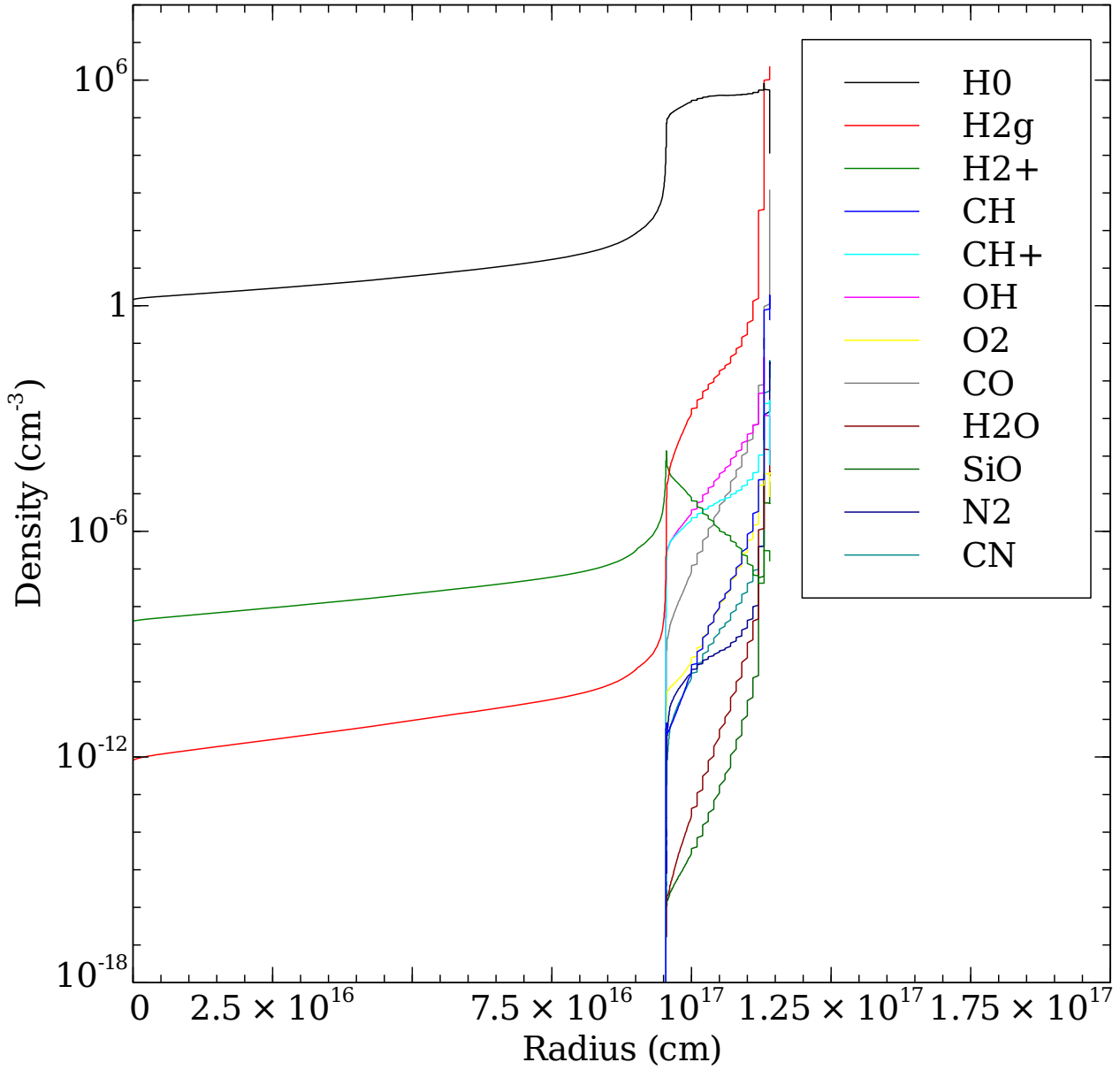Simulations of several types of astronomical objects are in the test suite. They can be grouped together by the first part of their filename. The following subsections describe important considerations for setting up each of these classes of simulations.

8.3.1 BLR of Active Nuclei

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

8.3.2 NLR of Active Nuclei

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

8.3.3 PDRs of star-forming regions

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

The **double** command should be entered if the calculation stops before the outer edge of the molecular cloud is reached. See the discussion in Section 3.5.1 on page 19.

The culture in the PDR modeling community is to treat the PDR as an isolated phenomenon rather than an extension of the H II region. Nature does not do this, but you can force the code to do it by removing all hydrogen-ionizing radiation from the incident continuum. This is done with the **extinguish** command (see Section 5.3.1 on page 24).

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

By default the code will stop when the gas kinetic temperature falls below 4000 K. PDRs are generally cooler than that. Many PDR calculations assume that the cloud thickness corresponds to

a visual extinction A_V of 10 mag. To do this we must tell the code not to stop when the gas becomes code, and to use this extinction. This would be done by

```
stop temperature off #do not use kinetic temperature as a stopping criterion
stop Av 10  #stop when visual extinction reaches 10 magnitudes
```

8.3.4   Galactic nebulae

Cosmic rays and the ISM background should be included. Dust is almost certainly present in the ionized gas and should be included along with depleted abundances of the refractory elements. This is important since the grains extinguish the radiation field, their photoelectrons can heat the gas, and the gas-phase depletions they cause change the cooling function (Kingdon et al., 1995). Grains are included by selecting a mix of gas and solids that includes dust (with one of the **abundances** commands) or by explicitly including these (with a combination of the **grains**, **abundances**, and **depletion** commands).

9   How to make this plot

This section includes examples plots that might be found in research papers on spectroscopy. It gives an example of the plot, along with an explanation of how to do the calculation and create it.

9.1 H_2 excitation diagrams

H_2 excitation diagrams show observed or computed level populations as a function of the excitation energy in the molecule, as shown in Figure 9. See AGN3 Chapter 8 and Appendix 4 for background information. The files used in this example were created with the test suite Baldwin et al. (1991) Orion HII/PDR sim and are included in the `docs/latex/QuickStart` directory with names starting with `H2_BFM`.

Run a simulation that includes the large H_2 molecule (with the **atom H2** command) and include the **save H2 column densities** command. The first line of the **save** output file gives column headers. The next three lines give the ortho, para, and total H_2 column density, which we don't need here. The remaining lines give the column densities and excitation energies for all levels in the ground electronic state of H_2.

Import the output of the **save H2 column densities** command into Veusz. In the import data dialogue specify that the data are tab delimited and to ignore the first three rows after the header. Import the rest of the data.

Create an x-y plot and choose column three, the excitation energy in K, as the x-axis. Choose **colden/stat wght** as the y-axis. We divide by the statistic weight to avoid having the data jump up and down on the plot by a factor of three due to the ortho/para distinction. For plot formatting, choose no line (set line color to while) and use a symbol for the points. Make the y-axis a log but leave x as linear.

Choose an x-axis range to suite your observations. The range 5×10^3 K $- 3 \times 10^4$ K includes most levels that can be detected in the 2 μm window.

Next we need to add lines indicating the excitation temperature. It is easiest to use a spreadsheet to create pairs of points to define this line. Export this data into a tab delimited file so that Veusz can read it. Import it into Veusz as a new data set.
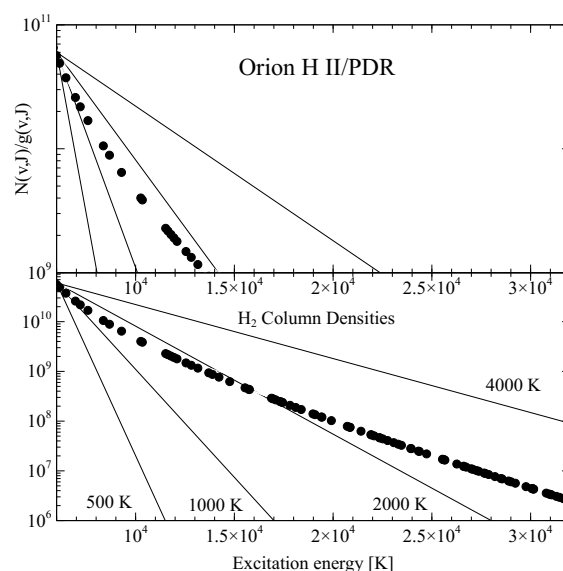
Figure 9: Predicted H_2 column densities are shown as a function of excitation energy for the Baldwin et al. (1991) Orion H II - PDR model.

To convert observations of H_2 lines into column densities you need to divide the observed line by a ratio of atomic constants.

REFERENCES

N. P. Abel, G. J. Ferland, G. Shaw, and P. A. M. van Hoof. The H II Region/PDR Connection: Self-consistent Calculations of Physical Conditions in Star-forming Regions. *ApJS*, 161:65–95, November 2005. doi: 10.1086/432913.

J. Baldwin, G. Ferland, K. Korista, and D. Verner. Locally Optimally Emitting Clouds and the Origin of Quasar Emission Lines. *ApJL*, 455:L119+, December 1995. doi: 10.1086/309827.

J. A. Baldwin, G. J. Ferland, P. G. Martin, M. R. Corbin, S. A. Cota, B. M. Peterson, and A. Slettebak. Physical conditions in the Orion Nebula and an assessment of its helium abundance. *ApJ*, 374: 580–609, June 1991. doi: 10.1086/170146.

M. Chatzikos, S. Bianchi, F. Camilloni, P. Chakraborty, C. M. Gunasekera, F. Guzmán, J. S. Milby, A. Sarkar, G. Shaw, P. A. M. van Hoof, and G. J. Ferland. The 2023 Release of Cloudy. *RMxAA*, 59: 327–343, October 2023a. doi: 10.22201/ia.01851101p.2023.59.02.12.

M. Chatzikos, S. Bianchi, F. Camilloni, P. Chakraborty, C. M. Gunasekera, F. Guzmán, J. S. Milby, A. Sarkar, G. Shaw, P. A. M. van Hoof, and G. J. Ferland. The 2023 Release of Cloudy. In *RMxAA* Chatzikos et al. (2023a), pages 327–343. doi: 10.22201/ia.01851101p.2023.59.02.12.

M. Chatzikos, S. Bianchi, F. Camilloni, P. Chakraborty, C. M. Gunasekera, F. Guzmán, J. S. Milby, A. Sarkar, G. Shaw, P. A. M. van Hoof, and G. J. Ferland. The 2023 Release of Cloudy. In *RMxAA* Chatzikos et al. (2023a), pages 327–343. doi: 10.22201/ia.01851101p.2023.59.02.12.

J. W. Ferguson, K. T. Korista, J. A. Baldwin, and G. J. Ferland. Locally Optimally Emitting Clouds and the Narrow Emission Lines in Seyfert Galaxies. *ApJ*, 487:122, September 1997. doi: 10.1086/304611.

G. Ferland. The Spectrum of a Single Photoionized Cloud. In G. Ferland and J. Baldwin, editors, *Quasars and Cosmology*, volume 162 of *Astronomical Society of the Pacific Conference Series*, page 147, 1999a.

G. Ferland. Reliability in the Face of Complexity - the Challenge of High-end Scientific Computing. In G. Ferland and D. W. Savin, editors, *Spectroscopic Challenges of Photoionized Plasmas*, volume 247 of *Astronomical Society of the Pacific Conference Series*, page 391, 2001.

G. Ferland. Molecular hydrogen in NLSys and its implications for the SED. In *Narrow-Line Seyfert 1 Galaxies and their Place in the Universe*, 2011.

G. J. Ferland. *Hazy, A Brief Introduction to Cloudy 80*. 1991.

G. J. Ferland. *Hazy, A Brief Introduction to Cloudy 84*. 1993.

G. J. Ferland. *Hazy, A Brief Introduction to Cloudy 90*. 1996.

G. J. Ferland. Hydrogen Emission from Low Column Density Gas: Case C. *PASP*, 111:1524–1528, December 1999b. doi: 10.1086/316466.

G. J. Ferland. Quantitative Spectroscopy of Photoionized Clouds. *ARA&A*, 41:517–554, 2003. doi: 10.1146/annurev.astro.41.011802.094836.

G. J. Ferland, K. T. Korista, D. A. Verner, J. W. Ferguson, J. B. Kingdon, and E. M. Verner. CLOUDY 90: Numerical Simulation of Plasmas and Their Spectra. *PASP*, 110:761–778, July 1998. doi: 10.1086/316190.

G. J. Ferland, R. L. Porter, P. A. M. van Hoof, R. J. R. Williams, N. P. Abel, M. L. Lykins, G. Shaw, W. J. Henney, and P. C. Stancil. The 2013 Release of Cloudy. *Revista Mexicana de Astronomia y Astrofisica*, 49:137–163, April 2013.

G. J. Ferland, M. Chatzikos, F. Guzmán, M. L. Lykins, P. A. M. van Hoof, R. J. R. Williams, N. P. Abel, N. R. Badnell, F. P. Keenan, R. L. Porter, and P. C. Stancil. The 2017 Release Cloudy. *Revista Mexicana de Astronomia y Astrofisica*, 53:385–438, October 2017.

C. Giammanco, J. E. Beckman, A. Zurita, and M. Relaño. Propagation of ionizing radiation in H II regions: The effects of optically thick density fluctuations. *A&A*, 424:877–885, September 2004. doi: 10.1051/0004-6361:20040468.

Chamani M. Gunasekera, Xihan Ji, Marios Chatzikos, Renbin Yan, and Gary Ferland. Self consistent grain depletions and abundances II: Effects on strong-line diagnostics of extragalactic H II regions. *MNRAS*, 520(3):4345–4355, April 2023. doi: 10.1093/mnras/stad322.

Chamani M. Gunasekera, Peter A. M. van Hoof, Marios Chatzikos, and Gary J. Ferland. Cloudy and the High-Resolution Microcalorimeter Revolution: Optical, UV, and X-ray Spectra of One-electron Systems. *arXiv e-prints*, art. arXiv:2412.01606, December 2024. doi: 10.48550/arXiv.2412.01606.

W. J. Henney, S. J. Arthur, R. J. R. Williams, and G. J. Ferland. Self-Consistent Dynamic Models of Steady Ionization Fronts. I. Weak-D and Weak-R Fronts. *ApJ*, 621:328–347, March 2005. doi: 10.1086/427491.

W. J. Henney, R. J. R. Williams, G. J. Ferland, G. Shaw, and C. R. O'Dell. Merged Ionization/Dissociation Fronts in Planetary Nebulae. *ApJL*, 671:L137–L140, December 2007. doi: 10.1086/525023.

J. Kingdon, G. J. Ferland, and W. A. Feibelman. Grains in ionized nebulae: Spectral line diagnostics. *ApJ*, 439:793–799, February 1995. doi: 10.1086/175217.

M. Moraes and M. Diaz. HR Del Remnant Anatomy Using Two-Dimensional Spectral Data and Three-Dimensional Photoionization Shell Models. *AJ*, 138:1541–1556, December 2009. doi: 10.1088/0004-6256/138/6/1541.

C. Morisset. Cloudy_3D, a new pseudo-3D photoionization code. In M. J. Barlow & R. H. Méndez, editor, *Planetary Nebulae in our Galaxy and Beyond*, volume 234 of *IAU Symposium*, pages 467–468,

2006. doi: 10.1017/S1743921306003772.

C. Morisset and G. Stasinska. An atlas of synthetic line profiles of Planetary Nebulae. *Revista Mexicana de Astronomia y Astrofisica*, 44:171–180, April 2008.

D. Osterbrock and E. Flather. Electron Densities in the Orion NEBULA.II. *ApJ*, 129:26, January 1959. doi: 10.1086/146592.

R. L. Porter, G. J. Ferland, S. B. Kraemer, B. K. Armentrout, K. A. Arnaud, and T. J. Turner. A Cloudy/XSPEC Interface. *PASP*, 118:920–923, June 2006. doi: 10.1086/506333.

G. Shaw, G. J. Ferland, N. P. Abel, P. C. Stancil, and P. A. M. van Hoof. Molecular Hydrogen in Star-forming Regions: Implementation of its Microphysics in CLOUDY. *ApJ*, 624:794–807, May 2005. doi: 10.1086/429215.

A   Veusz Cookbook

Most of the plots in this Guide were created with Jeremy Sanders' Veusz plotting package. Veusz is an open-source and runs on Windows, Mac OS X, and Linux. Veusz can be downloaded from https://veusz.github.io/. What follows are a few recipes for basic Cloudy plotting tasks that will get you started with Veusz. For more information, see the Veusz manual at https://veusz.github.io/help-support/. This document assumes that you are using Veusz 1.6 or later.

A.1   Steps to Import data

There is an important distinction in this step. If you create an "unlinked data set" Veusz will make your data part of the Veusz plot file. This is good if you want a single self-contained file. If you create a "linked data set" the plot will be linked to the original data file. If you update the data file the plot will be updated with no further work.

Veusz does not read spreadsheet files directly. If you work with your data in a spreadsheet you should save the data in a field-delimited format. CSV (comma separated values) or tabs are good choices.

Heads up: Data files created by Cloudy have a first line with column headings for the data that follow. These headings have spaces and special characters (/, + , -, ₋, etc.). If you encounter problems with Veusz not being able to open files with unlinked data sets, removing these characters from your data and re-importing may fix the problem.

1. Click "Data"

2. Select "Import"

3. Choose the file that contains your data clicking "Browse" and selecting the file

4. If the data is delimited, select the "CSV" tab.

5. Under the "File Preview" box select the correct delimiter

6. At the bottom, you have the option to link the dataset to the original file.

7. Click "Import" and each column of your data will be saved in a dataset designated by the column heading.

8. Click "Close"

A.2   Steps to plot your data (XY points)

1. Click "Insert"

2. Select "Add xy" : xy1 will be created in the top white box on the left (Editing Box)

3. The middle box (properties box) on the left side gives the properties of the xy1 pointset

4. In the properties box, click the blue down arrow to the right of "X data"

5. Select the dataset that you want to appear on the x-axis

6. Repeat steps 4 and 5 for the "Y data"

A.3   Steps to Export Plot to PDF

1. Click "File"

2. Select "Export"

3. Choose the file location with the "where" drop down.

4. Click the blue down arrow to the right of the "Files of Type" (at the bottom)

5. Choose "Portable Document Format (*.pdf)"

6. Choose the destination and file name then click "Save"

A.4   Change From Points and Line (default) to Line only

1. With the desired pointset selected in the Editing Box, select the "Main" tab in the Formatting Box (bottom on the left).

2. Click the blue down arrow on the "Marker" drop down box

3. Select "none"

A.5   Add x-axis title:

1. Select "X" (of the "axis" type) in the Editing Box

2. Enter the desired title for the x-axis in the "Label" textbox inside the Properties Box

A.6   Add y-axis title:

1. Select "Y" (of the "axis" type) in the Editing Box

2. Enter the desired title for the y-axis in the "Label" textbox inside the Properties Box

A.7   Add a second plot:

1. Repeat the Steps to plot your data (XY points)

A.8   Change Line Properties:

1. Select the desired pointset in the Editing Box

2. Click the second tab called "Plot line" in the Formatting Box

3. In this tab one can change the color, width, style, and transparency of the line for the selected pointset.

A.9   Adding a Key or Legend:

1. Click "Insert"

2. Select "Add Key" – This will create a blank key/legend box that can be dragged and dropped.

3. Select a pointset from the Editing Box

4. In the Properties Box, enter the text to appear in the key/legend box in the textbox called "Key text"

A.10   Create a 2D dataset (Needed for contour or image graphs):

1. Ensure that your individual data columns are already imported

2. Click "Data"

3. Select "Create 2D"

4. Type in a name for your 2D dataset in the "Name" textbox

5. Ensure that the middle option, "From x, y, and z . . . ", is selected in the "Method of Creating Dataset" Box.

6. In the "Values" section, choose the desired x, y, and z individual datasets

7. Click "Create" – It should tell you that the dataset was created at the bottom of the form.

8. Click "Close"

A.11   Create a Contour Plot:

1. Click "Insert"

2. Select "Add contour"

3. In the Properties Box, select the desired 2D dataset

Veusz automatically chooses contours. It is possible to change the number of contours by changing the "Number Levels" property. Changing the "Scaling" property changes the way that the contour levels are chosen. If the contours are chose automatically, it is important to specify the "min" and "max" contours.

A.12   Set Contour Levels Manually:

1. Ensure that the desired contour set is selected in the Editing Box.

2. In the Properties Box, change the "Scaling" property to "Manual"

3. Enter the desired contour levels in the "Manual Levels" textbox separated by commas.

A.13   Add contour labels:

1. Click on the 2nd tab in the Formatting Box, called "Contour labels"

2. Uncheck the "Hide" checkbox

3. The font, color, and size can also be adjusted here

A.14   A bit about contour plots:

With a contour pointset selected, the Formatting Box contains 5 tabs. The first tab, called "Main formatting", only allows you to hide the contours. The next tab over is called "Contour Labels". Here you can adjust the numerical labels for the contours.

The third tab from the left is called "Contour Lines". By default, there is one line style. This means that all of the contour lines will have the properties set by this line style. The "Add"button will add a new line style to the list. With 2 line styles, the top line style will be applied to the lowest contour line. The second line style will be applied to the next lowest contour. The next contour line will then take the properties of the top line style. Each successive contour line will continue to cycle between the two available line styles. The "Delete" button deletes the lowest line style on the list. The checkbox to the right of the line style properties hides the contour lines that are described by the given line style.

The fourth tab is called "Contour Fill". It works just like the "Contour Lines" tab except that it fills in the contour areas with different fillers (e.g. solid, vertical lines, etc.). It is also possible to adjust the color of the fillers.

The fifth tab is "Sub-contour Lines". This tab allows you to add a specific number of contour sections evenly spaced between the defined contour lines. The number of sections to add in between each defined contour line is set by the "Levels" textbox. The number of sub-contour lines that will appear between each defined contour is the number in the "Levels" textbox minus one. The "Line Style" section of this tab works the same way as the "Contour Lines" and "Contour Fill" tab. Note that you have to uncheck the "Hide" checkbox to view the sub-contours.

A.15   Filled contour maps

A variety of filled contour maps are available. Do insert / image and then select a 2D dataset. Several color and grey-scales are available.

If you want both the contour lines and the filled image then the contours must be on top of the filled contour.

Figure 10 shows an example, taken from the **grid_extreme** test in the slow test suite. This uses the spectrum2 colormap and goes from red to blue as the temperature goes from low to high values.

A.16   Multiple graphs

The graphs are made out of widgets that can be placed inside each other, e.g. graphs can be placed in pages, and different types of plotting widgets, e.g. xy or function, can be placed in graphs. You can also place graphs within a grid widget to get an arrangement of separate graphs.

Figure 11 shows an example, taken from the Orion HII region / PDR simulation, of a nested plot.

A.17   Documenting your work

The Properties widget includes a Notes field that can be used to keep a history of the data behind a plot. This is useful so that you can come back to a project after some time and recover the history of the work leading up to a result or the location of the helper files that were used along the way.
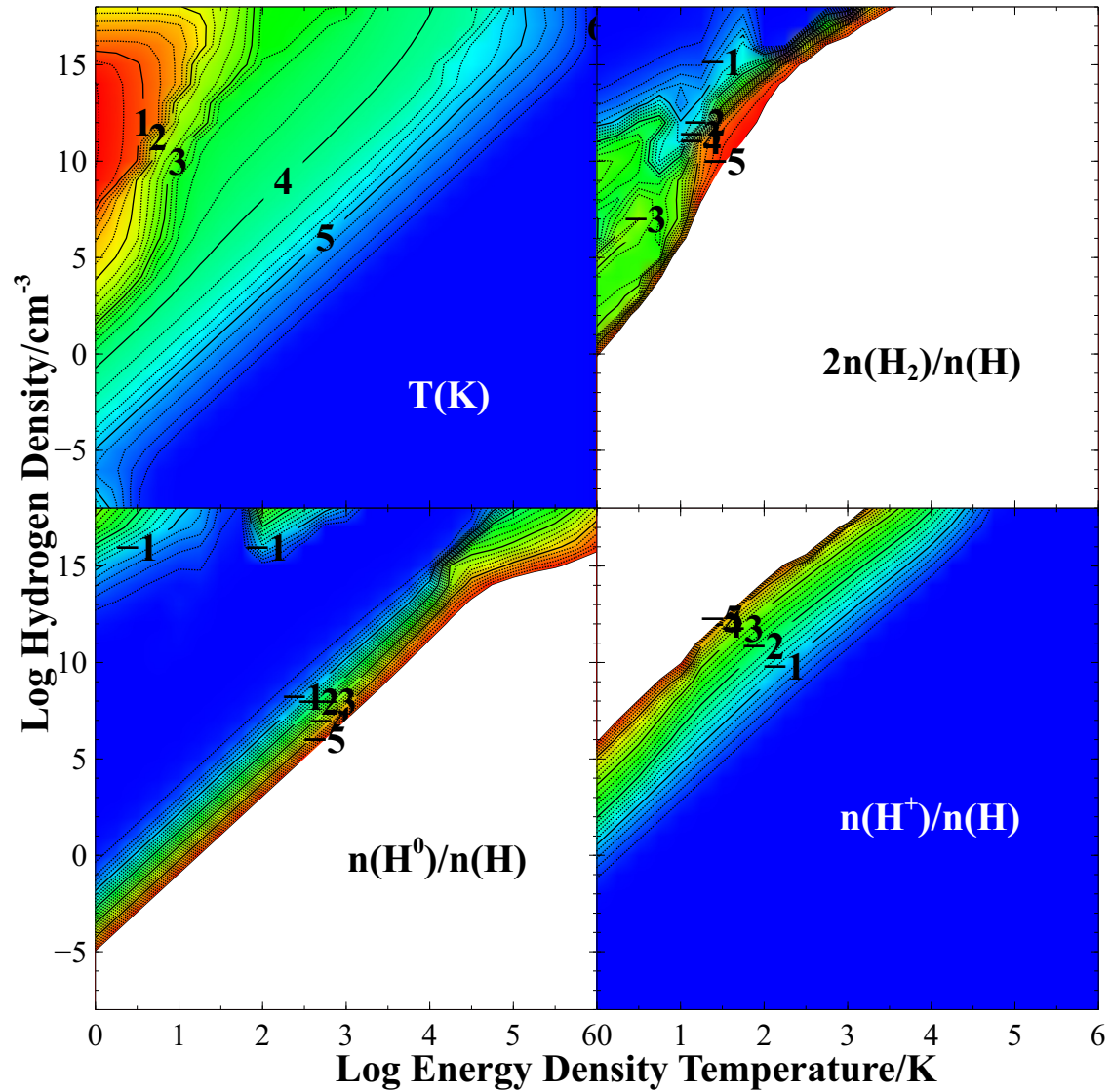
Figure 10: This shows physical conditions over a very broad range of hydrogen density and energy-density temperature. The log of the electron kinetic temperature is shown in the upper left, and hydrogen ionization and molecular fractions in the remaining panels. The gas goes to the Compton temperature in the upper left corner of each panel, is in the blackbody limit across the top, and is close to LTE across the right-hand edges of the figure.
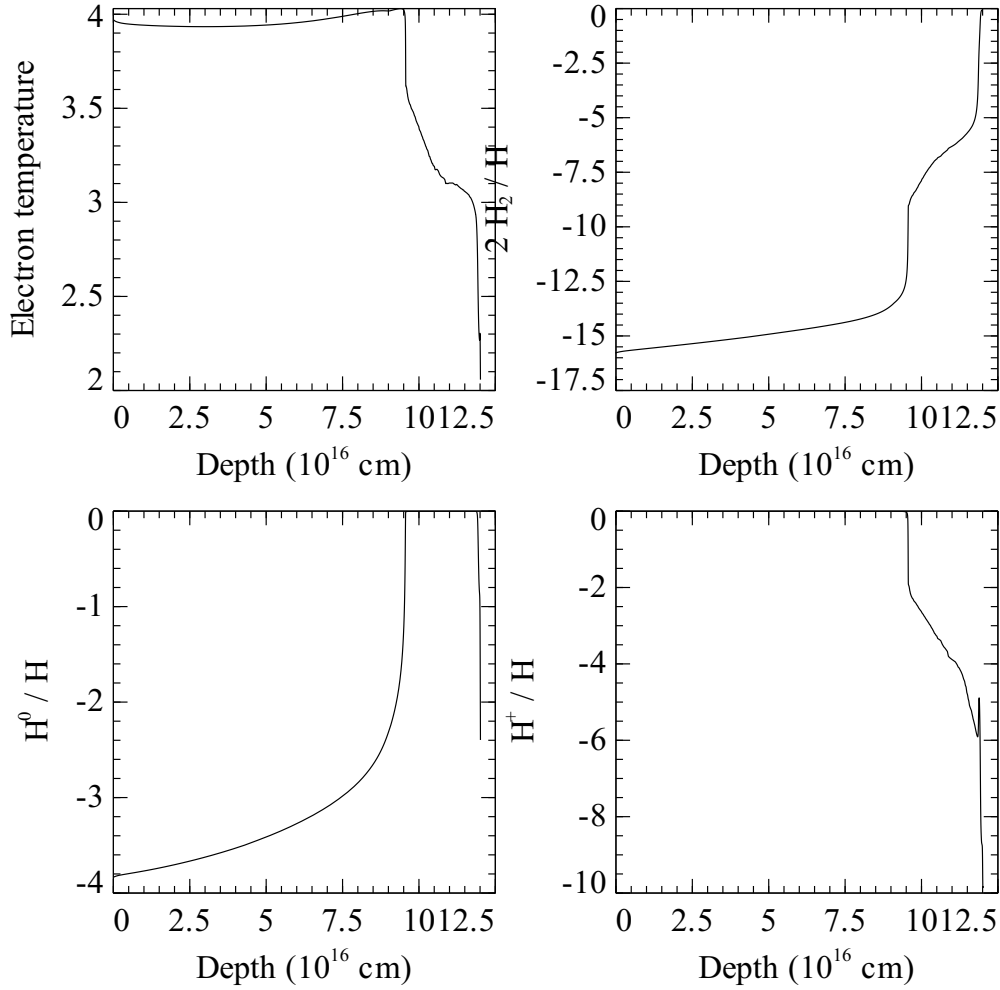
Figure 11: The electron kinetic temperature, and fractions of H in H_2, H^0, and H^+, are shown as a function of depth into the Orion H II region / PDR.