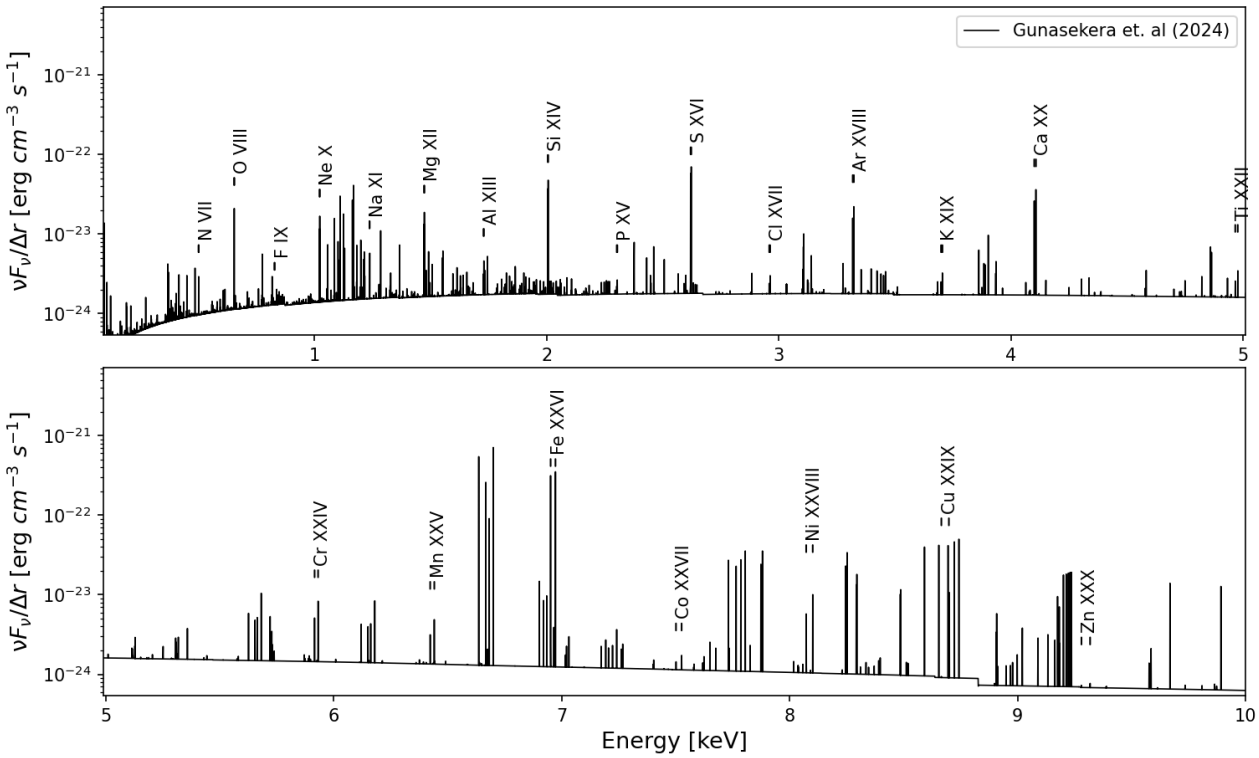

Hazy

a brief introduction to CLOUDY *C25*

1. Introduction and commands

Cloudy & Associates

www.nublado.org

April 2, 2025

Software: Copyright © 1978–2025 Gary J. Ferland and others. All rights reserved.

The software described in this documentation (CLOUDY) is subject to a FreeBSD-style software license contained in the file `license.txt` in the root directory of the distributed files. The list of co-authors is given in the file `others.txt` in the same directory. Use of this program is not restricted provided each use is acknowledged upon publication. The bibliographic reference to this version of CLOUDY is "version xx.xx of the code last described in Chatzikos et al. (2023)." The version number, shown here as "xx.xx", should be given. This version number, along with a complete citation, can be found by executing the code with the **print citation** command included in the input script.

CLOUDY is an evolving code. You should confirm that you have the most recent version of the code by checking the web site www.nublado.org. The code has a discussion board with emailing list. This will have announcements of any updates to the code.

Portions of the documentation have been published, and are copyrighted by, the American Astronomical Society, the Astronomical Society of the Pacific, and the Royal Astronomical Society. The remainder of the documentation is subject to the following FreeBSD format documentation license:

Documentation: Copyright © 1978–2025 Gary J. Ferland and others. All rights reserved.

Redistribution and use of the documentation (all parts of HAZY and the Quick Start Guide) in source (LaTeX) and 'compiled' forms (PDF, PostScript, HTML and so forth) with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code (LaTeX) must retain the above copyright notice, this list of conditions and the following disclaimer as the first lines of the file `license_doc.txt` in the root directory unmodified.

2. Redistributions in compiled form (converted to PDF, PostScript, XML, HTML and other formats) must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

This documentation is provided by the CLOUDY *project "as is" and any express or implied warranties, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose are disclaimed. In no event shall the* CLOUDY *project be liable for any direct, indirect, incidental, special, exemplary, or consequential damages (including, but not limited to, procurement of substitute goods or services; loss of use, data, or profits; or business interruption) however caused and on any theory of liability, whether in contract, strict liability, or tort (including negligence or otherwise) arising in any way out of the use of this documentation, even if advised of the possibility of such damage.*

Cover image: Cloudy's simulation of the outer region of the Perseus galaxy cluster. The figure shows resolved fine-structure one-electron doublets such as Fe XXVI. A spectral resolution of $R = 0.25$ eV and a log column density of 24 were used. J-splitting of the one-electron nP$-$1S lines is a major improvement in this release of Cloudy, as described in Gunasekera+2024.

CONTENTS

LIST OF FIGURES xxv

LIST OF TABLES xxvii

1 INTRODUCTION 1
- 1.1 Overview . . . 1
- 1.2 What must be specified . . . 1
  - 1.2.1 Incident radiation field . . . 2
  - 1.2.2 Chemical Composition . . . 2
  - 1.2.3 Geometry . . . 3
  - 1.2.4 Velocity Structure . . . 3
- 1.3 The input deck . . . 3
- 1.4 What is computed and reported . . . 4

2 DEFINITIONS 7
- 2.1 Overview . . . 7
- 2.2 Radiation fields . . . 7
  - 2.2.1 Incident radiation field . . . 7
  - 2.2.2 Diffuse radiation field . . . 7
  - 2.2.3 Transmitted radiation field . . . 8
  - 2.2.4 Reflected radiation field . . . 8
- 2.3 Geometry . . . 8
  - 2.3.1 Illuminated and shielded faces of the cloud . . . 10
  - 2.3.2 Depth and radius . . . 10
  - 2.3.3 Covering factor . . . 10
  - 2.3.4 Open vs. closed geometry . . . 10
  - 2.3.5 Filling factor . . . 12
  - 2.3.6 Hydrogen density . . . 13
  - 2.3.7 Column densities . . . 13
  - 2.3.8 Matter-bounded and radiation-bounded geometries . . . 13
- 2.4 Intensity & luminosity cases . . . 13
  - 2.4.1 Luminosity case . . . 14
  - 2.4.2 Intensity case . . . 14
  - 2.4.3 The units of the predicted emission lines . . . 14
- 2.5 “Species”, how we specify atoms, ions, molecules, and their spectra . . . 14

2.5.1 Overview . . . 14
2.5.2 How we specify species and spectra . . . 15
2.5.3 How we specify spectral lines . . . 16
2.5.4 What species are you using? . . . 17
2.6 Air vs vacuum wavelengths . . . 18

3 INTRODUCTION TO COMMANDS 19
3.1 Overview . . . 19
3.2 Command format . . . 19
3.2.1 Input and Output . . . 19
3.2.2 Command-line format. . . . 20
3.2.3 Units . . . 21
3.2.4 Number of commands. . . . 21
3.2.5 Output as input. . . . 21
3.2.6 Syntax used in this document . . . 21
3.2.7 and, because nobody ever reads this document. 22
3.2.8 The continue option . . . 22
3.2.9 Numerical input . . . 22
3.2.10 Comments . . . 23
3.2.11 Hidden commands . . . 23
3.2.12 Commands for experimental parts of the code . . . 24
3.2.13 Log vs linear quantities . . . 24
3.2.14 The **help** command . . . 24
3.2.15 An example . . . 24
3.3 Reading and Writing Files . . . 25
3.4 The Search Path for Input Files . . . 26
3.5 The **init** command . . . 27
3.6 Random numbers . . . 28

4 THE INCIDENT RADIATION FIELD 29
4.1 Overview . . . 29
4.2 The coarse and fine continua . . . 29
4.3 Defining a single component of the incident radiation field . . . 30
4.4 Combining several radiation fields . . . 31
4.4.1 The sum of several radiation fields . . . 31
4.4.2 Keeping shape and intensity commands together . . . 31
4.5 Check the incident radiation field! . . . 32

5 INCIDENT RADIATION FIELD LUMINOSITY 33
5.1 Overview . . . 33
5.2 Intensity and luminosity cases . . . 33
5.3 The range option . . . 33
5.4 Absolute [visual, bolometric] magnitude -2.3 . . . 34
5.5 Energy density 5e4 K [linear] . . . 35
5.6 f(nu) = -12.456 [at .1824 Ryd] . . . 35

5.7 Intensity 8.3 [range, linear] . . . 35
5.8 ionization parameter = −1.984 . . . 36
5.9 L(nu) = 24.456 [at .1824 Ryd] . . . 36
5.10 luminosity 38.3 [solar, total, range, linear] . . . 37
5.11 nuF(nu) = 13.456 [at .1824 Ryd] . . . 37
5.12 nuL(nu) = 43.456 [at .1824 Ryd] . . . 38
5.13 phi(H) = 12.867 [range...] . . . 38
5.14 Q(H) = 56.789 [range...] . . . 38
5.15 ratio -3.4 0.3645 Ryd to 147 Ryd [alphaox, log] . . . 38
5.16 xi -0.1 . . . 40

6 INCIDENT RADIATION FIELD SHAPE 41
6.1 Overview . . . 41
6.1.1 Isotropic and beamed continua . . . 41
6.2 AGN T =1.5e5 k, a(ox) = -1.4, a(uv)=-0.5 a(x)=-1 . . . 42
6.3 Background, z=1.825, [f=100; no CMB] . . . 44
6.4 Blackbody t=e5 K [linear, log, luminosity] . . . 44
6.4.1 Blackbody, t = 1e5 K, disk = 1e3 K . . . 45
6.4.2 Peter Martin's blackbody luminosity options . . . 45
6.4.3 Blackbody 5, luminosity = 38. . . . 45
6.4.4 Blackbody 5, radius = 10. . . . 45
6.4.5 Blackbody 5e4 K, energy density = 500K. . . . 45
6.4.6 Blackbody, t = 5e4 K, STE . . . 46
6.4.7 Blackbody, t = 1e5 K, dilution factor = -14 . . . 46
6.5 Bremsstrahlung, temp = 8 . . . 46
6.6 CMB [redshift 1000] . . . 47
6.6.1 The time option . . . 47
6.6.2 The density option . . . 47
6.7 Extinguish column = 23, leak = 0.05, low =4 Ryd . . . 47
6.7.1 Extinguish optical depth 1.2, [options] . . . 48
6.8 Interpolate [ν(Ryd) or log ν(Hz)], log(f_ν) . . . 48
6.9 Laser, frequency = 3.5 Ryd [rel width 0.02] . . . 49
6.10 Power law, slope =-1.4 [hi cut =6 Ryd low cut =.1, Kelvin] . . . 50
6.11 Table SED command . . . 50
6.11.1 SED file format . . . 51
6.11.2 Optional keywords within the SED data file . . . 51
6.11.3 Available SED datasets . . . 52
6.11.4 Table SED "akn120.sed" . . . 52
6.11.5 Table SED "cool.sed" . . . 53
6.11.6 Table SED "CrabHester.sed"/"CrabDavidson.sed" . . . 53
6.11.7 Table SED "Mrk509.sed" . . . 53
6.11.8 Table SED "NGC5548.sed" . . . 53
6.11.9 Table SED "pl1.sed" . . . 53
6.11.10 Table SED "Rubin.sed" . . . 53
6.11.11 Table SED "XDR.sed" . . . 55

6.12 Other Table commands . . . 55
6.12.1 Overview . . . 55
6.12.2 Table AGN . . . 55
6.12.3 Table Draine [factor=1.7] . . . 56
6.12.4 Table HM96 [factor=-1] . . . 56
6.12.5 Table HM05 redshift 2.21 [quasar; factor=-1] . . . 58
6.12.6 Table HM12 redshift 2.21 [factor=-1] . . . 58
6.12.7 Table ism [factor = 0.7] . . . 58
6.12.8 Table KS19 redshift 2.21 [factor=-1] [Q=18] . . . 59
6.12.9 Table power law [spectral index -1.4, low =.01, hi =20] . . . 59
6.12.10 Table read "contin.txt" [scale [= 0.5]] . . . 60
6.12.11 Table trapezium . . . 61
6.13 Table stars command . . . 62
6.13.1 Overview of available stellar SEDs . . . 63
6.13.2 High-energy component . . . 63
6.13.3 Starburst99 files . . . 64
6.13.4 PopStar and BPASS grids . . . 65

7 CHEMICAL COMPOSITION 67
7.1 Overview . . . 67
7.2 Absolute abundances and scale factors . . . 67
7.3 Precedence . . . 69
7.4 Abundances he c. 69
7.4.1 Arbitrary abundances . . . 70
7.4.2 Setting all at once . . . 70
7.4.3 Abundance "filename.abn" – using tables of abundances . . . 70
7.4.4 Older solar abundances . . . 72
7.4.5 Grains, gas-phase depletions, and quantum heating . . . 72
7.4.6 Interactions between the abundances and grains commands . . . 75
7.5 Abundances starburst, Z=10 . . . 75
7.6 Abundances isotopes . . . 75
7.7 Element name [scale, abundance, isotopes, off, log, table] . . . 77
7.7.1 Element name scale . . . 77
7.7.2 Element name abundance . . . 77
7.7.3 Element name isotopes . . . 77
Setting the D/H ratio . . . 78
Setting the 12C/13C ratio . . . 78
7.7.4 Element isotopes all . . . 78
7.7.5 Element name ionization . . . 78
7.7.6 Element name off . . . 79
7.7.7 Element limit off -12 . . . 79
7.7.8 Element name table . . . 80
7.8 Fluctuations abundances, period, max, min, phase . . . 80
7.9 Grains . . . 80
7.9.1 Using the built-in grain types . . . 82

7.9.2 Interactions between abundances and grains commands . . . 84
7.9.3 Creating your own grain types . . . 85
7.9.4 PAHs . . . 85
7.9.5 Variable grain abundances . . . 86
7.9.6 Line intensities with grains . . . 87
7.9.7 Extinction for point and extended sources . . . 88
7.9.8 Grains-related commands . . . 88
7.9.9 Effects of grains . . . 89
7.9.10 Grain-related quantities that are printed . . . 90
7.10 Metals 0.05 [log, linear, grains; deplete] . . . 90
7.10.1 Scaling grains and metals together . . . 91
7.11 Metals deplete commands . . . 91
7.11.1 The depletion factors, reference abundance, and grains . . . 91
7.11.2 External data files . . . 92
7.11.3 The print option . . . 92
7.11.4 metals deplete - specify a table of depletion scale factors . . . 92
7.11.5 metals deplete Jenkins 2009 Fstar=0.5 . . . 94
Command syntax . . . 94
Other details . . . 94

8 DENSITY LAWS 97
8.1 Overview . . . 97
8.2 Constant density, pressure, gas pressure . . . 97
8.2.1 Constant density . . . 97
8.2.2 Constant gas pressure [index = −1.1] . . . 98
8.2.3 Constant pressure [no continuum, no abort] . . . 98
8.2.4 Constant pressure set 6.5 . . . 99
8.2.5 Constant pressure timescale = 2.3e9 s, alpha=-1 . . . 99
8.2.6 The reset option . . . 99
8.3 Gravity [options] . . . 99
8.3.1 Gravity [spherical, plane-parallel] . . . 99
8.3.2 Gravity external [Mass = 10 $M_\odot$, extent = 3 parsecs, powerlaw = -2] . . . 100
8.4 Dark [options] . . . 100
8.4.1 Dark NFW virial radius = 18 [characteristic radius = 17] . . . 100
8.5 Dlaw [options] . . . 100
8.5.1 dlaw p1, p2, p3... . . . 101
8.5.2 Dlaw wind . . . 101
8.5.3 Dlaw table [depth, radius] . . . 102
8.6 Fluctuations density period 102
8.7 Globule [density =2, depth =16, power =2] . . . 103
8.8 hden 5.6, [proportional to R -2,...] . . . 103
8.8.1 Power-law dependence on radius . . . 103
8.8.2 Clouds extending to infinity . . . 104
8.8.3 Power-law dependence on depth . . . 104
8.8.4 Power-law dependence on column density . . . 104

9 GEOMETRY **107**
9.1 Overview . . . 107
9.2 Age 44 years [off] . . . 107
9.3 Aperture commands . . . 107
9.3.1 Aperture [slit, beam] . . . 107
9.3.2 Aperture size 2.5 . . . 109
9.3.3 Aperture covering factor 0.6 . . . 109
9.4 Clumping factor = 0.05 [index =-1] . . . 109
9.5 Covering factor 0.3 . . . 110
9.6 Cylinder log semi height =9.12 . . . 110
9.7 Distance 3.2 linear parsecs . . . 111
9.8 Filling factor = 0.05 [index =-1] . . . 111
9.9 Illumination isotropic | angle 45 deg [radian] . . . 112
9.10 Radius r(inner) [r(outer), thickness; parsec; linear] . . . 112
9.11 Sphere [options] . . . 113
9.11.1 Sphere expanding or static . . . 113
9.12 Stop depth. 114
9.13 Stop thickness. 114

10 OPTICAL DEPTHS AND RADIATIVE TRANSFER **115**
10.1 Overview . . . 115
10.2 Case A [options] . . . 115
10.3 Case B [tau ly alpha = 9; options] . . . 116
10.4 Case C [options] . . . 117
10.5 Diffuse fields [outward, OTS] . . . 117
10.6 Double optical depths . . . 117
10.7 Iterate [2 times, to convergence] . . . 118
10.7.1 Number of iterations . . . 118
10.7.2 Iterate to convergence [max =7, error =.05, all] . . . 118
10.7.3 Convergence problems . . . 119
10.8 No scattering opacity . . . 119
10.9 Turbulence = 100 km/s [log, dissipate] . . . 120
10.9.1 Definitions . . . 120
10.9.2 Turbulent pressure . . . 120
10.9.3 Energy dissipation . . . 121
10.9.4 Equipartition turbulent - magnetic pressures . . . 121
10.9.5 Turbulence command heads up! . . . 122
10.10 vlaw alpha=-1 . . . 122

11 THERMAL SOLUTIONS **123**
11.1 Overview . . . 123
11.2 cextra -14.231 [temp to the 1.5 power] . . . 123
11.3 Constant temperature, t=1e4 K [linear] . . . 123
11.4 Constant grain temperature 20K [linear] . . . 124
11.5 Coronal equilibrium, T=1e7 K [linear] . . . 124

11.6 Cosmic rays [options...] . . . 126
11.6.1 Cosmic rays background [1.2] . . . 126
11.6.2 Cosmic ray rate -17.3 . . . 127
11.6.3 Cosmic rays density =1.2 [index, etc.] . . . 127
11.6.4 Cosmic ray equipartition . . . 127
11.6.5 The CR background and low-density gas . . . 128
11.7 Failures 100 times [map] . . . 129
11.8 Force temperature to 3400 K . . . 129
11.9 Hextra -14 [depth, density, time, SS] . . . 129
11.9.1 Hextra -14 depth 18, thickness 12 . . . 130
11.9.2 Hextra -13 density . . . 130
11.9.3 Hextra time . . . 130
11.9.4 Hextra SS 0.2 1e8 3e17 . . . 130
11.9.5 Notes on the hextra command . . . 131
11.10 High temperature approach . . . 131
11.11 Magnetic field, log(B) = 5 [options] . . . 131
11.12 Map, zone 4 [range 2000, 5000] . . . 132
11.13 Neutrons -2 [efficiency =-2] . . . 132
11.14 Print coolants, zone 135 . . . 133
11.15 Print heating . . . 133
11.16 Set temperature [floor, convergence] . . . 133
11.17 tlaw [DB96, SN99] . . . 133
11.17.1 Tlaw table [depth, radius] . . . 133

12 CONTROLLING ATOMIC AND MOLECULAR MODELS 135
12.1 Species overview . . . 135
12.1.1 How many levels do we include? . . . 135
12.1.2 Species output options . . . 136
Database print . . . 136
Save species . . . 136
Save lines labels . . . 136
12.1.3 The g-bar approximation . . . 136
12.1.4 Generating atomic data bibliography . . . 136
12.2 Species “name” [options] . . . 137
12.2.1 Species “name” off . . . 137
12.2.2 Species “name” levels=[10,all] . . . 137
12.2.3 Species “name” dataset=“abc” . . . 137
12.2.4 Species “name” lte . . . 137
12.3 Database H2 options . . . 138
12.3.1 The **set H2** command . . . 138
12.3.2 H_2 output options . . . 138
12.3.3 Database H2 . . . 138
12.3.4 Database H2 chemistry [simple; full] . . . 138
12.3.5 Database H2 collisions [options] . . . 138
12.3.6 Database H2 gbar [off; on] . . . 139

12.3.7 Database H2 levels . . . 139
12.3.8 Database H2 limit -4 . . . 139
12.3.9 Database H2 matrix 43 . . . 140
12.3.10 Database H2 noise [mean, standard deviation] . . . 140
12.3.11 Database H2 thermal [simple; full] . . . 140
12.3.12 Database H2 trace [options] . . . 140
12.3.13 database H2 output options . . . 140
12.4 Database H-like | He-like [element, options] . . . 141
12.4.1 Introduction . . . 141
12.4.2 Structure of the model atoms . . . 141
The H-like sequence . . . 142
The He-like sequence . . . 142
Selecting collisional processes . . . 143
12.4.3 Database H-like collisions. 143
Selecting collisional processes . . . 144
Selecting *n*-changing collision theories . . . 144
Selecting *l*-changing collision theories . . . 144
12.4.4 Database He-like collisions 145
Selecting collisional processes . . . 145
Selecting *n*-changing collision theories . . . 145
Selecting *l*-changing collision theories . . . 146
12.4.5 Database H-like | He-like continuum lowering off . . . 146
12.4.6 Database H-like damping off . . . 146
12.4.7 Database He-like dielectronic recombination [off] . . . 147
12.4.8 Database H-like | He-like [element] masers off . . . 147
12.4.9 Database H-like | He-like error generation [pessimistic] . . . 147
12.4.10 Database He-like gbar options . . . 147
12.4.11 Database H-like | He-like levels options . . . 147
Database H-like | He-like levels resolved | collapsed 4 [element iron] . 147
Database H-like | He-like levels LTE . . . 148
Database H-like | He-like levels print . . . 148
Setting the number of levels with keywords . . . 148
12.4.12 Database H-like | He-like Lyman options . . . 149
Database H-like | He-like Lyman extra 1000 . . . 149
Database H-like Lyman extra resolution 0.25 . . . 149
Database H-like Lyman pump . . . 149
Database H-like Lyman pumping off . . . 150
Database H-like Lyman pumping scale 3 . . . 150
12.4.13 Database H-like | He-like NO RECOmbination INTErp . . . 150
12.4.14 Database H-like | He-like redistribution [options] . . . 151
12.4.15 Database H-like keep fine structure [off] . . . 151
12.4.16 Database H-like | He-like topoff [off] . . . 152
12.5 Overview to the Chianti, Lamda, and Stout databases . . . 152
12.6 Database Lamda [on , off][print][levels {j, MAX}] . . . 153
12.6.1 Database Lamda [on off] . . . 153

12.6.2 Database Lamda "masterlist.ini" . . . 153
12.6.3 Database Lamda print . . . 153
12.6.4 Database Lamda Levels n . . . 153
12.7 Database Chianti [options] . . . 154
12.7.1 Database Chianti "masterlist.ini" . . . 154
12.7.2 Database Chianti off . . . 154
12.7.3 Database Chianti [hybrid, no hybrid] . . . 154
12.7.4 Database Chianti print . . . 155
12.7.5 Database Chianti Levels . . . 155
12.7.6 Database Chianti [experimental theoretical] . . . 155
12.8 Database Stout [on , off, no hybrid][print][levels {j, MAX}] . . . 155
12.8.1 Database Stout [on off] . . . 155
12.8.2 Database Stout "masterlist.ini" . . . 155
12.8.3 Database Stout hybrid . . . 156
12.8.4 Database Stout print . . . 156
12.8.5 Database Stout Levels nFe, nOther . . . 156

13 CONTROLLING CHEMICAL MODELS 157
13.1 The Chemistry command . . . 157
13.1.1 Chemistry reaction . . . 157

14 DYNAMICAL & TIME-DEPENDENT CALCULATIONS 159
14.1 Overview . . . 159
14.2 cosmology . . . 159
14.2.1 cosmology omega . . . 160
14.2.2 cosmology Hubble . . . 160
14.3 A cooling or heating non-equilibrium gas . . . 160
14.4 Time-variable incident radiation field . . . 162
14.4.1 time command . . . 162
14.4.2 Time-step size increment . . . 163
14.4.3 Keywords on individual time-step entries . . . 164
14.4.4 Varying individual continua . . . 164
14.4.5 Controlling the number of time steps . . . 164
Stop temperature time [exceeds] . . . 164
14.4.6 Other commands . . . 164
14.5 Wind velocity=300 km/sec [mass =1.4 Msun, disk, ballistic, static] . . . 165
14.5.1 Ballistic solutions . . . 165
14.5.2 wind advection, velocity=-5 km/s . . . 166
14.5.3 Wind 5 km/s no continuum . . . 167
14.6 Miscellaneous dynamics commands . . . 167
14.6.1 Constant pressure time . . . 167
14.6.2 Set dynamics [options] . . . 167
14.7 Useful output options . . . 168
14.7.1 Print cumulative lines . . . 168
14.7.2 Save commands, general comments . . . 168

14.7.3 Save cumulative continuum . . . 168
14.7.4 Save time dependent . . . 168
14.7.5 Set cumulative . . . 168
14.7.6 Set save hash options . . . 168

15 STOPPING CRITERIA 169
15.1 Overview . . . 169
15.2 Danger! Understand why the calculation stopped! . . . 169
15.3 Radius inner =18 [thickness =16; parsec; linear] . . . 170
15.4 Stop Av 12.1 [point, extended] . . . 170
15.5 Stop column density = 23 [neutral; ionized; total; . . .] . . . 171
15.5.1 Stop column density 23 . . . 171
15.5.2 Stop neutral column density 23 . . . 171
15.5.3 Stop ionized column density 23 . . . 171
15.5.4 Stop atomic column density 21.3 . . . 171
15.5.5 Stop H/TSpin column density 17.2 . . . 172
15.5.6 Stop H2 column density 19.2 . . . 172
15.5.7 Stop CO column density 19.2 . . . 172
15.5.8 Stop column density “OH” 19.2 . . . 172
15.5.9 Stop effective column density 23 . . . 172
15.6 Stop continuum flux 200 micron 65 Jansky . . . 173
15.7 Stop depth 173
15.8 Stop eden 3 [linear] . . . 174
15.9 Stop efrac = 1.05 . . . 174
15.10 Stop line ”C 2” 157.636m 0.2 rel to ”O 3” 5006.84 . . . 174
15.11 Stop mass 32.98 . . . 175
15.12 Stop mfrac = 0.5 . . . 175
15.13 Stop molecule depletion -2 . . . 175
15.14 Stop optical depth -1 at 2.3 Ryd . . . 176
15.14.1 Stop Balmer optical depth = −.3 . . . 176
15.14.2 Stop Lyman optical depth = 5 . . . 176
15.14.3 Stop 21cm optical depth = linear 3 . . . 177
15.15 Stop pfrac = 0.23 . . . 177
15.16 Stop radius 17.3 [parsec; linear; 17.6 on sec iter] . . . 177
15.17 Stop temperature 1e3 K [linear, exceeds] . . . 178
15.18 Stop thickness 9.3 [parsec; linear; 23 on sec iter] . . . 178
15.19 Stop velocity < 1 km/s . . . 178
15.20 Stop zone 123 [21 on sec iteration,. . .] . . . 178

16 CONTROLLING OUTPUT 181
16.1 Overview . . . 181
16.2 No buffering . . . 181
16.3 Normalize to ”O 3” 5006.84 [scale factor = 100] . . . 181
16.4 Print ages . . . 182
16.5 Print arrays . . . 182

16.5.1 Print arrays ionization . . . 182
16.5.2 Print arrays levels "species" . . . 182
16.6 Print citation . . . 183
16.7 Print constants . . . 183
16.8 Print column densities [on; off] . . . 183
16.9 Print coolants, zone 135 . . . 183
16.10 Print continuum indices . . . 183
16.11 Print departure coefficients [He-like] [element] . . . 184
16.12 Print critical densities [He-like] [element] . . . 184
16.13 Print errors . . . 184
16.14 Print fixits . . . 184
16.15 Print every 1000 [5 37 93] . . . 185
16.16 Print heating . . . 185
16.17 Print populations [H-like carbon, to level 45] . . . 185
16.18 Print last . . . 185
16.19 Print line options . . . 185
16.19.1 Print line all . . . 186
16.19.2 Print line cell xx . . . 186
16.19.3 Print line collisions . . . 186
16.19.4 Print line column [linear] . . . 186
16.19.5 Print line cumulative . . . 186
16.19.6 Print line faint -2 [_off] . . . 186
16.19.7 Print line flux at Earth . . . 187
16.19.8 Print line heat . . . 187
16.19.9 Print line H2 electronic . . . 187
16.19.10 Print line inward . . . 187
16.19.11 Print line iso collapsed off . . . 187
16.19.12 Print line [intrinsic, emergent] off . . . 187
16.19.13 Print line optical depths [_off, faint] . . . 188
16.19.14 Print line precision [4, 5, or 6] . . . 188
16.19.15 Print line pump . . . 188
16.19.16 Print line sort wavelength [range 3500A to 1.2m] . . . 188
16.19.17 Print line sort intensity . . . 189
16.19.18 Print line sum . . . 189
16.19.19 Print line surface brightness [arcsec] . . . 189
16.19.20 Print line vacuum . . . 189
16.20 Print macros . . . 190
16.21 Print modules . . . 190
16.22 Print off . . . 190
16.23 Print on . . . 190
16.24 Print only [header, zones] . . . 191
16.25 Print path . . . 191
16.26 Print quiet . . . 191
16.27 Print recombination . . . 191
16.28 Print short . . . 191

16.29 Print starting at 61 . . . 191
16.30 Print UTA references . . . 191
16.31 Print version . . . 192
16.32 Print zone 1000 [5 37 93] . . . 192
16.33 Save commands . . . 192
16.33.1 Overview . . . 192
16.33.2 Save vs punch commands . . . 192
16.33.3 An output file name must appear inside double quotes . . . 192
16.33.4 Setting a prefix for all save files . . . 193
16.33.5 The “last iteration” option . . . 193
16.33.6 The “no buffering” option . . . 193
16.33.7 The “clobber/no clobber” option . . . 193
16.33.8 The “no hash” option . . . 193
16.33.9 The “title” option . . . 194
16.33.10 The “separate” option . . . 194
16.33.11 Log vs linear output quantities . . . 194
16.33.12 Depth versus radius . . . 194
16.34 Save abundances . . . 194
16.35 Save ages . . . 195
16.36 Save agn [options] . . . 195
16.37 Save arrays levels ”species” [fits or ppm] [iteration it] [zone nz] . . . 195
16.38 Save monitors . . . 195
16.39 Save average 196
16.39.1 Temperature average . . . 196
16.39.2 Ionization average . . . 196
16.39.3 Column density . . . 196
16.40 Save chemistry options . . . 197
16.40.1 Save chemistry rates ”filename” species ”molecule” [coef] . . . 197
16.41 Save column density . . . 197
16.42 Save continuum . . . 197
16.42.1 Lines in the spectrum . . . 197
16.42.2 Emission line – continuum contrast . . . 197
16.42.3 Pumped contributions to the lines . . . 199
16.42.4 Isotropic Continua . . . 200
16.42.5 The units option - changing the continuum units . . . 200
16.42.6 Save continuum uses vacuum wavelengths . . . 200
16.42.7 Units of the save output in intensity and luminosity cases . . . 200
16.42.8 Save continuum . . . 200
16.42.9 What is observed . . . 202
16.42.10 Save continuum bins . . . 203
16.42.11 Save cumulative continuum . . . 203
16.42.12 Save diffuse continuum . . . 203
16.42.13 Save continuum emissivity 12 [units micron] . . . 203
16.42.14 Save emitted continuum . . . 204
16.42.15 Save fine continuum [range, merge] . . . 204

16.42.16 Save grain continuum . . . 204
16.42.17 Save incident continuum . . . 205
16.42.18 Save interactive continuum . . . 205
16.42.19 Save ionizing continuum [options] . . . 205
16.42.20 Save raw continuum . . . 206
16.42.21 Save reflected continuum . . . 206
16.42.22 Save transmitted continuum [no isotropic] . . . 206
16.42.23 Save two photon continuum . . . 207
16.43 Save convergence [reason, error] . . . 207
16.43.1 Save convergence base . . . 207
16.43.2 Save convergence error . . . 207
16.43.3 Save convergence reason . . . 207
16.44 Save cooling . . . 207
16.44.1 Save cooling each . . . 208
16.45 Save charge transfer . . . 208
16.46 Save Chianti . . . 208
16.47 Save dr . . . 209
16.48 Save dt . . . 209
16.49 Save dynamics options . . . 209
16.49.1 Save dynamics advection . . . 209
16.50 Save element *name* . . . 209
16.51 Save enthalpy . . . 209
16.52 Save execution times . . . 209
16.53 Save FeII . . . 210
16.54 Save FITS . . . 210
16.55 Save gammas . . . 210
16.55.1 Save gammas element oxygen 1 . . . 210
16.56 Save Gaunt factors . . . 210
16.57 Save grains [options] . . . 211
16.57.1 Save grain abundance . . . 211
16.57.2 Save grain charge . . . 211
16.57.3 Save grain continuum . . . 211
16.57.4 Save grain D/G ratio . . . 211
16.57.5 Save grain drift velocity . . . 211
16.57.6 Save grain extinction . . . 211
16.57.7 Save grain H2rate . . . 212
16.57.8 Save grain heating . . . 212
16.57.9 Save grain opacity . . . 212
16.57.10 Save grain potential . . . 212
16.57.11 Save grain qs . . . 212
16.57.12 Save grain temperature . . . 213
16.58 Save grid . . . 213
16.59 Save heating . . . 213
16.60 Save [option] history . . . 214
16.60.1 Save pressure history . . . 214

16.60.2 Save temperature history . . . 214
16.61 Save H2 . . . 214
16.61.1 Save H2 column density . . . 214
16.61.2 Save H2 cooling . . . 214
16.61.3 Save H2 cooling per molecule . . . 215
16.61.4 Save H2 creation . . . 215
16.61.5 Save H2 destruction . . . 215
16.61.6 Save H2 heating . . . 215
16.61.7 Save H2 levels . . . 215
16.61.8 Save H2 lines . . . 215
16.61.9 Save H2 populations . . . 216
16.61.10 Save H2 PDR . . . 216
16.61.11 Save H2 rates . . . 216
16.61.12 Save H2 Solomon . . . 217
16.61.13 Save H2 special . . . 217
16.61.14 Save H2 temperatures . . . 217
16.61.15 Save H2 thermal . . . 217
16.62 Save helium [options..] . . . 217
16.62.1 Save helium line wavelengths . . . 217
16.63 Save hydrogen . . . 217
16.63.1 Save hydrogen 21 cm . . . 217
16.63.2 Save hydrogen conditions . . . 217
16.63.3 Save hydrogen ionization . . . 218
16.63.4 Save hydrogen lines [alpha] . . . 218
16.63.5 Save hydrogen Lya . . . 218
16.63.6 Save hydrogen populations . . . 218
16.64 Save ionization means . . . 218
16.65 Save ionization rates carbon . . . 218
16.66 Save ip . . . 219
16.67 Save Leiden . . . 219
16.68 Save lines, options . . . 219
16.68.1 Save lines, array . . . 219
16.68.2 Save lines, zone cumulative . . . 219
16.68.3 Save line data . . . 220
16.68.4 Save lines, emissivity . . . 220
16.68.5 Save lines, intensity [every 5 zones] . . . 221
16.68.6 Save line labels [long] [no index] . . . 221
16.68.7 Save line list [absolute] . . . 222
Filenames . . . 223
Line List data format . . . 223
Output format . . . 223
Units of the line brightness . . . 223
The ratio option . . . 224
16.68.8 Save line optical depths, limit=-2 . . . 224
16.68.9 Save line optical depth some . . . 226

16.68.10 Save line pressure. . . . 226
16.68.11 Save line populations [row [lower|upper|ratio|tspin]] . . . 227
16.68.12 Save line RT . . . 227
16.69 Save map, zone 3 [range 3999 to 4500] . . . 227
16.70 Save molecules . . . 228
16.71 Save opacities [total, grain, element] . . . 228
16.71.1 Save total opacity . . . 228
16.71.2 Save grain opacity . . . 228
16.71.3 Save fine opacities range 0.7 to 1 Ryd, coadd 15 cells . . . 229
16.71.4 Save element opacity [name] . . . 229
16.71.5 Save opacity figure . . . 230
16.71.6 Save opacity shell 26 5 3 . . . 230
16.71.7 Save brems opacity . . . 230
16.72 Save optical depths . . . 230
16.72.1 Save fine optical depths . . . 230
16.73 Save OTS . . . 231
16.74 Save overview . . . 231
16.75 Save PDR . . . 231
16.76 Save performance . . . 231
16.77 Save pointers . . . 232
16.78 Save physical conditions . . . 232
16.79 Save pressure . . . 232
16.80 Save qheat . . . 233
16.81 Save radius . . . 233
16.82 Save recombination [option] . . . 233
16.82.1 Save recombination coefficients . . . 233
16.82.2 Save recombination efficiency . . . 233
16.83 Save results . . . 233
16.84 Save secondaries . . . 234
16.85 Save source function [depth, spectrum] . . . 234
16.85.1 Save source function, spectrum . . . 234
16.85.2 Save source function, depth . . . 234
16.86 Save species [options] . . . 234
16.86.1 Saving derived physical quantities . . . 235
Specifying a particular species . . . 235
Level populations . . . 235
Other quantities . . . 235
Examples . . . 235
Save species column densities . . . 236
Save species departure coefficients . . . 236
Save species densities . . . 236
Save species optical depths . . . 237
Save species levels . . . 237
Save species continuum [units Ryd] [row [inward, outward]] . . . 237
Save species bands . . . 237

16.86.2 Save species optical depths . . . 238
16.86.3 Saving fundamental physical properties of the species . . . 238
Save species energies . . . 238
Save species labels . . . 239
Save species lines [options] . . . 239
Save species data sources . . . 239
16.87 Save special . . . 239
16.88 Save tegrid . . . 239
16.89 Save temperature . . . 240
16.90 Save time dependent . . . 240
16.91 Save TPredictor . . . 240
16.92 Save XSPEC atable | mtable [range 0.1 200] [normalize [2.3]] . . . 240
16.93 Save wind . . . 241
16.94 Title “This is a title” . . . 241
16.95 Trace zone 94 [iteration 2; options . .] . . . 242
16.95.1 Trace convergence level . . . 242
16.95.2 Trace H-like [element name, full] . . . 242
16.95.3 Trace He-like [element name, full] . . . 243

17 THE OPTIMIZE COMMAND 245
17.1 Running the optimizer . . . 245
17.2 Overview . . . 245
17.3 Commands with vary option . . . 246
17.4 What must be specified . . . 246
17.5 Observed quantities . . . 248
17.5.1 Optimize column density . . . 248
17.5.2 Optimize, [intensity, luminosity]=36.425 [error =0.1] . . . 248
17.5.3 Optimize continuum flux 350 micron 126 Jansky [error =0.1] . . . 249
17.5.4 Optimize lines . . . 250
17.5.5 Optimize temperature . . . 251
17.5.6 Optimize diameter . . . 251
17.6 Optimization methods . . . 252
17.6.1 Optimize phymir [sequential mode] [8 cpu] . . . 252
17.6.2 Optimize Subplex . . . 252
17.7 Controlling the optimizer . . . 252
17.7.1 Optimize file= “best.in” . . . 252
17.7.2 Optimize increment = 0.5 dex . . . 253
17.7.3 Optimize, iterations = 1200 . . . 253
17.7.4 Optimize range -2.3 to 3.9 . . . 253
17.7.5 Optimize tolerance = 0.02 . . . 254
17.7.6 Optimize continue . . . 254
17.8 Convergence criteria . . . 254
17.9 Other optimizer commands . . . 255
17.9.1 No vary . . . 255
17.9.2 Optimize, trace start 4 . . . 255

17.9.3 Optimize, trace flow . . . 255
17.10 Notes concerning commands with vary option . . . 255
17.10.1 AGN . . . 255
17.10.2 Cosmic rays . . . 255
17.10.3 Dlaw . . . 255
17.10.4 Element . . . 256
17.10.5 Fudge . . . 256
17.10.6 Illumination . . . 256
17.10.7 Metals . . . 256
17.10.8 Power law . . . 256
17.10.9 Save output . . . 256
17.10.10 Radius . . . 256
17.10.11 Ratio alphox . . . 257
17.10.12 Stop thickness . . . 257
17.10.13 Stop radius . . . 257
17.10.14 Table star . . . 257
17.11 Notes concerning the optimization process . . . 257
17.11.1 Use physically motivated initial conditions . . . 257
17.11.2 Change the increment size . . . 258
17.11.3 Set physically motivated limits to the variable quantities . . . 258
17.11.4 Don't give up! . . . 258
17.12 Other optimization methods . . . 258
17.13 The optimizer test cases . . . 259

18 THE GRID COMMAND 261
18.1 Introduction to grids . . . 261
18.2 Running the code with grids . . . 261
18.2.1 Command line setup . . . 261
18.2.2 Grids on parallel machines . . . 262
18.2.3 Forcing sequential runs . . . 262
18.2.4 Specifying the number of CPUs . . . 262
18.3 Grid start point, end point, increment [linear] . . . 263
18.4 Grid list [linear] "filename" . . . 264
18.5 Beware the grid command treatment of temperatures!! . . . 264
18.6 Grid output options . . . 264
18.7 Other grid options . . . 265
18.8 Notes on various commands . . . 265
18.8.1 Constant temperature . . . 265

19 MISCELLANEOUS COMMANDS 267
19.1 Overview . . . 267
19.2 Introduction to the compile commands . . . 267
19.3 Compile stars . . . 267
19.4 Compile opacities . . . 268
19.5 Compile recombination coefficients [H-like; He-like] . . . 268

19.6 Compile grains . . . 268
19.6.1 Compile grains . . . 269
19.6.2 Compile grain 10 bins, [filename, ism graphite] . . . 269
19.6.3 Some caveats . . . 271
19.6.4 Examples . . . 271
19.7 Crash [keywords] . . . 271
19.7.1 crash zero . . . 272
19.7.2 crash overflow . . . 272
19.7.3 crash long overflow . . . 272
19.7.4 crash NaN . . . 272
19.7.5 crash setnan [float, double] . . . 272
19.7.6 crash isnan [float, double] . . . 272
19.7.7 crash segfault . . . 272
19.7.8 crash assert . . . 272
19.7.9 crash bounds . . . 273
19.7.10 crash exception . . . 273
19.7.11 crash domain . . . 273
19.7.12 crash undefined . . . 274
19.7.13 crash abort . . . 274
19.7.14 crash grid . . . 274
19.8 Eden -2 . . . 274
19.9 Fudge factors 4.59678 [12.3 34.5 958 . . .] . . . 274
19.10 Init [“c84.ini”, path] . . . 275
19.11 No 275
19.11.1 No advection [H-like, He-like, . . .] . . . 275
19.11.2 No Auger effect . . . 276
19.11.3 No blends . . . 276
19.11.4 No buffering . . . 276
19.11.5 No charge transfer . . . 276
19.11.6 No collisional ionization . . . 276
19.11.7 No Compton effect . . . 276
19.11.8 No CTHeat . . . 277
19.11.9 No diffuse line pumping . . . 277
19.11.10 No FeII pumping . . . 277
19.11.11 No file opacity . . . 277
19.11.12 No fine structure line optical depths . . . 277
19.11.13 No free free . . . 277
19.11.14 No grain [process] . . . 277
19.11.15 No induced processes . . . 278
19.11.16 No ionization reevaluation . . . 278
19.11.17 No isotropic continua report . . . 278
19.11.18 No line transfer . . . 279
19.11.19 No lines isotropic continuum subtraction . . . 279
19.11.20 No Lya 21cm pumping . . . 279
19.11.21 No OTS [options] . . . 279

19.11.22 No level 2 lines . . . 280
19.11.23 No molecules [heavy] . . . 280
19.11.24 No on the spot . . . 280
19.11.25 No opacity reevaluation . . . 280
19.11.26 No outward [options] . . . 280
19.11.27 No photoionization . . . 280
19.11.28 No radiation pressure . . . 281
19.11.29 No recoil ionization . . . 281
19.11.30 No scattering opacity . . . 281
19.11.31 No scattering escape . . . 281
19.11.32 No absorption escape . . . 281
19.11.33 No scattering intensity . . . 281
19.11.34 No secondary ionizations . . . 281
19.11.35 No Stark broadening . . . 282
19.11.36 No TePredictor . . . 282
19.11.37 No static opacities . . . 282
19.11.38 No three body recombination . . . 282
19.11.39 No times . . . 282
19.11.40 No vary . . . 282
19.12 Monitor commands . . . 282
19.12.1 Monitored results in the main output . . . 283
19.12.2 Monitor set error 0.1 . . . 283
19.12.3 Monitor ⟨subcommand⟩ grid “filename” . . . 283
19.12.4 Monitor Case B [H-like, He-like] element error . . . 283
19.12.5 Monitor chemistry searches < 3 . . . 284
19.12.6 Monitor chemistry steps < 8 . . . 284
19.12.7 Monitor column ”species” 18.9 . . . 284
19.12.8 Monitor csupra −17.09 . . . 285
19.12.9 Monitor Ctot -12.5 . . . 285
19.12.10 Monitor departure coefficients mean=1, error=0.04 . . . 285
19.12.11 Monitor depth 13.2 . . . 286
19.12.12 Monitor eden 9 . . . 286
19.12.13 Monitor grain potential . . . 286
19.12.14 Monitor H2 grain rate -16.5 . . . 286
19.12.15 Monitor H2 ortho/para ratio 2.02 . . . 286
19.12.16 Monitor HHeicf -0.013 . . . 286
19.12.17 Monitor Htot -13.2 . . . 286
19.12.18 Monitor ionization fraction oxygen 3 -3.45, error 0.1, weight = radius . . . 286
19.12.19 Monitor itrmax < 20 . . . 287
19.12.20 Monitor itrzn < 3.5 . . . 287
19.12.21 Monitor line [Case B] “H 1” 4861.33 < 1.01 . . . 287
19.12.22 Monitor molecular fraction H2 -3.45, error 0.1 . . . 288
19.12.23 Monitor mpi . . . 288
19.12.24 Monitor niter < 4 . . . 288
19.12.25 Monitor nothing 0 . . . 288

19.12.26 Monitor nzone < 135 . . . 289
19.12.27 Monitor PRadMax 0.34 . . . 289
19.12.28 Monitor pressure error < 0.01 . . . 289
19.12.29 Monitor radius 18.2 . . . 289
19.12.30 Monitor temperature "species" 4.02 volume, error 0.01 . . . 289
19.12.31 Monitor 21cm temperature [mean, spin, optical] 50 error 0.1 . . . 289
19.12.32 Monitor grain temperature index 2, temperature 234 . . . 290
19.12.33 Monitor temperature at face 11400K . . . 290
19.12.34 Monitor thickness 13.2 . . . 290
19.12.35 Monitor velocity 7.6 km/s . . . 290
19.13 Set commands . . . 290
19.13.1 Set Lya 21cm [excitation, kinetic, constant] . . . 290
19.13.2 Set 12C13C -3.2 – DEPRECATED . . . 291
19.13.3 Set blend ["Custom"] [4622.23 [air | vacuum]] [quiet] . . . 291
19.13.4 Set charge transfer -11.5 . . . 292
19.13.5 Set chemistry . . . 292
19.13.6 Set check energy every zone . . . 293
19.13.7 Set collisional ionization [Dima, Hybrid] . . . 293
19.13.8 Set collision strength averaging on . . . 293
19.13.9 Set continuum options . . . 294
19.13.10 Set convergence 295
19.13.11 Set coverage [fast] . . . 295
19.13.12 Set csupra = -12.34 . . . 295
19.13.13 Set cumulative [mass, flux, off] . . . 295
19.13.14 Set D/H -3.2 – DEPRECATED . . . 296
19.13.15 Set density tolerance . . . 296
19.13.16 Set didz 0.05 . . . 296
19.13.17 Set dielectronic recombination [keywords] . . . 296
19.13.18 Set dr 11.2 . . . 297
19.13.19 Set drmax 11.2 . . . 297
19.13.20 Set drmin 11.2 . . . 297
19.13.21 Set dynamics commands . . . 297
19.13.22 Set eden [options ....] . . . 298
19.13.23 Set flxfnt -20 . . . 298
19.13.24 Set gbar off . . . 298
19.13.25 Set grain heating factor 2. . . . 298
19.13.26 Set H2 299
19.13.27 Set HCORR . . . 300
19.13.28 Set HO charge transfer [chemistry; ionization] . . . 300
19.13.29 Set ind2 [on, off] . . . 300
19.13.30 Set ionization tolerance 0.01 . . . 300
19.13.31 Set isotopes [ALL] – DEPRECATED . . . 300
19.13.32 Set kshell energy 1e6 . . . 301
19.13.33 Set Leiden hack . . . 301
19.13.34 set monitor scientific notation . . . 301

19.13.35 Set nchrg 3 . . . 301
19.13.36 Set negopc . . . 301
19.13.37 Set nend 500 . . . 302
19.13.38 Set nFnu [options] . . . 302
19.13.39 Set nFnu add 300 micron . . . 302
19.13.40 Set nmaps 50 . . . 303
19.13.41 Set PAH option . . . 303
19.13.42 Set phfit [1995, 1996] . . . 303
19.13.43 Set pressure convergence 0.01 . . . 303
19.13.44 Set pressure ionize 50 . . . 303
19.13.45 Set save commands . . . 304
19.13.46 Set species gbar 0.01 . . . 305
19.13.47 Set species continuum . . . 306
19.13.48 Set temperature [floor, convergence] . . . 306
19.13.49 Set test . . . 306
19.13.50 Set trimming -9 [upper lower, off] [old, new] . . . 306
19.13.51 Set tsqden 8 . . . 307
19.13.52 set update couplings every ion . . . 307
19.13.53 Set UTA . . . 307
19.13.54 Set WeakHeatCool 0.02 . . . 308
19.14 Table lines “name.dat” . . . 308
19.15 Test . . . 308
19.15.1 Test . . . 308
19.15.2 Test large . . . 308
19.15.3 Test H2 . . . 309
19.15.4 Test molecular . . . 309
19.16 Performance speedups . . . 309
19.16.1 Turn off minor elements . . . 309
19.16.2 Degrade the grain physics . . . 309
19.16.3 Turn off the chemistry . . . 310
19.16.4 Turn off level 2 lines . . . 310
19.16.5 Degrade the continuum resolution . . . 310
19.16.6 Make the iso-sequence model atoms as small as possible . . . 311

A USING THE GRAIN CODE IN CLOUDY 313
A.1 Introduction . . . 313
A.2 Using the grain model . . . 314
A.3 Description of the refractive index files . . . 316
A.3.1 rfi_tbl . . . 317
A.3.2 opc_tbl . . . 319
A.4 Description of the mixed medium files . . . 320
A.5 Description of the size distribution files . . . 321
A.5.1 ssize . . . 322
A.5.2 ncarb . . . 322
A.5.3 power . . . 322

A.5.4 exp1, exp2, exp3 . . . 322
A.5.5 normal . . . 323
A.5.6 lognormal . . . 323
A.5.7 table . . . 323

B INCLUDING USER-DEFINED ATMOSPHERE GRIDS IN CLOUDY 325
B.1 Introduction . . . 325
B.2 The ascii file . . . 325
B.3 Compiling the grid . . . 328
B.4 Runtime behavior . . . 328

C LATEX STYLE AND MACROS 329
C.1 Blocks of verbatim text . . . 329
C.2 Macros . . . 329

REFERENCES 331

LIST OF FIGURES

2.1 Contributors to total radiation field . . . 8
2.2 Incident and reflected radiation field . . . 9
2.3 Cloudy_3D simulation of a planetary nebula . . . 11
2.4 Open and closed geometries . . . 12

4.1 Orion Spectrum . . . 30

6.1 AGN SED . . . 43
6.2 Crab Nebula SED . . . 54
6.3 ISM radiation field . . . 57
6.4 Stellar radiation fields . . . 62

7.1 Quantum heating . . . 76

9.1 Cylindrical geometry . . . 110

11.1 X-ray emission spectrum . . . 125
11.2 Effects of cosmic ray background on gas temperature . . . 128

12.1 Resolved and collapsed levels . . . 142

14.1 Cooling flow . . . 161

16.1 Incident and net emission . . . 198
16.2 Radiation field components in save continuum . . . 202
16.3 Save line ratio example . . . 225

LIST OF TABLES

3.1 Default Conditions . . . 20
3.2 Interpretation of Numerical Input . . . 23

4.1 Commands specifying both shape *and* intensity . . . 32

6.1 Legacy SED keywords . . . 52
6.2 Akn 120 Continuum . . . 52
6.3 AGN Continuum . . . 55
6.4 ISM Radiation Field . . . 59
6.5 Power Law Continuum . . . 59

7.1 Default Solar Composition . . . 68
7.2 Stored Abundance Sets . . . 71
7.3 Old Solar Composition, default in versions C84 - C94 . . . 73
7.4 2010 Solar Composition, keyword **GASS** . . . 74
7.5 Isotope Fractions Data . . . 77
7.6 Solar composition sorted by abundance . . . 81
7.7 Standard Grain Types . . . 82
7.8 Depletions . . . 93

16.1 Other quantities which can be added as columns in **save species**. . . . 236
16.2 Trace convergence keywords and routines . . . 242
16.3 Trace Keywords and Effects . . . 244

17.1 Commands with vary option. Entries "var" are explained in Section 17.10. . . . 247
17.2 Optimizer results . . . 259

19.1 Grain keywords for size distribution (*.szd files) . . . 270
19.2 Refractive index data files in the CLOUDY distribution . . . 270

A.1 The definition of each of the material types hardwired into Cloudy. . . . 317
A.2 The various choices for the enthalpy function hardwired in Cloudy. . . . 318
A.3 The various choices for the minimum charge hardwired in Cloudy. . . . 318
A.4 The various expressions for the electron affinity hardwired in Cloudy. . . . 318
A.5 The various choices for the inverse attenuation length. . . . 318
A.6 The various expressions for the photoelectric yield hardwired in Cloudy. . . . 318
A.7 The various choices for splitting up the grain emissions. . . . 318

A.8 The expressions for the ro-vibrational distribution of H_2 formed on various grain surfaces. 318
A.9 Example of a mixed medium file for fluffy silicate. 320
A.10 Allowed choices for the mixing law. 321

B.1 Example of an ascii file, derived from the kwerner.ascii file. 327

C.1 Naming convention for labels 329
C.2 Macros defined in `macros.tex` 330

Chapter 1

INTRODUCTION

1.1 Overview

CLOUDY is a microphysics code. It is designed to simulate physical conditions within clouds ranging from the intergalactic medium to the high-density LTE and STE limits. The range of temperatures extends from the CMB temperature up to 1.001×10^{10} K and the physical state ranges between fully molecular to bare nuclei. It predicts the thermal, ionization, and chemical structure of a cloud which lies with these limits, and predicts its observed spectrum.

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

Many environments are encountered in which dilute gas is heated and ionized by the radiation field striking the gas. Under these circumstances it is possible to predict the physical conditions (that is, the distribution of ionization, density, and temperature) across the cloud, and its resulting spectrum, in a unique and self-consistent manner. This is done by simultaneously solving the equations of statistical and thermal equilibrium, equations that balance ionization-neutralization processes, and heating-cooling processes, respectively. The graduate text Osterbrock and Ferland (2006, hereafter AGN3) describe the physics governing such environments and Ferland (2003) provides additional details on current research topics.

The instructions for downloading and setting up the code are on the web site www.nublado.org. We maintain a discussion board at cloudyastrophysics.groups.io where questions can be posted.

1.2 What must be specified

One powerful asset of photoionization analysis is the large number of observables resulting from only a few input parameters. Intensities of a very large number of spectral lines[1] are predicted by

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

CLOUDY. These result from the specification of only a) the shape and intensity of the external radiation field striking a cloud, b) the chemical composition and grain content of the gas, and c) the geometry of the gas, including its radial extent and the dependence of density on radius. The following subsections describe the general philosophy of the specification of each.

1.2.1 Incident radiation field

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

Shape of the external radiation field The shape of the spectral energy distribution (SED) should be fully specified between a frequency of 10 MHz ($\lambda \approx 29.98$ m – this is the lowest frequency observable with LOFAR) and an energy of 7.354×10^6 Ryd ($h\nu = 100$ MeV) if possible.[2] A physically-motivated radiation field spanning the full energy range should be specified if possible. This can be specified as a fundamental form (such as blackbody emission, optically thin bremsstrahlung emission, or a power law with optional exponential cutoff), interpolated from tables of points, or as the radiation field transmitted through a cloud and predicted by previous calculations. Additionally, a set of built-in radiation fields (for instance, emission from many model atmospheres, the observed Crab Nebula spectral energy distribution (SED), or several typical AGN SEDs) can be specified as look-up tables.

Intensity or luminosity of the external radiation field The intensity of the external radiation field striking the cloud must be specified. This can be given either as the flux striking a unit surface area of cloud or as luminosity radiated by the central object into 4π sr. The code must be able to derive the flux of photons ($\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$) striking the illuminated face of the cloud. There are many commands that set these. Two cases, the intensity and luminosity cases, can be used to set the strength of the radiation field. These are described in the next Chapter.

Combining several radiation fields Up to 100 different radiation fields can be included as part of the total field striking the cloud. There must be exactly the same number of shape and luminosity specifications. The code will stop if there are not.

1.2.2 Chemical Composition

The program considers the lightest 30 elements in detail. Grains are not part of the default composition but can be included. All stages of ionization are treated, and all published charge transfer, radiative recombination, and dielectronic recombination processes are included as recombination mechanisms. Photoionization from valence and inner shells and many excited states, as well as collisional ionization by both thermal and supra-thermal electrons, and charge transfer, are included as ionization mechanisms. The chemistry includes many molecules and will

[2] In much of the following discussion photon energies will be given in Rydbergs. The ionization potential of hydrogen is nearly 1 Rydberg. See the discussion in Part 2 of this document for an exact definition and how to convert the Rydberg to other units.

go to the fully molecular limit (H in H_2 and C or O in CO). Although the default composition is solar several other standard mixtures can easily be specified and an arbitrary composition can be entered.

1.2.3 Geometry

The geometry is always 1D spherical but can be made effectively plane parallel by making the inner radius much larger than the thickness of the cloud. The default is for the gas to have constant density and to fully fill its volume Other pressure laws and strongly clumped clouds can be computed as well.

CLOUDY normally assumes an ***open geometry***, or one in which the gas has a very small ***covering factor*** (these terms are defined in the following chapter and in section 5.9 of AGN3). This can be changed with the **sphere** (see section 9.11) command which sets the covering factor to a large enough value for light escaping the cloud in the direction towards the central object to always interact gas on the other side. This is a ***closed geometry***. Line photons which cross the central hole interact with line-absorbing gas on the far side if **sphere static** is set but do not interact (because of a Doppler shift due to expansion) if **sphere expanding** is set. This case is the default when **sphere** is specified.

1.2.4 Velocity Structure

CLOUDY normally assumes that only thermal motions broaden spectral lines and that there is no internal velocity structure. These assumptions can be changed. A component of microturbulence can be added with the **turbulence** command. A flow can be computed with the **wind** command.

1.3 The input deck

CLOUDY is driven by a set of command lines. These normally reside in a small input file which the code reads when it starts. The first word on each line is called the command, which is usually followed by one or more keywords and/or numbers. The commands and keywords may be abbreviated to a minimum of 4 characters, provided that abbreviation is unique. Commands and keywords may be entered as either upper or lower case characters (i.e., they are case-insensitive) but text between double quotes (`"`) is always treated as case-sensitive. For file names that case may be ignored by the operating system though. Where the names of molecular species are specified, these will often need to be in quotation marks, so that, for example, carbon monoxide (CO) and cobalt (Co) can be distinguished. Numbers may be entered in the usual computer notation (e.g., −2, 1.23, or +4.33e5). Non-ASCII characters are forbidden in commands, but they are allowed in comments. The input script (and other files that CLOUDY needs) need to be typed in a plain text editor, such as `emacs` or `vi` on Linux/UNIX systems or `notepad` on Windows, but *not* word processors like `MS word`.

When CLOUDY is executed as a stand-alone program standard input (`stdin`) is read for input and standard output (`stdout`) is used for output. It may also be run using commands stored in an text input file. As an example, create a small file (say, called `simple.in`) containing the following lines:

```
title example input
hden 5 # log of hydrogen density, cm^-3
blackbody 5e4 K # spectral shape is a 50000K blackbody
# intensity of blackbody is set with
# ionization parameter so starting radius is not needed
ionization parameter -2
```

Suppose that the code has been compiled to create the executable `cloudy.exe`. Then, the model described by the parameters in this input file could be computed by typing the following at the command prompt:

```
cloudy.exe -r simple
```

Note that on many Linux systems the command will have to be written as

```
./cloudy.exe -r simple
```

for system security. The file `simple.in` will be read for input and the output will be in the file `simple.out`.

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.

1.4 What is computed and reported

The code creates an output file as the model is computed. The program begins by echoing the input commands (after stripping hidden comments, see Sect. 3.2.10 for details). Comments are always ignored by the parser. The input stream ends with either a blank line or the end-of-file. Some properties of the incident radiation field, such as luminosity and number of photons in certain frequency ranges, are then printed.

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

After the zone calculations are complete and the model is finished the code will explain why the calculation stopped. It is very important to confirm that the calculation stopped for the intended reason. Some warnings, cautions, or notes about the calculation may also be printed. The code is designed to be autonomous and self aware. This self checking will ensure that its range of validity

is not exceeded. It will complain if it goes outside its range of validity, if it feels that some parameter has been incorrectly set, or if something surprising has happened during the calculation. This is an essential core feature of the code since it is now often used to generate grids of thousands of models, making it impossible to validate individual models by hand.

The final print out begins with a recapitulation of the entered commands followed by the predicted emission-line spectrum. The first two columns of the emission-line spectrum give an identifying label and wavelength for each spectral line. The third column is the log of the luminosity [erg s^{-1}] or intensity [erg cm^{-2} s^{-1}] of the emission line, and the last column gives its intensity relative to the reference line. The reference line is Hβ by default and other emission lines can be chosen with the **normalize** command. The third column will be either the luminosity or intensity. The luminosity (energy radiated by a shell of gas covering Ω sr of the central object) is predicted if the incident radiation field is specified as a luminosity. The line intensity $4\pi J$ (the energy emitted per square centimeter of the gas slab) is predicted if the incident radiation field is specified as an intensity. If the geometry is spherical but the incident radiation field is specified as an intensity then the line intensities will be expressed per unit area at the inner radius. Only the strongest emission lines are printed; the relative intensity of the weakest line to print is adjusted with the **print faint** command.

The last pages of the output gives some averages of the ionization fractions over the slab, the optical depths in various lines and continua, and other properties of the nebula.

There is a vast amount of information that is computed but not given in the standard output file. Special output files are created with the **save** commands described below.

Chapter 2

DEFINITIONS

2.1 Overview

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

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

2.2 Radiation fields

Figure 2.1 shows several of the radiation fields computed in the calculation.

2.2.1 Incident radiation field

The ***incident radiation field*** is the external radiation field emitted by the central object that strikes the ***illuminated face*** of the cloud. It is specified in the commands that set up the calculation. Often the incident field is the only energy source for the cloud. As the incident radiation field is transmitted through the cloud it is diminished by extinction.

2.2.2 Diffuse radiation field

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

The main difference between the calculation of a stellar atmosphere and a photoionized cloud is in the treatment of the diffuse fields. The gas albedo, which gives the relative probability of a photon being scattered versus being absorbed, is generally small in a nebula. As a result, the diffuse fields must be far weaker than the attenuated incident continuum since the second is usually the cloud's only real energy source. The total radiation field is usually dominated by the attenuated incident radiation field. By contrast in a stellar atmosphere the nearly isotropic diffuse

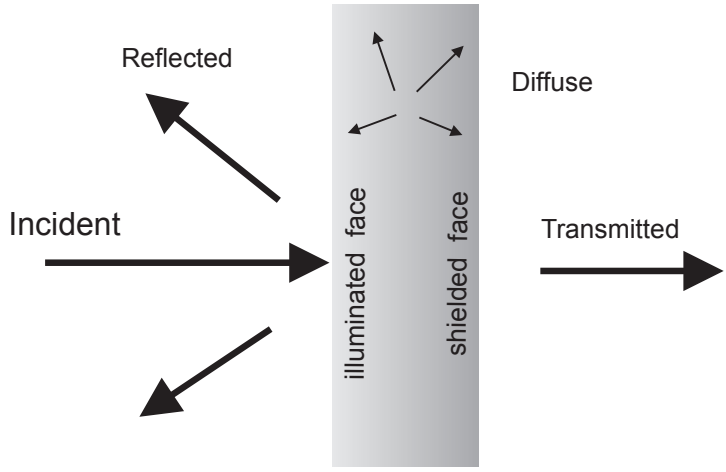

Figure 2.1: Several of the radiation fields that enter in the calculations.

field usually dominates the local intensity. As a result the diffuse fields can be treated by lower order approximations in a photoionized cloud than in a stellar atmosphere.

2.2.3 Transmitted radiation field

The ***transmitted radiation field*** is the net emission emergent from the shielded face of the cloud. It includes both the ***attenuated incident*** and the ***diffuse*** radiation fields.

2.2.4 Reflected radiation field

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

Figure 2.2 shows a plot of the incident and reflected radiation fields for the Compton reflector in an AGN. This is a large column-density cloud, $N(\mathrm{H}) = 10^{23.75}$ cm^{-2}, illuminated by a $f_\nu \propto \nu^{-1}$ power law. The incident radiation field is shown as a dashed line and the reflected radiation field, obtained from the **save continuum** command), is the solid line. The Compton reflector's peak at X-ray energies is clearly shown.

2.3 Geometry

In the simplest case the geometry is spherical, or plane parallel if the inner radius is much larger than the thickness of the cloud. The summary at the end of the calculation will say whether the

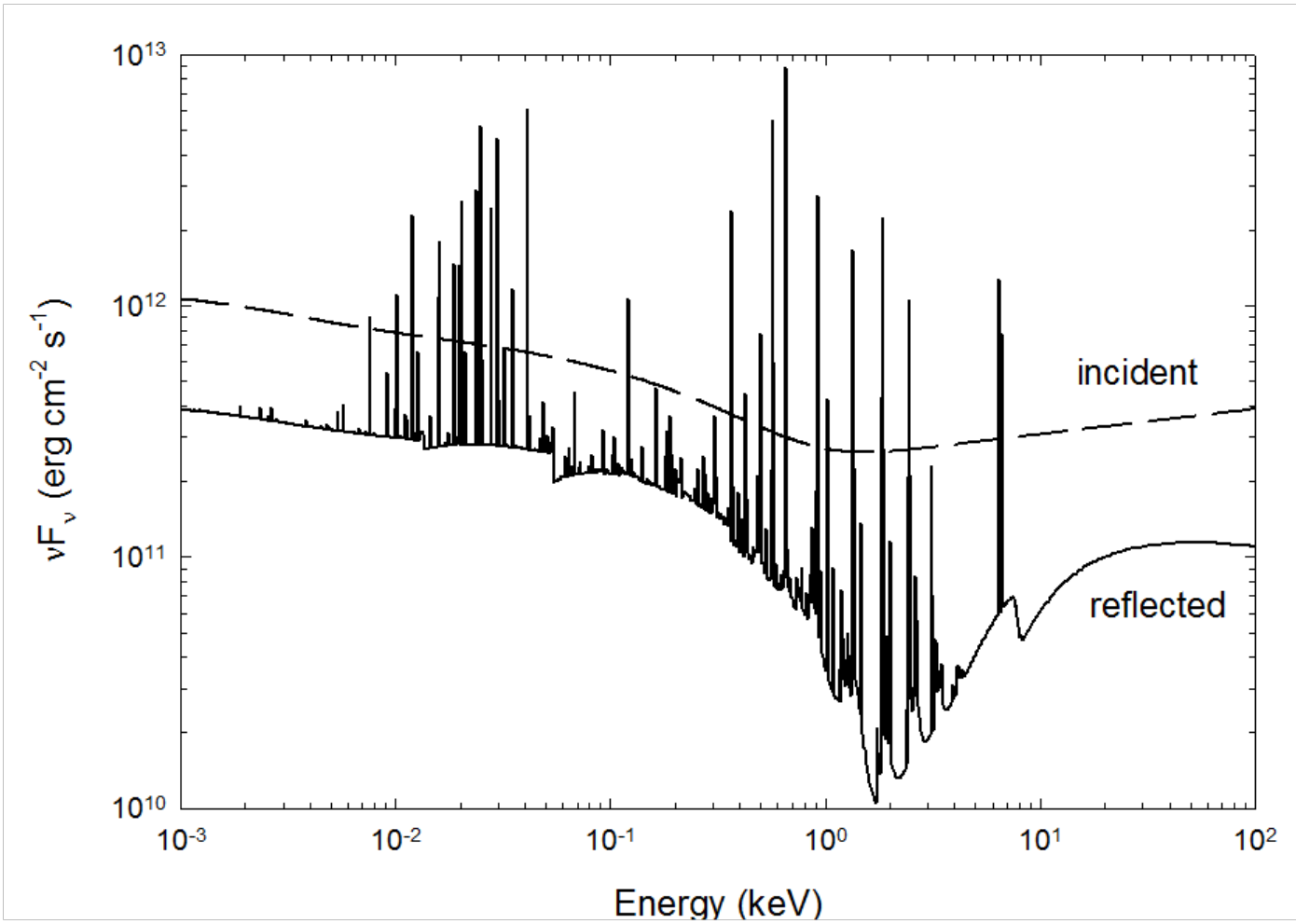

Figure 2.2: This incident (dashed) and reflected (solid) radiation fields, as computed by agn_blr_albedo.in in the test suite.

geometry was plane parallel (the ratio of the thickness to the inner radius, $\Delta r/r_o < 0.1$), a thick shell ($\Delta r/r_o < 3$), or spherical ($\Delta r/r_o \geq 3$).

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). Cloudy_3D was used to compute the images shown on the cover and in Figure 2.3. The RAINY3D code is another example (Moraes and Diaz, 2009).

2.3.1 Illuminated and shielded faces of the cloud

The side of the cloud facing the source of the external radiation field is the ***illuminated face*** of the cloud. The opposite side, in shadow, is the ***shielded face*** of the cloud. The illuminated face is generally hotter and more highly ionized than the shielded face. In nearly all cases the calculation starts at the illuminated face and stops at the shielded face.

2.3.2 Depth and radius

Figure 2.4 shows two possible geometries and some terms used to describe them. The ***radius*** is the distance from the center of symmetry, usually the center of the central object, to a given point. The ***depth*** is the distance between the illuminated face of the cloud and a point within the cloud. The ***inner radius*** is referred to as r_o, the depth is Δr, and the ***current radius*** is r. The depth or radius for a zone is the distance to the center of the zone.

2.3.3 Covering factor

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

2.3.4 Open vs. closed geometry

Two limiting cases, referred to as ***open*** and ***closed***, can be identified for the geometry and its influence upon the calculations. Figure 2.4 shows examples of both. The covering factor determines which is the best approximation. The choice mainly affects the transfer of the diffuse fields and has only second-order effects on predicted quantities.

Open geometry. An ***open*** geometry is one in which the covering factor of the gas is small. All radiation that escapes from the illuminated face of the cloud, towards the source of continuous radiation, then escapes from the system without further interaction with gas. This is thought to be the case in, for example, the broad-line region of active nuclei or a blister H II region. In this case Lβ and higher hydrogen Lyman lines and ionizing radiation

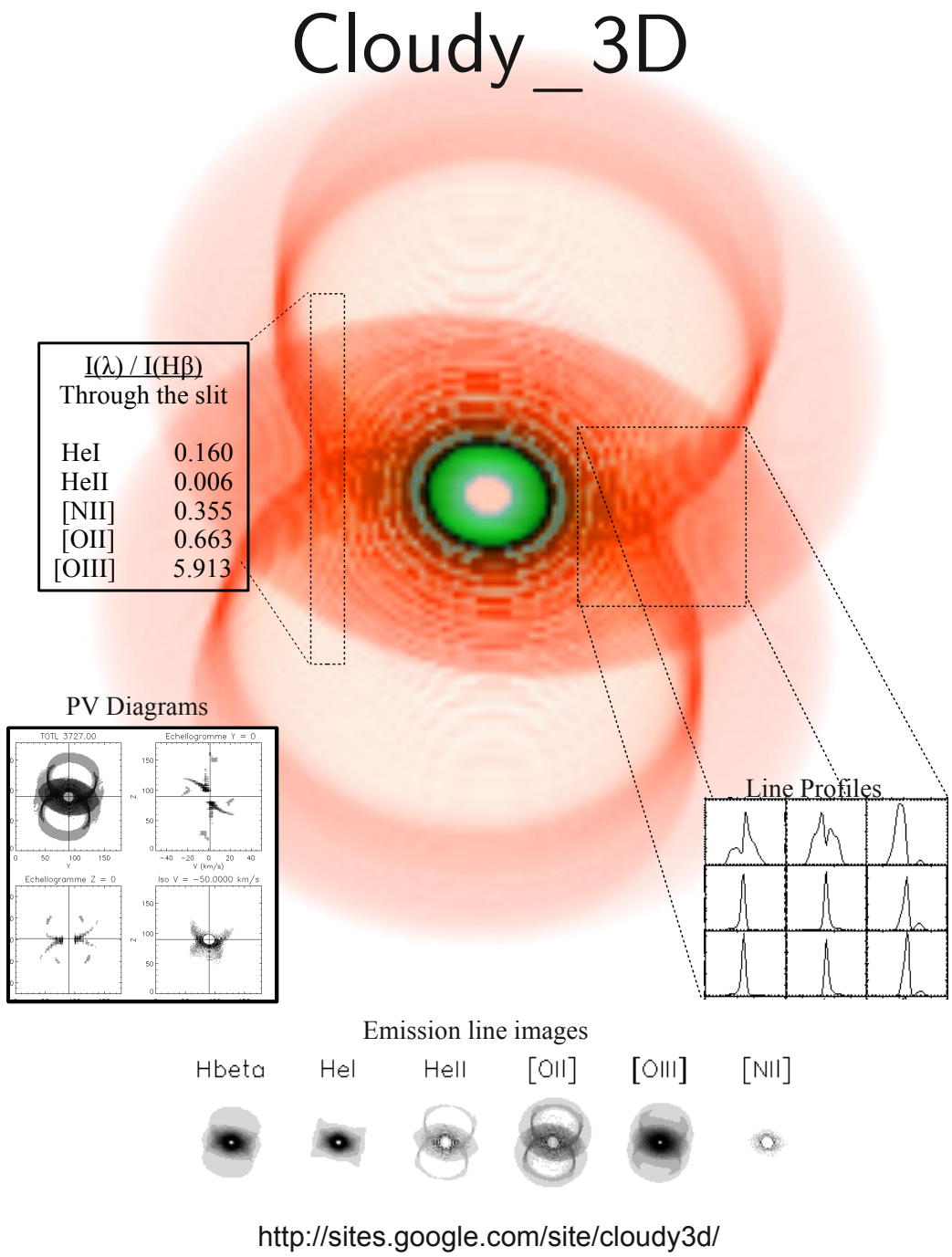

Figure 2.3: A 3-color image of a Hourglass-type nebula, obtained by running Cloudy 3D. Colors are [N II] (orange) and [O III] (green) emission. Emission line profiles are shown for [N II] lines. Intensities through any given slit can be obtained. Position-velocity diagrams are obtained as well as channel maps, for any line. Emission line surface brightness maps are also available for any line computed by CLOUDY. Statistical tools to analyze emission line properties are also provided.

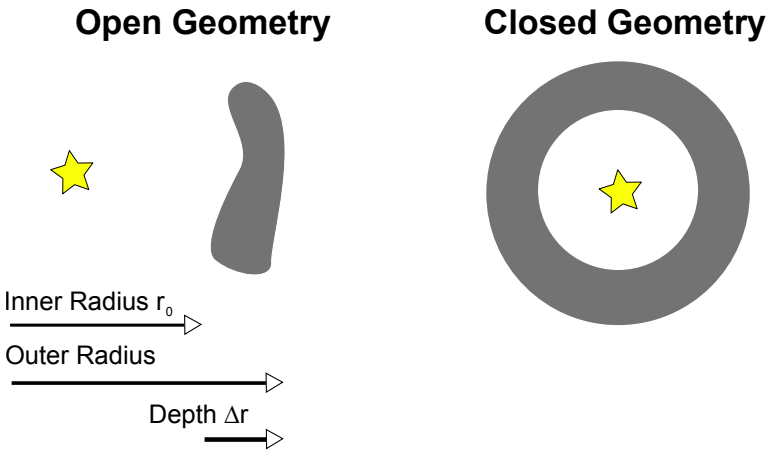

Figure 2.4: This figure shows the two limiting cases that can be assumed. The star is the source of ionizing radiation and the shaded area represents the cloud. An open geometry is the default, and a closed geometry will be computed if the **sphere** command (see section 9.11) is entered.

produced by recombinations can escape from the nebula. This geometry is the default and will be assumed if the **sphere** command is not specified.

Closed geometry. Emission-line gas covers $\sim 4\pi$ sr as seen by the central object in a ***closed*** geometry. The central object is small relative to the nebula then all diffuse fields which escape from the illuminated face of the cloud in the direction towards the central object will go on to strike the far side of the nebula. This geometry is implicitly assumed in most calculations of planetary nebulae and H II regions. This geometry will be assumed if the **sphere** command (see section 9.11) is entered.

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

Don't panic! These geometrical considerations (open vs closed, static vs expanding) make only second-order differences in the predicted emission-line spectrum, generally at the $\approx$10% level, largely because of the different treatments of the radiative transfer of the diffuse fields. If you are concerned about which geometry to use, try both, and compare the results. The differences are usually small.

2.3.5 Filling factor

The ***filling factor*** $f(r)$ accounts for clumping of emission-line gas. When a filling factor is set the hydrogen density is the density within regions containing gas while surrounding regions are

assumed to be a vacuum. The specific effects of a filling factor are described by Osterbrock and Flather (1959) and in AGN3 section 5.9.

2.3.6 Hydrogen density

The ***hydrogen density*** $n(\mathrm{H})$ is the total hydrogen density given by

$$n(\mathrm{H}) = n(\mathrm{H}^0) + n(\mathrm{H}^+) + 2n(\mathrm{H}_2) + \sum_{other} n(\mathrm{H}_{other}) \ [\mathrm{cm}^{-3}] \tag{2.1}$$

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

2.3.7 Column densities

The ***hydrogen column density*** $N(\mathrm{H})$ is given by

$$N(\mathrm{H}) = \int n(\mathrm{H})\, f(r)\, dr \ [\mathrm{cm}^{-2}] \tag{2.2}$$

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

2.3.8 Matter-bounded and radiation-bounded geometries

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

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

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

2.4 Intensity & luminosity cases

The external radiation field is usually specified with two different commands. One specifies the shape of the incident radiation field. A second command will set the brightness of the light. CLOUDY must be able to deduce the flux of photons [$\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$] striking the illuminated face of the cloud. There are two ways to specify the brightness of the external radiation field.

2.4.1 Luminosity case

If the brightness of the external radiation field is specified as a "luminosity", the light radiated by the central object into 4π sr, then it is also necessary to specify an inner or starting radius so that the flux of photons can be derived.

The emission lines will be predicted as luminosities. A covering factor will linearly change the luminosity of the entire spectrum but will have only second order effects on relative intensities.

2.4.2 Intensity case

The external radiation field can be specified as an "intensity", the energy incident upon a unit area of cloud, with units something like photons $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$. It is not necessary to specify a starting radius in the intensity case although it is possible to do so. Line intensities (energy emitted per unit area of cloud, related to $4\pi J$)), are predicted in this case.

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

The word "intensity" appears in quotes since it is not the I or J defined in radiative transfer texts, but rather the photon or energy equivalent of $4\pi J$.

2.4.3 The units of the predicted emission lines

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

2.5 "Species", how we specify atoms, ions, molecules, and their spectra

2.5.1 Overview

CLOUDY simulates gas ranging from fully ionized to molecular. Nomenclature varies considerably between chemical, atomic, and plasma physics. We adopted a nomenclature that tries to find a middle ground between these different fields.

We refer to a particular atom, ion, or molecule as a "species". A species is a baryon. Examples are CO, H_2, H^+, and Fe^{22+}. Species are treated using a common approach, as much as possible.

Our naming convention melds a bit of each of these fields because a single set of rules must apply to all species.

The species are taken from a number of databases and a large number of commands control how they are used. These commands are described in the Section beginning on page 135 below. The commands controling output options for the species are described in the Section starting on page 234 below.

A spectrum is a collection of photons. Examples that might appear in the literature are H I or C IV. It is not convenient to use these labels in computer input/output. We adopted the following rules.

2.5.2 How we specify species and spectra

Some rules for how these are specified:

- Labels are case sensitive, to distinguish between the molecule CO and the atom Co.
- At present we do not use _ to indicate subscripts, or ^ to indicate charge.
- Molecules are written pretty much as they appear in texts. H_2, CO, and H^- would be written as "H2", "CO", and "H-".
- Atoms are the element symbol by itself. Examples are "H" or "He" and *not* the atomic physics notation H^0 or He^0.
- Ions are given by "+" followed by the net charge. Examples are "He+2" or "Fe+22" and *not* the correct atomic physics notation, He^{2+} or Fe^{22+}. The latter would clash with notation for molecular ions. "C2+" indicates C_2^+ in our notation.
- We use ^ to specify isotopes, with ^ and the atomic weight placed before the atom to which it refers. For example, "^13CO" is the carbon monoxide isotopologue ^{13}CO.
- The **save species labels all** (page 239) command will produce a file containing the full list of species labels.
- We follow a modified atomic physics notation for the spectrum. H I, He II, and C IV are the spectra emitted by H^0, He^+, and C^{3+}. In atomic physics the notation "H I" indicates a collection of photons while H^0 is a baryon. Emission lines replace the Roman numeral with an integer. Examples are H 1 λ4861Å, He 2 λ4686Å, and C 4 λ1549Å.

  The spaces between the element and integer are significant. The spectrum label fills four characters so there are two spaces within H 1, one space within Si 2, and no spaces within Fe16.[1]
- To summarize, atomic hydrogen would be referenced as "H" while the Lα line would be "H I". The distinction is important because, depending on how it is formed, Lα can trace either H^0 or H^+.

[1] It is not safe to copy/paste these labels from the PDF file. Tests show that only one space is rendered. Try this experiment with the H 1. Copy/paste it into a text editor like vi. There will be one space. The original LaTex was correct and had two.

- The **save line labels** command (page 221 will create a file listing all spectral lines predicted by the code, along with a brief comment.
- The internal structure of a species, and associated cooling / emission, are computed using external databases, as described in the section beginning on page 135. We use our own Stout database, along with the Chianti (Dere et al. (1997); Landi et al. (2012)) and Lamda, the Leiden Atomic and Molecular Database (www.strw.leidenuniv.nl/ moldata/, Schöier et al. (2005)) databases.
- Each databases has its own "masterlist" file that specifies which models to use. The masterlist file follows the notation used within the databases. For Chianti and Stout, the internal structure of C^{3+}, which produces C IV emission, is called "c_4". The water molecule in Lamda is referenced as "H2O".
- If a particular species is specified in more than one masterlist file we will use Stout if it exists, then Chianti, followed by Lamda.

2.5.3 How we specify spectral lines

Several commands require you to specify a list of spectral lines, either in the input script or in a separate data file. One example is the **save line list** command. In all cases the same format is used. You need to specify exactly one spectral line on each input line, starting with a species label followed by a wavelength. If the species label is four characters or shorter, you can type it *as is* in the first four columns of the input line. If it is longer, it is mandatory that you surround the label with double quotes. You may also use double quotes for labels that are four characters or shorter, but this is optional. If you use double quotes, the string must start in the first column. After the label you need to enter the wavelength. The code assumes that wavelengths are in angstrom by default. You can also supply wavelengths in micrometer or centimeter by appending a "m" or "c" directly after the number (i.e., there may be no spaces inbetween). You may alternatively use upper case characters here. You can also append the character "a", which will explicitly indicate that you use the default unit angstrom. Note that the wavelength must start in the 5th column or later, even when the species label is shorter than four characters. There must always be whitespace between the label and the wavelength.

CLOUDY assumes by default that the wavelength you entered has the same type (air or vacuum wavelength) as is used in the main output. You can set the latter using the **print line vacuum** command described in Sections 2.6 and 16.19.20. This can be inconvenient, e.g. when you use the line identification in an init script that is intended to be used in both types of output. You can resolve this problem by adding the optional keyword **air** or **vacuum**. This keyword needs to be typed immediately after the wavelength, with at least one space between the wavelength and the keyword. The keyword will not affect how the wavelength of the line is printed in the output, only how the number immediately preceding it is interpreted. Below we show some examples of how you enter spectral lines.

```
O  3  5006.84  # green line of [O III], wavelength in angstrom
CO   2600.05m  # CO 1 -> 0 line, wavelength in micron
CO 2600.05m    # INVALID! wavelength starts in column 4!
CO   2600.05 m # INVALID! there should be no space before "m"!
```

```
"O  3" 834.5A  # double quotes are optional here
"^13CO" 2719.67m # double quotes are mandatory here!
O  3  5008.24 vacuum # force interpretation as vacuum wavelength
O  3  5006.84air # ERROR! no space between the wavelength and "air"
```

Sometimes specifying the label and the wavelength is not enough to find a unique match for the line. One example is the H I 4.65247 μm line, which can be either the $7 \rightarrow 5$ or $35 \rightarrow 7$ line since they have exactly the same wavelength. To disambiguate the match, you can optionally supply the indices of the lower and upper level (in that order), or the energy of the lower level (in cm^{-1}), as shown below:

```
H  1  4.65247m  index=11, 23  # this is 7 -> 5
H  1  4.65247m  Elow=105291.6 # also 7 -> 5
H  1  4.65247m  Elow=107440.4 # this is 35 -> 7
```

Only one of these methods can be used. Both methods have their advantages and disadvantages. Using the level indices guarantees a unique match, but the level indices can change if the data set changes (in the case of Stout, Chianti, or LAMDA) or if you use a different number of resolved levels in the iso sequences. The energy of the lower level on the other hand is more stable (e.g., changing the number of resolved levels for the iso sequences will not change the energy, but changing the data set could change it). The tolerance of the energy match may not be strict enough to disambiguate the lines in all cases (a match is declared if the relative difference is less than 10^{-6}, which may be insufficient for very close multiplet components). The necessary information can be found in the **save line labels** output. Note that disambiguation is not supported for all lines. If the **save line labels** output does not contain entries for index and Elow, then this type of disambiguation is not supported for that line. In the excerpt shown below, disambiguation is supported for the first line, but not the second.

```
 79502  Al 3   1.00203m  # index=4, 6 Elow=113957  Stout, 3d 2D1.5 -- 4s 2S0.5
216990  B  3   733.634A  # inner shell line
```

To clarify this, if you want to disambiguate a correct line that supports disambiguation from an incorrect line that does not support disambiguation, it will work as expected. But the other way around it will not work.

Nothing else may appear on the line except a comment starting with a comment character as is shown in the examples above. In general it is best to do a copy / paste of a line identification from the CLOUDY output (e.g., the line list produced with the **save line labels** command) but remember that species labels that are longer than 4 characters *must* be enclosed in double quotes.

Note that the intensity of the emission lines resulting from magnetic dipole transitions, also called M1 lines are not added in the sum of the corresponding line intensity. Instead, we keep these M1 lines seperate, and are distinguished by their line label ending with "M1". For example, "H 1 1215.67A" and "H 1 M1 1215.67A" are two separate lines.

2.5.4 What species are you using?

You can generate a report giving all species, the number levels in the model, and the database used for the species, by running CLOUDY with the following input;

```
test
species print
```

You can generate a list of all species labels by running the following input deck:

```
test
save species labels "test.slab"
```

You can generate a list of all emission lines by running:

```
test
save line labels "test.llab"
```

The following is an example using species labels to report the column densities of several ions and molecules, together with the visual extinction:

```
save species column densities "test.col"
"e-"
"CO"
"H2"
"H"
"H+"
"*AV"
end of species
```

2.6 Air vs vacuum wavelengths

The convention in spectroscopy, dating back to 19th century experimental atomic physics, is to quote line wavelengths in vacuum for $\lambda_{vac} \leq 2000$ Å and STP air wavelengths for $\lambda_{vac} > 2000$ Å. The **print line vacuum** command tells the code to print vacuum wavelengths throughout. See Section 16.19.20 for more details. The continuum reported by the family of **save continuum** commands, described in Section 16.42, is always reported in vacuum wavelengths to avoid a discontinuity at 2000 Å.

Chapter 3

INTRODUCTION TO COMMANDS

3.1 Overview

This section introduces the commands that drive CLOUDY. The following chapters group the commands together by purpose. Individual commands are discussed after examples of their use. This section begins by outlining conditions that are assumed by default and then goes on to discuss the various classes of commands (i.e., those that set the incident radiation field, composition, or the geometry).

Keeping this document parallel with the code is a very high priority. In case of any confusion, please consult the original source. The commands are parsed by the series of routines that have names beginning with *"parse"*. The list of routines can be seen by listing the files `parse*.cpp`. The second half of the name indicates the command that is parsed by that routine.

CLOUDY is designed so that a reasonable set of initial conditions to be assumed by default so that a minimum number of commands are needed to drive it. These default conditions are summarized in Table 3.1, which also lists the commands that change each assumption.

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

3.2 Command format

3.2.1 Input and Output

When executed as a stand-alone program CLOUDY reads `stdin` for input and produces output on `stdout`. From a command prompt, this would be done as **cloudy.exe** < **input** > **output** or **cloudy.exe -r prefix** (the latter form will read its input from `prefix.in` and write its main output to `prefix.out`). A third option is to supply the CLOUDY commands on the command line, e.g. **cloudy.exe -e test** which will execute the CLOUDY command **test** (the "smoke test"). This is useful for quickly executing short scripts. If you want to execute multiple commands, the commands need to be separated by a semicolon and embedded in single quotes (to protect the semicolon from the shell), e.g. **cloudy.exe -e 'set continuum resolution 0.1; compile grains'**. The code is also designed to be used as a subroutine of other, much larger, programs. In this case

Table 3.1: Default Conditions

Quantity	Value	Command
Default inner radius	10^{30} cm	**radius**
Default outer radius	10^{31} cm	**radius**
Highest allowed temperature	1.001×10^{10} K	
Stop calculation when T falls below this value	4000 K	**stop temperature**
Relative error in heating-cooling balance	0.005	**set temperature convergence**
Relative error in electron density	0.0	**set eden convergence**
Relative intensity of faintest line to print	10^{-3}	**print line faint**
Low-energy limit to continuum	3.040×10^{-9} Ryd	
High-energy limit to continuum	7.354×10^{6} Ryd	
Limiting number of zones	1400	**set nend**
Total hydrogen column density	10^{30} cm^{-2}	**stop column density**
H^+ column density	10^{30} cm^{-2}	**stop column density**
H^0 column density	10^{30} cm^{-2}	**stop column density**
Grains	No grains	**grains**
Spectral resolution	mesh resolution	**set save line width**
Background cosmic rays	No	**cosmic rays**
Cosmic background	No	**background**

the input stream is entered using the subroutine calls described in a section of Part 2 of this document. In either case, this input stream must contain all the commands needed to drive the program. The command format is the same whether used as a stand-alone program or as a subroutine.

3.2.2 Command-line format.

Commands are entered at the start of a line (i.e. starting in column 1). They may be abbreviated, as long as the abbreviation is unambiguous, with a minimum of four characters (unless the whole command is shorter of course). The command is usually followed by one or more numbers or keywords. The script can be either lower or upper case, i.e. it is case insensitive. All the characters of the command name that are typed will be checked by the parser, unlike keywords where usually only the first four characters are checked[1]. The code adheres to US spelling where appropriate.

The end of each command line is marked[2] by the end-of-line, or the start of a comment (see Sect. 3.2.10).

The command lines can generally be in any order and there is no limit to the length of an input line. The input stream ends with a blank line (i.e. an empty line, or a line with a space in the first column), the end-of-file, or a field of stars ("***", i.e. a minimum of three consecutive stars starting in the first column).

[1]This will be changed in a future version of the code such that also all characters of the keywords will be checked.

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

3.2.3 Units

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

3.2.4 Number of commands.

There is no limit to the number of commands that may be entered.

3.2.5 Output as input.

CLOUDY can read its own output as an input stream. As described in the section *Output* in Part 2 of this document, the code echoes the input command lines as a header before the calculation begins. These lines are centered on the page and surrounded by asterisks.

Sometimes a particular model will need to be recomputed. You can do this by making a copy of the printed command lines and using this copy as an input file. The input parser will handle removal of the leading spaces and asterisk. This is mainly a debugging aid.

3.2.6 Syntax used in this document

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

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

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

These square brackets indicate only that the parameters are optional. The brackets should not be placed on the command line. They will be totally ignored if they occur. The above example would actually be entered as follows:

```
# following gives flux density at energy of 0.1824 Ryd
f(nu) = -12.456 at .1824 Ryd
#
# the luminosity command with linear keyword
luminosity 2e38 solar linear
#
# the phi(h) command with the range option
phi(h) = 12.867 range 0.1 to 0.2 Ryd
```

Underscores indicate a space. Most commands and keywords require four character matches to be recognized. Keywords which start with a letter (i.e. A−−Z) must start following a space (or

other non-alphabetic character) in order to be recognized. Only one space is needed between words.

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

```
# blackbody with T=5e4 K, in strict TE
blackbody 5e4 K lte
#
# use ISM radiation field
table ism
```

The following is how the commands should actually be entered:

```
# blackbody with T=5e4 K, in strict TE
blackbody 5e4 K lte
#
# use ISM radiation field
table ism
```

The space must occur where the underscore is written.

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

The examples of commands that follow show the square brackets and underscores for optional parameters and required spaces. Many people put these characters into the input stream because they don't read documentation. As a service to the user, the command-line parser will usually replace any square brackets or underscores with the space character when the command lines are initially read. The exception is any part of a string that is surrounded by double quotes. The string between double quotes is likely to be a file name and an underscore can occur in such a name.

3.2.8 The continue option

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

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

3.2.9 Numerical input

Numerical parameters are entered on the command line as free-format numbers. Exponential notation can be used.[3] Numbers may be preceded or followed by characters to increase

[3]Exponential notation could not be used in versions 07 and before.

Table 3.2: Interpretation of Numerical Input

Typed Quantity	Interpreted as	Why
1e2	100	Exponentional notation is OK
1.2.3	two numbers 1.2 0.3	1.2 is parsed, then .3
.3 3	0.3 and 3.0	
10^3	1000	^ acts as exponentiation operator
$temp	variable value	set by $temp = 1000 previously in input

readability. The strings "T=1000000K", "1000000", and "T=1E6" are equivalent. A period or full stop (".") by itself is interpreted as a character, not numeral or number.

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

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

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

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

Table 3.2 shows how various typed input numbers will be interpreted. The first column gives the typed quantity, the second its interpretation, and the third the explanation.

3.2.10 Comments

Comments may be entered among the input data in several ways. Comments can be entered at the end of a command line after a sharp sign ("#") or a double sharp sign ("##"). Anything on a line after one of these characters is completely ignored. This can be used to document parameters on a line. Any line beginning with a # or ## is also a comment and the entire line is ignored. Any comment starting with a single sharp sign is "visible", which means that it is echoed in the main output and also included in the file `optimal.in` (see Sect. 17.7.1). Comments starting with a double sharp sign are "hidden" comments and will be omitted in the main output and the file `optimal.in`. There is also a **title** command, to enter a title for the simulation.

3.2.11 Hidden commands

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

3.2.12 Commands for experimental parts of the code

The code is in continuous development. When new versions are released there will be new or experimental parts of the code that are still being developed or which have not been fully debugged. The newly-developed physics is not included in a calculation which uses the default conditions. The commands which exercise these new features are included in this document and are indicated in two ways. First by the symbol in the section header, and second by a grey background. You are welcome to give these commands a try but should not expect robust results.

3.2.13 Log vs linear quantities

Most quantities are entered as the log of the number but some are linear. The following outlines some systematics of how these are entered.

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

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

Using the new notation `10^3.5` as a linear number is equivalent to the form **3.5 log**, and will be required to replace the **log** keyword in future versions of CLOUDY.

3.2.14 The help command

This command prints helpful information about Cloudy and exits. Don't expect much as yet, beyond a recommendation to read this document!

3.2.15 An example

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

```
title "this is the input stream for a planetary nebula"
#
# set the temperature of the central star
```

```
blackbody, temperature = 1e5 K
#
# set the total luminosity of the central star
luminosity total 38 # log(Ltot)- ergs/s
radius 17   # log of starting radius in cm
hden 4      # log of hydrogen density - cm^-3
sphere # this is a sphere with large covering factor
```

3.3 Reading and Writing Files

When CLOUDY is running, it needs to read many files and will usually also write new files to disk. In this section we will describe the rules used by the code to decide the location of these files. The skinny is that input files are always searched along a search path and output files are written in the local directory. Only one exception exists: the main input script, which must be located in the local directory. The default search path is to look first in the local directory, then the data directory of the CLOUDY distribution. Below we will give more details.

The first thing the code needs is the main input script. The name of this file can be supplied to the code on the command line (e.g., using the *-r* flag). This file always needs to be in the local directory (a.k.a. current working directory). The code will not work correctly if you try to supply a filename in a subdirectory or using an absolute path, and will abort in that case. All output files written by the code will be in the local directory as well. This of course includes the main output file.

It is sometimes necessary to read or write external files whose names are specified on a command line. File names are entered inside pairs of double quotes, as in `"name.txt"`.

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

If an input file is supplied in this way, the rule is to always search for the file along a search path. This allows the user more freedom to organize the files. The default search path is to look in the local directory first, and if it is not found there, look in the CLOUDY data directory. The code will issue a caution if multiple matches with the same file name were found along the search path. It will then continue using the first match. An example is shown below:

```
CAUTION: multiple matches for file continuum_mesh.ini found:
   ==/path/to/run/continuum_mesh.ini==
   ==/path/to/cxx.yy/data/continuum_mesh.ini==
Using the first match.
```

The user can override or extend the default search path. This is described in more detail below in Section 3.4.

As stated before, CLOUDY will always write output files in the local directory. If you try to supply a different path between double quotes (i.e., a file in a subdirectory or an absolute path) the code will abort, as in the following example:

```
save continuum "subdir/name.com" # ERROR, this will abort!
```

The search path described above will also be used when CLOUDY is reading data files from its distribution. This allows the user to place modified data files in the local directory and leave the files in the main data directory untouched. This is useful if you want to limit the scope of the modification to the current run only (to be precise, all subsequent runs in the directory that contains the modified file). One example would be when you want to alter the resolution of the CLOUDY frequency mesh by modifying the file `continuum_mesh.ini` (see Section 4.5.3 in Part 2 of HAZY). If you place the modified file in the local directory, it will only affect runs in that directory and not globally.

3.4 The Search Path for Input Files

The code uses a search path to locate input files, as was discussed in the previous section. The default search path is to look in the local directory first and then in the data directory included in the distribution. The user can override that setting by defining the environment variable *CLOUDY_DATA_PATH*. How you can do that depends on the operating system and the shell you are using. For csh/tcsh under Linux/UNIX use:

```
setenv CLOUDY_DATA_PATH ".:/path/to/cloudy/data" # (double quotes are optional)
```

For bash/sh/ksh/etc... you should use:

```
export CLOUDY_DATA_PATH=".:/path/to/cloudy/data" # (no spaces around the '='!)
```

These methods should also work under Mac OS X. The environment variable will be read when the code starts up and then override the standard setting compiled into the code. If you subsequently undefine the environment variable, the code will revert back to using the standard search path.

The environment value *CLOUDY_DATA_PATH* supports a search path with multiple components, similar to the search paths used in UNIX. You can supply multiple directory paths, separated by a colon (':') on UNIX and Mac systems, or separated by a semicolon (';') on Windows systems. When Cloudy searches for a file, it will search in each of the component directories until it finds the file. The search will be done in the same order as they were defined in the environment variable. The code will *not* do a recursive descent into subdirectories of the search path components. The examples shown above would be equivalent to the default setting as the special path `` `.' `` is shorthand for the local directory, irrespective of where you are.

When you modify the search path, you should realize that the default data directory must always be part of the search path. The code will not function otherwise! It is also strongly recommended to keep the local directory as part of the search path. To simplify this, CLOUDY supports the special path component `` `+' `` which will be expanded to the default search path for that release.

A typical example would be this:

```
export CLOUDY_DATA_PATH="+:/path/to/local_data"
```

This search path would consist of the default search path (i.e., the local directory and the CLOUDY data directory) followed by a custom directory `/path/to/local_data` where you can store local files, such as stellar atmosphere grids, or custom grain opacity files. This avoids "polluting" the standard data directory and also makes it easier to port the files from one release to the next.

You can make multiple local data directories and combine them in custom sequences. One example would be one directory containing general local files, while the other contains files specifically needed for a non-standard spectral resolution run (such as binary stellar atmosphere grids, or grain opacity files). Such an example could look like this:

```
export CLOUDY_DATA_PATH="/path/to/non_standard_data:+:/path/to/local_data"
```

Note that we put `/path/to/non_standard_data` up front (this is the directory containing data for a non-standard spectral resolution). This assures that the non-standard resolution version of a file is picked up first, before the standard version.

3.5 The init command

This is a special command that tells the code to read a set of commands stored in an ancillary file. This allows frequently-used commands to be stored in a separate file. The **init** command is fully described in later sections below.

The filename can be specified within a pair of double quotes, as in `"ism.ini"`. The default name for an initialization file is `cloudy.ini`. There is no limit to the number of commands that can be in an initialization file.

This provides an easy way to change the default behavior of the code. For instance, many of the elements now included in CLOUDY have negligible abundances and the code will run a bit faster if they are turned off with the **element off** command. Also, only about half of these elements were included before version 86 of the code. The file `c84.ini` in the CLOUDY data directory which will turn off many of these elements. The `c84.ini` file contains the following commands:

```
print off hide
element Lithium off
element Beryllium off
element Boron off
element Fluorine off
element Phosphor off
element Chlorine off
element Potassium off
element Scandium off
element Titanium off
element Vanadium off
element Chromium off
element Manganese off
element Cobalt off
element Copper off
element Zinc off
print on
```

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

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

were entered in the input stream.

A series of `*.ini` files are included in the data directory included in the CLOUDY distribution. Do an `ls *.ini` within the data directory to list the available files. Note that not all these files are init files, e.g., `continuum_mesh.ini` is not an init file in the sense we discussed here. Comments at the start of the files describe their purpose.

3.6 Random numbers

Some commands in CLOUDY need random numbers to work. Like all computer codes, CLOUDY uses a pseudo random number generator (PRNG) to create these numbers. A PRNG creates a deterministic sequence of numbers starting from a given seed. However, on each invocation the bits in the number are so efficiently scrambled that for all intents and purposes they can be considered a truly random sequence of numbers. The PRNG in CLOUDY is by default a fully vectorized version of xoshiro256** (Blackman and Vigna, 2018, in press)[4], while the random numbers with a normal distribution are now generated using the Ziggurat algorithm (Marsaglia and Tsang, 2000). Both methods are very fast. An additional advantage is that the new code is fully aware of parallelization in the code, meaning that parallel ranks created with MPI or fork will automatically have a completely different sequence of random numbers. CLOUDY generates a random seed at the start of execution by default (when available derived from `/dev/urandom`, otherwise using the system time). The executable accepts the `-s [ seed ]` flag, which will set a fixed seed for the random number generator and is intended for debugging and testing purposes. The parameter is a 64-bit seed in hexadecimal form. If the seed is omitted, a default fixed value will be used. It is not recommended to use this flag in normal CLOUDY runs.

[4] http://xoshiro.di.unimi.it.

Chapter 4

THE INCIDENT RADIATION FIELD

4.1 Overview

The incident radiation field should be defined between the low-frequency limit to the code, 10 MHz, corresponding to a wavelength of 29.98 m, and the high-energy limit of 7.354×10^6 Ryd, corresponding to an energy of 100 MeV. The shape and its intensity or luminosity are usually specified independently. There are many ways to do this.

4.2 The coarse and fine continua

The code uses multi-grid methods to compute the overall spectral emission while including interactions among the very large number of spectral lines (Shaw et al., 2005). The ***coarse continuum*** is used to define the incident and diffuse continua. A second ***fine continuum*** has much higher spectral resolving power and is used to treat line transfer. The fine continuum resolution is high enough to resolve overlapping spectral lines. It is used to define line blocking coefficients that are passed up to the coarse continuum. This permits the effects of line overlap and velocity shear to be treated automatically. The fine continuum does not include continuous emission or absorption from the cloud so photoelectric or grain absorption will not be present (this is treated in the coarse continuum). Only the very strongest absorption lines will be visible on the coarse continuum, but the fine continuum will often show thousands of lines. There are several options on the **save continuum** command (page 197) that report both of these continua while various **set continuum** commands described on page 294 change details of this output.

Figure 4.1 compares parts of the fine (left panel) and coarse (right panel) continua at a point near the H^0 - H_2 transition in the Orion H II region / PDR. The fine continuum shows thousands of lines, mainly H I and H_2, but no emission. The coarse continuum shows gas and dust emission and the effects of extinction of this emission.

The spectral resolving power is defined as $E/\delta E$, where E is the photon energy and δE the width of the resolution element. The number of energy bins needed to store the radiation field is proportional to the resolving power. The number of bins is a major pace setter for the code since it must continuously reevaluate various integrals over the radiation field. The resolving power of the coarse continuum is defined in the file `continuum_mesh.ini` in the data directory. This file is designed to be changed by the user and can be changed to much higher resolving power if need

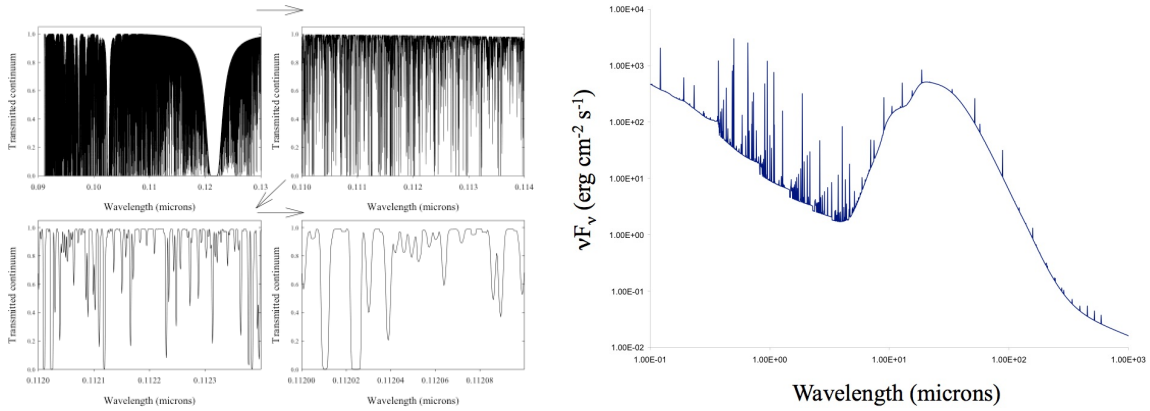

Figure 4.1: The spectrum of the Orion H II region / PDR at a point near the H^0 - H_2 transition. The left panel shows the fine continuum and is taken from Figure 20 of Shaw et al. (2005) It shows the extinction of the stellar SED by absorption lines formed in the gas. The right panel shows the coarse continuum for a cloud stopping at this point and mainly shows the extinguished stellar SEC also with dust and gas emission.

be. This is all described in Section 4.5.3 in Part 2 of this document. The resolution of the coarse continuum for a particular model can also be changed with the **set continuum resolution** command described on page 294. If you change the continuum resolution you will need to recompile the stellar continuua and grain opacities so that their energy grid agrees with that used by the code.

The resolving power of the fine continuum is established at the start of a calculation. It needs to resolve lines, so the resolution depends on the lowest possible kinetic temperature and any turbulence that may be present. The **set fine continuum** command described on page 294 can also adjust this resolution.

4.3 Defining a single component of the incident radiation field

Two quantities, the *shape* and *intensity*, specify a single component of the incident radiation field. The shape gives the form of the spectral energy distribution but not its intensity. The intensity can be specified as either the energy striking the illuminated face of the cloud, referred to as the intensity case, or both the source luminosity and the distance separating the source and the cloud. The latter is referred to as the luminosity case. The shape and intensity are specified independently in most cases, although some commands specify both (the command specifying the cosmic microwave background is an example of the latter).

In much of the following discussion we will refer to both luminosity and intensity commands as simply intensity commands. The distinction between the luminosity and intensity cases is discussed further on page 13.

4.4 Combining several radiation fields

4.4.1 The sum of several radiation fields

It is possible to combine up to 100 incident radiation fields.[1] CLOUDY will stop if more than 100 continua are entered. This limit is set by the variable *LIMSPC* that occurs in one of the included header files.

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

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

```
# this is the black body associated with the boundary layer
# this is a shape command
black body, temperature =5e5 K
# this gives its total luminosity
luminosity (total) 37.3
# the following shape is a rising power law,
# a simple approximation to the disk
power law, slope = 1.333, cutoff = 0.6 Ryd
# the total luminosity of this power law component
luminosity (total) 37.2
```

The 5×10^5 K blackbody has a total luminosity of $10^{37.3}\,\mathrm{erg\,s^{-1}}$ while the power-law has a total luminosity of $10^{37.2}\,\mathrm{erg\,s^{-1}}$.

4.4.2 Keeping shape and intensity commands together

It is not absolutely necessary to keep the ordered pairs of shape and intensity commands together but this is a good practice since some commands (those given in Table 4.1) specify *both* the shape *and* intensity of the incident radiation field. Problems arise if one of the commands giving both shape and intensity is entered between another pair of shape and intensity commands. For instance, the following will produce unintended results:

```
black body, temp = 5e5 K
CMB, z=2
luminosity (total) 37
```

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

Table 4.1: Commands specifying both shape *and* intensity

background
blackbody, energy density
blackbody, LTE
blackbody, luminosity
blackbody, radius
CMB
table Draine
table HM96, HM05, HM12
table ISM
table read scale

4.5 Check the incident radiation field!

It is important to check that the incident radiation field has been entered correctly. First set up the commands to do a simulation and include the **save continuum** command (described on page 197). Do the simulation and examine the file produced by the **save continuum** command. The first column gives the photon energy and the second gives the incident radiation field (as νJ_ν, erg cm^{-2} s^{-1}) at the illuminated face of the cloud. Plot this radiation field and check that it is correct.

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

Chapter 5

INCIDENT RADIATION FIELD LUMINOSITY

5.1 Overview

All commands setting the intensity or luminosity of the incident radiation field are defined in this Chapter.

5.2 Intensity and luminosity cases

The brightness of the incident radiation field can be specified as either an intensity, the energy per unit area of cloud, or as a luminosity, the power emitted by the central source of radiation into 4π sr. The ***intensity case*** and ***luminosity case*** are described on page 13. Each of the following commands is listed as an intensity or luminosity command. This distinction is important because the inner radius of the cloud must also be specified in the luminosity case.

5.3 The range option

Many of the intensity/luminosity commands specify the number of photons or integrated energy *in hydrogen-ionizing radiation* ($1\,\mathrm{Ryd} \leq h\nu \leq 7.354 \times 10^6\,\mathrm{Ryd}$). Other energy intervals can be specified with the **range** option, an optional keyword that can appear on most intensity / luminosity commands.

When the keyword **range** appears there are an additional two parameters, the low- and high-energy limits to the energy range in Rydbergs. These appear as the second and third numbers on the line (the first number gives the intensity or luminosity). The position of the keyword **range** on the command line does not matter but the order of the numbers on the line does matter. If both range parameters are omitted then the low ($3.040 \times 10^{-9}\,\mathrm{Ryd}$) and high ($7.354 \times 10^6\,\mathrm{Ryd}$) energy limit of the incident radiation field will be substituted. If both energies are specified then the second number must be larger than the first. If only one parameter appears then only the lower limit of the range will be changed and the high-energy limit will be left at its default of $7.354 \times 10^6\,\mathrm{Ryd}$. If the first optional number is negative or the keyword **log** appears then *both* of

the extra numbers are interpreted as logs. If you want to set the lower limit of the range to the low energy limit 3.040×10^{-9} Ryd of the incident radiation field, but want to supply your own upper limit, you can enter 0 followed by the upper limit if you use linear numbers, or -10 followed by the upper limit if you use log numbers (-10 is actually any number $\leq \log(3.040 \times 10^{-9}\,\mathrm{Ryd})$).

If **range total**, or simply **range** or **total**, appears with no parameters then the full energy range considered by the program, 3.040×10^{-9} Ryd to 7.354×10^{6} Ryd, will be used. In this case the number is the total integrated (bolometric) intensity or luminosity.

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

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

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

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

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

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

5.4 Absolute [visual, bolometric] magnitude -2.3

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

$$L_{total} = 3.826 \times 10^{33} \times 10^{(4.75-M_{bol})/2.5}\,[\mathrm{erg\,s^{-1}}]. \tag{5.1}$$

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

$$\nu L_\nu\,(5500\text{Å}) \approx 2.44 \times 10^{35} \times 10^{-M_V/2.5}\,[\mathrm{erg\,s^{-1}}]. \tag{5.2}$$

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

This is a luminosity command.

5.5 Energy density 5e4 K [linear]

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

This is an intensity command.

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

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

The optional second number is the frequency in Rydbergs where $4\pi J_\nu$ is specified. The default is 1 Ryd. In the example above the incident radiation field is specified at 0.1824 Ryd = 5000Å. The frequency can be any within the energy band considered by the code, presently 3.040×10^{-9} Ryd to 7.354×10^{6} Ryd. If the energy is less than or equal to zero then it is interpreted as the log of the energy in Rydbergs, and as the linear energy itself if positive.

This is an intensity command.

5.7 Intensity 8.3 [range, linear]

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

$$4\pi J = \int_{\nu 1}^{\nu 2} 4\pi J_\nu \, d\nu \, [\mathrm{erg\,cm^{-2}\,s^{-1}}]. \tag{5.3}$$

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

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

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

```
# a B star with an intensity roughly the Habing 1968 radiation field
blackbody 3e4 K
intensity -2.8, range 0.44 to 1 Ryd
# remove all H-ionizing radiation
```

```
extinguish by 24, leakage = 0
```

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

This is an intensity command.

5.8 ionization parameter = −1.984

The ionization parameter U is the dimensionless ratio of hydrogen-ionizing photon to total-hydrogen densities evaluated at the illuminated face of the cloud. It is defined as

$$U \equiv \frac{Q(\mathrm{H})}{4\pi r_o^2 n(\mathrm{H})\, c} \equiv \frac{\Phi(\mathrm{H})}{n(\mathrm{H})\, c} \tag{5.4}$$

(AGN3, equation 14.7, page 357). Here r_o is the separation [cm] between the center of the source of ionizing radiation and the illuminated face of the cloud, n(H) [cm^{-3}] is the total[1] hydrogen density (ionized, neutral, and molecular) at that point, c is the speed of light, Q(H) [s^{-1}] is the number of hydrogen-ionizing photons emitted by the central object, and Φ(H) [cm^{-2} s^{-1}] is the surface flux of ionizing photons at the illuminated face. The number on the command line is interpreted as the log of U unless the keyword **linear** appears. The ionization parameter is a useful quantity in plane-parallel, low-density, constant-density, models, because of homology relations between models with different photon and gas densities but the same ionization parameter (see Davidson, 1977).

This is an intensity command.

5.9 L(nu) = 24.456 [at .1824 Ryd]

This sets the monochromatic luminosity L_ν [erg s^{-1}Hz^{-1}] of the central object. The first number is the log of the luminosity. The optional second number is the frequency in Rydbergs where L_ν is specified. The default is 1 Ryd. In the example above the incident radiation field is specified at $0.1824\,\mathrm{Ryd} = 5000$Å. The frequency can be any within the energy band considered by the code, presently 3.040×10^{-9} Ryd to 7.354×10^{6} Ryd. If the energy is less than or equal to zero then it is interpreted as the log of the energy in Rydbergs, and the linear energy itself if positive.

This is a luminosity command.

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

5.10 luminosity 38.3 [solar, total, range, linear]

The number is the log of the integrated luminosity[2] emitted by the central object into 4π sr,

$$L = 4\pi R_{star}^2 \int_{\nu_1}^{\nu_2} \pi F_\nu \, d\nu [\mathrm{erg\,s^{-1}}]. \tag{5.5}$$

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

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

The following are examples of the luminosity command.

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

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

# both are a total luminosity 1000 times solar since
# solar specifies the total luminosity relative to the sun
luminosity solar 3
luminosity linear solar 1000

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

This is a luminosity command.

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

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

The optional second number is the energy (Ryd) where $4\pi\,\nu J_\nu$ is specified. The default is 1 Ryd. In the example above the incident radiation field is specified at $0.1824\,\mathrm{Ryd} = 5000\mathring{A}$. The energy can be any within the energy band considered by the code, presently 3.040×10^{-9} Ryd to 7.354×10^6 Ryd. If the energy is less than or equal to zero it is interpreted as the log of the energy in Rydbergs. It is the linear energy if it is positive.

This is an intensity command.

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

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

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

The optional second number is the energy (Ryd) where L_ν is specified. In the example above the incident radiation field is specified at 0.1824 Ryd = 5000Å. The frequency can be any within the energy band considered by the code, presently 3.040×10^{-9} Ryd to 7.354×10^{6} Ryd. If the energy is less than or equal to zero, it is interpreted as the log of the energy in Rydbergs, and the linear energy if positive.

This is a luminosity command.

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

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

$$\Phi(\mathrm{H}) \equiv \frac{Q(\mathrm{H})}{4\pi r_o^2} \equiv \frac{R_{star}^2}{r_o^2} \int_{\nu_1}^{\nu_2} \frac{\pi F_\nu}{h\nu}\, d\nu [\mathrm{cm}^{-2}\ \mathrm{s}^{-1}] \tag{5.6}$$

and is proportional to the optical depth in excited lines, such as the Balmer lines (Ferland et al., 1979; AGN3). The range option can be used to change the default energy range, given by the values of ν_1 and ν_2.

This is an intensity command.

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

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

$$Q(\mathrm{H}) = 4\pi R_{star}^2 \int_{\nu_1}^{\nu_2} \frac{\pi F_\nu}{h\nu}\, d\nu. \tag{5.7}$$

The default energy range is 1 Ryd to 7.354×10^{6} Ryd and the **range** option can be used to change the energy bounds ν_1 and ν_2. The photon flux (the number of photons per unit area of cloud surface) can be specified with the **phi(H)** command[3].

This is a luminosity command.

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

This specifies the intensity of a second radiation field (referred to as the current radiation field) relative to the intensity of the previous radiation field. The ratio of the intensities $4\pi J_\nu$ (erg

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

cm^{-2} s^{-1} Hz^{-1}) of the current to the previous radiation field is given by the first number on the command line. It is assumed to be the linear ratio unless it is less than or equal to zero, in which case it is interpreted as a log. If the keyword **log** appears then the positive number is interpreted as the log of the ratio.

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

```
blackbody 5e4 K         # the big blue bump
ionization parameter -2 # its ionization parameter
table power law          #an alpha =-1 power law
# now set intensity of power law relative to bump at 1 Ryd
ratio 0.001 at 1 Ryd
```

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

$$\frac{f_\nu\left(2\,\mathrm{keV}\right)}{f_\nu\left(2500\,\text{Å}\right)} = \left(\frac{\nu_{2\,\mathrm{keV}}}{\nu_{2500\,\text{Å}}}\right)^{\alpha_{ox}} = 403.3^{\alpha_{ox}}. \tag{5.8}$$

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

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

```
blackbody 5e4 K   # the big blue bump
ionization parameter -2
table power law    # an alpha =-1 power law
ratio alphaox -1.4 # set alpha_ox of the X-ray and UV continua
```

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

$$\alpha_{ox} = -1.32 - 0.088 \times \log\left(\frac{L_o}{10^{28}\,\mathrm{erg\,s^{-1}\,Hz^{-1}}}\right) \tag{5.9}$$

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

$$\alpha_{ox} = -1.11 - 0.111 \times \log\left(\frac{L_o}{10^{27}\,\mathrm{erg\,s^{-1}\,Hz^{-1}}}\right) \tag{5.10}$$

and by Wilkes et al. (1994):

$$\alpha_{ox} = -1.53 - 0.11 \times \log\left(\frac{L_o}{10^{30.5}\,\mathrm{erg\,s^{-1}\,Hz^{-1}}}\right). \tag{5.11}$$

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

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

In general, it is probably better to use the **AGN** command rather than this command.

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

5.16 xi -0.1

Tarter et al. (1969); Krolik et al. (1981); Kallman and Bautista (2001) define an ionization parameter ξ given by

$$\xi = (4\pi)^2 \int_{1R}^{1000R} J_\nu \, d\nu / n(\mathrm{H}) = \frac{F_{ion}}{n(\mathrm{H})\, r_0^2} \approx \frac{L_{ion}}{n(\mathrm{H})\, r_0^2} [\mathrm{erg\ cm\ s^{-1}}] \tag{5.12}$$

where $n(\mathrm{H})$ is the hydrogen density at the illuminated face of the cloud and r_0 is the source - cloud separation. The number is the log of ξ.

The original Tarter et al. (1969) paper defined ξ as the last term in equation 5.12, with L_{ion} defined as including all ionizing radiation. CLOUDY used this definition, integrating over all ionizing photon energies, through version C13.03.

The 1 - 1000 Ryd energy range first appears in Krolik et al. (1981), in the discussion above eqn 2.2a, but they introduce a new variable, F_{ion}, for the luminosity over this range. The XSTAR code (Kallman and Bautista, 2001) defines ξ over 1 - 1000 Ryd.

Beginning with C13.04 the XSTAR definition is used for ξ, for compatibility with that code. That is, we now integrate over 1 - 1000 Ryd. The middle and right terms in equation 5.12 are not equal for a hard SED.

You can easily define your own energy intervals by using the **luminosity**, page 36, or **intensity**, page 35, commands to specify L or $4\pi J$. They both accept the **range** option to change the limits in equation 5.12.

Chapter 6

INCIDENT RADIATION FIELD SHAPE

6.1 Overview

The spectral energy distribution (SED) of the incident radiation field should be specified between the energies of 3.040×10^{-9} Ryd ($\lambda \approx 29.98$ m) and 100 MeV $\approx 7.354 \times 10^{6}$ Ryd. The low-energy region is important for Compton cooling, photoionization from excited states of the elements, free-free heating, H^- heating, and grain heating. The high-energy portion is important for Auger and secondary ionization, Compton heating, and pair production. Energies greater than 100 MeV are not generally important since the Klein - Nishina electron-scattering cross section is small. CLOUDY will complain, but compute the model if possible, if the incident radiation field is not specified over the full energy range. An intensity of zero will be assumed for missing portions of the incident radiation field.

The ***plasma frequency***, given by

$$\nu_{pl} = \left(\frac{n_e q_e^2}{\pi m_e} \right)^{1/2} = 8.978 \times 10^3 \, n_e^{1/2} \, s^{-1} = 2.729 \times 10^{-12} \, n_e^{1/2} \, \text{Ryd}, \tag{6.1}$$

will move into the energy range considered by the code if the electron density is higher than $\sim 10^7 \text{cm}^{-3}$. The incident radiation field below the plasma frequency is totally reflected and does not enter the slab. The code will generate a comment if the plasma frequency occurs within the energy grid.

It is important to plot the incident radiation field using the output of the ***save continuum*** *command to make sure the SED has the expected shape.* It is easy to make a mistake in converting between f_ν, f_λ, νf_ν, and λf_λ when deriving the SED. The **save continuum** command will report the SED in νf_ν or λf_λ units (the two are equivalent).

6.1.1 Isotropic and beamed continua

Most radiation fields are assumed to strike the cloud as a single-directional beam. The beam is assumed to enter the slab normal to the surface, unless this is changed with the **illumination angle** command. Other commands, including **background**, **CMB**, **table HM96**, **table HM05**, **table HM12**, **table KS19**, and **table ISM**, generate continua that are isotropic. Each shape command will indicate whether the SED is isotropic or beamed.

Many spectrometers will automatically remove isotropic emission while conducting an observation. Some isotropic continua, especially the CMB at radio wavelengths, will dominate over local emission. The radiation field used in the calculation is reported by default, although there are two commands that will suppress isotropic emission. The **save continuum** command, section 16.42, includes a **no isotropic** option, described in section 16.42.4, to suppress isotropic emission in that file. The **no isotropic continua report** command described in section 19.11.17 will suppress Isotropic emission in both **save continuum** files and the continuum bands described in Section 2.2.3 *The radiation field integrated over a range of wavelengths*.

6.2 AGN T =1.5e5 k, a(ox) = -1.4, a(uv)=-0.5 a(x)=-1

This produces a multi-component continuum similar to that observed in typical Active Galactic Nucleus (AGN). An example is shown in Figure 6.1. The "Big Bump" component, peaking at ≈ 1 Ryd, is a rising power law with a high-energy exponential cutoff. It is parameterized by the temperature of the bump, the first argument on the command line. It is interpreted as the log of the temperature if it is less than or equal to 10 and the linear temperature otherwise. The temperature of the bump can by optimized by using the **vary** keyword. The second parameter is the X-ray to UV ratio α_{ox}. Note that there is no implicit negative sign in this exponent; typical AGN have $\alpha_{ox} \sim -1.4$, (Zamorani et al., 1981). The third (optional) argument is the low-energy slope of the Big Bump continuum, with the default $\alpha_{uv} = -0.5$ (Elvis et al., 1994; Francis, 1993). The last argument is the slope of the X-ray component with the default $\alpha_x = -1$. Optional parameters can be omitted from right to left.

The full continuum is the sum of two components, as in equation 6.2:

$$f_\nu = \nu^{\alpha_{uv}} \exp(-h\nu/kT_{BB}) \exp(-kT_{IR}/h\nu) + a\nu^{\alpha_x}. \quad (6.2)$$

The coefficient a is adjusted to produce the correct α_{ox} for the case where the Big Bump does not contribute to the emission at 2 keV. If the bump is very hot then it may contribute to the X-rays as well and the resulting continuum will have a more negative α_{ox} than specified. The X-ray power law is only added for energies greater than 0.1 Ryd to prevent it from extending into the infrared, where a power law of this slope would produce *very* strong free-free heating. The Big Bump component is assumed to have an infrared exponential cutoff at $kT_{IR} = 0.01$Ryd.

The last term in equation 6.2 is not extrapolated below 1.36 eV or above 100 keV. Below 1.36 eV the last term is simply set to zero (the bump dominates for these energies). Above 100 keV the continuum is assumed to fall off as ν^{-2}.

The exponential cutoff in the low-energy end of the Big Bump will cause the incident continuum to have zero intensity at very long wavelengths. The code will complain since a zero incident continuum is unphysical, but the model will be computed. You can prevent the continuum from going to zero by including the cosmic background with the **background** command or with the **CMB** command.

We used this command to generate the continuum used in a large atlas of BLR line intensities (Korista et al., 1997a). The specific parameters needed to reproduce that continuum are **AGN 6.00 -1.40 -0.50 -1.0**. Only Kirk Korista can remember all these numbers and the parameters were not explicitly given in the original paper in the format used by the code. The **kirk** option on the **AGN**

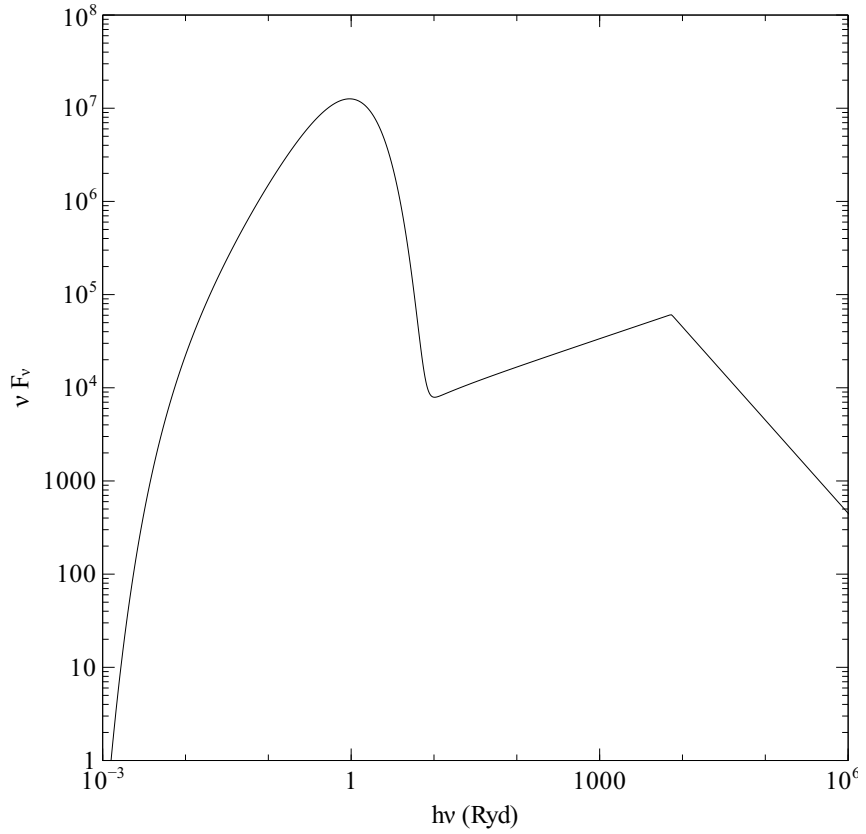

Figure 6.1: The continuum produced by the **AGN** continuum command. The Big Bump peaks at $\approx$ 1 Ryd, while the X-ray power law dominates at high energies. The two components are normalized by the second parameter, the value of α_{ox}.

command will generate that continuum. It is fairly similar to the Mathews and Ferland (1987) continuum but there is much greater flexibility in changing its details.

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

This specifies a radiation field shape and intensity chosen to mimic the cosmic radio to X-ray background (Ostriker and Ikeuchi, 1983, Ikeuchi and Ostriker, 1986, and Vedel et al., 1994). Their ultraviolet continuum shape is an $\alpha = -1$ power-law, with a mean intensity J_ν at 912Å given by

$$4\pi J_\nu\left(912\text{Å}\right) = 4\pi \times 10^{-21}\left(\frac{1+z}{3.5}\right)^4 \quad f[\text{erg Hz}^{-1}\text{cm}^{-2}\,\text{s}^{-1}] \tag{6.3}$$

where z is the redshift and f an optional scale factor entered as the second parameter. Its default value is $f = 1$, and $z = 0$ (i.e., now) is assumed if no redshift is entered. Judging from Bechtold et al. (1987), Bajtlik et al. (1988), and Vedel et al. (1994), f is confidently known to be within an order of magnitude of unity.

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

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

$$T_{CMB} = T_o\,(1+z)\ \text{[K]} \tag{6.4}$$

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

The **table HM96** command uses a more sophisticated form of the energetic continuum, but only at a redshift of $z = 2$. The **table HM05** and **table HM12** commands work at any redshift.

This command does not add the cosmic-ray background. This should be included if the simulation is to extend into molecular gas and is done separately with the **cosmic ray** command.

This radiation field is assumed to be isotropic rather than beamed.

6.4 Blackbody t=e5 K [linear, log, luminosity]

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

```
black body, Temp = 1e6 K
```

which is equivalent to

```
black 1000000
```

or

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

6.4.1 Blackbody, t = 1e5 K, disk = 1e3 K

This command produces a multi-color blackbody as if due to an accretion disk with inner and outer temperatures T_{in} and T_{out} as in Mitsuda et al. (1984). The temperatures can be in either order; the code will automatically take the greater temperature as T_{in}.

Neither the luminosity nor the intensity is set by this command.

6.4.2 Peter Martin's blackbody luminosity options

The luminosity of the black body can be specified with command-line options added by P.G. Martin. If the luminosity is specified with any of these options then it must not also be specified with another luminosity command for this radiation field. The keywords that can appear on the line are given in the following subsections.

6.4.3 Blackbody 5, luminosity = 38.

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

This is a luminosity command.

6.4.4 Blackbody 5, radius = 10.

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

This is a luminosity command.

6.4.5 Blackbody 5e4 K, energy density = 500K.

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

$$T_u \equiv \left(\frac{u}{a}\right)^{1/4} [\mathrm{K}] \tag{6.5}$$

where a is the Stefan's radiation density constant.

Both temperatures are interpreted as logs if they are less than or equal to 10 or the keyword **log** appears, and linear otherwise. They will be interpreted as the linear temperature rather than as a log if the keyword **linear** appears. Note that if the **linear** key is used then both temperatures are interpreted as linear. Similarly, if the keyword **log** appears, both temperatures are interpreted as log. Note also that the cosmic background radiation should also be included if $T_u \leq 2.725(1+z)K$. CLOUDY will complain, but compute the model, if the energy density of the incident continuum corresponds to a temperature less than the present energy-density temperature of the universe.

This is an intensity command.

6.4.6 Blackbody, t = 5e4 K, STE

The keyword **STE**[1] (note that a space must appear before "STE") with no second number is equivalent to the **energy density** option with T_u set to the color temperature of the radiation field. This is a quick way to check that ionization and level populations go to strict thermodynamic equilibrium in the high radiation-density limit.

This is an intensity command.

6.4.7 Blackbody, t = 1e5 K, dilution factor = -14

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

$$W \equiv \frac{J_\nu}{B_\nu} \approx \frac{\pi R_{star}^2}{4\pi r_o^2}, \tag{6.6}$$

where R_{star} is the radius of the star and r_o is the separation between the illuminated face of the cloud and the center of the star. The approximation on the RHS assumes that $R_{star} \ll r_o$. The dilution factor can be entered either directly or as a log. Any number > 0 will be interpreted as linear, while any number ≤ 0 will be interpreted as log. The example above is a rough approximation of the radiation field within a typical planetary nebula.

This is an intensity command.

6.5 Bremsstrahlung, temp = 8

The incident radiation field will be optically thin pure hydrogen bremsstrahlung emission. The shape is given by

$$f_\nu \propto \nu^{-0.2} \exp(-h\nu/kT) \quad [\mathrm{erg\,cm^{-2}s^{-1}Hz^{-1}}]. \tag{6.7}$$

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

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

A more realistic radiation field could be produced by combining the **coronal equilibrium** command with the **save transmitted continuum** command to generate a continuum which can be read in with the **table read** command.

6.6 CMB [redshift 1000]

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

$$T_{CMB} = T_o\,(1+z) \quad [\mathrm{K}] \tag{6.8}$$

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

6.6.1 The time option

The keyword **time** will follow the time-dependent recombination of an evolving universe. This is now being developed by Ryan Porter and will be described in Porter et al. (in preparation).

6.6.2 The density option

The keyword **density** will also specify the density at the redshift.

This CMB radiation field is assumed to be isotropic rather than beamed.

6.7 Extinguish column = 23, leak = 0.05, low =4 Ryd

This command will modify the shape of the incident SED by extinction due to photoelectric absorption by a cold neutral slab with column density given by the first argument (entered as a log, cm^{-2}). This extinction is applied *after* the radiation field has been fully generated and normalized to the correct intensity. The form of the extinction is a simple power-law fit to the absorption curves calculated by Cruddace et al. (1974). The extinguished continuum f'_ν is related to the initial continuum f_ν by

$$f'_\nu\,(\nu \geq 1\,\mathrm{Ryd}) = f_\nu \left\{ \eta + (1-\eta) \exp\left(-6.22 \times 10^{-18}\, \nu_{Ryd}^{-2.43}\, N\,(\mathrm{H}) \right) \right\} \tag{6.9}$$

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

The optional second number is the fractional leakage η through the absorber (see Ferland and Mushotzky, 1982). This has a default value of 0, i.e., no leakage. The leakage is interpreted as a

log if it is negative and linear otherwise. If unexpected or unphysical results occur when the **extinguish** command is given then it is likely that nearly all ionizing radiation has been attenuated. A plot of the generated continuum (with the **plot continuum** command) may prove interesting. The code will stop if the leakage is greater than 1.0 (100%).

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

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

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

This command should only be used as a quick test. A more physical way to extinguish the continuum would be to actually transmit it through a model of the absorbing slab, save that continuum with the **save transmitted continuum** command, then use this filtered continuum with the **table read** command.

6.7.1 Extinguish optical depth 1.2, [options]

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

6.8 Interpolate [ν(Ryd) or log ν(Hz)], log(f_ν)

This command will be maintained for backwards compatibility, but you should consider using the **table SED** command, described on page 50, which does the same thing but with much greater flexibility.

The shape of the incident radiation field will be interpolated from the table of points given by this command. The interpolation uses straight lines in log-log space. There is no limit to the number of ordered pairs of points that can be entered. Lines starting with the keyword **continue** can be used to continue entering values after the initial **interpolate** line is filled.

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

The numbers are ordered pairs giving the energy and intensity. The first number within each ordered pair of points is *either* the energy in Rydbergs (linear *or* as a log) or the log of the frequency (in Hertz). CLOUDY assumes that the log of the energy in Rydbergs was entered if the first number is negative; that the log of the frequency (Hz) was entered if the first number is

greater than 5; and linear Rydbergs otherwise. Any of the three styles can be chosen but it must be used consistently within the command. If the first energy is entered as zero then it is interpreted as the low-energy limit of the code. In this case the remaining energies will be interpreted as linear Rydbergs if the second number is positive and the log of the energies in Rydbergs if it is negative. An energy of zero Rydbergs is not allowed (except for the first), and the energies must be in increasing order.

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

Heads up! The intensity is J_ν *and not* νJ_ν. You should check that this command has produced the SED you expect by creating as plot by using the output from the **save incident continuum** command described on page 205.

The **interpolate** command can be freely mixed with other shape commands and a total of up to 100 **interpolate** and **table** commands can be entered.[2] Note that **table** and **interpolate** are actually two forms of the same command so that they store information in the same arrays. The total number of **table** and **interpolate** commands entered together cannot exceed 100.

As an example, the following approximates a metal-poor 4.5e4 K stellar atmosphere. The energies are entered in Rydbergs:

```
# following is 45000 K atmosphere from Shields and Searle
interpolate (0.00001 -11.106) (.58 -1.5792) (.99 -1.44) (1.01 -1.7018)
continue  1.8 -1.905) (1.81 -1.939) (2.57 -2.208) (2.59 -2.247) (3 -2.3994)
continue (3.02 -2.8193) (3.49 -2.9342) (3.51 -4.143) (3.99 -5.582)
continue (4.01 -6.3213) (6 -9.9) (10 -17.3) (20 -30) (1e7 -30)
q(h) = 52.778151
```

Note that the continuum should be specified between 3.040×10^{-9} Ryd and 7.354×10^{6} Ryd even if the intensity is small. If it is not fully specified then a warning will be issued and a model computed with the unspecified continuum set to zero. As a further note, it is important that the continuum be physically correct. For instance, stellar model atmospheres predict almost no X-ray emission while real OB stars *are* X-ray sources (although neglecting X-rays for these stars is generally a safe approximation). See page 63 below for a more detailed discussion.

This command sets the SED by specifying f_ν *points.* Other commands may use νf_ν or λf_λ units (the two are equivalent). You should plot the final SED used by CLOUDY with the **save continuum** command to make sure the SED has the expected shape. It is easy to make a mistake in converting between f_ν, f_λ, νf_ν, and λf_λ when deriving the SED. The **save continuum** command will report the SED in νf_ν or λf_λ units.

6.9 Laser, frequency = 3.5 Ryd [rel width 0.02]

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

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

The optional second number changes the relative width of the laser. The relative width is the ratio dE/E where dE is the half width of the laser. The laser will only be active within $\pm dE$ of E. The code would return an error condition if dE is so small that the laser does not happen to be within $\pm dE$ of E as evaluated by the code's energy mesh. The fractional width probably should not be made smaller than roughly 0.01 but the code does not protect against too small a value of dE.

Another way to make a laser is to save out a transmitted radiation field using the **save transmitted continuum** command, edit this file, and, by hand, increase the intensity of the continuum at particular cells. This is described where the **table read** command is defined.

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

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

$$f_\nu = \nu^{+\alpha} \exp\left(-\nu/\nu_{high\,cut}\right) \exp\left(-\nu_{low\,cut}/\nu\right) [\mathrm{erg\,cm^{-2}s^{-1}Hz^{-1}}]. \tag{6.10}$$

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

If the keyword **Kelvin** appears then both cutoff energies are interpreted as temperatures in Kelvin rather than energies in Rydbergs. The temperature is a log if it is less than or equal to 10 or the keyword **log** appears, and the linear temperature itself otherwise. If you do not use the keyword **Kelvin**, the cutoffs will always be interpreted as linear quantities.

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

It is *much* better to use the **interpolate** command and enter physically reasonable low-energy and high-energy radiation fields. There is a special version of the command, **table power law** for entering a well-behaved power-law continuum.

6.11 Table SED command

The **table SED** command uses an external data file that specifies a component of the incident radiation field as a table of points. The name of the file containing the SED data is given in

quotation marks. For example:
table SED "cf.sed"
The SED files described in this section can be used in this way, as can SED files you create. Details on individual SED files included in the distribution are given below.

6.11.1 SED file format

Each line must contain exactly one pair of numbers, the first the photon energy, and the second a measure of its intensity. By default the SED data have the photon energy given in Rydberg and the intensity specified as J_ν, although both can be changed. The photon energies must be entered either in strictly monotonically increasing or decreasing order.

The SED data can end with the end of file, or with a line containing a field of stars, like "***" (a minimum of three stars is required). This last option allows supplemental information to be stored after the end of data.

Optional keywords can be placed on lines giving the pair of numbers. The following is an example of a SED file:

```
#  Simple broken powerlaw
1.4   1e11 nuFnu units kev
19.5  3e11
255   6e11
**********

Anything after the field of stars is ignored.
```

In this example the **nuFnu** option indicates that the intensities are specified as νf_ν rather than the default f_ν. The **units** keyword changes the units of the photon energy. Lines after the ending field of stars are used to document the file. The first line is a comment since it starts with the sharp sign.

6.11.2 Optional keywords within the SED data file

The entire SED data file is scanned for the following optional keywords. Optional keywords can be placed anywhere in the file following the pair of numbers, and more than one keyword can occur on a line.

Comments. Any line (or part of a line) starting with "#" is ignored up to the end of the line.

Photon energy units. By default the photon energy is given in Rydbergs. The energy or wavelength units can be changed by including the **units** option, followed by one of the units given in Section 16.42.5. Both the keyword **units** and one of these units must appear if you wish to use other units. An example of the use of the **units** option in an SED file is shown in Section 6.11.1.

Intensity units. The flux in the SED file is given in relative f_ν units. If the keyword **nuFnu** is present, the flux can be given in relative νf_ν units (which is the same as λf_λ). If the keyword **Flambda** is present, the flux can be given in relative f_λ units. The values are not used to set the continuum brightness, only the shape. The brightness is set with the intensity or luminosity command described in the Chapter starting on page 33.

Extrapolating a low-energy tail is not done by default. If the table does not extend over the full energy bandwidth of the code then points outside the table are set to zero intensity. If the keyword **extrapolate** is present then the SED will be extrapolated to the low-energy limit of the code.

Table 6.1: Legacy SED keywords

keyword	filename
table AKN120	`akn120.sed`
table cooling flow	`cool.sed`
table Crab	`CrabHester.sed`
table Crab Davidson	`CrabDavidson.sed`
table Rubin	`Rubin.sed`
table XDR	`XDR.sed`

Table 6.2: Akn 120 Continuum

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

6.11.3 Available SED datasets

The primary documentation for the SED files included in the distribution is the set of files in the `data/SED` directory. Consult the `ReadMe.txt` file to find out more about what is available.

Many of the SEDs listed in the following subsections were available in CLOUDY versions C10 and before. To maintain backwards compatibility the keywords listed in Table 6.1, without the SED keyword, will use the file indicated.

6.11.4 Table SED "akn120.sed"

If the keyword **akn120** appears then the continuum summarized by Peterson (private communication) is used. The continuum is described by the observed flux at Earth [erg cm^{-2} s^{-1} Hz^{-1}] and is given in Table 6.2.

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

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

6.11.5 Table SED "cool.sed"

This generates the continuum shape described by Johnstone et al. (1992) and used in Ferland et al. (1994, 2002). It is a co-added series of Raymond-Smith collisional-equilibrium continua and was chosen to represent the spectrum at a point within a typical cooling flow.

6.11.6 Table SED "CrabHester.sed"/"CrabDavidson.sed"

This generates the SED of the synchrotron-emitting regions of the Crab Nebula. This is the net observed continuum, originating in both the pulsar and nebula, and not the pulsar continuum alone.

CrabHester.sed is the SED derived by Atoyan and Aharonian (1996) and shown in Hester (2008) is used. If **CrabDavidson.sed** appears then the continuum summarized by Davidson and Fesen (1985) is used. These two SEDs, shown in Figure 6.2, have substantial differences in the UV/FUV.

According to Davidson and Fesen (1985), the total luminosity of the Crab is $L_{tot} = 10^{38.14}$ erg s^{-1}. The Crab SED could be generated by combining the commands

```
luminosity (total) 38.14
table SED "CrabDavidson.sed"
```

6.11.7 Table SED "Mrk509.sed"

This is taken from Figure 3 of Kaastra et al. (2011).

6.11.8 Table SED "NGC5548.sed"

This is taken from the green line in Figure 10 of Mehdipour et al. (2015).

6.11.9 Table SED "pl1.sed"

This produces an $f_\nu \propto \nu^{-1}$ power-law continuum across the full spectral range considered by the code. This should only be used for testing since it may produce unphysically bright radio or γ-ray emission. See the discussion of the **power law** command on page 50 for more details, and a safe way to get a power-law continuum.

6.11.10 Table SED "Rubin.sed"

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

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

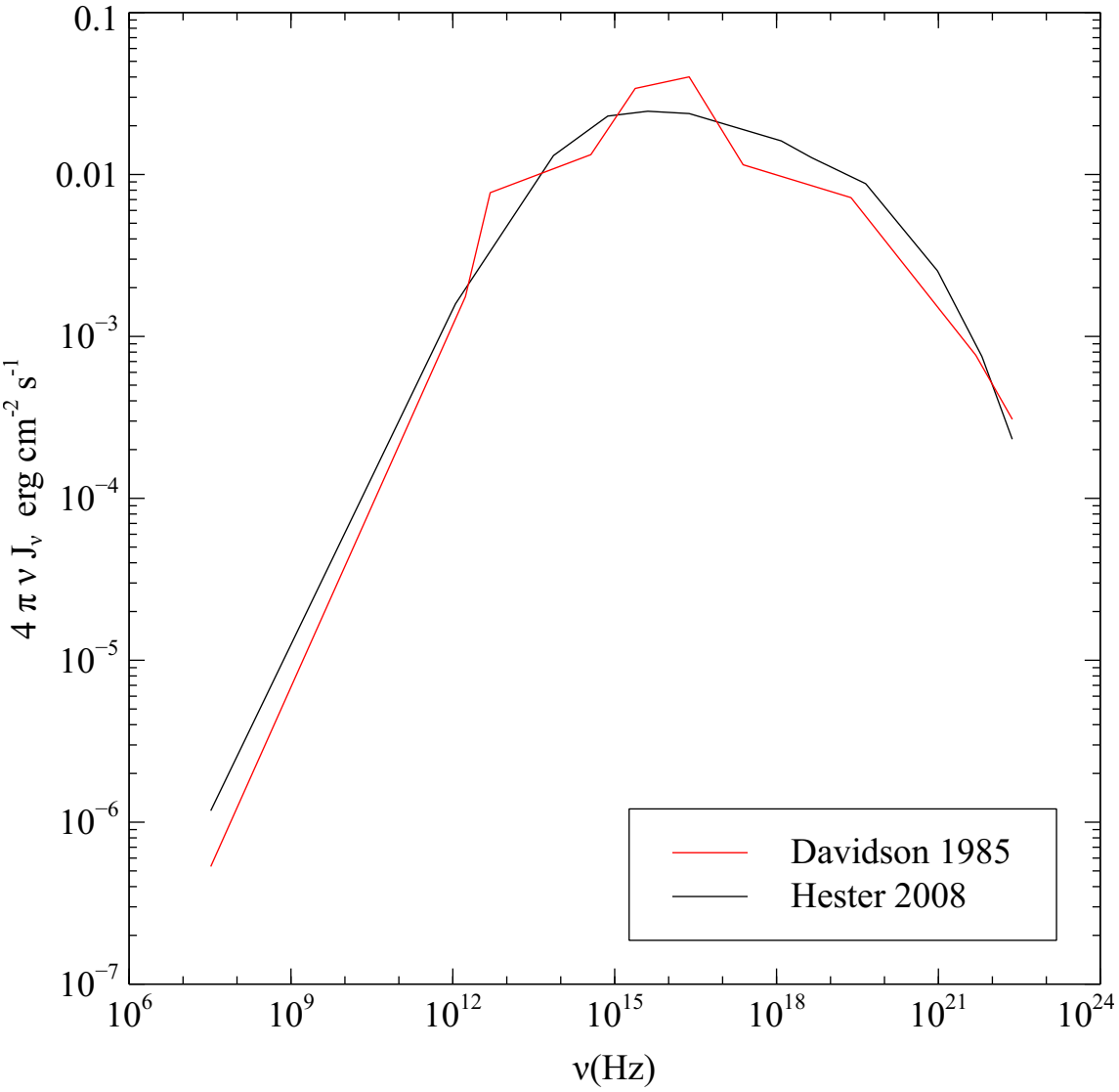

Figure 6.2: The SEDs produced by the **table Crab** commands are compared. The red line is the SED from Davidson and Fesen (1985) and was the default used by CLOUDY through versions C10. The black line is the current default, derived by Atoyan and Aharonian (1996) and shown in Hester (2008)

Table 6.3: AGN Continuum

ν(Ryd)	log (F_ν)	slope
1.00(−5)	−3.388	+2.50
9.12(−3)	4.0115	−1.00
0.206	2.6576	−0.50
1.743	2.194	−1.00
4.130	1.819	−3.00
26.84	−0.6192	−0.70
7.35(+3)	−2.326	−1.67
7.40(+6)	−7.34	−1.12

6.11.11 Table SED "XDR.sed"

This generates the X-ray SED described in Maloney et al. (1996),

$$f_\nu = f_0 \times \left(\frac{E}{100\,\mathrm{keV}} \right)^{-0.7} [\mathrm{erg\,cm^{-2}\,s^{-1}\ Hz^{-1}}], \tag{6.11}$$

over the energy range 1 – 100 keV. The radiation field has negligible intensity outside this range.

6.12 Other Table commands

6.12.1 Overview

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

6.12.2 Table AGN

If the keyword **AGN** appears (note the presence of a leading space) then a continuum similar to that deduced by Mathews and Ferland (1987) will be used. The continuum is meant to be similar to typical radio quiet active galaxies. The points used to describe this continuum are given in Table 6.3.

This radiation field differs from the Mathews and Ferland (1987) SED only in that the continuum is assumed to have a sub-millimeter break at 10 microns. For wavelengths longer than 10 μm the continuum is assumed to have a slope $f_\nu \propto \nu^{+2.5}$, appropriate for a self-absorbed synchrotron continuum (Rybicki and Lightman, 1979). Note that this represents a typical observed continuum, and may not be directly related to the continuum actually striking BLR gas (Korista et al., 1997b).

The energy of the sub-millimeter break is not well determined observationally but has a major impact on high density, high ionization parameter models, as discussed by Ferland and Persson (1989), Ferland et al. (1992), and Ferland (1999a). The energy of the infrared break can be

adjusted with the **break** keyword. The break can be adjusted between the limits of 0.2 Rydberg and 3.040×10^{-9} Ryd by entering the keyword **break** followed by a number specifying the energy of the break. The number is interpreted as the log of the energy in Rydbergs if it is negative and as linear Rydbergs if positive. It is interpreted as the linear wavelength of the break in micron if the keyword **micron** also appears. If no number appears, but the keywords **no break** does, then a break at the low-energy limit of the code (3.040×10^{-9} Ryd) is assumed. The following shows equivalent ways of generating a continuum with a break at 10 microns;

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

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

6.12.3 Table Draine [factor=1.7]

This enters the galactic background radiation field given by equation 23 of Draine and Bertoldi (1996). The radiation field is only defined over a very narrow wavelength range so it is only appropriate for certain very simple PDR calculations. It is shown in Figure 6.3 along with the radiation field produced by the **table ism** command. This command specifies both the shape and intensity of the continuum.

The optional scale factor changes the intensity of the continuum. By default it is a linear scale factor but the keyword **log** changes this.

This radiation field is defined up to an energy very close to 1 Rydberg. It may spill over into the hydrogen-ionizing continuum for certain choices of the continuum-bin resolution. Add the **extinguish** command to insure that H-ionizing radiation is extinguished if this is needed. The code will print a comment is this is not done.

This continuum should only be used for restricted tests since the vast majority of the electromagnetic spectrum is undefined. It was created to compute the Leiden PDR test cases (the `pdr_leiden*` simulations in the test suite) and the paper by Röllig et al. (2007).

This radiation field is isotropic. This is an intensity command.

6.12.4 Table HM96 [factor=-1]

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

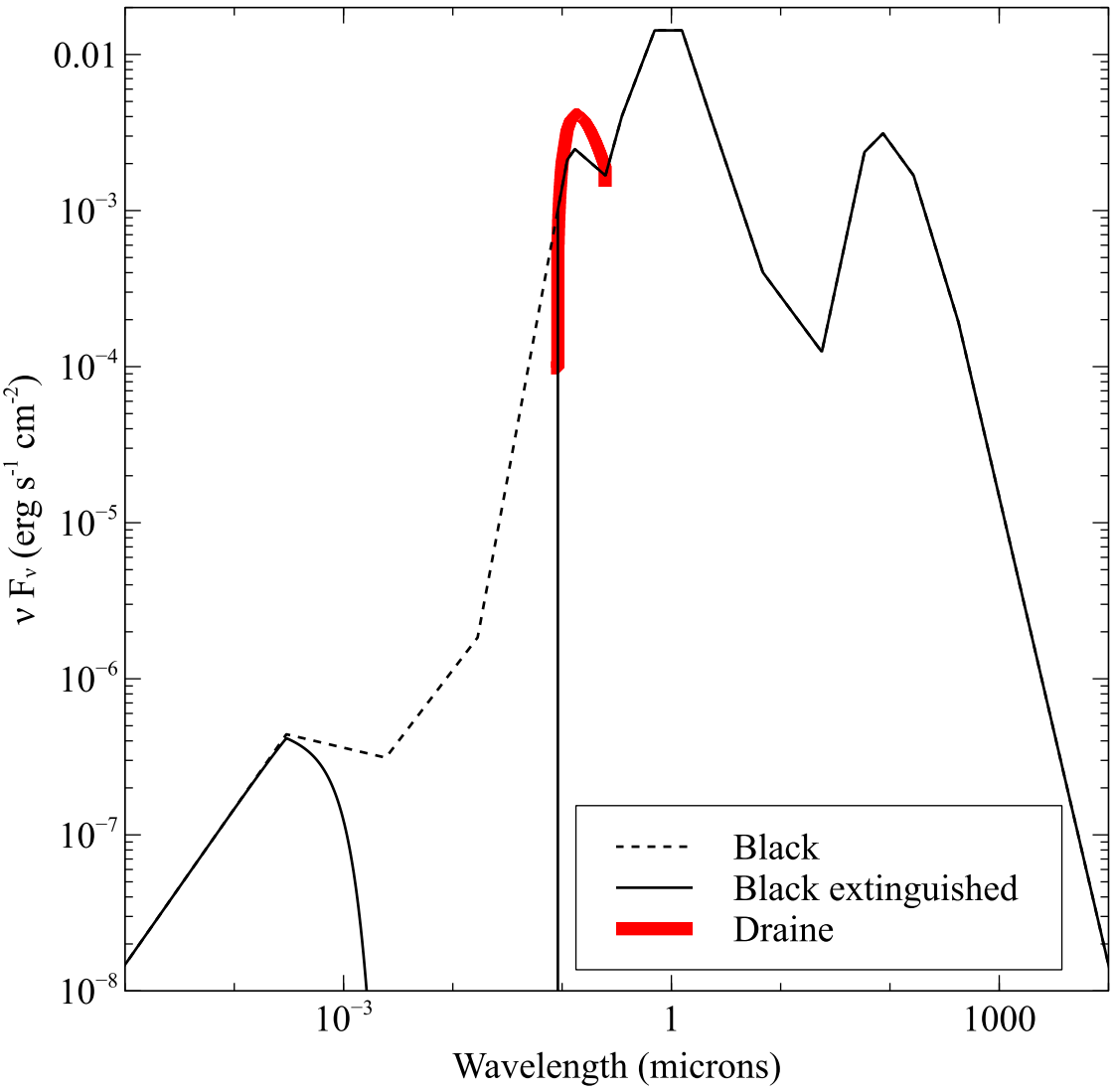

Figure 6.3: The SED produced by the **table ISM** command is the lighter line. The infrared cirrus is the peak at $\lambda \sim 100\,\mu$m and starlight dominates at shorter wavelengths. The points just shortward of the Lyman limit ($0.0912\,\mu$m) are interpolated—actually it is thought that interstellar extinction removes most of this continuum. The dashed line shows the interpolated SED and the solid line shows the effects of absorption introduced by adding the **extinguish** command. The heavy red line is the **table Draine** continuum.

This radiation field is isotropic. This is an intensity command.

6.12.5 Table HM05 redshift 2.21 [quasar; factor=-1]

This enters the Haardt & Madau (2005 private communication) radiation field at any redshift $0 \leq z \leq 9.479$. It interpolates on tables kindly provided by Francesco Haardt. There are two forms of the radiation field - the default includes both quasars and galaxies. If the keyword **quasar** occurs then a radiation field considering only quasars is used. The cosmic microwave background is not included - use the **CMB** command to add this component. Note that this table specifies both the shape and intensity of the radiation field. There is an optional multiplicative scale factor to change the intensity. If the scale factor is less than or equal to zero then it is interpreted as the log of the scale factor.

The radiation field is isotropic. This is an intensity command.

6.12.6 Table HM12 redshift 2.21 [factor=-1]

This enters the Haardt and Madau (2012) radiation field at any redshift $0 \leq z \leq 15.93$. It works the same way as the **table HM05** command described above, except that the keyword **quasar** is not supported because tables for the quasar-only case do not exist for this data set.

6.12.7 Table ism [factor = 0.7]

The local interstellar radiation field is generated with the keyword **ism**. This uses Figure 2 of Black (1987) to represent the *unextinguished* local interstellar radiation field (see Figure 6.3). This command specifies *both* the shape and luminosity of the radiation field. The continuum generated by CLOUDY is exactly that given by Black except that the radiation field between 1 and 4 Ryd is interpolated from the observed or inferred values. Actually it is thought that this part of the radiation field is heavily absorbed by gas in the ISM so that little 1 to 4 Ryd radiation exists, at least in the galactic plane. Such absorption can be introduced with the **extinguish** command.

The **table ism** command also specifies the intensity of the incident radiation field since this is directly observed. There is an optional scale factor to change the intensity of the entire radiation field. It is the log of the scale factor if less than or equal to zero and the scale factor itself if positive. The actual numbers used by CLOUDY to interpolate on Black's table are given in Table 6.4. The frequencies are in Hz, and the product νf_ν in erg cm^{-2} s^{-1}.

This does not include the cosmic microwave background so that it can be used at any redshift. Use the **CMB** command to add this component.

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

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

This radiation field is isotropic. This is an intensity command.

Table 6.4: ISM Radiation Field

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

Table 6.5: Power Law Continuum

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

6.12.8 Table KS19 redshift 2.21 [factor=-1] [Q=18]

This enters the Khaire and Srianand (2019) extragalactic background radiation at any redshift $0 \leq z \leq 15$. It works very similar to the **table HM12** command described above, except that there is a second optional parameter giving the integer Q parameter for the requested SED (see Table 1 of Khaire and Srianand, 2019). When omitted, it will default to $Q18$, the fiducial model recommended by Khaire and Srianand (2019). If the Q parameter is included, then adding the scaling factor is mandatory and the scaling factor should be entered before the Q parameter. Valid values for the Q parameter are between 14 and 20.

The radiation field is isotropic. This is an intensity command.

6.12.9 Table power law [spectral index -1.4, low =.01, hi =20]

This produces a power-law continuum that is well behaved at both the high and low energy ends. The default shape, assumed when no numbers occur on the command line, is the form $f_\nu \propto \nu^\alpha$. Here $\alpha = -1$ for the spectral midrange between 10 microns and 50 keV, and the continuum has slopes $f_\nu \propto \nu^{5/2}$ at lower energies (appropriate for self-absorbed synchrotron, eq 6.54, p.190, Rybicki and Lightman, 1979) and $f_\nu \propto \nu^{-2}$ at higher energies. Note that much of the X-ray literature will use an implicit negative sign in defining the energy index, as in equation 13.2 of AGN3. CLOUDY never uses implicit negative signs so a typical AGN will have an energy index of roughly −1.5. Table 6.5 summarizes the continuum that is produced by this command if no other parameters are specified.

Three optional numbers may appear on the command line. The first number sets the slope of the mid-range spectral component (infrared to X-ray) and has a default of -1 ($f_\nu \propto \nu^{-1}$). The next two numbers adjust the energy limits of the mid-range spectral component. Their default units are Rydbergs but the keyword **microns** will change the units to microns *for the first energy only.* The second number is the energy (in Rydbergs) of the infrared break. The default is 0.009115 Ryd (10 microns). If this second number is zero then the low energy limit to the continuum (3.040×10^{-9} Ryd) will be used. The number is interpreted as the log of the energy in Rydbergs if it is negative and linear otherwise. Note that, with no infrared break, free-free heating will probably be significant for denser clouds. A power-law continuum with a low energy break at 1 micron would minimize this heating and could be generated with the command

```
# a power-law with index -1 and 1 micron break
table power law slope -1, 1 micron break
```

The following uses the default power law

```
# another example, a power-law with index -1
# and 10 micron break, (the default)
table power law slope -1
```

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

6.12.10 Table read "contin.txt" [scale [= 0.5]]

This reads in the radiation field predicted from a previous CLOUDY calculation. The name of the file containing the spectrum predicted from the first calculation continuum must be enclosed in a pair of double quotes. The first calculation saves the radiation transmitted through a cloud with the **save transmitted continuum** command described in Section 16.42.22. Subsequent calculations use the **table read** command to include this field.

The transmitted spectrum of the first model will include all transmitted radiation, including line emission, and will write that out as a single spectral energy distribution. When the second model reads that in, it will have no knowledge of what was what in the first model, apart from the outer radius of that model (assuming a radius was specified in that model). It will treat it as an incident SED without further specifics. So the [O III] 5006.84 flux reported by the second model will be that produced by the second model only, but the 5006.84 photons from the first model will be included "anonymously" in the incident SED.

The **save transmitted continuum** command produces a file containing the frequency in Rydbergs and the transmitted continuum νf_ν [erg cm^{-2} s^{-1}]. This continuum is the sum of the attenuated incident continuum and the fraction of the cloud's diffuse emission that is transmitted in the outward direction. The first lines of the file contain header information and must not be deleted.

CLOUDY checks that the photon energies contained in the **save transmitted continuum** file are exactly that same as the photon energies used in the code's internal arrays. This is because the contents of the files are simply placed into these arrays with no interpolation. You should not try to change the photon energies.

The **table read** command can be freely mixed with all of the other radiation field shape commands. Any number of **table read** commands can be entered.[3] The **save continuum** file must have been produced by the same version of CLOUDY.

By default, this command does not set the intensity of the radiation field. If the **scale** option is not given, the intensity of the radiation field must be set explicitly, using any of the intensity or luminosity commands.

If the **scale** option is given, the intensity scale can be specified relative to the intensity of the previous calculation. As with some other **table** commands, the value specified is interpreted as a log scale if it is non-positive and linear otherwise. If no number is given, the scale factor defaults to unity.

If the model that generated the transmitted spectrum and the model that reads it both set a radius, then spherical dilution will implicitly be done. The scale factor will be a multiplicative factor on top of that, i.e. the transmitted spectrum will be multiplied by a factor $(r_{\rm out}/r_{\rm in})^2 s$ where $r_{\rm out}$ is the outer radius of the first model, $r_{\rm in}$ is the inner radius of the second model, and s is the scale factor. A warning will be printed if $r_{\rm in} < r_{\rm out}$, but the spherical correction will still be done (which would make the radiation field more intense). If the first model set a radius, but not the second (or *vice versa*) a caution will be printed and no spherical correction will be done (but the scale factor will still be applied).

The following gives an example of first creating a file containing the transmitted radiation field then using this file as the incident radiation field in a second calculation.

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

Now use this continuum in a second calculation:

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

The output continuum can also be diluted by a factor of 100 compared to the first calculation:

```
table read file = "absorber.txt" scale -2
hden 9
```

6.12.11 Table trapezium

This enters the radiation field of the Orion Trapezium stars.

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

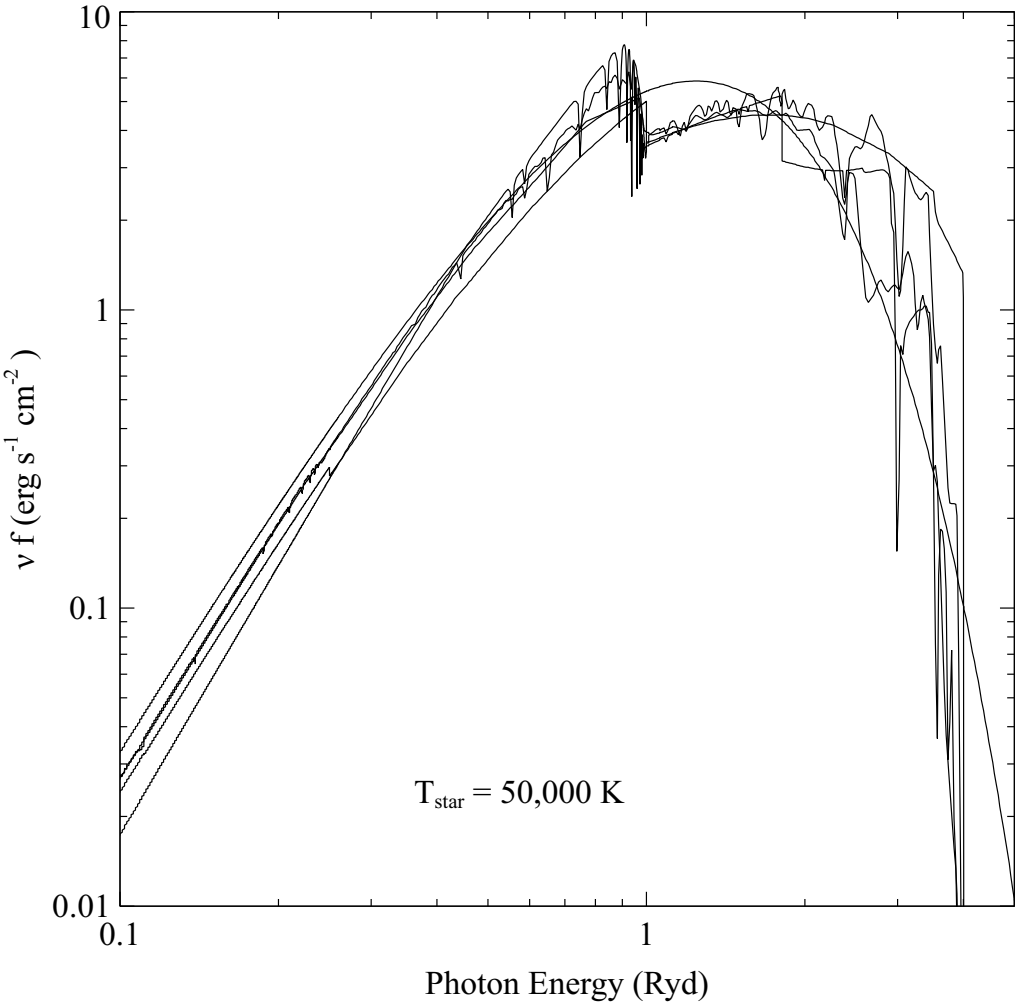

Figure 6.4: This figure shows the emergent radiation field predicted by five 50 kK stars included with the code. The smoothest is the blackbody, and the Kurucz (1991) and Rauch (1997) atmospheres show the most structure. stars

6.13 Table stars command

Several sets of emergent SEDs from stellar atmosphere calculations are accessible. The command has several sub-keywords that indicate which set of atmospheres to use. The syntax of the command varies depending on how the atmosphere grid was constructed. The easiest way to find out how to use this command is by entering the command **table star available** after the grids have been installed. The **table star** accepts the optional keyword **log** which indicates that the first number on the line is the logarithm of that parameter. Usually that is the effective temperature.

Figure 6.4 compares predictions for five of the 50 kK SEDs that are available. These include a blackbody and atmospheres computed by Mihalas (1972), Kurucz (1979), Kurucz (1991) and Rauch (2003). All were normalized to have the same total luminosity (10^{38} erg s^{-1}) observed from a distance of 10^{18} cm. Note the order of magnitude dispersion among the continua for energies around 4 Ryd.

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

Danger! Many forms of the **table stars** commands accept the **log** keyword to indicate that the log of the stellar temperature is entered. Many also accept the surface gravity, normally referred to as "log g", as a second parameter. The "log" in "log g" would cause the temperature to be interpreted as the log of the temperature. Either do not enter the string "log g", or write it as "logg" or "log(g)".

6.13.1 Overview of available stellar SEDs

In versions C06.02 and before the stellar continua were entered into the code on a case by case basis. New continua could only be introduced by writing new code. Beginning in C07.02 Peter van Hoof developed a unified treatment so that any continuum could be used by putting it into a standard format. This is far more flexible and allows more continua to be entered.

Usually grids have a range of surface temperature, and many also include a range of surface gravity or metallicity. But other types of parameters are also possible, e.g., age in the case of stellar population synthesis models. The **table star** command will interpolate on these tables to calculate an SED with the specified parameters. More information on the various grids that are supported can be found on our web site.

To list all the grids that have already been installed on your system, you can use the **table star available** command. This will list all the standard grids that were installed from the webpage above (including their parameters), but not the custom grids that you created yourself. The output will also list the **table HM05** and **table HM12** commands. They are not strictly part of the **table star** family of commands, but they internally use the stellar grid infrastructure to do their work.

To list all the SEDs that are contained in a single stellar atmosphere grid, you can use the **table star** <**grid**> **list** command. For more details see our webpage listed above.

6.13.2 High-energy component

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

$$\log(L_x) = 1.08\,(+0.06/-0.22)\,\log(L_{bol}) - 9.38\,(+2.32/-0.83)\,. \qquad (6.12)$$

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

6.13.3 Starburst99 files

It is possible to read in predictions from Starburst99 (Leitherer et al., 1999).[4] The Starburst99 SEDs are unified with the stellar atmosphere treatment. See our web site wiki.nublado.org/wiki/StellarAtmospheres and Appendix B.2 below for more details.

The procedure is to create your Starburst99 calculation on their web site (we tested the version at STScI). Store the Starburst99 output file "*.spectrum1" as a file with a "`.stb99`" or "`.stb`" extension. Compile it as in the following example

```
compile star "some_file.stb99"
```

You can optionally include the keyword **stellar** or **nebular** if you want to use only the stellar or the nebular component of the interstellar radiation field (these are the fourth and fifth column of the Starburst99 calculation, respectively). The default is to use the total spectrum, which is the sum of the stellar and nebular component. This would be the equivalent of including the emission from nearby star forming regions in your calculation. If you don't want that, you can use the stellar component only, as in the following example

```
compile star "some_file.stb99" stellar
```

The above commands will produce files `some_file.ascii` and `some_file.idx`, which can then be used in the **table stars** command:

```
table star "some_file.ascii" age=6e7 years
```

The number on the **table star** command is the age of the starburst. It has the same units as the ages given in the Starburst99 data file and is normally given in years. It is interpreted as a log if the keyword **log** appears, and as the linear age otherwise.

It is critical that the original Starburst99 file not be changed in any way at all. Doing so will trick the code into generating bogus results. In any case, it is a good idea to make a plot of the generated radiation field, using column 1 and 2 of the output from the **save continuum** command, to confirm that it is correct.

Starting with the C10 release it is possible to combine several Starburst99 runs with different metallicities into a single 2-dimensional grid. This allows interpolation in both age and $\log Z$. This has to be done by compiling each of the Starburst99 runs as described above and then manually merging the "`.ascii`" files into a single "`.ascii`" file. Appendix B.2 gives details how the merged file should be formatted. An example of a merged file is available on the StellarAtmospheres page of the Cloudy wiki linked at the start of this subsection. The resulting file can be compiled using

```
compile star "merged_file.ascii"
```

and used as follows

```
table star "merged_file.ascii" age=6.7 log(Z)=-1.2
```

[4] The **table starburst** command, which existed for this purpose in versions through C07, has been replaced with this **table stars** command.

6.13.4 PopStar and BPASS grids

You can convert PopStar and BPASS grids into "`.ascii`" files so that they can be used in CLOUDY. You can either use a single PopStar grid, allowing interpolation in age, or combine several PopStar grids for the same initial mass function but different metallicities allowing interpolation in age as well as $\log Z$. Detailed instructions can be found on wiki.nublado.org/wiki/StellarAtmospheres.

Chapter 7

CHEMICAL COMPOSITION

7.1 Overview

The default solar composition is summarized in Table 7.1. C and O abundances come from photospheric abundances of Allende Prieto et al. (2002, 2001), while N, Ne, Mg, Si, and Fe are from Holweger (2001). The helium abundance is a typical value for nebulae with near-solar compositions. The remainder of the first thirty elements comes from Grevesse and Sauval (1998). Meteoritic and photospheric abundances agree for most elements. They differ by significant amounts for P, S, Cl, and Mn. These are fairly volatile elements so may be deficient in meteorites. For these four the means of the meteoritic and photospheric abundances were used. The default solar abundances are stored in the file `data/abundances/default.abn` and can be changed by altering or overwriting that file.

These default abundances are maintained for backwards compatibility. The **abundances GASS** command described in section 7.2 will use values from Grevesse et al. (2010).

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

The following sections describe how to modify the chemical composition.

7.2 Absolute abundances and scale factors

Abundances can be specified as either *absolute abundances* or as *scale factors.* An absolute abundance gives the abundance of an element relative to hydrogen. An example might be $n(\mathrm{O})/n(\mathrm{H}) = 4 \times 10^{-4}$. A relative abundance is given relative to another value. An example might be the abundance relative to the solar value, $n(\mathrm{O})/n(\mathrm{H}) = 4 \times [n(\mathrm{O})/n(\mathrm{H})]_{\odot}$.

Table 7.1: Default Solar Composition

A			12+log	log	n/n(H)	ref
1	H	Hydrogen	12.00	0.00	1.00E+00	GS98
2	He	Helium	11.00	−1.00	1.00E-01	text
3	Li	Lithium	3.31	−8.69	2.04E-09	GS98
4	Be	Beryllium	1.42	−10.58	2.63E-11	GS98
5	B	Boron	2.79	−9.21	6.17E−10	GS98
6	C	Carbon	8.39	−3.61	2.45E-04	AP02
7	N	Nitrogen	7.93	−4.07	8.51E-05	H01
8	O	Oxygen	8.69	−3.31	4.90E-04	AP01
9	F	Fluorine	4.48	−7.52	3.02E-08	GS09
10	Ne	Neon	8.00	−4.00	1.00E-04	H01
11	Na	Sodium	6.33	−5.67	2.14E-06	GS98
12	Mg	Magnesium	7.54	−4.46	3.47E-05	H01
13	Al	Aluminium	6.47	−5.53	2.95E-06	GS98
14	Si	Silicon	7.54	−4.46	3.47E-05	H01
15	P	Phosphorus	5.51	−6.50	3.20E-07	GS98*
16	S	Sulphur	7.27	−4.74	1.84E-05	GS98*
17	Cl	Chlorine	5.28	−6.72	1.91E-07	GS98
18	Ar	Argon	6.40	−5.60	2.51E-06	GS98
19	K	Potassium	5.12	−6.88	1.32E-07	GS98
20	Ca	Calcicum	6.36	−5.64	2.29E-06	GS98
21	Sc	Scandium	3.17	−8.83	1.48E-09	GS98
22	Ti	Titanium	5.02	−6.98	1.05E-07	GS98
23	V	Vanadium	4.00	−8.00	1.00E-08	GS98
24	Cr	Chromium	5.67	−6.33	4.68E-07	GS98
25	Mn	Manganese	5.46	−6.54	2.88E-07	GS98*
26	Fe	Iron	7.45	−4.55	2.82E-05	H01
27	Co	Cobalt	4.92	−7.08	8.32E-08	GS98
28	Ni	Nickel	6.25	−5.75	1.78E-06	GS98
29	Cu	Copper	4.21	−7.79	1.62E-08	GS98
30	Zn	Zinc	4.60	−7.40	3.98E-08	GS98

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

7.3 Precedence

There are many ways to specify an abundance. It is even possible to specify different abundances for the same element in the same input script. If the absolute abundance is specified with more than one command then the last abundance is used. If the abundance is specified by both its absolute abundance and by a scale factor then both are used. Either of the following will multiply the default H II region nitrogen abundance by a factor of two:

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

or

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

The **element scale factor** option applies a scale factor while the **element abundance** option sets an absolute abundance. The **abundances** H II **region** command uses one of the absolute abundance mixtures that are stored by the code. This command enters the gas-phase abundances of all 30 elements and also includes interstellar grains. The **element** command changes some details about a particular element.

In the following example both commands set absolute abundances so the first nitrogen **abundance** will have no effect and the final nitrogen abundance will be the default H II region abundance

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

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

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

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

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

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

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

7.4 Abundances he c...

The abundances of all elements can be entered with the command **abundances** followed by: a) a complete set of abundances, b) the keyword **all** and a single number to set all of the abundances, or c) a second keyword to select one of the stored abundance sets.

7.4.1 Arbitrary abundances

The **abundances** command can be used to specify the abundances of a number of elements in one command. The composition can be specified on several lines with **continue** lines following the initial **abundances** line. Abundances of zero are not allowed; CLOUDY will stop if they are entered. Elements can be turned off with the **elements off** command.

The element symbol must be written before its abundance to identify what element the value belongs to. An optional equals sign is allowed in between the symbol and the value to improve readability. The abundances can be entered in any order.

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

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

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

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

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

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

7.4.2 Setting all at once

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

```
abundances all -10
abundances all 1e-10
```

The **metals** command will set abundances of all elements heavier than helium.

7.4.3 Abundance "filename.abn" – using tables of abundances

A set of abundances stored in an external file are used if there are no numbers on the **abundances** command but a file name occurs in quotation marks. The following gives some examples:

Table 7.2: Stored Abundance Sets

Keyword	Filename	Description	grains?
	`default.abn`	The default abundances	no
CAMEron	`Cameron.abn`	These are from Cameron (1982).	no
PRIMordial	`primordial.abn`	The primordial abundances.	no
CRAB	`crab.abn`	These are from model M1, Table 3, of Pequignot and Dennefeld (1983).	no
NOVA	`nova.abn`	These are roughly those derived by Ferland and Shields (1978).	no
HII, ORIOn	`hii.abn`	The HII region abundances are the mean of the Orion Nebula abundances given in three studies.	Orion
AGB, PLANetary	`pn.abn`	These abundances are from Aller and Czyzak (1983) and Khromov (1989), with high depletions assumed for elements they do not list.	ISM
ISM	`ism.abn`	The gas-phase abundances are an average for the warm and cold phases of the interstellar medium.	ISM
OLD SOLAR 84	`solar84.abn`	The composition will be the solar used in versions 84–94 of the code. These are defined in Table 7.3 and were taken from the meteoritic abundances of Grevesse and Anders (1989) with extensions by Grevesse and Noels (1993). All three of the keywords **old solar 84** must appear.	no
GASS	`solar_GASS10.abn`	This will use solar abundances from Grevesse et al. (2010). The abundances are listed in Table 7.4.	no
ALLEn	`allen73.abn`	These are the cosmic abundances of Allen (1973).	no

```
abundances "cameron.abn"
abundances "HII.abn" no grains
```

Table 7.2 lists some of the abundance sets that are included in the distribution. The keyword given in the first column of Table 7.2 can be used rather than the filename. This maintains compatibility with versions C13 and before. This abundance sets in this table have been part of the code's distribution for a number of years and more recent sets are not listed here. Do a listing of the `*.abn` files in `data/abundances` to get a full list of the available abundances.

Some of the stored abundance sets include grains, as indicated in Table 7.2. The **no grains** and **no QHeat** options are recognized, as described on the **grains** command (see section 7.9). For instance, ISM abundances but Orion grains would be specified with the combination

```
grains Orion
abundances "ISM.abn" no grains
```

Because of the ability to use keywords, this is equivalent to

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

The abundances files contain lines giving the name of an element and its abundance relative to hydrogen. Hydrogen may be in specified, or may not be. If the abundance of hydrogen is given and is not equal to unity the code will rescale the abundances by the entered value for hydrogen. If hydrogen is not specified it is assumed to have an abundance of 1.

All elements heavier than hydrogen are turned off before the abundance files are read. An element is turned on if it appears in the file. If an element is not included in the file it will not be included in the calculation.

The **abundances** command has a **print** option to report grain types and abundances entered for each active element.

7.4.4 Older solar abundances

The default solar abundances are stored in the file `data/abundances/solar.abn` and can be changed by altering or overwriting that file.

The keyword **old solar 84** uses the solar composition used in versions 84 - 94 of the code. These are defined in Table 7.3 and were taken from the meteoritic abundances of Grevesse and Anders (1989) with extensions by Grevesse and Noels (1993). The keyword **Cameron** uses the Cameron (1982) abundances. The keyword **Allen** uses the Allen (1973) abundances. These are maintained for historical reasons.

7.4.5 Grains, gas-phase depletions, and quantum heating

Certain elements, especially Si, Ca, Al, Mg, and Fe, are heavily depleted onto grains in the ISM. Kingdon et al. (1995) discuss the observational effects of grains upon an H II region. The abundance sets specified by the **H II region**, **ISM**, or **planetary nebula** keywords will include grains and the gas-phase mixtures given in Table 7.2. The bottom line of the table lists the type of grains that are included by default. These types are defined in the section on grains starting on page 80.

Grains can also be specified separately with the **grains** command (see section 7.9). If you use the **grains** command to specify another type of grains then you should include the keyword **no grains** on the abundances line to prevent the default grains from also being used.

If you set the keyword **no grains** to not include the default grains and do not specify another set of grains with the **grains** command then grains will not be included in the calculation but the depleted gas-phase abundances will still be used.[1] This is, of course, not self-consistent.

Quantum or single-photon heating will be considered for all grain species where it will be important. Quantum heating only affects the Wien tail of the grain thermal-emission spectrum but is computationally fairly expensive. The **no qheat** option on the **abundances** command will disable quantum heating. Figure 7.1 compares the continuum predicted in the test case `grains_qheat`, a part of the code's test suite. Quantum heating is normally included and the **no qheat** option was used to disable it to make this comparison.

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

Table 7.3: Old Solar Composition, default in versions C84 - C94

				Solar	
1	H	Hydrogen	1	0.00	12.00
2	He	Helium	0.1	−1.00	11.00
3	Li	Lithium	2.04E-09	−8.69	3.31
4	Be	Beryllium	2.63E-11	−10.58	1.42
5	B	Boron	7.59E-10	−9.12	2.88
6	C	Carbon	3.55E-04	−3.45	8.55
7	N	Nitrogen	9.33E-05	−4.03	797
8	O	Oxygen	7.41E-04	−3.13	8.87
9	F	Fluorine	3.02E-08	−7.52	4.48
10	Ne	Neon	1.17E-04	−3.93	8.07
11	Na	Sodium	2.06E-06	−5.69	6.31
12	Mg	Magnesium	3.80E-05	−4.42	7.58
13	Al	Aluminium	2.95E-06	−5.53	6.47
14	Si	Silicon	3.55E-05	−4.44	7.55
15	P	Phosphorus	3.73E-07	−6.43	5.57
16	S	Sulphur	1.62E-05	−4.79	7.21
17	Cl	Chlorine	1.88E-07	−6.73	5.27
18	Ar	Argon	3.98E-06	−5.40	6.60
19	K	Potassium	1.35E-07	−6.87	5.13
20	Ca	Calcicum	2.29E-06	−5.64	6.36
21	Sc	Scandium	1.58E-09	−8.80	3.20
22	Ti	Titanium	1.10E-07	−6.96	5.04
23	V	Vanadium	1.05E-08	−7.98	4.02
24	Cr	Chromium	4.84E-07	−6.32	5.68
25	Mn	Manganese	3.42E-07	−6.47	5.53
26	Fe	Iron	3.24E-05	−4.49	7.51
27	Co	Cobalt	8.32E-08	−7.08	4.92
28	Ni	Nickel	1.76E-06	−5.75	6.25
29	Cu	Copper	1.87E-08	−7.73	4.27
30	Zn	Zinc	4.52E-08	−7.34	4.66

Table 7.4: 2010 Solar Composition, keyword **GASS**

			Solar	
1	H	1	0	12
2	He	8.51E-02	-1.07	10.93
3	Li	1.12E-11	-10.95	1.05
4	Be	2.40E-11	-10.62	1.38
5	B	5.01E-10	-9.3	2.7
6	C	2.69E-04	-3.57	8.43
7	N	6.76E-05	-4.17	7.83
8	O	4.90E-04	-3.31	8.69
9	F	3.63E-08	-7.44	4.56
10	Ne	8.51E-05	-4.07	7.93
11	Na	1.74E-06	-5.76	6.24
12	Mg	3.98E-05	-4.4	7.6
13	Al	2.82E-06	-5.55	6.45
14	Si	3.24E-05	-4.49	7.51
15	P	2.57E-07	-6.59	5.41
16	S	1.32E-05	-4.88	7.12
17	Cl	3.16E-07	-6.5	5.5
18	Ar	2.51E-06	-5.6	6.4
19	K	1.07E-07	-6.97	5.03
20	Ca	2.19E-06	-5.66	6.34
21	Sc	1.41E-09	-8.85	3.15
22	Ti	8.91E-08	-7.05	4.95
23	V	8.51E-09	-8.07	3.93
24	Cr	4.37E-07	-6.36	5.64
25	Mn	2.69E-07	-6.57	5.43
26	Fe	3.16E-05	-4.5	7.5
27	Co	9.77E-08	-7.01	4.99
28	Ni	1.66E-06	-5.78	6.22
29	Cu	1.55E-08	-7.81	4.19
30	Zn	3.63E-08	-7.44	4.56

To access the full range of grain options it is best to combine the **no grains** option on the **abundances** command and follow that with the **grains** command and its options, described in section 7.9), as in

```
abundances ISM no grains
grains [options]
```

7.4.6 Interactions between the abundances and grains commands

It is possible to include grain species with both the **abundances** command, described here, and with the **grains** command. For simplicity you should only set the grain abundances only once. If you specify the same grains more than one time you are asking for trouble. The rules for how these commands interact are described elsewhere.

You should check that you have the expected grain abundances by looking at the list of grain abundances given at the start of the calculation and at the value of $A_V/N_{\rm H}$ given at the end.

7.5 Abundances starburst, Z=10

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

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

This command works by generating scale factors that multiply the current standard abundance. This is solar by default. The command does not derive absolute abundances. If you change the abundances with a second **abundance** command then the scale factors will multiply those new abundances.

The keyword **trace** will print the abundances of all elements as a function of metallicity between the full $10^{-3}Z_{sun}$ and $36Z_{sun}$ limits. The code will then stop.

7.6 Abundances isotopes

Isotope fractions may be specified with this command by providing one of the keywords entered in Table 7.5, or a filename enclosed in double-quotations. Note that all tables rely on the Rosman and Taylor (1998) terrestrial abundances for many isotopes. For instance, Asplund et al. (2009) make use of the "representative isotopic composition", while Lodders (2003) opts for the "best

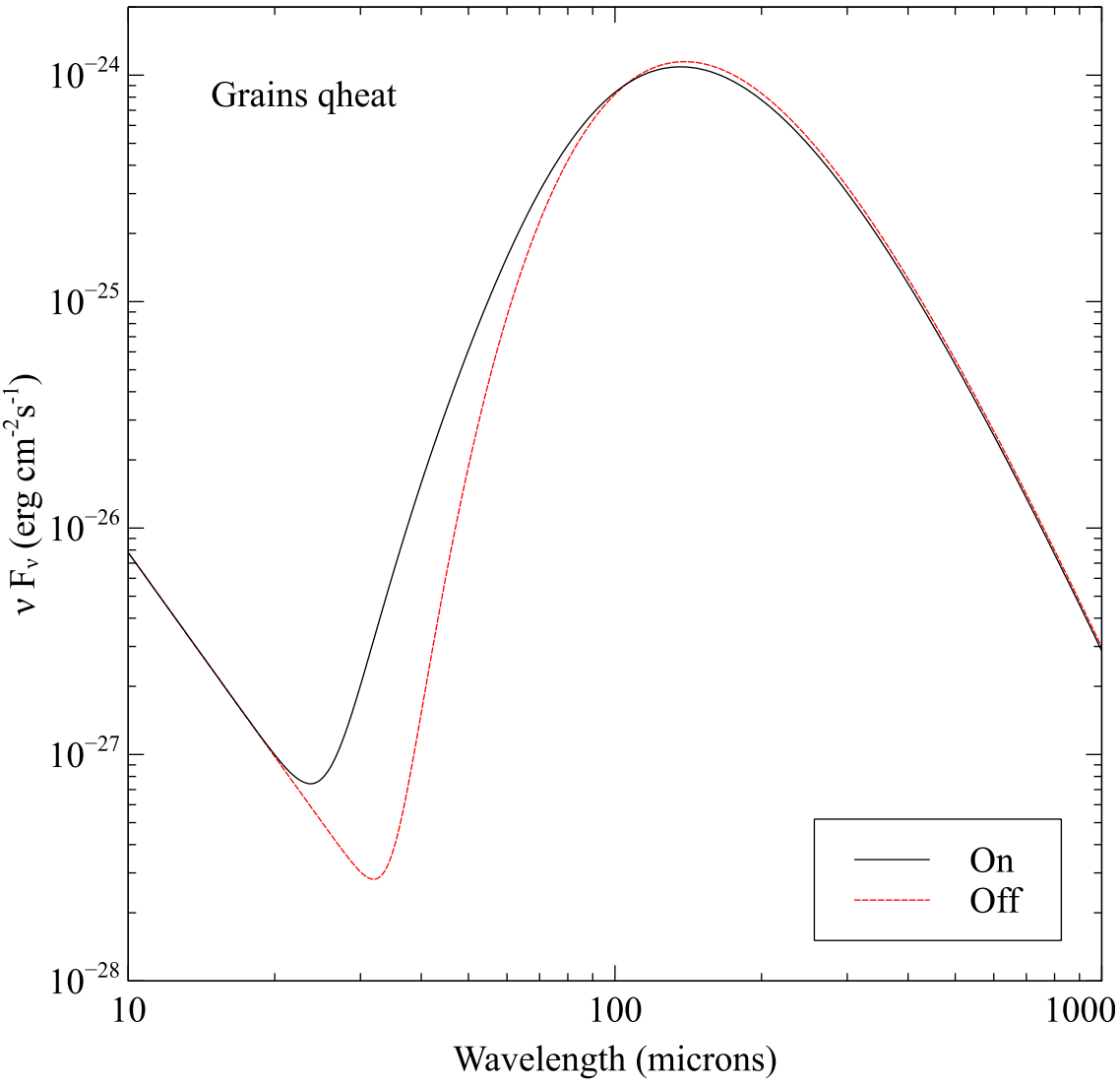

Figure 7.1: This figure shows the net emission computed by the test case `grains_qheat`. The solid line includes quantum heating while the dashed line has disabled it. The emission lines have been suppressed in this plot for clarity.

Table 7.5: Isotope Fractions Data

Keyword	Filename	Abundance Type	Reference
aspl	"Asplund09-iso.abn"	Protosolar	Asplund et al. (2009)
lodders09	"Lodders09-iso.abn"	Protosolar	Lodders et al. (2009)
lodders03	"Lodders03-iso.abn"	Solar System	Lodders (2003)
rosm	"Rosman98-iso.abn"	Terrestrial	Rosman and Taylor (1998)
	"default-iso.abn"	aspl + D/H + f(13C)	

measurement from a single terrestrial source". By default, the abundances of Asplund et al. (2009) are used, modified to match the primordial D/H ratio of 1.65×10^{-5} (Pettini and Bowen, 2001), and to reproduce Cloudy's historic 13C isotope fraction of 1/30.

Additional isotopic data used internally include masses derived from Audi and Wapstra (1995), and spins and magnetic moments obtained from Stone (2005).

Note that the command automatically resets the Deuterium and 13C abundances. The command **element name isotopes**, described below, should be issued to maintain a desired isotope ratio, following all instances of **abundances isotopes**.

A user-defined file must include fractions for all the isotopes present in the default files.

The default isotope abundances were used in our predictions for hyperfine structure line emission from hot and rarefied media, e.g., the intracluster medium in galaxy clusters, and the Orion million K gas; these are discussed in detail in Chatzikos et al. (2014).

7.7 Element name [scale, abundance, isotopes, off, log, table]

This sets the abundance of a particular element. The command has several keywords that change the behavior of the command.

7.7.1 Element name scale

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

7.7.2 Element name abundance

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

7.7.3 Element name isotopes

If the keyword **isotopes** appears, then the code expects a pair of numbers, the atomic mass number and the isotope fraction, for each of the elemental isotopes. Linear numbers are expected.

Setting the D/H ratio

This command should be used to set the D/H abundance ratio. For example, the following sets the ratio to 2×10^{-5}, close to the current solar-system value:

```
element hydrogen isotopes (1, 1) (2, 2e-5)
```

This mainly affects HD cooling. The default is the primordial abundance of 1.65×10^{-5} as measured by Pettini and Bowen (2001).

The command **set D/H** has been deprecated.

Setting the 12C/13C ratio

This command should be used to set the 12C/13C ratio. For instance, the following sets the ratio to 5:

```
element carbon isotopes (12, 5) (13, 1)
```

The code currently predicts the ^{13}CO rotation spectrum and the intensity of ^{13}C III λ1910Å (Clegg et al., 1997) using this ratio. The code does not now solve for ^{12}C and ^{13}CO abundances and ionic fractions but rather assumes that the ratio ^{12}CO/^{13}CO and the $^{12}C^{+2}/^{13}C^{+2}$ ionization fractions are equal to ^{12}C/^{13}C. This ignores chemical fractionation. A full treatment of this physics is high on the list of priorities.

The ^{12}C/^{13}C ratio depends on both location in the galaxy and age. It is ≈ 90 in the solar system (Asplund et al., 2009) and reflects the galactic value when the sun formed. The value toward the Galactic center is about 20 increasing to ≈ 77 at the solar radius (Wilson and Rood, 1994). The default value of 30 is a compromise.

The command **set 12C13C** has been deprecated.

7.7.4 Element isotopes all

This command enables the available isotopologues of all molecules. Currently, only singly-substituted isotopologues are available in Cloudy. Densities are simply scaling by fractionations of constituent elements.

Deuterium is not presently enabled by this command. Isotopes of all other elements are enabled.

The command **set isotopes all** has been deprecated.

7.7.5 Element name ionization

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

The total density of all ions must add up to the total density of the element. The code imposes several particle conservation rules to insure that mass is conserved. The density of the ions will be

renormalized so that the total abundance of the element is equal to the value specified by other commands.[2]

This command does not specify the chemical state. The H_2 fraction can be set with the **set H2** command. Chemistry is disabled if you set the ionization distribution of an element that is part of the chemical network. It will not be possible to obtain a chemical solution if the abundances of the atom and ions cannot respond to changes in the chemistry.

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

7.7.6 Element name off

If the keyword **off** appears (note the leading space) then the element is not included in the calculation. The ionization equilibrium, opacity, and cooling due to the element will not be computed.

The keyword **on** will include an element that was previously turned off in the same input stream.

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

Memory for the species that are present is dynamically allocated when the code starts. When the code is used to compute a large grid of models it is not possible to turn on an element that was turned off for the first model since the needed memory does not exist. You can turn off an element in later models since the element is simply not computed. In later calculations in a grid the code will ignore any attempt to turn on an element that was initially turned off.

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

Turning off an element with this command has other side effects. For instance, the following input stream would produce unexpected results:

```
element zinc ionization -3 0 -4
init "ism.ini"
element zinc on
```

because the `ism.ini` file turns off zinc. This destroys the memory of the **element zinc ionization** command, which sets the ionization of the element. So, when the element is turned back on with the **element zinc on** command the ionization will be determined self consistently rather than set to the constant values.

7.7.7 Element limit off -12

Elements with abundances that are smaller than the number on the line will be ignored. The number is the log of the limiting abundance by number relative to hydrogen. This is an easy way

[2] In versions C08 and before this command did not confirm that the sum of the ionization fractions was unity. The abundance of each stage of ionization was set to the product of the ionization fraction times the abundance set by other commands. If the fractions did not add up to unity the effect was to change the total abundance of the element.

to turn off elements that have trivial abundances. Table 7.6 lists the elements in terms of decreasing abundance. The command

```
elements limit off -7
```

would cause all elements at Cobalt and above in Table 7.6 to be turned off.

7.7.8 Element name table

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

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

The first number on each line is the log of the radius (the default) or depth (if the keyword **depth** also appears on the **element** line). The second number is the log of the abundance of the element at that point, by number relative to hydrogen. The table ends with a line starting with the keyword **end**. At least two pairs of data need to be entered to enable interpolation by the code. There is no upper limit to the number of pairs that may be entered. This command always specifies the absolute abundance and not the scale factor.

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

7.8 Fluctuations abundances, period, max, min, phase

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

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

The **fluctuations** command is more fully described in Section 8.6 on the density version.

7.9 Grains

This command controls how grains are included in the calculation. They are not included by default. Grains are included in the compositions set by some **abundances** commands (see Table

Table 7.6: Solar composition sorted by abundance

A	Symbol	Name	$\log n(x)/n(\mathrm{H})$
4	Be	Beryllium	-10.58
5	B	Boron	-9.21
21	Sc	Scandium	-8.83
3	Li	Lithium	-8.69
23	V	Vanadium	-8.00
29	Cu	Copper	-7.79
9	F	Fluorine	-7.52
30	Zn	Zinc	-7.40
27	Co	Cobalt	-7.08
22	Ti	Titanium	-6.98
19	K	Potassium	-6.88
17	Cl	Chlorine	-6.72
25	Mn	Manganese	-6.54
15	P	Phosphorus	-6.50
24	Cr	Chromium	-6.33
28	Ni	Nickel	-5.75
11	Na	Sodium	-5.67
20	Ca	Calcium	-5.64
18	Ar	Argon	-5.60
13	Al	Aluminium	-5.53
16	S	Sulphur	-4.74
26	Fe	Iron	-4.55
12	Mg	Magnesium	-4.46
14	Si	Silicon	-4.46
7	N	Nitrogen	-4.07
10	Ne	Neon	-4.00
6	C	Carbon	-3.61
8	O	Oxygen	-3.31
2	He	Helium	-1.00
1	H	Hydrogen	0.00

Table 7.7: Standard Grain Types

Keyword	Filename	Type	Size distribution
	graphite_0m010.opc	Graphite	Single 0.01 μm
	graphite_0m100.opc	Graphite	Single 0.1 μm
	graphite_1m000.opc	Graphite	Single 1 μm
ISM single graphite	graphite_ism_01.opc	Graphite	Unresolved ISM
ISM graphite	graphite_ism_10.opc	Graphite	ISM, 10 bins
Orion single graphite	graphite_orion_01.opc	Graphite	Unresolved Orion
Orion graphite	graphite_orion_10.opc	Graphite	Orion, 10 bins
grey single	grey_ism_01.opc	Grey	Unresolved ISM
grey	grey_ism_10.opc	Grey	ISM, 10 bins
PAH single	pah1_ab08_01.opc	PAH	Unresolved AB08
PAH	pah1_ab08_10.opc	PAH	AB08, 10 bins
	pah1_c120.opc	PAH	120 C atoms
	pah1_c15.opc	PAH	15 C atoms
	silicate_0m010.opc	Silicate	Single 0.01 μm
	silicate_0m100.opc	Silicate	Single 0.1 μm
	silicate_1m000.opc	Silicate	Single 1 μm
ISM single silicate	silicate_ism_01.opc	Silicate	Unresolved ISM
ISM silicate	silicate_ism_10.opc	Silicate	ISM, 10 bins
Orion single silicate	silicate_orion_01.opc	Silicate	Unresolved Orion
Orion silicate	silicate_orion_10.opc	Silicate	Orion, 10 bins

7.2 above). Grain physics was originally developed in collaboration with Peter van Hoof, Peter G. Martin, and Joe Weingartner. Peter van Hoof did the majority of the coding, and he is the maintainer of the code. Reviews are given by Spitzer (1948, 1978), and Martin and Zalubas (1979). Baldwin et al. (1991), Weingartner and Draine (2001b), van Hoof et al. (2004) and Weingartner et al. (2006) describe the theory incorporated in the code.

The **save grain** commands produce output giving details about the grains. Some details of the grain physics can be adjusted with the **no grain ...** family of commands. The **compile grains** command (page 268 below) is used to create new grain opacity files, or to recompile them if the code's energy mesh is changed.

7.9.1 Using the built-in grain types

The grain types summarized in Table 7.7 are included in the code's data files. The table gives the grain material, its size distribution, and the name of the grain opacity file that it uses. In most cases these will be sufficient - nothing more needs to be done to set up the grains.

Grains are composed of elements that are condensed from the gas phase. It would be inconsistent to assume solar abundances for all elements and also include grains. In reality certain elements, especially Ca, Al, Ti, and Fe, are strongly depleted from the gas phase in the ISM and are believed to be present as grains. The **metals deplete** command can be used to deplete elements that are included in grains.

Several of the standard abundance sets that are specified with the **abundances** command

include grains. The elements which make up the grains are depleted from the gas phase in these abundance sets. Elements are not depleted in sets that do not include grains by default.

- **grains ISM** specifies grains with a size distribution and abundance appropriate for the ISM of our galaxy. This includes both a graphitic and silicate component and generally reproduces the observed overall extinction properties for a ratio of extinction per reddening of $R_V \equiv A_v/E(B-V) = 3.1$. This is the default and will be used if no keywords occur on the **grains** command. If either keyword **graphite** or **silicate** also appears then only that grain type is included. Both species are included if neither keyword appears.

- **grains Orion** specifies graphitic and silicate grains with a size distribution and abundance appropriate for those along the line of sight to the Trapezium stars in Orion. The Orion size distribution is deficient in small particles and so produces the relatively grey extinction observed in Orion (Baldwin et al., 1991). If either keyword **graphite** or **silicate** appears then only that grain type is included. Both species are included if neither keyword appears.

- **grains PAH** turns on PAHs. See section 7.9.4 below for more details.

Grain abundances An optional scale factor on the command line changes the grain abundance relative to its default value. If the keyword **log** appears or if the number is less than or equal to zero then the number is the log of the scale factor. It is the linear factor if the keyword linear appears or if no keyword appears and the number is positive.

It is also possible to change the abundances of the grains with the **metals grains** command. The **metals** command scales the abundances of all elements more massive than helium by a scale factor. If the keyword **grains** occurs then the grain abundance will also be scaled up or down. This would keep the grains to metals abundance ratio constant.

The following example uses ISM gas-phase abundances, Orion silicates with dust to gas ratio twice the default value, and ISM graphite with default ISM abundances.

```
# use ISM abundances but DO NOT include ISM default grains
abundances ISM no grains
# Orion silicate with twice the Orion abundance
grains Orion silicate 2
# ism graphite with ISM dust to gas ratio
grains ISM graphite
```

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

Size resolved or averaged grains Grains in the ISM have a broad range of sizes, often described by a power-law distribution (Mathis et al., 1977). By default the grains will be resolved into ten size bins. If the keyword **single** appears then the grains will have properties determined by averaging over the entire size range. If the keyword **distribution** appears then the code will resolve the size distribution. This is the default and will be used if no keyword appears. The **single**

option may save some machine time but will give a less realistic representation of the grain physics. This is because many grain properties such as temperature and charge depend strongly on the size. In particular, the photoelectric heating of the gas can be underestimated if the single mean grain is used.

7.9.2 Interactions between abundances and grains commands

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

- An **abundance** command will override all grains set with a previous **abundances** command. So with the following pair

  ```
  abundances ism
  abundances old solar 84
  ```

  ISM grains stay in place since grains are not included in the old solar mixture.

  ```
  abundances ism
  abundances orion
  ```

  only Orion grains will be included

- An **abundances** command will NOT override any grains set with a previous **grains** command. It will not add any grains either. Instead it will implicitly behave as if the option **no grains** was given on the **abundances** command.

  ```
  grains orion
  abundances ism
  ```

  only Orion grains are included

- A **grains** command given after an **abundances** command or another **grains** command will add to those already included, even if it would cause the same set of grains to be included twice. A warning will be printed if the latter occurs. For instance, the following would add the ISM grains twice and generate a warning;

  ```
  abundances ism
  grains ism
  ```

- An **abundance** command that does not set any grains will not override any grains set with a previous **abundances** command.

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

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

The following will use ISM grains and Orion abundances, but will not include Orion grains since the **abundances** command comes after the **grains** command;

```
grains ism
abundances Orion
```

The order is swapped in the following. This results in Orion gas-phase abundances and both ISM and Orion grains

```
abundances Orion
grains ism
```

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

7.9.3 Creating your own grain types

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

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

7.9.4 PAHs

PAH's are not included by default but are added with the **PAH** option on the **grains** command. A detailed treatment of the physics of PAH's is implemented, including photoelectric heating and collisional processes as discussed by Weingartner and Draine (2001a), and stochastic heating effects following Guhathakurta and Draine (1989). A power-law distribution of PAH sizes with 10 size bins (Abel et al., 2008, hereafter AB08), and two single-sized PAH's, are included in the downloaded data files. AB08 is the default and will be used if no further options occur on the command line. If the keyword **single** also appears then an unresolved size distribution with their mean properties is used.

The PAH opacity functions were derived by Kevin Volk from a variety of sources. The original opacity function is from Desert et al. (1990) and Schutte et al. (1993). Kevin adapted the opacities from these papers to agree with the infrared plateaus seen in the Orion Bar (Bregman et al., 1989). The optical/far-UV opacity values have a gradual change to atomic cross sections so that you get the correct X-ray cross sections.

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

corresponding to $n(\mathrm{C})/n(\mathrm{H}_{tot}) = 2.979 \times 10^{-6}$. The command **grains PAH C120** will include a large PAH with 120 carbon atoms per molecule and an abundance relative to hydrogen of $n(\mathrm{PAH})/n(\mathrm{H}_{tot}) = 2.483 \times 10^{-8}$, also corresponding to $n(\mathrm{C})/n(\mathrm{H}_{tot}) = 2.979 \times 10^{-6}$.

PAHs appear to exist mainly at the interface between the H^+ region and the molecular clouds. Apparently PAHs are destroyed in ionized gas (Sellgren et al., 1990, AGN3 section 8.5) by ionizing photons and by collisions with ions (mainly H^+) and may be depleted into larger grains in molecular regions. By default the code assumes that the PAH abundance scales with the ratio $n(H^0)/n(H)$ although this can be changed. This produces very few grains in ionized and fully molecular gas, but the PAHs will have their default abundance when the gas is atomic. This is consistent with Sellgren's observations of the Orion Bar.

The **set PAH** command described on page 303 makes it possible to specify several other laws describing how PAH abundances depend on physical conditions. Note that if PAHs are present in predominantly molecular gas then they will soak up nearly all of the free electrons. This has major effects on the predictions of the chemistry network.

Tielens (2008) has determined the abundance of carbon locked up in PAHs containing 20 – 100 atoms as 14 parts per million hydrogen atoms. The average Galactic PAHs per hydrogen, according to Tielens (2008), is $10^{-6.52}$. This abundance can be generated with the commands

```
grains PAH 10 linear
set PAH "H,H2"
```

7.9.5 Variable grain abundances

The **function** option on the **grains** command makes it possible to vary the abundance of any grain species across a cloud. (The **abundances** command does not have the **function** option.) This option works by setting the local abundance of a grain species to the product of the intrinsic abundance and the value returned by the function *GrnVryDpth*. That routine can be modified by the user to specify a scale factor that might depend on other physical conditions or location. Alternatively, you can supply the keywords **function sublimation** which will turn on a predefined grain abundance function that mimics the effects of grain sublimation. It will steeply lower the grain abundance when the grain temperature is above the sublimation temperature for a given grain bin. The function that is used is as follows:

$$A_{\mathrm{g}} = \exp\left[-\left(\frac{T_{\mathrm{g}}}{T_{\mathrm{subl}}}\right)^3\right].$$

All the other multipliers for the grain abundance are still in effect when the keyword **function** is used. This means that the abundance multiplier supplied with the **grains** command is also included, as well as the factor supplied in the **metals grains** command. So the abundance of the grain is given by the default abundance multiplied with the two numbers mentioned here times whatever the grain abundance function returns.

If the **grains** command sets the abundance of a single grain species then the **function** option will only apply to that particular species. If the option occurs on a **grains** command that specifies more than one species of grains (as in the **ISM** keyword) then all species enabled by that command are affected. This is probably unphysical since small grains are easier to destroy than

large grains. However, it is possible to define a different behavior of the grain abundance in routine *GrnVryDpth* for each individual grain size bin.

The code does not attempt to conserve the mass of the grain constituents. The gas-phase abundances are not automatically changed where the grain abundances are changed. This can be done by entering abundances with a depth-dependent table of abundances using the **element ¡name¿ table** command.

7.9.6 Line intensities with grains

Two sets of line intensities, the intrinsic and emergent spectra, are included in the main output. These differ in how the effects of extinction by continuous opacity sources, especially dust, are treated. In general, the observed spectrum of an emission-line region containing dust depends on the geometry and the viewing angle, so some thought should go into the geometry when extinction is important.

It can be shown that the reddening or extinction across a photoionized H^+ layer will nearly always be small in the optical or IR (AGN3). If there is a noticeable amount of reddening, this must be happening in neutral material associated with the source, or (more likely) by dust inbetween the source and the observer. The effects of grains external to the emission-line region are very geometry dependent. One approach is to correct the observed spectrum for reddening to obtain an intrinsic spectrum, and to then compare this intrinsic spectrum with that computed by the code.

The intrinsic spectrum: This is the spectrum produced within the cloud and does not include effects of dust that lies outside the line-forming region. Photon destruction by all background opacity sources (including grains) are always fully treated (i.e., Hummer, 1968, Kalkofen, 1987), and the predicted intrinsic intensities always includes this destruction. These intensities *do not* include the reddening effects of any grains or other opacity sources that lie outside the line-forming region. For instance, this would not include the effects of grains in the PDR that lies outside the H^+ region.

The emergent spectrum: The emergent line intensities include the physics of the intrinsic spectrum but also account for opacity sources outside the region where the line forms. For instance, in a simulation of an open geometry that includes both the H^+ layer and an adjoining PDR the outward intensities of the Balmer lines, which form in the H^+ layer, will include absorption by grains and reflection from the PDR. In a closed geometry the spectrum would be that observed from outside the emission-line regions, looking through the cloud from its outermost layers.

In versions C13 and before we assumed that the ionized gas had a large molecular cloud behind the shielded face of the ionized gas and so the effects of absorption and reflection from this unmodelled molecular cloud were included in the emergent spectrum. This is appropriate for a star-forming H II region. Cloudy can now compute the structure and effects of the PDR and molecular cloud explicitly, so beginning in C17 the emergent intensity is computed for the computed geometry. The calculation can be extended into atomic and molecular regions by removing the kinetic temperature stopping criterion with the **stop temperature off** command (Section 15.17). The calculation needs to go deep enough into the molecular cloud for optical reflection to be included. This could be done with the command **stop Av 3** described in Section 15.4.

These issues are discussed further in Section Section 2.10, *Line intensities in a dusty cloud* in Part 2 of HAZY.

7.9.7 Extinction for point and extended sources

Grain extinction is given by the cross section [cm^2] per H nucleon: $\sigma = \kappa / n(\mathrm{H})$, where κ [cm^{-1}] is the opacity due to grains and $n(\mathrm{H})$[cm^{-3}] is the local density of H in all forms. The opacity includes both absorption and scattering. The scattering opacity depends on the geometry of the absorbing cloud, as described next and in AGN3 section 7.3.

Light scattering off grains is not isotropic. The angular dependence is often approximated by the Henyey-Greenstein function. The scattering theory predicts the fraction of scatterings, given by the grain asymmetry factor g, that are "forward scattering", that is, change the direction of the scattered photon by less than 90 degrees (i.e. into the forward 2π sr). The asymmetry parameter g is defined as the average value of $\cos(\theta)$, where θ is the angle between incident and scattered photon. So for isotropic scattering $g = 0$, while for strongly forward scattering, g will be close to 1.

Rather than the total scattering cross section σ_s, an effective scattering cross section $\sigma_{scat} = (\sigma_s(1-g)$ contributes to extinction for extended sources. This discounts light scattered into the forward 2π sr. The asymmetry parameter g approaches unity at high energies, so that σ_{scat} becomes much less than σ_{abs}. For an extended source such as a diffuse cloud the loss of photons by small-angle scattering will be compensated by a similar gain of photons from rays that are nearly parallel, so that total opacity is $\sigma_{abs} + \sigma_s(1-g)$. This is referred to as ***extended source extinction*** and would apply to a resolved HII region like the Orion Nebula.

For a point source such as a star even a small deflection of starlight by forward scattering removes light from the ray and so counts as extinction. In this case the total grain opacity is simply $\sigma_{abs} + \sigma_s$. This is referred to as the ***point source extinction*** and is the quantity measured in extinction studies of stars.

The code keeps track of both measures of extinction since a beam of light passing through an HII region is attenuated by the extended extinction while the observational stellar literature will quote a point source extinction.

7.9.8 Grains-related commands

A group of **no grain** commands turn off various physical processes.

There are several **save** commands that will output predicted properties of the grains.

save grains This command will give some additional information about the grain properties.

save continuum This produces files that include the infrared continuum emitted by the gas and grains.

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

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

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

Forcing quantum heating on or off By default, quantum (also called "stochastic" or "single-photon") heating is included for size-resolved species when it is significant. Guhathakurta and Draine (1989) describe the formalism used here. The method was originally implemented by Kevin Volk and subsequently revised and generalized by Peter van Hoof. This is considered for all species except for the unresolved size distributions. To save time quantum heating is only treated when the grain cooling time is sufficiently short compared with the time between heating events. Quantum heating is considered for a given size bin if the ratio of the volume of the largest and the smallest grain in that particular bin is less than 100. This means that quantum heating is considered for all of our resolved distributions, as well as all single-sized grains, but not for unresolved distributions (except the unresolved AB08 distribution, where it *is* used by default).

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

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

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

7.9.9 Effects of grains

Grains have several effects on interstellar gas (see AGN3, Chapter 7). Refractory elements are depleted from the gas phase, thus removing coolants from the gas (Kingdon et al., 1995). Grains absorb the incident continuum and reradiate an infrared continuum (Bottorff et al., 1998). Higher-energy photons ionize the grains, establishing a net charge, and so affect the charge balance of the gas (Baldwin et al., 1991). Photoejection of electrons following absorption of a UV photon heats the gas. Grains surfaces are sites of important chemical reactions, atomic/ion charge transfer, and molecular freeze-out when temperatures become low. Grains are a net sink of free electrons in fully molecular regions—this greatly influences PDR models. Several processes partially couple the temperatures of the gas and dust (van Hoof et al., 2004).

The heating and cooling of the grains and gas are done self-consistently. Grains are heated by direct absorption of the incident continuum, by the line and continuum emission within the cloud,

and by gas collisions. Grains cool by collisions with the gas, the thermionic effect, and by radiation. The balance between heating and cooling establishes the temperature for each grain size and type. Grains affect the gas temperature by heating, mainly by the grain photoelectric effect and the thermionic effect, and by cooling, mainly by free-particle capture onto the grain surface.

7.9.10 Grain-related quantities that are printed

Several quantities related to the grain properties are reported.

The chemical composition is reported as the calculation starts. This includes a block giving the abundances of the elements that are incorporated in grains. If the grain abundance varies with depth then this is for the conditions at the illuminated face. A typical printout appears as follows:

```
                    Grain Chemical Composition
C : -4.1243  O : -4.2374  Mg: -4.8394  Si: -4.8394  Fe: -4.8394

                  Number of grains per hydrogen
Carbonaceous: -7.144  Silicate: -7.418
```

After the calculation is complete there is a summary of some properties of the grains:

```
Average Grain Properties (over radius):
       sil-a0slm01* sil-a0slm02* sil-a0slm03* sil-a0slm04* sil-a0slm05* sil-a0slm06* sil-a0slm07* sil-a0slm08* sil-a0slm09* sil-a0slm10
   nd:      0            1            2            3            4            5            6            7            8            9
<Tgr>: 1.031e+02    9.156e+01    8.352e+01    7.854e+01    7.647e+01    7.556e+01    7.017e+01    6.218e+01    5.472e+01    4.989e+01
<Vel>: 1.620e+01    2.530e+02    5.102e+02    1.008e+03    1.908e+03    3.188e+03    4.218e+03    4.802e+03    5.183e+03    5.381e+03
<Pot>: 9.085e-01   -3.674e-01   -9.865e-01   -1.049e+00   -8.803e-01   -6.611e-01   -7.209e-01   -9.144e-01   -1.055e+00   -1.116e+00
<D/G>: 2.014e-05    2.969e-05    4.379e-05    6.464e-05    9.554e-05    1.416e-04    2.112e-04    3.186e-04    4.913e-04    3.264e-04

       gra-a0glm01* gra-a0glm02* gra-a0glm03* gra-a0glm04* gra-a0glm05* gra-a0glm06* gra-a0glm07* gra-a0glm08* gra-a0glm09  gra-a0glm10
   nd:     10           11           12           13           14           15           16           17           18           19
<Tgr>: 1.207e+02    9.886e+01    8.591e+01    8.088e+01    8.047e+01    7.164e+01    5.921e+01    4.733e+01    3.644e+01    3.030e+01
<Vel>: 2.500e+01    3.807e+02    9.346e+02    2.205e+03    4.155e+03    5.682e+03    6.422e+03    5.650e+03    5.099e+03    4.990e+03
<Pot>: 8.393e-01   -7.959e-01   -1.138e+00   -1.019e+00   -7.800e-01   -9.040e-01   -1.116e+00   -1.224e+00   -1.263e+00   -1.273e+00
<D/G>: 3.396e-05    3.691e-05    4.021e-05    4.401e-05    4.874e-05    5.554e-05    6.739e-05    9.234e-05    1.458e-04    6.589e-05
Dust to gas ratio (by mass): 2.374e-03, A(V)/N(H)(pnt):1.296e-022, (ext):8.245e-023, R:3.247e+000 AV(ext):7.158e-003 (pnt):1.125e-002
```

Finally, the **save grain abundances** command give the abundances as a function of depth.

7.10 Metals 0.05 [log, linear, grains; deplete]

This command multiplies the abundances of the entire mixture of metals (elements heavier than helium) by the scale factor entered on the line. This is useful when the effects of global enrichments or depletions of the elements are to be investigated. If the number is less than or equal to zero it is assumed to be the log of the scale factor and the linear scale factor if it is positive. The **linear** and **log** keywords force that interpretation of the number.

Combinations such as

```
abundances planetary nebula
metals 3
```

or

```
metals 3
abundances planetary nebula
```

would multiply the planetary nebula gas-phase abundances by three,[3] while

```
metals -10
```

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

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

7.10.1 Scaling grains and metals together

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

```
grains 0.5
metals and grains 0.5
```

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

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

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

7.11 Metals deplete commands

These specify depletion scale factors D_Z that account for elements that have condensed onto grains. The standard approach is to define the gas-phase abundance of an element Z as

$$A_Z(\text{gas phase}) = A_Z(\text{reference}) \times D_Z \tag{7.1}$$

where the reference abundance $A_Z(\text{reference})$ is the composition the user specifies with other commands and the depletion scale factor D_Z is set with these commands. The final gas-phase abundance results.

There are two versions of this command. The first uses a table of depletion factors that is based on older work and has been included in CLOUDY for some time. The second uses the empirical correlations given by Jenkins (2009) to relate depletions to a global variable he terms F_*.

7.11.1 The depletion factors, reference abundance, and grains

However specified, the depletion factors modify the reference abundances according to Equation 7.1. Other commands are used to specify the reference abundance. The most flexible are the **abundances** family of commands specified in Section 7.4.

The **metals deplete** commands can be combined with the regular **metals** command to rescale the depletion factors. This works because both commands define multiplicative factors to modify the abundances. In the following example, the depletion will be applied to a composition with metallicity that is twice our default. The two **metals** commands can be entered in any order.

```
metals deplete
metals 2.
```

The depleted elements form grains. These commands do not create a set of grain species that is consistent with the depletion. The grain properties are set separately, usually with one of the **grains** commands described in Section 7.9. Specifying grains by themselves (with the **grains** command) does not change the gas-phase abundances, which is not self-consistent. The code will complain if you do this, but still perform the simulation.

The number of depleted atoms that results from the **metals deplete** command are not directly related to the number of atoms in grains, as set with the **grains** commands. These commands do not attempt to conserve mass. Snow and Witt (1996) argue that standard grains have more mass than the heavy elements depleted from a solar abundance. Interstellar grains are likely to be porous with a part of the volume a vacuum. You can judge the potential mismatch by looking at the reported number of atoms in grains and also the number of atoms depleted from the gas phase. These are given in the chemical composition report at the top of the main output.

7.11.2 External data files

These **metals deplete** commands read data from one of the `*.dpl` files located in the `cloudy/data/abundances` directory. Our default file will be used if no filename is specified but the user can specify other files by placing the filename within a pair of double quotes, as in

```
metals deplete "myfile.dpl"
```

The default file that is read if no name as specified is documented in the `readme.md` file located in this directory.

The `readme.md` file in the `abundances` directory also specifies the names of other available data files. Another file, perhaps your own, will be read if a filename appears within double quotes on the command line. This new file must follow the format of the existing files. If you create additional depletion files for your work please consider documenting what you have done by placing comments inside the file and offering it back to the CLOUDY project by posting on the user forum.

7.11.3 The print option

A report giving the depletion factors produced by these commands is produced with the **print** keyword. This is a useful way to check on consistency with expectations. The following is an example.

```
metals deplete print
```

7.11.4 metals deplete - specify a table of depletion scale factors

This is the **metals deplete** command that has long been part of CLOUDY. It specifies a set of depletion scale factors that are roughly those appropriate for relatively dense ISM gas ($\sim 1\,\mathrm{cm}^{-3}$). Table 7.8 lists the depletions that will be assumed if the keyword **deplete** occurs on the **metals** command but no numbers are on the line. These are loosely based on the depletions listed by Jenkins (1987) and Cowie and Songaila (1986). A file specifying the depletions in the Table is included in the distribution and will be used by default. To use your own file, specify its name within double quotes on the command line.

Table 7.8: Depletions

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

7.11.5 metals deplete Jenkins 2009 Fstar=0.5

Jenkins (2009) presents a unified empirical relationship between depletion scale factors of different elements and other properties of the ISM. The overall depletion is parameterized by his depletion strength F_*, which we refer to as Fstar, with the physical limits $0 \leq F_* \leq 1$. In his model, each element is depleted by an empirically-determined depletion scale factor D_z that is related to F_* by his equation 10, the coefficients listed in his Table 4, and the special case of sulphur from Section 9 of his paper. Gunasekera et al. (2022, 2023) give more details and apply these depletions to a model of the Orion Nebula.

Command syntax

The full command syntax is

```
metals deplete Jenkins2009 Fstar=0.5 [vary]  ["MyFile.dpl"]
```

The year 2009 must be the first number on the command line and the next number, F_*, must be between 0 and 1. A space may or may not be placed between the name and year. A typical use case might be

```
abundances nova
metals deplete Jenkins 2009 Fstar=0.5
grains
```

which depletes the specified "nova" abundances by the Jenkins (2009) depletion scale factors with $F_* = 0.5$ and includes our default ISM grain mixture.

The value of F_* is linear by default but the **log** and **linear** keywords allows F_* to be specified either way. The **print** option will vary F_* between 0 and 1 and report the depletion factors.

This command includes the **vary** option, described in Section 17.3 allowing the **grid** command or the optimizer to be used. The **grid** command requires log arguments so a grid of models varying F_* between 10^{-2} and 1 could be generated by

```
metals deplete Jenkins2009 Fstar=-0.5 vary log
grid -2 0 0.1
```

Other details

The Jenkins original work was limited to elements he studied with HST. Some very highly depleted elements were not included in his study. Calcium is very highly depleted, was not studied by Jenkins, and would produce strong lines if left at solar abundances (Kingdon et al., 1995) . Solar-abundance calcium has a significant effect on other lines because of energy balance. Gunasekera et al. (2022, 2023) describe how we deplete elements not considered by Jenkins. Both their updated version of the Jenkins analysis, which includes missing elements, and our version of Jenkins' original file, are available.

Some elements in Jenkins (2009) have depletion scale factors that are greater than unity for small F_*. The physical meaning is that depletion onto grains has increased the abundance of those elements. Jenkins' values are used by default and a comment is produced after the last zone if the depletion scale factor of any element is greater than unity. We provide a **limit** option to specify an upper limit to the depletion scale factor. The following is an example in which D_Z is not allowed to be greater than unity

```
metals deplete Jenkins2009 Fstar=0.5 limit 1
```

The Jenkins (2009) study actually worked with ISM gas-phase abundances scaled relative to the Lodders (2003) solar abundances. We provide both the Lodders (2003) and the more recent Lodders et al. (2009) versions of her solar abundances in the CLOUDY distribution. The Lodders (2003) gas-phase ISM abundances could be specified by

```
abundances "solar_Lodders03.abn"
metals deplete Jenkins2009 Fstar=0.5
grains
```

using the composition stored in the `solar_Lodders03.abn` file in the `abundances` directory. This combination would reproduce the Jenkins work.

Chapter 8

DENSITY LAWS

8.1 Overview

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

A cloud can be isobaric, maintain constant pressure, if the timescale for changes, for instance in the continuum source or the cooling time, is short compared with the dynamical or sound-crossing time t_d

$$t_d = \frac{\Delta r}{c_s} \,[\mathrm{s}] \tag{8.1}$$

where Δr is the cloud thickness and c_s is the sound speed (AGN3 eq 6.25)

$$c_s = \left(\frac{\gamma k T_0}{\mu_0 m_\mathrm{H}} \right)^{\frac{1}{2}} [\mathrm{cm\,s^{-1}}]. \tag{8.2}$$

8.2 Constant density, pressure, gas pressure

This command specifies how the density changes across the cloud. The **hden** command usually specifies the initial hydrogen density. Several commands determine how the density changes with radius. These are described next.

8.2.1 Constant density

This is the default. The hydrogen density, the sum

$$n(\mathrm{H}) = n\left(\mathrm{H}^0\right) + n\left(\mathrm{H}^+\right) + 2n(\mathrm{H}_2) + \sum_{other} n(\mathrm{H}_{other}) \,[\mathrm{cm}^{-3}] \tag{8.3}$$

is kept constant. This is not quite an isochoric density law because the total particle density is not constant—the electron and molecular fractions can vary with depth. I prefer this type of model

because the homology relations with the ionization parameter (Davidson, 1977) are preserved. The hydrogen nucleon density is set with the **hden** command.

8.2.2 Constant gas pressure [index = −1.1]

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

$$P_{gas} = n_{tot} kT_e \, [\mathrm{dyne\,cm^{-2}}] \tag{8.4}$$

where n_{tot} is the total particle density [$\mathrm{cm^{-3}}$], is kept constant. The optional index α can be any value, positive or negative, and will force the pressure to change as a power-law of the radius;

$$P_{gas}(r) = P_{\mathrm{o}} \left(\frac{r}{r_{\mathrm{o}}} \right)^{\alpha} [\mathrm{dyne\,cm^{-2}}] \tag{8.5}$$

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

8.2.3 Constant pressure [no continuum, no abort]

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

$$\begin{aligned} P_{tot}(r) &= P_{tot}(r_o) + \int a_{rad}\, \rho \, dr + \int g \, \rho \, dr \\ &= P_{gas} + P_{ram} + P_{turb} + P_{mag} + P_{lines} + \Delta P_{rad} + \Delta P_{grav} \end{aligned} \quad [\mathrm{dyne\,cm^{-2}}] \tag{8.6}$$

where a_{rad} is the radiative acceleration [$\mathrm{cm\ s^{-2}}$] due to the attenuated incident continuum, g is the (negative) gravitational acceleration [$\mathrm{cm\ s^{-2}}$], and ρ is the gas density [$\mathrm{gm\ cm^{-3}}$].

$P_{gas} = n_{tot} kT_e$ is the thermal gas pressure. The ram pressure is $P_{ram} = \rho u_{wind}^2$, where u_{wind} is the flow velocity. This is only present for a wind geometry. The turbulent pressure is $P_{turb} = \rho \, u_{turb}^2/2$, where u_{turb} is the turbulent velocity set with the **turbulence** command. The magnetic pressure is $P_{mag} = B^2/8\pi$, where the magnetic field B is set with the **magnetic field** command. Both magnetic and turbulent pressures are zero by default. If either dominates the total pressure then the density will be nearly constant.

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

When the outward force due to the attenuation of the incident radiation field and the inward force of gravity are balanced by the pressure gradient, the cloud will be in hydrostatic equilibrium Ascasibar and Díaz, 2010). The required change to the total pressure is given by the integrals and referred to as ΔP_{rad} and ΔP_{grav}, respectively. The **no continuum** option will turn off the ΔP_{rad} term. Gravity must be turned on separately using the **gravity** command described below.

8.2.4 Constant pressure set 6.5

The **set** option specifies the initial pressure. A number, the log of the pressure in ISM nT units (K cm^{-3}), must appear.

An initial gas density must still be specified with one of the commands in this Chapter. This command works by first solving for the equilibrium temperature at the density that was specified. The density is then changed, using a Newton-method solver, to have the pressure match that specified. The solver will raise or lower the density assuming that pressure and density are directly related to one another.

There are possible ambiguities in this solution since the same pressure can sometimes be achieved with more than one density. This is the well-known three-phase stability phenomenon (Field et al., 1969).

8.2.5 Constant pressure timescale = 2.3e9 s, alpha=-1

The pressure changes as a function of time. It is given by

$$P_{\rm o}(t) = P_{\rm o}(t_{\rm o}) \left(1 + \frac{t}{t_{\rm scale}}\right)^{\alpha} [\text{dyne cm}^{-2}] \tag{8.7}$$

The timescale $t_{\rm scale}$ is the first number on the line and the power-law index α is the second. Here $P_{\rm o}(t_{\rm o})$ is the static pressure evaluated at the illuminated face of the cloud.

8.2.6 The reset option

This changes the behavior of the **constant pressure** command. By default the gas pressure in the first zone is derived from the specified hydrogen density and the resulting kinetic temperature. The hydrogen density is the fundamental quantity. In later iterations the initial hydrogen density is kept constant but the kinetic temperature may change as details of the line radiative transfer change. The resulting gas pressure, $\propto nT$, may not have exactly the same value on successive iterations. The total pressure in constant pressure clouds will change slightly as a result. The change in the pressure is small, typically only a few percent.

The **reset** option tells the code to keep the total pressure at the illuminated face constant, from iteration to iteration, rather than the hydrogen density. The **reset** option works by allowing the gas density to change to try to match the total pressure found in previous iterations.

8.3 Gravity [options]

The **gravity** command tells the code to balance the gravitational acceleration by varying the total pressure as described in Ascasibar and Díaz, 2010.

8.3.1 Gravity [spherical, plane-parallel]

One of the keywords **spherical** or **plane-parallel** must appear. Each of these commands specifies the symmetry of the mass distributions and activates self-gravitational pressure.

8.3.2 Gravity external [Mass = 10 $M_\odot$, extent = 3 parsecs, powerlaw = -2]

Additional mass components can be specified by any number of these commands. The first number of represents the mass. The units depend upon the symmetry (which must be specified separately – see above). In the spherical case, this number represents the mass (in solar masses) of a pointlike star located at $r = 0$; in the plane-parallel case, it corresponds to the surface density (in $M_\odot$ pc^{-2}) of a uniform mass sheet at $z = 0$. The number is interpreted as linear unless the keyword **log** is specified. Two optional parameters are the physical extent of the mass component (in parsecs, linear unless **log** appears) and a power-law index. Together these specify the mass accumulation function

$$M_{\rm enclosed} = M(r/r_0)^\alpha. \tag{8.8}$$

The last two parameters can be omitted from right to left. Point sources or infinitesimally thin sheets are the default. The default index is zero.

8.4 Dark [options]

This command provides a gravitational pressure term due to a dark-matter distribution. At present, there is only one option, a standard Navarro et al. (1996) (NFW) profile.

8.4.1 Dark NFW virial radius = 18 [characteristic radius = 17]

This command specifies an NFW profile uniquely specified by their virial and characteristic radii r_{200} and r_s, respectively. The second parameter can be omitted and has the default value $r_s = r_{200}/10$. Both parameters are interpreted as logarithms of distances in cm.

8.5 Dlaw [options]

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

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

It is possible to specify a change in densities so extreme that the code will have convergence problems. CLOUDY works by linearizing all equations. If the density changes dramatically over a very small radius the conditions may change too much for the solvers to converge. The code uses adaptive logic to adjust the zoning and should prevent this from happening but the heuristics may be fooled by drastic density laws. The code will generate a warning if the density does change by too much—if this happens the cure is to not use such large density contrasts.

8.5.1 dlaw p1, p2, p3...

This is the default form of the command. It passes the parameters on the command line to a user-provided function ***dense_fabden***, located in `dense_fabden.cpp`. There are up to ten parameters. A new function ***dense_fabden*** must be written by the user and the version of ***dense_fabden*** already in CLOUDY must be deleted. (The code will stop if the initial version of ***dense_fabden*** is not replaced.) CLOUDY will call ***dense_fabden*** as needed to determine the density as a function of depth. The arguments of the function are the radius and the depth. Both are in centimeters and are double precision variables. The function must return the hydrogen density (cm^{-3}) as a double precision variable. The code provided in the function must use the ten or fewer parameters in the structure *dense* to compute the density at the current position.

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

```
/*dense_fabden implements the dlaw command, returns density using
 * current position and up to ten parameters on dlaw command line */
#include "cddefines.h"
#include "rfield.h"
#include "dense.h"

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

8.5.2 Dlaw wind

This sets up a density profile consistent with a steady-state wind and the continuity equation. The steady-state velocity profile is parametrized as in Springmann (1994):

$$v(r) = v_\star + (v_\infty - v_0)\sqrt{\beta_1 x + (1-\beta_1)x^{\beta_2}}. \tag{8.9}$$

where $x \equiv 1 - r_\star/r$ and we set $r_\star$ to the inner radius of the cloud (except that x has a minimum value 0.01 for computational stability). The mass loss rate into 4π sterradians ($\dot{M}$) then allows the density via continuity:

$$n(r) = \dot{M}/(4\pi m_H \mu r^2 v(r)), \tag{8.10}$$

where μ is the mean molecular weight of the gas. The parameters must be specified in this order: $\dot{M}$, v_∞, β_2, β_1, v_0, $v_\star$. Only the first three are required. $\dot{M}$ should be given as $M_\odot\,yr^{-1}$. All velocities should be given as $km\,s^{-1}$. The final three may be omitted right to left and take default values $\beta_1 = v_0 = v_\star = 0$. All values are interpreted as linear values.

This command sets the hydrogen density but does not create a wind. It creates a density profile that mimics a wind. The **dlaw** command (with or without the **wind** keyword) and the **wind** command described in section 14.5 are mutually exclusive, and you will just end up with whichever one appeared last. There is no way to combine the two.

This command is experimental.

8.5.3 Dlaw table [depth, radius]

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

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

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

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

8.6 Fluctuations density period ...

This specifies a density that varies as a sine wave. It was introduced to investigate the effects of inhomogeneities upon the emission-line spectrum (see Mihalszki and Ferland, 1983; Kingdon and Ferland, 1995). The first number is the log of the period P of the sine wave in centimeters. The second two numbers are the logs of the largest and smallest hydrogen densities over the sine wave. Order is important here. The last optional number is a phase shift φ (in radians) which allows the initial zone to occur at any part of the sine wave. If it is omitted the calculation will begin at the maximum density. If the phase is set to π the calculation will start at the minimum density.

The density is scaled according to the relation

$$n(r) = \left(\frac{n_{\max} - n_{\min}}{2}\right) \times \cos\left(\Delta r \frac{2\pi}{P} + \varphi\right) + \left(\frac{n_{\max} + n_{\min}}{2}\right) \text{ [cm}^{-3}\text{]} \qquad (8.11)$$

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

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

The keyword **column** will replace the depth variable in equation 8.11 with the total hydrogen column density. The period should then be the column density of one cycle. This will more evenly weight the final column density over low- and high-density gas.

The **fluctuations abundances** command described in Section 7.8 proves a mechanism for varying the gas-phase abundances of the elements.

8.7 Globule [density =2, depth =16, power =2]

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

$$n(r) = n_o \left(\frac{R_{scale\,depth}}{R_{scale\,depth} - \Delta r} \right)^{\alpha} = n_o \left(1 - \frac{\Delta r}{R_{scale\,depth}} \right)^{-\alpha} [\mathrm{cm}^{-3}] \tag{8.12}$$

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

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

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

$$n(\mathrm{H}) = n\left(\mathrm{H}^0\right) + n\left(\mathrm{H}^+\right) + 2n(\mathrm{H}_2) + \sum_{other} n(\mathrm{H}_{other})\,[\mathrm{cm}^{-3}]. \tag{8.13}$$

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

For situations where the hydrogen atom is close to LTE and the gas is hot, there is a problem in defining the neutral hydrogen density because of the well-known divergence of the partition function, as discussed, for instance, by Mihalas (1978). The atomic hydrogen density is defined as the total population in all computed levels. In most circumstances, i.e., $n(\mathrm{H}) \leq 10^{15}$ cm^{-3} and $T \leq 10^4$ K, the ambiguity is much less than 1%.

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

8.8.1 Power-law dependence on radius

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

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

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

i.e.,

$$n(r) = n_{\rm o}(r_{\rm o}) \left(\frac{r}{r_{\rm o}}\right)^{\alpha} [{\rm cm}^{-3}]. \tag{8.14}$$

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

8.8.2 Clouds extending to infinity

For an inverse-square density law there is a critical value of the number of ionizing photons emitted by the central object, corresponding to an ionization front at infinite radius;

$$Q_{crit}({\rm H}) = \alpha_B(T_e)\, n_{\rm o}^2\, 4\,\pi\, r_{\rm o}^3\ [{\rm s}^{-1}]. \tag{8.15}$$

A hydrogen ionization front will not be present and the model will extend to infinite radius when $Q({\rm H}) \geq Q_{crit}(H)$. In this expression $\alpha_B(Te)$ is the hydrogen Case B recombination coefficient and n_o and r_o are the inner density and radius respectively. Generally, a hydrogen ionization front will not be present if the density falls off faster than an inverse square law, but rather the level of ionization will tend to *increase* with increasing radius. In either case, if a reasonable outer radius is not set, the calculation will extend to very large radii, an unphysically small density will result, and usually the code will crash due to floating point underflow, followed by division by zero. It is usually necessary to set an outer radius when the density falls off with an index $\alpha \leq -2$, since, for most circumstances, the cloud will remain hot and ionized to infinite radius and zero density.

8.8.3 Power-law dependence on depth

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

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

The density is given by

$$n(r) = n_{\rm o}(r_{\rm o}) \left(1 + \frac{\Delta r}{R_{scale}}\right)^{\alpha} [{\rm cm}^{-3}] \tag{8.16}$$

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

8.8.4 Power-law dependence on column density

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

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

Here the density is given by

$$n(r) = n_{\rm o}(r_{\rm o}) \left(1 + \frac{N({\rm H})}{N({\rm H})_{scale}}\right)^{\alpha} [{\rm cm}^{-3}] \tag{8.17}$$

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

Chapter 9

GEOMETRY

9.1 Overview

This section describes commands that determine the geometry of the emission-line region. Many of the quantities used below are defined in the section on geometry in the Chapter *DEFINITIONS*.

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

9.2 Age 44 years [off]

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

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

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

9.3 Aperture commands

9.3.1 Aperture [slit, beam]

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

One of the keywords **slit** or **beam** must appear. The keyword **beam** tells the code to simulate observing a resolved nebula with a small pencil beam centered very close to the central source. If the keyword **slit** appears then the computed structure is simulated to be observed with a long slit that extends over the *entire* width of the nebula, also positioned very close to the center. In the long slit case, it is assumed that the observer has added up all the flux from the source over the full length of the slit. It is not now possible to simulate an off-axis long slit or pencil beam.

The **aperture** command only affects how volume emissivities are added together to form the final line spectrum printed in the main CLOUDY output and saved with the **save lines** command. When optimizing line fluxes, or line flux ratios, the quantities observed through the aperture will be used. This command has no effect on any aspect of the calculation of conditions in the gas. It also does *not* affect the spectral energy distribution predicted with the **save continuum** commands. This will always be integrated over the entire nebula. The continuum bands in the file `data/continuum_bands.dat` are not predicted when the **aperture** command is used, for this reason.

This command only affects the line luminosities in the luminosity case. When the intensity case is used and the inner radius is not specified all integrations are for a pencil beam through a plane-parallel slab.

In the following a quantity α is defined with the following meaning: $\alpha = 0$ in the pencil beam case (we are observing along a single line of sight passing through the center of the nebula), $\alpha = 1$ in the long-slit case (we are observing through a narrow slit placed over the center of the nebula; the slit is longer than the nebula and the flux is integrated over the entire slit), and $\alpha = 2$ in the general case (we are observing the flux integrated over the entire nebula). The default index is $\alpha = 2$.

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

$$Q_\alpha(\lambda) = C_\alpha D_\alpha \int \left(\frac{r}{r_0}\right)^\alpha \varepsilon(\lambda)\, dr, \tag{9.1}$$

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

$$\begin{aligned} C_0 &= 2 \\ C_1 &= 2\pi r_0 \\ C_2 &= 4\pi r_0^2 \end{aligned}, \tag{9.2}$$

where r_0 is the inner radius of the nebula. The covering factor D_α depends on the geometry and is

$$\begin{aligned} D_0 &= 1/2,\ 1 \\ D_1 &= \Theta/2\pi \\ D_2 &= \Omega/4\pi \end{aligned}. \tag{9.3}$$

For the entire nebula ($\alpha = 2$) this is the familiar definition (see also Sect. 9.5). In the long slit case ($\alpha = 1$), D_1 is the fraction of a large circle in the plane that is being observed that actually passes through nebular material. In the pencil beam case ($\alpha = 0$), D_0 indicates whether only the front or back side of the line of sight is covered with nebular material ($D_0 = 1/2$) or if both sides are covered ($D_0 = 1$). The covering factor for the entire nebula is set with the **covering factor** command. This is the quantity D_2. You can set the quantities D_0 or D_1 using the **aperture covering factor** command described below.

In the luminosity case Q_α will have units erg s^{-1} for the entire nebula, erg cm^{-1} s^{-1} for the long slit case, and erg cm^{-2} s^{-1} for the pencil beam. In the intensity case the units are erg cm^{-2} s^{-1}. The luminosity and intensity cases are defined on page 13.

Q_α is the quantity printed in the main CLOUDY output and by the **save lines** command in the default setup. However, if the distance to the object is set with the **distance** command, we can convert this quantity into an observed flux at Earth. In the luminosity case we can construct the following relations between observed quantities and the quantities predicted by the code. These neglect interstellar extinction of course. The flux measured at the Earth will be

$$F_\alpha(\lambda) = \frac{A_\alpha}{4\pi D^2} Q_\alpha(\lambda) \text{ [erg cm}^{-2}\text{s}^{-1}\text{]}, \tag{9.4}$$

with

$$\begin{aligned} A_0 &= \Omega_b D^2 \\ A_1 &= \Theta_s D \quad . \\ A_2 &= 1 \end{aligned} \tag{9.5}$$

Here F_α is the observed flux [erg cm^{-2} s^{-1}] D is the distance of the object [cm], as specified with the **distance** command, Θ_s is the slit width [radian], and Ω_b is the surface area of the pencil beam [sr].

If the **print line flux at Earth** command is used, then CLOUDY will do this conversion for you and F_α will be reported in the main CLOUDY output as well as the **save lines** output. F_α will then also be used when optimizing line fluxes. You can set the value of Θ_s or Ω_b with the **aperture size** command described below.

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

$$S(\lambda) = \frac{\Omega\left(1 \text{ arcsec}^2\right)}{4\pi} Q_0(\lambda) \approx 1.87043 \times 10^{-12} Q_0(\lambda) \text{ [erg cm}^{-2} \text{ s}^{-1} \text{ arcsec}^{-2}\text{]}. \tag{9.6}$$

9.3.2 Aperture size 2.5

This command should be used in conjunction with the **aperture [slit, beam]** and **print line flux at Earth** commands. It will have no effect otherwise. It allows you to set the slit width [arcsec] or the surface area of the pencil beam [$arcsec^2$]. These are the quantities Θ_s and Ω_b used in Eq. 9.5. The quantity entered will always be treated as a linear quantity. If this command is not used, the code will assume a default slit width of 1 arcsec, or a beam area of 1 $arcsec^2$.

9.3.3 Aperture covering factor 0.6

This command should be used in conjunction with the **aperture [slit, beam]** command. It allows you to set the quantity D_α defined in Eq. 9.3. A single number should appear on the command line which should be between 0 and 1. The default value is 1.

9.4 Clumping factor = 0.05 [index =-1]

This is a another name for the **filling factor** command described on page 111. It takes the same options, and includes the same physics, as described there.

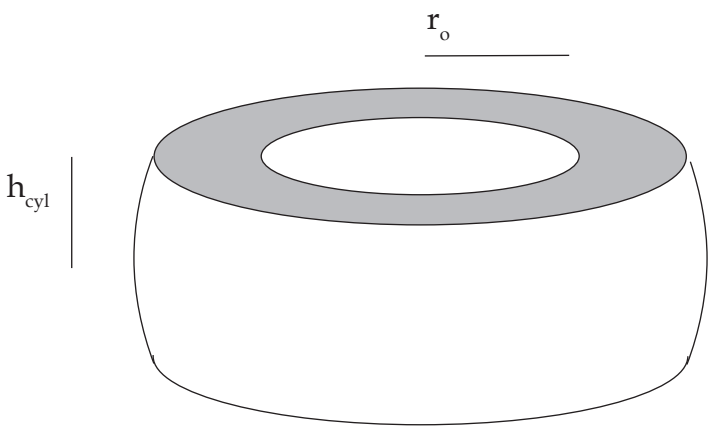

Figure 9.1: This figure shows the geometry assumed when the **cylinder** command is used.

9.5 Covering factor 0.3

This sets a covering factor $\Omega/4\pi$ for the emission-line region (see AGN3 section 5.9). The argument is the log of the covering factor if less than or equal to zero and the linear covering factor if positive. It is impossible to specify a covering factor of zero.

The covering factor affects both the luminosity of the emitted spectrum and the radiative transfer of lines and continua. If a covering factor is set and the luminosity case used then the luminosities will be for a shell covering Ω sr of the central object. Line luminosities will scale nearly linearly with the covering factor. The covering factor does not strongly affect the line intensities in the intensity case, where the emission per unit area of cloud is considered. The covering factor does have second-order effects on the spectrum because of changes in the transport of the diffuse radiation fields.

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

If no covering factor is entered and **sphere** is not set (see section 9.11) then the default is for a geometric covering factor of unity (the shell fully covers the continuum source) but a radiative covering factor of zero (i.e., an open geometry).

9.6 Cylinder log semi height =9.12

The model will be spherical, but truncated so as to simulate a cylinder (see Ferland et al., 1982). Figure 9.1 gives an example of the assumed geometry.

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

The effective volume element used to compute the emissivity of the gas is given by

$$dV = 4\pi r_o^2 \left(\frac{r}{r_o}\right) \left(\frac{\min\left(r,\, h_{cyl}\right)}{r_o}\right) f(r)\; dr\,[\mathrm{cm}^3] \tag{9.7}$$

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

Changing the emissivity as described by equation 9.7 is the only effect of this command. It does not alter the radiative transfer methods and is only formally correct when the continua and lines are optically thin.

When combined with the **aperture slit** command, it will be assumed that the slit is oriented along the rotational axis of the cylinder. If the slit is oriented at an angle, you can model this geometry by dividing the height of the cylinder by the cosine of the angle. If the slit is perpendicular to the rotational axis you can simply omit the cylinder command since it will have no effect on the predicted spectrum.

9.7 Distance 3.2 linear parsecs

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

If the distance is set then it is possible to predict the emission-line fluxes observed at Earth if the luminosity case is computed. If the distance is set with this command then the observed emission-line fluxes at the Earth will be printed if the **print line flux** command is also entered.

This command can be combined with the **aperture** command to simulate observing parts of a nebula from the Earth.

9.8 Filling factor = 0.05 [index =-1]

The first number is the filling factor $f(r)$ for a clumpy model (AGN3 Section 5.9). It can be either the filling factor itself (which is greater than zero and less than or equal to one) or the log of the filling factor (if it is less than or equal to zero). The second optional number is the index for a power-law variation of the filling factor $f(r)$, i.e.,

$$f(r) = f(r_o) \left(\frac{r}{r_o}\right)^{\alpha} \tag{9.8}$$

where r_o is the inner radius of the cloud.

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

$$dE = \varepsilon\, f(r)\; dV \frac{\Omega}{4\pi}\,[\mathrm{erg\,s}^{-1}] \tag{9.9}$$

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

$$d\tau = \alpha_{l,u} \left(n_l - n_u \frac{g_l}{g_u}\right) f(r)\, dr \tag{9.10}$$

(see Osterbrock and Flather, 1959).

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

9.9 Illumination isotropic | angle 45 deg [radian]

This command specifies the orientation of the radiation field with respect to the slab that is being illuminated. It applies *only* to the last radiation field that was specified preceding this command. If the keyword **isotropic** is present, it indicates that the radiation field is isotropic, i.e. coming from all directions. Alternatively, if this keyword is absent, you can specify that the slab is illuminated by a unidirectional beam at an angle θ away from the normal. The default is $\theta = 0$ (illumination perpendicular to the slab). The angle is in degrees unless the keyword **radian** appears.

The only effect of this command is to cause the beam of incident radiation to be attenuated by $\tau_n / \cos\theta$ where τ_n is the optical depth of the zone in the direction of the normal. The line and diffuse continua optical depth scale, which is defined normal to the plane, are not directly affected by this command as these are not part of the incident radiation.

This command also accepts the optional keywords **forward**, **reverse**, and **symmetric**, indicating whether the the front face, the rear face, or both faces, of the gas are illuminated. Illumination from the rear is currently not implemented, so these options have no effect and a caution to that effect will be printed.

9.10 Radius r(inner) [r(outer), thickness; parsec; linear]

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

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

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

The default outer radius is effectively infinite (actually 10^{31} cm). If the intensity case is used then a starting radius of 10^{30} cm will be set by default. Under most circumstances this will result in a plane-parallel geometry.

Page 256 describes a problem that can occur if the second parameter is used with the **vary** option. Please read it if you will vary the radius with the outer radius set. The **stop thickness** command provides a way to set a stopping thickness without specifying a starting radius.

9.11 Sphere [options]

CLOUDY normally assumes an open geometry, i.e., that the gas covering factor is small. The **sphere** command[1] [2] should be included to change this assumption for a closed geometry, one where the covering factor of the gas is large and the model spherical. This command tells CLOUDY to take into account ionization by the diffuse continua and lines produced in the far side of the nebula (i.e., from beyond the central object), and not to attenuate the ionizing continuum by pure scattering opacities, such as electron scattering, back scattering by grains, or Rayleigh scattering.

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

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

9.11.1 Sphere expanding or static

Two optional keywords, **expanding** and **static**, determine how line transfer is handled. If **expanding** (the default when **sphere** is entered) is set then CLOUDY assumes that line photons escaping from the illuminated face of the cloud are Doppler shifted away from lines of absorbing material on the far side of the shell. This will be the case if the expansion velocity exceeds the Doppler width by large amounts. If **static** is set then line photons do interact on both sides so that even line photons produced at the illuminated face of the cloud will be strongly trapped by material on the far side. $L\alpha$ radiation pressure in the H^+ region will probably be significant if **sphere static** is set and grains are not present.

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

The specific effects of **sphere** are the following: The total continuous optical depths are assumed to be twice the computed optical depths, and the initial optical depth is half the total. All diffuse reemission (bremsstrahlung, free-bound, etc.) is counted in the outward beam rather than only half. Scattering opacities are not considered in the attenuation of the incident radiation field. When **static** is set, the optical depth in Lα in the inner direction is set to 10^5 on the first iteration. Otherwise it is 10^{-20}. The total optical depths of lines are twice their computed depth for the static case. Finally, ionization by lines and continua produced in the far side of the nebula is included. At the end of the iteration all inward optical depths are set to half of the total value computed from the previous iteration.

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

[2] Before version 96 the **sphere** command included an option to change the covering factor. The covering factor was removed from the **sphere** command. Only the **covering factor** command changes the covering factor.

9.12 Stop depth...

9.13 Stop thickness...

The **stop depth** and **stop thickness** commands provide methods to set the thickness of a cloud without specifying its radius. They are described in other sections.

Chapter 10

OPTICAL DEPTHS AND RADIATIVE TRANSFER

10.1 Overview

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

It is necessary to iterate upon the solution if emission lines are optically thick since total optical depths are not known on the first iteration. The default is for a single iteration. This is often adequate for low-density nebulae such as planetary nebulae or H II regions. A second iteration is sometimes enough to establish a fairly accurate line optical depth scale for many resonance transitions. Further iterations are usually needed when subordinate lines are also optically thick. The **iterate to convergence** command will iterate until the optical-depth scale is well defined.

Line-radiation pressure cannot be computed accurately until the total line optical depths are known, so this quantity is only meaningful after the first iteration. CLOUDY will stop if the internal radiation pressure exceeds half of the surface gas pressure in a constant-pressure model since such a geometry is unstable unless it is self-gravitating. The radiation pressure is not allowed to exceed half the gas pressure on the initial iterations of a multi-iteration constant-pressure model. This is to prevent the calculation from stopping when the optical depth scale is not yet well converged.

The following sections outline various commands that affect the radiative transfer.

10.2 Case A [options]

This has the same options as the Case B command but sets the Lα optical depth to a very small value by default. This does not turn off induced processes, which are normally ignored when Case A is assumed. You would use the **no induced processes** command to do that.

10.3 Case B [tau ly alpha = 9; options]

This command is used to check the line emission from hydrogen-like and helium-like species in the Case B limit (AGN3 section 4.2). This command *should not* be used in any model that is supposed to represent a real physical environment. It is intended only to provide an easy way to check predictions of the code against simple, more limited, calculations. In particular, when this is used the Lyman-line optical depths will be made artificially large. This may affect the ionization and temperature of the gas.

With no options, this command sets the inward-looking optical depth of Lα for all atoms and ions of the H- and He-like iso-electronic sequences to 10^5 so that even a one-zone model will be close to Case B.[1] This optical depth does not depend on the abundance of the element, or even the presence of the species. The optional number is the log of the Lα optical depth. One-sided escape probabilities are used so the total escape probability is simply that for the inward direction. In keeping with the Case B approximation the **Case B** command suppresses excited-state line optical depths.

The Hummer option. The species include all collisional processes. Case B does not define the population of the ground or first excited state so a true comparison with Case B results should have collisions from these levels turned off. This is done with the Hummer and Storey option (with the key **Hummer**), to allow comparison with their 1987 and 1995 papers. Collisions from the ground and first excited states *are* included if this second option is not specified. Collisions between levels with $n \geq 3$ *are* included unless the **database H-like collision off** or **database He-like collision off** commands are given. Collisions between the $2s$ and $2p$ levels are always included unless the **database H-like collisions l-mixing off** command is given. Similarly, this option uses the Percival and Richards (1978) n-changing collisions and the classical Pengelly and Seaton (1964) theory for l-changing collisions.

In the case of the He-like isoelectronic sequence the **Case B** command sets the optical depths in the singlet Lyman lines to a large value. The Hummer & Storey option has no effect on the He-like sequence.

The **no Pdest** option turns off destruction of Lyman lines by background opacity.

There are several side effects of this command that may after the spectrum or physical conditions in unexpected ways. The large Lα optical depth will often result in an especially strong radiation field within this line. This affects the gas through photoionization of excited metastable states of H and He and of those elements with a small enough ionization potential. The **no photoionization** option on the **Case B** command tells the code not to include photoionization from excited states of H^0. But these strong diffuse fields will also strongly affect the level of ionization of the gas, making the resulting ionization equilibrium a fiction. Optically thin gas is actually described by Case C (Ferland, 1999b, Luridiana et al., 2009 and AGN3 Section 11.4) where continuum pumping enhances Balmer lines. The large Lyman-line optical depths that result from the **Case B** command will prevent continuum resonant pumping of the atom. Beware.

[1] Before version 96 the default optical depth was 10^9. This caused extreme Lα behavior in a grain-free H II region. The lower value is a better estimate of the physics that occurs in an actual H II region.

10.4 Case C [options]

This has the same options as the Case B command but sets the Lα optical depth to a very small value by default. Case C is described by Ferland (1999b) and Luridiana et al. (2009).

10.5 Diffuse fields [outward, OTS]

This specifies which method is to be used to transfer the diffuse fields, the emission from gas within the computed structure. The options are **outward only** and **OTS**.

The **OTS** option takes into account optical depths in both the inward and outward directions. The **OTS** option has a **SIMPLE** option which will do a very simple OTS approximation without taking optical depths into account. All diffuse fields with energies capable of ionizing hydrogen are assumed to do so, and those with smaller energies freely escape. This is intended as a debugging tool.

If **outward** is chosen then the code will check for a number. This determines which of the many forms of the outward-only approximation (Tarter, 1967) is used. The default[2] is 2. This is intended for testing the code.

This choice does not strongly affect the predicted emission-line spectrum but it does change the temperature at the illuminated face of the cloud.

10.6 Double optical depths

On second and later iterations the code uses the total optical depths of the computed structure to find the outwardly-directed radiation field. This command doubles the total optical depth so that the shielded face of the cloud becomes the mid-plane of a structure that is twice as thick as the computed cloud.

This original purpose of this command was to simulate a geometry in which ionizing radiation strikes the plane-parallel cloud from both sides. Examples are a Lα forest cloud or the diffuse ISM. The total line and continuum optical depths are set to twice the computed optical depth at the end of the iteration. The computed model is then one half of the cloud and the other half of the cloud is assumed to be a mirror image of the first half. Doubling the total line and continuum optical depths at the end of the iteration is the *only* effect of this command. Physical quantities such as the physical thickness, column densities, or line emission *are not* affected.

This approximation makes sense if the cloud is optically thick in lines but optically thin (or nearly so) in continua. Lines such as the Lα transitions of He I and He II can be important sources of ionizing radiation. Their transport will be handled correctly in this limit when this command is used. Continuum transport out of the cloud will also be treated correctly, but attenuation of the incident continuum will *not* be if the cloud is optically thick in the continuum.

The second use of this command is when the outer edge of a computed structure is not the other edge of the cloud. A typical PDR calculation is an example. The calculation starts at the illuminated face and continues until the gas becomes cool and molecular. The stopping point often does not correspond to the outer boundary of the molecular cloud, but rather is a point that is

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

"deep enough" for a given study. The optical depths are always computed self-consistently. On second and later iterations the total optical depths are normally those of the computed structure. Near the shielded face the outward optical depths will be small and radiation will freely escape in the outward direction. The gas temperature may fall dramatically due to the enhanced cooling resulting from the free escape of line photons. In real PDRs considerable neutral or molecular material probably extends beyond the stopping point so that line photons do not freely escape. The shielding effects of this unmodeled extra material can be included with this command. Then, the shielded face of the cloud will correspond to the mid-plane of the overall structure and lines will not artificially radiate freely into the outer (unmodeled) hemisphere.

10.7 Iterate [2 times, to convergence]

This specifies the number of iterations to be performed. The default is a single iteration, a single pass through the model. At least a second iteration should be performed in order to establish the correct total optical depth scale when line transfer or radiation pressure is important. Two iterations are sometimes sufficient and will be done if no numbers are entered on the command line. A comment will be printed after the last iteration if the total optical depth scale has not converged and further iterations are needed.

10.7.1 Number of iterations

There is a slight inconsistency in how the code counts the number of iterations. The way it functions in practice is what makes the most sense to me.

The word *iterate* is from Latin for "again." So the true number of "agains" should be one less than the total number of calculations of the cloud structure. When the **iterate** command is not entered there is one calculation of the structure and so formally no iterations. If any one of the following commands is entered:

```
iterate
iterate 0
iterate 1
iterate 2
```

then exactly two calculations of the structure will be done. If the number on the line is two or greater, then the number will be the total number of calculations of the structure.

10.7.2 Iterate to convergence [max =7, error =.05, all]

This is a special form of the **iterate** command in which the code will continue to iterate until the optical depths have converged or a limit to the number of iterations has been reached. The optional first number on the line is the maximum number of iterations to perform with a default of 10. The second optional number is the convergence criterion. The default is for relative optical depths to have changed by less than 0.20 between the last two iterations. The optional numbers may be omitted from right to left. If all transitions are optically thin then only a second iteration is performed.

If the **all** option is given, the convergence test will be made on all transitions, rather than a selected set.

Section 10.7.3 discusses some reasons the simulation may not converge. There are likely to be convergence problems if the outer edge of the cloud is set by the lowest allowed temperature rather than an outer radius of column density.

10.7.3 Convergence problems

The code generally will not converge if it has not done so within ten or so iterations. The most common reason for convergence problems is that the outer edge of the cloud changes from iteration to iteration. To prevent this from happening it is important to understand why the calculation stopped (Section 15).

Convergence problems often happen if the outer edge of the cloud is set by the lowest allowed temperature, as set with the **stop temperature** command (Section 15.17), or the default lowest allowed temperature. The temperature at the shielded face of the cloud can be affected by the total optical depths. As a result the point where the lowest temperature is reached can change from iteration to iteration, so the column densities and optical depths also change, independently of the radiative transfer solution. It will not be possible to converge the optical depths since so much is changing. The code will generate a caution if the solution is not converged after the limit to the number of iterations is reached. In that case it will also check if the calculation stopped due to the low-temperature limit being reached, and suggest changing the stopping criteria in that case.

Another common reason for convergence problems is that the specified column density or thickness causes the simulation to end within a prominent ionization front. In this case very small changes in the physical conditions result in large changes in the optical depths.

These are physical, not numerical, problems. To prevent them, understand what sets the outer edge of the cloud. The code should not have convergence problems if the outer edge is determined by the outer radius, total column density, or an equivalent, such as A_V.

10.8 No scattering opacity

This turns off several pure scattering opacities. These include scattering by grains, electron scattering, and the extreme damping wings of Lyman lines (Rayleigh scattering). When scattering opacity is included and an open geometry is computed the scattering opacity is assumed to attenuate the incident radiation field as $(1+0.5\,\tau_{scat})^{-1}$ rather than $\exp(-\tau)$ (Schuster, 1905).

Scattering can be neglected in a spherical geometry with gas fully covering the source of ionizing radiation. Scattered photons are not really lost but continue to diffuse out with (perhaps) a slight shift in energy. Electron scattering is generally the most important scattering opacity in a grain-free mixture. If $\tau_{scat} \leq 1$ then it is reasonable to consider electron scattering as a heating and cooling process but not as an absorption mechanism if the energy shifts are not large (i.e., $h\nu \ll mc^2$) and the geometry is spherical (this is not correct for γ-ray energies, of course). CLOUDY is not now designed to work in environments that are quite Compton thick, but should work well for clouds where the electron scattering optical depths are less than or of order unity.

When this command is entered scattering processes such as Compton energy exchange and grain scattering are still included as heating, cooling, and ionization processes, but not as

extinction sources. (Thermal and ionization effects of Compton scattering are turned off with the **no Compton** command). The **no scattering opacity** command is automatically generated when **sphere** (see section 9.11) is specified.

10.9 Turbulence = 100 km/s [log, dissipate]

This enters a microturbulent velocity u_{turb}. The velocity is given in km s^{-1} on the command line although the code works with cm s^{-1} internally. The turbulent line width u_{turb} is zero by default, but the value that is entered must be $u_{turb} \geq 0$. If the optional keyword **log** appears then the number is interpreted as the log of the turbulence. Alternatively, you can also enter the keyword **equipartition** which is discussed further below.

Turbulent pressure is included in the equation of state when the total pressure is computed.[3] The **no pressure** option on this command says not to include turbulent pressure in the total pressure.

Turbulence affects the shielding and pumping of lines. Fluorescent excitation of lines becomes increasingly important for larger turbulent line widths since a larger part of the continuum can be absorbed by a line. Line pumping is included as a general excitation mechanism for all lines using the formalism outlined by Ferland (1992) and described further in a section of Part 3. The line-center optical depth varies inversely with the line width velocity u so the effects of line optical depths and trapping become smaller with increasing line width. Larger u inhibits self-shielding.

10.9.1 Definitions

The Doppler width of any line that is broadened by both thermal motions and turbulence that is given by a Gaussian is given by the quadratic sums of the thermal and turbulent parts,

$$u = \sqrt{u_{th}^2 + u_{turb}^2} = b = \sqrt{2}\sigma \, [\mathrm{cm\,s^{-1}}]. \tag{10.1}$$

Here b is the Doppler parameter used in much of the UV absorption-line literature, σ is the standard deviation (often called the dispersion) for a normal distribution, and σ^2 is the variance. For a thermal distribution of motions the average velocity along the line of sight (Mihalas, 1978, equation 9-35, page 250) and the most probable speed (Novotny, 1973, p 122) of a particle with mass m are both given by

$$u_{th} = \sqrt{2kT/m} \, [\mathrm{cm\,s^{-1}}]. \tag{10.2}$$

This corresponds to $16.3\sqrt{t_4/m_{AMU}}$ km s^{-1} for pure thermal motions. This command sets u_{turb}. Note that with these definitions the full-width half-maximum (FWHM) of a line is equal to

$$u\,(FWHM) = u\sqrt{4\ln 2} \, [\mathrm{cm\,s^{-1}}]. \tag{10.3}$$

10.9.2 Turbulent pressure

For an ideal gas the thermal pressure is $P = nkT$ while the energy density is $1/2nkT$ per degree of freedom, so for a monatomic gas is $U = 3/2nkT$. Turbulent motions add pressure and energy terms that are analogous to thermal motions if the turbulence has a Gaussian distribution. Note

[3] Turbulence was not included as a pressure term in versions 06.02 and before.

that the turbulent velocities may be organized, or lined up in certain directions, if the gas is magnetically controlled. This presents a complication that is ignored.

Heiles and Troland (2005) discuss both turbulence and magnetic fields. Their equation 34 gives the turbulent energy density in terms of the one-dimensional turbulent velocity dispersion $\Delta V^2_{turb,1D}$ (or standard deviation). Note that we write velocities in terms of u with the relationship $u^2 = 2\Delta V^2_{turb,1D}$. Their energy density can be rewritten as a pressure:

$$\begin{aligned} P_{turb} = \tfrac{F}{6}\rho\, u^2_{turb} &= 3.9\times10^{-10} F \left(\tfrac{n_{tot}}{10^5\ \mathrm{cm^{-3}}}\right)\left(\tfrac{u_{turb}}{1\ \mathrm{km\ s^{-1}}}\right)^2 \quad \left[\mathrm{erg\ cm^{-3}};\ \mathrm{dyne\ cm^{-2}}\right] \\ &= 2.8\times10^{6}\ F \left(\tfrac{n_{tot}}{10^5\ \mathrm{cm^{-3}}}\right)\left(\tfrac{u_{turb}}{1\ \mathrm{km\ s^{-1}}}\right)^2 \quad \left[\mathrm{cm^{-3}\ K}\right] \end{aligned} \tag{10.4}$$

where n_{tot} is the total hydrogen density, u_{turb} is the turbulent velocity, and He/H= 0.1 was assumed. The term F accounts for how the turbulent velocity field is ordered (Heiles and Troland, 2005, their equation 34). F is 2 for turbulent velocities that are perpendicular to the magnetic field as in Alfven waves and F is 3 for isotropic turbulent motions. The default value of $F = 3$ is changed by entering a new value as a second number on this command line.

10.9.3 Energy dissipation

The **dissipate** option on the **turbulence** command provides a way in include conversion of wave energy into heat (see Bottorff and Ferland, 2002). When the option is used a third number, the log of the scale length for the dissipation in cm, must appear. Then the turbulent velocity will have the form

$$u_{turb}(r) = u_{turb}(r_o)\exp(-\Delta r/r_{scale}) \quad [\mathrm{cm\,s^{-1}}] \tag{10.5}$$

where $u_{turb}(r_o)$ is the turbulence at the illuminated face and Δr is the depth into the cloud. The wave mechanical energy is assumed to have been converted into heat with a local heating rate given by (Bottorff and Ferland, 2002)

$$G(r) = 3.45\times10^{-28} 2^{-3/2} u^3_{turb}(r) \quad [\mathrm{erg\,cm^{-3}s^{-1}}] \tag{10.6}$$

10.9.4 Equipartition turbulent - magnetic pressures

The **equipartition** option on the **turbulence** command sets the turbulent velocity to an equipartition between magnetic and turbulent energy densities. That is,

$$P_{turb} = \frac{F}{6}\rho\, u^2_{turb} = P_{mag} = \frac{B^2}{8\pi}\,[\mathrm{erg\,cm^{-3}}]. \tag{10.7}$$

The **magnetic field** command sets B. The turbulent velocity is determined from the magnetic field assuming equation 10.7. Heiles and Crutcher (2005) argue that the correlation between turbulent and magnetic pressures, while true on average, does not hold in detail for specific regions of the ISM. When this option is used, you can optionally add the parameter F on the command line (the default is $F = 3$). The keyword **no pressure** is supported, but not **dissipate**.

10.9.5 Turbulence command heads up!

N.B.! In the default (non-equipartition) form of the command, the turbulent velocity is the first number on the command line. The F parameter is the second number. The energy-dissipation scale length is the third number and must appear after u_{turb} and F. These numbers can only be omitted from right to left. In the equipartition case you can only specify F, which then is the first number on the command line. The energy-dissipation scale length is not supported in the equipartition case. The keyword **vary** is only supported in the non-equipartition case, in which case the turbulent velocity is varied.

The turbulent velocity must be less than the speed of light. The code will stop if $u_{turb} \geq c$ is specified.

10.10 vlaw alpha=-1

This specifies a turbulent velocity that is a power law in radius. The number is the power law α on radius. It must be negative. The turbulence will be given by

$$u = u_0 \left(r / r_0 \right)^{\alpha} \tag{10.8}$$

where r_0 in the inner radius and u_0 is the initial turbulence specified with the **turbulence** command.

Chapter 11

THERMAL SOLUTIONS

11.1 Overview

This Chapter describes options that affect the thermal solution and the gas kinetic temperature.

The accuracy of the thermal solution is set by the error tolerance in the heating—cooling balance. This is set with the **set temperature tolerance** command. Other commands can change some details of the thermal solution.

11.2 cextra -14.231 [temp to the 1.5 power]

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

$$\Lambda = 10^{c_1} \times \left(\frac{T_e}{10^4 \text{ K}} \right)^{c_2} [\text{erg cm}^{-3}\text{s}^{-1}] \tag{11.1}$$

where c_1 and c_2 are the two numbers entered with this command. If the second optional argument c_2 is not specified then zero (i.e., constant cooling) is assumed.

The function evaluating **cextra** sets the variable *cextxx*. The expression can be easily changed to other forms by changing how this variable is set.

11.3 Constant temperature, t=1e4 K [linear]

A constant gas kinetic temperature calculation will be performed. The number is the electron temperature and is interpreted as the log of the value if it is ≤ 10 or if the keyword **log** is present. The optional keyword **linear** forces interpretation as a linear quantity. The temperature must be within the temperature limits of the code, currently $2.8\,\text{K} \leq T \leq 1.001 \times 10^{10}\,\text{K}$.

By default the temperature is specified in Kelvin. The keywords **eV** and **keV** are also accepted.

Collisional ionization of all atoms and ions is included so this option can be used to produce clouds in coronal or collisional equilibrium.

WARNING! It is also necessary to specify a stopping criterion of some kind when this command is used. Most thermal-equilibrium calculations stop when the electron temperature falls below some lowest value, set with the **stop temperature** command and with the default value 4000 K. This cannot happen with a constant-temperature model. For instance, a constant temperature model of a planetary nebula will continue until the default limit to the number of zones (now 1400) is reached. The *vast* majority of the cloud will consist of predominantly neutral gas well outside the hydrogen Strömgren sphere. This gas will have a small ambient level of ionization and emission due to collisional ionization. The resulting emission-line spectrum might be surprising since the neutral gas contributes significant emission.

Any of the stopping criteria described in the Chapter *Stopping Criteria* can be used. A more physical constant temperature model of a ionized cloud could be done by using the **stop efrac** command to stop the calculation when the hydrogen ionization front is reached.

11.4 Constant grain temperature 20K [linear]

Normally the temperature of each grain material and size is determined by balancing heating and cooling. This command sets the grain temperature to the indicated quantity.

If the **linear** keyword appears then the number is interpreted as the linear temperature. Otherwise numbers ≤ 10 are interpreted as a log of the temperature.

Other aspects of the grain physics are controlled with the **grain** command.

11.5 Coronal equilibrium, T=1e7 K [linear]

Coronal equilibrium, in which the gas is collisionally ionized, is assumed. The number is interpreted as the log of the gas kinetic temperature if the argument is ≤ 10 or the keyword **log** is present, and the linear temperature otherwise. The optional keyword **linear** will force the interpretation as a linear temperature. This holds the electron temperature at the specified value.

It is necessary[1] to also specify some sort of stopping criteria. The calculation will probably continue until the default limit to the number of zones is reached if a stopping criterion is not specified. As an example, the following would compute the properties of $1\,\mathrm{cm}^{-3}$ of a low density gas at a million degrees Kelvin:

```
coronal, T=1e6K
hden 0      # H density 1 cm^-3
set dr 0    # set zone thickness of 1 cm
stop zone 1 # do only one zone
```

If this is the only energy source specified then it is considered an intensity command. Emission-line intensities will be predicted, as described on page 13, unless a radius is also specified.

Figure 11.1 shows the soft X-ray line and continuum emission predicted from the input stream in the test case `ism_hot_brems.in`.

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

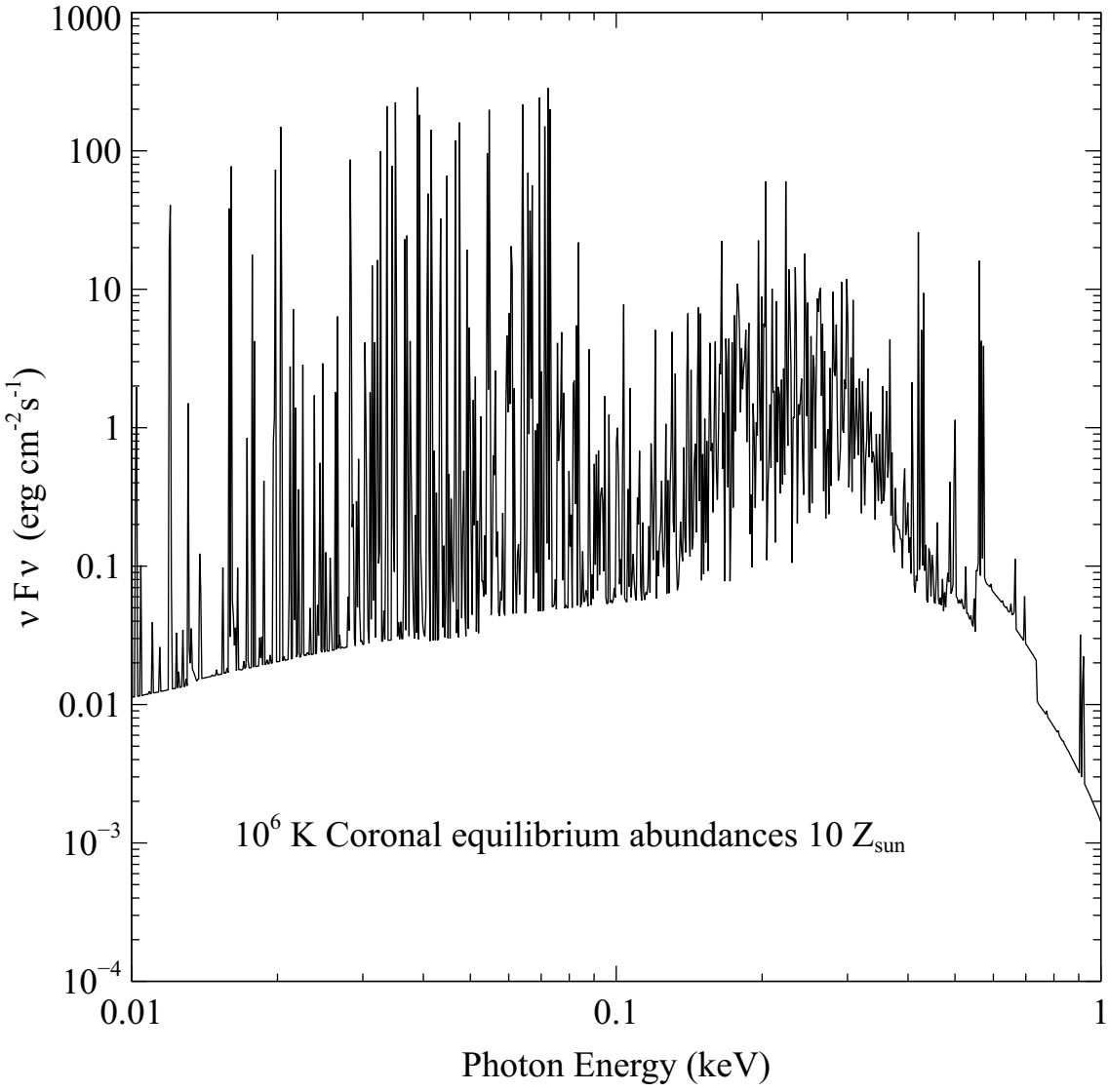

Figure 11.1: This shows the soft X-ray emission from the hot phase of a metal-rich ISM in coronal equilibrium. The input script `ism_hot_brems.in` was used to create the predicted spectrum.

The **time init** option provides a way to set initial conditions for a time-dependent cloud. This is described beginning on page 162 where the **time** command is discussed.

11.6 Cosmic rays [options...]

This adds cosmic rays and their heating and ionization. The command must be specified with at least the first five letters to avoid ambiguity with the **cosmology** command.

This physics is described by Ferland and Mushotzky (1984), the subsection *Cosmic Ray Interactions* in the section *Other Physical Processes* in Part 3 of this document, and section 11.3 of AGN3. Cosmic rays mainly heat ionized gas and both heat and create secondary ionizations in neutral gas. All of this is done self-consistently (Dalgarno et al., 1999).

The **set csupra** command provides a way to specify an H^0 secondary ionization rate. This has many of the same effects as introducing cosmic rays and uses the same code variables. The rate introduced by the **set csupra** command is not self-consistent with the rest of the calculation but does provide a way to test the code in certain simple limits.

Cosmic rays must be included if the calculation extends into molecular regions. The ion-molecule chemistry that occurs in the cold ISM requires a source of ionization (Dyson and Williams, 1997). In fact, most estimates of the galactic cosmic ray background are based on the abundances of chemical ions such as H_3^+ (McCall et al., 2003; Shaw et al., 2008). The chemistry network will probably collapse if the gas becomes molecular but cosmic rays are not present. The code will complain but try to compute the model if a simulation without cosmic rays extends into cold gas. If cosmic rays are not included in the calculation and the neutral hydrogen ionization rate falls below $10^{-17}\,s^{-1}$ the code will print a comment stating that the ionization rate fell below the galactic background rate.

11.6.1 Cosmic rays background [1.2]

This includes galactic background cosmic rays. We adopt the Indriolo et al. (2007) mean H^0 cosmic ray ionization rate of $2\times10^{-16}\,s^{-1}$. The H_2 secondary ionization rate is then $4.6\times10^{-16}\,s^{-1}$,[2]. (Glassgold and Langer (1974) give the relationship between H^0 and H_2 ionization rates.) CLOUDY determines the heating and ionization rates of cosmic rays self consistently, as appropriate for the local molecular, atomic, and electron densities (see AGN3 Section 11.3).

An optional scale factor specifies the cosmic ray ionization rate relative to this background value. The scale factor is assumed to be a log unless the keyword **linear** also appears.

The background rate is an active research area. McCall et al. (2003) find a background rate $1.2\times10^{-15}\,s^{-1}$ for a galactic sight line. Shaw et al. (2008) find a rate about five times larger from detailed modeling of the sight line to Zeta Per. Pellegrini et al. (2007) find a significantly enhanced ionization rate in M17. Suchkov et al. (1993) find cosmic ray ionization rates that are several dex larger than the galactic background in the starburst galaxy M82.

[2]Before 2004 the code used a background ionization rate of $7.4\times10^{-18}\,s^{-1}$ quoted by Tielens and Hollenbach (1985a, Table 10) and McKee (1999). The rate of $2.5\times10^{-17}\,s^{-1}$ given by Williams et al. (1998) was used through version C10.

The abundance of PAHs (see Section 7.9.4) has a direct effect on the electron density and hence on the cosmic ray seconday ionization rate of H. The cosmic ray ionization rate of H_2 is nearly twice the rate for H since two atoms are present. This command changes the atomic hydrogen ionization rate. Shaw and Ferland (2021) suggest that, for the average Galactic PAH abundance, the cosmic ray ionization rate of atomic hydrogen is $3.9\pm 1.9\times10^{-16}$ s^{-1}. This rate can be specified with the command

```
cosmic rays background 1.95 linear
```

Shaw and Ferland (2021) show that the hydrogen cosmic ray ionization rate derived using H_3^+ is much higher, $47.8\pm 23.3\times10^{-16}$ s^{-1}, if PAHs are not present.

11.6.2 Cosmic ray rate -17.3

If the keyword **rate** appears then the command specifies the log of the H^0 ionization rate [s^{-1}] in predominantly neutral gas. The default is $2.5\times10^{-17}s^{-1}$ taken from Williams et al. (1998).

11.6.3 Cosmic rays density =1.2 [index, etc.]

If the keyword **density** appears then the first number is the log of the cosmic ray density $n(cr)$ [cm^{-3}]. Expressions given in Ferland and Mushotzky (1984) convert this into an ionization rate.

The second optional number is a power-law index α that describes the variation of the cosmic ray density with radius, i.e.,

$$n(cr,\ r) = n(cr,\ r_o)\left(\frac{r}{r_o}\right)^{\alpha}\ [\mathrm{cm}^{-3}]. \tag{11.2}$$

The default value of the index is $\alpha = 0$, corresponding to a constant cosmic-ray density. The third optional number is the log of the temperature of the fast electrons, if they are not relativistic. If this third number is specified then expressions from Balbus and McKee (1982) will be used to evaluate the electron heating rates. The options can be omitted from right to left.

11.6.4 Cosmic ray equipartition

The cosmic rays are assumed to be in equipartition with the magnetic field. Then the cosmic-ray density is given by

$$n(cr,\ B) = n(cr,\ B_0)\left(\frac{U(B)}{U(B_0)}\right)\ [\mathrm{cm}^{-3}] \tag{11.3}$$

where $U(B)$ and $U(B_0)$ are the energy densities of the local magnetic field relative to the galactic background magnetic field. The **magnetic field** command sets B. Webber (1998) gives a cosmic-ray energy density of 1.8 eV cm^{-3}, which corresponds to a magnetic field of 8.5 μG if equipartition holds. The background cosmic-ray ionization rate is scaled by the ratio of the magnetic energy densities to find a local cosmic ray ionization rate and energy density.

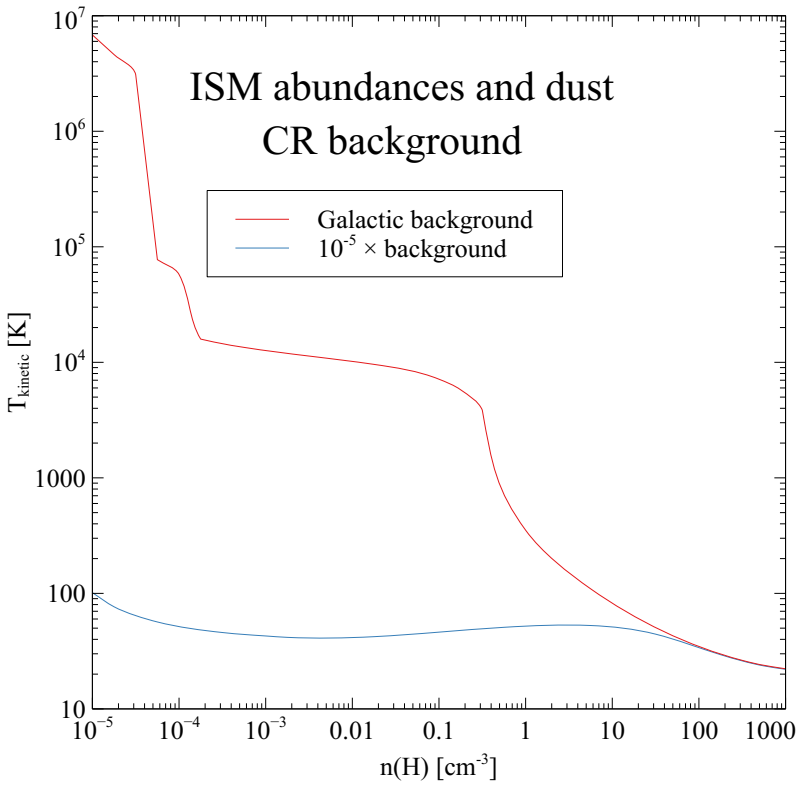

Figure 11.2: This shows the gas kinetic temperature as a function of hydrogen density for a parcel of ISM gas and dust exposed to ISM background starlight and cosmic rays. The upper curve includes cosmic rays with the galactic background density while the lower curve has a density 10^5 times smaller. The comsic rays strongly affect the conditions in the gas when the hydrogen density is below $\sim 30\,\mathrm{cm}^{-3}$.

11.6.5 The CR background and low-density gas

The photoelectric heating and radiative cooling of low-density gas varies as the square of the density while the CR heating varies linearly with density. This means that CR effects become increasingly important, indeed dominant, at low enough densities. Figure 11.2 shows this.

The figure shows calculations in which both the galactic starlight and cosmic backgrounds are included. One set of calculations has the full cosmic ray background density while the second has 10^{-5} of that value. The two nearly agree at high densities where CRs have little effect. The effects of the CRs become increasingly dominant at lower densities, causing the gas to reach temperatures of nearly 10^7 K at the lowest hydrogen column density shown.

Something to think about before adding cosmic rays to simulations of low-density gas.

11.7 Failures 100 times [map]

A converge failure occurs when the heating-cooling balance, the electron density, or the pressure, is not within the tolerance set by the **set convergence** commands. Normally CLOUDY will punt[3] after an excessive number of convergence failures (presently 20) occur. This command increases the number of allowed failures to the value entered as a parameter.

When CLOUDY stops because of excessive failures and the **map** option is specified[4] as a keyword on the **failures** command it first produces a map of (heating-cooling) versus temperature to give an indication of where the equilibrium temperature should have been. A section in Part 2 describes thermal failures in more detail and describes this output.

Failures most often occur near a thermal front. This occurs when the solution jumps over peaks in the cooling function that occur near $\sim 2000\,\mathrm{K}$ and $\sim 10^5\,\mathrm{K}$. A warning will be issued at the end of the calculation if the global thermal balance is wrong.

It should not be necessary to use this command. Please post the example on the discussion board if you find a simulation where you need to increase the number of convergence failures.

11.8 Force temperature to 3400 K

This forces the initial estimate of the temperature of the first zone to the value entered. The temperature is interpreted as a log if it is ≤ 10 and the linear temperature otherwise. The keywords **log** and **linear** will override this.

This command is useful if more than one initial temperature solution is possible. It forces the first guess of the temperature to the specified value but *does not* hold the temperature constant. The temperature is determined by energy balance thereafter. (A constant kinetic temperature is set with the **constant temperature** command.)

CLOUDY may have trouble finding a valid first temperature if the initial solution is forced well away from the equilibrium value. This is an inevitable consequence of the complete linearization methods that are intrinsic to the code's solvers. If a large number of thermal failures or warnings result from the use of this command then it is likely that the code has been forced too far away from the solution. A better guess of the initial temperature would fix this problem.

11.9 Hextra -14 [depth, density, time, SS]

This includes extra heating due to some unspecified energy source. The first number H_0 is the log of the volume-heating rate [erg $\mathrm{cm}^{-3}\mathrm{s}^{-1}$]. Optional keywords specify how this heating depends on density, depth into the cloud, or time. If these optional keywords do not occur then the heating is constant.

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

[4] In versions 94 and before, the default was to produce the map, and the **no map** option turned this off. With version 95 the option is not to produce a map, and this must be requested with the **map** option.

11.9.1 Hextra -14 depth 18, thickness 12

The keyword **depth** makes the heating vary with depth into the cloud. The second number is the log of the scale radius r_{scale} [cm]. The extra heating rate varies as[5]

$$H = H_0[\exp(-depth/r_{scale}) + \exp(T - depth)/r_{scale})][\mathrm{erg\,cm^{-3}\,s^{-1}}]. \quad (11.4)$$

The third optional parameter T is the total thickness of the slab. If this is given then the second exponential term will be added. This will mimic an external heat source that warms the cloud from both the illuminated and shielded faces. If the third parameter is not entered then the rightmost term is not included.

11.9.2 Hextra -13 density

The **density** option makes the heating scale with the hydrogen density. The heating will be given by

$$H = H_0\,[n\,(\mathrm{H})\,/n_0]\,[\mathrm{erg\,cm^{-3}\,s^{-1}}]. \quad (11.5)$$

where n(H) is the local hydrogen density [$\mathrm{cm^{-3}}$] and n_0 is a scale density. The scale density is unity by default and is changed by specifying the optional second number, the log of the scale density.

11.9.3 Hextra time

If the **time** keyword appears then the variable scale factors that appear on the **time** command will multiply the heating rate specified with the **hextra** command. This makes it possible to follow the effects of time-dependent heating sources.

11.9.4 Hextra SS 0.2 1e8 3e17

When the keyword **SS** appears, the standard α-disk model (Shakura and Sunyaev (1973)) will be used. The first parameter is α, the dimensionless parameter of an α-disk model. The second number is the mass of center black hole M in solar mass units. The last one is the distance [cm] from the center of the disk r. In this model, heating rate is calculated as

$$H = \alpha P\sqrt{\frac{MG}{r^3}}[\mathrm{erg\,cm^{-3}\,s^{-1}}] \quad (11.6)$$

where P is the local gas pressure [erg $\mathrm{cm^{-3}}$] and G is gravitational constant. All three parameters must appear. There are spaces to either side of "**_SS_**".

[5] In versions through 94.00 the heating rate varied as $\exp[-r_{scale}/(r-r_o)]$ and went to infinity as the illuminated face. The radial dependence was changed to its current form in 94.01.

11.9.5 Notes on the hextra command

The heating can only depend on depth or density but the **time** option can also be specified for each.

By default a calculation will stop when the kinetic temperature falls below 4000 K. It will be necessary to set other stopping criteria such as the cloud thickness if the extra heating keeps the gas warmer than this limit.

11.10 High temperature approach

This command tells the code to search for the first temperature by approaching the thermal solution from the high temperature extreme of 10^6 K. Normally the approach is from low temperatures. This can be useful when more than one thermal solution is possible. An example is shown in Figure 10 of Ferland et al. (2009a).

11.11 Magnetic field, log(B) = 5 [options]

Magnetic fields are not included by default. This specifies the strength and geometry of the magnetic field. A number, the log of the magnetic field strength in Gauss, is the first number on the line. The physical effects of magnetic fields are discussed in Part 3 of this document.

Ordered and tangled fields can be specified. The field is assumed to be tangled by default. If the keyword **ordered** appears then the angle between the radiation field of the central object and the magnetic field is also specified. This angle is zero if the field is in the outward radial direction. The angle is given in degrees by default but the **radian** keyword will cause it to be radians instead.

In the case of a tangled field the code will look for a second number, the index for the gamma-law relation between the magnetic field and the gas density. If a second number is not found then an index of $\gamma = 4/3$, appropriate for a tangled field, will be assumed. This index appears in the relationship

$$B_{\rm tangled} = B^0_{\rm tangled} \left(\frac{\rho}{\rho_0} \right)^{\gamma/2} [\rm Gauss] \tag{11.7}$$

where the term in parenthesis is the ratio of the current density to the density at the illuminated face of the cloud. For a spherical constant-velocity flow, where $\rho/\rho_0 \propto (r/r_0)^{-2}$, the field will vary as $B/B_0 \propto (r/r_0)^{-\gamma}$. This is in contrast with a dipolar field, in which the radial dependence is $B/B_0 \propto (r/r_0)^{-3}$.

Both ordered and tangled fields can be specified on separate commands. If more than one ordered or tangled field is specified then the second will take precedence over the first.

The major thermal effect of a field is to add cyclotron cooling. This is only important at very high temperatures.

Magnetic pressure and enthalpy terms corresponding to the magnetic energy density are included in the equation of state when a constant-pressure geometry is assumed. The magnetic pressure is

$$P_B/k \approx \frac{B^2}{8\pi k} \approx B^2_{100\,\mu G}\, 2.9 \times 10^6 [\rm cm^{-3}\,K] \tag{11.8}$$

where $B_{100\,\mu G}$ is the magnetic field in units of $100\,\mu G$. The plasma β parameter, defined as the local ratio of thermal to magnetic energy densities, $\beta = P_{gas}/P_{mag}$, is printed in the summary of pressure contributions. The field affects the dynamics of the gas when $\beta \leq 1$.

Supersonic turbulence in rough energy equipartition with the magnetic field is often observed in the cold ISM (Heiles and Crutcher, 2005). The **turbulence** command has an **equipartition** option to set the turbulent velocity to a value in energy equipartition with the magnetic field. Although a general correlation is observed in the cold neutral medium of the ISM, it is not thought to be a fundamental relationship (see the Heiles & Crutcher paper).

11.12 Map, zone 4 [range 2000, 5000]

This computes the heating-cooling vs temperature relation (a "thermal map") of the zone specified as the first number on the line. This is one way to check for the existence of more than one thermal solution. If no zone is specified, or if the zone is less than or equal to 0, then only a thermal map is produced for the illuminated face of the cloud and the calculation then stops. The calculation of the heating and cooling is self-consistent across this map. A section in ***Problems*** in Part 2 of this document explains how to interpret the map output.

The map produced by this command is not directly comparable to the more typical plot that shows the equilibrium temperature as a function of ionization parameter (Krolik et al., 1981). That map can be produced by successively calling CLOUDY with the same radiation field but different densities, say, with the **grid** command. In that second case each deduced temperature is a valid equilibrium temperature. In the map produced by the **map** command only one temperature, where heating and cooling are equal, is a valid equilibrium temperature. The map produced by this command is useful for checking for more than one thermal solution, to check that the heating and cooling curves smoothly flow as the temperature changes, or to investigate why the code had convergence problems (it was originally introduced for only this latter purpose).

The optional keyword **range** specifies the temperature range of the map. If this option is specified then the first number on the line must be the zone for the map, zero if only a map of the illuminated face, and the next two numbers must be the lower and upper temperature limits to the map. These temperatures will be interpreted as logs if the first temperature is ≤ 10. Normally about 20 steps occur between the lowest and highest temperature in the map. The number of steps can be reset with the **set nmaps** command.

The thermal map can be saved as a separate file with the **save map** command. This produces output that is suitable for processing by other software.

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

11.13 Neutrons -2 [efficiency =-2]

This adds energy deposition and ionization by secondaries due to the fast neutrons proposed by Sikora et al. (1989). The argument is the luminosity in fast neutrons expressed as a fraction of the *total* photon luminosity of the incident continuum. It is interpreted as a log if ≤ 0 and a linear scale factor if positive.

The second optional argument is the heating—ionization efficiency of the neutrons, unity by default. Both quantities are interpreted as logs if ≤ 0 and linear if positive.

11.14 Print coolants, zone 135

See the **print ...** commands below.

11.15 Print heating

See the **print ...** commands below.

11.16 Set temperature [floor, convergence]

See the **set temperature ...** commands below.

11.17 tlaw [DB96, SN99]

This specifies a temperature law. Two options currently are implemented.

The **DB96** option tells the code to use the temperature—column density law used by Draine and Bertoldi (1996):

$$T = T_0 / \left[1 + N(\mathrm{H})\, \sigma_{d,1000}\right] [\mathrm{K}] \tag{11.9}$$

where T_0 is 500 K and σ is $6 \times 10^{-22}\,\mathrm{cm}^{-2}$.

The **SN99** option tells the code to use the temperature - H_2 fraction relationship assumed by Sternberg and Neufeld (1999);

$$T = \frac{500}{1 + 9\left[2n\left(\mathrm{H}_2\right)/n\left(\mathrm{H}_{tot}\right)\right]^4} [\mathrm{K}]. \tag{11.10}$$

11.17.1 Tlaw table [depth, radius]

If the keyword **table** appears on the **tlaw** command then the code will read in a set of ordered pairs of radii and temperatures. There must be two numbers per line as in the example below. The first number is the log of the radius or depth [cm] and is followed by the log of the temperature [K]. If the keyword **depth** also appears on the command line then the first number is interpreted as the log of the depth from the illuminated face and the table must begin with a depth smaller than 10^{-30} cm, the first point where the depth is evaluated. The first number is interpreted as the log of the radius if **depth** does not appear. The ordered pairs end with a line with the keyword **end** in columns 1 through 3.

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

```
tlaw table depth
continue -35 4
continue 12 4
```

```
continue 13 5
continue 14 6
continue 15 7
end of tlaw
```

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

Chapter 12

CONTROLLING ATOMIC AND MOLECULAR MODELS

12.1 Species overview

CLOUDY includes models of the internal structure of atoms, ions, and molecules, which are used to predict line spectra. We collectively refer to these models as "species", as described on page 14 above. The **database** and **species** commands, described here, change details of the physical treatment of these models.[1] The **database** commands change properties specific to different atomic and molecular databases, while the **species** commands change properties specific to an individual atomic or molecular species.

The commands described in this section allow many details concerning the treatment of a species to be changed. For instance, it is possible to change which atomic or molecular database will be used to model a species and the number of levels to include.

12.1.1 How many levels do we include?

Some models can include many hundreds to thousands of levels. The strongest lines tend to come from lower levels, although high levels can be quite important at high densities. Very large models, with the greatest number of levels, give the best spectroscopic accuracy but can take quite some time to compute. By default we include an intermediate number of levels, chosen as a compromise between execution time and an adequate model of the emission and cooling. The default number of resolved and collapsed levels can be found in LineLabels.out in the CLOUDY docs directory.

Lykins et al. (2013) describe the logic behind our selection of levels. CLOUDY can do models in either photo or collisional ionization. For a particular ion the photoionization case will generally be cooler than the collisional case. For most elements the default limits are 50 levels for collisional and 15 for photoionization models. For iron, which tends to have many more levels due to the number of orbiting electrons, we settled on a default limit of 100 levels for the collisional case and 25 levels for the photoionization case.

The maximum number of levels for various species can be adjusted with the commands

[1] The **database** command was the **atom** series of commands in C13 and before. The code currently accepts **atom** as a pseudonym for **database**.

described in this section. The actual number of levels computed at a point in the cloud is determined by the local physical conditions since high levels may not be populated if, for instance, the temperature is low.

12.1.2 Species output options

Database print

This generates a report summarizing properties of all species. The species label, the number of levels used, and the database which defined it, are all included. To see which CLOUDY uses by default run this simple test:

```
test
database print
```

The commands described in this section allow changes to the detailed treatment of each species.

Save species

This family of commands, described on 234, reports column densities, densities, and other properties of the species.

Save lines labels

In models obtained from external databases, the levels connected by photon transitions are listed in the output of **save lines labels**, described on 221.

12.1.3 The g-bar approximation

Many transitions do not have collision rates (collision rates are for more difficult to determine than radiative rates). We use various forms of the "g-bar" approximation, as originally suggested by Seaton (1962) and van Regemorter (1962), to fill in missing data.

The **set gbar** (see page 298) controls aspects of the g-bar approximation.

12.1.4 Generating atomic data bibliography

The `scripts/` directory contains tools to generate a summary of the references to the original atomic data used by the code. `db-ref-bib2json.pl` crawls through the database, gathers the relevant citations, and saves them in a suitable JSON file in the database directory (e.g., `data/stout/refs.json`). This file ships with Cloudy, but it may need to be updated. These citations can be converted to a LaTeX table with `db-ref-json2tex.pl`. Currently, only the Stout database is fully supported with these tools.

`db-species-tex.pl` may be used to produce a LaTeX table that lists the database from which the atomic and molecular data of a particular run are drawn. The script operates on the output of the **save species labels** command, which must be issued in that run.

Please consult the comments at the top of each script for usage instructions and functional details.

12.2 Species "name" [options]

The **species** command provides various options for controlling behaviour of individual species (atoms, ions or molecules). The options are discussed in the following sub-sections. The species names should be provided in chemical, not spectroscopic, notation, i.e. `"CO"` or `"O+"`, *not* `O 2` or `"O 2"` – the species names are case-sensitive.

The parsing of the **species** is stricter than other commands in CLOUDY, to allow multiple options to be specified at a time. The options must be spelt out completely (although they remain case-insensitive).

12.2.1 Species "name" off

This option allows individual *molecular* species to be disabled (i.e. not atoms or ions at present). This option is useful for debugging, and investigating the importance of specific species to chemical pathways. It should be used with care, as disabling molecules compromises the physical validity of the model, and may in some circumstances lead to numerical instability.

12.2.2 Species "name" levels=[10,all]

This option allows the number of levels used in modelling the species to be altered from the default value, within the bounds of the transition rate data available to CLOUDY. The command

```
species "O+" levels=10
```

runs a model with 10 levels for the O^+ ion, rather than the default value.

Using **=all** rather than a numeric argument requests the maximum available number of levels. The equal sign is part of the keyword and must be specified with no space between it and **all**.

12.2.3 Species "name" dataset="abc"

This option allows you to use an alternate dataset (designated by a nickname) of atomic or molecular data for a given species. It is only supported for the Stout database as other databases are developed by third parties and we have a policy of not altering those databases. The data files belonging to the alternate dataset are stored in the same directory as the default Stout data files, but will have a modified filename. For example, if you have a dataset with the nickname `abc` for species S+2, the full path to the files will be `data/stout/s/s_3/s_3_abc.nrg`, etc.

Note that the nicknames are case-sensitive as they are part of a filename. This is why they need to be enclosed in double quotes. See HAZY 2 for further details on the datasets that are supported by CLOUDY.

12.2.4 Species "name" lte

This option specifies LTE relative populations will be used for the species, rather than detailed level balance equations.

12.3 Database H2 options

The large model of the H_2 molecule, described in Shaw et al. (2005), will be included if any of the family of **database H2** commands appears. The much faster three-level model outlined by Tielens and Hollenbach (1985a), Burton et al. (1990), Draine and Bertoldi (1996), and expanded by Elwert et al. (2005), is used by default. The large molecule is represented by several thousand levels producing roughly half a million lines. AGN3 describes some properties of H_2 in section 8.3 and appendix A6.

NB The number "2" appears in the keyword for this command. Any numerical parameters that appear on the **database H2** commands must appear after this two—the code will check that the first number parsed off the command line is the number 2.

12.3.1 The set H2 command

Some details of the physical treatment of the H_2 molecule can be changed with the **set H2** command described on page 299 below.

12.3.2 H_2 output options

Strong H_2 lines will appear in the main emission-line output. The **save H2** command has many other output options. The section in Part 2 of this document describing calling the code as a subroutine describes several routines that will return predictions of the large molecule. The **print line precision** command will add more significant figures to the wavelengths printed in the main printout. This may help isolate a particular line.

12.3.3 Database H2

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

12.3.4 Database H2 chemistry [simple; full]

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

12.3.5 Database H2 collisions [options]

These commands change various collisional processes within the H_2 molecule.

database H2 ortho para collisions on/off

This turns off ortho-para changing collisions with gas particles.

database H2 orH2 collisions options

database H2 paH2 collisions options

These commands determine which of the H_2 – H_2 collision data sets is used. The default is the Lee et al. (2008) set, which can also be chosen with the **ORNL** option. The keyword **Le Bourlot**

selects the Le Bourlot et al. (1999) data set. The keyword **ohH2** adjusts the ortho data and the keyword **paH2** adjusts the para data set. One of these keywords must be specified. Both cannot be adjusted with the same command.

database H2 H collisions (year)

This command sets collision rates for H_2 – H^0 collisions. There are three data sets. The default is **2007**, which uses the rates given by Wrathmall et al. (2007). The **1999** option uses the rates given by Le Bourlot et al. (1999). The **2015** option uses the rates given by Lique (2015). By default we use the Wrathmall et al. (2007) data.

database H2 He collisions options

This sets collision rates for H_2 - He^0 collisions. There are two data sets. The default is **ORNL**, which uses the rates given by Lee et al. (2006). The **Le Bourlot** option uses the rates given by Le Bourlot et al. (1999).

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

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

By default the code will uses guesses of collisional rate coefficients using the g-bar method. Collisional deexcitation for the g-bar transitions are turned off with the **database H2 gbar** command.

12.3.6 Database H2 gbar [off; on]

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

12.3.7 Database H2 levels

This changes the number of electronic levels within the H_2 molecule. The default is seven and includes the ground and first six bound singlet electronic states. This is also the largest number of levels. At minimum is three levels, which is sufficient to include the Lyman and Werner bands in the UV. This is the necessary minimum number of levels to include the correct photodissociation processes. If no number appears but the keyword **large** does then the code will use the upper limit.

12.3.8 Database H2 limit -4

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

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

12.3.9 Database H2 matrix 43

Populations of the lower ro-vibration states of the ground electronic level are determined by two schemes. The first, and most straightforward, is the solution of a complete set of balance equations by solving the system of balance equations with a master equation approach. The time needed to solve the linear algebra increases as a power of the number of levels so we need to keep the number of levels as small as possible. High levels are treated by back substitution, starting from the highest level with X and proceeding downwards. This command sets the total number of levels that are computed within the matrix. Levels higher than this will be treated with back substitution.

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

12.3.10 Database H2 noise [mean, standard deviation]

This multiplies the rates for collisional processes within the H_2 molecule by a Gaussian random number so that $r' = r\ 10^{\rm rand}$. Here r is the correct rate coefficient and ***rand*** is a Gaussian distributed random number. There are two optional numbers on the command line used for generating the Gaussian random numbers. The first optional number is the mean, with a default of 0. The second optional number is the standard deviation with a default of 0.5.

12.3.11 Database H2 thermal [simple; full]

This changes how the heating and cooling by the H_2 molecule are treated. By default, or if the keyword **full** appears, then the fully self-consistent heating and cooling rates are used when the large H_2 molecule is enabled. If the keyword **simple** occurs then expressions from Tielens and Hollenbach (1985a) are used instead.

12.3.12 Database H2 trace [options]

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

12.3.13 database H2 output options

Roughly half a million lines are predicted when the large H_2 molecule is included. The main emission-line printout includes all significant lines produced in the ground electronic state but does not include electronic transitions. The **print line H2 electronic** command will include these lines. The family of **save H2** commands provides ways to save information such as column

densities, the emission-line spectrum, and details of the effects of H_2 on the conditions in the cloud.

12.4 Database H-like | He-like [element, options]

12.4.1 Introduction

These commands are used to change some details in the treatment of atoms of the H-like and He-like isoelectronic sequences.[2] Atoms of the H-like sequence have one bound electron and include H^0, He^+, Li^{+2}, through Zn^{+29}. Atoms of the He-like sequence have two bound electrons and include He^0, Li^+, through Zn^{+28}.

This implementation of the H-like sequence was initially part of Jason Ferguson's PhD thesis and is described in Ferguson and Ferland (1997), Ferguson et al. (2001), and Bottorff et al. (2002). The physics was expanded to resolve *nl* terms by Ryan Porter during a visit to IoA Cambridge in Fall 2008, as was summarized in Luridiana et al. (2009). Any number of levels up to *n* of 400 can be computed.

The He-like sequence was developed by Ryan Porter as part of his thesis and it is described in Bauman et al. (2005), Porter et al. (2005), Porter and Ferland (2007), and Porter et al. (2012). Ryan Porter unified the two sequences and expanded the H-like sequence to include all of the physics included in the He-like sequence.

The two sequences are now unified so the same **database AA-like** commands largely work for both.

The name of one of the iso-sequences must appear on the command line. The options are **H-like** and **He-like**. Only one iso-sequence can be modified with a single command. By default all ions of the iso-sequence are modified. If the keyword **element** and the name of an element appears then only that ion is modified. The following are some examples

```
# only change hydrogen itself
database H-like element hydrogen resolved levels 8
# change the entire He-like sequence
database He-like resolved levels 9
```

12.4.2 Structure of the model atoms

The model atoms include a certain number of resolved and collapsed levels as shown in Figure 12.1. The lower $n_{resolved}$ levels are *nl* resolved. Another $n_{collapsed}$ levels, which replace the *nl* terms with an *n* configuration, are above the resolved levels. The default number of resolved and collapsed levels can be found in LineLabels.out in the docs directory.

The physics of the resolved levels is exact while that of the collapsed levels is more approximate since the entire *n* configuration is treated as a single level assuming that the *nl* terms within the *n* configuration are populated according to statistical weight. This is a good assumption at high densities, as shown by Pengelly and Seaton (1964).

The total recombination coefficient is the sum to over all possible bound levels. The model has a finite, often small, number of levels. Recombination coefficients to each level are derived from

[2] This was the **hydrogenic** command in versions 90 and before.

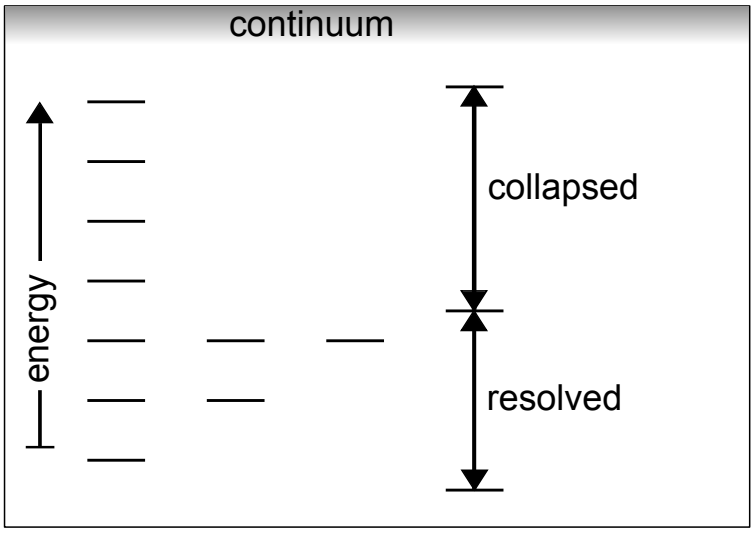

Figure 12.1: This shows resolved and collapsed levels for a typical H- or He-like species. The set of horizontal bars on the left are energy levels with principal quantum numbers ranging from 1 to 7, increasing from bottom to top. The lowest three configurations are resolved into *nl* terms while higher shells are treated as collapsed levels, in which the *nl* terms are assumed to be populated according to their statistical weight. The levels are adjusted with the **database AA-like levels** command. This is a cartoon and does not represent any particular species.

the photoionization cross section using the Milne relation (AGN3). The recombination to these levels is less than the total. We "topoff" the model by adding the remaining recombination coefficient to the model atom, as controlled with the **database xx-like topoff** command.

By default, the lines produced by the highest levels are not included in the printout since their intensities may be unreliable. The **print line iso collapsed** command, described in 16.19.11, will cause lines produced by the highest levels to be included in the printout. See the discussion of the **print line iso collapsed** command for more details.

The H-like sequence

Figure 12.1 shows an H-like system. Resolved configurations are split into *nl* levels. Collapsed levels each represent one principal quantum number. By default there are 10 resolved and 15 collapsed levels for H^0 and He^+, and 5 resolved and 2 collapsed levels for heavier elements.

The H-like iso sequence models atoms can extend up to any principle quantum number between the limits $4 < n \leq 400$. The size is limited mainly by the available memory and computer time. Increasing the number of levels allows a better representation of the collision physics that occurs within higher levels of the atom at the expense of longer execution times and greater memory requirements.

The He-like sequence

The model atom resolves *n*-levels into *nLS* levels, and the $2\ ^3P$ term is split into three *nLSJ* levels. For a given maximum principal quantum number $n_{\rm max}$, there will be a total of $n_{\rm max}^2 + n_{\rm max} + 1 nLS$

levels. The default number of resolved levels for He^0 is 6, resulting in 43 *nLS* levels, there are five resolved levels for C, N, O, Fe, and Zn, and three resolved levels for all remaining elements. Increasing $n_{\max}$ allows a better representation of the atom's emission, especially the collision physics that occurs within higher levels of the atom, but at the expense of longer execution times and greater memory requirements.

There are also 20 collapsed levels for He^0 and one collapsed level for heavier elements. These collapsed levels bring together all of the individual *nLS* terms as one pseudo-level. The recombination coefficient into this pseudo-level is the sum of recombination coefficients into the individual terms plus the recombination topoff. Transition probabilities from this pseudo-level are calculated as follows.

$$A(n_u \rightarrow n_l L_l S) = \frac{\sum_{L_u=L_p m1} g_{L_u S} A(n_u L_u S \rightarrow n_l L_l S)}{\sum_{L_u=L_l \pm 1}}. \tag{12.1}$$

This causes the collapsed level to behave exactly as if it were a set of resolved terms populated according to statistical weight.

There is no limit to the number of levels other than computational resources like memory and time.

Selecting collisional processes

database H-like collisions thermal This option will do the more precise averaging for all densities, will be far slower, and should be used when precision is needed.

database H-like collisional ionization off This command turns off collisional ionization by thermal electrons of all levels. Ionization by cosmic rays is not affected by this command.

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

database H-like collisions off This command turns off collisions for all elements along the H-like isoelectronic sequence. All three collisional processes will be turned off if none of the three keywords are recognized.

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

12.4.3 Database H-like collisions...

Collisional processes, both between levels and ionization, are controlled with this command. This command is mainly used for testing. Separate collisional processes can be modified with the

following options. Only one option is recognized per command line so multiple commands are needed to turn off several processes.

Selecting collisional processes

database H-like collisions thermal This option will do the more precise averaging for all densities for n-changing and l-changing collisional data coming from cross sections. It will be far slower, and should be used when reference calculations with high precision is needed. The default is to use the effective coefficients at KT energy instead of Maxwell averages.

database H-like collisional ionization off This command turns off collisional ionization by thermal electrons of all levels. Ionization by cosmic rays is not affected by this command.

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

database H-like collisions off This command turns off collisions for all elements along the H-like isoelectronic sequence. All three collisional processes will be turned off if none of the three keywords are recognized.

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

Selecting n-changing collision theories

Guzmán et al. (2019) has compared different hight Rydberg collisional excitation theories. The commands below select between them.

database H-like collisions Lebedev The semiclassical straight-trajectory Born approximation from Lebedev and Beigman (1998) is used by default.

database H-like collisions Vriens This turns off the default Lebedev and Beigman (1998) and turns on Vriens and Smeets (1980) for n-changing collisions.

database H-like collisions Fujimoto This turns off the default Lebedev and Beigman (1998) and turns on Fujimoto (1978) for n-changing collisions.

database H-like collisions van regemorter This turns off the default Lebedev and Beigman (1998) and turns on van Regemorter (1962) for n-changing collisions.

Selecting l-changing collision theories

This selects the l-changing collision theory. As described in Badnell et al. (2021) we recommend our default option, the modified Pengelly and Seaton (1964), PSM20.

database H-like collisions l-mixing off This turns off all l-mixing collisions, including $2s-2p$.

database H-like collisions l-mixing Pengelly PSM20 The modified Pengelly and Seaton (1964) formalism, PS-M, described by (Badnell et al., 2021), is used. This is the default option when no command is included.

database H-like collisions l-mixing Pengelly [Classic/PSM17] If the key **Classic** is used, then the original Pengelly and Seaton (1964) is used. If the key **PSM17** is used, then the slightly different modified theory from (Guzmán et al., 2017) is used.

The **Case B** command, described in Section 10.3, uses PSM20 by default. That command has a key **Hummer** to allow comparison with Hummer and Storey (1987), that switches to the original Pengelly and Seaton (1964) formalism. This formalism is providing Maxwell averaged rate coefficients so **thermal** option does not apply.

database H-like collisions [thermal] l-mixing Vrinceanu This changes to the semi-classical formalism of Vrinceanu and Flannery (2001), which reports cross sections. By default, the rate is derived from the cross section at the most likely collision energy. Here the density is so low that l-changing collisions are unimportant, or so high that the levels are l-mixed. See the **database H-like collisions [thermal]** command above for a discussion of the thermal averaging options.

database H-like collisions l-mixing VOS12 Semiclassical This changes to the approximation shown in equation (9) of Vrinceanu et al. (2012). This formalism is providing Maxwell averaged rate coefficients so **thermal** option does not apply.

database H-like collisions [thermal] l-mixing [thermal] VOS12 Quantal This changes to the approximation shown in equation (2) of Vrinceanu et al. (2012), corresponding to a pure quantal treatment of l-mixing collisions. See the **database H-like collisions [thermal]** command above for a discussion of the thermal averaging options.

12.4.4 Database He-like collisions ....

This command controls He-like collsional processes. Separate collisional processes can be modified with these options. Only one option is recognized per command line so multiple commands are needed to turn off several processes. This command modifies collisions for all elements along the He-like isoelectronic sequence.

Selecting collisional processes

As in section 12.4.3:

database He-like collisions thermal
database He-like collisional excitation off
database He-like collisional ionization off

Selecting n-changing collision theories

The same commands used for H-like explained in section 12.4.3 are used here.

database He-like collisions Lebedev The semiclassical straight-trajectory Born approximation from Lebedev and Beigman (1998) is used by default.

database He-like collisions Vriens This turns off the default Lebedev and Beigman (1998) and turns on Vriens and Smeets (1980) for n-changing collisions.

database He-like collisions Fujimoto This turns off the default Lebedev and Beigman (1998) and turns on Fujimoto (1978) for n-changing collisions.

database He-like collisions van regemorter This turns off the default Lebedev and Beigman (1998) and turns on van Regemorter (1962) for n-changing collisions.

Selecting l-changing collision theories

The same commands used for H-like explained in section 12.4.3 are **only used for** $l > 2$. For $l \leq 2$ l-shells are highly non degenerate and cannot be adequately treated by l-changing theories (see Guzmán et al. (2017)).

database He-like collisions l-mixing [options] Where as in section 12.4.3 **options** are **off**, **Pengelly [Classic]**, **[thermal] Vrinceanu** , **VOS2012 Semiclassical** or **[thermal] VOS2012 Quantal**

database He-like collisions l-mixing all degenerate [options] This disables non degenerate calculations and all $l - l'$ (also $l \leq 2$) collisions are calculated with the model chosen in **options**, assuming all the levels are degenerate. This is mainly a testing command. The **options** are either **Pengelly**, **Vrinceanu** or **VOS12**.

database He-like collisions l-mixing PSM20 Standard calculations from (Badnell et al., 2021), with the criteria used in Guzmán et al. (2017), will be performed. This option will be performed as default if no command is included.

database He-like collisions l-mixing [B72—no degeneration] [options] This sets up the criterion for degeneration on collisions. If **B72** is used, criteria from equation 3.12 of Brocklehurst (1972) is adopted. If **no degeneration**, Pengelly and Seaton Pengelly and Seaton (1964) smoothed criteria is followed. The **options** are either **Pengelly**, **Vrinceanu**, **VOS12**, or **PSM17** (see above).

database He-like collisions l-mixing S62 [off] [degeneration ops] [options] This enables/disables calculations based on Seaton (1962) for $l \leq 2$ where electrons dominate. If disabled these values are calculated with the model chosen in the **options**. The **options** are either **Pengelly**, **Vrinceanu** or **VOS12**. Differently to the **all degenerate** command, **this does not disable non-degenerate l-shells**.

12.4.5 Database H-like | He-like continuum lowering off

This command disables continuum lowering processes due to particle packing, Debye shielding, or Stark broadening. The processes are active by default and implemented following Bautista and Kallman (2000). This command applies to the entire isoelectronic sequence.

12.4.6 Database H-like damping off

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

This only disables H I Rayleigh scattering. The **He-like** option is not supported.

12.4.7 Database He-like dielectronic recombination [off]

This option turns dielectronic recombination off for the entire isoelectronic sequence. Dielectronic recombination is included by default. This command only turns off DR. It affects all elements of the He-like iso sequence.

12.4.8 Database H-like | He-like [element] masers off

This command turns off masers from collapsed levels for the specified iso-sequences. Masers are allowed by default for all species. If no element is given masers are turned off for all species of that isosequence.

12.4.9 Database H-like | He-like error generation [pessimistic]

Monte Carlo error analysis can be performed with this command. Randomly generated Gaussian errors are applied to atomic data.

The pessimistic option will choose the largest of two standard deviations for each piece of atomic data. The full command would look something like

```
database He-like error generation pessimistic
```

Without the pessimistic keyword the default optimistic values are used.

12.4.10 Database He-like gbar options

The code employs various forms of the $\bar{g}$ approximation to fill in collision strengths for those transitions with no quantal calculations. This command changes which approximation is used. The options are **Vriens** for the Vriens and Smeets (1980) and **off** to set this to zero. This only exists for the He-like iso sequence.

12.4.11 Database H-like | He-like levels options

This set of commands controls the model atomic structure shown in Figure 12.1.

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

Database H-like | He-like levels resolved | collapsed 4 [element iron]

This sets the number of resolved or collapsed levels. One of the keywords **resolved** or **collapsed** must appear. The argument gives the principal quantum number n for the number of levels. In the case of collapsed levels the number of added levels will be equal to the argument. In the case of resolved levels a large number of resolved terms may be needed to produce the requested number of principal quantum numbers.

The following sets most iso sequence models to a small number of levels but does a high quality simulation of the H- and He- like Fe ions.

```
# most ions have small models
database H-like levels small
database He-like levels small
# resolve the lowest n=10 shells, these take well more than 10 levels
database H-like levels element iron resolved levels 10
database He-like levels element iron resolved levels 10
# 15 collapsed levels on top
database H-like levels element iron collapsed levels 15
database He-like levels element iron collapsed levels 15
```

Database H-like | He-like levels LTE

This will set the level populations to their LTE values.

Database H-like | He-like levels print

This produces a summary of the number of levels for all ions of the iso sequences. It reports the highest principal quantum number of the resolved levels, the number of *nls* terms in these *n* configurations, and the number of collapsed levels.

Setting the number of levels with keywords

If no numbers occur then we look for a keyword to specify preset sizes.

The keyword **large** specifies a preset size of 40 resolved and 160 collapsed levels. The code will make sure that the model atom uses the larger of either the preset size or the current size (i.e., it sets a minimum size). The keyword **small** specifies a preset size of 5 resolved levels (for the H-like sequence) or 3 resolved levels (for the He-like sequence) and 10 collapsed levels. The code will make sure that the model atom uses the smaller of either the preset size or the current size (i.e., it sets a maximum size).

The following would set the full H-like isoelectronic sequence to a small number of levels, then reset hydrogen, helium, and iron to a large number.

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

Actually, the atoms are coded so that the number of levels can be made even larger than discussed above. However, the memory and computer time requirements will be substantial for very large model atoms, so in most cases that will set the upper limit to the size you can model.

Use the **database xx-like levels print** command to confirm that you are using the expected number of levels.

A word of caution. Using a small model atom for He-like species (and especially for helium itself) can have unexpected consequences. Only the levels up to $n = 3$ are resolved, meaning that lines coming from higher levels will be collapsed into one entry with an unusual wavelength. One particular example for a well-known line is the He I $1s4d\ ^3\mathrm{D} \rightarrow 1s2p\ ^3\mathrm{P}^\circ$ triplet line at 4471.49 Å. This line will be absent when using a small model atom since it comes from $n = 4$. But a line at 4469.79 Å will be present, which includes all lines from $n = 4$ to $1s2p\ ^3\mathrm{P}^\circ$. Make sure to study the

output of the **save line labels** command when you are using a small model atom to understand what the entries on the line stack mean.

Warning! Note that the command

```
database H-like levels large
```

will set all 30 hydrogenic atoms to a large number of levels. This would require at least 12 gigabyte of memory for the H-like sequence alone, and would be very slow on today's computers. It is best to only use large model atoms for the most important elements.

12.4.12 Database H-like | He-like Lyman options

This set of commands changes the treatment of the Lyman lines. For this purpose, the Lyman lines are all resonance lines, lines whose lower level is the ground state, of the iso sequences. (Actually, only the H I resonance lines are Lyman lines.)

Database H-like | He-like Lyman extra 1000

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

This changes the number of extra levels for the entire iso-electronic sequence.

Database H-like Lyman extra resolution 0.25

The nP to 1s lines of the H-like isoelectronic sequence are resolved into the fine-structure components. This commands determines down to which atomic species Z and upto which principle quantum number n the nP levels are resolved to nPj=1/2 and nPj=3/2 levels. The default resolution has been set to 1/10 in eV of the Athena high-resolution X-ray mission, 0.25eV.

This command does not affect the model atoms for H I and He II.

Database H-like Lyman pump

These commands control how continuum fluorescent excitation, "line pumping", is treated. They only modify the treatment of H I so the **H-like** keyword must appear.

Database H-like Lyman pumping off

This turns off continuum radiative pumping of the Lyman lines of H I. This is one way to take into account the possible presence of Lyman absorption lines in the incident continuum.

This command is used in the test suite to perform classical PDR simulations which ignore the H^+ region and assume that the PDR is illuminated by a radiation field with no hydrogen-ionizing radiation. Classical PDR calculations do not include a full treatment of H or He so their ions are only produced by cosmic rays. CLOUDY does a full collisional-radiative model of the physics of H^0 and He^0 and will find a thin layer of highly ionized gas. This is produced by Lyman-line pumping into excited np levels, some of which decay and population the metastable $2s$. That level is then photoionized by relatively low-energy light. The process stops when the Lyman line optical depths become large enough for the atom to become self-shielding. This command will stop this process from ever becoming important by disabling Lyman line pumping. (The process would not happen in a realistic calculation which included the H^+ region since Lyman line optical depths through the H^+ region are large.)

This command is included in the "homework problem" PDR simulations in the test suite to stop Lyman line pumping from occurring. A better solution would be to include the PDR as a additional layer outside the H^+ region. The Lyman line optical depths across the ionized gas in the H II region will be large and prevent continuum pumping from occurring in the PDR. Combining the H II region and PDR also results in a self-consistent calculation (Abel et al., 2005 and Abel et al., 2008).

Database H-like Lyman pumping scale 3

The continuum radiative pumping rate of the Lyman lines of H I will be multiplied by the scale factor that appears on the line. This is meant to take into account the possible presence of Lyman lines in emission or absorption in the incident continuum. The pumping rate is normally set by the intensity of the coarse continuum at the line wavelength. The number that appears on the command line is a scale factor that multiplies this continuum.[3] Numbers ≤ 0 are interpreted as a log and the **log** keyword will force interpretation as a log. A value of greater than 1 would correspond to Lyman lines in emission in the incident radiation field. A scale factor of 2 would simulate the presence of stellar Lyman emission lines with peak intensities twice those of the neighboring continuum.

12.4.13 Database H-like | He-like NO RECOmbination INTErp

The code normally derives recombination coefficients by interpolating on a table that lives in the data directory. This command tells the code to compute recombination coefficients on the fly by integrating over the photoionization cross section.

[3] In versions C08 and before a value of 0 would produce black Lyman lines. Starting in C10 values ≤ 0 are interpreted as logs so a value of 0 will result in a scale factor of unity.

12.4.14 Database H-like | He-like redistribution [options]

This changes the form of the redistribution function for various lines within the model atom. This command is only used if you are not happy with the default redistribution functions. The command changes the redistribution functions for entire iso-sequences.

A keyword to specify which set of lines to change is required. This is one of **alpha** (the H-like $2p-1s$ transition or the He-like $2\,^1P-1\,^1S$ transition)), **resonance** (any higher Lyman line decaying to the ground term), or **subordinate** (all Balmer, Paschen, etc lines). The type of redistribution function to use must also be specified. The options are **PRD** (partial redistribution), **CRD** (complete redistribution with Doppler core only), and **CRDW** (complete redistribution with damping wings). (The underscore indicates a space.) However, note that this command has no effect on H Lα line.

The keyword **show** prints the current default redistribution functions at the time when the command is parsed.

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

12.4.15 Database H-like keep fine structure [off]

The default behavior of CLOUDY is to sum all the fine-structure components of a hydrogenic line together and present them as a single (blended) line on the line stack. When this command is used, the individual fine structure components are also pushed onto the line stack. These additional entries will have a special label to distinguish them from the main hydrogenic line, as is shown below.

```
H  1 s+   6562.81A  -16.989    0.3827
H  1 p-   6562.81A  -16.761    0.6472
H  1 d-   6562.81A  -16.372    1.5858
```

The extra characters in the label are the l state of the upper level and a $+$ or $-$ sign indicating whether the l' value for the lower level is either $l+1$ or $l-1$. So, e.g., `H 1 d- 6562.81A` is the $3d \rightarrow 2p$ transition of hydrogen. If $l > 20$ for the upper level, no character designation for the quantum number exists and the l value itself will be printed, e.g., `H 1 22-`. Note that only fine-structure components for resolved to resolved transitions will be saved, and not for collapsed to resolved transitions. If the optional keyword **off** is entered, saving the fine-structure

components will be disabled, which is also the default. This command can only be used for the H-like iso sequence.

12.4.16 Database H-like | He-like topoff [off]

Atoms have a very large number of levels at low densities. This model has a finite number of levels, often far smaller than the number of levels present. We introduce "topoff"; to maintain the correct total recombination coefficient and allow for the proper collisional coupling of the highest levels to the continuum.

This command turns topoff off or on. The default is to include topoff and the keyword **off** disables it. This changes the treatment for the entire iso-sequence.

Topoff is necessary to obtain the correct total radiative recombination rate coefficient with a finite number of levels. Because only a finite number of levels can be computed the sum of the total recombination coefficient will be less than the sum to infinity. This difference must be added somewhere to conserve the total recombination rate. Similarly, the very highest physical levels should be coupled to the continuum by collisional ionization / three-body recombination. When topoff is included the collisional ionization rate for the highest level is increased by a large factor to mimic the effects of much higher levels (three body recombination is obtained from this by detailed balance).

At high densities the continuum is lowered by pressure effects. If the continuum is lowered to include levels in the model atom then topoff is not done. In this case topoff is not needed since we have a complete model of the atom.

12.5 Overview to the Chianti, Lamda, and Stout databases

Three databases, described below, are used by CLOUDY to model line emission from atoms and molecules. Stout is the database we developed for CLOUDY. Chianti (Dere et al. (1997); Landi et al. (2012)) was developed to study outer layers of the sun. LAMDA, the Leiden Atomic and Molecular Database (www.strw.leidenuniv.nl/ moldata/, Schöier et al. (2005)), contains many molecules.

The following sections document how we use these databases. Similar options are available for all three. Our "species", how we specify atoms, ions, and molecules, and their lines, is discussed on 14.

- Each database has a masterlist file, located in **data/DATABASE**, that lists the species to be used with that database.
- If a species is present in more than one database the higher priority database is used. The Stout database has the highest priority, followed by Chianti and LAMDA.
- You can create your own masterlist file to cherry pick the species and database you want with the **database DATABASE masterlist** command.
- In general, we use only a fraction of the levels available in the database. The **print** option on the commands will report the species and number of levels for that database.

- The **levels max** option will use all available levels for a particular database. The calculation will take significantly more time. Lykins et al. (2013) discusses the strategy we use to limit the number of levels.

- It is possible to specify the minimum number of levels for a particular species by entering a number following the species name in the masterlist file for Chianti and Stout.

12.6 Database Lamda [on , off][print][levels {j, MAX}]

The command controls the use of the molecular models from Lamda, the Leiden Atomic and Molecular Database (www.strw.leidenuniv.nl/ moldata/, Schöier et al. (2005)). Note that some molecules in LAMDA are not currently known to CLOUDY's chemistry. CLOUDY therefore has no means by which to predict the abundance of those molecules. In such cases, the LAMDA models are simply ignored.

The **database Chianti** command, described on page 154, and the **database Stout** command, described on page 155, do similar work for the Chianti and Lamda databases.

12.6.1 Database Lamda [on off]

This will enable or disable the use of the Lamda database. It is on by default.

12.6.2 Database Lamda "masterlist.ini"

The species included in the file `filename`, given between quotes, will be used instead of the list in the default location in the `data` directory. The masterlist file must begin with the magic number used in our default masterlist files. By default we use species listed in the `Lamda.ini` file in the `data/lamda/masterlist` directory. This makes it possible to create `ini` files that select portions of Lamda.

12.6.3 Database Lamda print

This prints a report giving which Lamda species are being used along with the number of levels.

12.6.4 Database Lamda Levels n

This sets the maximum number of Lamda energy levels. By default we limit the number of energy levels for Lamda species to 70 . The **levels** option will overwrite this limit. The number, *n*, is the level limit. If no numbers appear but the keyword **MAX** does, we use all of the available energy levels.

The **database Chianti** command, described on page 154, and the **database Stout** command, described on page 155, do similar work for the Chianti and Stout databases.

12.7 Database Chianti [options]

This controls the use of the Chianti database (Dere et al. (1997); Landi et al. (2012)). The **database Lamda** command, described on page 153, and the **database Stout** command, described on page 155, do similar work for the Lamda and Stout databases.

The **save chianti** command described on page 208 can be used to export parts of the Chianti data.

The Chianti database does not contain collision strengths for subordinate transitions. By default, CLOUDY will use the gbar approximation to estimate collision strengths for allowed transitions that do not have data. The **Set Gbar** command controls the use of gbar in Chianti and is described on page 298.

The database contains both experimental and theoretical energies. By default we use only experimental energies, although this can be changed with the **Set Chianti Experimental** command described on page 155.

By default we limit the number of energy levels for Chianti iron species to 25 and all other elements to 15. This can be changed with the **Database Chianti Levels** command described on page 155.

12.7.1 Database Chianti "masterlist.ini"

By default, CLOUDY uses selected Chianti species between phosphorus and zinc, specifically including Fe VI through Fe XXIV. These species are listed in the `CloudyChianti.ini` masterlist file in the `data/chianti/masterlist` directory. Files in that directory give other sets of species.

If a double quote appears on the line then the filename between the pair of double quotes will be used instead of the `CloudyChianti.ini` file. This makes it possible to create masterlist files that select portions of Chianti.

12.7.2 Database Chianti off

will disable all of Chianti.

12.7.3 Database Chianti [hybrid, no hybrid]

Hybrid, the merging of the Chianti and Opacity Project data, is done by default if Chianti is included. The **no hybrid** option disables the Opacity Project data and only uses Chianti. See Lykins et al. (2013) for more details.

Our Opacity Project line set is contained in the file `data/level2.dat` and includes a large number of permitted electronic transitions. These often extend to higher excitation energies and shorter wavelengths than Chianti, so can add important lines. However, the Opacity Project data have greater uncertainties than Chianti. The OP energies are entirely theoretical and there are no matching collision strengths. The line wavelengths have an uncertainty of roughly 10% and we use various forms of the g-bar approximation to produce collision rates.

You can judge the various sources of atomic data by running the model `docs/LineList.in` and examining the resulting file `docs/LineList.txt`. It lists all lines with their sources.

12.7.4 Database Chianti print

will print which Chianti ions are being used along with the number of levels.

12.7.5 Database Chianti Levels

This sets the maximum number of Chianti energy levels. Some ions have a very large number of levels. These have a significant impact on the execution time but produce only weak lines and have a minor effect on the cooling for photoionized clouds. By default we limit the number of energy levels for Chianti iron species to 25 and all other elements to 15. Plasmas in collisional equilibrium tend to be hotter, for a given stage of ionization, than a photoionized plasma. The **Coronal** command, described on page 124, is used for the collisional ionization case. The limits to the number of levels are then 100 for iron and 50 for all others. The **levels** option will overwrite these limits. The first number, i, is the iron level limit, while the second number, j, is for all other Chianti elements. If no numbers appear but the keyword **MAX** does, we use all of the available energy levels. See Lykins et al. (2013) for more details.

12.7.6 Database Chianti [experimental theoretical]

By default we use experimental Chianti energies. The **theoretical** option causes CLOUDY to only use theoretical energies. This provides significantly more levels and transitions, however the error in wavelength is larger.

12.8 Database Stout [on , off, no hybrid][print][levels {j, MAX}]

The command controls the use of the Stout database. The **database Chianti** command, described on page 154, and the **database Lamda** command, described on page 153, do similar work for the Chianti and Lamda databases.

12.8.1 Database Stout [on off]

command will enable or disable the use of the Stout database. It is on by default.

12.8.2 Database Stout "masterlist.ini"

The species included in the file `filename`, given between quotes, will be used instead of the list in the default location in the `data` directory. The masterlist file must begin with the magic number used in our default masterlist files. By default we use species listed in the `Stout.ini` file in the `data/stout/masterlist` directory. This makes it possible to create `ini` files that select portions of Stout.

12.8.3 Database Stout hybrid

Hybrid, the merging of the Stout data with the Opacity Project data, is on by default. The **no hybrid** option disables the Opacity Project data and only uses Stout.

12.8.4 Database Stout print

will print which Stout species are being used along with the number of levels.

12.8.5 Database Stout Levels nFe, nOther

This sets the maximum number of Stout energy levels. Some ions have a very large number of levels. These have a significant impact on the execution time but produce only weak lines and have a minor effect on the cooling for photoionized clouds. By default we limit the number of energy levels for Stout iron species to 25 and all other elements to 15. Plasmas in collisional equilibrium tend to be hotter, for a given stage of ionization, than a photoionized plasma. The **Coronal** command, described on page 124, is used for the collisional ionization case. The limits to the number of levels are then 100 for iron and 50 for all others. The **levels** option will overwrite these limits. The first number is the iron level limit, while the second number is for all other Stout elements. If no numbers appear but the keyword **MAX** does, we use all of the available energy levels.

Chapter 13

CONTROLLING CHEMICAL MODELS

13.1 The Chemistry command

Several chemistry sub-commands can be specified on a single line, separated by commas, e.g.
chemistry reaction "S+,Mg => S,Mg+" off, reaction "S+,Fe => S,Fe+" off

13.1.1 Chemistry reaction

The keyword **reaction** takes a string argument for the reaction which is being referred to. Care is taken to ensure that the effect of this command does not depend on the order in which reactants or products are specified.

At present, it has a single option, **off**, which disables the reaction.

A list of reactions can be specified for the command, i.e.
chemistry reaction ("S+,Mg => S,Mg+", "S+,Fe => S,Fe+") off
The brackets, and commas between entries, are required.

Chapter 14

DYNAMICAL & TIME-DEPENDENT CALCULATIONS

14.1 Overview

This section describes the dynamical and time-dependent calculations that can be done with the code. This physics was developed in collaboration with Robin Williams and Will Henney, and Henney et al. (2005) and Henney et al. (2007) describe first applications.

These calculations begin by doing several iterations to compute the static structure. This establishes the optical depth scale that is used in later calculations. The number of static calculations is 3 by default and is reset with the **set dynamics relax** command.

Advective terms are included in the balance equations by adding upstream quantities that are taken from the previous iteration. Problems would arise if we needed to look upstream beyond the outer edge of the computed radial grid. To prevent this the outer radius found in the static iterations will be the outer radius of later calculations. This overrides the behavior of the **radius** command.

Warning! The physics behind these commands is still being developed. The heuristics that are needed to control changes in the time steps or the advancing upstream parameters are not complete. Output options to capture the information in the flow or over time are not complete. All of this is an area of active development.

In the following discussion commands that are still under development are introduced by the symbol and are in a grey box.

14.2 cosmology

This follows the time dependent recombination of the early universe. It was added by Ryan Porter. It has several options.

14.2.1 cosmology omega

This specifies an omega parameter. It accepts the keywords **baryon**, **radiation**, **matter**, **lambda**, and **K**.

14.2.2 cosmology Hubble

This specifies the Hubble constant in units of 100 $\mathrm{km\,s^{-1}\ Mpc^{-1}}$.

14.3 A cooling or heating non-equilibrium gas

This case is a non-equilibrium gas that cools down or heats up after an initial state is established. The usual case is a grain-free gas which is not exposed to an external radiation field. A unspecified process has established the gas in an equilibrium state, and the gas is allowed to freely cool or heat. Recent papers which studied this case are Chatzikos et al. (2015) and Gnat and Sternberg (2007).

This calculation is set up by doing a coronal equilibrium calculation with the **coronal 1e4 K init time** command described on page 124. This establishes an equilibrium collisionally ionized gas by doing a number of iterations with the kinetic temperature held fixed at the specified value. By default the number of iterations is 3, but may be changed with the command **set dynamics relax [niter]**.

The initial timestep is by default set to a small fraction (10^{-4}) of the cooling time, while subsequent timesteps are set to 4% of the cooling time.

The **iterate** command tells the code to evolve the system until some stopping criterion is met. The command **stop time when temperature below [value]** tells the code to stop when the lowest temperature falls below the specified value. The is a reasonable simulation of a constant pressure cooling flow.

As an example, the following input script evolves a parcel of gas as it cools isobarically from X-ray temperatures. The emitted X-ray spectrum is shown in Figure 14.1.

```
coronal 11.654e6 K init time
hden 0.1 linear
constant gas pressure reset
set dr 0
set nend 1
stop zone 1
iterate 300
stop time when temperature falls below 1e4 K
cosmic rays background
#
# commands controlling output    =========
set cumulative mass
set trimming off
set save prefix "cp-cool-1keV"
save time dependent   ".tim"  no hash
save continuum units angstrom ".concum" cumulative
```

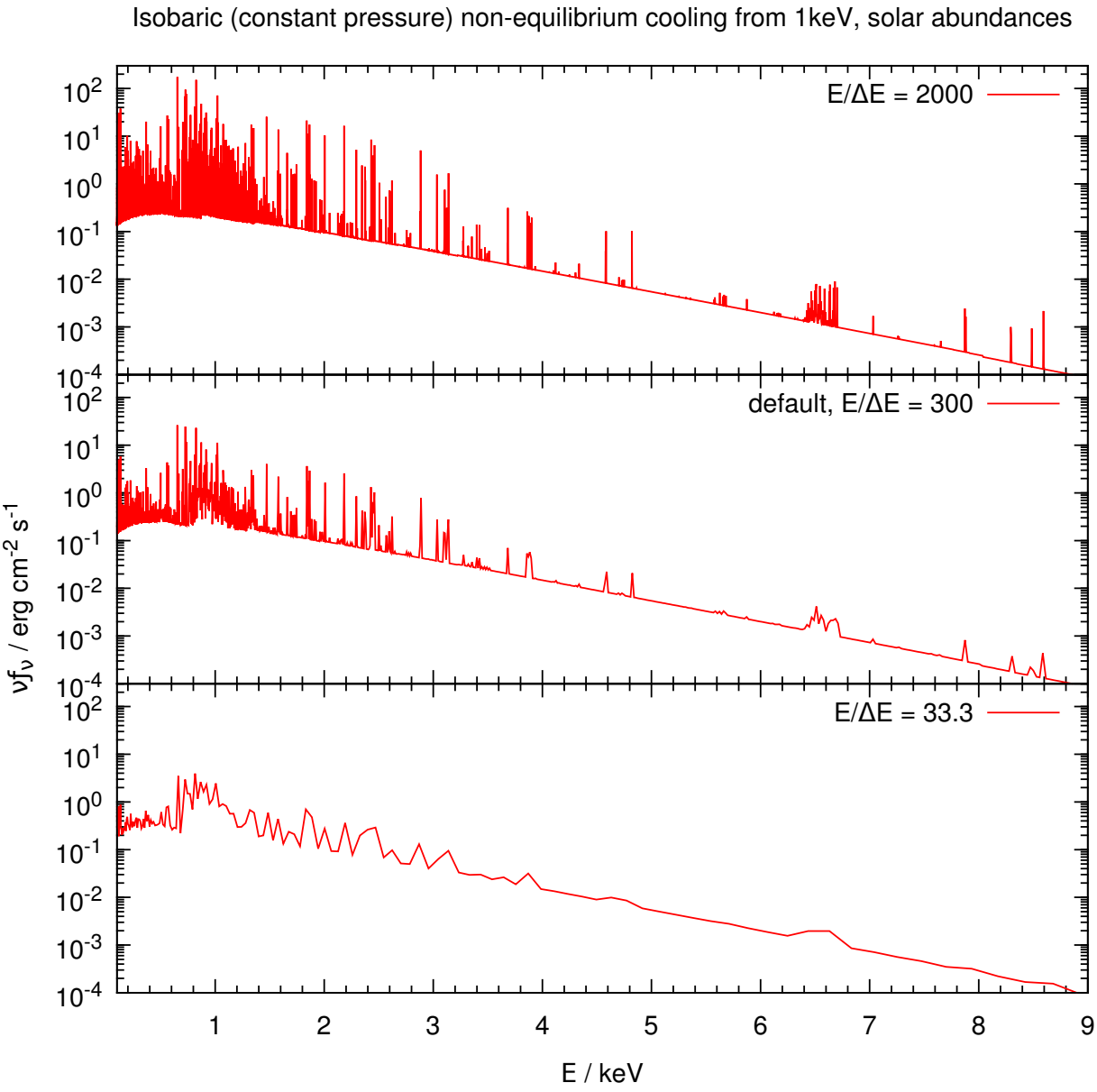

Figure 14.1: The X-ray spectrum of a unit-volume parcel of gas cooling under conditions of constant pressure (isobarically) from X-ray temperatures (1 keV), computed at three resolving powers. In the top panel, the resolving power is comparable the proposed *Athena* resolving power around 6 keV. In the bottom panel, the resolving power is comparable to the *Chandra* non-dispersive resolving power around 1 keV. In the middle panel, the resolving power is at the code default. This functionality is described in Chatzikos et al. (2015).

14.4 Time-variable incident radiation field

This follows the time-dependent evolution of a cloud near a variable continuum or heat source.

Time-dependent calculations are done by setting up an initial geometry using the usual commands that specify a cloud and the incident radiation field. The **time** keyword establishes which energy sources vary with time. The keyword can appear on any of the commands that specify the luminosity of the radiation field and on the **hextra** command. The **trace** option will turn on trace printout.

If a **time** command also occurs then all radiative and heat sources with the **time** keyword will change with time. The **time** command sets a time step and stopping time. Each iteration is a time step and the number of time steps is set with the **iterate** command. The calculation will stop when either the number of iterations or the stopping time is reached.

14.4.1 time command

The first argument on the **time** command is the log of the initial time step in seconds. The calculation starts at time zero and the first time step will be this long. Later time steps will have their size adjusted by the rate of change in conditions in the gas. The optional second number gives the log of the stopping time in seconds. The calculation will stop when either this time or the number of iterations is reached.

The **time** command is followed by a table giving the logs of the elapsed time (seconds) and the log of a scale factor giving the brightness of the variable radiation source relative to the value specified in the luminosity command.

The variation of luminosity is linearly interpolated between the points specified: the first point gives the initial scale factor. The code will not extrapolate to times beyond the end of the table, but for one exception. If the keyword **cycle** appears on any line specifying an elapsed time, the code will treat that time as a period and repeat all previous time steps with that period (the specified period must be greater than all previously-specified elapsed times). It will stop if it needs to look up times beyond the last tabulated value. The following is a simple example:

```
# set the continuum and a variable flux of H-ionizing photons
blackbody T=4e4 K
phi(H) 13.5 time
# set a limit of 50 time steps.  The calculations will actually
# stop when we reach total age
iterate 50 times
# the time command sets a time step of 1e7 s and a stop time of 1e8 s.
# This requires ten time steps.  We set 50 iterations above but the
# calculation will actually stop when the age of 1e8 s is reached.
# The time command is followed by a table giving the log of the age and the
# log of the brightness of the energy source relative to its initial value.
# The table ends with the line saying "end of times"
time steps=7 s stop at 8 s
age = 0 scale = 0
age = 7.3 scale = 0
```

```
age = 7.35 scale = 5
age = 7.5 scale = 5
age = 7.55 scale = 0
age = 10 scale = 0
end of times
# commands to set hydrogen density, only done one zone
hden 5
stop zone 1
```

In this example only one radiation field is given and the flux of hydrogen-ionizing photons varies with time. A constant time step of 10^7 s is used. The **time** command is followed by the series of lines giving the brightness of the incident radiation field as a function of time. The series ends with the line starting with **end**. Each of the lines in the age—brightness table starts with **age** but these characters are ignored. They are only present to improve readability.

14.4.2 Time-step size increment

```
# first set the continuum
blackbody T=4e4 K
# this continuum will vary with time
ionization parameter -2 time
# the CMB will be constant
CMB
# do 50 time steps
iterate 50 times
# the time command itself, followed by a table of times & continua
time 7.5
 7     scale= 0
 7.1   scale=-5 recombination
 9.3   scale=-5
 9.301 scale= 0 ionization, factor = 1.1
 20    scale= 0
end of time
```

In this example the incident radiation field is a blackbody with its intensity set by an ionization parameter. The code does fifty iterations with a time step of $10^{7.5}$ s since this is specified on the initial **time** command line. In the table the blackbody is left at its initial brightness for the first 10^7 s then lowered to 10^{-5} of the initial value for a total of $10^{9.3}$ s when it is raised back to the initial value.

As given above, the time steps always have the same size, the size given on the initial **time** command. It is also possible to specify the size of the time step for each elapsed time that appears in the table that gives the continuum brightness. If all of the optional time-step sizes are specified then they are used in place of the time step given on the **time** command. The time step sizes are multiplicatively increased by the time-step scale factor. It is also possible to give a time-step scale factor saying how to increase the time step.

If any are missing then the time step on the command is used and is kept constant. The table ends with a line saying **end**. If the optional end time is not specified on the **time** command then the calculation will continue until the specified number of iterations has been performed.

14.4.3 Keywords on individual time-step entries

Most of the letters on the lines giving the times and scale factors are totally ignored. They are there to make the table easier to read. There are keywords which indicate that a particular time step marks the start of a special case, such as the recombination of a highly ionized gas or the propagation of an ionization front into neutral gas.

The keyword **recombination** indicates that this time step is the first where the continuum source will be dramatically fainter. Certain heuristics are used to follow the recombination and cooling of the gas.

The recombination heuristics are not complete and the ionization heuristics do not exist..

14.4.4 Varying individual continua

The keyword **time** is used to specify which continua will be scaled with time and can appear on any of the luminosity / intensity commands given in Chapter 5. This makes it possible to vary only some of the continua while others are left constant. For instance, a constant CMB and a variable star can be treated. At least one of the luminosity commands must have a **time** keyword for time-dependent calculations to be done.

The **hextra** command includes the **time** keyword. The variable scale factor will apply to the extra heating in this case.

14.4.5 Controlling the number of time steps

Each iteration is a time stop. The **iterate to convergence** command has a special meaning in time-dependent calculations. In this case it tells the code to keep performing iterations, stepping through time, until one of the special criteria set with a **stop time** command is satisfied. The following **stop time** commands are implemented.

Stop temperature time [exceeds]

The **stop temperature** command is described on page 178. It has two forms. Usually it specifies the lowest temperature to allow in a calculation. If the keyword **exceeds** appears then it specifies the highest temperature to permit.

The **time** keyword tells the code to stop time when the highest or lowest temperature occurring anywhere in the computed structure is above or below the limits set with this command.

14.4.6 Other commands

The **coronal** command has the **time init** option. This will include thermal collisional ionization for the initial evaluations of the simulation. The kinetic temperature will be free when the time-dependent calculation starts.

14.5 Wind velocity=300 km/sec [mass =1.4 Msun, disk, ballistic, static]

The geometry will be a wind. The first number is the initial velocity in $\mathrm{km\,s^{-1}}$. The second optional number is the central mass in solar mass units. Both are linear quantities. The previous form of this command without a **velocity** keyword is now deprecated.

The equations of motion are solved including the inward pull of gravity. The gravitational acceleration in the default non-rotating case is evaluated as

$$g = -\frac{GM}{r^2}\,[\mathrm{cm\,s^{-2}}]. \tag{14.1}$$

The mass of the central object is given in solar masses on the command line but is evaluated in gm within the code. If the mass is not specified then zero is assumed. If the mass is specified and the keyword **disk** also appears then the geometry is assumed to be a rotating disk. In this case the inward gravitational acceleration is

$$g = -\frac{GM}{r^2}\left(1-\frac{r_\mathrm{o}}{r}\right) \quad [\mathrm{cm\,s^{-2}}] \tag{14.2}$$

where r_o is the inner radius.

The line widths and escape probabilities are evaluated in the Sobolev or Large Velocity Gradient (LVG) approximation. The effective line optical depth is given by

$$\tau_{l,u}(R) = \alpha_{l,u}\min(r,\Delta r)\left(n_l - n_u\frac{g_l}{g_u}\right)\left(\frac{u_{th}}{\max(u_{th},u_\mathrm{exp})}\right) \quad \text{[Napier]} \tag{14.3}$$

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

14.5.1 Ballistic solutions

A **ballistic** flag (or *positive velocity* in the case with no **velocity** keyword) indicates the hypersonic wind solution that has been a part of the code since its beginnings. (Note that the negative velocity case has however not yet been tested thoroughly.) The equations of motion of the gas are solved, not including forces due to material pressure gradients. Acceleration due to line and continuous opacity of the gas and deceleration due to the gravity of the central object are included. The calculation will stop if the gas comes to rest or if any of the other stopping criteria is met. The initial velocity must be above the sonic point. Further details are presented in a section in Part 2.

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

Without a **ballistic** flag (or A *negative velocity* in the case with no **velocity** keyword) the code simulates the flow including corrections due to pressure gradients in the flow, appropriate to a weak-D or D-critical H II region (Henney et al., 2005). (Note that the positive velocity case has however not yet been tested thoroughly.) This physics is currently under development in collaboration with Will Henney and Robin Williams. The central object mass is set to zero.

14.5.2 wind advection, velocity=-5 km/s

The **advection** keyword says to include the effects of advection on the thermal and ionization equations. Currently advection is only treated in the case where the velocity is negative, corresponding to a flow off a molecular cloud towards the source of ionizing radiation. The argument is the initial gas flow speed in $\mathrm{km\,s^{-1}}$. This physics is being developed in collaboration with Robin Williams and Will Henney and is not yet mature.

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

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

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

```
wind advection velocity -20 center 1e16 index -2
```

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

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

wind advection dfdr- 20.90 index 2

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

Convergence- the level of convergence can be estimated by comparing the discretization error and convergence error as a function of iteration, as shown in Figure 4 of Henney et al. (2005).

14.5.3 Wind 5 km/s no continuum

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

The **no induced processes** command will turn off line radiative acceleration. It has other physical consequences since all fluorescent excitation processes are turned off.

14.6 Miscellaneous dynamics commands

14.6.1 Constant pressure time

This allows the pressure to vary with time. See Section 8.2.5 *Constant pressure timescale = 2.3e9 s, alpha=-1* for more details.

14.6.2 Set dynamics [options]

This changes some aspects of the dynamics solutions when advective winds are used.

set dynamics advection length [fraction] This sets the look-back distance used for the advection terms in the initial iterations of the solutions. Its optimum value is determined by a compromise between speed of convergence (which is improved for large lengths) and accuracy (which is improved for small lengths). The code automatically cuts the look-back distance when good convergence is obtained so a reasonably large value can be assumed initially.

If the keyword **fraction** occurs then the number on the line will be the fraction of the total depth of the model. It is the log of the fraction if it is negative. If fraction does not occur then the number is the log of the scale length in cm.

set dynamics antishock depth 16 would put an anti-shock at a depth of 10^{16} cm.

set dynamics antishock Mach 1.01 would put in an anti-shock when the mach number fell below 1.01.

set dynamics population equilibrium This uses the equilibrium solution rather than the non-equilibrium solution that is done in a dynamical or time-dependent model.

set dynamics pressure mode This changes some aspects of how the pressure solver deals with the sonic point. The options are: **original**, **subsonic**, **supersonic**, and **strongd**. The default chooses subsonic or supersonic based on a comparison of the ram pressure and gas pressure at the face. **Original** mode uses similar criteria to choose in each zone but is liable to oscillation. **Subsonic** and **supersonic** will force a globally subsonic or supersonic solution. **Original** and **strongd** can pass through a sonic point. **Strongd** does this by following the supersonic branch until there is no valid pressure solution, then following the minimum of the pressure-density relation until it can leave along the subsonic branch. By varying the

face conditions, the depth of the internal region of unphysical solution values can be minimized, to converge on a physical strong-D type solution.

set dynamics shock depth. The argument is the log of the shock depth in cm.

set dynamics relax 4. This gives the number of static iterations to be performed before dynamical or time-dependent behavior begins. The initial iterations are done to establish the equilibrium geometry, optical depth scale, and other properties. The default is to do 3 iterations to relax the solution.

14.7 Useful output options

14.7.1 Print cumulative lines

This gives the time-integrated energy in the emission lines reported in the main printout. See page 186 for a description of this command.

14.7.2 Save commands, general comments

Save commands count every time step counting as an iteration. If you specify the **last** option you will only obtain save output for the last time step.

14.7.3 Save cumulative continuum

This gives the time-integrated cumulative energy in the spectrum reported in the **Save continuum** command. See page 203 for a description of this command.

14.7.4 Save time dependent

This gives some information about each time step. See page 240 for more information.

14.7.5 Set cumulative

Time-integrated calculations for emission lines and continuua are permitted. This command controls the weight each timestep receives in the integration, and it is fully described in Section 19.13.13.

14.7.6 Set save hash options

The **set save hash** command (page 304) has a **time** option to include the time on every time step in save output.

Chapter 15

STOPPING CRITERIA

15.1 Overview

CLOUDY will stop at some depth into the cloud. The physics that sets this depth is important since it can directly affect predicted quantities. This chapter describes various stopping criteria.

Two geometries, matter bounded and radiation bounded, can be identified for an ionized gas.

A radiation-bounded cloud is one where the outer edge of the emitting gas is a hydrogen ionization front. In this case the calculation stops because nearly all ionizing radiation has been attenuated and the temperature falls below 4000 K, the default lowest-allowed kinetic temperature. This choice of lowest temperature was made with optical emission lines in mind. Setting another outer limit is not necessary unless molecules or lines with very low ionization and excitation potentials (e.g., the [C II] or [O I] far infrared lines) are of interest. You lower the stopping temperature with the **stop temperature** command.

In a matter-bounded cloud the gas is optically thin to energetic radiation and the outer radius of the cloud must be specified. This could be a column density, physical thickness in cm, or optical depth. More than one stopping criterion can be specified and the calculation will stop when the first one is met.

CLOUDY will say why it stopped after the results of the last zone calculation are printed. It is very important to make sure that the calculation stopped for the reason you intended.

If no stopping criteria are set the calculation will usually stop because the default lowest temperature (4000 K) or the default greatest number of zones (1400) was reached.

15.2 Danger! Understand why the calculation stopped!

Sometimes the predicted emission-line spectrum will depend strongly on the thickness of the cloud. The cloud thickness is set by the stopping criteria. The predicted intensity will depend on thickness if the outer edge of the cloud is within a line's creation region. This is often the case for some lines in an X-ray irradiated gas and for any radiation field and molecular or low-ionization infrared lines.

There are several checks that should be made to confirm that the spectrum is the one expected and not an artifact of the cloud thickness or stopping criteria. The first and most important is to understand *why* the calculation stopped. This is explained in the first comment after the last zone

is printed. First locate the print out for the last zone. The following example, from the `pn_paris` simulation, shows the printout that includes the last zone's results and the start of the calculation's summary. The first line after the ionization distribution of iron gives the title for the model and the line after that gives the reason that the calculation stopped. In this case the calculation stopped because the kinetic temperature fell below the lowest temperature, which was left at its default of 4000 K. This occurs very near the hydrogen H^+ - H^0 ionization front. Optical and UV lines form at higher temperatures so the calculation would include all contributors to the optical/UV spectrum. Lines such as [C II] 158 μm or [Si II] 34.8 μm form in cool neutral gas, so these lines would become stronger if the calculation went more deeply into cold neutral gas.

```
####150  Te:3.978E+03 Hden:3.000E+03 Ne:1.276E+02 R:4.062E+17 R-R0:3.062E+17 dR:5.658E+13 NTR:  5 Htot:4.094E-19 T912: 9.97e+07###
 Hydrogen      9.70e-01 2.97e-02 H+o/Hden 1.00e+00 3.57e-09 H-    H2 1.57e-07 5.07e-10 H2+ HeH+ 8.59e-08 Ho+ ColD 3.27e+19 8.86e+20
 Helium        8.77e-01 1.23e-01 1.81e-04 He I2SP3 4.78e-08 5.43e-16 Comp H,C 1.39e-26 3.59e-27 Fill Fac 1.00e+00 Gam1/tot 6.57e-01
 Carbon        2.50e-04 9.97e-01 3.12e-03 0.00e+00 0.00e+00 0.00e+00 0.00e+00 H2O+/O   0.00e+00 OH+/Otot 0.00e+00 Hex(tot) 0.00e+00
 Sulphur    0  7.02e-05 9.64e-01 3.56e-02 2.36e-06 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 Argon      0  7.82e-01 1.96e-01 2.24e-02 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
 Iron       0  9.25e-06 9.97e-01 2.80e-03 3.42e-06 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00 0.00e+00
    parispn.in Meudon Planetary nebula
   Calculation stopped because lowest Te reached.    Iteration  1 of  1
   The geometry is spherical.
  !Some input lines contained [ or ], these were changed to spaces.
  !Non-collisional excitation of [OIII] 4363 reached   2.35\% of the total.
  !AGE: Cloud age was not set.  Longest timescale was 3.68e+012 s = 1.17e+005 years.
   Suprathermal collisional ionization of H reached 9.48\% of the local H ionization rate.
   Charge transfer ionization of H reached 8.94\% of the local H ionization rate.
```

Left on its own, the code will probably stop when the temperature falls below the default lowest temperature of 4000 K. This is what happened in the preceding example. This temperature was chosen for two reasons; a) collisionally excited optical and ultraviolet lines generally form in gas hotter than this (but infrared lines will form at far lower temperatures) and b) more than one thermal solution is possible for temperatures around 3000 K (Williams, 1967), so thermal instabilities may result if the gas extends to cooler temperatures.

It is a good idea to check whether the predictions would change if the model were made thicker or thinner. It is safe to assume that a line's intensity does not depend on the thickness of the cloud if the final temperature is well below the excitation potential of the line or the gas is more neutral than the species of interest.

The chapter *Output* in Part 2 of this document lists all possible reasons for stopping.

15.3 Radius inner =18 [thickness =16; parsec; linear]

The optional second number on the **radius** command sets the thickness or outer radius of the cloud.

15.4 Stop Av 12.1 [point, extended]

This stops a calculation at a specified visual extinction A_V. The value is the extinction in magnitudes at the isophotal wavelength of the V filter (5500 Å). The number is the linear extinction in magnitudes unless it is negative, when it is interpreted as the log of the extinction. Note that there must be spaces before and after the key "**AV**."

Properties of grains are determined by the **grains** command. The distinction between the extinction for a point versus an extended source is described in section 7.6 of AGN3. By default the A_V specified with this command will be for a point source, which is the quantity measured in

extinction studies of stars. The extended-source extinction, appropriate for extinction of a spatially-resolved emission-line region,, is specified with the keyword **extended**.

15.5 Stop column density = 23 [neutral; ionized; total; . . .]

This stops the calculation at the specified hydrogen column density $N(\mathrm{H})$ [cm^{-2}]. There are several optional keywords which determine whether the column density is the total (the default), the ionized hydrogen column density, the neutral hydrogen column density, or several other column densities. The default stopping column density is 10^{30} cm^{-2}.

By default the number on the line is interpreted as the log of the column density. The keyword **linear** forces that interpretation.

15.5.1 Stop column density 23

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

$$N(\mathrm{H}) = \int \left\{ n\left(\mathrm{H}^0\right) + n\left(\mathrm{H}^+\right) + 2n\left(\mathrm{H}_2\right) + \sum_{other} n\left(\mathrm{H}_{other}\right) \right\} f(r)\, dr\, [\mathrm{cm}^{-2}] \tag{15.1}$$

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

15.5.2 Stop neutral column density 23

The number is the log of the atomic hydrogen column density

$$N\left(\mathrm{H}^0\right) = \int n\left(\mathrm{H}^0\right) f(r)\, dr\, [\mathrm{cm}^{-2}]. \tag{15.2}$$

15.5.3 Stop ionized column density 23

The number is the log of the ionized hydrogen column density

$$N\left(\mathrm{H}^+\right) = \int n\left(\mathrm{H}^+\right) f(r)\, dr\, [\mathrm{cm}^{-2}]. \tag{15.3}$$

15.5.4 Stop atomic column density 21.3

In some PDR literature the atomic hydrogen column density is defined as the sum of H^0 and H_2. This command allows the calculation to stop as this summed column density, defined as

$$N\left(\mathrm{H}^0 + 2\mathrm{H}_2\right) = \int \left[n\left(\mathrm{H}^0\right) + 2n\left(\mathrm{H}_2\right) \right] f(r)\, dr\, [\mathrm{cm}^{-2}]. \tag{15.4}$$

This command was added by Nick Abel. Note that this counts each H_2 as two hydrogen atoms.

15.5.5 Stop H/TSpin column density 17.2

This calculation stops at the specified integral

$$N\left(\mathrm{H}^0\right)/T_{spin} = \int \frac{n\left(\mathrm{H}^0\right)}{T_{spin}} f(r)\, dr \,[\mathrm{K}^{-1}\,\mathrm{cm}^{-2}]. \tag{15.5}$$

Note that T_{spin} is computed self-consistently including the effects of pumping by the background continuum (usually the CMB) and the local Lα radiation field.

15.5.6 Stop H2 column density 19.2

The calculation stops at the specified molecular hydrogen column density

$$N(\mathrm{H_2}) = \int n(\mathrm{H_2})\, f(r)\, dr \,[\mathrm{cm}^{-2}]. \tag{15.6}$$

The 2 in **H2** must come before the log of the column density in the command. This is really the H_2 column density, not twice it.

15.5.7 Stop CO column density 19.2

The calculation stops at the specified column density in CO

$$N(\mathrm{CO}) = \int n(\mathrm{CO})\, f(r)\, dr \,[\mathrm{cm}^{-2}]. \tag{15.7}$$

This command was added by Nick Abel.

15.5.8 Stop column density "OH" 19.2

The calculation stops at the specified column density in the species given in quotes. If the species is not recognized, the command will have no effect. Species are described on page 14.

15.5.9 Stop effective column density 23

This is actually a form of the **stop optical depth** command. Usually, low-energy cutoffs in X-ray spectra are parameterized by the equivalent column density of a cold neutral absorber with cosmic abundances. Actually what is measured is an optical depth at some energy, generally around 1.0 keV. If the gas is ionized then a much larger column density will be needed to produce the observed absorption. The difference can be more than an order of magnitude. This command stops the calculation when the incident continuum has been attenuated by the appropriate absorption at 1.0 keV. The calculation stops when the absorption optical depth at 1.0 keV (neglecting scattering opacities) reaches a value of

$$\tau_{abs}\,(1.0\,\mathrm{keV}) = N_{effec} 2.14 \times 10^{-22}\,[\mathrm{Napier}] \tag{15.8}$$

at 73.5 Ryd. The argument of this command is the log of the effective column density N_{effec}. The absorption cross-section per proton for cold neutral gas is taken from Morrison and McCammon

(1983). Scattering opacities *are not* included in this optical depth. No attempt is made to use realistic physical conditions or absorption cross sections—this command follows the Morrison & McCammon paper.

If the gas is highly ionized then the actual column density will be greater than the effective column density. It will be less if the abundances of the heavy elements are greatly enhanced.

15.6 Stop continuum flux 200 micron 65 Jansky

This commands allows you to stop the calculation when an observed continuum flux or surface brightness at an arbitrary wavelength is reached. Note that both the keywords **continuum** and **flux** need to be present on the command line. The first parameter must be the vacuum wavelength or frequency of the observation. The second must be the flux or surface brightness. The syntax for these parameters is exactly the same as for the **optimize continuum flux** command described in Section 17.5.3. If the flux is ≤ 0, or the keyword **log** appears on the line, it will be assumed that the flux is logarithmic (but not the wavelength or frequency!).

The code will implicitly behave as if a **set nFnu add** command had been given to add the requested frequency point. Observed fluxes must be dereddened for any extinction inbetween the source and the observer. The same normalization will be used as in the rest of the Cloudy output. So if you include the **print line flux at earth** command, you should enter the observed flux at Earth here. See Section 16.19.7 for further details. See also Section 19.13.38 for a discussion of the components that are included in the continuum flux prediction.

There is no limit to the number of **stop continuum flux** commands. The following gives some examples of its use:

```
# make sure we normalize the flux as observed
print line flux seen at earth
# and set the distance
distance 980 linear parsec

# stop when a dust continuum flux, e.g. measured by IRAS is reached
stop continuum flux 60 micron 220 Jy

# stop when a measured radio continuum flux is reached
stop continuum flux 6 cm 50 mJy
# same for an OCRA measurement...
stop continuum flux 30 GHz 0.1 jansky

# this will stop on a flux of 1e-21 erg s^-1 cm^-2 Hz^-1 (100 Jy) at 100 micron
stop continuum flux 100 micron -21 erg/s/sqcm/Hz
```

15.7 Stop depth . . .

This is another name for the **stop thickness** command described on page 178 below.

15.8 Stop eden 3 [linear]

The calculation stops when the electron density falls below the indicated value. The number is the log of the electron density [cm^{-3}]. The optional keyword **linear** will force the argument to be interpreted as the quantity itself, not its log. This command is one way to stop constant-temperature models. For instance, the calculation can be forced to stop at the H^+ - H^0 ionization front by setting the stopping electron density to approximately half of the hydrogen density.

The following examples show a case that will stop near the He^{2+}-He^+ ionization front and a case that will stop near the H^+-H^0 ionization front for solar abundances.

```
#
# stop at the He++ - He+ ionization front
hden 9
stop eden 9.06 # stop when helium (10\% by number) is He+

#
# stop at H+ - H0 ionization front
hden 5
stop eden 4.5 # stop when electron dens falls below H density
```

The default is an electron density of -10^{30} cm^{-3}. (The negative sign is not a typo.)

15.9 Stop efrac = 1.05

The model will stop when the electron fraction, defined as the ratio of electron to total hydrogen densities, falls below the indicated value. This is another way to stop calculations at ionization fronts. This is useful if the hydrogen density there is not known beforehand as occurs in constant-pressure calculations. The argument is the fraction itself if it is greater than zero and the log of the fraction if it is ≤ 0.

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

15.10 Stop line "C 2" 157.636m 0.2 rel to "O 3" 5006.84

The calculation will stop when the emission line with the label given within the first pair of quotes and the wavelength given by the first number exceeds an intensity given by the second number, relative to an optional second emission line. In this example the calculation will stop when the intensity of [C II] 157.636 μm reaches 0.2 relative to [O III] λ5006.84. If a second optional line is not entered it will be Hβ.[1] This can be a useful way to stop matter-bounded models. The results of this command are not exact; the final intensity ratio will be slightly larger than the ratio specified.

[1]The scaling of the line intensities can be changed with the **normalize** command. That command can change both the normalization line (usually Hβ) and its relative intensity (usually 1). If the second line is not set with the **stop line** command then Hβ is the denominator in the ratio. The **stop line** command always uses the ratio of the two line intensities on the scale that is set with the **normalize** command. In versions C07.02 and before the **normalize** command did not interact with the stop line command.

Intrinsic intensities are used by default. Emergent intensities will be used if the keyword **emergent** appears on the command line.

The line label and wavelength should be entered exactly as is discussed in Section 2.5.3 with the added note that the label should always be enclosed in double quotes since it will never start in the first column. The line label within the quotes must have the same number of spaces as in the output. "C 2" and "C 2" are not the same (the first has only one space and the second is correct with two spaces). The wavelength is assumed to be in angstrom if no letter follows it. The wavelength can be changed to micron or centimeter by immediately following the wavelength with 'm' or 'c' so that the wavelength of the [C II] 157.636 μm line would be written as "157.636m". If the keyword **air** or **vacuum** is used, it will *only* affect the wavelength immediately preceding it, so in this example:

```
stop line "c  2" 157.636m air 0.002 rel to "o  3" 5008.24 vacuum
```

the wavelength of the [C II] line will be assumed to be in air, while the wavelength of the [O III] line will be assumed to be in vacuum.

There is no limit to the number of **stop line** commands that may be entered. If more than one **stop line** command is entered, the code will stop as soon as one of the conditions is met.

15.11 Stop mass 32.98

The calculation will stop when the total mass of the computed structure exceeds the quantity entered. If the inner radius is specified (the luminosity case) then the entered number is the log of the mass in gram. If the inner radius is not specified (the intensity case) then it is the log of the mass per unit area, g cm^{-2}.

At the current time no attempt is made to make the computed mass exactly equal to the entered number. The calculation will stop after the zone where the mass is first exceeded.

15.12 Stop mfrac = 0.5

The calculation stops when the hydrogen molecular fraction, defined as $2n(\mathrm{H2})/n(\mathrm{H}_{tot})$, exceeds the indicated value. This is a way to stop calculations within a PDR. The argument is interpreted as the molecular fraction itself if it is greater than zero and as the log of the fraction if it is less than or equal to zero.

The default is a molecular fraction of 10^{37}cm^{-3}.

15.13 Stop molecule depletion -2

Nick Abel incorporated the condensation of molecules onto grain surfaces. Currently CO, H_2O, and OH condensation are treated. The density of molecules on grains is given as a species with the molecule's usual label, "CO", "H2O", or "OH", followed by the string "gr". The rates of UV and cosmic ray desorption along with accretion come from Hasegawa et al. (1992) and Hasegawa and Herbst (1993). The grain surface chemistry mentioned in these papers is not included so our

treatment of condensation is closer to the work of Bergin et al. (1995), who also use the Hasegawa et al. rates. Use the **no grain molecules** command to turn off this condensation.

15.14 Stop optical depth -1 at 2.3 Ryd

This command stops the calculation at an arbitrary continuum absorption optical depth. The first number is the log of the optical depth. The optical depth is interpreted as a log by default. If the **linear** keyword occurs then the number is interpreted as the linear value. The optical depth is only for absorption and does not include scattering opacities. The second number is the energy in Rydbergs. It is interpreted as a log if it is negative, as linear if positive, and must be within the energy bounds considered by the code (presently 3.040×10^{-9} Ryd to 7.354×10^{6} Ryd). At present only one stopping optical depth can be specified. If more than one is entered then only the last is honored.

It is traditional in X-ray astronomy to characterize low-energy cut-offs as the equivalent *completely neutral* column density for *solar* abundances. This is not correct when the gas is ionized (since the high-energy absorption opacity is diminished) or when the abundances of the heavy elements are enhanced (since the high-energy opacity is increased). For extreme cases these effects can change the opacity by more than an order of magnitude. The deduced column density is underestimated by the same amount. It is better to convert the deduced column density back into an optical depth at 0.5 or 1 keV (this is actually the observed quantity) and use this optical depth and energy as the stopping criteria than to use the deduced column density as a stopping criterion. Either this command, or the **stop effective column density** command (which is actually a form of the **stop optical depth** command can be used to stop the calculation at an X-ray optical depth corresponding to a certain low-energy absorption.

The optical depth used in this command is the absorption optical depth and does not include scattering opacities. In general, the effects of scattering opacities are much more geometry dependent than absorption opacities.

15.14.1 Stop Balmer optical depth $= -.3$

This command is a special case of the **stop optical depth** command in which the energy does not need to be specified but the keyword **Balmer** is given. It will stop the calculation when the log of the absorption optical depth at the Balmer continuum threshold ($\nu = 0.250$ Ryd) reaches the specified value. The default is $\tau_{Bac} = 10^{20}$ and the optical depth is always interpreted as a log. This is the *total absorption* optical depth at the Balmer edge and includes all computed absorption opacity sources such as grains or free-free absorption, but neglects scattering.

15.14.2 Stop Lyman optical depth = 5

This is a special case of the **stop optical depth** command in which the energy does not need to be specified but the keyword **Lyman** is given. The number is the log of the Lyman limit optical depth, τ_{912}. The default value is $\tau_{912} = 10^{20}$. The stopping criterion is *really* the *total* 912Å *absorption* optical depth and *not* the hydrogen Lyman limit optical depth at 912Å. These are not exactly the same, especially when grains are present or the abundances of the heavy elements are enhanced.

15.14.3 Stop 21cm optical depth = linear 3

This stops the calculation when the *line center* optical depth of H I λ21 cm reaches the entered value. The populations of the hyperfine levels are determined by solving the full level populations with the effects of pumping by Lα and the external continuum included. The calculation stops when the optical depth from the continuum source to the outer edge of the cloud exceeds the entered value. The actual optical depth with be greater than or equal to the indicated value.

Note that CLOUDY always uses the dimensionless *line center* optical depth. The *integrated* optical depth is the standard used in radio astronomy. The *integrated* optical depth is called the *mean* optical depth in optical spectroscopy. The dimensionless mean optical depth is $\sqrt{\pi}$ larger than the line center optical depth. In radio astronomy, the integrated optical depth is often given in velocity units, km s^{-1}.

15.15 Stop pfrac = 0.23

The calculation will stop when the proton fraction, defined as the ratio of proton (ionized hydrogen) to total hydrogen densities, falls below the indicated value. This is another way to stop calculations at ionization fronts and is useful if the hydrogen density there is not known beforehand. This occurs in constant pressure calculations, for instance. The argument is interpreted as the fraction itself if it is greater than zero and the log of the fraction if it is less than or equal to zero.

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

15.16 Stop radius 17.3 [parsec; linear; 17.6 on sec iter]

This sets an upper limit to the radius of the cloud. The argument is interpreted as the log of the radius unless the keyword **linear** appears. The default unit is centimeter but it will be interpreted as the log of the radius in parsec if the keyword **parsec** appears on the line.

The **stop radius** command has the same effect as the optional second number on the **radius** command.

Any number of radii may be entered on the command line. Each will be the ending radius for consecutive iterations. The limit to the number of stopping values is set by the limit to the number of iterations that can be performed. If fewer numbers are entered than iterations performed then the last number will be used for all further iterations.

This command is useful if you want to vary the inner radius in an optimizer run, but want to keep the outer radius fixed. If you do so, make sure that you set the upper limit of the inner radius to a value less than the outer radius. You can do this with the **optimize range** command. If you did not set an upper limit and the optimizer would try to move the inner radius beyond the outer radius, the code would abort.

15.17 Stop temperature 1e3 K [linear, exceeds]

The calculation will stop when the kinetic temperature drops below T_{low}, the argument of this command. The argument is interpreted as the log of the temperature if ≤ 10 and as the linear temperature if > 10 or if the **linear** keyword appears.

The default value is $T_{low} = 4000\,\mathrm{K}$. Gas cooler than this produces little optical emission, but may be a strong emitter of infrared lines such as the [C II] 158 μm or the [O I] ^{3}P lines. The lowest temperature allowed, T_{low}, should be adjusted with this command so that the excitation potential $h\nu$ is $\cong kT_{low}$ for the lowest excitation potential transition considered. Note that more than one temperature is sometimes possible when $T \approx 10^3$ K (Williams, 1967) so thermal stability problems may develop if T_{low} is lowered below a few thousand degrees Kelvin. This issue is discussed further in the section *Problems* in Part 2 of this document.

If the keyword **exceeds** appears on the line then the specified temperature will be the highest allowed temperature. The calculation stops if the temperature exceeds the value on the command. This might be necessary when a grid of models is computed but those in the high temperature phase (i.e., $T > 10^5$ K) are not of interest. The other rules for the command are unchanged.

If no number appears on the command line, but the keyword **off** does, then temperature will not be used as a stopping criterion.

15.18 Stop thickness 9.3 [parsec; linear; 23 on sec iter]

This sets an upper limit to the thickness of the cloud. The thickness is the distance [cm] between the illuminated and shielded faces. The argument is interpreted as the log of the thickness unless the keyword **linear** appears. The default units are centimeters but it will be interpreted as the log of the thickness in parsec if the keyword **parsec** appears on the line.

The **stop thickness** command has the same effect as the optional second number on the **radius** command. This command makes it possible to set a cloud thickness when the inner radius is not specified, such as when the ionization parameter is given.

Any number of thicknesses may be entered on the command line. Each will be the ending thickness for consecutive iterations. The limit to the number of stopping values is set by the limit to the number of iterations that can be performed. If fewer numbers are entered than iterations performed then the last number will be used for all further iterations.

The keyword **depth** can be used instead of **thickness**.

15.19 Stop velocity $<$ 1 km/s

The calculation will stop if the absolute value of the wind velocity falls below the value. The stopping velocity is entered in km s^{-1} but is converted to cm s^{-1} within the code.

15.20 Stop zone 123 [21 on sec iteration,...]

This sets a limit to the number of zones that are computed on each iteration. It is not normally used. In this example the calculation will stop after computing 123 zones. The default value is

1400. Any number of zones may be entered, each being the ending zone for consecutive iterations. This limit is set by the limit to the number of iterations that can be performed. If fewer numbers are entered than iterations performed then the last number will be used for all further iterations.

The code checks that it did not stop because the default number of zones was reached. A warning will be generated if this happens since it was probably not intended. Use the **set nend** command to increase the default number of zones while keeping this checking active.

The code allocates memory to store a great deal of information for the limiting number of zones. Increasing the number of zones will also increase the memory needed to run the code. Don't do this unless you really need to use the zones.

Chapter 16

CONTROLLING OUTPUT

16.1 Overview

CLOUDY is capable of generating lots of output although its default output is minimal. Commands that control the output are described here. A Chapter in Hazy 2 describes the meaning of the output. The output commands are split into two broad classes:

- **print** commands generate output in the default output of the code,
- **save** commands generate output in separate files, typically once per zone, or once per iteration.

16.2 No buffering

This command is described in Section 19.11.4.

16.3 Normalize to "O 3" 5006.84 [scale factor = 100]

The strength of an emission line in the standard output will be given in intensity or luminosity units and as its intensity relative to a reference line. Emission-line intensities are usually listed relative to the intensity of Hβ λ4861.32Å, the default reference line. By default the reference line has an intensity of unity. This command can change the reference line to any of the other predicted lines and can change the relative intensity of the reference line to another value. The relative intensities of all lines in the spectrum will then be relative to the intensity of the line defined here. The normalization line can be specified as outlined in Section 2.5.3. As a minimum you should specify a label within double quotes and a wavelength given by the first number on the line.[1] The wavelength must match the printed wavelength to all significant figures and can be either in air or vacuum, as discussed in Section 2.5.3.

The optional second number sets the relative intensity of the reference line. If it is equal to 100, as in this example, then all intensities will be relative to a reference line intensity of 100. The

[1] The label was optional in versions 94 and before of the code, but now is required due to the large number of lines, making unique wavelengths unusual.

default is for an intensity of unity. The example given above will cause the relative intensities to be expressed relative to an [O III] λ5006.84 intensity of 100. The scale factor must be greater than zero.

The code works by finding the first line in the emission-line stack whose wavelength and label matches the line on this command. How to disambiguate multiple matches is discussed in Section 2.5.3.

The following shows some examples of the **normalize** command:

```
# normalize to spectrum to Pa
normalize to "H  1" 1.875m

# normalize spectrum to the [OI] IR line on a scale where it is equal to 100
normalize to "O  1" 63.17m = 100

# specify air wavelengths and disambiguate the line
normalize to "H  1" 4.65247m air index=11, 23
```

16.4 Print ages

The code normally assumes that the system is old enough for microphysical processes to have become time steady. This tells the code to print all of the timescales tracked by the code. These are the same timescales considered by the **age** command. Normally only the shortest timescale is printed at the end of the calculation.

If a physical process is not significant, for instance, the H_2 formation timescale in a 10^6 K gas, the age is still computed but is set to a negative number. This retains the value while not including the process when the important timescales are determined.

16.5 Print arrays

These commands are intended mainly as debugging aids.

16.5.1 Print arrays ionization

This prints the array elements that enter into solution of the ionization balance. By default it will print this information for all elements. If the keyword **only** appears then it will also look for the name of an element and will only print the array elements for that element. **Print arrays ionization only element** commands are additive. If more than one appears then only the information for the requested elements will be printed. This is a debugging aid.

16.5.2 Print arrays levels ”species”

This prints the matrix that enters the solver for the level populations of a species. Only one species per simulation may be specified. The output consists of a header line followed by the rows of the linear algebra system to solve. The syntax for specifying levels is that of **save species levels populations**, Section 16.86.1. For instance, to obtain the rows and columns for all levels the

species name must be followed by the "[:]" notation, e.g., "C+[:]". A subset may be specified by entering the desired levels in the brackets, e.g., "C+[1:5]" will print rows 1–5 and columns 1–5 of the matrix.

16.6 Print citation

CLOUDY is a research project that involves the creative efforts of many people. It should be cited as follows: "Calculations were performed with version yy.mm.dd of CLOUDY, last described by Ferland et al. (yyyy)." The numbers represent the release date and the citation is to a review paper. The citation should mention the version of the code since some predictions changes as the atomic data and treatment of physical processes improve. Old versions of the code are never deleted from the web site so it is possible to recover a version that produced a given result.

This command will print the current version number of the code and give the full bibliographic citation for the review paper.

16.7 Print constants

The physical constants stored in the header file `physconst.h` will be printed along with sizes of some variables.

16.8 Print column densities [on; off]

This controls whether the column densities of the various constituents are printed. The keywords are _**ON**_ and _**OFF**. The default is to print the column densities.

The column densities of several excited states within ground terms of some species are printed as well. The meaning of the labels for the excited states column densities is given in the discussion of ***cdColm*** in Part 2 of this document.

16.9 Print coolants, zone 135

This prints the coolants for the specified zone. If no zone number or 0 appears on the line then the coolants for *all* zones will be printed. The total cooling and the fractional contribution of the strongest coolants are printed. For each coolant a label gives an indication of the spectroscopic origin of the coolant and the following number gives its vacuum wavelength, with a 0 to indicate a continuum. The last number of the group is the fraction of the total cooling carried by that agent.

16.10 Print continuum indices

The file created by the **save continuum** command identifies a line that occurs within each continuum bin. This can be used to understand what lines contribute to the predicted spectrum. The line label is for the first line that was entered in a particular cell. It is not the strongest line and

there may be many lines contributing to a particular cell. This command will print the line energy (rydberg), continuum array index, and the line's spectroscopic notation, for every line that is included in the calculation. The lines will be printed in the order in which they are entered into the continuum. The printout can then be sorted by energy or array index to discover all lines that occur within a particular cell. This is a debugging aid.

There are two optional numbers which give the lower and upper limit to the energy range (rydberg) to be printed. All lines are printed if these numbers do not appear then.

16.11 Print departure coefficients [He-like] [element]

LTE departure coefficients for levels within an element along the H-like or He-like isoelectronic sequences will be printed. The **print populations** command controls printing individual level populations.

If the keywords **H-like** or **He-like** appear, then an element on the hydrogen-like or helium-like isoelectronic sequences will be printed. Otherwise all iso-sequences of the chosen element will be printed. The code will search for the name of an element, and if it finds one, will print only that element. If no elements are recognized then departure coefficients for all elements of the chosen iso-sequence are printed. If neither elements and iso-sequences are given or recognized, departure coefficients for all elements and all iso-sequences will be printed.

16.12 Print critical densities [He-like] [element]

Critical densities for angular momentum changing (l-changing) collisions for each element will be printed from $n = 3$ for the H-like of He-like iso-sequences.

If the keywords **H-like** or **He-like** appear, then an element on the hydrogen-like or helium-like isoelectronic sequences will be printed. Otherwise all iso-sequences of the chosen element will be printed. The code will search for the name of an element, and if it finds one, will print only that element. If no elements are recognized then departure coefficients for all elements of the chosen iso-sequence are printed. If neither elements and iso-sequences are given or recognized, critical densities for all elements and all iso-sequences will be printed.

16.13 Print errors

The code will always identify problems by either printing comments during the calculation or warnings after the calculation is complete. This says to also print these warnings to `stderr`. On many systems this output can be redirected to the screen. The **no buffering** command describes how to handle `stderr` output.

16.14 Print fixits

Print a list of issues within the code which have been noted using the `fixit()` macro in the source, and where this code has been used in the current run.

16.15 Print every 1000 [5 37 93]

This will be replaced with the **print zone** command.

16.16 Print heating

The relative heating due to each stage of ionization or physical process is printed. This is the fraction of the total heating due to this particular stage of ionization and is printed directly below the relative abundance of that stage.

16.17 Print populations [H-like carbon, to level 45]

Level populations are normally not printed for the atoms and ions of the H-like or He-like isoelectronic sequences. This will print them. If no numbers appear on the line then the first 15 levels will be printed. Enter the highest level to print on the line as an integer if more are desired.

If the keywords **H-like** or **He-like** appear, then an element on the hydrogen-like or helium-like isoelectronic sequences will be printed. Otherwise all iso-sequences of the chosen element will be printed. The code will search for the name of an element, and if it finds one, will print only that element. If no elements are recognized then departure coefficients for all elements of the chosen iso-sequence are printed. If neither elements and iso-sequences are given or recognized, populations for all elements and all iso-sequences will be printed.

The departure coefficients are printed with the **print departure coefficients** command.

16.18 Print last

Normally results for every iteration are printed as they are computed. This command says to print only results for the last iteration.

16.19 Print line options

A large block of emission-line intensities is printed after the calculation is complete.[2] This controls details of that printout. Note that each option is intended to be used on a standalone command. If multiple options are required, they should be specified on separate lines to ensure proper execution.

Some options change the layout of this information. These include options to print a single column, to sort the lines by wavelength or intensity, or to print only the strongest lines, or those within a certain wavelength range.

[2] In versions 87 and before, the code printed some relative line intensities for each zone. An extra line could be added with the **print line** command. This command, and that printout, no longer exists. Use the **save line intensities** command instead.

Other options indicate line-formation processes. A great deal of information about line formation and beaming is stored within the code but not normally printed to save space. The section of Part 2 of this document *The Emission Lines* gives more information.

Some spectra have so many lines that several different transitions may appear to have the same wavelength. This occurs to some extent for most H_2, Fe II, and He-like spectra. The **print line precision** command allows you to change the number of digits in the printed line wavelength. It may be necessary to increase the wavelength precision if the default line wavelengths are ambiguous and more than one transition appears with the same wavelength.

16.19.1 Print line all

All of the contributions to line formation, including collisions, pumping, and heating, will be printed.

16.19.2 Print line cell xx

More than one line can occur within a continuum cell in the output produced by the **save continuum** command. This command will print the label for every line that falls into a particular continuum cell. The number of the cell, with the lowest energy cell being 1, must appear.

16.19.3 Print line collisions

Collisions are often the dominant contributor to an optically thick resonance line. This adds an entry with the label **Coll** followed by the wavelength and the collisional contribution.

16.19.4 Print line column [linear]

The main block of emission lines is normally printed with four lines across the page. This command says to print lines as a single column to make it easier to enter into a spreadsheet. The keyword **linear** will cause the intensities to be printed as the linear flux in exponential format rather than as the log.

16.19.5 Print line cumulative

In a time-dependent simulation the main block of emission lines give the emission for the current time step. This command says to also print the time-integrated line emission, referred to as the cumulative emission. This "spectrum" is the total energy emitted in the lines, with units $\mathrm{erg\,cm^{-2}}$.

16.19.6 Print line faint -2 [_off]

CLOUDY will normally print the intensities of all emission lines with intensities greater than 10^{-3} of the reference line, which is usually Hβ. This changes the limit to the relative intensity of the weakest line to be printed. The argument is either the log (if ≤ 0) or the linear (if positive) intensity of the weakest line to print, relative to the reference line. The **_log** option will force interpretation as a log. The reference line is usually Hβ, and can be changed with the **normalize**

command. In the case shown here, only lines with intensities greater than 1% of Hβ will be printed.

If no numbers are entered, but the keyword **_off** appears then all lines are printed, even those with zero intensity.

16.19.7 Print line flux at Earth

If the distance to an object is set with the **distance** command and line luminosities are predicted then this command says to print the observed flux at the Earth rather than the line luminosity. The units are erg cm^{-2} s^{-1}. (No correction for interstellar extinction is included, of course). Both the keywords **flux** and **Earth** must appear. This command can be combined with the **aperture** command to simulate observing only part of a spatially-resolved object.

16.19.8 Print line heat

Fluorescent excitation is included as a line formation process. If a line is radiatively excited but then collisionally deexcited it will heat rather than cool the gas. This option prints the heating due to line collisional de-excitation. The entry will have the label **Heat** followed by the wavelength.

16.19.9 Print line H2 electronic

By default only ro-vibrational lines within the ground electronic state of H_2 are included in the emission-line printout when the large H_2 molecule is included with the **database H2** command. This command tells the code to also print electronic transitions.

16.19.10 Print line inward

Optically-thick emission lines are not emitted isotropically. The "inward" fraction of the line is the part that is emitted from the illuminated face of the cloud into the direction towards the source of ionizing radiation. This will generally be greater than 50% of the total intensity if the line is optically thick. This command prints this inward fraction with the label "Inwd" followed by the wavelength.

The optical depth scale must be fully converged for the inward intensity to be predicted. Use the **iterate to convergence** to do this.

16.19.11 Print line iso collapsed off

The model atoms for the iso-electronic sequences have both resolved and collapsed levels. Predictions from the collapsed levels can be unreliable below the critical density for *L*-mixing at a given *n*. This command will disable printing predictions from the collapsed levels.

16.19.12 Print line [intrinsic, emergent] off

By default the main line block is printed in the main output. This command will inhibit the reporting of line intensities in the main output, leading to significantly smaller output files. If the

keywords **intrinsic** or **emergent** appear, then only the intrinsic or the emergent, respectively, block of lines will be inhibited.

16.19.13 Print line optical depths [_off, faint]

Mean line optical depths are not printed by default[3]. The option tells the code to print them at the end of the iteration. For each emission line, the information printed is the species label followed by the wavelength, the total line optical depth, which includes contributions from overlapping emission lines, and finally the line optical depth due to that emission line only.

There are two optional keywords.

If **_off** appears then line mean optical depths will not be printed. This is useful if turned on in a previous iteration and no longer needed.

The keyword **faint** sets the smallest line mean optical depth to print. The default smallest mean line optical depth to print is 0.1. The log of the limit must be given. Optical depths for all lines that mase are printed.

16.19.14 Print line precision [4, 5, or 6]

This changes the number of decimal places for line wavelengths in the final pages of output. Emission-line wavelengths are normally printed with six digits, as in "4861.36A". This command can change the number of digits to 4, 5 or 6. The lines are printed in four columns when four digits are chosen. With five or six digits the lines will no longer fit across the page so the number of columns is automatically reduced to three.

16.19.15 Print line pump

All lines include fluorescent excitation by the attenuated incident continuum as a line formation process. Continuum pumping will often be the dominant formation mechanism for optically-thin high-excitation lines. This option prints an estimate of the contribution to the total line intensity from this process. The entry will have the label "Pump" followed by the wavelength.

16.19.16 Print line sort wavelength [range 3500A to 1.2m]

The output spectrum to be sorted by wavelength rather than by ion.[4] It was originally added by Peter G. Martin. If the **range** option appears then two more numbers, the lower and upper bounds to the wavelength range, must also appear. Each number is interpreted as the wavelength in angstrom by default, but is interpreted as the wavelength in micron or centimeter if the wavelength is immediately followed by an "m" or "c", respectively. The two wavelengths must be positive and in increasing wavelength order. By default the wavelengths will be assumed to have the same type (air or vacuum) as used in the output (see Section 16.19.20). Interpretation as air or vacuum wavelengths can be forced by appending the optional keyword **air** or **vacuum** directly behind the

[3]Line center optical depths were printed through version C10. Mean line optical depths are now reported. Line center optical depths are $\sqrt{\pi}$ times smaller than mean optical depths.

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

wavelength. Note that the keyword only applies to the wavelength directly preceding it, so it needs to be repeated, as shown below.

```
print line sort wavelength range 1200A vacuum to 1.2m air
```

16.19.17 Print line sort intensity

The predicted emission lines will be sorted in order of decreasing intensity.

16.19.18 Print line sum

This prints the sum of the intensities of an arbitrary set of emission lines. This can be useful for applications such as the Stoy (1933) energy balance method of determining stellar temperatures, which rely on the sum of a set of observed line intensities relative to a recombination line (see also Kaler and Jacoby, 1991 and section 5.10 of AGN3). The sum is printed as the last entry in the emission-line array as an entry with the label "Stoy" and a wavelength of 0.

Each emission line included in the sum is entered on its own line. This list begins on the line after the **print line sum** command and continues until a line with **end** in the first three columns appears. The format for entering the spectral lines is described in Section 2.5.3. The following gives an example of its use.

```
print line sum
o  3 5006.84
Blnd 3727
o  1 6300
O  3  51.80m
S  3  18.67m
s  3 9532
end of lines
```

An arbitrary number of lines can be entered into the sum.

16.19.19 Print line surface brightness [arcsec]

By default the line intensities that are printed after the calculation is complete is given as L [erg s^{-1}] for the luminosity case and $4\pi J$[erg $cm^{-2}s^{-1}$] for the intensity case. This command will change these intensities into surface brightness units. The default is per steradian but if the keyword **arcsec** appears then the surface brightness will be per square arcsec.

16.19.20 Print line vacuum

By default we follow the atomic physics convention that vacuum line wavelengths are used for $\lambda_{vac} \leq 2000$ Å and STP air wavelengths for $\lambda_{vac} > 2000$ Å. This command will change the behavior to use vacuum wavelengths for all wavelengths. Note that this will only affect the wavelengths of atomic and molecular lines printed in the output. Internally the code always uses

vacuum wavelengths.[5] This includes the **save continuum** family of commands which will always report the SED (including emission lines) in vacuum wavelengths.

Air wavelengths are shorter than vacuum wavelengths, which creates an ambiguity. Lines with a vacuum wavelength of 2000 Å would be reported with an air wavelength 1999.3521 Å. So for lines inbetween 1999.3521 Å and 2000 Å it is not clear whether these are air or vacuum wavelengths. To avoid the ambiguity, CLOUDY will assume that lines with wavelengths $\leq$ 2000 Å are always vacuum wavelengths, which creates problems for some lines. This problem can be avoided by explicitly stating the vacuum wavelength of the line even when using air wavelengths in the output, e.g.:

```
"Ne 9" 2000.31A vacuum
```

16.20 Print macros

This prints the name and status of the macros that are used in the `cddefines.h` header file. These macros are either set by the user at compiler time with the `-DMACRO` option on the compile command or by the compiler itself.

16.21 Print modules

This generates a list of all initialization modules in the current coreload. This is mainly a debugging aid.

16.22 Print off

This turns off the print out, as with the **print quiet** command. This is normally paired with a later **print on** command to avoid printing parts of the output.

There is a possible problem. The code can read its own output as input, to make it easy to rerun a model. In many initialization files the following pair of commands appears:

```
print off
commands ....
print on
```

The resulting output will print the first **print off** command, but will not print the commands or the **print on** command. If this output is used as input no further output will be created for the new model. This problem will not occur if the **print off** command includes the keyword **hide**.

16.23 Print on

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

[5] This was not the case in Cloudy versions C23 and older where the code would use air wavelengths internally unless this command was used.

16.24 Print only [header, zones]

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

16.25 Print path

The path giving the location of the data files will be printed.

16.26 Print quiet

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

16.27 Print recombination

This reports the data sources for the dielectronic (DR) and radiative recombination (RR) rate coefficients, their values at the current temperature, and the density suppression factors computed at the current density. Where the DR rate data source is listed as "mean", the value used is a guessestimate, constant with temperature to guarantee the absence of discontinuities.

The **set recombination** command described on page 296 allows details to be changed.

16.28 Print short

This shortens the detailed final printout. Only the emission lines and a short summary of some thermal properties of the model will be printed.

16.29 Print starting at 61

This option turns off *all* printout *until* the specified zone is reached. This should come last in the input stream since command lines appearing after it will not be printed.

16.30 Print UTA references

This reports the references to the UTA data used for each ionic species in the simulation. See **set UTA**, Section 19.13.53, for details on how to choose among various UTA data sets.

16.31 Print version

This prints the code, compiler, and operating system versions, along with other information.

16.32 Print zone 1000 [5 37 93]

CLOUDY will always print the results for the first and last zones. This command varies the number of zones printed between these two. If more than one number is entered then each applies to successive iterations. The example above will print every 1000 zones on the first iteration, every 5 zones on the second iteration, 37 on the next, etc. If there are fewer numbers entered than iterations performed then the last number entered will be used for all further iterations.

16.33 Save commands

16.33.1 Overview

Save commands save results into a file that can be used later. They are the primary output mechanism for CLOUDY. There are many options. For instance, physical quantities as a function of depth into the cloud, including temperature, ionization, and density, can be saved for later plotting. The emitted spectrum, or other quantities predicted by the code, can be output. The general idea is for the file produced by this command to then be post-processed by other plotting or analysis programs to produce final results.

One keyword must appear and only one keyword per line is recognized. Up to 100 **save** commands can be entered.

16.33.2 Save vs punch commands

In versions C08 and before the **save** command was called **punch**. "Punch" was an output option in FORTRAN IV and was implemented by machines that produced holes on Hollerith cards. Those machines and cards now exist only in museums. This version of CLOUDY continues to accept **punch** as an alias for **save**.

16.33.3 An output file name must appear inside double quotes

Each **save** command must specify a file name[6] for the resulting output. This file name must appear between a pair of double quotes as in `"output.txt"`. It must be a valid file name for your operating system. The following is an example.

```
save overview "model.ovr"
```

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

[6] In versions 90 and before Fortran default save units, with names like fort.9, could be used for save output. The filename must be specified with versions 91 and later.

16.33.4 Setting a prefix for all save files

A prefix can be set for all filenames with the **set save prefix** command (see Section 19.13.45). This makes it possible to set a prefix only one time for several save files, as in

```
set save prefix "Den11"
save overview ".ovr"
save continuum units micron ".con"
```

The files `Den11.ovr` and `Den11.con` will be created.

16.33.5 The "last iteration" option

Each **save** command also has a keyword **last** that will cause the output to only be produced on the last iteration. It this keyword does not appear then output will be produced for every iteration. The results of each iteration are separated by a line of hash marks ("###"). In grid runs this behavior will be slightly modified. See the description in Section 18.6.

16.33.6 The "no buffering" option

If the option **no buffering** appears then file buffering will be turned off for that file. This slows down the output considerably but ensures that all output will exist if the code crashes. There is also a stand-alone **no buffering** command to turn off buffering for the code's standard output.

16.33.7 The "clobber/no clobber" option

When the code is used as a stand-alone program to compute a single model it will open the save file at the start of the calculation and close it at the end. In a sequence of models as in an optimization run this will happen for each new model and so will overwrite results of all previous calculations.

The **no_clobber** keyword on the **save** command will produce one long file containing results of consecutive models. It tells the code to never close the file at the end of any but the last calculation and not try to reopen this file once it is open.

The default, with one exception, is to overwrite files. The **grid** command computes a series of simulations with a single input file. The entire set of output is usually needed so the default in this single case is to not overwrite files, but rather produce one large file.

Include the **clobber** keyword if you want to overwrite files, and the **no clobber** keyword if you want one large save file with successive predictions.

16.33.8 The "no hash" option

When more than one iteration is done the results of each iteration end with a series of hash marks, "###", to make the start of each iteration easy to find in an editor. These hash marks can cause problems if the file is then read in by other programs. The hash marks will not be produced if the **no hash** keyword appears.

The character string that is printed between iterations can be changed with the **set save hash** command described on page 304.

16.33.9 The "title" option

The title[7] of the model and the version number of the code will be printed on the first line of the save file.

16.33.10 The "separate" option

The default behavior of the code is to concatenate save output from different models in a grid run into a single large file. The only exception to this is the **save FITS** command since it would violate the FITS standard to combine multiple FITS spectra into a single file. If the keyword **separate** is included on the save command line, each model in the grid will produce a separate save file. They will have names `grid000000000_filename`, `grid000000001_filename`, etc., where `filename` is the file name you supplied between double quotes. The meaning of the grid index embedded in the filename can be found with the **save grid** command described in Section 16.58. When the save output is split up, only the first file will contain the save header.

16.33.11 Log vs linear output quantities

Many of the older **save** commands reported the log of the quantity. Starting with the first release after C13 we are, on a case by case basis, changing the output to give only linear quantities. Those **save** commands which have been changed recognize a **_log** keyword to give quantities in the format used in C13 and before. The commands which have been changed will indicate this with the sentence *This command accepts the **_log** option*.

16.33.12 Depth versus radius

The code and this documentation make a consistent distinction between depth and radius. The ***radius*** is the distance from a point in the cloud to the center of symmetry, generally the center of the central object. The ***depth*** is the distance from a point in the cloud to the illuminated face of the cloud. In both cases the distance is to the center of the current zone.

The output from each **save** command is described in the following sections. In those cases where quantities are given as a function of position into the cloud, the first column will usually give the depth, not the radius. You need to add the inner radius of the cloud to the depth to get the radius.

16.34 Save abundances

The log of the gas-phase densities [cm^{-3}] of the elements will be saved for each zone. This is the sum of the abundances of a chemical element in atoms, ions, molecules, and ices, but does not include grains. This provides a check for the effects of the **element table** and **fluctuations abundances** commands.

[7] The title was printed by default in versions 95 and before of the code. The title was generally deleted so that the save file could be used to make plots so it is now missing by default.

16.35 Save ages

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

16.36 Save agn [options]

This produces output files that were used to create data tables in the 2nd edition of *Astrophysics of Gaseous Nebulae*, referred to as AGN3 here. The options are the following: **charge** transfer, **recombination** coefficients, **recc** for hydrogen recombination cooling, **opacity**, **hemis**, and **hecs** (for He^0 collision strengths).

16.37 Save arrays levels "species" [fits or ppm] [iteration it] [zone nz]

This is a companion command to **print arrays levels**, and it is also a debugging aid. The syntax for the species is the same as in Section 16.5.2. Using this command generates either a FITS file or a PPM image, depending on the keyword given, **fits** or **ppm**. The image will be generated for the given species for all iterations and zones. This can be limited to specific iteration with the keyword **iteration** followed by an iteration number, and likewise to a specific zone with the keyword **zone** followed by a zone number.

The benefit of FITS images is that they can be manipulated with standard software (e.g., with ds9, FTOOLS, python, etc.) in post-processing. The **fits** option builds upon, and extends the **save fits** command. Whereas the latter generates a table, the former creates a FITS file that contains three *images*, one for each of the rate matrix, the vector of creation rates, and the computed level populations. In terms of the linear algebra system, $Ax = b$, which Cloudy solves, these images hold the A matrix, the b vector, and the computed $x = A^{-1}b$ vector, respectively.

Notice that if a subset of the energy levels is specified, only those levels will appear in the output images. For instance, "Fe+[10:20]" will generate an 11×11 image for the rate matrix (centered on the diagonal), and two 11×1 images for the creation rates, and computed populations. It is recommended that the full matrix be output at first, and then the range of levels be limited at a later stage, if needed.

Also note that a FITS file with the three images mentioned above will be created if negative populations are computed for a species when solving the relevant linear algebra equation. The output file name contains the failing species, the iteration and zone where the failure occurred, as well as a substring to distinguish the FITS file from those generated with this command; e.g., `Fe+2_it1_nz0_NEG_POP.fits`. This functionality is offered as a debugging aid.

16.38 Save monitors

The **monitor** command provides an automated way to validate the predictions of the code. Normally the results from these checks will be printed on the standard output. If this command appears then the same output will also be sent to a file.

16.39 Save average . . .

This reports averages of various quantities. It was included as a way to bring together information generated with the **grid** command.

The **save** command is followed by a series of lines which say which average to generate. These end with a line that starts with the word **end**. The following is an example:

```
save averages, file="hii.avr" last no clobber
temperature, hydrogen 1 over volume
ionization, helium 2 over radius
column density oxygen 3
end of averages
```

All reported quantities are the value itself. This command accepts the **_log** option (page 194) to report quantities in the log style used in C13 and before.

16.39.1 Temperature average

The keyword **temperature** begins the line. The average temperature can be weighted with respect to any atom or ion. The name of one of the elements and the ionization stage, 1 for the atom, 2 for the first ion, etc, then follow.

The code computes averages weighted over radius or over volume. If the keyword **volume** appears then the volume-weighted mean temperature will be reported. The default is weighting over radius.

The command works by calling *cdTemp* described in Part 2 of this document.

16.39.2 Ionization average

This reports the average ionization fraction of an atom or ion. The keyword **ionization** begins the line. It is followed by the name of one of the elements, then the ionization stage, 1 for the atom, 2 for the first ion, etc.

The code computes ionization fractions weighted over radius or over volume. If the keyword **volume** appears then the volume-weighted fraction will be reported. The default is weighting over radius.

The hydrogen molecular fraction $2n(H_2)/n(H_{tot})$ can be obtained by asking for ionization stage 0 of the element hydrogen. This is a special case. Other molecular fractions cannot now be obtained.

The **save ionization means** command described on page 218 will save the mean ionization of all elements in the save form as that given at the end of the standard output.

16.39.3 Column density

The column density of any atom or ion is reported. The keyword **column density** begins the line. It is followed by the name of one of the elements, then the ionization stage, 1 for the atom, 2 for the first ion, etc.

16.40 Save chemistry options

16.40.1 Save chemistry rates "filename" species "molecule" [coef]

This command saves rates for molecular reactions involving a specific species. Due to current design constraints, the molecule label must come second at present. These labels are case-sensitive.

By default, all non-catalytic rates will be printed. There are several optional keywords to change this. The keywords **creation** and **destruction** select only those reactions which create or destroy the species, respectively. The keyword **catalytic** selects all reactions for which the specified species acts as a catalyst. Finally, the keyword **all** prints all reactions, including the catalytic ones.

The keyword **coef** will print rate coefficients [$\mathrm{cm^3\,s^{-1}}$, for two-body reactions] instead of rates [$\mathrm{s^{-1}}$].

16.41 Save column density

This command is now one option of the **save species** command, see section 16.86. Please use that command.

The **save H2 column density** command is described elsewhere.

16.42 Save continuum

This command is a primary mechanism for saving the predicted spectrum.

16.42.1 Lines in the spectrum

Emission lines are included in the output for all **save continuum** commands. They are visible in the net emitted spectrum. Labels giving the strongest lines contributing to each wavelength are given in the third to last column in the save output for most versions of the **save continuum** command. More than one line will contribute to many wavelength cells and the last column indicates the number of lines within that cell.

The **print continuum indices** command will list the labels for all lines that enter into each cell. This provides a way to see all lines that contribute. The **print line sort wavelength** can be used to understand the relative contributions when multiple lines contribute at a particular wavelength or energy.

Figure 16.1 shows the incident SED as the smooth red line, while the black line gives the net emission with a warm absorber long the line of sight.

16.42.2 Emission line – continuum contrast

In a real spectrometer the line-to-continuum contrast depends on the spectrometer resolution if a line is unresolved. The higher the resolution, the higher the line will appear in the spectrum. In CLOUDY, lines are unresolved on the coarse continuum mesh that is reported in most versions of

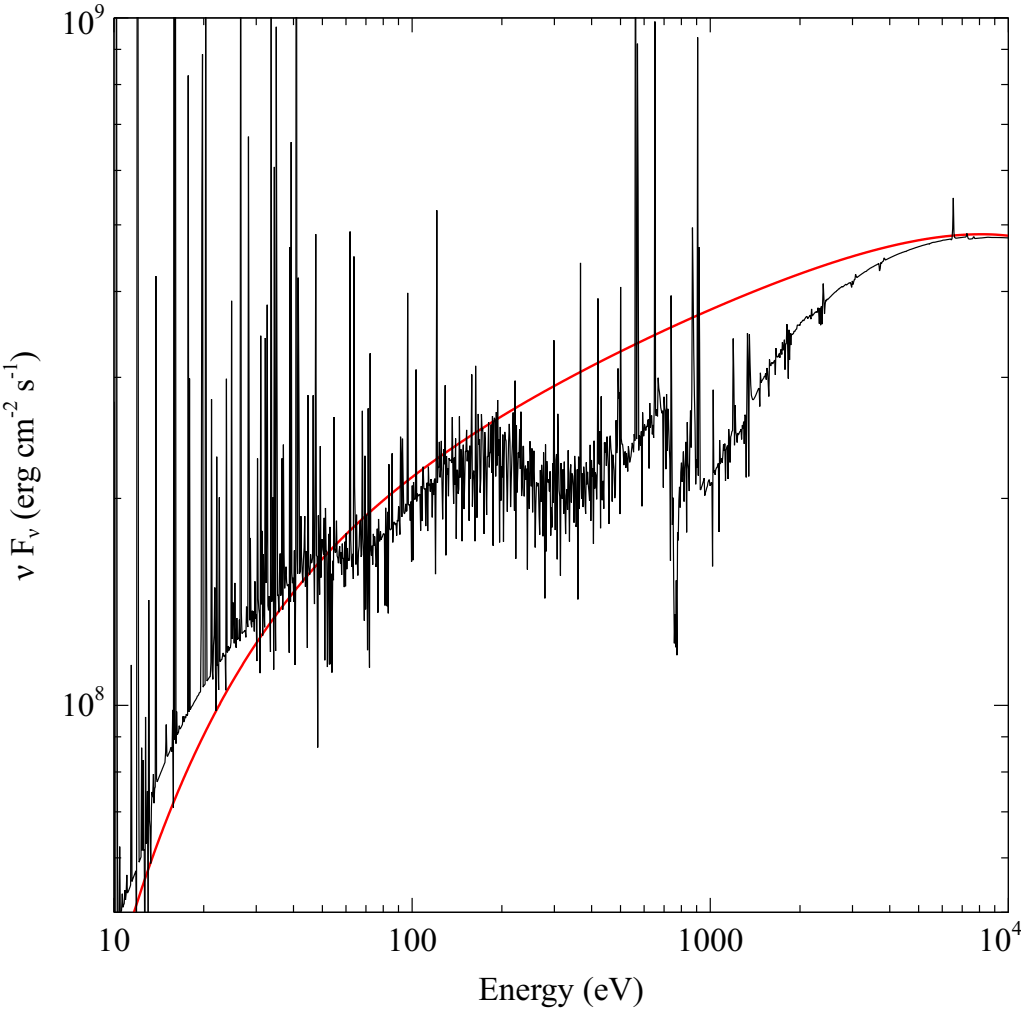

Figure 16.1: The predicted X-ray spectrum of a warm absorber in an Active Galactic Nucleus. Prominent emission and absorption lines are present along with broad UTA absorption features. The parameters are from Reynolds & Fabian (1995, MNRAS, 273, 1167).

the **save continuum** command. CLOUDY adds the emission-line intensities into the continuum, including the resolving power, using Equation 19.2.

When you plot the SED that is reported by a **save continuum** command, emission lines will appear to have a triangular shape (assuming they are not blended with lines in adjacent cells). We will treat this triangular shape as the unresolved line profile. The total flux in that line will then be given by the integral over this line profile, i.e. the area of the triangle. This implies that when you perform an integral over the spectrum (e.g. to do synthetic photometry) the line fluxes will automatically be included correctly in the integral.

Real spectrometers may have significantly higher or lower resolution than CLOUDY's coarse continuum. The **set save line width / resolution** command, described on page 304, changes the lines relative to the continuum to make the spectrum look like that observed with a spectrometer with a resolution that is different from the coarse continuum. Changing the velocity width with the **set save line width** command has the same effect as changing the velocity resolution of a spectrometer measuring an unresolved line. Smaller velocity widths will make the line rise higher above the continuum, as shown in Equation 19.2. Alternatively you can also use the **set save resolution** command to specify the spectral resolution of a spectrograph. Higher values for the resolution will make the line rise higher above the continuum.

Note that this command will *only* adjust the height of the line, not the width. The latter will always be the width of one cell in the coarse continuum mesh (*even if the line is broader than that cell*). This implies that the line flux in the continuum array is artificially changed by the **set save line width / resolution** command. Changing the spectral resolution can be useful to emphasize weak lines in a plot, but should never be used when energy conservation is important, e.g. when you want to postprocess the file to fold the saved continuum with a photometric passband.

16.42.3 Pumped contributions to the lines

Continuum pumping and fluorescence are included as excitation processes for all lines. These contributions are usually not printed as a separate quantity but will be if the **print line pump** command is entered. Whether or not the pumped contribution actually adds to the observed line emission depends on the geometry. Continuum pumping increases the line emission if no related continuum absorption is seen by the observer. This will be the case if the continuum source is either not observed or not covered by absorbing gas on the observer's line of sight. If absorbing gas covers an observed continuum source then the situation is like the P Cygni problem, emission produced by absorption and pumping will not increase the total intensity of the line at all.

The line intensity includes fluorescent excitation unless the **no induced processes** command is entered. That command is unphysical since it turns off all induced processes. You can judge how great the contribution of the pumped part of the line is by printing it with the **print line pump** command.

In general the treatment of scattering is very geometry dependent. The output produced by the **save continuum** commands *does not* include the pumped part of the line contribution. This is correct if the continuum source is included in the beam, but is not if only the gas is observed.

16.42.4 Isotropic Continua

By default the radiation field used in the calculation is reported. The CMB will often dominate certain portions of the spectrum, overwhelming local emission, when it is included. Most spectrometers in the radio and infrared will automatically remove isotropic emission while conducting an observation. The **no isotropic** option provides a way to do a similar subtraction. Each of the commands that set an SED shape, described in Section 6, will say whether that component is isotropic or beamed. This option will remove all sources of isotropic emission.

The commands **save continuum** and **save transmitted continuum** have a **no isotropic** option to not include attenuated isotropic continua in the total. It is also possible to remove isotropic continua with the **no isotropic continua report** command described in Section 19.11.17.

16.42.5 The units option - changing the continuum units

By default, the energy units for the first column, which gives the wavelength or energy for each point in the continuum, are rydberg. The units can be changed to any of several energy or wavelength units with the **units** keyword that appears on a **save continuum** command. The following keywords are recognized: **_micron**, **_eV_**, **_keV**, **_MeV**, **wavenumber**, **centimeter** (also **_cm_**), **_mm_**, **_nm_**, **angstrom**, **_Hz_**,**_kHz**, **_MHz**, **_GHz**, **kelvin** (also **_K_**), **erg_**, and **_rydberg**. Both the keyword **units** and one of these units must appear for the units of the energy scale to be changed.

16.42.6 Save continuum uses vacuum wavelengths

Vacuum wavelengths are always used in **save continuum** output. This is to avoid a discontinuity at 2000 Å, the point where line wavelengths switch from vacuum to air. The **print line vacuum** command, described in Section 16.19.20, controls whether line wavelengths are given in air or vacuum, has no effect on **save continuum** output.

16.42.7 Units of the save output in intensity and luminosity cases

The units of the predicted continuum depend on whether the intensity or luminosity case is used. In the intensity case continua are given as the intensity per octave $4\pi\,\nu J_\nu [\mathrm{erg\,cm^{-2}\,s^{-1}}]$. In the luminosity case they are νL_ν [$\mathrm{erg\,s^{-1}}$]. The emission from the cloud includes a covering factor if one was specified.

Through C13, the **save continuum** output in the luminosity case was per unit cloud area at the inner radius rather than the true luminosity, and so had units $\mathrm{erg\,cm^{-2}\,s^{-1}}$. This was to avoid processor limits on older computers. Beginning in C16 output is in true luminosity units, νL_ν, in the luminosity case. The old behavior will the used if the **set save luminosity old** option, described in Section 19.13.45, is used. This is to provide backwards compatibility.

16.42.8 Save continuum

The **save continuum** command, with no other keywords, produces a file with the following information. The different contributors to the continuum are defined in the Chapter *Definitions*.

Column 1. The first column gives the photon energy in the units set with the **units** option. The default units are rydberg.

Column 2. This is the incident continuum at the illuminated face of the cloud.

Column 3. This is the transmitted incident continuum and does not include diffuse emission from the cloud. The cloud is assumed to fully cover the continuum source along our sight line, so a covering factor has no effect. If the option **no isotropic** has been issued, the transmitted continuum does not include contributions from isotropic incident radiation sources.

Column 4. This is the outward portion of the emitted continuum and line emission. This includes a covering factor if one was specified. This does not include the attenuated or reflected portions of the incident continuum.

Column 5. This gives the net transmitted continuum, the sum of the attenuated incident (column 3) and diffuse (column 4) continua and lines. This would be the observed continuum if the continuum source were viewed through the gas. Emission from the cloud includes a covering factor if one was specified. If the option **no isotropic** has been issued, the net transmitted continuum does not include contributions from isotropic incident radiation sources. This is the same as Column 2 of the **save transmitted continuum** command, described in Section 16.42.22.

Column 6. This is the reflected continuum and is only predicted for an open geometry.

Column 7. This is the total, the sum of the transmitted and reflected continua and lines. The attenuated incident continuum is included. If the option **no isotropic** has been issued, the total continuum does not include contributions from isotropic incident radiation sources.

Column 8. The sum of all reflected line emission.

Column 9. The sum of all outward line emission.

Column 10 and 11. Line and continuum labels indicate the lines and continuum edges that might contribute at that energy. The line label gives the label for the strongest line in the total spectrum (reflected plus outward) the line-center of which lies in that bin. (This is new in C10. All previous versions simply reported the first line encountered, as is still the case with the continuum label.) The continuum labels are established when the code is set up and they do not mean that the continuum feature is actually present in the spectrum.

Column 12. This gives the number of emission lines within that continuum bin, divided by the ratio of the energy width of the cell to the cell's central energy, dE/E. So this is the number of lines per unit relative energy.

This command will save the continuum at the outer edge of the cloud unless the keyword **every** appears on the command line, in which case it will save the continuum for every zone.

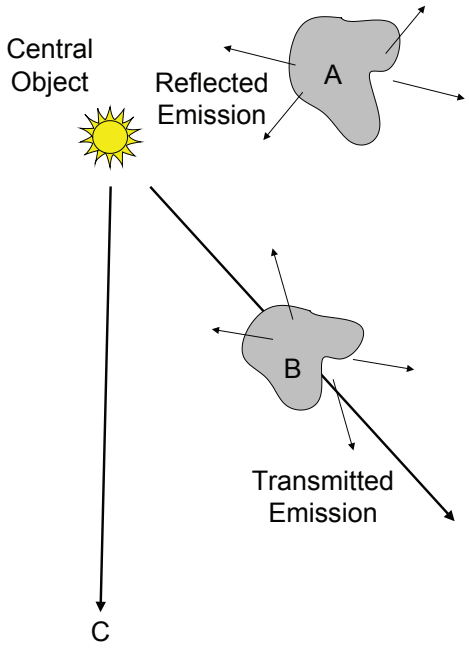

Figure 16.2: This figure illustrates several components of the radiation field that enter in the calculations.

16.42.9 What is observed

Figure 16.2 illustrates several possible geometries. Two lines of sight to the central object are shown, and two clouds are shown. Each cloud produces both a reflected and transmitted emission component.

Three possible geometries, indicated by the letter on the figure, occur depending on how we view the central source and clouds: a) we do not directly observe the central object although we may see it by reflection from the illuminated face of a cloud, b) we observe the transmitted continuum and the outward emission from the emitting cloud, and c) we observe the unattenuated continuum directly without absorption. Column 2 gives the unattenuated continuum, and column 3 gives the attenuated continuum.

There are also three possible situations for the line emission. First, we might only observe clouds that lie on the near side of the continuum source. In this case we see the "outward" line emission. Second, we might only observe clouds that lie on the far side of the continuum source. In this case we only see the "reflected" or inward component. Lastly, we might observe a symmetric geometry with reflected emission from the far side and outward emission from the near side.

In most cases an observer at large distance from the structure would observe *both* the central object and the cloud and would measure the quantity listed in column 5 (if only transmitted

emission is detected) or column 7 (if both reflected and transmitted continua are seen). If the central object is not seen then the quantity in column 4 would be observed.

16.42.10 Save continuum bins

This saves the continuum energy bins. The first column is the center of the bin in whatever unit was set after the **units** keyword (see Section 16.42.5 for details). The second column is also the center of the bin, but this time always in Ryd. The third column is the cell width $\delta\nu$, also in Ryd. The bin extends from $\nu - \delta\nu/2$ to $\nu + \delta\nu/2$.

16.42.11 Save cumulative continuum

This gives the continuum integrated over a time-dependent simulation. The usual **save continuum** command gives the continuum for the current time step in these calculations. The units of the usual **save continuum** command are flux per octave, νf_ν [erg cm^{-2} s^{-1}]. This command gives the time-integrated energy, νE_ν [erg cm^{-2}].

Note that using this command with the option **no isotropic** will cause CLOUDY to exit immediately with an error message.

16.42.12 Save diffuse continuum

This reports the local diffuse line and continuous emission coefficient $4\pi\nu j_\nu$ [erg cm^{-3} s^{-1}]. Optical depth effects are not included.

By default this reports the diffuse emission from the last zone. The first column of the output gives the photon energy including the **units** option. The second column gives the diffuse continuous emission. The third column gives the line emission in the same units, including the effects of the **set save line width / resolution** command described on page 304 when appropriate. The last column gives the total.

If the keyword **zone** appears then the diffuse emission *from every zone* will be reported. The first row gives the wavelength or energy scale. The remaining rows give the total (line and continuum) emission coefficients $4\pi\nu j_\nu$ at each energy.

16.42.13 Save continuum emissivity 12 [units micron]

This command will save the continuum volume emissivity $4\pi\nu j_\nu$ (in erg cm^{-3} s^{-1}), as well as the local absorption and scattering opacity (in cm^{-1}) as a function of radius and depth (in cm). This output can be used by an external program to do more specialized radiative transfer, e.g. to determine the continuum flux through an aperture. To make this process second-order accurate, the radius and depth in the middle of each zone is reported. One number should be supplied on the command line, which is the wavelength / frequency at which the emissivity and opacity will be evaluated. You can use the keyword **units** as described in Section 16.42.5. By default the frequency is assumed to be in rydberg. If a wavelength unit is chosen, the number will always be interpreted as a vacuum wavelength.

16.42.14 Save emitted continuum

The continuum emitted and reflected from the nebula is saved. The first column is the photon energy. The second column is the reflected spectrum. The third column is the outward diffuse emission. The fourth column is the total emission (the sum of the inward and outward emission). This would be the observed emission from the nebula if the central continuum source was not in the beam but clouds uniformly cover the continuum source. The last two columns are labels for lines and continua contributing at each energy. The attenuated incident continuum is not included in any of these components. The effects of a covering factor are included. The continua have units νf_ν [erg cm^{-2} s^{-1}] or νL_ν [erg s^{-1}] depending on whether the intensity or luminosity case is specified.

16.42.15 Save fine continuum [range, merge]

The code transfers the continuum on a coarse mesh, needed for speed in evaluating photo-interaction rates, and on a fine mesh, needed for automatic treatment of line overlap. The coarse continuum is the one given in output from the **save continuum** command and it includes all the components shown in Figure 16.2. The fine continuum is designed to account for line transfer. It shows a normalized attenuated incident continuum and does not include continuous emission or absorption from the cloud. This multi-grid approach is needed to combine precision and speed.

This command reports the energy (see below) of the fine continuum bin in the first column, and the continuum transmission coefficient, $I_{transmitted}/I_{incident}$, for the fine continuum in the second column. What follows is a list of absorption lines centered on that energy bin. Each line is identified by its spectral label (species and wavelength), and is followed by its mean optical depth. The latter is the same as the last entry reported by **print line optical depths**, see Section 16.19.13. All lines centered on that bin are reported, sorted from most-to-least opaque.

The resulting output will be huge if the entire fine continuum is saved. *Users are advised to make use of the* **range** *option to limit the size of the output file.* If the keyword **range** appears then the lower and upper limits to the range of the fine continuum must be entered. The command also accepts a **units** option, described on page 200, to change the units used in the specified energy range, and in the resulting output. If wavelength units are chosen, the limits to the range will be interpreted as vacuum wavelengths.

The optional third numerical parameter gives the number of fine continuum cells to average together, again with the intent of reducing the size of the output file. The default is to average over 10 cells. If the number of cells to be combined is specified then it must be the third number on the command line, following the limits of the range. Note that in a given energy range, the same number of features is reported, irrespective of the number of cells merged.

The resolving power of the fine continuum is adjusted when the calculation begins so that lines of the heavy elements can be resolved. The resolving power is reported in the main printout with the string **R(F Con)**.

16.42.16 Save grain continuum

The thermal emission by grains is part of the emergent continuum. This command saves only the grain emission in the optically thin limit for the last zone. It is mainly intended as a debugging aid.

The first column gives the photon energy in the units specified with the **units** option. The second column gives the total emission from carbonaceous grains (including amorphous carbon and PAH's). The third column gives all emission from all constituents that are not carbonaceous. In practice this will be mainly the silicates. The last column gives the sum of the two.

16.42.17 Save incident continuum

The incident continuum, that emitted by the central object and striking the illuminated face of the cloud, will be saved. The first two columns give the photon energy and the continuum specific intensity $\nu 4\pi J_\nu$ [erg cm^{-2} s^{-1}] or specific luminosity νL_ν [erg s^{-1}]. The photon occupation number

$$\eta_\nu \equiv J_\nu(\nu)\left(\frac{2h\nu^3}{c^2}\right)^{-1} \tag{16.1}$$

concludes the line.

16.42.18 Save interactive continuum

This gives the product of the internal radiation field times the gas opacity integrated over the full continuum. The results are produced for each zone and give the attenuated incident continuum, the OTS line, the OTS continuum, the outward continuum and the outward lines. The first optional number is the lowest energy in the output. If missing or zero, the lowest energy considered by the code will be used. Numbers less than 100 are interpreted as the energy in rydberg and those greater than 100 as the cell number.

16.42.19 Save ionizing continuum [options]

This creates a file that can be used to indicate ionization sources. If the keyword **every** occurs then this is saved for every zone, otherwise it is only saved for the last zone.

The first optional number on the command line is the lowest energy to include in the output. If it is missing or zero then the lowest energy considered by the code will be used. It is interpreted as the energy in rydberg if the number is less than 100 and as the cell number if it is greater than 100. The second optional number is the threshold for the faintest interaction to print. The default is one percent of the quantity given in the `rate/tot` column. Enter zero for this number if you want all interactions to be printed. The optional numbers may be omitted from right to left.

The first column in the resulting output is the cell number of the C scale, so the first cell is zero. Next comes the photon energy in rydberg unless changed with the **units** option. The third column, with the header `flux`, is the flux of photons within this frequency bin (*not* per unit frequency) with units s^{-1} cm^{-2}. The forth column gives the photo-interaction rate, the product of the flux multiplied by the gas absorption cross section [cm^{-2}] and so with units s^{-1}. The next five numbers are the fractions of this photo-interaction rate due to the attenuated incident continuum, the OTS line, the OTS continuum radiation fields, and the outward lines and continua. The 10$^{\text{th}}$ column, with the heading `rate/tot`, is the ratio of this photo rate to the total integrated radiation-field interaction rate. The next column, with the heading `integral`, is the integrated cumulative interaction, the integral of the previous column over energy. This makes it easy to

identify the portions of the radiation field that have the dominant interaction with the gas. The last two labels on the line indicate which lines and continua contribute at that energy.

16.42.20 Save raw continuum

This saves the ***raw*** continua. This is exactly the continuum used within the code. The first number is the photon energy. The next columns are the contents of many of the continuum arrays at each energy. Consult the source to see which arrays are now saved. Each gives the number of photons stored in that cell with units photons s^{-1} cm^{-2} $cell^{-1}$. The last number is the number of lines within that cell.

16.42.21 Save reflected continuum

The reflected continuum is output if **sphere** is not set. The first column is the photon energy, the second the reflected continuum $4\pi \nu J_\nu$ [erg cm^{-2} s^{-1}] at that energy. The third gives the reflected lines and the fourth is the sum of these two. Someone who could only see the illuminated face of the cloud would observe this. The next column is the effective albedo of the cloud, the ratio of the reflected to incident continuum (the **save total opacity** command will give the true albedo). The last column gives the label for continuum processes with thresholds at each energy.

16.42.22 Save transmitted continuum [no isotropic]

This saves the transmitted spectrum (i.e., the attenuated incident SED and the outward component of the diffuse continuum and line emission) predicted at the end of the calculation. It is affected by the **no isotropic** option. The data are the same as Column 5 of the **save continuum** command described in Section 16.42.

This save file can be used as part of the incident continuum in a later calculation by reading it with the **table read** command described in Section 6.12.10.

The code will implicitly behave as if the keyword **last** had been included. This is needed since the code would become confused if it tried to read a file containing information from multiple iterations. So the output would be useless without this keyword. In a grid run it will also implicitly save the continuum into a separate file for each grid point. This is needed for the same reason: the code cannot read a concatenated output file.

Two cautions apply when reading this file as an input continuum. First, save output should not be written into the same file as the input file during the second calculation. The file containing the first continuum will be overwritten if this occurs. Second, the first couple of lines contain header information that is needed by the code. They should not be deleted.

This command is unusual in that it always saves the SED in intensity units, even when the simulation is done in luminosity mode. If the radius is set, the continuum will be an intensity normalized at the outer radius. If the radius is not set, the intensities will have the usual normalization also used in other **save continuum** commands.

The line-to-continuum contrast factor *Resolution* set by the **set save line width / resolution** command (see page 304) is not used in this command. This is to insure that lines have the correct intensity in the save file, as needed for energy conservation. The **units** keyword is also ignored. The output will always use the standard unit (rydberg) as the frequency unit.

16.42.23 Save two photon continuum

This saves the total two-photon continuum and is mainly a debugging aid. The photon energy is followed by the number of photons emitted per rydberg per second, then by νF_ν. The details of two photon emission, including induced processes, are discussed by Bottorff et al. (2006).

16.43 Save convergence [reason, error]

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

16.43.1 Save convergence base

The gives computed values of many quantities that are converged during a calculation. The values are given as they are at the exit to the base convergence routine. Depending on the convergence properties of the solution, there will be between only a few up to many dozens of evaluations in each zone.

16.43.2 Save convergence error

This will produce information concerning the quality of the converged pressure, electron density, and heating-cooling solution. The correct value, converged value, and percentage error, (correct-converged) $\times$ (100/correct), will be produced for each zone.

16.43.3 Save convergence reason

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

16.44 Save cooling

This saves the cooling agents for each zone. The first number is the depth [cm] followed by the temperature [K]. The next two numbers are the total heating and cooling rates [erg cm^{-3} s^{-1}], not including advective terms; advective cooling is reported separately in the fifth column. The remaining labels indicate contributors to the cooling array and the fraction of the total cooling (ignoring advective cooling) carried by that agent. At least four (4) agents are reported, or more if they are "bright" enough. The faintest coolant saved is normally 0.05 of the total (sans advective), and can be reset with the **set WeakHeatCool** command.

This is mainly used for debugging and its output is not well documented. The line labels may, or may not, correspond to labels used in the main emission-line output. If they are, the wavelength reported here will be the vacuum wavelength of the line, irrespective of whether the **print line vacuum** command has been used or not. The code adds individual coolants to the total cooling by calling routine *CoolAdd*. The arguments to that call include the string and a number that follows it. To discover the process for a particular label you will need to search over the entire code to find where that string occurs in a call to *CoolAdd*. The string will be within a pair of double quotes. As

an example, two coolants that might appear are ***dust 0***, and ***H2cX 0***. A search on the string `"H2cX"` (include the beginning and ending double quotes) will find the string occurring in two places, where the cooling is added with the call to *CoolAdd* and another place where the cooling itself is evaluated. Comments around the call to *CoolAdd* indicate that ***H2xC*** is cooling due to collisional excitation within the ground electronic state of H_2. A search on the string `"dust"` finds its call to *CoolAdd* with the comment that it is cooling due to grain recombination cooling.

Cooling due to the unified iso-electronic sequences is different since the label that occurs in the call to *CoolAdd* is generated by the routine that evaluates the cooling and does not explicitly occur in the source. The iso-electronic cooling labels begin with `"IS"`, followed by a string indicating the process, and followed by strings giving the iso-electronic sequence and the element. Search for the start of the string but not for the element name. An example might be the string `"IScionH H"`. The `"IS` indicates that it is cooling due to an ion along the iso-electronic sequences and the two ending H's indicate that the coolant is H-like hydrogen. The string `"IScion"` occurs in the source and comments indicate that the process is collisional ionization cooling. The iso-electronic cooling strings do not include a pair of double quotes in the source since the ending double quote occurs after the element names. You would search for the string `"IScion"`.

16.44.1 Save cooling each

If keyword **each** appear after **save cooling** command, the cooling of each element and for certain special cases will be saved for each zone. The unit of the cooling rate is [erg cm^{-3} s^{-1}]. Notice that the output metal bremsstrahlung cooling is the real metal bremsstrahlung cooling minus the total bremsstrahlung heating. If the bremsstrahlung heating is comparable with the real metal bremsstrahlung cooling, the output may not be precise. The cooling due to molecules is listed separately. The total cooling listed in column 3 does not include the dynamics cooling (adve, column 45).

16.45 Save charge transfer

Charge transfer recombination and ionization rate coefficients [$cm^3 s^{-1}$] for hydrogen onto heavier elements will be output. The rates will be evaluated at the temperature of the last computed zone. Rates for recombination ($A^{+x} + H \rightarrow A^{+x+1} + H^+$) are first, followed by the rates for the opposite ionization process. The first number is the atomic number of the species.

16.46 Save Chianti

This saves collision strengths for Chianti transitions over a certain temperature range. The output gives species, lower level, upper level, vacuum wavelength in angstrom, and collision strengths for each temperature given in the header row. This reports data for all active Chianti species in that particular run.

16.47 Save dr

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

16.48 Save dt

The logic behind the choice of time step in time-dependent simulations will be described.

16.49 Save dynamics options

This produces some information concerning the dynamics solutions.

16.49.1 Save dynamics advection

This produces information about advection terms.

16.50 Save element *name*

This will save the ionization structure of any element. The output will have one line per zone giving the ion fraction[8] of each successive stage of ionization. The keyword **element** must be followed by the element name.

The first number on the resulting output is the depth [cm] into the cloud. The remaining lines give the relative ionization fraction of the $n+1$ possible stages of ionization where n is the atomic number of the element. These may be followed by some of the more abundance molecules.

If the keyword **density** appears on the command line then the density [cm^{-3}] will be given rather than the dimensionless fraction.

16.51 Save enthalpy

The file will list the depth into the cloud, followed by the total enthalpy, and various contributors to it.

16.52 Save execution times

The code will output the zone number, the time required to compute that zone, and the elapsed time since the start of the calculation. This is intended as a mechanism to identify zones that require large amounts of work to converge. This command is being superseded by the **save performance** command.

[8] Before version 96 the ionization fractions only included atoms and ions. They now include molecules. The sum of the atomic and ionic fractions will not add up to unity if a significant fraction of the element is in molecules.

16.53 Save FeII

These commands are obsolete. See the **save species** family of commands, Section 16.86.

16.54 Save FITS

This command produces photon energies and the total transmitted continuum into a two-column file in FITS (Flexible Image Transport System) format. FITS is a standard format used in astronomy, and endorsed by both NASA and the International Astronomical Union. The most current definition of the FITS format is by Hanisch et al. (2001). This was added by Ryan Porter and its first application is given in Porter et al. (2006). The saved spectrum will by default include the isotropic continua. They can be excluded using the **no isotropic continua report** command. The saved spectrum will always conserve energy, i.e. the **set save resolving power** command has no effect.

Since only one spectrum can be stored in a FITS file, the command will implicitly behave as if the keyword **last** was included on the command line.

The following creates a FITS file with a blackbody continuum

```
blackbody 1e5 K
ionization parameter -2
save FITS "filename"
```

This command is often used with the **grid** command which creates grids of calculations. In this case, the command produces separate FITS files for each grid point. They will have names `grid000000000_filename`, `grid000000001_filename`, etc., where `filename` is the file name you supplied between double quotes. The meaning of the grid index embedded in the filename can be found with the **save grid** command described in Section 16.58.

16.55 Save gammas

This gives the photoionization rates for all subshells of all ions for the last zone. The numbers are the element, ion, and subshell numbers, followed by the photoionization and heating rates from that subshell. The remaining numbers are the fractional electron Auger yields.

16.55.1 Save gammas element oxygen 1

If the **element** keyword appears then the detailed contributors to the photoionization rate for the valence shell of a particular element will be produced by calling *GammaPrt*. The ionization stage must also appear, with 1 the atom, 2 the first ion, etc.

16.56 Save Gaunt factors

This produces a series of tables showing the thermally averaged free-free Gaunt factors as a function of the log of the photon energy (vertical, in Ryd) and the log of the electron temperature

(horizontal, in K). Such a table is produced for each charge state treated by CLOUDY. Below each table, the Gaunt factors are repeated as triplets, but this time as a function of the customary dimensionless units $\log \gamma^2$ and $\log u$ rather than temperature and photon energy.

16.57 Save grains [options]

These commands give grain properties. Several grain species are usually included in a calculation. Often there are several size bins per grain type. These commands will print a line giving a list of the grain labels, followed by a line giving the grain radius in μm. The following lines then give the individual grain properties (temperature, potential, etc) for each size and type.

Details of the grain physics are given in Baldwin et al. (1991), Weingartner and Draine (2001a), van Hoof et al. (2004), and Weingartner et al. (2006).

16.57.1 Save grain abundance

The grain abundance [g cm^{-3}] for each grain bin and the total grain abundance is given as a function of depth.

16.57.2 Save grain charge

The first number is the electron density [cm^{-3}] contributed by grains. A positive number indicates that grains where a net source of electrons so they have a net positive charge. The remaining columns print the charge of each grain size and type, in number of elementary charges per grain, for each zone.

16.57.3 Save grain continuum

This is listed in the *save continuum* section.

16.57.4 Save grain D/G ratio

The dimensionless dust to gas ratio for each grain bin and the total grain abundance is given as a function of depth.

16.57.5 Save grain drift velocity

The drift velocity [km s^{-1}] of each grain species is printed for each zone.

16.57.6 Save grain extinction

The grain extinction at the V filter will be saved as a function of depth. This extinction only includes grains, although they should provide nearly all the extinction when they are present. The first column gives the depth into the cloud [cm], the second is the extinction [mag] at the V filter for an extended source, like a PDR, and the third number is the extinction for a point source like a

star. This distinction is discussed in Sections 7.2 and 7.6 of AGN3. The extended source extinction discounts forward scattering by writing the scattering opacity as $\sigma(1-g)$, where σ is the total scattering opacity and g is the grain asymmetry factor. The quantity in the last column does not include the $(1-g)$ term.

16.57.7 Save grain H2rate

The grain H_2 formation rate is given for each size and type of grain and for each zone.

16.57.8 Save grain heating

The grain heating [erg $cm^{-3}s^{-1}$] is output for each zone.

16.57.9 Save grain opacity

This gives the grain opacity as a function of the photon energy in the last zone of the model. The first column is the photon energy in the units set with the **units** option, the second the total (absorption plus scattering) cross section, followed by the absorption and scattering cross sections. These are the summed cross section per proton for all grain species in the calculation. Column 5 is the pure scattering cross section summed over all grains and column 6 is the grain albedo. Columns 2, 4, and 6 use the scattering cross section multiplied by $(1-g)$, while column 5 does not. The opacities have units [cm^2 H^{-1}] using the actual grain abundance from the last zone.

16.57.10 Save grain potential

The grain floating potential [eV] is output for each grain size bin and each zone. The grain potential is defined as

$$\varphi_g = \frac{(\langle Z\rangle + 1)\, e^2}{a} \tag{16.2}$$

where $\langle Z\rangle$ is the average grain charge of the grain size bin. In reality there is a distribution of grain charges in each bin, so this quantity does not relate to any individual grain. It is approximately the amount of energy needed to move an electron from the grain surface (after it has been lifted out of the grain potential well) to infinity, averaged over all charge states. Weingartner and Draine (2001a), van Hoof et al. (2004) and Weingartner et al. (2006) provide further details.

The grain potential given here and the (average) grain charge (see **save grain charge**) differ only by constants. It is probably best to use the latter since that quantity will be far easier to define. Refer to Weingartner and Draine (2001a) and van Hoof et al. (2004) for more details.

16.57.11 Save grain qs

The photon energy in rydberg is followed by the absorption and scattering efficiency for each grain species. The absorption efficiency $Q_{\rm abs}$ is defined as the ratio of the absorption cross section $\sigma_{\rm abs}$ and the geometric projected grain surface area $\sigma_{\rm g} = \pi a^2$. This yields: $Q_{\rm abs} = \sigma_{\rm abs}/\sigma_{\rm g}$. A similar definition holds for the scattering efficiency, keeping in mind that the scattering cross

section has been multiplied by $(1-g)$, i.e., we print: $Q_{\rm scat}(1-g) = \sigma_{\rm scat}(1-g)/\sigma_{\rm g}$. See also Chapter 7 of AGN3.

16.57.12 Save grain temperature

The temperature of each grain species is printed for each zone.

16.58 Save grid

This is used with the **grid** command to create a file giving the parameters for each simulation in a large grid of calculations. This can be combined with other **save** commands to create files giving the derived quantities for a large set of calculations.

The first row gives tab-separated column titles. The column title for the parameters that are varied will give the first few characters of the command line for that parameter. Each of the following lines summarizes a particular grid point. The first column gives the index number in the grid. This is useful if you have separate output for each grid point (e.g. from the **save FITS** command). The index number will be part of the file name in that case. It is also useful for finding back the output from a particular grid point in a save file (see Section 18.6 for more details). The next two columns indicate whether the calculation had a failure or had any warnings. A successful calculation will have "F" (for false) indicating no failure or warning occurred. If either of these is "T" (for true) then that simulation had major problems. The following column is a string giving a bit more detail about the exit code. It says "ok" if everything went fine and "warnings" if warnings were present in the run, but nothing more serious. If the simulation failed completely, you could see strings like "cloudy abort" or "early termination". You should look at the detailed output to find out what went wrong. The next two columns give the number of the MPI rank that executed the simulation (will be zero in non-MPI runs) and the sequence number of the simulation on that rank (they are executed in random order). These numbers are used for debugging the code. The next columns give numerical parameters for each simulation in the grid. The last column gives all the grid parameters as a single string. Many plotting programs will allow you to add this string next to each of the plotted data points.

This command does not produce any output if the **grid** command is not used. In grid runs it is highly recommended to always include this command. It allows you to see in one glance if the run was successful and where the problems are in case it was not. It is also useful to include this output in case you want to file a bug report.

16.59 Save heating

The code will save the heating agents for each zone. The first two columns give the depth [cm] and temperature [K]. The next two columns give the total heating and cooling rates [erg cm^{-3} s^{-1}], not including advective terms; advective heating is reported separately in the fifth column. This is followed by a set of labels for various heat sources and the fraction of the total heating (ignoring advective heating) carried by that agent. At least four (4) agents are reported, or more if they are "bright" enough. The faintest agent saved is normally 0.05 of the total (sans advective), and can be reset with the **set WeakHeatCool** command.

The heating labels will probably not correspond to any entries in the emission-line list. If the identification of a heat source is in question the best recourse is to search for the heating label over the entire code.

16.60 Save [option] history

16.60.1 Save pressure history

This follows the pressure and density convergence history.

16.60.2 Save temperature history

This follows the temperature and heating—cooling convergence history.

16.61 Save H2

Some details of the large H_2 molecule are output as a save file. One of the following options must appear. The large H_2 model is not enabled by default but is turned on with the **database H2** command.

Many studies of H_2 properties will use plots showing populations as a function of excitation energy. The slope gives a population temperature, which may be related to the kinetic temperature under some circumstances. Theoretical plots can be made using the **save H2 column density** or **save H2 populations** commands described below. The level energy in K, given in the save output, is the x-axis and the column density or population in each level is the y-axis.

The **save grain H2rate** command gives the H_2 formation rate for each of the grain species included in the calculation.

16.61.1 Save H2 column density

This saves the column density of ro-vibrational states within the ground electronic state. This command recognizes the same options as the **save H2 populations** command.

The file begins with the total H_2 column densities in the ortho and para forms, followed by the total H_2 column density. The remainder of the file gives the v and J quantum indices, followed by the excitation energy of the level in K, the total column density in that level, and finally the column density divided by the statistical weight of the level.

16.61.2 Save H2 cooling

This produces a file containing heating and cooling rates [cm^{-3} s^{-1}] as a function of depth.

16.61.3 Save H2 cooling per molecule

This produces a file containing the LTE cooling (for transitions within the X band only), and the net H2 cooling (cooling minus heating) as a function of temperature. Each measure of cooling is reported per H2 molecule in units of $\mathrm{erg\,s^{-1}}$.

16.61.4 Save H2 creation

This command has been superseded by the **save chemistry rates** command.

16.61.5 Save H2 destruction

This command has been superseded by the **save chemistry rates** command.

16.61.6 Save H2 heating

This produces a file containing the depth, total heating rate [erg $\mathrm{cm^{-3}s^{-1}}$], and H_2 heating predicted by an expression in Tielens and Hollenbach (1985a), together with the heating predicted by the large model of the H_2 molecule.

16.61.7 Save H2 levels

This creates an energy-ordered list of the levels within X. The first column gives the energy in wavenumbers relative to the lowest level, the second gives the statistical weight, and the next two columns give the vibration and rotation quantum numbers. The sum of the transition probabilities is given, followed by the sum of collisional de-excitation rate coefficients ($\mathrm{cm^3\ s^{-1}}$) of the level for each of the colliders included in the calculation.

16.61.8 Save H2 lines

The intensities of all significant H_2 emission lines that are produced within the ground electronic configuration are given. The file also gives the information needed to convert an emission-line spectrum into a level population vs excitation energy diagram.

Each line of output begins with a spectroscopic designation of the line, followed by the upper and lower electronic, vibration, and rotation quantum numbers. This is followed by the wavelength of the line in angstrom. Next the line wavelength is printed as it appears in the output. The log of the intensity or luminosity in the line (depending on whether the intensity or luminosity case is specified), and the intensity of the line relative to the normalization line, follow. The excitation energy of the upper level of the transition in Kelvin and the product $g_u h\nu A_{ul}$ for each line are also given. This product is needed to convert intensities into excited-state column densities.

Only lines brighter than 10^{-4} of the reference line are given by default. The intensity of the faintest line, relative to the normalization line, is set with the first optional number that can appear on the command line. If the number is negative then it is interpreted as the log of the limit.

By default only lines within the ground electronic system are saved but this can be changed if the keyword **electronic** appears on the command line. If the keyword **all** also appears then lines

within all electronic systems are saved. If the keyword **ground** appears then only lines the ground system are saved. This is the default. If a number appears then it is the number of electronic systems to be saved, 1 for only the ground electronic state. If the number of systems is specified then it must be the second number on the line—the first being the intensity of the faintest line to print.

The following give examples. The first sets the faintest H_2 line to save. Only lines within the ground electronic system will be produced.

```
save H2 lines, faintest -4 "filename"
```

This sets the faintest line, and also requests all lines produced by all electronic configurations.

```
save H2 lines, faintest -4 all "filename"
```

This example requests only lines for the lowest three electronic configurations.

```
save H2 lines, faintest -4 electronic 3 "filename"
```

16.61.9 Save H2 populations

The level populations for the ground electronic state are given for the last computed zone. The populations are relative to the total H_2 abundance.

There are several optional parameters. The quantum numbers of the highest vibrational and rotational levels to save can be specified as consecutive numbers on the command line. These occur in the order v then J. If no numbers occur, or if a limit that is less then or equal to zero is entered, then populations of all levels will be saved.

If the keyword **zone** appears then the level populations will be saved for every zone. These are all given along one long line, a format that is different from the other output options. Otherwise the populations are only saved at the end of the iteration using the output options described next.

If output is only saved for the last zone then it can be in either a triplet format, with the vibration and rotation quantum numbers followed by the population, or as a matrix, with all populations of a given vibration quantum number lying along a single row. The triplet form is done by default, and the second will occur if the keyword **matrix** occurs on the command line.

The **save H2 column densities** command provides similar output options but for integrated column densities.

16.61.10 Save H2 PDR

The output contains six columns giving the depth (in cm), the o-H_2 density, the p-H_2 density (both in cm^{-3}), and the Solomon dissociation rate following Tielens and Hollenbach (1985a), Bertoldi and Draine (1996), and the CLOUDY big H_2 model, respectively (in s^{-1}).

16.61.11 Save H2 rates

The output contains useful information regarding H_2 formation and destruction rates.

16.61.12 Save H2 Solomon

The output will give the total rate for photo-excitation from the ground to electronic excited states and then identify those levels which are the dominant contributor to the total rate.

16.61.13 Save H2 special

This provides infrastructure to save debugging information. It can be easily changed to suite particular needs.

16.61.14 Save H2 temperatures

The depth [cm], 21 cm spin temperature [K], gas kinetic temperature [K], and several temperatures derived from relative populations of J levels within the H_2 $v = 0$ ground electronic state [K], are saved for each zone.

16.61.15 Save H2 thermal

A variety of heating—cooling processes involving H_2 are given for each zone.

16.62 Save helium [options..]

16.62.1 Save helium line wavelengths

This gives wavelengths of lines from $n = 2$ levels of He-like ions.

16.63 Save hydrogen

16.63.1 Save hydrogen 21 cm

This gives some information related to the spin temperature of the 21 cm line. The keyword is either **21 cm** or **21cm** and does not depend on whether air or vacuum wavelengths are used. The level populations within 1s are determined including radiative excitation by Lα, pumping by the external and diffuse continua, collisions, and radiative decay. Several of the resulting populations and temperatures are included in the output.

16.63.2 Save hydrogen conditions

This gives the physical conditions and populations of various forms of hydrogen are given as a function of depth. The depth [cm], temperature [K], hydrogen density [cm^{-3}], and electron density [cm^{-3}] are followed by the densities of H^0, H^+, H_2, H_{2^+}, H_{3^+}, and H^- relative to the total hydrogen density.

16.63.3 Save hydrogen ionization

This gives rates for processes affecting the hydrogen ionization as a function of depth. The columns give the ground state photoionization rate [s^{-1}], the total and Case B recombination coefficients [cm^3 s^{-1}], the predicted ratio of $n(H^+)$ to $n(H^0)$, and the theoretical ratio for the simple case. Finally, contributors to the ground state photoionization rate are produced with a call to *GammaPrt*.

16.63.4 Save hydrogen lines [alpha]

The upper and lower quantum indices, the line wavelength, the optical depth, and the intensity of the emitted line, will be saved. If the **alpha** option is used, only α transitions ($n \rightarrow n-1$) will be saved.

16.63.5 Save hydrogen Lya

This produces some information related to the excitation, destruction, and escape of the Lα line of hydrogen. The optical depth from the illuminated face to the outer edge of the current zone is followed by the excitation temperature, electron kinetic temperature, and the ratio of these.

16.63.6 Save hydrogen populations

This gives the depth [cm], the densities of H^0 and H^+ [cm^{-3}], followed by the level population densities [cm^{-3}] for the levels in the order $n = 1, 2s, 2p, 3, 4, 5, \ldots$ for each zone.

16.64 Save ionization means

The mean[9] ionization of all elements included in the calculation will be output. The format is exactly the same as the mean ionization printout produced at the end of the standard output.

The **save averages** command described on page 196 can save the average ionization fraction for a particular species.

16.65 Save ionization rates carbon

The total ionization and recombination rates for a specified element will be saved as a function of depth. The name of an element must appear on the command line. Each line of output will have the depth [cm], electron density [cm^{-3}], and the sink timescale for loss of particles due to advection out of the region [s^{-1}]. The remaining numbers give quantities for each possible stage of ionization of the element. For each ionization stage the set of numbers that are printed give the density of atoms in that ionization stage [cm^{-3}], the total ionization rate [s^{-1}], the total recombination rate [s^{-1}], and the rate new atoms are advected into the region [s^{-1}].

[9] Before version 96 the ionization fractions only included atoms and ions. They now also include molecules. The sum of the atomic and ionic fractions will not add up to unity if a significant fraction of the element is in molecular form.

16.66 Save ip

This gives the ionization potentials of all shells of all ions and atoms of the 30 elements included in the code. The first row is the spectroscopic designation of the ion. Each additional row gives the subshell and ionization potential of that subshell in eV.

16.67 Save Leiden

This command produces an output file designed for the comparison calculations presented in the 2004 Leiden meeting on PDR calculations (Röllig et al., 2007).

16.68 Save lines, options

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

16.68.1 Save lines, array

This gives the intensities of all lines in a form in which they can easily be plotted by other software. The first column lists the line energy in rydberg. The next gives the spectroscopic designation of the ion. The following two columns give the log of the intrinsic and emergent intensities or luminosities of the line. These will be in the same units as the main printout and are the absolute intensities, luminosities, surface brightness, etc and not the relative intensities. Only lines with non-zero intensity are included. All lines that appear in the CLOUDY main output will also appear in this save file.[10]

Line energies are in rydberg by default. This command recognizes the **units** option. Note that when wavelength units are chosen, e.g., **angstrom**, the wavelengths will always be printed in vacuum regardless of whether the **print line vacuum** command is used or not. This makes the behavior compatible with the **save continuum** command.

16.68.2 Save lines, zone cumulative

This option tells the code to save the cumulative intensity of up to 100 emission lines as a function of depth into the cloud. The emission lines are specified on the following input lines and end with a line with the keyword **end** in columns 1–3. The format for specifying the spectral lines is described in Section 2.5.3.

The code distinguishes between intrinsic and emergent line intensities. This command reports the intrinsic line intensity by default. If the keyword **emergent** appears then it will report the emergent line intensity. The keyword "zone" distinguishes the spatial integration performed here from the time integration performed in time-dependent problems.

The line labels and wavelengths are followed by the depth into the cloud and the integrated intensities of the lines [erg cm^{-2} s^{-1}] for each zone. This information can be used to follow the build up of emission lines across a computed structure.

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

The following illustrates its use;

```
save lines, zone cumulative, "lines.cum"
H  1 4861.36A
o  3 5006.84A
blnd 3727.00A
o  1 6300.30A
end of lines
```

If the optional keyword **relative** is specified then the quantities will be given relative to the normalization line. The default is to give the intensity $4\pi J$ [erg cm^{-2} s^{-1}].

The **save lines zone cumulative** and **save lines emissivity** commands can now be used in the same calculation.

16.68.3 Save line data

This saves some atomic data for all lines included in the line transfer arrays. It can be used to generate a table listing many lines and their atomic parameters. The code will stop after the data have been saved since it is left in a disturbed state.

The first set of lines consists of recombination lines from Nussbaumer and Storey (1984) and Pequignot et al. (1991). For these the spectroscopic designation and wavelength are given followed by the log of the recombination rate coefficient.

The remaining sets of lines are those that are treated with full radiative transfer. The first are the ***level 2*** lines, which use Opacity Project wavelengths and various g-bar approximations to generate approximate collision strengths. These are followed by the hydrogen and helium iso-electronic sequences, then the ^{12}CO and ^{13}CO lines. The H_2 lines come last if this large model is included.

By default the atomic parameters will be evaluated at a temperature of 10^4 K. Other temperatures can be selected by entering a **constant temperature** command. The number of H-like, He-like, CO, and H_2 lines that are printed is controlled by the size of the relevant atoms when the **save line data** command is executed.

This command recognizes the **units** option. The line can be given in any of the wavelength or energy units it understands. Note that if you use the **units** option and choose a wavelength type unit, e.g., **micron**, then the saved wavelengths will always be in vacuum, regardless of what is used in the main output. If you do not use the **units** option, wavelengths will be saved in the same format as in the main output, which means they can be either in air or vacuum, depending whether you use the **print line vacuum** command.

16.68.4 Save lines, emissivity

This saves the emissivity of an arbitrary number of lines as a function of depth into the cloud. This information can then be used by other codes to reconstruct the surface brightness distribution of a resolved emission-line object. The ***emissivity*** is the net emission $4\pi\bar{J} = n_u A_{ul} P_{ul} h\nu$ [erg cm^{-3} s^{-1}] produced at a point and escaping the cloud. This includes the escape probability P_{ul}.

The code distinguishes between intrinsic and emergent line intensities. This command reports the intrinsic line intensity by default. If the keyword **emergent** appears then it will report the emergent line intensity.

The emission lines are specified on the input lines that follow the command and end with a line with the keyword **end** in columns 1–3. The format for specifying spectral lines is described in Section 2.5.3.

The save output begins with emission-line labels and wavelengths. The remaining output gives the emission structure. The first column is the depth [cm] into the cloud. The remaining columns give the volume emissivity [erg cm^{-3} s^{-1}] for each line. The intensity is for a fully filled volume so the saved intensity should be multiplied by the filling factor to compare with observations of a clumpy medium.

The following illustrates its use;

```
save lines, emissivity, "lines.str"
H  1 4861.36A
o  3 5006.84A
blnd 3727.00A
o  1 6300.30A
end of lines
```

The **save lines cumulative** and **save lines emissivity** commands can now be used in the same calculation.

16.68.5 Save lines, intensity [every 5 zones]

This gives the intrinsic or emergent intensities of all lines with intensities greater than zero in the format used for the final printout (line label, wavelength, intensity).

The default is to give the intrinsic intensities. If the keyword **emergent** appears then those will be given instead.

The default is for this to be done only after the last zone is computed. Results for intermediate zones can be saved if the additional keyword **every** appears. In this case the first number on the line is the interval between zones to save, as in the **print every** command.

The save output will have the line information spread over 6 columns. For some data base applications it would be better to have a single column of results. If the keyword **column** appears then a single column is produced.

Lines with non-zero intensities are reported by default. If the keyword **all** appears then all lines are reported.

16.68.6 Save line labels [long] [no index]

This creates a file listing all emission-line labels and wavelengths in the same format as they appear in the main output's emission-line list. This is a useful way to obtain a list of lines to use when looking for a specific line. The file is tab-delimited, with the first column giving the line's index within the large stack of emission lines, the second giving the character string that identifies the line in the output (the line label), and the third giving the line's wavelength in any of several units. The line ends with a comment that describes the line in more detail. The format of the comment varies from line to line, but in general tries to give a spectroscopic identification of the line. The necessary information is not always available though.

There are a vast number of emission lines predicted by the code and many lines will have the same wavelength. The line label can usually be used to distinguish between various lines with the

same wavelength. This is seldom the case for contributions to the line however. For many lines the code supports further disambiguation if you supply the level indices of the lower and upper level, or the energy of the lower level. If this is supported for the line in question, the save output will contain the necessary information. See Secion 2.5.3 for further details.

The line index can also be used to resolve this degeneracy in cases where you want to obtain a line's intensity with a call to a routine. Routine *cdLine_ip* (described in the header file `cddrive.h` and also in Part 2 of this document) uses the line index to find the relative intensity and luminosity of a particular line. But note that this index is not necessarily the same in different calculations. It will always be the same for a particular set of input conditions but it depends on the sizes of various atoms and which chemical elements are used in the calculation.

By default only the total intensity of a line is reported, so a part of the output might appear as

```
3653  H  1      6562.81A    # index=2, 5 Elow=82259.08   H-like,   2^2S - n=  3
3658  H  1      4861.33A    # index=2, 8 Elow=82259.08   H-like,   2^2S - n=  4
```

Most emission lines share a common structure which contains a great deal of information about line details. The keyword **long** will include these details in the report. In this case the report would include

```
3653  H  1      6562.81A    # index=2, 5 Elow=82259.08   H-like,   2^2S - n=  3
3654  Inwd      6562.81A    # H-like,   2^2S - n=  3
3655  Coll      6562.81A    # H-like,   2^2S - n=  3
3656  Pump      6562.81A    # H-like,   2^2S - n=  3
3657  Heat      6562.81A    # H-like,   2^2S - n=  3
3658  H  1      4861.33A    # index=2, 8 Elow=82259.08   H-like,   2^2S - n=  4
3659  Inwd      4861.33A    # H-like,   2^2S - n=  4
3660  Coll      4861.33A    # H-like,   2^2S - n=  4
3661  Pump      4861.33A    # H-like,   2^2S - n=  4
3662  Heat      4861.33A    # H-like,   2^2S - n=  4
```

The entries marked "Inwd", "Coll", "Pump", and "Heat" indicate the inward fraction, the collisional and radiative pumped contribution to the line, and the line heating.

The command also accepts the **no index** keywords, which can be used to suppress the line index in the output. This can be useful when comparing the line stacks for two versions of an atomic database (e.g., Chianti), as spectral lines whose wavelength remains unchanged may appear distinct to a differencing tool (e.g., `sdiff`) on account of the indices.

This output includes some transitions that are completely forbidden by radiative selection rules. These transitions can still have collisional rates and so contribute to the heating or cooling. You can judge whether a transition is truly forbidden by looking at the atomic data reported with the **save line data** command, described in Section 16.68.3.

16.68.7 Save line list [absolute]

This reads in a list of emission lines from a file and reports the predicted line intensities. It is designed as a way to obtain predictions for a subset of the lines that are predicted during a series of calculations. It is often used together with the **grid** command when doing grids of calculations or with the **time** command when following the evolution of a time-varying continuum source.

Filenames

These are tricky since two filenames appear in this command. All **save** commands have the name of an output file in double quotes. This is the first filename on the command line. The file containing the list of emission lines is within the second pair of quotes. In the following example

```
save line list "output.txt"  "LineListHII.dat"
```

the save output will go to `output.txt` and `LineListHII.dat` contains the set of emission lines. Predicted intensities for the list of lines contained in the second file will be output into the first file.

This command works by calling routine *cdGetLineList* which is described in Part 2 of this document. The filename within the second pair of quotes is sent to this routine. You can build your own file. The format is described below.

Line List data format

Several sample files giving lists of emission lines are included in the data directory. They have names that are of the form `LineList*.dat` and are intended for different circumstances. Any line that starts with a comment character ("#") in the first column is ignored. Lines containing data start with the line label in the first column, which is followed by the wavelength in the usual CLOUDY format and optionally either the level indices or the energy of the lower level to disambiguate the line (see Section 2.5.3 for details). Nothing else may appear on the line except a comment at the end, also starting with a comment character. The following is an example.

```
# the [O III] green line
O  3  5006.84
O  3  5008.24 vacuum # same as above, but using an explicit vacuum wavelength
Blnd  5007A # Blend: "O  3  5006.84A"+"O  3  4958.91A"+"O  3  4931.23A"
H  1  4.65247m  Elow=105291.6 # Is it 7->5 or 35->7? Use Elow to disambiguate
"Fe 2b" 7445
```

You can use the field of stars ("***", i.e. a minimum of three consecutive stars starting in the first column), or a blank line as an end-of-file sentinel. Any input after such a line will be ignored.

Output format

By default results are presented as rows of emission lines. The output file will have the list of emission-line labels in the first row. The first column gives the iteration number and the remaining entries across the row give the line predictions. The **column** option will produces columns instead.

Units of the line brightness

Intrinsic line intensities are given by default. The keyword **emergent** will give those instead. The lines will be relative to the reference line by default. If the keyword **absolute** appears then they will be given in absolute units, the same units as the third column in the main emission-line output. By default these are $\mathrm{erg\,cm^{-2}\,s^{-1}}$ for the intensity case and $\mathrm{erg\,s^{-1}}$ for the luminosity case. The **print line surface brightness** command, described on page 189, can change the absolute units for all the lines to surface brightness, either $\mathrm{sr^{-1}}$ or $\mathrm{arcsec^{-2}}$.

The ratio option

If the keyword **ratio** appears then the ratio of adjacent lines will be output. There must be an even number of lines in the line-list file. The output will have the ratio of the intensity of the first divided by the second, the third divided by the fourth, etc. This provides a quick way to look at line ratios as a function of other parameters. The **grid** command can produce grids of calculations.

Suppose the file `linelist.dat` contains the following:

```
#
# the [O III] temperature indicator
o  3     5006.84
o  3     4363.21
```

The command

```
save line list "o3.lin" ratios from "linelist.dat"
```

would report the ratio of the [O III] λ5006.84 to the λ4363.21 line.

The following input script uses the **grid** command to predict the [O III] line ratio as a function of n and T.

```
# produce the save output
save line list "O3.lin" "linelist.dat" ratio no hash
save grid "O3.grd"
# following three commands do H and O-only, and set O
# ionization so that little O+3 is present (to prevent
# formation of 4363 by recombination).
init ``honly.ini''
element oxygen on
element oxygen ionization 1 1 1 0.01
# set the continuum
blackbody 4e4 K
ionization parameter -2
# next 4 commands vary Te and n
constant temperature 4 vary
grid 4000 17000 3000
hden 4 vary
grid 2 6.1 1
# must stop this constant Te model
stop zone 1
```

This produces two output files, `O3.grd` containing the densities and temperatures, and `O3.lin`, containing the ratio I(5006.84) / I(4363.21) for each density and temperature. These could be plotted to show this line ratio as a function of density and temperature, as shown in the Figure 16.3.

16.68.8 Save line optical depths, limit=-2

This gives the mean optical depths for all lines. By default all lines with optical depths greater than 0.1 will be included. The lower limit can be reset with the optional number that appears on the line—it is the log of the smallest optical depth to be printed (including line overlap). This command recognizes the **units** option so the line wavelengths can be in any of the wavelength or

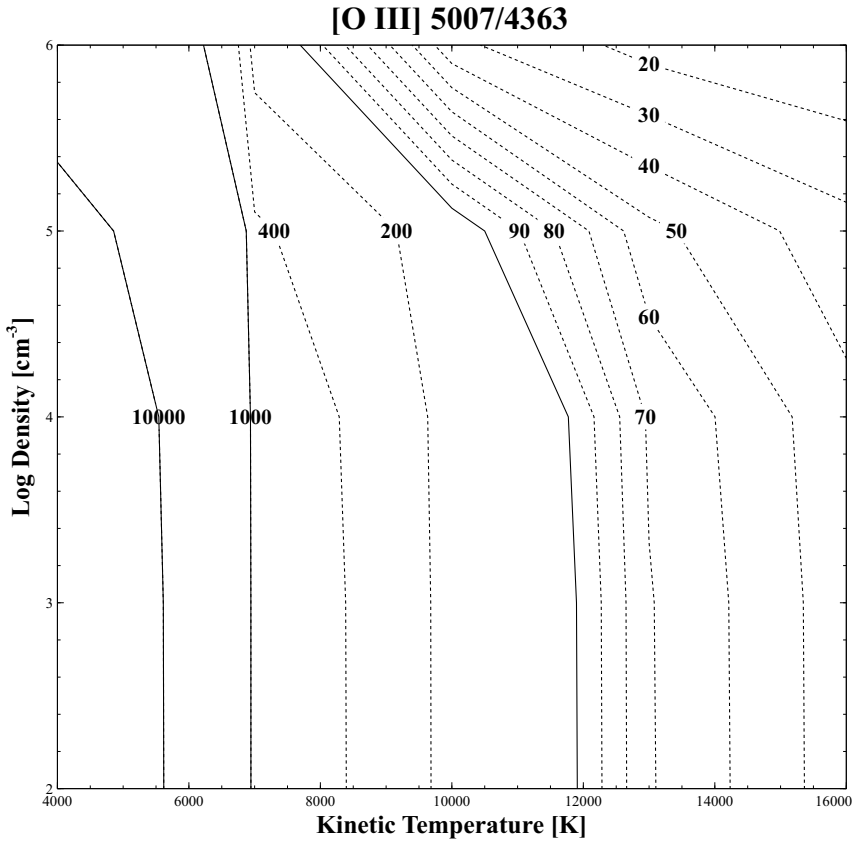

Figure 16.3: The [O III] line ratio as a function of density and temperature.

energy units understood by this option. Note that if you use the **units** option and choose a wavelength type unit, e.g., **micron**, then the saved wavelengths will always be in vacuum, regardless of what is used in the main output. If you do not use the **units** option, wavelengths will be saved in the same format as in the main output, which means they can be either in air or vacuum, depending whether you use the **print line vacuum** command.

The line identification, element and ion, starts the output line, in the form used in the usual emission-line printout. Next follows the line's wavelength or energy. Next, the line's total (including line overlap), and specific (ignoring line overlap) optical depths are printed. Finally, the damping constant is printed. The optical depth is integrated over the model, but the damping constant is evaluated for the conditions in the last zone. Results are reported for the last computed zone unless the **every** option appears.

In an open geometry (the default) the optical depths are integrated over the slab. In the case where **sphere static** (see section 9.11) is given we integrate over the near and far side of the sphere, but in the case of an expanding sphere (the default for the **sphere** command) it is assumed that the approaching and receding parts of the sphere are Doppler shifted so much that there is no overlap between the line profiles any more.

16.68.9 Save line optical depth some

This saves the mean optical depth of an arbitrary number of emission lines as a function of depth into the cloud. These must be requested for single line components: the optical depth for blends will be reported as 0.

For converged models, the total optical depth through the cloud is the same regardless of how it is accumulated (inward or transmitted). The direction of integration becomes significant only when considering the optical depth as a function of depth through the cloud. The command integrates the optical depth from the illuminated face up to the current depth into the cloud.

The emission lines are specified on the input lines that follow the command and end with a line with the keyword **end** in columns 1–3. The format for entering spectral lines is described in Section 2.5.3.

The save output begins with emission-line labels and wavelengths. The remaining output gives the optical depth structure. The first column is the depth [cm] into the cloud. The remaining columns give the optical depth for each line.

The following illustrates its use;

```
save lines, optical some, "lines.opt"
o  3 5006.84A
o  1 6300.30A
end of lines
```

16.68.10 Save line pressure.

This produces a list of the most important contributors to the line radiation pressure for each zone.

16.68.11 Save line populations [row [lower|upper|ratio|tspin]]

This saves the upper and lower level populations of the requested spectral lines.[11] The command expects two filenames to be given, each enclosed in double quotes. The first specifies the output file, while the second contains the spectral lines to be tracked. For example,

```
save line populations "output.txt" "linelist.dat"
```

will read lines from `linelist.dat` and save the output in `output.txt`.

By default, there is one line of output per spectral line per zone. Each output line lists the depth, the transition line label, the lower and upper level densities, the ratio $(n_u/g_u)/(n_l/g_l)$, and the corresponding excitation temperature. An example drawn from work on hyperfine lines in galaxy clusters is the following:

```
#depth       emline          nl           nu           (nu/gu)/(nl/gl)  Tspin
5.00000e-01  Fe26 348.000m   9.3587e-11   9.0239e-15   3.2141e-05       3.9964e+00
5.00000e-01  Fe24 3068.00m   1.2614e-08   1.0044e-08   2.6540e-01       3.5353e+00
```

If the **row** keyword appears, the command will generate one line of output for each zone, listing the depth, and for each spectral line, one of the columns above. By default, the population ratio is reported, but the lower level, upper level populations, or the spin temperature may be reported, instead; the relevant keywords are **ratio**, **lower**, **upper**, and **tspin**, respectively. For example, using the **row** option with the previous command, modifies the output to

```
#depth       Fe26 348.000m  Fe24 3068.00m
5.00000e-01  3.2141e-05     2.6540e-01
```

These results may be used in post-processing. For instance, the population ratio, $(n_u/g_u)/(n_l/g_l)$, may be used by external programs to correct absorption coefficients for stimulated emission.

Note that results for lines that are not associated with an atomic model (e.g., blends, recombinations lines) are always given as zeros.

16.68.12 Save line RT

This produces a file containing information concerning line radiative transfer. A series of lines that specify which emission lines to produce follow the command. The format for entering spectral lines is discussed in Section 2.5.3. The line list ends with a line that starts with **end**.

16.69 Save map, zone 3 [range 3999 to 4500]

This produces a map of the heating and cooling rates as a function of temperature. The details of the map are described in the description of the **map** command. The first number is the zone for the map, zero if only a map of the first zone

The optional keyword **range** specifies the temperature range of the map. If this option is specified then the zone number must come first and is followed by the lower and upper

[11] Until 2022, the command reported the level populations of (almost) all lines known to the code. This caused the output to be several MB even for a single zone model. The current output was chosen as more user-friendly.

temperature limits to the map. Both temperatures will be interpreted as logs if the first number is ≤ 10. The keyword **linear** will force that interpretation. If the temperature range is specified then there must be three numbers on the line—the stopping zone number followed by the lower and upper limits to the temperature.

Normally 20 steps occur between the lowest and highest temperature in the map. The number of steps is reset with the **set nmaps** command.

16.70 Save molecules

The densities [cm^{-3}] of all molecules will be saved. The depth and the point and extended visual extinction into the cloud are given in the first three columns. The first line of the output gives header labels for the molecules.

The **save species** command, section 16.86, allows the molecules to be printed to be chosen (or all molecules to be printed, if the **all** option is used), so will generally be more useful unless both the extinctions and *all* molecules are required. The **save molecules** command will be removed in a future release.

16.71 Save opacities [total, grain, element]

This gives some of the opacity sources considered by the code. The photon energy is given in the first column and the opacities are in the columns that follow. This command recognizes the **units** option to change the energy scale. One of the keywords described in the following sections must appear.

The opacities are only defined over the energy range over which the incident radiation field is defined. The resulting opacities will not extend to high energies for softer continua. A hard continuum such as **table agn** should use used in the input stream to obtain the opacities over the full energy range.

Results are reported for the last computed zone unless the **every** option appears.

16.71.1 Save total opacity

If the keyword **total** appears then the total opacity κ, the absorption cross section multiplied by the density of the absorber, summed over all constituents, will be saved. This has units cm^{-1}. The optical depth would be κ multiplied by a physical depth. This opacity includes all constituents, gas phase and grains, for the last computed zone, with unit filling factor. The first column is the photon energy and the second is the total opacity. The absorption and scattering opacities follow. The fifth column gives the albedo, the ratio $\kappa_s/(\kappa_s+\kappa_v)$. The κ's are the scattering and absorption parts of the total continuous opacity. The last column is a label indicating the ionization edge for each species included in the calculation.

16.71.2 Save grain opacity

The total grain cross section per hydrogen [cm^2/H], for all grain types that are included, will be saved after the last zone is calculated. Successive columns give the photon energy, the total

(absorption plus scattering) cross section per hydrogen, the absorption cross section per hydrogen, and the scattering cross section per hydrogen. The total and scattering opacity discount forward scattering so are for the extended source case. The last column gives the scattering cross section per hydrogen with forward scattering included so is for the stellar case.

16.71.3 Save fine opacities range 0.7 to 1 Ryd, coadd 15 cells

The code's execution time is partially set by the resolution of the continuum mesh due to the need for frequent reevaluations of opacities and rates. A very fine continuum mesh, with resolution of 1 km s^{-1} or better, is used to automatically treat line overlap. The main opacity array cannot use this resolution because single models would then have *very* long execution times. Instead, the code uses a multi-grid approach where a coarse continuum is used for most integrated quantities but a fine continuum grid is also present to handle the line overlap problem. This command will output the current contents of the fine opacity array. This only includes lines, not continuous emission or absorption from the cloud.

The resulting output file will be huge. Several options make the file smaller. If the keyword **range** occurs then the first two numbers on the command line are the lower and upper bounds to the energy range in the output. If neither is specified then the full energy range is reported. The **units** keyword, described on 200, can be used to change the units of both the range and the resulting output.

The last optional number says how many neighboring cells to co-add. If no number appears then 10 are coadded to reduce the amount of output. This must be the third number on the line, so the **range** option must also be specified to change the number of cells to coadd.

By default, only cells with non-zero opacity will be output. The keyword **all** will report all cells.

16.71.4 Save element opacity [name]

If neither the **total** or **grains** keywords appear then the name of an element must be specified. The keyword consists of the first four characters of any one of the 30 elements now incorporated in the code. The photon energy (eV) and total photoionization cross section (Megabarns [$10^{-18}$$cm^{-2}$]) for all stages of ionization of the specified element will be saved. A save file name must still be specified to get past the command line parser but is totally ignored.

The photoionization cross section of each stage of ionization is saved in a series of files. The name of the file will start with the first four characters of the element's name, followed by the stage of ionization (the atom is one), ending with `.opc`. Examples are `CARB1.opc` or `CARB6.opc`.

The code stops after producing these files. Only one element can be treated at a time because of this. If you want to examine the opacity files for more than one element you will need to run the code one time with each element.

This version of the **save opacity** command is misnamed. It actually saves the photoionization cross section, not the opacity due to the element.

16.71.5 Save opacity figure

This version of the command creates the file needed to generate one of the figures used in another Part of HAZY. The output gives the energy in rydberg, then keV, following by the hydrogen, helium, and total gas opacities. The opacities are in units of 10^{24}cm^{-2} and have been multiplied by the cube of the energy in rydberg.

16.71.6 Save opacity shell 26 5 3

This saves the state-specific photoionization cross section for a subshell of any species. The first number is the atomic number of the element, the second number the ionization stage, 1 for an atom, and the third number the subshell, between 1 and 7 representing $1s, 2s, 2p$, etc. The save file will contain the incident photon energy in rydberg followed by the cross section [cm^2].

16.71.7 Save brems opacity

This saves the free-free (a.k.a. brems) opacity of the gas as a function of frequency in the last computed zone. The first column gives the frequency in rydberg, the second column the regular free-free opacity (between electrons and ions), and the third column the electron-electron brems opacity. The latter two quantities both have units cm^{-1}.

16.72 Save optical depths

This gives the total, absorption, and scattering continuum optical depths for the computed geometry. For a spherical geometry this is the optical depth from the illuminated face to the outer edge of the cloud and not the total optical depth. The photon energy is followed by the total absorption and scattering optical depths. This command recognizes the **units** option to change the energy scale.

16.72.1 Save fine optical depths

The code's execution time is partially set by the resolution of the continuum mesh, due to frequent reevaluations of opacities and rates. For problems related to line overlap, a very fine continuum mesh, with resolution of 1 km s^{-1} or better, is used. The main continuum array cannot use this resolution because single models would then have *very* long execution times. Instead, the code uses a multi-scale approach, where a coarse continuum is used for most integrated quantities, but a fine continuum grid is also present to handle the line overlap problem. This command gives the current contents of the fine optical depth array. This only includes lines, not the continuum.

The resulting output file will be huge. Several options make the file smaller. If the keyword **range** occurs then the first two numbers on the command line are the lower and upper bounds to the energy range in the output. If neither is specified then the full energy range is reported. The **units** keyword, described on 200, can be used to change the units of both the range and the resulting output.

The last optional number says how many neighboring cells to co-add. If no number appears then 10 are coadded to reduce the amount of output. This must be the third number on the line, so the **range** option must also be specified to change the number of cells to coadd.

By default, only cells with non-zero opacity will be output. The keyword **all** will report all cells.

16.73 Save OTS

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

16.74 Save overview

This saves an overview of the thermal and ionization structure of the cloud and is a major output mechanism for the code. The first numbers are the depth [cm], temperature [K], local heating [erg cm^{-3} s^{-1}], total hydrogen density [cm^{-3}], and electron density [cm^{-3}].

These are followed by various ionization fractions. The H_2 molecular fraction is expressed as $2n(H_2)/n(H)$. Neutral and ionized hydrogen fractions are followed by the ionization fractions for the three stages of ionization of helium, the carbon molecular fraction $n(CO)/n(C)$, the first four stages of ionization of carbon, C^0 through C^{3+}, and the first six stages of oxygen. The last columns give the visual extinction (for point and extended extinction) and the optical depth at 912Å from the illuminated face to the current position.

All reported quantities are the value itself. This command accepts the **_log** option (page 194) to report quantities in the log style used in C13 and before.

16.75 Save PDR

This gives some quantities relevant to a photodissociation region (PDR). The first column gives the depth into the cloud [cm]. The second is the total hydrogen column density [cm^{-2}]. The gas kinetic temperature follows. These are followed by the ratios of densities of H^0, $2H_2$ and $2H_2$* to total hydrogen, C^0 and CO to total carbon, and water to total oxygen. The next number is the (dimensionless) intensity of the UV continuum relative to the Habing background. The last columns give the total extinction in magnitudes in the V filter measured from the illuminated face of the cloud for a point and extended source.

16.76 Save performance

The code will output the zone number, the time required to compute that zone, the elapsed time since the start of the calculation, and the number of times per zone that the most nested (ionization) solver was called. This is intended as a mechanism to identify zones that require large amounts of work to converge.

16.77 Save pointers

This gives the element number, ion stage, and the shell number, for all shells of the elements heavier than helium. This is followed by the energy of the lower and upper ranges of this shell and the photoionization cross sections at these bounds.

16.78 Save physical conditions

The physical conditions as a function of depth will be saved. The depth into the cloud [cm] is followed by the temperature [K], hydrogen and electron densities [cm^{-3}], heating [erg cm^{-3} s^{-1}], the radiative acceleration [cm s^{-2}], and filling factor.

16.79 Save pressure

Various contributors to the total pressure in the gas equation of state will be saved. Pressures are given in dynes cm^{-2}. Successive columns give the following:

depth is the depth [cm] from the illuminated face to the center of the current zone.

Pcorrect is the correct pressure at the current position. This is the sum of all contributors to the total pressure. This may include, among other terms, gas, radiation, turbulent, magnetic, and ram pressure.

Pcurrent is the current pressure. It will be equal to the correct total pressure, the previous term in the output, if the pressure has been converged. The tolerance on the pressure convergence is controlled with the **set pressure convergence** command.

PIn+Pinteg is the sum of the gas and radiation pressure due to the incident continuum evaluated at the current position. In a hydrostatic model, such as the Orion model proposed by Baldwin et al. (1991), this will be the total pressure and will be equal to the sum of the gas pressure at the illuminated face and the momentum absorbed from the incident continuum.

Pgas(r0) is the gas pressure at the illuminated face of the cloud.

Pgas is the current gas pressure.

Pram is the current ram pressure. This is only non-zero if the gas is moving, which only occurs when the wind geometry is computed.

Prad(line) is the current line radiation pressure.

Pinteg is the current integrated pressure due to attenuation of the incident continuum.

V(wind km/s) is the wind velocity in km s^{-1}. This is only non-zero for a wind geometry.

cad(wind km/s) is adiabatic sound speed in km s^{-1}.

P(mag) is magnetic pressure at the current position.

V(turb km/s) is the turbulent velocity in km s^{-1}.

P(turb) is the turbulent pressure.

int thin elec the integrated radiation force for electron scattering opacity only, in the absence of any absorption.

conv This is "T" if the pressure is conserved and "F" if it is not.

16.80 Save qheat

The probability distribution for the grain temperatures is saved. The first column is the grain temperature and the second column gives $dP/d\ln T$ as defined using the nomenclature of Guhathakurta and Draine (1989). Peter van Hoof added this command.

16.81 Save radius

The zone number is followed by the distance to the central object, the depth to the illuminated face of the cloud, and the zone thickness, all in cm. If the keyword **outer** appears then it will only write this information for the outer radius.

16.82 Save recombination [option]

16.82.1 Save recombination coefficients

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

16.82.2 Save recombination efficiency

This gives the radiative recombination rate and escaping fraction for the 1S and 2P terms of hydrogen.

16.83 Save results

The intrinsic intensities of all emission lines with non-zero intensities, and all column densities, can be saved at the end of the calculation by entering the command **save results last iteration**. This is one way to save the results of a grid of models. The resulting file contains the entire input stream as well.

The resulting save output will have the line information spread over 6 columns. For some data base applications it would be better to have a single column of results. If the keyword **column** appears then a single column is produced. If no keyword occurs, or the keyword **array** does, then the wide format is produced.

16.84 Save secondaries

The rate of secondary ionization of H^0, dissociation of H_2, and excitation of H I Lα, are given as a function of depth into the cloud.

16.85 Save source function [depth, spectrum]

16.85.1 Save source function, spectrum

The continuum source function for the local diffuse radiation field will be saved. The first column is the photon energy. The second column is the diffuse radiation field at that energy, in units of photons per second per rydberg. Column three contains the total absorption opacity [cm^{-1}] at that energy. Column 4 contains the source function, the ratio of the diffuse field to opacity (both have the units described above). The last column gives this ratio relative to the Planck function at the local electron temperature.

The last column is a measure of the local source function relative to the local Planck function. This will generally be nearly unity for a thermal plasma close to LTE. Ground states of atoms of hydrogen and helium generally have departure coefficients greater than unity so this ratio will be less than unity at energies when their emission dominates. The helium ion can have departure coefficients much smaller than unity for nebular conditions so the source function at helium-ionizing energies can be greater than the Planck function.

16.85.2 Save source function, depth

The source function of the diffuse field at an energy slightly greater than the Lyman continuum edge will be saved for all depths into the cloud. All quantities are evaluated at the lowest energy continuum cell that lies within the Lyman continuum. The first column gives the integrated optical depth from the illuminated face of the cloud to the current position. The second is the ratio of the diffuse field (photons per rydberg per second) to the absorption opacity. The third is the OTS continuum at this frequency, and the last is the ratio of the OTS continuum to the opacity, a photon measure of the OTS source function.

16.86 Save species [options]

These commands provide information about the molecules, atoms, and ions that are treated by a unified "species" approach. Options described below specify what information, for instance, population or column density, to report. If an ion or molecule is resolved into levels then the quantity for each level is reported. **Note that the reported levels are in the same order as in the input atomic data models.**[12] For unresolved species only the total is given. The rules for how different species are defined is given on page 14 above.

[12] CLOUDY sorts the levels internally, so the calculations are correct and self-consistent. We report results in the input order to keep them intelligible. Users are advised to review the input database models when level-resolved output is requested.

Warning about ortho/para resolved LAMDA models Several molecule models in the third-party database LAMDA separate species into ortho- and para- systems. The **save species** command in this version of Cloudy has the limitation that only the last system in the LAMDA masterlist will be reported in the output. This affects only the output of these commands. All recognized systems will be included in the simulation.

The **save species** commands are of two types. The first group reports derived physical quantities such as the column density in various levels within the species. The second group reports fundamental physical properties of the model, such as the levels, line collision strengths, etc.

16.86.1 Saving derived physical quantities

Specifying a particular species

By default this command reports details about a list of species, each on a new line and without quotes, followed by **end**.

It is possible to specify only one species matcher by including its label in double quotes, as well as a single-element list. Some rules:

- The file name for the saved data must appear as the first item in double quotes. Any species label comes second.
- The species label must match one of the labels given with the **save species labels** command. These are case sensitive.
- The rules for how the species labels work are defined on page 14 above.

The **all** option will print all molecules and ions which exist in the calculation.

Level populations

In addition, individual states of level-resolved species can be accessed by including the level index in square brackets after the species name, e.g. `"He+[1]"` or `"CO[2]"`, or a range of levels if range indexing is used e.g. `"He+[1:5]"`. The indices are on the physics scale so count from a lowest level of 1. If the lower bound is omitted, the range starts at the lowest available level, likewise if the upper bound is omitted, so `"He+[:]"` prints all levels. The individual levels are not reported by the **save species labels** command.

The total for the species will be reported if the species name is specified without the square brackets, as in `"CO"`. The population of the lowest level will be reported if `"CO[1]"` is specified.

Other quantities

Other quantities can also be printed, as shown in Table 16.1.

Examples

The following gives examples:

Table 16.1: Other quantities which can be added as columns in **save species**.

`*depth`	zone depth
`*AV`	point AV
`*AVx`	extended AV
`*temp`	temperature
`*time`	elapsed time (for time-dependent calculations)

```
#
# save total water column density
save species column density "pdr.col" "H2O"
#
# ions use baryon, not spectroscopic, notation.  This is C IV == C+3
save species populations "flow.pop"  "C+3"
#
# save set of species, some for particular levels
# total in C+, populations in the first two levels of C+, the lowest level of CO,
# the electron density and the point visual extinction
save species densities "pdr.pop"
"C+"
"C+[1:2]"
"CO[1]"
"e-"
"*AV"
end
```

Save species column densities

This reports the column densities [cm^{-2}] at the end of the iteration.

Save species departure coefficients

This reports the departure coefficients [dimensionless] for all zones. The depth is given first, and then the departure coefficient for each species.

```
# save all levels
save species departure coefficients "sim.dep" "H[:]"
# only save total for atom
save species departure coefficients "sim.dep" "H"
```

Save species densities

This reports the density of species and of levels [cm^{-3}] for all zones. The depth is given first, and then the density for each species.

Save species optical depths

This saves the optical depths of all lines. The lower and upper index of the transition is followed by the line optical depth. We report line-center optical depths.

Save species levels

This reports the number of levels used for each species, for each zone. If a species is not resolved, that is, it has no internal structure, the number of levels will be given as zero. This provides a way to watch the number of levels in a model change as the physical conditions change.

Tables giving the energy levels, their designation, and statistical weight, are contained in "energy" files located within each database.

Save species continuum [units Ryd] [row [inward, outward]]

This reports the pseudo-continuum of a species, that is, its (line) emission in a given wavelength range. The output file name, and the species must be specified in that order in double quotations, as in

```
save species continuum "feii.pseudo-cont" "Fe+"
```

By default, the continuum band is from 1000 Å to 7000 Å, split in 1000 logarithmic bins. These defaults can be modified with the **set species continuum** command, described in Section 19.13.47.

The output file lists the photon energy, the total intensity, and the intensity directed inward, and outward, in that order, in four tab-delimited columns. The photon energy is reported in Ryd, unless other units are specified with the **units** option. If a wavelength unit is chosen (e.g., **micron**) vacuum wavelengths will be reported over the entire wavelength range. The units of the intensity in the intensity and luminosity cases follow those for the **save continuum** command, described in Section 16.42.

Because for large grids of calculations, this format may prove inconvenient, the **row** option is provided. When issued, the wavelengths are listed once, in a single row at the top of the file, with subsequent rows holding the intensities for various grid points. Only one intensity sum is then reported per grid point, by default the total intensity. This may be altered by issuing the **inward** or **outward** options, which will cause the corresponding intensity to be reported instead.

The distinction between total, inward- and outward-directed emission is important, e.g., for Fe II lines in quasars, as discussed by Ferland et al. (2009b).

Save species bands

This command accepts a file that contains a number of bands over which the line emission of the indicated species is computed. The results are stored in the specified output file, and are also reported in the emission line list in the main output.

For example, the following computes the emission of the "Fe+" species over the bands found in `data/FeII_bands.dat`,

```
save species bands "bands.feii" "FeII_bands.dat" "Fe+"
```

The first file to appear in the command is the output file, the second is the bands file, followed by the species of choice.

A user-defined bands file may be used instead, but it must follow the format of file `data/FeII_bands.dat`. Briefly, each line holds one band only, described by three (positive) wavelengths, all in angstrom. The first column holds the wavelength that will be used with reporting and the following two columns hold the minimum and maximum vacuum wavelengths of the band, in that order. The remainder of the line is ignored. Comment lines, which begin with a hash (#), are also ignored. Bands are independent of each other, which means that they are not expected in any specific order, and they may also overlap.

The output created in the file specified by the **save** command has four tab-delimited columns. The first column reports the wavelength of the band as defined in the first column of the (input) bands file. The following three columns report the total, inward, and outward species line emission over that band. The intensity units follow those for the **save continuum** command, Section 16.42. By default, intrinsic intensities are reported; for emergent intensities the keyword **emergent** may be used.

The bands will appear in the main output only if a **save** command is entered. The only exception to this rule is the Fe II bands file: it is always parsed, and the summed up emission always appears in the main output.

There, the bands are identified by the spectral label of the species, followed by a single 'b'; e.g., the band for "Fe+" is "Fe 2b". This is then followed by the wavelength of the band, defined in the first column of the bands file. This entry contains the sum of the inward and outward flux. For each of the bands a second entry will be created that only contains the inward flux. These entries will have the label "InwdBnd" followed by the wavelength of the band. See also Hazy 2, Section 9.2.8.

16.86.2 Save species optical depths

This reports the optical depths of all lines for the specified species. The first two columns give the index of the upper and lower levels, in that order, on the physics (not C) scale, the third column gives the vacuum wavelength of the transition in angstrom, and the last column gives the mean optical depth.

16.86.3 Saving fundamental physical properties of the species

Save species energies

This reports the level energies. The default is to give level energies in rydberg but this can be changed with the **units** keyword. The energies are reported relative to the ground state of the species. This command is useful for identifying excitation levels from their index numbers, and should generally be given a range of levels as its argument, e.g.

```
save species energies "test_h_all.nrg" "H[:]"
save species energies "test_h_some.nrg" "H[:10]"
```

The energies for molecular species are reported as zero.

Save species labels

This gives a list of the species labels, a useful way to find out how we refer to a particular atom, ion or molecule. This command will typically be used with the **all** option, to give an inclusive list. Each entry also lists the external database from which the atomic data are derived, or the iso-sequence the species belongs to, if appropriate. If no database name exists then the species does not have internal structure.

Save species lines [options]

This saves the atomic data that CLOUDY uses in a given simulation. If the keyword **all** appears then data for all active species are reported. Otherwise, a species is expected, e.g., "Si+5". Multiple instances of the command may be issued, but each species is processed once across all files. That is, if a species is requested for two output files, it will appear only in the first file.

By default for each transition the spectral label is reported, followed by its wavelength, its lower/upper level indices, the statistical weights of these levels, the Einstein A, the collision strength, and a column that holds a flag, in that order. The flag indicates whether the g-bar approximation was used in the calculation of the collision strength. Values of -1 are appropriate for transitions from iso-sequence species. All other species have values of 0 for transitions with collisional data, and 1 for transitions without collisional data, for which the g-bar approximation has been used.

Optional keywords change this behavior. The transition energy in wavenumbers is reported instead of the wavelength, if the keyword _**wn**_ appears. The oscillator strength is reported instead of the Einstein A, if the keyword _**gf**_ appears. In addition, downward deexcitation rates for all possible colliders are reported, if the keyword **rates** appears. These rates are reported at the end of each line, after the g-bar column.

Save species data sources

This saves a table giving the data sources for atomic and molecular species that are treated with external databases.

16.87 Save special

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

16.88 Save tegrid

The history of the last *NGRID* evaluations of the heating and cooling will be saved. *NGRID* is set in a header file. This is one way to evaluate the stability of the thermal solutions.

16.89 Save temperature

The radius is followed by the temperature and the derivative of the cooling with respect to temperature dC/dT. Next come the first and second spatial derivatives of the temperature with respect to depth, dT/dr and d^2T/dr^2. These derivatives set the rates of conductive heat flow and heat loss respectively.

16.90 Save time dependent

This gives some information on a time-dependent model.

16.91 Save TPredictor

The code estimates the temperature of the next zone from the changes in temperature that have occurred in previous zones. This is only attempted in a constant density geometry. This output gives the old temperature, the estimated new temperature, and the final equilibrium temperature.

16.92 Save XSPEC atable | mtable [range 0.1 200] [normalize [2.3]]

This command produces a file in FITS format for input into the spectral analysis program XSPEC (Arnaud, 1996). This command was added by Ryan Porter and its first application is given in Porter et al. (2006).

Either **atable** or **mtable** must be specified. These keywords refer to additive and multiplicative tables respectively. This command is used with the **grid** command to produce grids of calculations. The file produced by this command contains predictions pertaining to the last iteration of each of the simulations (i.e. the keyword **last** is implicitly in effect). A file produced with **mtable** contains the energy bins and total optical depth, $\exp(-\tau)$, from the illuminated face to the last zone of each simulation in the grid including line absorption, but excluding any scattering processes. A file produced with **atable** contains the energy bins and some component of the spectrum, as determined by the following options:

total - saves the sum of the outward spectrum and the reflected spectrum, including diffuse and incident, lines and continua.

[attenuated | reflected] incident continuum - saves the incident continuum, the attenuated incident continuum or the reflected incident continuum;

[reflected] diffuse continuum - saves the outward diffuse continuum or the reflected diffuse continuum;

[reflected] lines - saves outward lines or reflected lines; and

[reflected] spectrum - saves the total outward spectrum or the total reflected spectrum.

If none of these options are present, the total outward spectrum will be saved. By "reflected" we mean components escaping into the 2π sr subtended by the illuminated face toward the continuum source. If no options are specified for the **atable** command then the transmitted spectra will be saved. Note that the command can be specified multiple times, allowing the user to individually save any or all of the separate components of the predicted spectra.

By default the full frequency range will be saved. The **range** option will limit the range. Both lower and upper limits must be specified. The energies must be in keV.

By default the SEDs stored in an **atable** will be equal to the equivalent SEDs from the **save continuum** command, except that they have different units [photons $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}\,\mathrm{bin}^{-1}$] and normalization will be forced on the inner radius (this is to prevent numerical overflow). The saved spectrum will always conserve energy, i.e. the **set save resolving power** command has no effect. There is an optional keyword **normalize** that will cause the SED to be normalized to 1 [photons $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}\,\mathrm{keV}^{-1}$] at a specific frequency. You can specify a number in keV to set the normalization frequency, or it will default to 1 keV if that number is omitted. If both the **range** and **normalize** keywords are used, **range** (with its associated numbers) must come first. By default the isotropic continua will be included. They can be excluded using the **no isotropic continua report** command.

In the example below we specify that a blackbody is to be varied. The number on the **blackbody** line is ignored. The **range** command specifies that the log of the temperature of the blackbody will range from 4 to 6. The **grid** command specifies that we are calculating a regularly spaced grid with 1 dex steps between 4 and 6. In this case since we only have one parameter varied with the **vary** keyword, there will only be one dimension in the grid and only three simulations. Each of the two **save xspec atable** commands produces a FITS format file. The first file, named `filename1`, will contain the incident continuum for each of the three simulations. The second file, named `filename2`, will contain the transmitted spectrum for each of the three simulations. The commands are as follows:

```
blackbody 1e5 K vary
grid range 4 to 6 in 1 dex steps
save xspec atable incident continuum "filename1"
save xspec atable spectrum "filename2"
```

16.93 Save wind

The radius and depth [cm] are followed by the velocity [cm s^{-1}], total radiative acceleration [cm s^{-2}], the accelerations due to lines and continua alone, and the dimensionless force multiplier. If the keyword **terminal** appears then this is only produced for the last zone, otherwise it is for every zone. This option can be combined with the **save grid** command to save predictions from a series of grid calculations.

16.94 Title "This is a title"

The argument is a title for the calculation and can be useful for organizing the models. The preferred form is a quoted string, but if this is not present, the whole of the line after the command will be used for backward compatibility.

Table 16.2: Trace convergence keywords and routines

Keyword	Routine
pressure	ConvPresTempEdenIoniz
temperature	ConvTempEdenIoniz
eden	ConvEdenIoniz
ionization	ConvIoniz

The title is reprinted several times to label output.

16.95 Trace zone 94 [iteration 2; options . .]

This command turns on ***trace*** information to follow the logical flow within CLOUDY. The code uses adaptive logic to control the calculation and this option provides a way to follow these internal decisions.

The trace begins *after* the zone given by the first number on the line. If the zone is zero, or if no numbers occur on the line, then the trace is turned on at the beginning of the calculation. The second (optional with default of 1) number is the iteration on which the trace should be started. It should be set to 2 to turn on the trace for the second iteration. So the command **trace 0 2** would start the trace at the beginning of the second iteration.

Table 16.2 lists the trace keywords in the first column. The four-character part of the key that must be matched is capitalized. The purpose of each is indicated in the last column.

A string including the word DEBUG is printed when the **trace** command is parsed. This will not be printed if the option **no print** occurs on the **trace** command.

16.95.1 Trace convergence level

This is a special form of the **trace** command that prints an overview of the decisions made during the calculation. The physical state of the gas is determined by nested pressure, temperature, electron density, and ionization solvers. This **trace** option makes it possible to view the decisions made by any of these solvers.

The code will check for one of the keywords **pressure**, **temperature**, **eden**, or **ionization**, and produce output explaining the decisions made by that solver and all higher ones. With no keyword all levels are printed. Successively deeper layers are obtained with the keywords listed in Table 16.2.

There are further keywords that create additional printout. The **OTS** keyword will identify source of OTS fields. The **ESOURCE** option will identify sources of free electrons.

16.95.2 Trace H-like [element name, full]

This turns on extensive printout describing the physics of one of the model hydrogenic atoms. The same code is used to compute atomic properties for all hydrogenic species. If the keyword **full** appears then the printout will be far more detailed. If no element is specified then this will only be

for hydrogen itself. If the name of any other element appears then the printout will be for that element.

16.95.3 Trace He-like [element name, full]

This turns on extensive printout describing the physics of one of the model helium-like sequence atoms. The same code is used for all helium-like species. If the keyword **full** appears then the printout will be far more detailed. If no element is specified then this will only be for helium itself. If the name of any other element appears then the printout will describe that element.

Table 16.3: Trace Keywords and Effects

Keyword	Quantity traced
BETA	OI 8446-Lα problem
CARBon	carbon ionization equilibrium
CALCium	calcium ionization balance
COMPton	Compton heating, cooling, and ionization
CONTinuum	prints out photon arrays, pointers
CONVergence	convergence loop, no other printout
COOLants	cooling
DIFFuse fields	sum of recombination coef in DIFFEM
DR	choice of next zone thickness
EDEN	changes in electron density
GRAIN	details dealing with grain treatment
HEATing	heating agents
HEAVies	heavy element balance
HELIum	helium ionization equilibrium
HELIum ATOM	Helium singlets ionization equilibrium
HELIum IONized	helium ion ionization equilibrium
HELIum SINGlet	Helium singlets ionization equilibrium
HELIum TRIPlet	helium ion ionization equilibrium
HYDRogen	Minimal trce of the H ionization
IRON	Fe abundance, K-alpha emission
LINEes	line pointers, opacity. A's, etc
leveln	LevelN n level atom routine
Ly BETA	Lβ - OI 8446 pumping problem
MOLEcules	rate coefficients for molecules
NEON	recombination, ionization for neon
OPTICal depths	inner, outer optical depths in STARTR
OPTIMizer	Steps in optimize command driver
OTS	ots ionization rates
POINters	pointers for element thresholds
THREe body	three-body recombination rates for metals
TWO photon	induced two photon processes
WIND	Wind geometry

Chapter 17

THE OPTIMIZE COMMAND

17.1 Running the optimizer

If you want to run the optimizer, it is *mandatory* to use one of the following commands

```
# non-MPI runs
cloudy.exe -r optrun
# alternative
cloudy.exe -p optrun

# MPI runs
mpirun -np <n> /path/to/cloudy.exe -r optrun
# alternative
mpirun -np <n> /path/to/cloudy.exe -p optrun
```

This assumes that the input script is in the file `optrun.in`. The normal input and output redirection shown elsewhere will not work in this case and CLOUDY will stop with an error message if you try to use it. The command line switches are described in a section of HAZY 2. Using the **-r** or **-p** command line switches is necessary because the code needs to know the name of the input script in order to generate intermediate files with its name embedded. It is also necessary for the correct functioning of the code in MPI mode. Running in MPI mode is described further in Section 17.6.1.

17.2 Overview

The **optimize** command and its keywords tell the code to vary one or more parameters to try to reproduce e.g. an observed emission-line spectrum, line flux or luminosity, a set of column densities, or temperatures, etc. R.F. Carswell wrote the original code and first implemented the method in CLOUDY and Peter van Hoof extended the methods. The user can choose a minimization method (see Section 17.6) to obtain a best fit to a set of observed quantities. The latter are specified by a series of **optimize** commands. These are described in Section 17.5. The keyword **vary** can appear on many of the commands used to specify physical parameters (Table 17.1). This indicates that this parameter is to be varied in order to obtain a best fit to the observed quantities, and also sets the initial value for the parameter. Section 17.10 below discusses details

of some commands. Other commands exist to influence the behavior of the optimizer, e.g. to set the initial stepsize, limit the range of a physical parameter, set the convergence tolerance, etc. See Section 17.7 for more details.

17.3 Commands with vary option

This section lists all commands which have the **vary** option. Table 17.1 lists all commands, and Section 17.10 gives notes concerning commands with the **vary** option.

17.4 What must be specified

At least one observed quantity must be given, e.g. the desired emission-line spectrum, a line luminosity, kinetic temperature, or a column density. You also specify which physical parameters should be varied. These parameters are specified by a keyword **vary** which may appear on any of the commands listed in Table 17.1. Up to 20 parameters may be varied at a time. Note that it makes no sense to vary more parameters than the number of observables. In that case there will be multiple solutions and the optimizer will not be able to converge. Preferably you should have significantly more observables than quantities to vary so that observational errors do not unduly influence the optimization process. The physical quantities are actually varied as logs within the code and increments (the first steps away from the initial guess) are also logarithmic. The only exceptions to this rule are the **illumination** command, and commands varying a power-law index (**power law** and **ratio alphox**). They are varied as linear quantities. The parameters of the **dlaw** and **fudge** will have the meaning given to them by the user and may be either linear or logarithmic (though the user is strongly encouraged to make them logarithmic).

Several examples of the **vary** option in action are given in the `optimize_*.in` simulations in the test suite. A typical input stream follows:

```
# tell the code to vary the ionization parameter
# and hydrogen density
blackbody, 5e4 K
hden 4 vary
ionization parameter -2 vary
stop zone 1
#
# the following specifies the observed emission lines, order is
# label, wavelength, intensity relative to H-beta, tolerance
optimize lines
O  3 5006.84 intensity =13.8 error =0.1
Blnd 3727 < 2.1 # (only upper limit)
end of lines
#
```

This example tells the code to vary the density and ionization parameter to reproduce the observed intensities of two emission lines.

Information concerning the optimization process is fed to the code as a series of keywords on the **optimize** command. These are described next. Only one keyword will be recognized per **optimize** command.

Table 17.1: Commands with vary option. Entries "var" are explained in Section 17.10.

Command	Quantity varied	Min	Max	Inc.
abundances starburst	metallicity	-3	$\log(35.5)$	0.2
AGN	Big Bump temperature	$-\infty$	$+\infty$	0.5
blackbody	temperature	$-\infty$	$+\infty$	0.5
bremsstrahlung	temperature	$-\infty$	$+\infty$	0.5
constant pressure set	pressure	$-\infty$	$+\infty$	0.1
constant temperature	temperature	$\log(2.8)$	10	0.1
coronal equilibrium	temperature	$\log(2.8)$	10	0.1
cosmic rays	cosmic ray density	$-\infty$	$+\infty$	0.2
database H-like Lyman pumping scale	line blocking scale factor	$-\infty$	$+\infty$	0.1
distance	distance	$-\infty$	$+\infty$	0.3
dlaw	first dlaw parameter	$-\infty$	$+\infty$	0.5
eden	electron density	$-\infty$	20	0.1
element ...	abundance of an element	$-\infty$	$+\infty$	0.2
energy density	energy density temp	$-\infty$	$+\infty$	0.1
filling factor	filling factor	$-\infty$	0	0.5
fudge	first fudge factor	$-\infty$	$+\infty$	var
globule	density	$-\infty$	$+\infty$	0.2
grains	grain abundance	$-\infty$	$+\infty$	1.0
hden	hydrogen density	$-\infty$	$+\infty$	1.0
hextra	extra heating	$-\infty$	$+\infty$	0.1
illumination	Angle of illumination	0	$\pi/2$	0.1
intensity	intensity of source	$-\infty$	$+\infty$	0.5
ionization parameter	ionization parameter	$-\infty$	$+\infty$	0.5
luminosity	luminosity of source	$-\infty$	$+\infty$	0.5
magnetic field ...	magnetic field	$-\infty$	$+\infty$	0.5
metals	metallicity	$-\infty$	$+\infty$	0.5
metals deplete Jenkins 2009 Fstar 0.5	Fstar	0	1	0.1
phi(H)	photon flux	$-\infty$	$+\infty$	0.5
power law	see below	var	var	0.2
Q(H)	no. of ionizing photons	$-\infty$	$+\infty$	0.5
radius	inner radius	$-\infty$	$+\infty$	0.5
ratio	intensity ratio	$-\infty$	$+\infty$	0.2
ratio alphaox	power-law index	$-\infty$	$+\infty$	0.2
set H2 Jura scale	scale factor for Jura rate	$-\infty$	$+\infty$	0.3
set temperature floor	lowest allowed temperature	$\log(2.8)$	10	0.3
stop ... column density	column density	$-\infty$	$+\infty$	0.5
stop mass	total mass	$-\infty$	$+\infty$	0.5
stop optical depth	optical depth at specified energy	$-\infty$	$+\infty$	0.5
stop radius	outer radius	$-\infty$	$+\infty$	0.5
stop thickness	cloud thickness	$-\infty$	$+\infty$	0.5
table Draine	scale factor	$-\infty$	$+\infty$	0.2
table ISM	scale factor	$-\infty$	$+\infty$	0.2
table star ...	usually temperature	var	var	0.1
turbulence	turbulent velocity	$-\infty$	$+\infty$	0.1
xi	ionization parameter	$-\infty$	$+\infty$	0.5

17.5 Observed quantities

This section describes how to specify the observed properties we want to match.

17.5.1 Optimize column density

This specifies a set of column densities to be optimized. A series of species and constraints are read, ending with a line with the keyword **end** in columns 1 to 3, will be read in from subsequent lines. One species is entered per line and there is no limit to the number of entries.

The species can be specified in one of two ways:

Atomic ions: Columns 1 to 4 of the column density lines must contain the first four characters of the name of the element spelled as in the output from the zone results. The first number on the line is the ionization stage, 1 indicates Atom I, 3 indicates Atom III, etc. The second number on the line is the log of the column density [cm^{-2}] and the last optional number is the relative uncertainty. It has a default of 0.05 (5 percent). A column density can be specified as an upper limit by entering $<$ anywhere on the line after column 4.

The following gives an example of its use:

```
optimize column densities
hydrogen 1  < 17 # make optically thin in H^0 Lyman continuum
carbon 4 17.4 error =.001 # match the C+3 column density closely
silicon 3 14.6 # The Si+2 column density
end of column densities
```

Molecular and level column densities: The species is specified as a quoted, case-sensitive string, and no ionization stage is specified. (This is different from previous versions of Cloudy, where the species did not need to be quoted, but only the first 4 characters were used, and an ionization stage of 0 had to be specified.) The species names which are parsed from the lines giving the column density constraints are simply passed to routine ***cdColm***. That routine is described in Part 2 of HAZY. This optimization command accepts any species that ***cdColm*** recognizes. Column densities of molecules and excited states of some ions can be specified by special syntax, and one of the cases that is described where ***cdColm*** is shown in the example below.

The following shows some examples of optimizing molecular and state column densities:

```
optimize column densities
"H-"      < 17 # The H- column density is a limit
"H2"      17.4 error =.001 # Match the total H_2 column density exactly
"H2+"     14.6 # The H2+ molecular ion
"O11*"    16.3 # This is an excited level within the O^0 ground term
"He+[2]"  16.3 # This selects just one level from the He+ ion
"CO[2]"   16.3 # This is a level in the CO molecule
"CO[1:3]" 18.3 # This is the sum of levels 1,2,3 in the CO molecule
end of column densities
```

17.5.2 Optimize, [intensity, luminosity]=36.425 [error =0.1]

This specifies the luminosity or intensity of a line. The code will try to match the predicted intensity or luminosity of the normalization line (usually Hβ and set with the **normalize**

command) match this value. The sub-keyword is either **intensity** or **luminosity** and both have exactly the same effect. The number is the log of either the intensity or luminosity of the line in the same units as found in the third column of the final print out. The second (optional) number is the fractional error for the fit between the observed and computed values. If a tolerance is not specified then a fractional uncertainty of 0.05 is assumed.

By default the intrinsic luminosity of the line will be matched. If the keyword **emergent** appears then the emergent luminosity will be considered.

Comments may be entered by placing any of the special characters described on page 23 in column 1.

The following gives some examples of its use:

```
# this will request an H beta intensity of 0.5 erg cm^-2 s^-1
# it applies to Hbeta since this is the default normalization
# line
optimize intensity -0.3

# the following resets the normalization line to 5006.84, then
# asks the code to reproduce its luminosity
normalize to "O  3"  5006.84
# we want a 5006.84 luminosity of 10^34.8 erg / s
optimize luminosity 34.8
```

17.5.3 Optimize continuum flux 350 micron 126 Jansky [error =0.1]

This commands allows you to optimize an observed continuum flux or surface brightness at an arbitrary wavelength. Note that both the keywords **continuum** and **flux** need to be present on the command line. The first parameter must be the vacuum wavelength or frequency of the observation. The syntax for this parameter is exactly the same as for the **set nFnu add** command described in Section 19.13.39. The code will implicitly behave as if a **set nFnu add** command had been given to add the requested frequency point. The second parameter on the command line is the observed flux. The unit of the flux should also be supplied. There are many choices. You can supply any combination of the following keywords separated by slashes: **erg/s** or **W** for the power unit, and **sqcm** (cm^{-2}) or **sqm** (m^{-2}) for the surface area unit. This will define a flux per decade νF_ν. If you want to supply a flux density F_ν, you can combine this with the keyword **Hz**, while for F_λ you can use **A** (Å), **nm** or **micron** (μm). If you want to supply a surface brightness you can combine this with **sr** or **sqas** (square arcsec), otherwise it will be assumed to be the intensity radiated into 4π sr. Some examples of valid units would be **erg/s/sqcm/Hz**, **W/sqm/A**, or **W/sqcm/micron/sr**. Alternatively the following keywords are also recognized: **Jansky** or **Jy** (Jy), and **mJy** (mJy) for supplying a flux density F_ν, while the keyword **MJy/sr** can be used for supplying a surface brightness. A keyword for the unit must be present. If the flux is ≤ 0, or the keyword **log** appears on the line, it will be assumed that the flux is logarithmic (but not the wavelength or frequency!). Observed fluxes must be dereddened for any extinction inbetween the source and the observer. The same normalization will be used as in the rest of the Cloudy output. So if you include the **print line flux at earth** command, you should enter the observed flux at Earth here. See Section 16.19.7 for further details. See also Section 19.13.38 for a discussion of the components that are included in the continuum flux prediction. The third (optional) parameter on the command line is the relative uncertainty in the flux. If it is not specified then an uncertainty

of 5% is assumed. The optimizer will print both the model and observed flux in the same units as were used on the command line. A continuum flux can be specified as an upper limit by entering < anywhere on the line after column 4.

For wavelengths longer than 1 mm this command is counted in the absolute flux category, for shorter wavelengths in the photometry category (see Section 17.8). This distinction is useful so that you can use a radio continuum flux measurement (e.g. at 6 cm) as a replacement for the Hβ intensity, while you would not want the same to be done for dust continuum measurements since the latter do not scale with the number of ionizing photons.

There is no limit to the number of **optimize continuum flux** commands. The following gives some examples of its use:

```
# make sure we normalize the flux as observed
print line flux seen at earth
# and set the distance
distance 980 linear parsec

# optimize a dust continuum flux, e.g. measured by IRAS
# this will be counted in the photometry class
optimize continuum flux 60 micron 220 Jy error 0.10

# optimize a 6 cm radio continuum measurement
# this will be counted in the absolute flux class
optimize continuum flux 6 cm 50 mJy
# an OCRA measurement...
optimize continuum flux 30 GHz 0.1 jansky

# this will set an upper limit of 1e-21 erg s^-1 cm^-2 Hz^-1 (100 Jy) at 100 micron
optimize continuum flux < 100 micron -21 erg/s/sqcm/Hz
```

17.5.4 Optimize lines

This command tells the code to read a list of observed lines that it will try to reproduce. There is no limit to the number of lines that may be entered.

```
#
# the following specifies observed emission lines, order is
# label, wavelength, intensity relative to H-beta, tolerance
optimize lines
O  3 5006.84 intensity =13.8 error =0.1
Blnd 3727 < 2.1 # (only upper limit)
O  3 88.3323m 1.2
O  1 145.495m 1.6
H  1 4.65247m  Elow=105291.6  0.33
end of lines
#
```

One emission line is specified per line and the line must contain information in a specific order. The line starts with a species label and wavelength. Optionally line disambiguation is supported. See Sect. 2.5.3 for details. The next quantity is the observed relative line intensity. This will be in the same units as the relative intensities printed at the end of the calculation. Intensities are normally given relative to Hβ, but can be changed to other reference lines with the **normalize**

command. The last (optional) number is the relative error in the observed line. If an error is not specified then a fractional uncertainty of 0.05 (5%) is assumed. A line can be specified as an upper limit by entering < before the intensity. Nothing may be entered after the last number, except a comment starting with a comment character. The default is to use the intrinsic line intensities. If the keyword **emergent** appears than those will be used instead. This keyword needs to be added to the **optimize lines** command, as shown below.

```
optimize lines emergent
O  3 88.33m 1.2
O  1 145.5m 1.6
end of lines
```

The emission lines end with a line that has the keyword **end** in columns 1 to 3. If this end does not appear correctly then the code will continue reading lines until the end-of-file is encountered.

Comments may be entered using any of the special characters in column 1 that were described on page 23.

17.5.5 Optimize temperature

This specifies a set of observed mean temperatures. Each of the lines that follow gives a species, ion stage, mean temperature, an error, and a specification of whether to weight the temperature with respect to radius or volume. The following is an example:

```
optimize temperature
Hydrogen  1  36200K  volume
H2   0   150K     radius
end of temperatures
```

This behaves as do the previous commands. The "<" option is recognized. Numbers ≤ 10 are interpreted as logs of the temperature. The keywords **volume** and **radius** specify how to weight the mean temperature. The default is to weight over radius and will be used if no keyword is recognized. The label for the element must appear in the first four columns. The ion stage follows with the temperature and an optional error coming next. An ion stage of zero indicates a molecule.

This command simply copies the labels and parameters on the each line and passes them to ***cdTemp***, a routine that determines the mean temperature and is described in a section of Hazy 2. That documentation describes how to request mean temperatures for various ions, molecules, and derived quantities.

17.5.6 Optimize diameter

The specifies the (angular) diameter of the nebula, defined either as 2 times the Strömgren radius for an ionization bounded nebula, or 2 times the outer radius for a density bounded nebula. If the distance to the object is set with the **distance** command, the number will be interpreted as a linear diameter in arcsec. If the keyword **cm** is present, the number will be interpreted as a linear diameter in cm. The keyword **log** can be used to force interpretation as a logarithmic quantity. If the distance to the object is not set, the number will always be interpreted as a diameter in cm. A diameter can be specified as an upper limit by entering < anywhere on the line after column 4.

17.6 Optimization methods

Two optimization methods are in CLOUDY. The ***phymir*** method (van Hoof, 1997) is the default.

17.6.1 Optimize phymir [sequential mode] [8 cpu]

This uses Peter van Hoof's ***phymir*** optimization method (van Hoof, 1997). In particular it tries to avoid being upset by the inevitable numerical noise that is present in any simulation. It has the further option of being able to run on multiple CPUs on multiprocessor UNIX machines, or in an MPI environment. This optimizer is the default.

Phymir by default runs in parallel mode (using more than one CPU) on UNIX systems and in sequential mode on non-UNIX systems. When the code is compiled with MPI support enabled (see a section of Hazy 2), it will also run by default in parallel mode on any (distributed) MPI cluster. The keyword **sequential** forces ***phymir*** to sequential mode. On most systems phymir will be able to detect the number of available CPUs. By default it will use all these CPUs. The command parser searches for a number on the line. If one is present this will be the maximum number of CPUs that are used simultaneously. In MPI runs the number of CPUs will be picked up automatically from the MPI environment. In this case setting the number of CPUs on the command line will have no effect.

The continue option. The ***phymir*** method includes an option to restart the optimization. During an optimization run, the code's state is regularly saved. The optimization can be restarted using this state file by including the command **optimize continue** described below.

17.6.2 Optimize Subplex

This uses the Rowan (1990) implementation of the subspace-searching simplex optimization method. This method does not support the **continue** option described above.

17.7 Controlling the optimizer

These commands change the behavior of the optimizer.

17.7.1 Optimize file= "best.in"

At the end of the optimization process the derived input parameters are written into a file for later use.[1] The default filename is "`optimal.in`" but can be changed with this command. The file name must be valid for the system in use and must be enclosed in double quotes.

It is possible to use this file to do later calculations in which various quantities might be saved for plotting. Also it is generally a good idea to confirm that a single run with CLOUDY does reproduce the same results as the many calls of the code made by the optimizer. The two should agree *exactly* but might not if the code became corrupted during the many calls made during the process.

[1] This replaces the **optimize save** command, that was in versions 90 and before.

17.7.2 Optimize increment = 0.5 dex

Increments are the logarithmic quantities that will be added to or subtracted from the initial guess as the optimizer makes the first steps away from the initial value. The default increments preset in the code and listed in Table 17.1 were chosen with typical conditions in mind. It may be necessary to change these if the process is unable to identify a solution. If a zero or no number is entered as an increment then the default increment will not be changed.

The increments entered with this command affect *only* the *previously* entered **vary** command. The following gives some examples of changing the increments.

```
hden 4 vary
optimize increments .1  # this sets .1 dex changes in hden
brems 6 vary            # increments left at default
radius 13.6 vary
optimize increments .05 # this sets changes in radius
```

17.7.3 Optimize, iterations = 1200

This specifies the upper limit to the number of iterations to be performed. The default is 400. This command is quite similar to the **stop zone** command in that it is a fail-safe method to prevent runaway infinite loops. The optimization process should not stop because of this limit. It may be necessary to increase the limit if the process is still making progress at the end of the calculation.

The number of iterations that the optimizer needs depends on the number of parameters that are varied. To work well there must be enough iterations to allow each of the free parameters to be varied over a wide range. Make sure that this happens by looking at why the optimization process ended. If it ran out of iterations then you should check whether all of the parameters been varied over a significant range. If they have not then you should increase the number of iterations or vary fewer parameters. If you are using the ***phymir*** method, it is possible to salvage a prematurely terminated run by increasing the number of iterations and adding the **optimize continue** command described below. CLOUDY will then continue where the previous run stopped.

17.7.4 Optimize range -2.3 to 3.9

The preset limits to the range over which parameters can be varied are indicated in Table 17.1.

It is sometimes necessary to establish physical limits to parameters. For instance metallicities may be limited to the range $-1 \leq \log(\mathrm{Z}) \leq 0$ by observations or physical plausibility. This command sets the bounds that the optimizer will use. The arguments are the lower and upper limit to the range of variation of the *previously* entered **vary** commands. Examples follow.

```
hden 4 vary
# the following sets limits to range of density
optimize range from 3 to 5
# There will be no range for this one
brems 6 vary
radius 13.6 vary
# this sets limits to radius
optimize range from 13 to 14
```

If the optimizer tries to go outside of the allowed range, a very large χ^2 is returned without actually calculating the model. This should steer the optimizer back to the allowed range.

17.7.5 Optimize tolerance = 0.02

The tolerance is a measure of the desired fractional accuracy of the parameters that are varied and is set with this command. The default value of 0.10 should be sufficient for initial trial runs but the final values should be made lower for more precision.

17.7.6 Optimize continue

The ***phymir*** method includes an option to restart the optimization. During an optimization run, the code's state is regularly saved in the file `continue.pmr`. The optimization can be restarted by including the command **optimize continue**. In this case it does not start from scratch, but re-starts using information in the file `continue.pmr` instead. This can be useful when your machine crashed during a lengthy run, you hit the maximum number of iterations before the optimization was finished, or if you want to continue optimizing with a tighter tolerance. The code will be in exactly the same state when it continues as when it wrote the last state file. So in general you will need to edit the input script before you restart the calculation. For instance, the first optimization may have ended because it hit the limit to the number of iterations. If you restart an optimization with a `continue.pmr` file you will need to increase the maximum number of iterations in the input script. Otherwise the code would stop immediately, thinking it was already at the iteration limit.

When you start a run with the **optimize continue** command, the code will immediately rename the file `continue.pmr` to `continue.bak` to avoid it being overwritten by the state files that are produced by the new run. So if you make a mistake, you can rename `continue.bak` to `continue.pmr` and try again.

The **optimize continue** command is *only* supported by the ***phymir*** optimizer.

17.8 Convergence criteria

Observables are placed into one of six categories - "relative flux," "column density," "mean temperature,", "absolute flux", "photometry", and "diameter" (see van Hoof, 1997). For the i^{th} observable the error estimate is

$$\chi_i^2 = \left(\frac{F_i^m - F_i^0}{\min\left(F_i^m, F^0\right)\sigma} \right)^2 \tag{17.1}$$

where F^m and F^0 are the model and observed values and σ is the relative error in the observed value. The uncertainty σ is specified when the observed quantities are entered and has a default value of 0.05. The average of the error estimate is computed for each category. The error estimate summed over all categories is minimized.

17.9 Other optimizer commands

17.9.1 No vary

It is sometimes useful to be able to turn off the optimizer for a given input stream without having to change the (possibly many) occurrences of the **vary** keyword. If the **no vary** command is entered then the **vary** keyword on the other commands will be ignored and a single model will be computed.

The **no vary** command can also be used with the **grid** command to tell the code to ignore the **grid** command and **vary** options.

17.9.2 Optimize, trace start 4

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

17.9.3 Optimize, trace flow

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

17.10 Notes concerning commands with vary option

The keyword **vary** can appear on the commands in Table 17.1. All other keywords of the command (except **trace**) are supported (though in some cases there may be an implicit unit conversion, e.g. if you supply a radius in parsec, the optimizer will vary the equivalent radius in cm). If multiple parameters are entered on the command line, only the first parameter will be varied while the rest will be held fixed. The parameter will be varied as a logarithmic quantity. Any exceptions to these rules will be noted below.

17.10.1 AGN

Only the Big Bump temperature, the first parameter, can be varied.

17.10.2 Cosmic rays

All keywords except **equipartition** are supported.

17.10.3 Dlaw

The first parameter will be varied as entered by the user (i.e. no log will be taken). The user is however strongly encouraged to use logs in the definition of the command if the parameter is not of order 1.

17.10.4 Element

Only the keywords **scale** and **abundance** are supported.

17.10.5 Fudge

The first parameter will be varied as entered by the user (i.e. no log will be taken). The user is however strongly encouraged to use logs in the definition of the command if the parameter is not of order 1.

The initial increment is set to 20% of the initial estimate, or to 1 if the initial estimate is zero. Use the **optimize increment** command to override this default.

17.10.6 Illumination

The illumination angle is varied as a linear quantity in radian.

17.10.7 Metals

The **grains** option can be used to vary the grain abundance with the metallicity. The keyword **depletion** is not supported.

17.10.8 Power law

The **vary** keyword appears in three forms, **vary**, **varyb**, and **varyc**. If **vary** appears then the first parameter, the slope of the power law, is varied. If **varyb** appears then the second parameter, the high-energy cutoff temperature in Kelvin, is varied. If **varyc** appears then the last parameter, the low-energy cutoff, is varied. Only one parameter may be varied at a time. The power law index is varied as a linear quantity since it is usually negative. If either of the cutoff energies is varied, the optimizer limits will be set such that the lower cutoff cannot move beyond the upper cutoff, or vice versa.

17.10.9 Save output

It is not possible to use some of the save output options while optimizing. Save output is designed to save predictions so that they can be analyzed after the calculation is complete. It makes little sense to do this during an optimization run with its hundreds of calculations. After the optimization is complete, rerun the optimal simulation with **save** commands included.

Most **save** options will work during optimization. Those that read a series of lines to determine what quantities to output will not work. This is because the command parser is not used in its usual mode when commands are read during an optimization run.

17.10.10 Radius

It is possible to specify the stopping radius or depth on the line but it is not possible to vary it. Only the starting radius is varied.

There could be a major source of confusion if the second parameter is entered and the two numbers are of the same order of magnitude. The logic used to interpret the second number is described on page 112 above. If the second number is greater than the first then it is interpreted as an outer radius; if less than, then the depth. As a result the interpretation of the second number can change while the first number is varied. It is safer to set an outer radius with the **stop thickness** or **stop radius** command rather than using the second number. If you vary the inner radius while keeping the outer radius fixed with the **stop radius** command, make sure that you use the **optimize range** command to keep the inner radius inside the outer radius. Failing to do so may result in premature termination of the optimization run.

17.10.11 Ratio alphox

The power law index is varied as a linear quantity since it is usually negative.

17.10.12 Stop thickness

Only one thickness can be specified. Additional parameters for the second and subsequent iterations will be ignored.

17.10.13 Stop radius

Only one radius can be specified. Additional parameters for the second and subsequent iterations will be ignored.

If you want to vary the outer radius while keeping the inner radius fixed, make sure that you use the **optimize range** command to keep the outer radius outside the inner radius. Failing to do so may result in premature termination of the optimization run when the optimizer takes the outer radius inside the inner radius. Alternatively, you can also use a combination of the **radius** and **stop thickness** commands.

If you want to vary the inner radius as well as the outer radius of a nebula, always use a combination of the **radius** and **stop thickness** commands. Varying the inner and outer radius simultaneously can lead to premature termination of the run if the optimizer takes the inner radius outside the outer radius, and is therefore not recommended.

17.10.14 Table star ...

The first parameter on the command line will always be varied, even if that is not the temperature. The minimum and maximum of the varied parameter depend on the chosen grid and are set in accordance.

17.11 Notes concerning the optimization process

17.11.1 Use physically motivated initial conditions

The algorithm will not be able to find a solution if one is not physically possible. For instance, an observed He II λ4686/Hβ intensity ratio of 0.5 cannot be produced by a 2e4 K blackbody no

matter how many other parameters are varied (it produces no He^+-ionizing radiation). It is probably necessary to start with parameters in the general area of the successful model. When far from the solution, it is also a good idea to use a large tolerance (using the **optimize tolerance** command) to stop it from over-optimizing a bad solution. Then do follow-up runs with a smaller tolerance once you have a better idea where the optimum solution is.

17.11.2 Change the increment size

The initial increment will often be the largest step ever taken during the optimization process. If the initial parameters are far from the solution then it may be wise to increase the increments. Depending on the optimization method used, it may not be possible to find solutions more than one or two increments away from the initial guess. If the increments are too big it may jump over valid solutions.

17.11.3 Set physically motivated limits to the variable quantities

The optimizer driver uses very simple methods and understands surprisingly little modern astrophysics. For instance, while trying to reproduce an observed He II λ4686/Hβ intensity ratio of 0.5 by varying the temperature of a blackbody radiator, the algorithm may just as well examine the consequences of photoionization by a 100 K radiation field. Physically, it is known that He II emission only occurs for stars hotter than $\sim$5e4 K (AGN3 section 2.5), so there is little purpose in examining temperatures lower than this. The process will converge more quickly if reasonable bounds to the range of the varied quantities are set using the **optimize range** command.

This advice is dangerous, of course, since you may limit yourself to solutions close to those you anticipate. Experiments should also be performed far from the anticipated solution.

17.11.4 Don't give up!

My experience is that this process works about a quarter of the time. The problem is that the algorithm can easily home in on a local minimum which is actually a very bad global solution. When this occurs the best idea is to restart the optimization process with a different set of initial conditions. Better yet is to start the process with parameters that give answers known to be close to the solution, although there is some danger of limiting the outcome to be what you expect. Finally, don't be afraid to use CPU time.

17.12 Other optimization methods

Much of astrophysics involves solving the inverse problem—knowing an answer (the spectrum) and trying to deduce the question (the conditions that caused it). However, optimizing a multi-dimensional function is more an art than a science. We would be interested in learning about, and possibly adopting, other promising efficient optimization methods (especially if they can be parallelized). BSD-compatible license code is necessary.

Table 17.2: Optimizer results

Method	Iterations	Density	Temperature
Initial condition	—	5	4
Phymir	50	4.988	3.998
Subplex	64	5.001	3.997

17.13 The optimizer test cases

The suite of test cases that comes with the code includes scripts to drive each of the optimizers. These scripts are `optimize_*.in`. These were produced by first running the code at a hydrogen density of 10^5 cm^{-3} and a temperature of 10^4 K. The spectrum of [O II] and [O III] emission lines was taken from this calculation. Each optimizer starts at a density and temperature some distance away from this solution and tries to reproduce the spectrum. Table 17.13 shows the results of this test.

Chapter 18

THE GRID COMMAND

18.1 Introduction to grids

The **grid** command varies one or more of the input parameters to compute a grid of models. It is actually a form of the **optimize** command and uses much of the same code. It was added by Ryan Porter and its first use was described in Porter et al. (2006). The grid models will be run in parallel on multi-core machines.

In a typical setup one or more parameters will be varied and output saved. A typical **grid** run might look something like the following:

```
# set up a simple AGN calculation
table agn
# vary the ionization parameter
ionization parameter -1 vary
# this grid command says how to vary the value of the ionization parameter in the
# preceding command.  Go from log U =-2 to log U = 0 in 0.5 dex steps
grid -2 0 0.5
stop column density 19
hden 10
iterate to convergence
# save the grid points, the value of the ionization parameter of each model
save grid "gridrun.grd" last no hash
# save intensities of selected lines
save line list "gridrun.lin" "LineList_BLR.dat" last no hash
```

This script computes a series of AGN models in which the ionization parameter is varied between two limits. The **save grid** command saves the values of the ionization parameter and gives some information about the calculation. The **save line list** command saves a set of lines for each grid point.

18.2 Running the code with grids

18.2.1 Command line setup

It is *mandatory* to run CLOUDY using one of the following commands to use the **grid** command:

```
# non-MPI runs, using fork
cloudy.exe -r gridrun
# alternative
cloudy.exe -p gridrun

# MPI runs
mpirun -np <n> /path/to/cloudy.exe -r gridrun
# alternative
mpirun -np <n> /path/to/cloudy.exe -p gridrun
```

This assumes that the input script is in the file `gridrun.in`.

Using the **-r** or **-p** command line switches is necessary because the code needs to know the name of the input script in order to generate intermediate files with its name embedded. It is also necessary for the correct functioning of the code in MPI mode. The **-r** option is included in the recommended setup for running CLOUDY.

18.2.2 Grids on parallel machines

On most systems the **grid** command will run in parallel by default using all available cores / hyperthreads. This includes all UNIX systems, as well as Cygwin under Windows. On most machines there are twice as many threads as CPUs, and the additional threads do not add much computing power. Tests show that a Mac will become unresponsive if all threads are used, so for OS X we use all cores rather than all threads, by default.

Grids on MPI machines By default the **grid** command will use the UNIX fork function to run multiple jobs, and nothing more need be done. MPI can be used to run grids on distributed systems. When the code is compiled with MPI support enabled (see a section of HAZY 2), the **grid** command will run in parallel mode on any (distributed) MPI cluster. This is the preferred method if you want to run very large grids and you need more cores than a single node can provide. The second example above shows how to run on a typical MPI-capable machine.

18.2.3 Forcing sequential runs

You can disable the parallel behavior by adding the keyword **sequential** to the **grid** command. The **sequential** option causes the grid to be executed using only a single core rather than in parallel mode. This option is ignored when running in MPI mode.

18.2.4 Specifying the number of CPUs

You can choose a different number of cores by adding the keyword **ncpus** at the end of the **grid** command. The **ncpus** option allows you to set the number of cores that will be used in parallel grid runs. This option is only effective on systems that support parallelization based on the *fork()* system call. It is ignored in MPI runs since the number of cores is set as an option to **mpirun** in that case. The number of CPUs should be added as a fourth number on the command line, e.g. as follows:

```
blackbody 5.e4 vary
grid 4 5 0.1 ncpus 4  ## limit # of cpus to 4
```

18.3 Grid start point, end point, increment [linear]

Parameters for those commands with the **vary** keyword (see Table 17.1 on page 247) can be varied within a grid. Each command with a **vary** option must be followed by a **grid** command. The value entered on the command with the **vary** option is ignored but must be given to satisfy the command parser.

The **grid** command line must have three numbers. The first and second are the start and end points of the range of variation. The last number is the value of the increment for each step. The increment can be either positive or negative. This behaves much the same way as a *do* loop in Fortran or a *for* loop in C. The **grid** command always changes variables from the start point to the end point. In other words, the start point must be less than the end point when the increment is positive, while, when the increment is negative, the start point must be larger than the end point. Note that the grid points are calculated in random order, so it no longer matters if you use a positive or negative increment[1]. The following varies the blackbody temperature over a range of 10^4 to 10^6 K with three points.

```
# the blackbody command with vary option, the 1e5 K is needed to
# get past the command parser, but is otherwise ignored
blackbody 1e5 K vary
# this varies the blackbody temperature from 1e6 to 1e4 K with -1 dex
# steps, so that 1e6K, 1e5K, and 1e4K will be computed
grid, range from 6 to 4 with -1 dex increments
```

The following is a 2D grid in which both the ionization parameter and metallicity are varied. There will be five values of the ionization parameter and three values of the metallicity computed.

```
ionization parameter -2 vary
grid range from 0 to -4 with -1 dex steps
metals 0 vary
grid range from -1 to 1 with 1 dex steps
```

For nearly all commands, the quantity will be varied logarithmically (current exceptions are the **illumination**, **ratio alphox**, **dlaw**, and **fudge** commands). If the quantity is varied logarithmically, the lower / upper limit and the step size also need to be given as logarithms, as shown above. If the keyword **linear** is included on the **grid** command, then these numbers will be interpreted as linear quantities. As an example, the following will produce a grid of models with a constant electron temperature of 5000, 10000, 15000, and 20000 K.

```
constant temperature 4 vary
grid range from 5000 to 20000 step 5000 linear
```

For some commands the unit of the quantity will be implicitly changed by the grid code. An example is the **radius** command. You can enter a radius in parsec using the **parsec** keyword. However, the grid code will always vary the radius in cm. In this case the lower / upper limit will also need to be the (logarithm of) a radius in cm. There is currently no way to overrule this. In general the lower / upper limit needs to have the exact same units as are used on the command line that is generated by the grid, *not* the command line as originally typed by the user.

[1] In versions C10 and earlier of the code, the grid points were calculated in sequential order (at least for non-MPI runs) so it made a difference then. The option of choosing a negative increment is maintained for backward compatibility.

18.4 Grid list [linear] “filename”

This **list** keyword allows parameter values to be individually listed in an external file that must be given in quotes on the same line. The file should not contain any other text, but may contain comments starting with “#”. The numbers should be separated by whitespace and/or newlines. The numbers will be interpreted as log values unless the **linear** keyword is included. This option is useful when a regularly-spaced grid as outlined above is too restrictive.

18.5 Beware the grid command treatment of temperatures!!

The following will crash with an fpe

```
constant temperature 4 vary
grid range from 5000 to 20000 step 5000  ## wrong, this will crash!
```

This is because of the rule stated above that the **grid** command treats temperature ranges as logs unless the keyword **linear** occurs.

18.6 Grid output options

Several **save** output options were developed to retrieve information from **grid** runs. The **save XSPEC** command will generate FITS files for input into XSPEC (Porter et al., 2006). The **save grid** and **save line list** commands save the grid points and emission-line intensities.

When computing a grid, the default is for save files to not overwrite results of previous grid points (to not “clobber” themselves) and to append the results to the save file instead. Conversely, in optimizer runs the default is for save files to overwrite one another since you are usually not interested in output along the unpredictable convergence path. That way you only get the save output from the fully converged model.

When you include the **separate** keyword on a **save** command, the output of each grid point will be written to a separate file. See Section 16.33.10 for further details.

In the output from **save** commands the default behavior is to put separators consisting of hash marks in between the output from different iterations (unless the **no hash** option is used). In grid runs this behavior is slightly changed. There will be an additional separator at the very end of the output, which will also have the additional text “GRID_DELIMIT” appended, as well as the number of the grid point in the format “grid000000000”. This makes it easy to find back the output from a specific grid point, as well as identify where there is output missing in case some of the grid points have serious problems.

During a grid run the main output for each grid point will be written to a separate file with names like `grid000000000_yoursimname.out`, `grid000000001_yoursimname.out`, etc. (assuming your input script is named `yoursimname.in`). The meaning of the grid index embedded in the filename can be found with the **save grid** command described in Section 16.58. At the end of the grid run these files will be gathered into a big main output file called `yoursimname.out` and the intermediate files will be deleted. For huge grids this operation can take a very long time. However, if you include the

separate option on any of the **grid** commands, the gathering step will be skipped and the main output will remain in separate files, thus speeding up the calculation considerably.

18.7 Other grid options

The **no vary** command can be used to ignore all occurrences of the **grid** command and **vary** option in an input stream.

The **repeat** option on the **grid** command causes the expected number of grid steps to be computed but the initial value of the variable is not incremented. This is mainly a debugging aid.

The **cycle** option causes the grid to be executed multiple times. If no additional number is found on the command line, the grid will be executed twice. You can also add a fourth number on the command line, which will be the number of times the grid will be executed. That number needs to be at least two. This is a debugging aid and should never be used for production work.

18.8 Notes on various commands

18.8.1 Constant temperature

Section 18.5 describes the rules governing how the temperature can appear on the **constant temperature** command.

Chapter 19

MISCELLANEOUS COMMANDS

19.1 Overview

This Chapter describes commands that are used to disable physical processes, change the code's internal behavior, or to take care of housekeeping activities. These are not usually used.

19.2 Introduction to the compile commands

The following subsections describe how to set up and then compile some external data files that are needed by the code. In most cases, the stellar atmosphere files must be compiled if you wish to use some of the **table star** emergent radiation fields. Most of the other files are included in the code's distribution and only need to be compiled if you change the code.

The files produced by the compilation process must be accessible to the code when it is executed from other directories. For this the files need to be somewhere on the search path. This is most easily done if you place the compiled files in the code's `data` directory. See Section 3.3 for further options.

The **compile** command will be executed after the command parser is finished examining the input deck. You must specify a complete model to get past the parser's validation of the input stream. The easiest way to do this is to also include the **test** command in addition to **compile** commands.

The code stops after executing the first **compile** command it comes across. It will not do more than one compilation in a single run.

The script `data/make_data.sh` runs all the **compile** commands mentioned in this Section. It can be used to update the relevant data files.

19.3 Compile stars

Kevin Volk incorporated several large grids of stellar atmosphere continua into CLOUDY in the 1990's. Peter van Hoof has generalized the treatment and made several extensions. The **table star** commands use these atmospheres. The data files can be very large and "direct access" is used to read these files quickly. The result is that the final files are not portable although the code used to

read or write them is. The process of converting the stellar atmosphere files from their original format into a form that can be read by CLOUDY is referred to as compiling the stellar atmospheres.

The **table star** commands will only function if the atmosphere files are installed as described on the code's web site. This is only done once while installing CLOUDY although it will have to be done again if you ever change the continuum energy mesh. This step does not need to be done if you don't want to use these stellar atmospheres.

Full instructions on how to download and install the stellar continua are given at gitlab.nublado.org/cloudy/cloudy/-/wikis/StellarAtmospheres.

19.4 Compile opacities

N.B. This command does not now function and may be removed.

When the code is initialized it spends some time evaluating numerical fits to the needed opacities. This initialization time can be saved if the opacities are compiled and the resulting file placed on the path. To do this, execute the code and enter only the command **compile opacities**. The code will generate a binary file named **opacity.opc** containing the needed opacities and array indices.

N.B. It is not really necessary to compile the opacities— the code will generate them when it starts up if the file does not exist. This may actually slow down the calculation if you are using a fast computer on a slow network.

19.5 Compile recombination coefficients [H-like; He-like]

The code reads in a table of recombination coefficients for the hydrogen and helium isoelectronic sequences when it initializes. This command regenerates that file. It was introduced by Ryan Porter. One of the iso-electronic sequences must be specified with either keyword **H-like** or **He-like**.

19.6 Compile grains

This command prepares the grain opacity files that are used by the **grains** command. This does not need to be done if you only want to use the types of grains included in the data distribution. You can create other species, with their own size distribution and refractive indices, by compiling them with this command. It is also necessary to compile the grains if you change the code's continuum energy mesh.

This command uses a spherical Mie code originally developed by Peter G. Martin using code written by Hansen and Travis (1974), and implemented into CLOUDY by Peter van Hoof, to generate sets of grain opacities from a description of their size distribution and grain material optical properties (van Hoof et al., 2004). The set of grain opacities created by this **compile** command are then used to compute temperature, charge, drift velocity, and emitted spectrum, for

each bin within the grain-size distribution. Section A.2 describes how to create new grain opacity files.

A grain model is specified by the optical properties of the material and by a description of the distribution of grain sizes. The optical properties are given by a file, with a name ending in "`.rfi`", giving the refractive indices as function of photon energy for the full energy range considered by the code. The grain-size distribution is specified in a separate data file with a name ending in "`.szd`". Sample refractive index and size distribution files are included in the data directory or you can create your own. In all cases the **compile** command combines the refractive index data with the size distribution to produce an opacity file (with a name ending in "`.opc`").

Section A includes far more information and should be consulted.

19.6.1 Compile grains

With no options this compiles the minimum number of grain types needed for the **grains** command to function. Examine the output to check for problems. If you change the continuum mesh the grains need to be recompiled. This must be done in the data directory.

The following steps are for running "Compile grains":

1) `cd` into the data directory

2) create a `.in` file and paste the command `Compile grains` in the file

3) Then run Cloudy like you normally do:

```
~/path to source/cloudy.exe  -r  <name of the .in file>
```

19.6.2 Compile grain 10 bins, [filename, ism graphite]

This version of the command produces opacities of a single grain species and is used to create new types of grains. Both the optical properties and size distribution must be specified. This is done by giving keywords (to use one of the standard files located in the data distribution) or a filename between double quotes (to read properties for a new species).

The number on the command line specifies the number of grain size bins to compute. If no number appears then 10 is set by default.

Keywords for standard data sets One of a set of keywords may be used to specify refractive index files and grain size distributions. The keyword is usually the name of a file included in the CLOUDY distribution without the filename extension.

A grain-size distribution may be specified with one of the keywords given in Table 19.1. The first three are single-sized distribution functions that are mainly used for testing the code. For single-sized grains the supplied number of cells is ignored since one cell is always used. These files have names that end in "`.szd`".

The keywords given in Table 19.2 may be used to specify a file containing refractive index information. These files have names that end in "`.rfi`".

Creating new data files If none of the keywords listed in the tables occur then the name of the file must be given explicitly. This may be a user-generated file. A pair of double quotes must surround the filename, as in "`name.rfi`". Names ending in "`.rfi`" are refractive index files while those ending in "`.szd`" are size distribution files. Filenames and keywords can be mixed on a single line as shown in the examples below.

Table 19.1: Grain keywords for size distribution (*.szd files)

Keyword	Size distribution
0m010	0.01 μm
0m100	0.1 μm
1m000	1 μm
ab08	Abel et al., 2008
c15	Small PAH, 15 C atoms
c120	Large PAH, 120 C atoms
ism	ISM distribution, Mathis et al., 1977
orion	Orion distribution, Baldwin et al., 1991

Table 19.2: Refractive index data files in the CLOUDY distribution

Filename	Keyword	Grain type	Reference
ac1-amcarb.rfi	ac1-amcarb	amorphous carbon	Rouleau and Martin, 1991
be1-amcarb.rfi	be1-amcarb	amorphous carbon	Rouleau and Martin, 1991
gdraine.rfi	—	graphite	Laor and Draine, 1993
gdraine03.rfi	—	graphite	Draine, 2003
graphite.rfi	graphite	graphite	Martin and Rouleau, 1991
grey.rfi	grey, gray	grey grain	—
pah1.rfi	pah	PAH	Volk, private comm.
ph2c.rfi	—	charged PAH	Li and Draine, 2001
ph2n.rfi	—	neutral PAH	Li and Draine, 2001
ph3c.rfi	—	charged PAH	Draine and Li, 2007
ph3n.rfi	—	neutral PAH	Draine and Li, 2007
sdraine.rfi	—	astronomical silicate	Laor and Draine, 1993
sdraine03.rfi	—	astronomical silicate	Draine, 2003
sic.rfi	sic	α-SiC	Laor and Draine, 1993
silicate.rfi	silicate	astronomical silicate	Martin and Rouleau, 1991
vacuum.rfi	—	vacuum	—

19.6.3 Some caveats

- If you run the code in another directory than where you compiled the grains, you must make that the files are moved to a location on the search path. See Section 3.3 for details.
- The grains will need to be recompiled if the energy mesh of the radiation field is changed. The input script `compilegrains.in` in the `data` directory will do this.

19.6.4 Examples

```
# compile all grain types
compile grains

# only ism graphite
compile grains ism graphite

# use 20 bins for a silicate
# with Orion size distribution
compile grains 20 Orion silicate

# the grey.rfi file is the one read
# with the key grey, so the following
# is equivalent to grey ism
compile grains "grey.rfi" ism

# explicitly request the graphite.rfi
# (default with graphite) refractive index
# file, and the Orion size distribution
compile grains "graphite.rfi"  "orion.szd"

# these refractive index files are also
# present in the standard distribution
compile grains "gdraine.rfi" ism # graphite, Laor & Draine (1993)
compile grains "gdraine03.rfi" ism # graphite, Draine (2003)
compile grains "sdraine.rfi" ism # silicate, Laor & Draine (1993)
compile grains "sdraine03.rfi" ism # silicate, Draine (2003)
compile grains "ph2n.rfi" ab08 # neutral PAH, Li and Draine (2001)
compile grains "ph2c.rfi" ab08 # charged PAH, Li and Draine (2001)
compile grains "ph3n.rfi" ab08 # neutral PAH, Draine and Li (2007)
compile grains "ph3c.rfi" ab08 # charged PAH, Draine and Li (2007)
```

19.7 Crash [keywords]

This command is intended to test that the code handles various error conditions in the desired manner. These include floating point exceptions, but also other errors like array bounds violations or the use of uninitialized variables. Note that not all error conditions will be caught in a standard setup. Array bounds violations and the use of uninitialized variables will not be caught by default since that would slow down the code too much. One of the following keywords must appear.

The IEEE standard for floating point arithmetic says to not throw an exception when a floating-point exception occurs. Instead the result is set the result to **+/-inf** or **NaN** (not a number)

and have the calculation continue. The code will only crash if this is explicitly requested by setting compiler options or enabling traps on the CPU. The source includes logic that will do this on most platforms. CLOUDY should crash on division by zero, overflow, or invalid operation. The **zero**, **overflow**, and **NaN** options on the **crash** command confirms this.

19.7.1 crash zero

This command causes the code to divide a positive number by zero. This causes a division by zero exception.

19.7.2 crash overflow

This command causes the code to divide a very large number by a very small number. The result will overflow.

19.7.3 crash long overflow

This command causes the code to convert a floating number that is larger than the largest long int into a long int. Some machines may throw an exception in this case (mine does not).

19.7.4 crash NaN

This command will divide zero by zero. This will cause an invalid operation exception.

19.7.5 crash setnan [float, double]

This command will test that the routine ***set_NaN*** sets variables to NaN. A variable is set to NaN with these routines and is then used in a multiplication.

19.7.6 crash isnan [float, double]

This command will test that the routine ***is_nan*** correctly determines whether a variable is NaN. A variable is set to NaN with ***set_nan*** then is asserted not to be NaN. The monitor should fail.

19.7.7 crash segfault

This command causes the code to dereference a ***NULL*** pointer. This will trigger a segmentation violation (segfault) and should always cause the code to crash.

19.7.8 crash assert

This command causes the code to assert that a positive number is less than zero. This should cause an exception when the code is compiled in debug mode, but will have no effect when the optimization is set to a high level or if the flag -DNDEBUG is included with the compiler options.

If the code does indeed crash with a failed assert then it may request that the output be posted on the web site. This is normal and only shows that the asserts are working properly.

19.7.9 crash bounds

This command will cause an array to be evaluated with an index that exceeds the array bounds. In some cases the code may crash when this occurs.

There are multiple types of arrays within the code and keywords say which one to test. Allocated arrays are created on the heap by using ***new***, ***vector***, or ***valarray***. Bounds errors for these types of arrays can be trapped with the compiler or external tools such as ***valgrind*** (http://valgrind.org) and ***Purify***. Declared arrays are explicitly given a size when the variable is declared in the source, as in

```
double c[5];
array<double,5> c;
```

Modern versions of the GNU and LLVM C++ compilers have options to check that limits are not exceeded on a declared array. Finally, the *multi_arr*, *flex_arr*, and *avx_ptr* classes were introduced after the code migrated to C++ and can check on array bounds if the macro **BOUNDS_CHECK** is set when the code is compiled. This is done with the compiler option **-DBOUNDS_CHECK** on most compilers.

A keyword specifying the type of array to test must appear. The keyword **static** will use a static array. Either keyword **stack** or **auto** will use an array created off the stack. The keyword **array** or **vector** will use an array created using the corresponding STL class. Code generated by a bounds-checking compiler should catch these errors. The keyword **multi** will check an array created from the *multi_arr* class. The two keywords **multi iter** will test an iterator in the *multi_arr* class. These should crash if the macro **BOUNDS_CHECK** is set. The **crash bounds multi iter** command does the same as the **crash bounds multi** command except that it tests access through iterators (which behave like pointers) rather than direct access to the array. Finally the keyword **avxptr** allows you can test an array created using the *avx_ptr* class. This should crash if the macro **BOUNDS_CHECK** is set.

The **crash bounds** command has an optional offset to add to the array size **ARR_SIZE**. If *n* is positive then the code will access **array[ARR_SIZE+n]**. If it is negative then the code will access **array[n]**. The default is zero, that is, to access **array[ARR_SIZE]**. If either of two keywords, **low** or **high**, occur with no number then the code will assume **n = 2** and check only the low or high bounds.

19.7.10 crash exception

This command will cause the code to throw an exception. The exception should be caught by the handler in the main program.

19.7.11 crash domain

This command will call a vectorized math routine with invalid arguments. This routine will then throw an exception. The exception should be caught by the handler in the main program. This test will only be executed if AVX vectorization has been enabled by the compiler.

19.7.12 crash undefined

This command multiplies a valid constant by an undefined float created off the stack. Special tools such as *valgrind* can detect this and will report an error. In normal execution, the fact that the variable is not initialized will not be detected, though in some circumstances the multiplication may result in a floating point exception.

19.7.13 crash abort

This command will cause the code to simulate a CLOUDY abort. This exception should be caught and a message giving the reason for the abort should be printed.

19.7.14 crash grid

This command will cause random grid points in a grid to fail due to a randomly selected signal (e.g., a floating point exception) or a CLOUDY abort. On average one in four grid points should fail. This command is intended to test the robustness of the grid command against such failures and should never be used in normal grid runs. The correct behavior of the code is that the grid will finish normally. The **save grid** output should be complete and correctly identify each failure. The output from other **save** commands should also be complete and contain default (invalid) entries for each of the failed grid points.

19.8 Eden -2

This adds free electrons to the gas. The argument is the log of the electron density [cm^{-3}]. This command accepts the **vary** keyword. It is mainly intended to test the behavior of CLOUDY in the limit of very low Compton temperatures. When the color temperature $T_{color} \ll 10^4$ K the gas is almost entirely neutral and free electrons must be artificially added to test the Compton energy exchange problem in the strict TE limit. (Remember, charge conservation is a horrible thing to violate.)

Note: This command adds an extra component of electrons. The **set eden** commands, described on page 298, set the electron density. Both commands violate charge conservation.

19.9 Fudge factors 4.59678 [12.3 34.5 958 . . .]

The numbers appearing on the line can be communicated to any part of the code by calling routine *fudge*. That routine has a single integer argument that is an index to the array of numbers entered on this command line. A call to *fudge*(0) would return the first number[1] and, in the example given above, a call to *fudge*(1) would return the value 12.3. Up to ten numbers can be entered on the command line.

This command is used to pass numbers to temporary or trial parts of the code. Routine *fudge* is a permanent part of CLOUDY and a warning is given at the end of the calculation if this function is

[1] In version 90 and before the array index was on the Fortran scale, so fudge(1) returned the first number. The code is now written in C++ and the array index is interpreted on the C scale. The first number is fudge[0].

evaluated. The code will stop if the index to the array of stored values exceeds the number of values entered. Extra numbers are simply ignored.

The code will abort if *fudge* is asked to return a parameter that was not defined. For instance if the command line reads

```
fudge 4.0
```

then ***fudge***(1) does not exist and the code will terminate.

fudge will return the number of parameters entered on the command line if it is called with a negative argument. The following code fragment shows how to check on the number of parameters and then evaluate *fudge* if *fudge* factors were entered.

```
factor = 1.;
if( fudge(-1) > 0 )
    factor = fudge(0);
```

19.10 Init [“c84.ini”, path]

This reads a set of commands from an initialization file. Frequently-used commands can be stored in this file and easily accessed.

There is no limit to the number of commands that can be in this initialization file. An arbitrary number of **init** commands can appear in a single input stream, as well as inside another initialization file.

The default name for the initialization file is `cloudy.ini`. This will be used if no double quotes occur on the command line. Other file names are specified by including a file name within a pair of double quotes as in “`special.ini`”. The name can be any name allowed by the operating system in use.

The data directory includes the `ini` files that are used in the test suite. They were set up with particular applications in mind. They may or may not be appropriate for your purpose. Please review the contents of these files before using them.

19.11 No . . .

It is possible to disable physical processes as a test of their importance. If a physical process is turned off then a flag is set to indicate that the treatment of a physical process has been disabled.

A warning will be printed at the end of the calculation as a reminder that the results of the calculation are not to be trusted. This warning will not be printed if the four-character keyword (**OK**) appears on the command line. The parenthesis is part of this keyword.

19.11.1 No advection [H-like, He-like, . . .]

This turns off the effects of advection on the ionization, recombination and energy of the gas. Keywords make it possible to turn off these terms for only certain parts of the calculation. The **metals** option will turn off the effects of advection on the ionization of any species not treated as one of the iso-electronic sequences. **Cooling** will turn off thermal effects of advection.

19.11.2 No Auger effect

This turns off the Auger effect. Inner-shell ionizations will eject only one electron.

19.11.3 No blends

The code keeps a list multiplets and blends in the file `blends.ini` in the data directory. This list is an init file that is automatically read by the code when the input script is parsed (i.e., the code behaves implicitly as if the command **init "blends.ini"** had been included at the end of the script). Including the command **no blends** prevents the list from being read. This can speed up the code when running large grids of single-zone models.

19.11.4 No buffering

Programs produce output by first writing into a buffer. Information is written to disk once the buffer is full. If a C++ program crashes before this buffer is "flushed" the information within the buffer will be lost. This poses a problem if the printout generated just before the crash is needed for debugging. The **no buffering** command will redirect the standard output to `stderr` which is not buffered. **Save** commands also have a **no buffering** option to turn off file buffering for the files they create.

On most systems `stderr` can be directed to a file by redirection of file descriptor 2 as in the following example (this assumes you are running a **bash** shell or something equivalent).

```
cloudy.exe < sim.in > sim.out 2> sim.err
```

When the **-r** or **-p** command line flag is used, the situation is different. In parallel runs, multiple output files would be open at once and redirecting to `stderr` would create havoc. In this case, the output file is closed and immediately reopened again in append mode. Buffering can then safely be turned off. In this case you run the code as you would normally.

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

19.11.5 No charge transfer

This turns off all charge transfer interactions.

19.11.6 No collisional ionization

This turns off collisional ionization from ground states of all atoms and ions treated as an equivalent two-level system (Ferland et al., 2017). Specifically, this disables the collisional ionization rate coefficients computed with the Voronov (1997) results. This should only be used in experimental conditions.

19.11.7 No Compton effect

This command turns off Compton heating and cooling of free electrons and Compton recoil ionization of bound electrons. Electron scattering opacity is *not* turned off.

19.11.8 No CTHeat

This turns off heating due to charge transfer. Kingdon and Ferland (1999) describe the process.

19.11.9 No diffuse line pumping

The diffuse continuum produced by gas within the cloud is included as a general line-excitation mechanism. This turns it off and is useful as a check on the importance of the process. (This command is currently disabled. The code will issue a statement to that effect and exit if this command is entered. Pumping by local diffuse emission is not included. Pumping by diffuse emission from previous, downstream zones is always included.)

19.11.10 No FeII pumping

This turns off H I Lα pumping of Fe II.

19.11.11 No file opacity

The code can generate a file of stored opacities with the **compile opacity** command. This file will be used to generate the opacities in later calculations. The **no file opacity** command tells the code to ignore this opacity file even if it exists.

19.11.12 No fine structure line optical depths

Fine structure lines, such as the ^3P 52, 88μm lines of [O III], can become optically thick under certain high-luminosity conditions (see, for example, Rubin, 1983). They can absorb the incident continuum and be a significant heating source for photodissociation regions (Tielens and Hollenbach, 1985a). Radiative transfer effects, including stimulated and maser emission, are fully treated for all lines. This turns off optical depths and line transfer for fine structure lines by setting the line opacity to zero.

This command does not turn off line heating—which will then be maximized since the lines will remain optically thin. The **no induced processes** command turns continuum pumping off for all lines and should be included to disable heating due to line absorption of the continuum.

The line transfer arrays are permanently injured by the **no fine structure** command. Subsequent runs with the same core load in a grid will still have the line optical depths disabled.

19.11.13 No free free

Free-free heating, cooling, and emission are turned off with this command.

19.11.14 No grain [process]

This series of commands disables various processes involving grains.

no grain electrons This turns off the effects of grain charging, either negative or positive, on the overall charge balance in the gas. This violates charge conservation.

no grain gas collisional energy exchange Collisional energy exchange between grains and the gas is treated as in van Hoof et al. (2004) and Abel et al. (2005). This processes acts to bring the grains and gas toward the same temperature. The two temperatures will be equal if collision energy exchange is fast. Often the gas is hotter than the grains in the H^+ region and this process heats the grains and cools the gas. In fully molecular regions typically the reverse is true as the remaining continuum is very ineffective in heating the gas, but can still heat the grains. This command disables this physics.

no grain molecules Condensation of molecules onto cold grain surfaces was added by Nick Abel and is treated using the formalism outlined by Hasegawa et al. (1992) and Hasegawa and Herbst (1993). This turns off that process.

no grain neutralization Ions can become neutralized or ionized, by several stages of ionization, following a collision with a grain. The treatment of these collisions is described in van Hoof et al. (2004). It is partially based on Draine and Sutin (1987)'s description of Coulomb interactions between the grains and gas. The effects of grain neutralization upon the ionization balance of the gas are turned off with this command although grain charging is still computed. This is not self consistent.

no grain physics This turns off all grain physics. The only remaining effect of grains is their continuous opacity. The resulting simulation will be much faster but incorrect. This is only intended to provide a method to do quick and dirty calculations. The spectrum emitted by the grains is not calculated so this command violates energy conservation.

no grain qheat This turns off quantum heating for all grain species.

no grain x-ray treatment This turns off the extensive treatment of inner shell X-ray processes within grains. This makes the X-ray physics more like Weingartner and Draine (2001a).

19.11.15 No induced processes

This turns off all processes involving the photon occupation number, including induced recombination and its cooling, stimulated two-photon emission and absorption (Bottorff et al., 2006), continuum fluorescent excitation, photoexcitation, and stimulated emission of all lines.

19.11.16 No ionization reevaluation

This tells the code to not constantly reevaluate the ionization rates. This option can only be set when constant density is assumed.

19.11.17 No isotropic continua report

By default, the code reports the total outward flux emitted by a cloud, including any attenuated isotropic continua. This is effectively what telescopes observe in the optical through X-ray bands. However, in the infrared through radio bands, it is customary (if not automated) to remove any background radiation with a variety of methods (e.g., source or frequency switching). This is part of the *Herschel* pipeline, for instance.

This command removes isotropic continua from the *reported* continua and band fluxes, to ease comparison against observations at longer than IR wavelengths. This only affects the output: Isotropic sources of radiation are still included in the physics problem CLOUDY solves.

This command affects the output produced by some **save continuum** commands, as described in section 16.42.4, as well as the fluxes through the bands described in Section 2.2.3 *The radiation field integrated over a range of wavelengths*, and defined in the file `data/continuum_bands.dat`. It has no effect on the continua reported in time-dependent integrations.

19.11.18 No line transfer

This turns off line transfer for all lines except the Lα transitions of the species treated with the iso-sequences. This is the approximation made in most codes designed to consider classical nebulae and may be appropriate in some cases. This command also turns off line radiation pressure.

19.11.19 No lines isotropic continuum subtraction

By default, reported line intensities are modified by the diminution factor due to pumping by isotropic continuum radiation, most notably the CMB. The physics of the diminution is explained in detail in Chatzikos et al. (2013), in connection to the escape probability formalism. The main effect is best understood in a two-level atom, where isotropic continuua sustain an upper level population that reduces the number of spontaneous deexcitations relative to the non-isotropic pumping case, which is driven by collisions. This diminution factor has been computed in connection to hyperfine structure lines by D'Cruz et al. (1998), and has been invoked ad hoc by Goldreich and Kwan (1974).

With this command the correction is not applied. This command affects the output of **save lines cumulative**, **save lines intensity**, and **save linelist**.

19.11.20 No Lya 21cm pumping

Pumping of the H I 21 cm line by Lα is treated as in Deguchi and Watson (1985). This turns off the process. All three keywords must match.

Note that the text **21cm** is interpreted as a keyword and not as a precise wavelength of the line.

19.11.21 No OTS [options]

no Lya OTS This turns off H I Lα OTS rates.

no HeII OTS This turns off He II Lα and recombination continua OTS rates.

no line OTS This turns off all line OTS rates.

19.11.22 No level 2 lines

This turns off the large block of Opacity Project ("level 2") lines. This should only be done in exploratory "quick look" calculations. The lines should be included for final results. These lines should not be disabled at higher densities since they may carry a large fraction of the cooling as the cloud approaches the black-body limit. They are important in UV photoexcitation of excited levels within split ground terms like O I* or C II*. Excited-state column densities will not be predicted if the level 2 lines are disabled.

19.11.23 No molecules [heavy]

CLOUDY includes a chemistry network for predicting molecular abundances that is described in Abel et al. (2005) and Röllig et al. (2007). The **no molecules** command will completely turn off this network. If the keyword **heavy** is also present on the command line, the chemistry network will be severely limited by excluding any molecular species containing elements heavier than hydrogen.

19.11.24 No on the spot

This turns on all ground-state recombination coefficients and turns off ionization by helium resonance lines. This last variable is then used to deduce the ionization efficiency of lines and continua.

19.11.25 No opacity reevaluation

Opacities are normally reevaluated every time the cooling function is reevaluated. When this is set opacities are only evaluated one time per zone.

19.11.26 No outward [options]

no outward lines This turns off the outward-only (Tarter, 1967) contribution by lines.

no outward continuum This turns off the outward-only contribution by continua.

19.11.27 No photoionization

This turns off photoionization of the ground states of all elements. It is designed to test the code in the collisional-ionization equilibrium limit.

This command can cause problems. Several iso-sequences are treated as full multi-level atoms. This includes all processes that affect internal excitations. The level-populations solver may have difficulty in finding a solution when photoionization is turned off with this command but the gas is very weakly collisionally ionized. Often internal continuum pumping of the Lyman lines will dominate excitations. When these radiative terms overwhelm the collisional-ionization term the ionization rates out of the ground state may underflow. In this case the solver will find negative level populations and the code will stop announcing a catastrophic failure. One solution is to turn

off continuum pumping, with the **no induced processes** command. Only do this if the calculation fails.

19.11.28 No radiation pressure

This command turns radiation pressure completely off. Radiation pressure due to trapped lines will be counted in the total pressure when the constant pressure option is used. The default is for a constant-density model. Radiation pressure is not included if constant gas pressure is specified.

19.11.29 No recoil ionization

This command turns off Compton recoil ionization. Compton heating and cooling of free electrons is included, but this is the only remaining thermal effect of electron scattering. Bound-electron scattering opacity is still included when this command is issued.

19.11.30 No scattering opacity

This command turns off all scattering opacity sources in the calculation. It is described in detail in Section 10.8 above. This can have a major effect on the physics of the simulation.

19.11.31 No scattering escape

This turns off line escape following scattering off a free electron. Free electrons move far more quickly in a thermal gas than nuclei. A line photon scattered off an electron will receive a large Doppler shift away from line center and escape the cloud. This command turns off that process.

19.11.32 No absorption escape

Lines can be removed by absorption by background opacities such as grain or photoelectric absorption. This is a true absorption process and the line energy goes into heating the grain or ionizing the gas. This command turns off that process so line photons are no longer destroyed by absorption opacities.

19.11.33 No scattering intensity

Line escape following scattering off a free electron is still included as a photon escape mechanism but is not included in the reported line intensity due to it large line width.

19.11.34 No secondary ionizations

This command will turn off the effects of knock-on supra-thermal electrons. Normally these are treated as in Spitzer and Tomasko (1968), Bergeron and Souffrin (1971), Shull (1979), Shull and van Steenberg (1985), Xu and McCray (1991), and Dalgarno et al. (1999). This command will make X-rays 100% effective in generating heat and produce no secondary ionizations or Lα excitations.

19.11.35 No Stark broadening

Stark broadening (important for densities larger than $\sim 10^{10}\text{cm}^{-3}$) is treated for hydrogen lines using the formalism described by Puetter (1981). This turns Stark broadening off.

19.11.36 No TePredictor

The code tries to predict the temperature of the next zone for constant density calculations. This stops the predictions from being used.

19.11.37 No static opacities

This forces all opacities to be reevaluated constantly within each zone. The default is to only evaluate minor opacities one time per zone.

19.11.38 No three body recombination

This turns off three-body recombination for the heavy elements that are treated as equivalent two-level systems. Three-body recombination for atoms and ions of the isoelectronic sequences is turned off by turning off collisional ionization (its time reversal) with the **database [H-like or He-like] collisional ionization off** command.

19.11.39 No times

This turns off printout that would introduce differences between output files which have identical results. The code will not report execution times or one-time notices about initializing databases. This allow exact text comparisons of results between models run at different times or on different machines.

19.11.40 No vary

This command turns off the **vary** option set on various optimization commands.

19.12 Monitor commands

These tell the code a set of results that are expected. At the end of the calculation the expected and actual results are compared and an error condition is returned if the two disagree by more than an uncertainty. This set of commands is the foundation for automated testing of the code. In Lexington the full test suite is computed every night and a log is kept of all monitored results. An email warning is sent if any monitor fail. This insures the reliability of the code and guarantees that bugs are caught almost as soon as they are introduced.

If the characters "<" or ">" appear on the command line the expected result is taken as an upper or lower limit. The default is to check for equality.

The relative error is the second optional number on the line. If this is not specified then a relative error of 0.05 will be assumed.

19.12.1 Monitored results in the main output

The monitored quantities will be reported at the end of the main output, in the following format.

```
                  Label         line    computed     asserted Rel Err Set err
ChkMonitor        H  1       4861.36A  -11.2479 =   -11.2530  -0.012   0.050
ChkMonitor -      H  1       3889.07A    0.1084 =     0.1070  -0.013   0.050
ChkMonitor        Ca B       3889.07A    0.1046 =     0.1050   0.004   0.050
ChkMonitor -      He 1       5875.64A    0.1494 =     0.1470  -0.016   0.050
ChkMonitor        Q(H)       4861.36A    0.8953 >     0.7800  -0.148   0.050
```

The **save monitors** command, described in section 16.38, can be used to save these in an external file.

19.12.2 Monitor set error 0.1

This provides a way to change the default relative error for all monitors of physics-based quantities that occur later in the input stream. The default error is normally the second optional number on each **monitor** command. The default is changed when this command is parsed so this should be the first **monitor** command in the input stream if you want to change the default error for all the monitors.

The majority of the monitors confirm quantities that are related to predictions. There are a special set of **monitor** commands which track the code's performance. The **monitor itrzn** command is an example. A default relative error of 0.2 is assumed for performance monitors. This default can be changed with the **Monitor set performance error xxx** command.

19.12.3 Monitor ⟨subcommand⟩ grid "filename"

For each of the commands discussed below you can add the keyword **grid**. If this keyword appears, then a file name should be supplied between double quotes. This file should contain a series of quantities to be monitored. There must be one value for each grid point in a grid calculation (the **grid** command is described elsewhere). The numbers in the file have the same meaning as the number on the command line when the **grid** keyword is not used. See the discussion for each subcommand for further details. The file should not contain any other text, but may contain comments starting with "#". The numbers should be separated by whitespace and/or newlines.

19.12.4 Monitor Case B [H-like, He-like] element error

The H and He recombination emission from lower density photoionized clouds that are optically thick in the Lyman continuum may be close to Case B values (AGN3). This command uses stored tables of Case B emission coefficients to compare the code's computed emission with Case B expectations. Assuming that the cloud conditions are set up properly, and that certain physical processes are disabled, this provides a way to check the accuracy of models of some atoms.

One of the iso-sequences, H-like or He-like, must be specified. The default relative error can be changed by specifying a number on the line. The code will compare Case B with computed values for all lines that are included in the emission-line output and a botched monitor will be declared if

any of the code's predictions differ from Case B predictions by more than this error. A line will be printed that gives the mean error for the Case B comparisons.

monitor Case B H-like [element error] The H-like option will compare the code's predictions with the Storey and Hummer (1995) results. An element name, any between hydrogen and oxygen, must also appear.

monitor Case B He-like error This will compare the predictions with Case B predictions for He I emission given by Porter et al. (2012, 2013). The Case B values have the label "+Col" in the main emission-line printout. The element name must still appear but currently only helium itself can be tested.

The range option The keyword **range** tells the code to only check the monitored values over a restricted wavelength range. The wavelengths can be entered in the standard CLOUDY format. Interpretation as air or vacuum wavelengths can be forced by appending the optional keyword **air** or **vacuum** directly behind the wavelength. Note that the keyword only applies to the wavelength directly preceding it, so it needs to be repeated, as shown below.

```
monitor Case B H-like hydrogen 0.15 range 1000A vacuum to 4.00m air
```

This was added as a way to exclude the intensities of unobservable MIR - FIR lines. These lines come from higher levels in the atom and generally have a larger error due to the finite size of the model.

19.12.5 Monitor chemistry searches < 3

This performance monitor checks the convergence properties of the chemistry solver. The quantity is the mean step length searches/chemistry step as reported in the main output under the header "Convergence statistics". This monitor will never trip in release code.

19.12.6 Monitor chemistry steps < 8

This performance monitor checks the convergence properties of the chemistry solver. The quantity is the mean steps/chemistry solve as reported in the main output under the header "Convergence statistics". This monitor will never trip in release code.

19.12.7 Monitor column "species" 18.9

This checks the predicted column density for the molecular species included in quotes.

An alternative form with unquoted keywords **CO**, **H-**, **H2**, **H2+**, **H3+**, **H2g**, **H2***, **HeH+**, **O2**, **SiO**, **C2**, **C3**, **OH**, **CN**, **CH+**, or **CH** is available at present but it is planned will be removed. To use this form, the labels must be surrounded by spaces. It is important that the log of the column density [cm^{-2}] appear *after* the label because of the 2 in **H2** or **O2**, etc.

The command accepts a **log** option to interpret the error as a log. The column density itself is always interpreted as a log.

For H_2 there is a **levels** option which tells the code to return the column density in a specific v, J level. This only works when the large H_2 molecule is included. The numbers on the line must appear in the following order: first the 2 in **H2**, then v, and then J, and finally log $N(H_2)$. The "wavelength" of the monitored quantity will be $100v+J$.

19.12.8 Monitor csupra −17.09

The secondary ionization rate is given by the variable *csupra*. This command checks the value of the secondary ionization rate. The number is the log of the secondary ionization rate [s^{-1}].

19.12.9 Monitor Ctot -12.5

This checks that the local cooling rate [$\mathrm{erg\,cm^{-3}\,s^{-1}}$] of the last zone is equal to the expected value. The argument is the log of the cooling rate.

19.12.10 Monitor departure coefficients mean=1, error=0.04

This confirms that the departure coefficients predicted by one of the model atoms are correct. The computed quantity is the mean departure coefficient for all levels in the model atom and is the first number on the line. The maximum allowed error from the mean is the second number on the line.

An alternative form for the command may be used to test against upper and lower limits for the departure coeffient. For example,

monitor departure coefficients "C[4]" $>$ **0**
monitor departure coefficients "C[4]" $<$ **1**

may be used to test the usual case that the departure coefficient is within the range 0–1. No error is expected in this use case.

monitor departure coefficient "C+4" If a species (in chemical notation) is given in quotes, as `"O+6"` or `"H2O"`, then departure coefficients for that species will be checked. The species designation may include a range of levels, e.g., "C+4[3:10]", or a specific level, e.g., "C+4[3]". The rules of Section 16.86.1 apply.

monitor departure coefficients, H-like helium, error=0.08 The keyword **H-like** tells the code that the departure coefficients predicted by an atom of the hydrogenic isoelectronic sequence should be checked. The name of one of the elements must also appear on the command line.

If the keyword **excited** appears then the ground state will not be included. The keyword **ZeroOK** allows departure coefficients of exactly zero to pass the test. This is useful in large grids of calculations where the species being monitored actually has zero abundance at some points in the grid.

monitor departure coefficients, He-like helium, error=0.08 The keyword **He-like** tells the code to check the predictions for the specified element of the helium isoelectronic sequence. It operates like the H-like keyword, and accepts the **excited** and **ZeroOK** keywords, as well.

monitor departure coefficient, hminus, error=0.08 The keyword **hminus** tells the code to check the departure coefficient of H^-.

19.12.11 Monitor depth 13.2

This checks that the cloud's thickness [cm] is equal to the expected value. The argument is always the log of the depth. The thickness is the length between the illuminated and shielded faces of the cloud.

19.12.12 Monitor eden 9

The number is the log of the electron density [cm^{-3}] of the last zone.

19.12.13 Monitor grain potential

The grain bin number is the first number on the line. Next comes the grain potential. This is linear eV and can be positive or negative. The third number is the optional error.

19.12.14 Monitor H2 grain rate -16.5

The number is the total rate [$cm^{-3}\ s^{-1}$] that H_2 forms on grain surfaces for the last computed zone. The number is a log if it is negative.

19.12.15 Monitor H2 ortho/para ratio 2.02

This checks the ratio of the ortho to para H_2 densities in the last computed zone. This is only meaningful if the large H_2 molecule has been turned on.

19.12.16 Monitor HHeicf -0.013

This checks the helium—hydrogen ionization correction factor. The number is the linear difference between the atomic fractions of helium and hydrogen.

19.12.17 Monitor Htot -13.2

This checks that the local heating rate [$erg\,cm^{-3}\,s^{-1}$] of the last zone is equal to the expected value. The argument is the log of the heating rate.

19.12.18 Monitor ionization fraction oxygen 3 -3.45, error 0.1, weight = radius

This checks the mean fractional ionization of an ionization stage of an element. An element name must appear somewhere on the line. The first number is the ionization stage, 1 for the atom, 2 for the first ion, etc. The second number is the expected ionization fraction. It is interpreted as a log if negative and as the linear ionization fraction if it is positive. The average can be with respect to radius or volume. The default is an average weighted with radius. If the keyword **volume** appears then the average will be weighted over volume.

If the keyword **grid** appears then a file name should be supplied between double quotes. This file should contain a series of ionization fractions to be monitored. There must be one value for each grid point in a grid calculation (the **grid** command is described below). The file should not contain any other text, not even comments. The numbers should be separated by whitespace and/or newlines.

An optional error may be supplied on the command line. The same error is assumed for each grid point.

19.12.19 Monitor itrmax < 20

This performance monitor checks the convergence properties of a calculation. The quantity is the maximum number of calls to *ConvBase* in any given zone and iteration (including the initial search phase). This monitor will never trip in release code.

19.12.20 Monitor itrzn < 3.5

This performance monitor checks the convergence properties of a calculation. The quantity is the number of iterations required to converge each zone. This monitor will never trip in release code.

19.12.21 Monitor line [Case B] "H 1" 4861.33 < 1.01

This checks the emission in a line. The string that gives the line label must appear between two double quotes. The line wavelength is the first number after the label. Both label and wavelength must appear exactly as they do in the output produced by the **save line labels** command or in the standard emission-line output. These two entries may optionally be followed by additional information to disambiguate the line. The information given in Section 2.5.3 applies here as well, with the obvious exceptions that the line label should not start in the first column, and that additional information is required after the line information, as described below.

If the keywords **Case B** appear, the code will compute the ratio of the intensity of the specified line over the Case B intensity of the same line, which is interpolated from the Storey and Hummer (1995) tables. Compared to **monitor Case B [H-like, He-like]**, which goes over all lines on the tables, this command allows for individual lines to be monitored.

The sub-keyword is actually "**line**_" to avoid confusion with the keyword "**linear**" that appears on some commands. The command will not be recognized if the trailing space is missing.

The second number that appears in the command is the expected intensity of the line. Its meaning is as follows:

- *Relative intensities*

  If the keywords **luminosity** and **intensity** do not appear, the second number is interpreted as the emission line intensity relative to the normalization line intensity. It is always a *linear* quantity.

- *Absolute intensities*

  If the keyword **luminosity** or **intensity** appears, then the absolute luminosity [erg s^{-1}] or intensity [erg cm^{-2} s^{-1}] of the line is checked. The monitored quantity is the *log* of the

luminosity or intensity of the transition, and not the intensity relative to the normalization line. For example,

```
monitor line [luminosity intensity] "Q(H)" 4861 38.91
```

- *Line types: intrinsic, emergent, and cumulative*

  We predict four types of line intensities: intrinsic, emergent, and their cumulative (time-integrated) counterparts, i.e., cumulative intrinsic, and cumulative emergent. The first two are described in Hazy 2, Section 2.10, *Line intensities in a dusty cloud*. The cumulative intensities are integrated according to the weight set by the **set cumulative** command, described in full in Section 19.13.13.

  In the absence of any of these keywords, the monitored intensity is interpreted as **intrinsic**. If only the keyword **cumulative** appears, the intensity is interpreted as **cumulative intrinsic**. Checks against the (instantaneous, non-integrated) emergent intensity are issued with the **emergent** keyword, while checks against the time-integrated emergent intensity are issued with the **cumulative emergent** keywords.

- *Upper or lower limits*

  Upper and lower limits may be requested by using the usual '<' and '>' symbols, respectively, in front of the desired value, as in:

```
monitor line luminosity "Q(H)" 4861 < 38.91
```

19.12.22 Monitor molecular fraction H2 -3.45, error 0.1

This checks the mean molecular fraction. The label for a molecule, currently only H_2, must appear somewhere on the line. The next number is the expected molecular fraction. If this number is ≤ 0 it is interpreted as a log and as the linear ionization fraction if it is positive. The average can be with respect to radius or volume. The default is an average weighted with radius but if the keyword **volume** appears then the average will be weighted over volume.

19.12.23 Monitor mpi

This checks that the code is running in MPI mode. The command is used in the `mpi` test suite. It has no additional parameters.

19.12.24 Monitor niter $<$ 4

This checks the number of iterations required in a calculation. This is usually an upper limit. This monitor will never trip in release code.

19.12.25 Monitor nothing 0

This will enter a clean monitor into the output. This is used for those simulations in the test suite which have nothing to monitor but are included for general testing.

19.12.26 Monitor nzone $<$ 135

This checks how many zones were needed. This is usually an upper limit. This monitor will never trip in release code.

19.12.27 Monitor PRadMax 0.34

The monitored quantity is the maximum ratio of radiation to gas pressure.

19.12.28 Monitor pressure error $<$ 0.01

The number is the ratio of the standard deviation of the pressure to its mean value. It is interpreted as a log if it is negative.

19.12.29 Monitor radius 18.2

This checks that the outer radius [cm] of the computed model is equal to the expected value. The radius is the distance from the center of the central object to the shielded face of the cloud.

19.12.30 Monitor temperature "species" 4.02 volume, error 0.01

This checks the mean temperature of any species. An ionic or molecular species is expected within quotes, e.g., "H", "He+2", or "CH2+". If the species is recognized, the code will compare the computed mean temperature for that stage of ionization with the monitored value. The number on the line is the temperature, interpreted as a log if it is ≤ 10. It will be linear if the keyword **linear** appears. The last number is the relative error.

The code computes means weighted over radius and over volume. If the keyword **volume** appears then the temperature will be compared with the volume-weighted mean. The default is weighting over radius.

If the keyword **grid** appears, then a file name should be supplied between double quotes. *The file name must be given before the double-quoted species.* The syntax is:

```
monitor temperature grid "InputFile.txt" "species" 4.02 volume, error 0.01
```

This file should contain a series of temperatures to be monitored. There must be one value for each grid point in a grid calculation (the **grid** command is described elsewhere). The file should not contain any other text, not even comments. The numbers should be separated by whitespace and/or newlines. An optional error may be supplied on the command line; the same error is assumed for each grid point. For example,

```
monitor temperature grid "ism_grid.dat" "H+" .05
```

19.12.31 Monitor 21cm temperature [mean, spin, optical] 50 error 0.1

This monitors the temperature of the 21cm hyperfine transition of hydrogen. The **mean** option tracks the kinetic temperature of the gas weighted by hydrogen, **spin** tracks the spin temperature

of the 21cm line, while **optical** tracks the temperature of the gas computed from the ratio of the optical depths in the Lα and 21cm lines, see AGN3, section 5.5.

The first number is the temperature, interpreted as a log if it is ≤ 10. It will be linear if the keyword **linear** appears. The last number is the relative error.

Note that the text **21cm** is interpreted as a keyword and not as a precise wavelength of the line.

19.12.32 Monitor grain temperature index 2, temperature 234

If no element name appears but the keyword **grains** does then the code will compare the computed and monitored grain temperatures. The first number on the line is an index giving the grain type. This index is from the order in which the grains were specified in the input stream and is given in the output. The first grain that occurs in the input stream is number 1. The second number is the temperature and is interpreted as a log if ≤ 10. The **linear** keyword forces smaller numbers to be interpreted as linear quantities. The optional error is the third number on the line. The temperature is always averaged over radius.

19.12.33 Monitor temperature at face 11400K

This checks the temperature at the illuminated face of the cloud.

19.12.34 Monitor thickness 13.2

This checks the thickness [cm] of the computed structure. The argument is always the log of the thickness. The thickness is the length between the illuminated and shielded faces.

19.12.35 Monitor velocity 7.6 km/s

This checks the final velocity of a wind model. The quantity is the expected final velocity in km s^{-1}.

19.13 Set commands

These commands change internal variables used by CLOUDY. These are mainly intended for testing and are not normally used.

19.13.1 Set Lya 21cm [excitation, kinetic, constant]

This changes the form of the Lyα source function S_ν at line center. The shape of the source function has a major impact on the excitation and resulting spin temperature of the 21 cm line. Three options are available. The first two are characterized by a Planck function at one of the temperatures that are present, while the last shape assumes $S_\nu \sim constant$.

The Lyα excitation temperature, as set by the $n(2p)/n(1s)$ population ratio, is the default and can be selected with the keyword **excitation**. The gas kinetic temperature is used if the keyword

kinetic appears. A frequency-independent source function is assumed if the keyword **constant** appears.

Note that the text **21cm** is interpreted as a keyword and not as a precise wavelength of the line.

19.13.2 Set 12C13C -3.2 – DEPRECATED

This command has been deprecated. Use **element carbon isotopes** instead.

19.13.3 Set blend ["Custom"] [4622.23 [air | vacuum]] [quiet]

With this command you can define a custom blend of lines. In order to identify the blend, you need to enter a label between double quotes and a wavelength. If the label is omitted, it will default to "Blnd". If it is too long, it will be truncated to 9 characters. If the wavelength is omitted, it will default to the $g_k A_{ki}$-averaged wavelength of the blend components. The latter is only meaningful if the blend components are all from the same ion, which in general need not be the case. If the wavelength is specified on the command line, you can add an optional keyword **air** or **vacuum** to indicate whether the specified wavelength is in air or vacuum. This keyword needs to be typed immediately after the wavelength, with at least one space between the wavelength and the keyword. If the keyword is omitted, the wavelength type will be assumed to be the same as the one used in the main output. Note that this keyword only applies to the wavelength of the blend itself and is not used for the individual blend components. How to enter these components is described below.

Each emission line included in the blend is entered on its own line. This list begins on the line after the **set blend** command and continues until a line with **end** in the first three columns appears. The format for entering the spectral lines is described in Section 2.5.3. An arbitrary number of lines can be entered into the blend, and there is also no limit to the number of blends that can be defined in this way. An example is shown below.

```
set blend 3968
Ne 3  3967.47
H  1  3970.07
end
```

The blend components need to have type 't' as defined in the **save line labels** output and also be associated to a transition. The latter can be recognized from the fact that the comment in the **save line labels** output has entries for "index" and "Elow". The custom blends will be stored on the line stack and can be used like any other entry on that stack.

If the keyword **quiet** appears on the **set blend** command line, and the database model for any of the species in the blend has been turned off, the blend will be quietly skipped rather than producing error messages about missing lines. This is useful if the **set blend** command is included in an init file that will be shared between many scripts. However, note that the code cannot discriminate between a species that has been disabled and a typo in the label, so make sure that the labels are correctly typed before using this keyword.

An init file called `blends.ini` with commonly used multiplets and blends is included in the data directory[2]. CLOUDY automatically parses this file unless you include the **no blends**

[2]For CLOUDY versions C17 and older these blends were hardwired into the source code.

command in your script. It is possible to modify this file to add your own blends (but note that this file is used by the test suite, so modifying or deleting existing blends could affect subsequent test suite runs). You can also use this file as a template to create your own init file with custom blends, or you can enter the blends directly in your script.

19.13.4 Set charge transfer -11.5

This command establishes the coefficient in the statistical hydrogen charge transfer rate coefficients used for species more than four times ionized (Ferland et al., 1997). If the number is negative then it is assumed to be the log of the coefficient, if zero then this estimate is turned off, and if positive the number is the coefficient itself. It is stored as the variable *HCTMin* and has the default of 1.92×10^{-9} cm^3 s^{-1}. This is used to set a rate coefficient of *HCTMin %q* where q is the excess charge of the heavy element.

19.13.5 Set chemistry

This changes some aspects of the chemistry network.

set chemistry Federman [ON , OFF] This determines whether the rate coefficients derived by Zsargó and Federman (2003) are used. Their chemical network consists of a small subset of reactions important to the formation of CH, CO, HCO^+, CH_{2+}, and CH_{3+} in regions with large amounts of CH^+. The default is "**ON**", that is, to use these rates instead of UMIST in the reaction network. If **OFF** is specified then UMIST rates are used.

set chemistry non equilibrium This uses the chemical model of Federman et al. (1996) to examine the effects of nonthermal chemistry on diffuse cloud chemistry. This command uses an effective kinetic temperature for a chemical reaction that is caused by ion / molecule slip when turbulence in present. The effective temperature is given by

$$T_{eff} = \frac{3}{2}kT + \frac{1}{2}\mu\Delta u_{turb}^2 \tag{19.1}$$

where k is Boltzmann's constant, T is the gas kinetic temperature, μ is the reduced mass of the two reactants, and Δu_{turb} is the turbulent velocity. Zsargó and Federman (2003) summarize the physical mechanism of non-thermal chemistry as:

> "The origin of the nonthermal motion is assumed to be the result of the propagation of Alfvén waves, constantly entering the cloud from the intercloud medium. The dissipation of these waves is concentrated primarily in a diffuse cloud boundary layer, the transition zone between the cloud and the intercloud medium."

The effect of T_{eff} is to increase the rates of those reactions which have a strong $\exp(-c/T)$ temperature barrier. This command primarily effects the formation of CH^+ (the abundance of which cannot be explained by equilibrium chemistry models) by eliminating the temperature barrier for the $C^+ + H_2 \rightarrow CH^+ + H$ formation channel, allowing CH^+ to form. The formation of CH^+ has trickle down effects on the predicted abundances of other

molecules, such as CH, C_2, CN, and OH (see equations 7 & 8 of Zsargó and Federman, 2003). This option was added by Nick Abel. The keyword can be **non equilibrium** or **non-equilibrium**.

set chemistry non equilibrium neutrals [ON , OFF] If the option **neutrals off** is also included then T_{eff} will equal T for all neutral-neutral reactions, so non-equilibrium chemistry is not included for these species. The physical assumption is that neutrals are not coupled to the magnetic field and therefore do not experience the effects of nonthermal chemistry. This was used by Federman et al. (1996) to explain low OH abundances in regions with high CH^+, such as ξ Per. By default, neutral-neutral reactions are included in the non-equilibrium chemistry.

set chemistry proton elimination [ON , OFF] In "proton elimination" reactions, an atom-molecule non-charge transfer reaction leads to the formation of either H or H^+. An example is C^+ + OH which leads to either CO + H^+ or CO^+ + H. Huntress (1977) states that the energetics of the reaction makes the H^+ formation channel highly unlikely. This command sets the rates of these reactions leading to H^+ (which currently only includes the example) to zero. By default, proton elimination is included.

19.13.6 Set check energy every zone

The code checks that it has not used more energy than is available in the incident radiation field when the calculation ends. This command will cause that check to occur for every zone. This does slow down the calculation somewhat, so it a debugging tool to use when energy conservation problems occur.

19.13.7 Set collisional ionization [Dima, Hybrid]

The collisional ionization rates of Voronov (1997), as implemented by Dima Verner soon after publication of that paper, were used through version C10. These are complete over the range of temperatures considered by CLOUDY. Dere (2007) presents updated rates but these do not cover the temperature range we need. The Voronov (1997) and Dere (2007) data are generally in excellent agreement. We scale the Voronov (1997) rates by the ratio of the Dere (2007) to Voronov (1997) to obtain updated rates over the full temperature range. We refer to this as the hybrid scheme. The scaling factor for each species is the Dere coefficient from the center of the temperature range given in Dere (2007) divided by the Voronov (1997) coefficient at that same temperature. The **set collisional ionization Dima** uses the Voronov (1997) rates while **set collisional ionization Hybrid**, the default, uses thsi scaling.

19.13.8 Set collision strength averaging on

This controls whether the l-mixing collision strengths used for hydrogenic levels are evaluated at kT (the default) or are an average over the thermal Maxwellian. If the keyword **on** appears then averaging will be done—this is very slow due to the large number of levels and evaluations involved. Tests show that the default, the evaluation at a single energy of kT, is significantly faster and produces very similar predictions.

19.13.9 Set continuum options

set continuum resolution 0.1 This changes the resolution of the coarse continuum energy mesh by a constant scale factor. The resolution of the continuum mesh as a function of energy is defined by the data file `continuum_mesh.ini`, which is located in the data directory. Permanent changes to the continuum resolution should be made there. This command allows the continuum mesh to be changed by a temporary scale factor. The number on the command line multiplies the resolution used by the code. Factors less then unity make the cells smaller, for higher resolving power, while factors greater than unity make the resolution coarser. For instance, an entry of 0.1 would make the resolution ten times finer or the resolving power ten times greater. If the number is less than or equal to zero it is interpreted as the log of the resolution scale factor.

This was originally added so that the entire continuum resolution could be improved by large factors when running some of the fundamental tests of hydrogenic and helium-like emission. The code's execution speed is affected by the number of coarse continuum cells due to the frequent reevaluations of the opacity and photo-interaction rates. For factors near unity the execution time will scale roughly inversely linearly.

The code can be sped up by making the factor larger than unity. But there is a limit where the computed results will be strongly affected and the code may even no longer pass its internal sanity checks. A factor near 2 will cut the execution time by about 40%. This will affect the predictions and should only be done in exploratory calculations.

This will change the line to continuum contrast since emission lines are not resolved in the coarse continuum. See the discussion around Equation 19.2 for more details.

set continuum shielding This changes the treatment of shielding of the incident continuum by line optical depths. There are several options. **Federman** is the default and uses the function given in the appendix of Federman et al. (1979) with a bug fixed in the wing shielding; **Fbug** uses the original function from Federman et al. (1979) without the bug fix, as used by versions of CLOUDY up to 13.02; **Ferland** uses the function given by Ferland (1992); **Rodgers** uses a function derived from Rodgers and Williams (1974), as used by Draine and Bertoldi (1996). **Integral** uses an explicit numerical integral over the Voigt function, which should be highly accurate but is also very slow. **Pesc** uses the inward-looking escape probability (though note that the escape probability includes an integral over which is not needed in this case). **None** turns off self-shielding.

set FeII continuum The command is obsolete. See **set species continuum**, Section 19.13.47.

set fine continuum This changes the logic used to determine the resolution of the fine continuum mesh. The command has two arguments. The name of an element must appear on the line—the a tomic weight of this element will be used to determine the narrowest lines that must be resolved. The number of resolution bins over a HWHM must also appear on the line.

A **range** option can be used with the **set fine continuum** command. The first number is interpreted as the lower limit (in Rydbergs) of the fine continuum. An optional second number modifies the default upper limit. For both parameters, negative numbers are interpreted as logs. This command is experimental and can appreciably change results in conditions where line overlap is important.

19.13.10 Set convergence . . .

The error in the heating-cooling balance is set with the **set temperature convergence** command. The error in the electron density is set with the **set eden convergence** command. The error in the local pressure is set with the **set pressure convergence** command. In cases where the code is having trouble converging the **set presioniz** command will limit the number of ionization evaluations before declaring a failure.

19.13.11 Set coverage [fast]

This limits the number of zones and iterations that will be done so that the code runs more quickly. It allows quick runs of the test suite with tools such as *Purify* and *valgrind*. If the **fast** option appears then only one and one iteration will be performed. Otherwise up to ten zones and two iterations will be done. This limit is imposed on top of whatever other limits may exist in the input stream.

19.13.12 Set csupra = -12.34

This sets the H^0 secondary ionization rate due to supra-thermal electrons to the number on the line. The number is the log of the rate [s^{-1}]. The excitation rate of Lα is assumed to be the same. This option is used to test the code in secondary-ionization dominated cases. Normally the secondary ionization rate is determined self-consistently from the high-energy radiation field or the cosmic ray density.

19.13.13 Set cumulative [mass, flux, off]

Set the type of accumulation to be used for **save cumulative** output. Options are

- **mass**, which is appropriate for single-element cooling calculations and generates output in $\mathrm{erg\,g^{-1}}$,
- **flux**, which is appropriate for calculations on fixed domains and generates output in $\mathrm{erg\,cm^{-2}}$, and
- **off** which disables the cumulative calculation, and will generate an error if a 'save cumulative' output is encountered.

19.13.14 Set D/H -3.2 – DEPRECATED

This command is deprecated. Use **element hydrogen isotopes** instead.

19.13.15 Set density tolerance

Set the maximum relative error in the conservation of the gas phase density for each element, permitted by the ionization and molecular balance solvers.

19.13.16 Set didz 0.05

The thickness of the first zone is chosen so that the largest continuous optical depth through it is one percent of the entered value. Thereafter the zone thickness is continuously adjusted by checking that the optical depth at the maximum continuum-gas interaction energy is set to this value. The default is 0.15. If the value is less than or equal to zero, then it is interpreted as the log of the quantity, and linear if greater than zero.

19.13.17 Set dielectronic recombination [keywords]

This changes the treatment of various details of the ionic recombination physics. The **print recombination** command described on page 191 prints a summary of the data in use.

set dielectronic recombination mean [options] At present rate coefficients for dielectronic recombination through low-lying autoionizing states have not been computed for lower stages of ionization of most elements on the third row (of the periodic table) and higher. The **mean** option modifies the treatment of the estimates of the missing dielectronic recombination coefficients.

The default is to use means of all existing DR rates for each ion stage to replace missing data for that stage. For instance, $Fe^{+} \Rightarrow Fe^{0}$ does not have reliable DR rates. Its rate will be the mean DR rate for all first ions of all elements with published rates.

set dielectronic recombination mean off will turn off guesses for DR rates. These guesses are normally used for those ions that have no rate coefficients because no better can be done at present.

set dielectronic recombination mean scale will multiply the DR guesses by scale factors. One or more scale factors may appear on the line, and each will multiply the mean rate for successive ions. If there are fewer scale factors than ions then the last scale factor entered will be used for all higher ions. The scale factors must be greater than or equal to zero.

set dielectronic recombination mean noise [dispersion] will introduce Gaussian noise with the indicated dispersion. This provides a way to quantify the uncertainty caused by the need to use mean rates. This adds a component of Gaussian random noise to the means of the dielectronic recombination rate coefficients. The number on the line is the dispersion. The resulting random number will multiply the rate coefficient. As a result the same DR rates will be used throughout any single calculation but the noise multiplying the rate will change from run to run.

set dielectronic recombination suppression off will turn off suppression of dielectronic recombination. The DR rate depends on both the gas temperature and density, such that as the density increases the DR rate decreases. This is parametrized as a suppression factor relative to the zero-density value, according to Nikolić et al. (2013). Unless this command is issued, the suppression will be applied.

19.13.18 Set dr 11.2

This forces the zone thickness to a prescribed value. A space is required after the "**dr**_". The argument is interpreted as a logarithm by default. The argument is interpreted as the linear value if the keyword **linear** appears. If the keyword **relative** is present the value will be interpreted as a relative fraction of the current radius. In that case the value needs to be between 0 and 1. The default is that this command is not in effect and the adaptive stepsize algorithm will set the zone thickness in accordance with the local physical conditions.

Use this command with great caution! Forcing the code to take steps that are too large can lead to inaccurate results, or even destabilize the code.

19.13.19 Set drmax 11.2

This sets the largest allowed zone thickness [cm]. The zone thickness will not become larger than this. The argument is a logarithm if it is less than 38 or if the keyword **log** appears. It is linear if the keyword **linear** appears. If the keyword **relative** is present the value will be interpreted as a relative fraction of the current radius. In that case the value needs to be between 0 and 1. The default for the maximum zone thickness is 10^{30} cm.

19.13.20 Set drmin 11.2

This sets the minimum zone thickness [cm]. The zone thickness will not become smaller than this, except possibly in the last zone to avoid overshooting the outer radius. The argument is a logarithm if it is less than 38 or if the keyword **log** appears. It is linear if the keyword **linear** appears. If the keyword **relative** is present the value will be interpreted as a relative fraction of the current radius. In that case the value needs to be between 0 and 1.

The keyword **depth** works similarly to the keyword **relative** but sets the minimum zone thickness relative to the depth into the cloud. There is a default value of 10^{-5}. This minimum is not active if a **set dr** command is used.

The default for the minimum zone thickness is the greater of approximately 1.2×10^{-36} cm and the default relative to depth.

Use this command with great caution! Forcing the code to take steps that are too large can lead to inaccurate results, or even destabilize the code.

19.13.21 Set dynamics commands

These commands change aspects of dynamical or time-dependent calculations. They are described in 14 *DYNAMICAL & TIME-DEPENDENT CALCULATIONS*

19.13.22 Set eden [options ....]

These commands provide several ways to set the electron density or change its convergence criterion.

set eden 2.3 If no other keywords are recognized then this command sets the electron density. The number is the log of the density [cm^{-3}].

set eden convergence 0.01 This sets the convergence criterion for the electron density. The number is the largest relative error in the electron density. It is interpreted as the log of the relative error if it is negative. The default value is 0.01. The old form of the command, **set eden error**, is also recognized.

set eden fraction 1e-3 This sets the electron fraction, $n_e/n(\mathrm{H})$, where $n(\mathrm{H})$ is the density of hydrogen in all forms. The number is the electron fraction and is interpreted as a log if it is is ≤ 0.

set eden solver [vWDB , SECAnt] Switch between vWDB electron density solver (the default) and experimental limited secant solver.

Note: The **eden** command, described on page 274, add an extra source of electrons. These **set eden** commands set the electron density itself. These commands violate charge conservation.

19.13.23 Set flxfnt -20

The highest photon energy that must be considered is lower for relatively soft continua such as a star than for X-ray sources such as AGNs. The criterion used to choose the highest energy to be considered ν_{high} is that $\nu f_\nu(\nu_{high})/\nu f_\nu(\nu_{peak}) < \textbf{\textit{flxfnt}}$, where ν_{peak} is the frequency where the continuum reaches its maximum νf_ν. ***FluxFaint*** is normally 10^{-10}. This command changes the value of ***FluxFaint***. The argument is the log of the value.

19.13.24 Set gbar off

By default the g-bar approximation is used for transitions in the Chianti and Stout databases which have no collision strengths. G-bar values are taken from Mewe (1972) for both allowed and forbidden transitions. For simplicity, transitions with gf values greater than or equal to 10^{-8} are considered to be allowed.

This command says not to use g-bar for any database transition. If there is no collision data and gbar is turned off , then 0.01 is used for the collision strength.

19.13.25 Set grain heating factor 2.

This command allows you to enter a fudge factor to alter the amount of photoelectric heating from the grains (also when the **set PAH Bakes** command is in effect). The number you enter is a multiplicative factor and is always interpreted as a linear number. This command violates energy conservation and is intended for testing only.

19.13.26 Set H2 ...

This command changes some aspects of the H_2 molecule. Other aspects of this molecule are changed with the **database H2** command described on page 138 above.

set H2 fraction This sets the ratio $n(H_2)/n(H)$. The number is interpreted as the log of the ratio if it is less then or equal to zero and as the linear ratio otherwise. An upper limit of 0.5 will be silently imposed. This command turns off the normal H_2 molecular network and forces the H2 density to the hydrogen density multiplied by this fraction. This is totally unphysical and is only intended to provide an easy way to test the H_2 molecule.

set H2 formation scale This scales the H2 formation rate (via three-body recombination). The number is interpreted as the log of the scaling factor if it is less then or equal to zero and as the linear factor otherwise. In the fully molecular or fully ionized limits, this command should be used rather than the **set H2 fraction** command because it allows the same functionality without breaking numerous consistency checks.

set H2 Jura [TH85; CT02; SN99; ELRD] This changes the "Jura rate", the rate H_2 forms on grains (Jura, 1975). The default rates include the Eley-Rideal Effect (Le Bourlot et al., 2012; Röllig et al., 2013) as implemented by Röllig et al. (2013). These depend on grain temperature and abundance. The option **TH85** will use rates from Tielens and Hollenbach (1985a), **SN99** will use rates for Sternberg and Neufeld (1999), **CT02** uses Cazaux and Tielens (2002). Finally, **ELRD** will use the rates that include the Eley-Rideal Effect (Le Bourlot et al., 2012; Röllig et al., 2013) and is the default.

set H2 Jura scale The number is a linear scale factor that multiplies the formation rate obtained by one of the above methods. If the scale factor is negative or the keyword **log** appears then the number is interpreted as the log of the scale factor. Habart et al. (2004) suggest that the rate can vary across the galaxy by large factors.

set H2 Jura rate The number is the log of the H_2 grain formation rate k [cm^3 s^{-1}]. This is used to give a simple grain formation rate $R = k\,n(H_{tot})\,n(H^0)$ [cm^{-3} s^{-1}]. If no number appears on the line then the Jura (1975) rate of 3×10^{-17} will be assumed.

set H2 Solomon [BD96; TH85; ELWERT] This changes how the Solomon process is treated when the small H_2 molecule is used. The fits to the dissociation rate derived by Elwert et al. (2005) are the default and will be used if the keyword **ELWERT** appears. The keyword **BD96** will use Bertoldi and Draine (1996) and the keyword **TH85** will use the Tielens and Hollenbach (1985a) rate. These rates differ by ≥ 1 dex for some H_2 column densities. When the large H_2 molecule is used, as set with the **database H2** command, the Solomon process is treated in a self-consistent manner. Abel et al. (2004) show that H_2 column densities are quite sensitive to the correct treatment of this physics for translucent clouds.

set H2 continuum dissociation [Stancil; AD69] This command changes how direct photodissociation of H_2 is treated. By default, CLOUDY computes the photodissociation rate by integrating the cross-section for photodissociation over the energy range for dissociation. The cross sections used for this process come from Stancil et al. (in preparation) and are

specified by the keyword **Stancil**. The other option is to use a constant H_2 cross section for this process, which comes for the work of Allison and Dalgarno (1969) and is used when **AD69** appears on the command line. This process, while not the dominant destruction process for H_2 in a PDR, may be important in some environments.

set H2 grain formation pumping This changes the distribution function for populations of v, J levels of H_2 following formation on grain surfaces. There are several possible distributions: Takahashi (2001, **Taka**, this is the default); Draine and Bertoldi (1996, **DB96**); and thermal (**thermal**).

The keyword **OFF** will put all newly formed H_2 in the lowest ortho and para states with a 3:1 ratio. This can be used to test the importantence of grain formation pumping. For low to moderate temperatures in a purely collisional gas the higher levels within X may have no source of populations when this is done. The code may then fail because the level population matrix will become ill conditioned. This option should only be used for experiments.

set H2 Tad 420 This changes the binding energy, T_{ad}, in the formalism described by Le Bourlot (2000) to account for deexcitation of H_2 while on a grain surface. The number is the binding energy expressed as a temperature. The default value of 800 K maximizes the rates of H_2 collisional deexcitation and ortho-para conversion. If the number is ≤ 10 it is interpreted as a log of the temperature unless the keyword **linear** appears.

19.13.27 Set HCORR

The density used in electronic collisions is actually $n_e + 1.7 \times 10^4 n\left(H^0\right) f$. The second term is an approximate correction for collisions by neutral hydrogen. This is only a guestamite based on Drawin (1969), (see also Weisheit, 1974, Steenbock and Holweger, 1984, Lambert, 1993, and Kiselman, 2001). The number on this command adjusts the scale factor f. You can turn off the process by setting the number to zero.

19.13.28 Set HO charge transfer [chemistry; ionization]

This command is no longer supported.

19.13.29 Set ind2 [on, off]

This provides a way to turn off the affects of induced two-photon emission or absorption. One of the keywords **on** or **off** must appear to turn induced two-photon processes on or off. It is on by default.

19.13.30 Set ionization tolerance 0.01

Sets the allowed error tolerance on ionization solver solutions.

19.13.31 Set isotopes [ALL] – DEPRECATED

This command has been deprecated. Use **element isotopes all** instead.

19.13.32 Set kshell energy 1e6

This changes the energy of the highest continuum point considered for photoelectric opacity. The default is 1 MeV, sufficiently high that Compton recoil and pair production are the dominant opacity sources and K-shell opacity may safely be ignored. Setting this limit to smaller values will save some compute time since the evaluation of the photoionization rate integrals will not extend to as high an energy. The argument is the energy in Rydbergs and it must be greater than 194 Ryd. If zero is entered then the high-energy limit of the continuum will be used.

19.13.33 Set Leiden hack

This replaces certain physical processes with the simple prescriptions that were specified for the 2004 Leiden PDR workshop (Röllig et al., 2007). These are only intended for testing and would not be used in a real calculation. If any **set Leiden hack** commands are entered the code will complain that a physical process has been disabled.

This enables all hacks. Individual hacks can be turned on with additional keywords. They are:

set Leiden hack H2* off The chemistry network used in the workshop did not include H_2* (see Tielens and Hollenbach, 1985a for a definition). This turns off reactions between that species and the CO network (although it is still present in the H-H_2 network).

set Leiden hack CR off This turns off cosmic-ray excitation for the large H_2 molecule. This is normally treated as in Dalgarno et al. (1999).

set Leiden hack UMIST This forces the code to use parts of the UMIST database that (we feel) we do better with other methods.

19.13.34 set monitor scientific notation

monitors are commands that tell the code what value for a predicted quantity to expect. These are part of the test suite and allow automatic testing of the code's results. This command causes the printed monitored values to be printed in scientific notation rather than a float format. The **monitor** command is described on page 282 above.

19.13.35 Set nchrg 3

The grain physics uses an n-charge state model where, for each bin, the charge distribution is resolved in exactly n charge states, independent of grain size. This is discussed in van Hoof et al. (2004). The default value of n is 2 but can be reset with this command to any value between 2 and 10. Higher values of n will give more accurate results at the expense of greater compute time.

19.13.36 Set negopc

Negative opacities may occur during a calculation if a level happens to mase (Ferland, 1993). The code will generate a comment at the end if this happens. This command tells the code to save the optical depth array when negative opacities occur. The output will go to the file `negopc.txt`.

19.13.37 Set nend 500

This sets the default limit to the number of zones that will be computed. The preset default limiting number of zones is 1400 but more zones may be needed in large column density clouds or ones exposed to very intense radiation fields.

The limit to the number of zones that will be computed can be set with either this command or with the s**top zone** command. The only difference is in the level of warning that will be generated if the code stops after reaching the limiting number of zones. If the code stops because it reached the number of zones set by the **stop zone** command then it thinks that this was the intended stopping criterion and no comment is generated. However the code generates a ***warning*** if it stops because it reaches the default limit to the number of zones since this probably *was not* intended. By using this command the limit can be increased while still retaining the checking and warnings that are generated if the code stops for an unintended reason.

The code allocates memory to store a great deal of information for each zone. Increasing the number of zones will also increase the memory needed to run the code.

19.13.38 Set nFnu [options]

By default, CLOUDY will print the continuum flux at certain wavelengths as a series of entries in the standard emission line array. The **set nFnu** command changes which parts of the radiation field are included in the entry with the *nFnu* label. There are four possible radiation fields, the transmitted and reflected incident radiation field, and the transmitted and reflected diffuse emission produced by the nebula. The default is to include only the reflected incident radiation field and the transmitted and reflected diffuse emission. The transmitted incident is not included by default. Any or all of these may be entered on the command line, but at least one must occur. Any component left off the command will not be included.

The four keywords, and the component that is included, are the following:
incident reflected or **incident_reflected** includes the incident reflected continuum.
incident transmitted or **incident_transmitted** includes the incident transmitted continuum.
diffuse inward or **diffuse_inward** includes the diffuse inward continuum.
diffuse outward or **diffuse_outward** includes the diffuse outward continuum.
total includes all four components of the continuum.

All desired components must be specified on a single **set nFnu** command. The following shows how to only include the outward incident and outward diffuse emission in the *nFnu* entry in the printout:

```
set nFnu diffuse outward, incident transmitted
```

This command does *not* change the quantities printed with the *nInu* label—these are always the sum of the reflected plus transmitted incident continuum.

19.13.39 Set nFnu add 300 micron

You can also add custom vacuum wavelengths or frequencies at which the *nFnu* and *nInu* entries in the emission-line printout are evaluated. You can do that by adding one **set nFnu add** command for each point. You can enter as many as you like. Each line must contain a number and

a unit to identify the wavelength or frequency. Adding the unit is mandatory. The following keywords are supported: **eV**, **keV**, **MeV**, **Hz**, **kHz**, **MHz**, **GHz**, **angstrom**, **nm**, **micron**, **mm**, **cm** or **centimeter**, **wavenumbers**, **erg**, **ryd**, **K** or **Kelvin**. Some examples are:

```
set nFnu add 22 GHz
set nFnu add 350 micron
set nFnu add 4200 angstrom
```

19.13.40 Set nmaps 50

This controls the number of steps in the heating-cooling map that results from either the **map** or **save map** commands. Normally about 20 steps are taken between the lowest and highest temperatures.

19.13.41 Set PAH option

The **grains PAH** command described on page 85 adds PAHs to the calculation. These **set** commands change some details of their treatment.

set PAH constant or "H" This changes the function that describes how PAH abundances depend on physical conditions. The default, which can also be specified with the string `"H"`, is for the PAH abundance to scale as $n(H^0)$ / $n(H_{tot})$. You can also specify the PAH abundance to scale as the total abundance of neutral atomic and molecular hydrogen with the string `"H,H2"`. The abundance will be constant if the keyword **constant** appears.

set PAH Bakes will replace our self-consistent treatment of photoelectric heating by all grains with the PAH photoelectric heating equation given by Bakes and Tielens (1994). This command no longer guarantees energy conservation and is only intended for testing.

19.13.42 Set phfit [1995, 1996]

The key **1995** tells the code to use photoionization cross sections from Verner and Yakovlev (1995). The key **1996** is the default and uses Verner et al. (1996), which is partially based on Opacity Project cross sections.

19.13.43 Set pressure convergence 0.01

This sets the convergence criterion for the total pressure. The number is the largest relative error in the pressure. The number is interpreted as the linear error if it is positive and the log of this error if negative. The default value is 0.01.

19.13.44 Set pressure ionize 50

This sets a limit to the number of times the bottom ionization solver will be called from the top pressure solver within a zone. This is a debugging aid as well as a means to limit the time it takes the code to declare convergence failure. The limit has a default value of 3000. This is

approximately 500 more calls than is ever needed in any test suite simulations. The vast majority ($\sim 99\%$) of zones are converged in less than 100 calls. Users should exercise caution in increasing the limit. Encountering the default limit is a strong indication that something is wrong with the simulation. It is unlikely the simulation will converge if more calls are allowed.

19.13.45 Set save commands

Save output is the primary output mechanism for the code. This command sets various details about how this output is generated.

set save flush will cause the file buffers to be flushed after every iteration.

set save hash "newstring" When more than one iteration is saved into a file each ends with a series of hash marks that makes it easy to locate with an editor. This sets the hash string to something else. A new string must appear within the pair of double quotes.

The special string "**return**" is replaced with a single carriage return which translates into the pair of carriage returns that ***gnuplot*** uses to separate plots. The special string "**time**" will give the elapsed time since the start of time-dependent calculations, followed by a pair of carriage returns. In both cases the keyword must appear within the pair of double quotes.

The hash can be completely turned off with the **no hash** option on the **save** command.

set save line width / resolving power, suppress, absorption The observed contrast between emission lines and the continuum depends on the intrinsic line width and, for an unresolved line, the resolution of the spectrometer. Lines are included in the radiation field produced by the **save continuum** commands. The default behavior is to do this in such a way that the energy is conserved in the resulting output.[3] This command allows you to adjust the contrast between the lines and continuum by, in effect, changing the velocity resolution of the spectrometer which views the unresolved lines.

Lines and continua are stored separately throughout the code. They are combined only when the output from the **save continuum** command is produced using the expression

$$\nu F_\nu(\text{total}) = \nu F_\nu(\text{continuum}) + RI(\text{line}) \tag{19.2}$$

where $R \equiv \lambda/\Delta\lambda$ is the resolving power and $\Delta\lambda$ is the resolution.

The default behavior is to match R to the resolving power of the coarse continuum at the frequency of the line as set in the the file `continuum_mesh.ini`. This assures energy conservation. If you want to boost weak lines in a plot, you can artificially change the resolution of the spectrum by using this command.

When you use the **set save line width** command, you should supply a velocity width Δv in km s^{-1}. This will set the resolving power $R = c/\Delta v$, where c is the speed of light. If you use the alternative **set save resolving power** command, you can specify R directly. Supplying a smaller value of Δv or a higher value of R will increase the line-to-continuum ratio and thereby make lines appear stronger.

[3] In C90.00 the standard was to use a line width equal to the speed of light. In C90.04 the standard was changed to using a line width of 1000 km s^{-1} . Finally, in version C10 the standard was changed again to using energy conservation. Versions prior to C90.00 did not include emission lines in the saved spectral energy distribution.

```
## set velocity resolution to 100 km/s => R = 3000 (approximately)
set save line width 100
## alternatively set spectral resolving power directly
set save resolving power 3000
```

Note that this command will *only* adjust the height of the line, not the width. The latter will always be the width of one cell in the coarse continuum mesh (even if the line is broader than that cell). This implies that the line flux in the continuum array is artificially changed by this command. Changing the spectral resolution can be useful to emphasize weak lines in a plot, but should never be used when energy conservation is important, e.g. when you want to fold the saved continuum with a photometric passband.

If the keyword **absorption** occurs then the effects of absorption lines on the transmission of the incident continuum will be modified by the factor given in equation 19.3. The transmitted radiation field is changed as

$$\nu f_\nu = \nu f_\nu^0 \max\left[1-(1-trans)R\frac{\Delta\nu}{\nu},0\right] \tag{19.3}$$

where *trans* is the correct transmission factor as propagated from the fine to coarse mesh, and $\nu/\Delta\nu$ is the resolution of the coarse continuum mesh. The keyword **absorption** makes the effects of absorption appear larger than than actually are, but gives an impression their appearance would have if the spectrum were viewed at a higher resolution. This approach uses the opacities stored in the much higher resolution fine continuum. As is the case for emission lines, using this option will violate energy conservation.

If you include the keyword **suppress** on the command line, then emission lines will not be included in the saved spectral energy distribution. If you combine the keywords **suppress** and **absorption**, then the effects of absorption lines will also be suppressed and only a pure continuum will be saved.

The only effect of this command is to change the line-to-continuum contrast in output from all **save** commands except for that produced by the **save transmitted continuum** command.

set save luminosity old In versions of Cloudy between roughly 1983 and C13 the **save continuum** output reported continuum and line luminosities per unit area of cloud at the inner radius when the luminosity case was used. This was to avoid exponential limits on floating point numbers. Today the code will report luminosities when appropriate. This command will use the old style for backwards compatibility. If this is used the **save continuum** output will be $\nu L_\nu/4\pi r_{inner}^2$ and have units $\mathrm{erg\,cm^{-2}\,s^{-1}}$.

set save prefix "test" This adds a prefix to all filenames and so provides an easy way to change all file names at one time. The prefix is any string that appears between the pair of double quotes on the command line. This command should precede all save commands.

19.13.46 Set species gbar 0.01

This sets a default collision strength for database transitions that have no collision or radiative data. You can turn this feature off by setting the value to zero.

19.13.47 Set species continuum

This changes the lower and upper bounds, and the resolution, of the species pseudo-continuum reported with the **save species continuum** command, Section 16.86.1. The species must be specified in double quotations, followed by three numbers: the short and long wavelength limits (vacuum wavelengths, both in angstrom) and the number of logarithmic bins to divide this into. All three numbers must be specified. The default are lower and upper wavelength limits of 1000 Å and 7000 Å, broken into 1000 bins. For example,

```
set species "Fe+" continuum 3000 5000 200
```

would accumulate the Fe II emission from 3000 Å to 5000 Å in 200 logarithmic bins.

19.13.48 Set temperature [floor, convergence]

These commands change some details of the thermal solution.

set temperature floor This sets a lowest kinetic temperature to allow in a calculation. When the temperature falls below this floor the code goes over to a constant temperature solution at the floor temperature. This provides a way to mimic having a minimum temperature that is set by some external and unspecified source of heat. The temperature will be interpreted as a log if it is less than or equal to 10 or the keyword **log** is present, and as linear if the keyword **linear** is present.

set temperature convergence The gas kinetic temperature is set by the balance between the heating and cooling. This command sets the relative error allowed in the heating-cooling match. The number is the largest fractional error allowed. It is interpreted as the error itself if it is positive and the log of the error if it is less than or equal to zero. The default is 0.005. This will be the error allowed in each zone. The total error or energy conservation mismatch integrated over a cloud will be much smaller, usually of order ten times smaller than the tolerance specified.

19.13.49 Set test

This sets the logical variable *lgTestCodeEnabled* to true. It provides the facility to conditionally run test code somewhere in the main body of CLOUDY.

19.13.50 Set trimming -9 [upper lower, off] [old, new]

The code saves execution time by not computing the population of stages of ionization with trivial abundances. The thresholds for excluding an ionization stage are chosen with photoionization equilibrium in mind. These may not be appropriate for some other conditions or if your definition of trivial is different from mine.

This command changes the limit for the smallest fractional abundance, $n(\mathrm{A}^{+x})/n(\mathrm{A})$, to consider. The smallest relative abundance to be considered for a stage of ionization higher than the ionization peak is changed with the **upper** keyword. The command **set trimming upper off** will turn off trimming of the highest stage of ionization. The smallest relative abundance of ions

below the peak is changed with the **lower** keyword. The default relative ionization fractions are 10^{-6} and 10^{-10} respectively. If no keyword appears then both are changed to the number entered. The argument is the log of the lowest or highest ionization fraction to consider.

Generally, line excitation energies for atoms with stages of ionization higher than the peak ionization will strongly exceed the ambient temperature so these ions will have little influence on the calculated temperature or spectrum. This is not true for lines formed by ions below the peak distribution. The keyword **off** sets the smallest abundance to just above the machine's floating precision limit.

The keyword **old** resets the algorithm used for trimming to the legacy approach; the keyword **new** resets this to the revised approach (the default), which includes some hysteresis in the trimming controls to prevent zone-to-zone oscillations in the number of active ionization states.

19.13.51 Set tsqden 8

The code performs an analysis of the predicted emission-line spectrum at the end of the calculation. This analysis will find the structural t^2 as well as a t^2 deduced from the [O III] and H I spectrum (Kingdon and Ferland, 1995 discuss this at length). Such an analysis only makes sense for densities below the critical density of the [O III] atom, which is $\sim 10^5$ cm^{-3}. The results of this analysis will not be printed if the density is higher than the value of *tsqden*, currently 10^7cm^{-3}. The number on the line is the log of the highest hydrogen density for which this analysis will be performed.

19.13.52 set update couplings every ion

This updates the couplings between the ionization networks fully after every ion balance solution, rather than only when all ions and the chemical network have been calculated. This may be necessary to allow the solution to converge in cases where the coupling sources dominate collisional and radiative processes, but will increase run times in general.

19.13.53 Set UTA

This determines the UTA line data to be used. Data for several species are available. Most of them come from N. Badnell's work (Badnell and Seaton, 2003; Badnell et al., 2005), and they consist of data for all elements up to Zn along all iso-sequences from Li-like to Mg-like. In the case of iron, these data cover the ionization stages Fe^{+14} through Fe^{+23}.

There are two additional sets of UTA data for lower stages of ionization of iron. The data of Kisielius (Kisielius et al., 2003; Ferland et al., 2013a) concern Fe^{+13} though Fe^{+15}, and they are the data of preference, as described in Ferland et al. (2013b), because of their higher accuracy and greater number of emission lines. The Gu et al. (2006) data concern Fe^{0} through Fe^{+15}. By default, the Kisielius data are used instead of the Badnell and Gu data for the relevant ionization stages of iron.

The following commands may be used to adjust the UTA data employed in a simulation:

set UTA off – disable **all** UTA lines. This is provided as a debugging tool. The simulation is considered unphysical when this command is given.

set UTA Kisielius off – disable the Kisielius et al. (2003) data for Fe^{+13} though Fe^{+15}. They are enabled by default. When disabled, the Badnell data for Fe^{+14} and Fe^{+15}, and the Gu data for Fe^{+13} are used instead.

To check which data set is used for each ionic species, see **print UTA references**, Section 16.30.

19.13.54 Set WeakHeatCool 0.02

This command sets the threshold for the weakest cooling or heating source included in output from the **save heating** or **save cooling** commands. The default is 0.05. The number entered is interpreted as a log if it is negative.

19.14 Table lines "name.dat"

When the code is used as a subprogram for other, larger, codes, it is possible to read in a series of lines whose intensity will later be extracted by first calling *cdGetLineList*, then by calling *cdLine*. The procedure is described in Part 2 of this document. A set of lines that occur within any of several data files is then read. This method is used in most of the large grid programs I use.

This command provides a quick way to confirm that the list of lines in the line data files have valid names. The command takes the name of a file as its argument and checks that all lines can be found. This command can also be used to check the standard "`LineList*.dat`" files that are part of the distribution. This is actually a form of the **monitor** command since it will announce a botched monitor if it cannot find any of the lines. Note that all 5 characters of the keyword **lines** need to be entered to avoid confusion with the keyword **linear**. See the discussion of *cdGetLineList* in Part 2 for further information.

19.15 Test

This command runs a simple model to do a "smoke test", a quick sanity check on the code. The single command produces a series of commands to run a two-zone constant-temperature model.

19.15.1 Test

The **table agn** command is used with the density and temperature constant at 10^4 K. The ionization parameter is $U = 10^{-2}$ and the calculation stops after two zones and one iteration. Many **monitor** commands are used to verify predictions. The commands that set up the model will be printed if the keyword **print** occurs on the line.

19.15.2 Test large

When the **large** keyword appears the model hydrogen atom will be set to a large number of levels.

19.15.3 Test H2

This tests the large H_2 molecule.

19.15.4 Test molecular

This does a simulation that is cold and nearly fully molecular. The can be combined with the **H2** keyword to test the H_2 molecule in the fully molecular limit.

19.16 Performance speedups

It is possible to speed up the calculation at the loss of some physical fidelity or stability. This section outlines the commands that make this possible. Each command is described in greater detail in sections earlier in this document.

Many of these options are included in the `fast.ini` initialization file. Using this `ini` file will decrease the execution time by roughly a factor of two. This is meant as a way to roughly explore vast ranges of parameters or as a debugging aid.

N.B.!! The fidelity of the simulation is compromised! These turn off significant physical processes.

19.16.1 Turn off minor elements

The ionization distribution and the effects of an element on the gas emission, cooling, and opacity are all determined self-consistently. This includes the first 30 elements and many of these have trivial abundances. The calculation can be sped up with little loss of physical fidelity elements with very small abundances are turned off. This is done with the **element off** command.

The **element limit off** command described on page 79 will turn off all elements which have an abundance smaller than a specified limit. The following will turn off all elements with abundances less than 10^{-7} of hydrogen.

```
element off limit -7
```

Specific elements can be turned off by giving that element on the **element off** command. The ini files `c84.ini`, `fast.ini`, and `ism.ini`, among others, include a set of **element off** commands to do this.

Turning off an element will affect results if the opacity, heating, or cooling due to the element is significant. Minor elements can have major effects in very dense, optically thick, gas, where all coolants are thermalized and so are close to the black body limit.

19.16.2 Degrade the grain physics

The treatment of grain physics is today's state of the art (van Hoof et al., 2001, 2004) and can be CPU-intensive. Grains have several major effects. Elements locked up in solids are missing from the gas phase and so their cooling is also missing (Kingdon et al., 1995). Grains provide the dominant opacity across much of the optical, near IR, and UV spectrum. This absorbs the incident

continuum, affects the radiative acceleration of the gas, and the transfer of optically thick lines. Radiative and collisional interactions between grains and the gas affect the gas temperature. In PDR and molecular regions, grains have an important effect on the charge balance of the gas. This affects both the ionization and chemistry (Abel et al., 2005). Grain surface reactions and molecular freeze-out change the chemistry and charge balance of the gas.

The first effect can be included at no expense by setting the gas-phase abundance of an element to a depleted value. Gas-phase abundances can be decreased by typical ISM depletion factors using the **metals deplete** command. This does not turn on grains and applying a depletion factor has no impact on the execution time.

Turn off quantum heating. Quantum heating is the process where a grain's temperature spikes after absorbing a single photon. This temperature spiking has an effect on the Wien tail of the grain's emitted spectral energy distribution but no other effect. Quantum heating can be disabled with the **no qheat** option on the **grain** command if the thermal continuum is not something you need to predict. Quantum heating only affects the emitted spectrum so turning if off will not affect results if the emitted IR continuum does not affect the gas. The IR continuum will affect the gas if it is absorbed by the gas or dust, which can occur with some line, or if the source is optically thick in the near IR continuum.

Use single rather than distributed grains. If the **single** keyword appears on the **grain** command then a single mean grain size will take be used instead of the size-resolved grain distribution. This affects the entire simulation, including the gas temperature and emission line spectrum, because the grain - gas interactions depend on the grain size.

19.16.3 Turn off the chemistry

The chemical network requires a fair amount of time due to the size of the matrix that must be solved. The chemical state is determined even in ionized regions to ensure continuity in the transition between the H^+ - H^0 - H_2 regions. The chemistry can be ignored if the simulation only includes ionized regions. This is done with the **no molecules** command.

19.16.4 Turn off level 2 lines

This is done with the **no level2 lines** command. The level 2 lines are a large set of transitions that came from the Opacity Project. They are important in dense or very high ionization gas, and for pumping excited levels (such as C I* or C II*) of many heavy elements.

19.16.5 Degrade the continuum resolution

The resolution of the coarse continuum mesh affects the execution time since photoionization rates and opacities involve repeated sums over it. The resolution is changed with the **set continuum resolution** command.

If you change the continuum resolution then the standard stellar SEDs and grains that are part of the code distribution will need to be recompiled. This is done with the **compile grains** and **compile stars** commands.

19.16.6 Make the iso-sequence model atoms as small as possible

The code treats several iso-sequences with a unified model atom. The execution time increases when the number of levels within the model atoms is increased. Less time will be used if the smallest possible number of levels is used. The number of levels is adjusted with the **levels** option on the **database** command. The **database levels** commands for the H-like and He-like atoms accepts the keywords **very small** to make the atom as compact as possible. The line spectrum predicted by these atoms is likely to be highly inaccurate, and Case B intensities are far better when a compact model is used.

Appendix A

USING THE GRAIN CODE IN CLOUDY

A.1 Introduction

This release of Cloudy contains a grain model that constitutes a significant upgrade from the original model that was described in Baldwin et al. (1991). The code has been written by Peter van Hoof, except where indicated differently. These are the main features:

- The code resolves the size distribution of the grains in an arbitrary number of size bins (chosen by the user), and calculates all grain parameters such as temperature, charge distribution, emitted flux, etc., separately for each bin. Since grain opacities depend strongly on the grain radius, they need to be calculated separately for each bin as well.

- For this purpose Cloudy contains a Mie code for spherical grains (written by P.G. Martin, based on a code by Hansen and Travis (1974)). This code allows the user to calculate grain opacities using either a pre-defined or a user-supplied set of optical constants or opacities, an arbitrary size distribution, and an arbitrary number of grain size bins. The code can handle any number of grain bins, only limited by the memory on your machine.

- The code fully treats quantum heating of grains using a robust and efficient algorithm (which is a comprehensively upgraded version of a code originally written by K. Volk), implementing an improved version of the procedure described in Guhathakurta and Draine (1989). Combined with resolved size distributions, this will lead to a much more realistic modeling of the grain emission under all circumstances. Quantum heating is turned on automatically for all resolved size distributions[1] and single-sized grains (including PAH's, but excluding grey grains for which quantum heating is never used). The user can change the default behavior of the code by including the keyword **qheat** (to enable quantum heating) or **no qheat** (to disable quantum heating) with the **grains** command. If the relative grain abundance $A_{\rm rel}(r)$ in a particular zone (variable *gv.bin[nd]->dstAbund*) drops below a preset threshold value (variable *gv.dstAbundThreshold*, default value 10^{-5}), quantum heating will be disabled for that zone only. Note that quantum heating usually only

[1] The actual criterion is that the ratio of the volumes of the largest and smallest grain in each bin is smaller than 100 for quantum heating to be the default. Note that this criterion is very generous and that a much smaller ratio (i.e., a higher number of size bins) will usually be necessary to achieve proper convergence of the emitted spectrum.

influences the energy distribution of the emitted spectrum, nothing else[2]. So switching off quantum heating can be a good idea to speed up the modeling, provided the correct shape of the emitted spectrum has no influence on the results the user is interested in.

- The treatment of the grain physics has been completely overhauled, following the discussion in Weingartner and Draine (2001a), van Hoof et al. (2004), and Weingartner et al. (2006). The grain model has a detailed treatment of the photo-electric effect and collisional processes, and includes thermionic emissions. The code uses the n-charge state model where for each bin the charge distribution is resolved in exactly n discrete charge states, independent of the grain size. The default value for n is 2, but the user can choose any number between 2 and *NCHS*/3 = 10 using the **set nchrg** <**n**> command (note that there is no "a" in **nchrg**!). Choosing a higher value will usually give more accurate results at the expense of computing time. Using the default $n = 2$ should give sufficient accuracy for most astronomical applications. The maximum value of n can be increased by altering the definition of *NCHS* in `grainvar.h` and recompiling the code. It is however extremely unlikely that a value of $n > 10$ would ever be needed. A detailed description of the n-charge state model can be found in van Hoof et al. (2004).

  Extensive comparisons in collaboration with Joe Weingartner done in 2001 show that the photo-electric heating rates and collisional cooling rates predicted by Cloudy agree very well with the results from the Weingartner and Draine (2001a) model for a wide range of grain sizes (between 5 Å and 0.1 μm), and using various choices for the incident radiation field. A detailed discussion of this comparison can be found in van van Hoof et al. (2001).

A.2 Using the grain model

In order to use the grain model, two steps need to be taken. In the first step, the grain opacities are calculated using the **compile grain** command. In the second step these opacities can be used with the **grains** command to create the actual photo-ionization model. The Cloudy distribution comes with a number of pre-compiled opacity files which cover a number of standard combinations of grain materials and size distributions. If these are sufficient for your needs, you can skip the skip the first step, and use these opacities directly in the **grains** command. If you wish to use different grain materials and/or size distributions, you will have to compile the grain opacities first with the **compile grain** command described on page 268. In order to do this, you type in a single command line, e.g.:

```
compile grain "silicate.rfi"  "ism.szd" 10
```

followed by an extra carriage return. The Mie code needs refractive index or mixed medium, and size distribution files as input. These are files with names ending in "`.rfi`", "`.mix`", or "`.szd`", resp. It will generate an opacity file, ending in "`.opc`". The command shown above will instruct the Mie code to calculate opacities using optical constants for astronomical silicate and a Mathis et al. (1977) standard ISM size distribution. The size distribution will be resolved in 10 bins. This

[2]Quantum heating can influence the emission line spectrum as well if continuum pumping of the lines by the short wavelength end of the dust spectrum is important. Typically these will be molecular lines.

will produce a file `silicate_ism_10.opc` which contains all the opacities. This opacity file is in human readable form and contains many comments to clarify its contents. It also contains a table of the size distribution function $a^4 n(a)$ as a function of the grain radius a for reference. The format of the table is such that it could be used directly to define a size distribution table (see section A.5.7). Note that only a single **compile grain** command can be given in a single Cloudy run. Note furthermore that the opacity file produced in the example above is already included in the Cloudy distribution!

In the previous example a refractive index file was used to define the optical properties of a pure grain material (e.g., astronomical silicate or graphite). A second example could be:

```
compile grain "fluffy.mix"  "ism.szd" 10
```

(the mixed medium file `fluffy.mix` is included in the Cloudy distribution, and the text is also shown in Table A.9). Here the mixed medium file is used instead of a refractive index file to define a mixture of two or more grain materials (possibly including vacuum for defining fluffy grains). In this case the optical properties of the constituting materials are combined with a mixing law (also called effective medium theory or EMT) to create a set of optical constants that represents the mixture and can be fed into the Mie theory. Hence the refractive index and mixed medium files play similar roles in that both define a set of optical constants to be used by the Mie code. They differ in the fact that refractive index files are used to define pure materials, while mixed medium files define impure materials. As a consequence the internal format of both files is very different and will be described in more detail in Sections A.3 and A.4. The opacities generated from both types of input files are fully compatible though and can be used in exactly the same way, as shown below.

In the second step, you can use these opacities in a subsequent run of Cloudy with the **grains** command described in Hazy. An example could be:

```
set nchrg <n> # optional
grains "silicate_ism_10.opc" +0.100 log # quantum heating on
grains "fluffy_ism_10.opc" no qheat # quantum heating off
```

Several **grains** commands may be used simultaneously to define mixtures of grains. One can freely mix **set nchrg** and **grains** commands, and different grain types may be calculated using a different number of charge states. The **set nchrg** command will only affect **grains** commands that come later in the input file. For ease of use, the filenames of certain refractive index, size distribution, and opacity files in the standard Cloudy distribution may be replaced by keywords, as described in Hazy.

The refractive index, mixed medium, and size distribution files may be replaced by user-defined versions, giving the user considerably more freedom to define grain properties compared to the old grain model in Cloudy. The format for each of those files will be defined in Sections A.3, A.4, and A.5.

A.3 Description of the refractive index files[3]

In order for the Mie code in Cloudy to work, it needs to know the optical properties of the grains under consideration. If the grain consists of a single, pure material, these have to be defined in a separate file with a name that must end in "`.rfi`". In this section we will describe the format of this file. It is helpful to compare with e.g. the `graphite.rfi` or `silicate.rfi` file in the standard distribution while reading this section. This document pertains to refractive index files with magic number *1030103*. Mixtures of grain materials can also be defined using mixed medium files which are described in Section A.4.

As is the case with all files connected with the Mie code in Cloudy, the user has the freedom to add comments to the file provided they start with a sharp sign (#). These comments may either occupy an entire line (in which case the sharp sign has to be in the first column), or be appended to some input value. Comments have been liberally added to the refractive index files that come with the standard Cloudy distribution in an effort to make them self-explanatory. All refractive index files start with a magic number for version control. This number should simply be copied from the files in the standard distribution. Next comes the chemical formula of the grain material, for graphite this would simply be "`C`"; for a certain type of silicate this could be "`Mg0.6Fe0.4SiO3`" indicating $Mg_{0.6}Fe_{0.4}SiO_3$. Note that the formula is case sensitive. For simplicity I will call this elementary building block the grain molecule, even though this term is not always appropriate. The molecular weight will be calculated by Cloudy using the chemical formula. The next two lines in the refractive index file define the default abundance of the grain molecule. The first number gives the maximum number density $A_{\rm max}$ of the grain molecule (relative to hydrogen = 1) that can be formed, assuming it completely depletes at least one of the constituting atoms from the gas phase. Let us assume that the initial abundances in the gas phase (i.e., abundances *before* grains were formed) were $A(X)$. Then , for the silicate example above, $A_{\rm max}$ should be $\min(A({\rm Mg})/0.6, A({\rm Fe})/0.4, A({\rm Si}), A({\rm O})/3)$. The second number gives the fraction $A_{\rm eff}$ of the maximum amount that is actually formed (i.e., the efficiency of the process), and should be a number between 0 and 1. The default abundance of the grain molecule is then given by the product of these two numbers: $A_{\rm eff}A_{\rm max}$. The actual grain abundance used in the Cloudy modeling can be set with the **grains** and the **metals** command (see Hazy for details). This essentially defines an additional multiplier $A_{\rm rel}$ which may be either smaller or larger than 1, and may depend on position r. The actual grain molecule abundance used in the Cloudy model is then given by $A_{\rm rel}(r)A_{\rm eff}A_{\rm max}$. For the silicate example above, the number density of iron locked up in these grains would be given by $0.4A_{\rm rel}(r)A_{\rm eff}A_{\rm max}$ (relative to hydrogen = 1), or $0.4A_{\rm rel}(r)A_{\rm eff}A_{\rm max}n_{\rm H}(r)$ (in atoms/cm^3, $n_{\rm H}$ is the hydrogen number density as defined by the **hden** command). The next line in the refractive index file gives the specific density of the grain material in g/cm^3. The grain code needs to know the material type in order to determine certain grain properties that are currently hardwired into the code. Examples would be the grain enthalpy as a function of temperature and the ro-vibrational distribution of H_2 formed on the grain surface. The next line indicates which material type to use. Currently six choices are available. They are outlined in Table A.1. The next two lines give the work function and the bandgap between the valence and conduction band in Rydberg respectively. For conductors the bandgap should be set to

[3]Note that the term "refractive index file" is used rather loosely here. It also pertains to materials for which no refractive index data in the strict sense exist, such as grey grains and PAH's.

Table A.1: The definition of each of the material types hardwired into Cloudy. The code in column 1 needs to be entered in the refractive index file. Each of the entries in columns 3 and higher will be explained in the table indicated in the header of that column.

code	mnemonic	Table A.2	Table A.3	Table A.4	Table A.5	Table A.6	Table A.7	Table A.8
1	*MAT_CAR*	*ENTH_CAR*	*ZMIN_CAR*	*POT_CAR*	*IAL_CAR*	*PE_CAR*	*STRG_CAR*	*H2_CAR*
2	*MAT_SIL*	*ENTH_SIL*	*ZMIN_SIL*	*POT_SIL*	*IAL_SIL*	*PE_SIL*	*STRG_SIL*	*H2_SIL*
3	*MAT_PAH*	*ENTH_PAH*	*ZMIN_CAR*	*POT_CAR*	*IAL_CAR*	*PE_CAR*	*STRG_CAR*	*H2_CAR*
4	*MAT_CAR2*	*ENTH_CAR2*	*ZMIN_CAR*	*POT_CAR*	*IAL_CAR*	*PE_CAR*	*STRG_CAR*	*H2_CAR*
5	*MAT_SIL2*	*ENTH_SIL2*	*ZMIN_SIL*	*POT_SIL*	*IAL_SIL*	*PE_SIL*	*STRG_SIL*	*H2_SIL*
6	*MAT_PAH2*	*ENTH_PAH2*	*ZMIN_CAR*	*POT_CAR*	*IAL_CAR*	*PE_CAR*	*STRG_CAR*	*H2_CAR*
7	*MAT_SIC*	*ENTH_SIC*	*ZMIN_CAR*	*POT_CAR*	*IAL_CAR*	*PE_CAR*	*STRG_CAR*	*H2_CAR*

zero, for insulators such as silicates, a non-zero value should be used. The next line gives the efficiency of thermionic emissions. This parameter is usually unknown for materials of astrophysical interest, and using 0.5 should be a reasonably safe guess. Next comes the sublimation temperature in Kelvin. Then comes a line with a keyword which identifies what type of refractive index file is being read. It determines what the remainder of the file will look like. Allowed values are *rfi_tbl*, *opc_tbl*, *grey*, *pah1*, *ph2n*, *ph2c*, *ph3n*, and *ph3c*. In the *grey*, *pah1*, *ph2n*, *ph2c*, *ph3n*, and *ph3c* cases, no further data is needed and the refractive index file ends there (i.e., the information needed to calculate the opacities is hardwired in the code).

A.3.1 rfi_tbl

This format is used to define the refractive index as a function of wavelength, which will then be used by the Mie code to generate the opacities when combined with a size distribution file.

The remaining lines in the refractive index file define the optical constants. First the user has to enter a code to define how the complex refractive index n is written up; the following choices are supported:

1. – supply Re(n^2), Im(n^2) (the dielectric function),
2. – supply Re($n-1$), Im(n),
3. – supply Re(n), Im(n).

The next line gives the number of principal axes N_a for the grain crystal and should be a number between 1 and 3. For amorphous materials one should always choose 1 axis. For crystalline materials the number may be 2 or 3. Next comes a line with N_a numbers giving the relative weights for each of the axes. These numbers will be used to average the opacities over each of the axes in crystalline materials. For materials with only one axis, this number is obviously redundant, and a single 1 should be entered. For graphite it is appropriate to enter "1 2" indicating that the first axis will have relative weight 1/3 and the second 2/3 (i.e., the relative weights are 1:2). Next come N_a chunks of data defining the optical constants for each axis. Each chunk starts with a line giving the number of data points N_d for that axis, followed by N_d lines containing 3 numbers: the vacuum wavelength in μm, and the real and imaginary part of the complex number defined above.

Table A.2: The various choices for the enthalpy function hardwired in Cloudy.

mnemonic	type	reference
ENTH_CAR	graphite	Guhathakurta and Draine (1989), Eq. 3.3
ENTH_SIL	silicate	Guhathakurta and Draine (1989), Eq. 3.4
ENTH_PAH	PAH	Dwek et al. (1997), Eq. A4
ENTH_CAR2	graphite	Draine and Li (2001), Eq. 9
ENTH_SIL2	silicate	Draine and Li (2001), Eq. 11
ENTH_PAH2	PAH	Draine and Li (2001), Eq. 33
ENTH_SIC	α-SiC	Chekhovskoy (1971)

Table A.3: The various choices for the minimum charge hardwired in Cloudy.

mnemonic	type	reference
ZMIN_CAR	graphite	Weingartner and Draine (2001a), Eq. 23a+24
ZMIN_SIL	silicate	Weingartner and Draine (2001a), Eq. 23b+24

Table A.4: The various expressions for the electron affinity hardwired in Cloudy.

mnemonic	type	reference
POT_CAR	graphite	Weingartner and Draine (2001a), Eq. 4
POT_SIL	silicate	Weingartner and Draine (2001a), Eq. 5

Table A.5: The various choices for the inverse attenuation length. Note that these choices will *only* be used if no refractive index data is contained in the refractive index file.

mnemonic	type	description
IAL_CAR	graphite	use `graphite.rfi` to calculate the inverse attenuation length
IAL_SIL	silicate	use `silicate.rfi` to calculate the inverse attenuation length

Table A.6: The various expressions for the photoelectric yield hardwired in Cloudy.

mnemonic	type	reference
PE_CAR	graphite	Weingartner et al. (2006), Sect. 4 – 6
	graphite	Weingartner and Draine (2001a), Eq. 16 (with **no grain x-ray treatment**)
PE_SIL	silicate	Weingartner et al. (2006), Sect. 4 – 6
	silicate	Weingartner and Draine (2001a), Eq. 17 (with **no grain x-ray treatment**)

Table A.7: The various choices for splitting up the grain emissions.

mnemonic	type	description
STRG_CAR	graphite	store emitted spectrum as graphitic emission.
STRG_SIL	silicate	store emitted spectrum as silicate emission.

Table A.8: The expressions for the ro-vibrational distribution of H_2 formed on various grain surfaces.

mnemonic	type	reference
H2_ICE	ice mantle	Takahashi and Uehara (2001), Table 2
H2_CAR	graphite	Takahashi and Uehara (2001), Table 2
H2_SIL	silicate	Takahashi and Uehara (2001), Table 2

Note that the wavelengths may be either monotonically increasing or decreasing, and that the number of data points or the wavelength grid may be different for each axis. The grid of wavelengths need not coincide with the Cloudy grid, nor does it need to cover the entire range of energies used in Cloudy. Logarithmic interpolation or extrapolation will be performed where needed. *It is the users responsibility to ensure that each wavelength grid contains sufficient points to make this process meaningful; only minimal checks will be performed by Cloudy.* Note that if your refractive index data do not have sufficient wavelength coverage (as will usually be the case if they come from laboratory experiments) you cannot simply merge data from another refractive index file to extend the range. This is because the refractive index data need to obey the Kramers-Kronig relations. If you take data from two different sources, this will not be the case. In most cases you need to extend your data into the EUV and X-ray regime. The proper procedure in this case is to calculate the absorption opacity by adding the photoionization opacities (including inner-shell ionizations) for each of the atoms that make up the grain using data for neutral atoms. You then need to combine these data with the Kramers-Kronig relations to reverse engineer the refractive index data. This is the procedure that was used to create the refractive index files that are included in the Cloudy distribution.

A.3.2 opc_tbl

This format can be used to directly define the opacities as a function of photon energy. It is mainly useful to define alternative prescriptions for PAH's. Note that the refractive index file created this way still needs to be combined with a size distribution file in the usual way with the **compile grain** command. This step will not alter the opacities themselves, but is necessary to compute certain quantities that are related to the size distribution which are not defined in the refractive index file itself. It is not allowed to resolve the size distribution when using an opacity table. If you want to define a resolved size distribution, you will have to supply an opacity table and corresponding size distribution file for each size bin separately.

The remaining lines in the refractive index file define the opacities. The first line contains the number of data values N_v supplied on each line in the table. Allowed values are:

1. – supply $\sigma_{\rm abs}$ only (in cm^2/H),
2. – supply $\sigma_{\rm abs}$, $\sigma_{\rm sct} \times (1-g)$ (in cm^2/H),
3. – supply $\sigma_{\rm abs}$, $\sigma_{\rm sct}$ (in cm^2/H), and $(1-g)$ separately.
4. – supply $\sigma_{\rm abs}$, $\sigma_{\rm sct}$ (in cm^2/H), $(1-g)$, and the inverse attenuation length (in cm^{-1}).

Here $\sigma_{\rm abs}$ and $\sigma_{\rm sct}$ are the absorption and scattering cross sections, and g is called the asymmetry factor or phase function. The latter is needed since the radiative transfer in Cloudy assumes that photons that are scattered in a forward direction (at an angle of less than 90°) are not lost from the beam (i.e., they still make it out of the cloud). The asymmetry factor g gives the average value of the cosine of the scattering angle, so that $(1-g)$ is an approximate correction factor for the anisotropy of the scattering by grains. If no scattering cross sections are supplied, $\sigma_{\rm sct} \times (1-g)$ will default to 10% of the absorption cross section. If no asymmetry factor is supplied, g will default to zero. If no inverse attenuation lengths are supplied, they will be derived from the material type supplied previously in the file. On the next line is a keyword with two allowed

Table A.9: Example of a mixed medium file for fluffy silicate (`fluffy.mix`).

```
# toy model of fluffy silicate; for test purposes only!
4030103  # magic number for version control
0.61     # default depletion
2        # number of separate materials in grain
#
  50  "vacuum.rfi"
  50  "silicate.rfi"
#
br35     # which EMT should be used
```

values: **log** or **linear**, defining whether the opacity table will contain logarithmic or linear data values. The next line gives the number of data points N_d, followed by N_d lines of actual data. Each line of the table should contain $N_v + 1$ numbers, starting with the photon energy in Rydberg, followed by the absorption cross section, the scattering cross section, the factor $(1-g)$, and the inverse attenuation length (the last three values may be omitted, depending on the value of N_v). The photon energies must either be strictly monotonically increasing or decreasing. The grid of photon energies need not coincide with the Cloudy grid, nor does it need to cover the entire range of energies. Logarithmic interpolation or extrapolation will be performed where needed. *It is the users responsibility to ensure that the frequency grid contains sufficient points to make this process meaningful; only minimal checks will be performed by Cloudy.*

A.4 Description of the mixed medium files

In order to define the optical properties of a grain consisting of a mixture of materials[4], a separate file with a name that must end in "`.mix`" has to be used. In this section we will describe the format of this file. An example of a mixed medium file is shown in Table A.9. This document pertains to mixed medium files with magic number *4030103*.

As is the case with all files connected with the Mie code in Cloudy, the user has the freedom to add comments to the file provided they start with a sharp sign (#). These comments may either occupy an entire line (in which case the sharp sign has to be in the first column), or be appended to some input value. All mixed medium files start with a magic number for version control, as shown in Table A.9. The chemical formula and the abundance at maximum depletion of the mixed material will be derived by Cloudy using the information supplied in the refractive index files of the constituting materials. However, the default depletion cannot be calculated and needs to be supplied by the user on the next line. This number is defined as a fraction of the maximum depletion, and should be a number between 0 and 1 (see also the discussion in Section A.3). When the opacities are calculated, the code will print the chemical formula and the abundances (both at maximum and default depletion), so that the user can check whether the default abundance is correct. The next line gives the number n of separate materials in the grain (n should be at least 2; it is allowed to specify the same material several times), followed by n lines giving information

[4] In this section the term material can also mean vacuum so that the user can define fluffy grains by mixing one or more materials with vacuum.

Table A.10: Allowed choices for the mixing law.

mnemonic	reference
BR35	Bruggeman (1935)
FA00	Voshchinnikov and Mathis (1999)
	Farafonov (2000)
MG04	Maxwell Garnett (1904)
ST95	Stognienko et al. (1995)

about the materials. Each of these lines should give the relative fraction of the volume occupied by that particular material, followed by the name of the refractive index file between double quotes. Note that the refractive index files have to be of type *rfi_tbl* since mixing laws need optical constants as input and cannot work with opacities. The normalization of the relative volumes can be on an arbitrary scale. The last line in the file should be a keyword identifying which mixing law is to be used. Allowed values are listed in Table A.10. In the case of Farafonov (2000), the mixing law assumes that the grain consists of concentric layers of material. It is assumed that the first material supplied in the mixed medium file identifies the innermost layer, and the following lines identify the subsequent layers towards the outer edge of the grain. The outermost layer will determine the material type as defined in Table A.1. None of the layers are allowed to be crystalline, i.e., have more than one principal axis, and the outermost layer is also not allowed to be vacuum. In the case of Bruggeman (1935) and Stognienko et al. (1995), the grain is assumed to be a random (possibly fractal) mixture of materials and the sequence in the mixed medium file is irrelevant, except that the material type of the mixture is still defined by the last entry in the list of materials. For this reason it is not allowed to have vacuum as the last entry. In the case of Maxwell Garnett (1904) the grain is assumed to consist predominantly of a matrix which is defined by the first entry in the mixed medium file, which encapsulates small, random inclusions which are defined by the remaining entries. The material type, as well as many other properties of the grain will be determined by the matrix material, i.e., the first entry in the file. The matrix material is not allowed to be vacuum or a crystalline material, but the inclusions can be.

A.5 Description of the size distribution files

In order for the Mie code in Cloudy to work, it needs to know the size distribution of the grains under consideration. This distribution has to be defined in a separate file with a name that must end in "`.szd`". In this section I will describe the format of this file. This document pertains to size distribution files with magic number *2010403*.

If we denote the number of grains $n_g da$ with radii between a and $a+da$ as $n_g da = n(a)da$, the purpose of the size distribution file is to define $n(a)$, or alternatively $a^4 n(a)$ which is more commonly used.

As is the case with all files connected with the Mie code in Cloudy, the user has the freedom to add comments to the file provided they start with a sharp sign (#). These comments may either occupy an entire line (in which case the sharp sign has to be in the first column), or be appended to some input value. Comments have been liberally added to the size distribution files that come with the standard Cloudy distribution in an effort to make them self-explanatory. All size distribution

files start with a magic number for version control. This number should simply be copied from the files in the standard distribution. The next line should contain a keyword indicating which type of size distribution will be entered. The following choices are currently supported: *ssize* - a single sized grain, *ncarb* - a PAH molecule with a specified number of carbon atoms, *power* - a simple power law, *exp1*, *exp2*, *exp3* - power laws with an exponential cutoff, *normal*, *lognormal* - a Gaussian distribution in a or $\ln(a)$, *table* - an arbitrary size distribution supplied as a table. This keyword is case insensitive. The rest of the file contains the parameters needed to fully define each of those choices. I will now describe these choices in more detail. It should be noted that at this stage the absolute normalization of the size distribution is irrelevant; that will be defined in the refractive index file by the default grain abundance. Each parameter mentioned below should be entered on a separate line, unless indicated otherwise. All size parameters should be entered in μm.

A.5.1 ssize

In this case the size distribution is given by a simple delta function:

$$n(a) \propto \delta(a - a_0)$$

The only parameter that needs to be supplied is the radius of the grain a_0.

A.5.2 ncarb

This case also describes a single-sized grain, but this time the number of carbon atoms is the only parameter that is supplied. This is convenient for defining a PAH molecule of a specific size. This size distribution is obviously only meaningful when combined with carbonaceous material.

A.5.3 power

In this case the size distribution is given by a simple power law:

$$n(a) \propto a^{\alpha} \;\; a_0 \leq a \leq a_1$$

Hence this distribution needs three parameters, which need to be supplied in the order a_0, a_1, α.

A.5.4 exp1, exp2, exp3

In this case the size distribution is given by a power law with a first-, second-, or third-order exponential cutoff:

$$n(a) \propto a^{\alpha} F(a;\beta) C_l(a;a_l,\sigma_l) C_u(a;a_u,\sigma_u) \;\; a_0 \leq a \leq a_1.$$

The function F is included to give extra curvature in the power-law region, the functions C_l and C_u define the cutoff of the distribution below a_l and above a_u. These functions are defined as follows:

$$F(a;\beta) = \begin{cases} (1-\beta a)^{-1} & \text{if } \beta < 0 \\ 1 & \text{if } \beta = 0 \\ (1+\beta a) & \text{if } \beta > 0 \end{cases}$$

$$C_l(a;a_l,\sigma_l) = \begin{cases} \exp\left[-\left(\frac{a_l-a}{\sigma_l}\right)^n\right] & \text{if } a < a_l \\ 1 & \text{if } a \geq a_l \end{cases}$$

$$C_u(a;a_u,\sigma_u) = \begin{cases} 1 & \text{if } a \leq a_u \\ \exp\left[-\left(\frac{a-a_u}{\sigma_u}\right)^n\right] & \text{if } a > a_u \end{cases}$$

The values of σ_l or σ_u may be set to zero, in which case a straight cutoff in the size distribution will be used. Note that when β, σ_l, and σ_u are all set to zero, this size distribution degenerates to the simple power law discussed above. The parameters need to be supplied in the following order: a_l, a_u, α, β, σ_l, σ_u, a_0, a_1. The value of n is determined by the keyword used: $n = 1$ for *exp1*, etc. The parameter β should be entered in μm^{-1}. Note that this size distribution extends infinitely beyond a_l and a_u, so additional cutoffs at a_0 and a_1 are needed. Either of these values may be set to zero, in which case Cloudy will calculate a safe default value such that only a negligible amount of mass is contained in the tail beyond that limit.

A.5.5 normal

In this case the size distribution is given by a Gaussian distribution in a:

$$n(a) \propto \frac{1}{a}\exp\left(-\frac{1}{2}\left[\frac{a-a_c}{\sigma}\right]^2\right) \quad a_0 \leq a \leq a_1$$

The parameters for this distribution need to be supplied in the order: a_c, σ, a_0, a_1. As was discussed in the previous section, the values of a_0 or a_1 may be set to zero in which case Cloudy will calculate a safe default.

A.5.6 lognormal

This case is completely analogous to the *normal* case discussed above, except that the distribution is now given by:

$$n(a) \propto \frac{1}{a}\exp\left(-\frac{1}{2}\left[\frac{\ln\{a/a_c\}}{\sigma}\right]^2\right) \quad a_0 \leq a \leq a_1$$

A.5.7 table

This option allows the user to define an arbitrary size distribution in the form of a table of $a^4n(a)$ as a function of a. First values for the lower and upper size limit a_0 and a_1 should be supplied. These values need not coincide with the lower and upper size limit of the table, although the range of the table should be at least that large (no extrapolation will be performed). Next the number of data pairs n in the table should be supplied, followed by n lines each containing two numbers: a (in μm) and $a^4n(a)$ (in arbitrary units). Note that the values for a in the table must be strictly monotonically increasing.

Appendix B

INCLUDING USER-DEFINED ATMOSPHERE GRIDS IN CLOUDY

B.1 Introduction

CLOUDY now features the possibility of including user-defined stellar atmosphere grids with an arbitrary number of dimensions (this is the number of parameters that are varied in the grid). Using such a grid requires two steps. The first is to create an ascii file with all the necessary information. The format of this file will be described in Section B.2. The second (optional) step is to compile this file into an index file. This is described in Section B.3. The runtime behavior of the code is described in Section B.4.

B.2 The ascii file

Each stellar atmosphere grid needs to be defined in a single ascii file containing all the necessary information. The format is illustrated in Table B.1. The aim of this format is to keep the data in the ascii file as close as possible to the original calculations. The file starts with some general information, followed by a block of model parameters, the frequency/wavelength grid, and finally each of the spectral energy distributions (SEDs). No comments of any sort are allowed in the file, except at the start (i.e., before the magic number). It is implicitly assumed that each SED has been rebinned onto the same frequency/wavelength grid. This grid does not need to coincide with the CLOUDY grid. The grid will be automatically rebinned onto the CLOUDY grid. The supplied frequency/wavelength grid doesn't need to cover the entire CLOUDY frequency range. Extrapolation to lower frequencies will be done using a Rayleigh-Jeans slope. It is allowed for flux points in the Wien tail to have zero flux. Above the highest frequency present in the ascii file, fluxes will be set to zero. It is the users responsibility to assure that sufficient data are supplied to enable a safe extrapolation of the spectra to the entire CLOUDY grid.

I will now describe each of the items in the ascii file in more detail. You can start the ascii file with an optional block of comments, consisting of lines with a hash symbol '#' in the first column. Such comments are *only* allowed at the start of the file and cannot be embedded amongst the data! The first line after the comment block contains the magic number that identifies the syntax version. It is checked when CLOUDY reads the ascii file. This document describes versions

20060612 and *20160614*. These two versions are almost identical, except for one extra model parameter in the latter version. So unless noted explicitly, all of the discussion below applies to both versions. The second line contains the number of parameters *ndim* that are varied in the grid. Typically this number is 2, but it may be any number ≥ 1 and $\leq$ *MDIM* (the latter parameter is defined in `stars.h`). The default value for *MDIM* is 4, which should be sufficient for all current grids. The parameter can however be increased if the user wishes so. The code will then have to be recompiled. The third line contains the number of parameters *npar* that are supplied for each atmosphere model. This number may be larger than *ndim*, but not smaller. The additional parameters are printed in the output, so using *npar* > *ndim* can be a useful way of supplying additional information about the models. The number of parameters is limited by the parameter *MDIM* defined in `stars.h`. Next follow *npar* lines, each giving a label for that parameter. This is used in the CLOUDY output. The labels are completely free. However, if the first label matches the string *Teff*, and the second label matches *log(g)* (case sensitive), then special rules apply. We will refer to this as a Teff-log(g) grid. See Sections B.3 and B.4 for further details. The next line gives the number of atmosphere models *nmod* in the grid, followed on the next line by the number of frequency/wavelength points *nfreq* in each of those models. In order to keep the format of the ascii file as close as possible to the original models, it is allowed to use either a frequency or wavelength grid in arbitrary units. The same holds for the dependent variable, which may either be a flux per frequency or wavelength unit. Both the dependent and independent variable should be entered as linear quantities, so logarithmic units like magnitudes are not supported. Wavelengths are assumed to be in vacuum. With this amount of freedom, the code needs to know how to convert these variables to CLOUDY's internal units. This is defines on the next 4 lines. The next line should contain either *nu* to state that the file contains a frequency grid, or *lambda* to state that it contains a wavelength grid. The next line contains a conversion factor to convert the numbers in the ascii file to a frequency in Hz (case *nu*) or a wavelength in Å (case *lambda*). The conversion factor is multiplied with the numbers in the ascii file. So, if the ascii file would contain a wavelength grid in nanometer, then the conversion factor would have to be 10 to convert it into a grid in Å. The next line defines the data type of the dependent variable. Legal values are *F_nu, H_nu* for fluxes per frequency unit, or *F_lambda, H_lambda* for fluxes per wavelength unit. The next line contains the conversion factor to convert the numbers in the ascii file to a flux in $\mathrm{erg\,cm^{-2}\,s^{-1}\,Hz^{-1}}$ (case *F_nu, H_nu*) or $\mathrm{erg\,cm^{-2}\,s^{-1}\,Å^{-1}}$ (case *F_lambda, H_lambda*). Again the conversion factors are multiplied with the numbers in the ascii file. In the cases *H_nu* and *H_lambda* no implicit factor 4π will be used! So if the ascii file contains Eddington fluxes *H_nu* in $\mathrm{erg\,cm^{-2}\,s^{-1}\,Å^{-1}\,sr^{-1}}$, then the conversion factor should be 12.56637... All fluxes are eventually renormalized in CLOUDY, but nevertheless it is important to get this conversion correct. The effective temperature of interpolated models is internally checked by evaluating the integral:

$$\int_0^\infty F_\nu d\nu = \sigma T_{\mathrm{eff}}^4.$$

If the result deviates too much from the expected value an error will be printed. If all models generate such an error, you should check your normalization.

This concludes the general setup of the grid. The remainder of the file contains the actual parameters and the data of the atmosphere models. These numbers can be formatted in any way you like, as long as it complies with C formatting rules. More in particular, you can supply more than 1 number per line, as long as they are separated by whitespace. Several forms of Fortran

Table B.1: Example of an ascii file, derived from the kwerner.ascii file. **Comments are not allowed to be embedded in the file and are only shown here for clarity.**

```
  20060612       # magic number
  2              # ndim
  2              # npar
  Teff           # label par 1
  log(g)         # label par 2
  20             # nmod
  513            # nfreq
  nu             # type of independent variable (nu or lambda)
  3.28984196e+15 # conversion factor for independent variable
  F_nu           # type of dependent variable (F_nu/H_nu or F_lambda/H_lambda)
  1.00000000e+00 # conversion factor for dependent variable
# nmod sets of npar model parameters (syntax version 20060612)
# nmod sets of npar model parameters plus track ident (syntax version 20160614)
   80000. 5.0  80000. 6.0  80000. 7.0  80000. 8.0
  100000. 5.0 100000. 6.0 100000. 7.0 100000. 8.0
  120000. 6.0 120000. 7.0 120000. 8.0 140000. 6.0
  140000. 7.0 140000. 8.0 160000. 7.0 160000. 8.0
  180000. 7.0 180000. 8.0 200000. 7.0 200000. 8.0
# the frequency/wavelength grid, nfreq points
  1.0100000e-05  3.0396596e-04  6.0762797e-04  9.1129000e-04  9.8728144e-04
                   ...  <505 more frequency points>
  1.7655824e+02  1.7940598e+02  1.8225368e+02
# the SED for model 1, nfreq flux points
  6.2701018e-11  5.5260639e-08  2.1959737e-07  4.9197354e-07  5.7711065e-07
                       ...  <505 more flux points>
  9.0885815e-37  2.7039771e-37  7.9828291e-38
                 ... <the SED's for models 2 ... 19>
# the SED for model 20, nfreq flux points
  1.4960523e-10  1.3383193e-07  5.3344235e-07  1.1975753e-06  1.4049202e-06
                       ...  <505 more flux points>
  2.1177483e-13  1.3848120e-13  9.0415034e-14
```

number formatting are not recognized in C! First, using "D-format" numbers (e.g. 1.0d+20) is not allowed as C does not recognize this notation. Second, Fortran may omit the "e" in numbers with large exponents (e.g. 1.0-102). This notation is not recognized either by C. The "e" in the exponential notation may be either lowercase or uppercase. In the ascii file, first the model parameters must be supplied. For syntax version 20060612 this means that *nmod* $\times$ *npar* numbers must be entered. First all parameters for model #1, then the parameters for model #2, etc. For syntax version 20160614 *nmod* $\times$ *npar+1* items must be entered. First all parameters for model #1 followed by a track ident (e.g. A1, in general a character followed by a number), then all parameters for model #2 followed by a different track ident, etc. The latter format is used for the CoStar grids. Next follows a block of *nfreq* numbers defining the frequency/wavelength grid. Finally, *nmod* blocks, each containing *nfreq* numbers, must be entered which give the fluxes for each of the atmosphere models. This then concludes the ascii file.

B.3 Compiling the grid

Ascii files containing the stellar atmosphere grid can have arbitrary names as long as the filename extension is "`.ascii`". In the remainder we will assume the file has been named `usergrid.ascii`. Cloudy will read this file every time it needs to interpolate an SED. For small files this is a negligible overhead, but for large files this will noticeably slow down the code. The alleviate this problem, you can compile the ascii file into an index file with a name ending in "`.idx`". This file holds offsets into the ascii file telling the code where each new SED starts. Using that, the code can simply skip the SEDs it doesn't need, making the reading process much faster. The presence of this file is optional, but is strongly recommended for large grids. The index file is in principle OS and compiler dependent, so it needs to be recreated for every computer that you wish to use it on if either the OS or the compiler are different. In practice the index files will often be identical, but nothing in the standard guarantees that. The files will definitely be different on UNIX/Linux and Windows systems. In order to compile the file, type the following command in your shell:

```
cloudy.exe -e 'compile stars "usergrid.ascii"'
```

CLOUDY will now create a file `usergrid.idx`. You may need to move this file to a location in your search path. See Section 3.3 for details. You can now use the grid by including the following in your input file:

```
table star "usergrid.ascii" 2.4e4 4.3
```

In the case of a Teff-log(g) grid (see Section B.2 for a definition) you can supply only Teff, and log(g) will default to the highest value in the grid. In all other cases you need to supply exactly *ndim* parameters. See Section 6.13 for further details.

B.4 Runtime behavior

When CLOUDY encounters the following command in its input:

```
table star "usergrid.ascii" 24300 4.3
```

it will try to interpolate the requested model using using as few models as possible. This means that if an exact match could be found in the grid, only that model will be used. If an exact match could be found for one parameter, but not another, then only interpolation in the latter parameter will be performed. In general no extrapolation will be performed. The only exception is the log(g) parameter in Teff-log(g) grids for which more relaxed rules are used. More on that later. First we will return to the example above. Let's assume that the usergrid contains models for Teff = {... 24000, 25000 ...} and log(g) = {... 4.0, 4.5 ...}. Then CLOUDY will perform interpolation using the following 4 models: (Teff, log(g)) = (24000, 4.0), (24000, 4.5), (25000, 4.0), and (25000, 4.5). It will perform linear interpolation on those models using the log of the flux. If any of the models it needs for the interpolation is not present in the grid, the interpolation will fail and CLOUDY will stop. However, more relaxed rules exist for log(g) in Teff-log(g) grids (see Section B.2 for a definition). If it cannot find a model with a given log(g), it will substitute the model with the nearest log(g) instead. The reasoning behind this is that in general the spectral energy distribution is not very sensitive to the value of log(g).

Appendix C

LATEX STYLE AND MACROS

C.1 Blocks of verbatim text

```
3198CellPeak1.00E+00   Lo 9.99e-09=912.21cm   Hi-Con:1.20E+02 Ryd   E(hi):7.35E+06Ryd     E(hi):     100.01 MeV
```

Table C.1 gives naming convention for labels.

C.2 Macros

Table C.2 lists some of the macros defined in `macros.tex`.

Table C.1: Naming convention for labels

Tables	tab:TableLabel
Equations	eqn:EquationLabel
Figures	fig:FigureLabel
Section	sec:SectionLabel

Table C.2: Macros defined in `macros.tex`

\Cloudy	CLOUDY	the name of the code
\Hazy	HAZY	the name of this document
\cdCommand{command}	**command**	a command
\cdFilename{file name}	`file name`	a file name
\cdTerm{jargon}	***jargon***	a defined term
\experimental		indicates experimental code
\cdVariable{variable name}	*variable name*	name of variable
\cdRoutine{routine name}	*routine name*	routine name
\cdSectionTitle{Section of Hazy}	*Section of Hazy*	title of a section of Hazy
\cdMono{typewriter text}	`typewriter text`	text in monospaced format

REFERENCES

N. P. Abel, C. L. Brogan, G. J. Ferland, C. R. O'Dell, G. Shaw, and T. H. Troland. Physical Conditions in Orion's Veil. *ApJ*, 609:247–260, July 2004. doi: 10.1086/421009.

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.

N. P. Abel, P. A. M. v. Hoof, G. Shaw, G. J. Ferland, and T. Elwert. Sensitivity of PDR Calculations to Microphysical Details. *ApJ*, 686:1125–1136, October 2008. doi: 10.1086/591505.

T. F. Adams. The Escape of Resonance-Line Radiation from Extremely Opaque Media. *ApJ*, 174:439, June 1972. doi: 10.1086/151503.

C. W. Allen. *Astrophysical quantities*. London: Athlone, 1973.

C. W. Allen. *Astrophysical Quantities*. London: Athlone (3rd edition), 1976.

C. Allende Prieto, D. L. Lambert, and M. Asplund. The Forbidden Abundance of Oxygen in the Sun. *ApJL*, 556:L63–L66, July 2001. doi: 10.1086/322874.

C. Allende Prieto, D. L. Lambert, and M. Asplund. A Reappraisal of the Solar Photospheric C/O Ratio. *ApJL*, 573:L137–L140, July 2002. doi: 10.1086/342095.

L. H. Aller and S. J. Czyzak. Chemical compositions of planetary nebulae. *ApJS*, 51:211–247, February 1983. doi: 10.1086/190846.

A. C. Allison and A. Dalgarno. Photodissociation of Vibrationally Excited H_2, HD, and D_2 by Absorption into the Continua of the Lyman and Werner Systems. *Atomic Data*, 1:91, 1969. doi: 10.1016/S0092-640X(69)80021-5.

N. Arimoto and Y. Yoshii. Chemical and photometric properties of a galactic wind model for elliptical galaxies. *A&A*, 173:23–38, February 1987.

K. A. Arnaud. XSPEC: The First Ten Years. In G. H. Jacoby and J. Barnes, editors, *Astronomical Data Analysis Software and Systems V*, volume 101 of *Astronomical Society of the Pacific Conference Series*, page 17, 1996.

Y. Ascasibar and A. I. Díaz. Photoionized gas in hydrostatic equilibrium: the role of gravity. *MNRAS*, 404: 275–282, May 2010. doi: 10.1111/j.1365-2966.2010.16270.x.

M. Asplund, N. Grevesse, A. J. Sauval, and P. Scott. The Chemical Composition of the Sun. *ARA&A*, 47: 481–522, September 2009. doi: 10.1146/annurev.astro.46.060407.145222.

A. M. Atoyan and F. A. Aharonian. On the mechanisms of gamma radiation in the Crab Nebula. *MNRAS*, 278:525–541, January 1996.

G. Audi and A. H. Wapstra. The 1995 update to the atomic mass evaluation. *Nuclear Physics A*, 595: 409–480, February 1995. doi: 10.1016/0375-9474(95)00445-9.

Y. Avni and H. Tananbaum. X-ray properties of optically selected QSOs. *ApJ*, 305:83–99, June 1986. doi: 10.1086/164230.

Y. Avni, D. M. Worrall, and W. A. Morgan, Jr. The Underlying Distribution of Optical to X-Ray Spectral Index for Quasars in the Presence of Detection Thresholds. *ApJ*, 454:673, December 1995. doi: 10.1086/176520.

N. R. Badnell and M. J. Seaton. On the importance of inner-shell transitions for opacity calculations. *Journal of Physics B Atomic Molecular Physics*, 36:4367–4385, November 2003. doi: 10.1088/0953-4075/36/21/015.

N. R. Badnell, M. A. Bautista, K. Butler, F. Delahaye, C. Mendoza, P. Palmeri, C. J. Zeippen, and M. J. Seaton. Updated opacities from the Opacity Project. *MNRAS*, 360:458–464, June 2005. doi: 10.1111/j.1365-2966.2005.08991.x.

N. R. Badnell, F. Guzmán, S. Brodie, R. J. R. Williams, P. A. M. van Hoof, M. Chatzikos, and G. J. Ferland. H, He-like recombination spectra - IV. Clarification and refinement of methodology for l-changing collisions. *MNRAS*, 507(2):2922–2929, October 2021. doi: 10.1093/mnras/stab2266.

S. Bajtlik, R. C. Duncan, and J. P. Ostriker. Quasar ionization of Lyman-alpha clouds - The proximity effect, a probe of the ultraviolet background at high redshift. *ApJ*, 327:570–583, April 1988. doi: 10.1086/166217.

E. L. O. Bakes and A. G. G. M. Tielens. The photoelectric heating mechanism for very small graphitic grains and polycyclic aromatic hydrocarbons. *ApJ*, 427:822–838, June 1994. doi: 10.1086/174188.

S. A. Balbus and C. F. McKee. The evaporation of spherical clouds in a hot gas. III - Suprathermal evaporation. *ApJ*, 252:529–552, January 1982. doi: 10.1086/159581.

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.

R. P. Bauman, R. L. Porter, G. J. Ferland, and K. B. MacAdam. J-Resolved He I Emission Predictions in the Low-Density Limit. *ApJ*, 628:541–554, July 2005. doi: 10.1086/430665.

M. A. Bautista and T. R. Kallman. Recombination Spectra of Helium-like Ions. *ApJ*, 544:581–591, November 2000. doi: 10.1086/317206.

J. Bechtold, R. J. Weymann, Z. Lin, and M. A. Malkan. The integrated ultraviolet radiation field from QSOs. *ApJ*, 315:180–197, April 1987. doi: 10.1086/165123.

J. Bergeron and S. Souffrin. Optical line spectrum of a gas heated by hard UV radiation or energetic particles. *A&A*, 14:167–188, September 1971.

E. A. Bergin, W. D. Langer, and P. F. Goldsmith. Gas-phase chemistry in dense interstellar clouds including grain surface molecular depletion and desorption. *ApJ*, 441:222–243, March 1995. doi: 10.1086/175351.

F. Bertoldi and B. T. Draine. Nonequilibrium Photodissociation Regions: Ionization-Dissociation Fronts. *ApJ*, 458:222, February 1996. doi: 10.1086/176805.

E. Bica. Population synthesis in galactic nuclei using a library of star clusters. *A&A*, 195:76–92, April 1988.

L. Binette, A. Prieto, E. Szusziewicz, and W. Zheng. Relation between the ionizing continuum and the emission lines in Fairall 9. *ApJ*, 343:135–141, August 1989. doi: 10.1086/167691.

J. H. Black. Heating and cooling of the interstellar gas. In *Interstellar Processes*, volume 134 of *Astrophysics and Space Science Library*, pages 731–744. Hollenbach, D. J. and Thronson, Jr., H. A., 1987. doi: 10.1007/978-94-009-3861-8_27.

K. J. Borkowski and J. P. Harrington. A grain-heated, dusty planetary nebula in M22. *ApJ*, 379:168–176, September 1991. doi: 10.1086/170493.

M. Bottorff and G. Ferland. Dissipative Heating and Quasar Emission Lines. *ApJ*, 568:581–591, April 2002. doi: 10.1086/339058.

M. Bottorff, J. Lamothe, E. Momjian, E. Verner, D. Vinković, and G. Ferland. Luminosity Indicators in Dusty Photoionized Environments. *PASP*, 110:1040–1045, September 1998.

M. C. Bottorff, J. A. Baldwin, G. J. Ferland, J. W. Ferguson, and K. T. Korista. He II Reverberation in Active Galactic Nucleus Spectra. *ApJ*, 581:932–947, December 2002. doi: 10.1086/344408.

M. C. Bottorff, G. J. Ferland, and J. P. Straley. Two-Photon Transitions and Continuous Emission from Hydrogenic Species. *PASP*, 118:1176–1179, August 2006. doi: 10.1086/506974.

J. D. Bregman, L. J. Allamandola, F. C. Witteborn, A. G. G. M. Tielens, and T. R. Geballe. The infrared emission bands. II - A spatial and spectral study of the Orion bar. *ApJ*, 344:791–798, September 1989. doi: 10.1086/167844.

M. Brocklehurst. The line spectra of helium in gaseous nebulae. *MNRAS*, 157:211, 1972.

D. A. G. Bruggeman. Berechnung verschiedener physikalischer Konstanten von heterogenen Substanzen. I. Dielektrizitätskonstanten und Leitfähigkeiten der Mischkörper aus isotropen Substanzen. *Annalen der Physik*, 416:636–664, 1935. doi: 10.1002/andp.19354160705.

M. G. Burton, D. J. Hollenbach, and A. G. G. M. Tielens. Line emission from clumpy photodissociation regions. *ApJ*, 365:620–639, December 1990. doi: 10.1086/169516.

A. G. W. Cameron. Elemental and Nuclidic Abundances in the Solar System. In *Essays in Nuclear Astrophysics*, page 23, 1982.

J. A. Cardelli. The Abundance of Heavy Elements in Interstellar Gas. *Science*, 265:209, July 1994. doi: 10.1126/science.265.5169.209.

J. A. Cardelli, B. D. Savage, F. C. Bruhweiler, A. M. Smith, D. C. Ebbets, K. R. Sembach, and U. J. Sofia. First results from the Goddard High-Resolution Spectrograph - Elemental abundances in the diffuse clouds toward XI Persei. *ApJL*, 377:L57–L60, August 1991. doi: 10.1086/186116.

S. Cazaux and A. G. G. M. Tielens. Molecular Hydrogen Formation in the Interstellar Medium. *ApJL*, 575: L29–L32, August 2002. doi: 10.1086/342607.

F. H. Chaffee, Jr. and R. E. White. A survey of interstellar neutral potassium. I - Abundances and physical conditions in clouds toward 188 early-type stars. *ApJS*, 50:169–198, November 1982. doi: 10.1086/190824.

M. Chatzikos, G. J. Ferland, R. J. R. Williams, R. Porter, and P. A. M. van Hoof. Effects of External Radiation Fields on Line Emission—Application to Star-forming Regions. *ApJ*, 779:122, December 2013. doi: 10.1088/0004-637X/779/2/122.

M. Chatzikos, G. J. Ferland, R. J. R. Williams, and A. C. Fabian. On the Observability of Optically Thin Coronal Hyperfine Structure Lines. *ApJ*, 787:96, June 2014. doi: 10.1088/0004-637X/787/2/96.

M. Chatzikos, R. J. R. Williams, G. J. Ferland, R. E. A. Canning, A. C. Fabian, J. S. Sanders, P. A. M. van Hoof, R. M. Johnstone, M. Lykins, and R. L. Porter. Implications of coronal line emission in NGC 4696*. *MNRAS*, 446:1234–1244, January 2015. doi: 10.1093/mnras/stu2173.

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 2023. doi: 10.22201/ia.01851101p.2023.59.02.12.

V. Ya. Chekhovskoy. Enthalpy and thermodynamic properties of SiC at temperatures up to 2900 K. *J. Chem. Thermodynamics*, 3:289–296, 1971.

J. Clavel and M. Santos-Lleo. The turn-over of the 'big bump' in Fairall 9 and HS 1700 + 6416 - A genuine difference between Seyfert galaxies and quasars? *A&A*, 230:3–10, April 1990.

R. E. S. Clegg and J. P. Harrington. The photo-ionization of He I (2 3S) in nebulae. *MNRAS*, 239:869–883, August 1989.

R. E. S. Clegg, P. J. Storey, J. R. Walsh, and L. Neale. Measurement of the $^{12}C/^{13}C$ ratio in planetary nebulae. *MNRAS*, 284:348–358, January 1997.

L. L. Cowie and A. Songaila. High-resolution optical and ultraviolet absorption-line studies of interstellar gas. *ARA&A*, 24:499–535, 1986. doi: 10.1146/annurev.aa.24.090186.002435.

G. Crinklaw, S. R. Federman, and C. L. Joseph. The depletion of calcium in the interstellar medium. *ApJ*, 424:748–753, April 1994. doi: 10.1086/173927.

R. Cruddace, F. Paresce, S. Bowyer, and M. Lampton. On the opacity of the interstellar medium to ultrasoft X-rays and extreme-ultraviolet radiation. *ApJ*, 187:497–504, February 1974. doi: 10.1086/152659.

A. Dalgarno, M. Yan, and W. Liu. Electron Energy Deposition in a Gas Mixture of Atomic and Molecular Hydrogen and Helium. *ApJS*, 125:237–256, November 1999. doi: 10.1086/313267.

K. Davidson. On photoionization analyses of emission spectra of quasars. *ApJ*, 218:20–32, November 1977. doi: 10.1086/155653.

K. Davidson and R. A. Fesen. Recent developments concerning the Crab Nebula. *ARA&A*, 23:119–146, 1985. doi: 10.1146/annurev.aa.23.090185.001003.

N. L. D'Cruz, C. L. Sarazin, and J. Dubau. Excitation of the 3.071 mm Hyperfine Line in Li-like ^{57}Fe in Astrophysical Plasmas. *ApJ*, 501:414–424, July 1998. doi: 10.1086/305806.

S. Deguchi and W. D. Watson. Excitation of the hyperfine transitions of atomic hydrogen, deuterium, and ionized helium 3 by Lyman-alpha radiation. *ApJ*, 290:578–586, March 1985. doi: 10.1086/163014.

K. P. Dere. Ionization rate coefficients for the elements hydrogen through zinc. *A&A*, 466:771–792, May 2007.

K. P. Dere, E. Landi, H. E. Mason, B. C. Monsignori Fossi, and P. R. Young. CHIANTI - an atomic database for emission lines. *A&AS*, 125, October 1997. doi: 10.1051/aas:1997368.

F.-X. Desert, F. Boulanger, and J. L. Puget. Interstellar dust models for extinction and emission. *A&A*, 237: 215–236, October 1990.

B. T. Draine. Scattering by Interstellar Dust Grains. II. X-Rays. *ApJ*, 598:1026–1037, December 2003. doi: 10.1086/379123.

B. T. Draine and F. Bertoldi. Structure of Stationary Photodissociation Fronts. *ApJ*, 468:269, September 1996. doi: 10.1086/177689.

B. T. Draine and A. Li. Infrared Emission from Interstellar Dust. I. Stochastic Heating of Small Grains. *ApJ*, 551:807–824, April 2001. doi: 10.1086/320227.

B. T. Draine and A. Li. Infrared Emission from Interstellar Dust. IV. The Silicate-Graphite-PAH Model in the Post-Spitzer Era. *ApJ*, 657:810–837, March 2007. doi: 10.1086/511055.

B. T. Draine and B. Sutin. Collisional charging of interstellar grains. *ApJ*, 320:803–817, September 1987. doi: 10.1086/165596.

H. W. Drawin. Influence of atom-atom collisions on the collisional-radiative ionization and recombination coefficients of hydrogen plasmas. *Zeitschrift fur Physik*, 225:483–493, October 1969. doi: 10.1007/BF01392775.

E. Dwek, R. G. Arendt, D. J. Fixsen, T. J. Sodroski, N. Odegard, J. L. Weiland, W. T. Reach, M. G. Hauser, T. Kelsall, S. H. Moseley, R. F. Silverberg, R. A. Shafer, J. Ballester, D. Bazell, and R. Isaacman. Detection and Characterization of Cold Interstellar Dust and Polycyclic Aromatic Hydrocarbon Emission, from COBE Observations. *ApJ*, 475:565, February 1997. doi: 10.1086/303568.

J. E. Dyson and D. A. Williams. *The physics of the interstellar medium, 2nd ed.* The graduate series in astronomy. Bristol: Institute of Physics Publishing, 1997. ISBN 0750303069.

M. Elitzur and G. J. Ferland. Radiation pressure and emission clouds around active galactic nuclei. *ApJ*, 305:35–44, June 1986. doi: 10.1086/164226.

M. Elvis, B. J. Wilkes, J. C. McDowell, R. F. Green, J. Bechtold, S. P. Willner, M. S. Oey, E. Polomski, and R. Cutri. Atlas of quasar energy distributions. *ApJS*, 95:1–68, November 1994. doi: 10.1086/192093.

T. Elwert, G. J. Ferland, W. J. Henney, and R. J. R. Williams. Modeling dissociation of molecular hydrogen. In *IAU Symposium*, volume 235 of *IAU Symposium*, page 132P, 2005.

V. G. Farafonov. Light Scattering by Multilayer Ellipsoids in the Rayleigh Approximation. *Optics and Spectroscopy*, 88:441–443, March 2000. doi: 10.1134/1.626814.

S. R. Federman, A. E. Glassgold, and J. Kwan. Atomic to molecular hydrogen transition in interstellar clouds. *ApJ*, 227:466–473, January 1979. doi: 10.1086/156753.

S. R. Federman, Y. Sheffer, D. L. Lambert, and R. L. Gilliland. Detection of boron, cobalt, and other weak interstellar lines toward Zeta Ophiuchi. *ApJL*, 413:L51–L54, August 1993. doi: 10.1086/186957.

S. R. Federman, J. M. C. Rawlings, S. D. Taylor, and D. A. Williams. Synthesis of interstellar CH^+ without OH. *MNRAS*, 279:L41–L46, April 1996.

J. W. Ferguson and G. J. Ferland. Accurate Hydrogen Spectral Simulations with a Compact Model Atom. *ApJ*, 479:363, April 1997. doi: 10.1086/303880.

J. W. Ferguson, K. T. Korista, D. A. Verner, and G. J. Ferland. Spectral Simulations from a Model Hydrogenic Atom. In G. Ferland and D. W. Savin, editors, *Spectroscopic Challenges of Photoionized Plasmas*, volume 247 of *Astronomical Society of the Pacific Conference Series*, page 287, 2001.

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. Quasars and Starbursts in Young Giant Elliptical Galaxies. In E. Perez, R. M. Gonzalez Delgado, and G. Tenorio-Tagle, editors, *Star Formation Through Time*, volume 297 of *Astronomical Society of the Pacific Conference Series*, page 69, 2003.

G. J. Ferland. N III line emission in planetary nebulae - Continuum fluorescence. *ApJL*, 389:L63–L65, April 1992. doi: 10.1086/186349.

G. J. Ferland. A masing forbidden Fe XI line. *ApJS*, 88:49–52, September 1993. doi: 10.1086/191814.

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 and M. Elitzur. Radiation pressure and the stability of broad-line region clouds. *ApJL*, 285: L11–L14, October 1984. doi: 10.1086/184354.

G. J. Ferland and R. F. Mushotzky. Broad line region clouds and the absorbing material in NGC 4151. *ApJ*, 262:564–577, November 1982. doi: 10.1086/160448.

G. J. Ferland and R. F. Mushotzky. Cosmic rays and the emission-line regions of active galactic nuclei. *ApJ*, 286:42–52, November 1984. doi: 10.1086/162574.

G. J. Ferland and S. E. Persson. Implications of Ca II emission for physical conditions in the broad-line region of active galactic nuclei. *ApJ*, 347:656–673, December 1989. doi: 10.1086/168156.

G. J. Ferland and G. A. Shields. Heavy element abundances of Nova Cygni 1975. *ApJ*, 226:172–185, November 1978. doi: 10.1086/156597.

G. J. Ferland, G. A. Shields, and H. Netzer. Asymmetries of the emission lines of QSOs, Seyfert galaxies, and novae. *ApJ*, 232:382–388, September 1979. doi: 10.1086/157297.

G. J. Ferland, D. L. Lambert, M. L. McCall, G. A. Shields, and M. H. Slovak. Physical conditions in the accretion disk of V603 Aquilae. *ApJ*, 260:794–806, September 1982. doi: 10.1086/160298.

G. J. Ferland, B. M. Peterson, K. Horne, W. F. Welsh, and S. N. Nahar. Anisotropic line emission and the geometry of the broad-line region in active galactic nuclei. *ApJ*, 387:95–108, March 1992. doi: 10.1086/171063.

G. J. Ferland, A. C. Fabian, and R. M. Johnstone. The Physical Conditions Within Dense Cold Clouds in Cooling Flows. *MNRAS*, 266:399, January 1994.

G. J. Ferland, K. T. Korista, D. A. Verner, and A. Dalgarno. Charge Transfer between Neutral Atoms and Highly Ionized Species: Implications for ISO Observations. *ApJL*, 481:L115+, June 1997. doi: 10.1086/310655.

G. J. Ferland, A. C. Fabian, and R. M. Johnstone. The physical conditions within dense cold clouds in cooling flows - II. *MNRAS*, 333:876–884, July 2002. doi: 10.1046/j.1365-8711.2002.05470.x.

G. J. Ferland, A. C. Fabian, N. A. Hatch, R. M. Johnstone, R. L. Porter, P. A. M. van Hoof, and R. J. R. Williams. Collisional heating as the origin of filament emission in galaxy clusters. *MNRAS*, 392: 1475–1502, February 2009a. doi: 10.1111/j.1365-2966.2008.14153.x.

G. J. Ferland, C. Hu, J.-M. Wang, J. A. Baldwin, R. L. Porter, P. A. M. van Hoof, and R. J. R. Williams. Implications of Infalling Fe II-Emitting Clouds in Active Galactic Nuclei: Anisotropic Properties. *ApJL*, 707:L82–L86, December 2009b. doi: 10.1088/0004-637X/707/1/L82.

G. J. Ferland, R. Kisielius, F. P. Keenan, P. A. M. van Hoof, V. Jonauskas, M. L. Lykins, R. L. Porter, and R. J. R. Williams. Expanded Iron UTA Spectra—Probing the Thermal Stability Limits in AGN Clouds. *ApJ*, 767:123, April 2013a. doi: 10.1088/0004-637X/767/2/123.

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 2013b.

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.

G. B. Field, D. W. Goldsmith, and H. J. Habing. Cosmic-Ray Heating of the Interstellar Gas. *ApJL*, 155: L149+, March 1969. doi: 10.1086/180324.

P. J. Francis. The continuum slopes and evolution of active galactic nuclei. *ApJ*, 407:519–524, April 1993. doi: 10.1086/172533.

T. Fujimoto. Semi-empirical cross sections and rate coefficients for excitation and ionization by electron collision and photoionization of helium. Technical Report IPPJ-AM-8, Nagoya University, 1978.

A. E. Glassgold and W. D. Langer. Model calculations for diffuse molecular clouds. *ApJ*, 193:73–91, October 1974.

O. Gnat and A. Sternberg. Time-dependent Ionization in Radiatively Cooling Gas. *ApJS*, 168:213–230, February 2007. doi: 10.1086/509786.

P. Goldreich and J. Kwan. Molecular Clouds. *ApJ*, 189:441–454, May 1974. doi: 10.1086/152821.

N. Grevesse and E. Anders. Solar-system abundances of the elements - A new table. In C. J. Waddington, editor, *Cosmic Abundances of Matter*, volume 183 of *American Institute of Physics Conference Series*, pages 1–8, 1989. doi: 10.1063/1.38013.

N. Grevesse and A. Noels. Cosmic abundances of the elements. In S. Kubono and T. Kajino, editors, *Origin and Evolution of the Elements*, pages 15–25, January 1993.

N. Grevesse and A. J. Sauval. Standard Solar Composition. *Space Science Reviews*, 85:161–174, May 1998. doi: 10.1023/A:1005161325181.

N. Grevesse, M. Asplund, A. J. Sauval, and P. Scott. The chemical composition of the Sun. *Ap&SS*, 328: 179–183, July 2010. doi: 10.1007/s10509-010-0288-z.

M. F. Gu, T. Holczer, E. Behar, and S. M. Kahn. Inner-Shell Absorption Lines of Fe VI-Fe XVI: A Many-Body Perturbation Theory Approach. *ApJ*, 641:1227–1232, April 2006. doi: 10.1086/500640.

P. Guhathakurta and B. T. Draine. Temperature fluctuations in interstellar grains. I - Computational method and sublimation of small grains. *ApJ*, 345:230–244, October 1989. doi: 10.1086/167899.

Chamani M. Gunasekera, Xihan Ji, Marios Chatzikos, Renbin Yan, and Gary Ferland. Self-consistent grain depletions and abundances I: the Orion Nebula as a test case. *MNRAS*, 512(2):2310–2317, May 2022. doi: 10.1093/mnras/stac022.

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.

F. Guzmán, N. R. Badnell, R. J. R. Williams, P. A. M. van Hoof, M. Chatzikos, and G. J. Ferland. H-, He-like recombination spectra – II.l-changing collisions for He Rydberg states. *MNRAS*, 464: 312–320, January 2017. doi: 10.1093/mnras/stw2304.

F. Guzmán, M. Chatzikos, P. A. M. van Hoof, Dana S. Balser, M. Dehghanian, N. R. Badnell, and G. J. Ferland . H-, He-like recombination spectra - III. n-changing collisions in highly excited Rydberg states and their impact on the radio, IR, and optical recombination lines. *MNRAS*, 486(1): 1003–1018, Jun 2019. doi: 10.1093/mnras/stz857.

F. Haardt and P. Madau. Radiative Transfer in a Clumpy Universe. II. The Ultraviolet Extragalactic Background. *ApJ*, 461:20, April 1996. doi: 10.1086/177035.

F. Haardt and P. Madau. Radiative Transfer in a Clumpy Universe. IV. New Synthesis Models of the Cosmic UV/X-Ray Background. *ApJ*, 746:125, February 2012. doi: 10.1088/0004-637X/746/2/125.

E. Habart, F. Boulanger, L. Verstraete, C. M. Walmsley, and G. Pineau des Forêts. Some empirical estimates of the H_2 formation rate in photon-dominated regions. *A&A*, 414:531–544, February 2004. doi: 10.1051/0004-6361:20031659.

H. J. Habing. The interstellar radiation density between 912 A and 2400 A. *BAN*, 19:421, January 1968.

F. Hamann and G. Ferland. The Chemical Evolution of QSOs and the Implications for Cosmology and Galaxy Formation. *ApJ*, 418:11, November 1993. doi: 10.1086/173366.

R. J. Hanisch, A. Farris, E. W. Greisen, W. D. Pence, B. M. Schlesinger, P. J. Teuben, R. W. Thompson, and A. Warnock, III. Definition of the Flexible Image Transport System (FITS). *A&A*, 376:359–380, September 2001. doi: 10.1051/0004-6361:20010923.

J. E. Hansen and L. D. Travis. Light scattering in planetary atmospheres. *Space Science Reviews*, 16: 527–610, October 1974. doi: 10.1007/BF00168069.

J. P. Harrington. The scattering of resonance-line radiation in the limit of large optical depth. *MNRAS*, 162: 43, 1973.

T. I. Hasegawa and E. Herbst. New gas-grain chemical models of quiescent dense interstellar clouds - The effects of H2 tunnelling reactions and cosmic ray induced desorption. *MNRAS*, 261:83–102, March 1993.

T. I. Hasegawa, E. Herbst, and C. M. Leung. Models of gas-grain chemistry in dense interstellar clouds with complex organic molecules. *ApJS*, 82:167–195, September 1992. doi: 10.1086/191713.

C. Heiles and R. Crutcher. Magnetic Fields in Diffuse HI and Molecular Clouds. In R. Wielebinski and R. Beck, editors, *Cosmic Magnetic Fields*, volume 664 of *Lecture Notes in Physics, Berlin Springer Verlag*, page 137, 2005. doi: 10.1007/11369875_7.

C. Heiles and T. H. Troland. The Millennium Arecibo 21 Centimeter Absorption-Line Survey. IV. Statistics of Magnetic Field, Column Density, and Turbulence. *ApJ*, 624:773–793, May 2005. doi: 10.1086/428896.

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. J. Hester. The Crab Nebula: An Astrophysical Chimera. *ARA&A*, 46:127–155, September 2008.

D. J. Hollenbach, T. Takahashi, and A. G. G. M. Tielens. Low-density photodissociation regions. *ApJ*, 377: 192–209, August 1991. doi: 10.1086/170347.

H. Holweger. Photospheric Abundances: Problems, Updates, Implications. In R. F. Wimmer-Schweingruber, editor, *Joint SOHO/ACE workshop "Solar and Galactic Composition"*, volume 598 of *American Institute of Physics Conference Series*, page 23, 2001. doi: 10.1063/1.1433974.

E. N. Hubbard and R. C. Puetter. Theoretical quasar emission-line ratios. VII - Energy-balance models for finite hydrogen slabs. *ApJ*, 290:394–410, March 1985. doi: 10.1086/162997.

D. G. Hummer. Non-coherent scattering-III. The effect of continuous absorption on the formation of spectral lines. *MNRAS*, 138:73, 1968.

D. G. Hummer and P. J. Storey. Recombination-line intensities for hydrogenic ions. I - Case B calculations for H I and He II. *MNRAS*, 224:801–820, February 1987.

W. T. Huntress, Jr. Laboratory studies of bimolecular reactions of positive ions in interstellar clouds, in comets, and in planetary atmospheres of reducing composition. *ApJS*, 33:495–514, April 1977. doi: 10.1086/190439.

S. Ikeuchi and J. P. Ostriker. Evolution of the intergalactic medium - What happened during the epoch Z = 3-10? *ApJ*, 301:522–543, February 1986. doi: 10.1086/163921.

N. Indriolo, T. R. Geballe, T. Oka, and B. J. McCall. H_3^+ in Diffuse Interstellar Clouds: A Tracer for the Cosmic-Ray Ionization Rate. *ApJ*, 671:1736–1747, December 2007. doi: 10.1086/523036.

E. B. Jenkins. Element abundances in the interstellar atomic material. In D. J. Hollenbach and H. A. Thronson, Jr., editors, *Interstellar Processes*, volume 134 of *Astrophysics and Space Science Library*, pages 533–559, 1987.

Edward B. Jenkins. A Unified Representation of Gas-Phase Element Depletions in the Interstellar Medium. *ApJ*, 700(2):1299–1348, August 2009. doi: 10.1088/0004-637X/700/2/1299.

R. M. Johnstone, A. C. Fabian, A. C. Edge, and P. A. Thomas. The spectral signature of the cooling flow in Abell 478. *MNRAS*, 255:431–440, April 1992.

M. Jura. Interstellar clouds containing optically thin H2. *ApJ*, 197:575–580, May 1975. doi: 10.1086/153545.

J. S. Kaastra, P.-O. Petrucci, M. Cappi, N. Arav, E. Behar, S. Bianchi, J. Bloom, A. J. Blustin, G. Branduardi-Raymont, E. Costantini, M. Dadina, R. G. Detmers, J. Ebrero, P. G. Jonker, C. Klein, G. A. Kriss, P. Lubiński, J. Malzac, M. Mehdipour, S. Paltani, C. Pinto, G. Ponti, E. M. Ratti, R. A. N. Smith, K. C. Steenbrugge, and C. P. de Vries. Multiwavelength campaign on Mrk 509. I. Variability and spectral energy distribution. *A&A*, 534:A36, October 2011. doi: 10.1051/0004-6361/201116869.

J. B. Kaler and G. H. Jacoby. Central star temperatures of low-excitation planetary nebulae. *ApJ*, 372: 215–224, May 1991. doi: 10.1086/169967.

W. Kalkofen, editor. *Numerical radiative transfer*. Cambridge: University Press, 1987.

T. Kallman and M. Bautista. Photoionization and High-Density Gas. *ApJS*, 133:221–253, March 2001. doi: 10.1086/319184.

T. Kato. Radiation from a hot, thin plasma from 1 to 250 A. *ApJS*, 30:397–449, April 1976. doi: 10.1086/190367.

V. Khaire and R. Srianand. New synthesis models of consistent extragalactic background light over cosmic time. *MNRAS*, 484:4174–4199, April 2019. doi: 10.1093/mnras/stz174.

G. S. Khromov. Planetary nebulae. *Space Science Reviews*, 51:339–423, December 1989. doi: 10.1007/BF00172024.

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.

J. B. Kingdon and G. J. Ferland. Temperature Fluctuations in Photoionized Nebulae. *ApJ*, 450:691, September 1995. doi: 10.1086/176175.

J. B. Kingdon and G. J. Ferland. The Effects of Charge Transfer on the Thermal Equilibrium of Photoionized Nebulae. *ApJL*, 516:L107–L109, May 1999. doi: 10.1086/312008.

D. Kiselman. NLTE effects on oxygen lines. *New Astronomy Review*, 45:559–563, July 2001. doi: 10.1016/S1387-6473(01)00127-0.

R. Kisielius, A. Hibbert, G. J. Ferland, M. E. Foord, S. J. Rose, P. A. M. van Hoof, and F. P. Keenan. Inner-shell photoexcitation of FeXV and FeXVI. *MNRAS*, 344:696–706, September 2003. doi: 10.1046/j.1365-8711.2003.06801.x.

K. Korista, J. Baldwin, G. Ferland, and D. Verner. An Atlas of Computed Equivalent Widths of Quasar Broad Emission Lines. *ApJS*, 108:401, February 1997a. doi: 10.1086/312966.

K. Korista, G. Ferland, and J. Baldwin. Do the Broad Emission Line Clouds See the Same Continuum That We See? *ApJ*, 487:555, October 1997b. doi: 10.1086/304659.

J. H. Krolik, C. F. McKee, and C. B. Tarter. Two-phase models of quasar emission line regions. *ApJ*, 249: 422–442, October 1981. doi: 10.1086/159303.

R. L. Kurucz. Model atmospheres for G, F, A, B, and O stars. *ApJS*, 40:1–340, May 1979. doi: 10.1086/190589.

R. L. Kurucz. New Lines, New Models, New Colors. In A. G. D. Philip, A. R. Upgren, and K. A. Janes, editors, *Precision Photometry: Astrophysics of the Galaxy*, page 27, 1991.

F. La Franca, A. Franceschini, S. Cristiani, and R. Vio. On the relationship between the optical and X-ray luminosities of quasars. *A&A*, 299:19, July 1995.

D. L. Lambert. Quantitative stellar spectroscopy with large optical telescopes. *Physica Scripta Volume T*, 47:186–198, 1993.

E. Landi, G. Del Zanna, P. R. Young, K. P. Dere, and H. E. Mason. CHIANTI - An Atomic Database for Emission Lines. XII. Version 7 of the Database. *ApJ*, 744:99, January 2012. doi: 10.1088/0004-637X/744/2/99.

A. Laor and B. T. Draine. Spectroscopic constraints on the properties of dust in active galactic nuclei. *ApJ*, 402:441–468, January 1993. doi: 10.1086/172149.

J. Le Bourlot. Ortho to para conversion of H_2 on interstellar grains. *A&A*, 360:656–662, August 2000.

J. Le Bourlot, G. Pineau des Forêts, and D. R. Flower. The cooling of astrophysical media by H_2. *MNRAS*, 305:802–810, May 1999. doi: 10.1046/j.1365-8711.1999.02497.x.

J. Le Bourlot, F. Le Petit, C. Pinto, E. Roueff, and F. Roy. Surface chemistry in the interstellar medium. I. H_2 formation by Langmuir-Hinshelwood and Eley-Rideal mechanisms. *A&A*, 541:A76, May 2012. doi: 10.1051/0004-6361/201118126.

V. S. Lebedev and I. L. Beigman. *Physics of Highly Excited Atoms and Ions*. Springer, 1998.

T.-G. Lee, N. Balakrishnan, R. C. Forrey, P. C. Stancil, D. R. Schultz, and G. J. Ferland. State-to-state rotational transitions in H_2+H_2 collisions at low temperatures. *JChPh*, 125(11):114302, September

2006. doi: 10.1063/1.2338319.

T.-G. Lee, N. Balakrishnan, R. C. Forrey, P. C. Stancil, G. Shaw, D. R. Schultz, and G. J. Ferland. Rotational Quenching Rate Coefficients for H_2 in Collisions with H_2 from 2 to 10,000 K. *ApJ*, 689: 1105–1111, December 2008. doi: 10.1086/592560.

C. Leitherer, D. Schaerer, J. D. Goldader, R. M. G. Delgado, C. Robert, D. F. Kune, D. F. de Mello, D. Devost, and T. M. Heckman. Starburst99: Synthesis Models for Galaxies with Active Star Formation. *ApJS*, 123:3–40, July 1999. doi: 10.1086/313233.

A. Li and B. T. Draine. Infrared Emission from Interstellar Dust. II. The Diffuse Interstellar Medium. *ApJ*, 554:778–802, June 2001. doi: 10.1086/323147.

F. Lique. Revisited study of the ro-vibrational excitation of H_2 by H: towards a revision of the cooling of astrophysical media. *MNRAS*, 453:810–818, October 2015. doi: 10.1093/mnras/stv1683.

K. Lodders. Solar System Abundances and Condensation Temperatures of the Elements. *ApJ*, 591: 1220–1247, July 2003. doi: 10.1086/375492.

K. Lodders, H. Palme, and H.-P. Gail. Abundances of the Elements in the Solar System. *Landolt Börnstein*, page 44, 2009. doi: 10.1007/978-3-540-88055-4_34.

V. Luridiana, S. Simón-Díaz, M. Cerviño, R. M. G. Delgado, R. L. Porter, and G. J. Ferland. Fluorescent Excitation of Balmer Lines in Gaseous Nebulae: Case D. *ApJ*, 691:1712–1728, February 2009. doi: 10.1088/0004-637X/691/2/1712.

M. L. Lykins, G. J. Ferland, R. L. Porter, P. A. M. van Hoof, R. J. R. Williams, and O. Gnat. Radiative cooling in collisionally ionized and photoionized plasmas. *MNRAS*, 429:3133–3143, March 2013. doi: 10.1093/mnras/sts570.

D. C. V. Mallik and M. Peimbert. Filling factor determinations and their effects on planetary nebula studies. *Revista Mexicana de Astronomia y Astrofisica*, 16:111–121, December 1988.

P. R. Maloney, D. J. Hollenbach, and A. G. G. M. Tielens. X-Ray–irradiated Molecular Gas. I. Physical Processes and General Results. *ApJ*, 466:561, July 1996. doi: 10.1086/177532.

G. Marsaglia and W. W. Tsang. The Ziggurat Method for Generating Random Variables. *Journal of Statistical Software*, 5:8, October 2000. doi: 10.18637/jss.v005.i08.

P. G. Martin and F. Rouleau. Extreme Ultraviolet Opacity with Interstellar Dust. In R. F. Malina and S. Bowyer, editors, *Extreme Ultraviolet Astronomy*, page 341, 1991.

W. C. Martin and R. Zalubas. Energy levels of aluminum, Al I through Al XIII. *Journal of Physical and Chemical Reference Data*, 8:817–864, July 1979. doi: 10.1063/1.555608.

J. C. Mather, D. J. Fixsen, R. A. Shafer, C. Mosier, and D. T. Wilkinson. Calibrator Design for the COBE Far-Infrared Absolute Spectrophotometer (FIRAS). *ApJ*, 512:511–520, February 1999. doi: 10.1086/306805.

W. G. Mathews and G. J. Ferland. What heats the hot phase in active nuclei? *ApJ*, 323:456–467, December 1987. doi: 10.1086/165843.

J. S. Mathis. Abundances of N, S, and He, and relative stellar temperatures, in low-excitation nebulae. *ApJ*, 261:195–199, October 1982. doi: 10.1086/160330.

J. S. Mathis, W. Rumpl, and K. H. Nordsieck. The size distribution of interstellar grains. *ApJ*, 217: 425–433, October 1977. doi: 10.1086/155591.

F. Matteucci and L. Greggio. Relative roles of type I and II supernovae in the chemical enrichment of the interstellar gas. *A&A*, 154:279–287, January 1986.

F. Matteucci and A. Tornambe. Chemical evolution of elliptical galaxies. *A&A*, 185:51–60, October 1987.

J. C. Maxwell Garnett. Colours in metal glasses and metal films. *Philos. Trans. R. Soc. London, Sect. A*, 203:385–420, June 1904. doi: 10.1098/rsta.1904.0024.

B. J. McCall, A. J. Huneycutt, R. J. Saykally, T. R. Geballe, N. Djuric, G. H. Dunn, J. Semaniak, O. Novotny, A. Al-Khalili, A. Ehlerding, F. Hellberg, S. Kalhori, A. Neau, R. Thomas, F. Österdahl, and M. Larsson. An enhanced cosmic-ray flux towards ζ Persei inferred from a laboratory study of the H_3^+-e^- recombination rate. *Nature*, 422:500–502, April 2003. doi: 10.1038/nature01498.

C. F. McKee. The Dynamical Structure and Evolution of Giant Molecular Clouds. In C. J. Lada and N. D. Kylafis, editors, *NATO ASIC Proc. 540: The Origin of Stars and Planetary Systems*, page 29, 1999.

M. Mehdipour, J. S. Kaastra, G. A. Kriss, M. Cappi, P.-O. Petrucci, K. C. Steenbrugge, N. Arav, E. Behar, S. Bianchi, R. Boissay, G. Branduardi-Raymont, E. Costantini, J. Ebrero, L. Di Gesu, F. A. Harrison, S. Kaspi, B. De Marco, G. Matt, S. Paltani, B. M. Peterson, G. Ponti, F. Pozo Nuñez, A. De Rosa, F. Ursini, C. P. de Vries, D. J. Walton, and M. Whewell. Anatomy of the AGN in NGC 5548. I. A global model for the broadband spectral energy distribution. *A&A*, 575:A22, March 2015. doi: 10.1051/0004-6361/201425373.

R. Mewe. Interpolation Formulae for the Electron Impact Excitation of Ions in the H-, He-, Li-, and Ne-Sequences. *A&A*, 20:215, August 1972.

D. Mihalas. *Non-LTE model atmospheres for B and O stars*, volume NCAR/STR-76 of *Colorado, Tech. Notes*. Boulder, Colorado: Natl. Cent. Atmos. Res., 1972. 7+114 p.

D. Mihalas. *Stellar atmospheres /2nd edition/*. San Francisco, W. H. Freeman and Co., 1978. 650 p.

J. S. Mihalszki and G. J. Ferland. Density inhomogeneities and the deduced chemical composition of planetary nebulae. *PASP*, 95:284–288, May 1983. doi: 10.1086/131159.

K. Mitsuda, H. Inoue, K. Koyama, K. Makishima, M. Matsuoka, Y. Ogawara, K. Suzuki, Y. Tanaka, N. Shibazaki, and T. Hirano. Energy spectra of low-mass binary X-ray sources observed from TENMA. *PASJ*, 36:741–759, 1984.

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.

R. Morrison and D. McCammon. Interstellar photoelectric absorption cross sections, 0.03-10 keV. *ApJ*, 270:119–122, July 1983. doi: 10.1086/161102.

J. F. Navarro, C. S. Frenk, and S. D. M. White. The Structure of Cold Dark Matter Halos. *ApJ*, 462:563, May 1996. doi: 10.1086/177173.

D. Nikolić, T. W. Gorczyca, K. T. Korista, G. J. Ferland, and N. R. Badnell. Suppression of Dielectronic Recombination due to Finite Density Effects. *ApJ*, 768:82, May 2013. doi: 10.1088/0004-637X/768/1/82.

E. Novotny. *Introduction to stellar atmospheres and interiors*. New York: Oxford University Press, 1973.

H. Nussbaumer and P. J. Storey. Dielectronic recombination at low temperatures. II Recombination coefficients for lines of C, N, O. *A&AS*, 56:293–312, June 1984.

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

D. E. Osterbrock and G. J. Ferland. *Astrophysics of gaseous nebulae and active galactic nuclei, 2nd. ed.* Sausalito, CA: University Science Books, 2006.

J. P. Ostriker and S. Ikeuchi. Physical properties of the intergalactic medium and the Lyman-alpha absorbing clouds. *ApJL*, 268:L63–L68, May 1983. doi: 10.1086/184030.

P. J. E. Peebles. *Physical cosmology*. Princeton Series in Physics. Princeton, N.J.: Princeton University Press, 1971.

E. W. Pellegrini, J. A. Baldwin, C. L. Brogan, M. M. Hanson, N. P. Abel, G. J. Ferland, H. B. Nemala, G. Shaw, and T. H. Troland. A Magnetically Supported Photodissociation Region in M17. *ApJ*, 658: 1119–1135, April 2007. doi: 10.1086/511258.

R. M. Pengelly and M. J. Seaton. Recombination spectra, II. *MNRAS*, 127:165, 1964.

D. Pequignot and M. Dennefeld. The Crab Nebula. *A&A*, 120:249–262, April 1983.

D. Pequignot, P. Petitjean, and C. Boisson. Total and effective radiative recombination coefficients. *A&A*, 251:680–688, November 1991.

I. C. Percival and D. Richards. Cross-sections and rates for electron excitation of excited positively-charged hydrogen and hydrogenic ions. *MNRAS*, 183:329–334, May 1978.

M. Pettini and D. V. Bowen. A New Measurement of the Primordial Abundance of Deuterium: Toward Convergence with the Baryon Density from the Cosmic Microwave Background? *ApJ*, 560:41–48, October 2001. doi: 10.1086/322510.

R. L. Porter and G. J. Ferland. Revisiting He-like X-Ray Emission-Line Plasma Diagnostics. *ApJ*, 664: 586–595, July 2007. doi: 10.1086/518882.

R. L. Porter, R. P. Bauman, G. J. Ferland, and K. B. MacAdam. Theoretical He I Emissivities in the Case B Approximation. *ApJL*, 622:L73–L75, March 2005. doi: 10.1086/429370.

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.

R. L. Porter, G. J. Ferland, P. J. Storey, and M. J. Detisch. Improved He I emissivities in the case B approximation. *MNRAS*, 425:L28–L31, September 2012. doi: 10.1111/j.1745-3933.2012.01300.x.

R. L. Porter, G. J. Ferland, P. J. Storey, and M. J. Detisch. Erratum: 'Improved He I emissivities in the Case B approximation'. *MNRAS*, 433:L89–L90, June 2013. doi: 10.1093/mnrasl/slt049.

R. C. Puetter. Theoretical quasar emission line ratios. IV - General asymptotic escape probabilities and the effects of linear Stark broadening. *ApJ*, 251:446–450, December 1981. doi: 10.1086/159482.

T. Rauch. Implication of light metals (Li-Ca) on NLTE model atmospheres of compact hot stars. *A&A*, 320:237–248, April 1997.

T. Rauch. A grid of synthetic ionizing spectra for very hot compact stars from NLTE model atmospheres. *A&A*, 403:709–714, May 2003. doi: 10.1051/0004-6361:20030412.

C. D. Rodgers and A. P. Williams. Integrated absorption of a spectral line with the Voigt profile. *JQSRT*, 14:319–323, April 1974. doi: 10.1016/0022-4073(74)90113-7.

M. Röllig, N. P. Abel, T. Bell, F. Bensch, J. Black, G. J. Ferland, B. Jonkheid, I. Kamp, M. J. Kaufman, J. Le Bourlot, F. Le Petit, R. Meijerink, O. Morata, V. Ossenkopf, E. Roueff, G. Shaw, M. Spaans, A. Sternberg, J. Stutzki, W.-F. Thi, E. F. van Dishoeck, P. A. M. van Hoof, S. Viti, and M. G. Wolfire. A photon dominated region code comparison study. *A&A*, 467:187–206, May 2007. doi: 10.1051/0004-6361:20065918.

M. Röllig, R. Szczerba, V. Ossenkopf, and C. Glück. Full SED fitting with the KOSMA-τ PDR code. I. Dust modelling. *A&A*, 549:A85, January 2013. doi: 10.1051/0004-6361/201118190.

K. J. R. Rosman and P. D. P. Taylor. Isotopic Compositions of the Elements 1997. *Journal of Physical and Chemical Reference Data*, 27:1275–1287, November 1998. doi: 10.1063/1.556031.

F. Rouleau and P. G. Martin. Shape and clustering effects on the optical properties of amorphous carbon. *ApJ*, 377:526–540, August 1991. doi: 10.1086/170382.

T. Rowan. *Functional Stability Analysis of Numerical Algorithms*. PhD thesis, Univ. Texas at Austin, January 1990.

R. H. Rubin. The effect of heavy element opacity on the structure of H II regions. *ApJ*, 274:671–676, November 1983. doi: 10.1086/161480.

R. H. Rubin, J. P. Simpson, M. R. Haas, and E. F. Erickson. Axisymmetric model of the ionized gas in the Orion Nebula. *ApJ*, 374:564–579, June 1991. doi: 10.1086/170145.

G. B. Rybicki and A. P. Lightman. *Radiative processes in astrophysics*. New York, Wiley-Interscience, 1979. 393 p.

F. L. Schöier, F. F. S. van der Tak, E. F. van Dishoeck, and J. H. Black. An atomic and molecular database for analysis of submillimetre line observations. *A&A*, 432:369–379, March 2005.

A. Schuster. Radiation Through a Foggy Atmosphere. *ApJ*, 21:1, January 1905. doi: 10.1086/141186.

W. A. Schutte, A. G. G. M. Tielens, and L. J. Allamandola. Theoretical modeling of the infrared fluorescence from interstellar polycyclic aromatic hydrocarbons. *ApJ*, 415:397–414, September 1993. doi: 10.1086/173173.

S. Sciortino, G. S. Vaiana, F. R. Harnden, Jr., M. Ramella, C. Morossi, R. Rosner, and J. H. M. M. Schmitt. Relationship between optical and X-ray properties of O-type stars surveyed with the Einstein Observatory. *ApJ*, 361:621–643, October 1990. doi: 10.1086/169225.

M. J. Seaton. The Impact Parameter Method for Electron Excitation of Optically Allowed Atomic Transitions. *Proceedings of the Physical Society*, 79:1105–1117, June 1962. doi: 10.1088/0370-1328/79/6/304.

K. Sellgren, A. T. Tokunaga, and Y. Nakada. The 3.3 micron feature, H2, and ionized gas in the Orion bar. *ApJ*, 349:120–125, January 1990. doi: 10.1086/168299.

F. H. Sellmaier, T. Yamamoto, A. W. A. Pauldrach, and R. H. Rubin. A possible solution for the [Ne III] problem in HII regions. *A&A*, 305:L37+, January 1996.

N. I. Shakura and R. A. Sunyaev. Black holes in binary systems. Observational appearance. *A&A*, 24: 337–355, 1973.

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.

G. Shaw, G. J. Ferland, R. Srianand, N. P. Abel, P. A. M. van Hoof, and P. C. Stancil. On the Enhanced Cosmic-Ray Ionization Rate in the Diffuse Cloud toward ζ Persei. *ApJ*, 675:405–412, March 2008. doi: 10.1086/526395.

Gargi Shaw and G. J. Ferland. Role of Polycyclic Aromatic Hydrocarbons on the Cosmic-Ray ionization rate in the Galaxy. *arXiv e-prints*, art. arXiv:2101.03732, January 2021.

J. M. Shull. Heating and ionization by X-ray photoelectrons. *ApJ*, 234:761–764, December 1979. doi: 10.1086/157553.

J. M. Shull and M. E. van Steenberg. X-ray secondary heating and ionization in quasar emission-line clouds. *ApJ*, 298:268–274, November 1985. doi: 10.1086/163605.

M. Sikora, M. C. Begelman, and B. Rudak. Relativistic neutrons in active galactic nuclei. *ApJL*, 341: L33–L36, June 1989. doi: 10.1086/185451.

T. P. Snow and A. N. Witt. Interstellar Depletions Updated: Where All the Atoms Went. *ApJL*, 468:L65+, September 1996. doi: 10.1086/310225.

T. P. Snow, Jr. and S. L. Dodgen. A search for interstellar scandium. *ApJ*, 237:708–710, May 1980. doi: 10.1086/157918.

T. P. Snow, Jr. and D. G. York. The detection of interstellar fluorine in the line of sight toward Delta Scorpii. *ApJL*, 247:L39–L41, July 1981. doi: 10.1086/183585.

L. Spitzer. *Physical processes in the interstellar medium*. New York Wiley-Interscience, 1978. 333 p.

L. J. Spitzer. The Temperature of Interstellar Matter. I. *ApJ*, 107:6, January 1948. doi: 10.1086/144984.

L. J. Spitzer and M. G. Tomasko. Heating of H i Regions by Energetic Particles. *ApJ*, 152:971, June 1968. doi: 10.1086/149610.

U. Springmann. Multiple resonance line scattering and the 'momentum problem' in Wolf-Rayet star winds. *A&A*, 289:505–523, September 1994.

G. Stasińska and R. Szczerba. The dust content of planetary nebulae: a reappraisal. *A&A*, 352:297–307, December 1999.

W. Steenbock and H. Holweger. Statistical equilibrium of lithium in cool stars of different metallicity. *A&A*, 130:319–323, January 1984.

A. Sternberg and D. A. Neufeld. The Ratio of Ortho- to Para-H_2 in Photodissociation Regions. *ApJ*, 516: 371–380, May 1999. doi: 10.1086/307115.

R. Stognienko, T. Henning, and V. Ossenkopf. Optical properties of coagulated particles. *A&A*, 296:797, April 1995.

N. J. Stone. Table of nuclear magnetic dipole and electric quadrupole moments. *Atomic Data and Nuclear Data Tables*, 90:75–176, May 2005. doi: 10.1016/j.adt.2005.04.001.

P. J. Storey and D. G. Hummer. Recombination line intensities for hydrogenic ions-IV. Total recombination coefficients and machine-readable tables for Z=1 to 8. *MNRAS*, 272:41–48, January 1995. on the web at http://adc.gsfc.nasa.gov/adc-cgi/cat.pl?/catalogs/6/6064/.

R. H. Stoy. The temperatures of the nuclei of planetary nebulae. *MNRAS*, 93:588, June 1933.

A. Suchkov, R. J. Allen, and T. M. Heckman. Cosmic-ray-dominated dense molecular gas in normal and starburst galaxies. *ApJ*, 413:542–547, August 1993. doi: 10.1086/173023.

J. Takahashi. The Ortho/Para Ratio of H_2 Newly Formed on Dust Grains. *ApJ*, 561:254–263, November 2001. doi: 10.1086/322954.

J. Takahashi and H. Uehara. H_2 Emission Spectra with New Formation Pumping Models. *ApJ*, 561: 843–857, November 2001. doi: 10.1086/323364.

C. B. Tarter. *Radiative Transfer in a Gas Excited by X-Rays*. PhD thesis, Cornell University, 1967.

C. B. Tarter, W. H. Tucker, and E. E. Salpeter. The Interaction of X-Ray Sources with Optically Thin Environments. *ApJ*, 156:943, June 1969. doi: 10.1086/150026.

A. G. G. M. Tielens. Interstellar polycyclic aromatic hydrocarbon molecules. *ARA&A*, 46:289–337, September 2008. doi: 10.1146/annurev.astro.46.060407.145211.

A. G. G. M. Tielens and D. Hollenbach. Photodissociation regions. I - Basic model. II - A model for the Orion photodissociation region. *ApJ*, 291:722–754, April 1985a. doi: 10.1086/163111.

A. G. G. M. Tielens and D. Hollenbach. Photodissociation Regions - Part Two - a Model for the Orion Photodissociation Region. *ApJ*, 291:747, April 1985b. doi: 10.1086/163112.

C. A. Tout, O. R. Pols, P. P. Eggleton, and Z. Han. Zero-age main-seqence radii and luminosities as analytic functions of mass and metallicity. *MNRAS*, 281:257–262, July 1996.

P. A. M. van Hoof. *Photo-ionization studies of nebulae*. PhD thesis, Rijksuniversiteit Groningen, 1997.

P. A. M. van Hoof, J. C. Weingartner, P. G. Martin, K. Volk, and G. J. Ferland. Grains in Photo-Ionized Environments. In G. Ferland and D. W. Savin, editors, *Spectroscopic Challenges of Photoionized Plasmas*, volume 247 of *Astronomical Society of the Pacific Conference Series*, page 363, 2001.

P. A. M. van Hoof, J. C. Weingartner, P. G. Martin, K. Volk, and G. J. Ferland. Grain size distributions and photoelectric heating in ionized media. *MNRAS*, 350:1330–1341, June 2004. doi: 10.1111/j.1365-2966.2004.07734.x.

H. van Regemorter. Rate of Collisional Excitation in Stellar Atmospheres. *ApJ*, 136:906, November 1962. doi: 10.1086/147445.

H. Vedel, U. Hellsten, and J. Sommer-Larsen. Formation of disc galaxies in the presence of a background UVX radiation field. *MNRAS*, 271:743, December 1994.

D. A. Verner and D. G. Yakovlev. Analytic FITS for partial photoionization cross sections. *A&AS*, 109: 125–133, January 1995.

D. A. Verner, G. J. Ferland, K. T. Korista, and D. G. Yakovlev. Atomic Data for Astrophysics. II. New Analytic FITS for Photoionization Cross Sections of Atoms and Ions. *ApJ*, 465:487, July 1996. doi: 10.1086/177435.

G. S. Voronov. A Practical Fit Formula for Ionization Rate Coefficients of Atoms and Ions by Electron Impact: Z = 1-28. *Atomic Data and Nuclear Data Tables*, 65:1, 1997. doi: 10.1006/adnd.1997.0732.

N. V. Voshchinnikov and J. S. Mathis. Calculating Cross Sections of Composite Interstellar Grains. *ApJ*, 526:257–264, November 1999. doi: 10.1086/307997.

L. Vriens and A. H. M. Smeets. Cross-section and rate formulas for electron-impact ionization, excitation, deexcitation, and total depopulation of excited atoms. *PhRvA*, 22:940–951, September 1980. doi: 10.1103/PhysRevA.22.940.

D. Vrinceanu and M. R. Flannery. Classical and quantal collisional Stark mixing at ultralow energies. *PhRvA*, 63(3):032701, March 2001. doi: 10.1103/PhysRevA.63.032701.

D. Vrinceanu, R. Onofrio, and H. R. Sadeghpour. Angular Momentum Changing Transitions in Proton-Rydberg Hydrogen Atom Collisions. *ApJ*, 747:56, March 2012. doi: 10.1088/0004-637X/747/1/56.

W. R. Webber. A New Estimate of the Local Interstellar Energy Density and Ionization Rate of Galactic Cosmic Cosmic Rays. *ApJ*, 506:329–334, October 1998. doi: 10.1086/306222.

J. C. Weingartner and B. T. Draine. Photoelectric Emission from Interstellar Dust: Grain Charging and Gas Heating. *ApJS*, 134:263–281, June 2001a. doi: 10.1086/320852.

J. C. Weingartner and B. T. Draine. Dust Grain-Size Distributions and Extinction in the Milky Way, Large Magellanic Cloud, and Small Magellanic Cloud. *ApJ*, 548:296–309, February 2001b. doi: 10.1086/318651.

J. C. Weingartner, B. T. Draine, and D. K. Barr. Photoelectric Emission from Dust Grains Exposed to Extreme Ultraviolet and X-Ray Radiation. *ApJ*, 645:1188–1197, July 2006. doi: 10.1086/504420.

J. C. Weisheit. X-Ray Ionization Cross-Sections and Ionization Equilibrium Equations Modified by Auger Transitions. *ApJ*, 190:735–740, June 1974. doi: 10.1086/152933.

R. E. White. Interstellar lithium - Differential depletion in diffuse clouds. *ApJ*, 307:777–786, August 1986. doi: 10.1086/164463.

B. J. Wilkes, H. Tananbaum, D. M. Worrall, Y. Avni, M. S. Oey, and J. Flanagan. The Einstein database of IPC x-ray observations of optically selected and radio-selected quasars, 1. *ApJS*, 92:53–109, May 1994. doi: 10.1086/191959.

D. T. Wilkinson. Measurements of cosmic microwave radiation. In M. P. Ulmer, editor, *13th Texas Symposium on Relativistic Astrophysics*, pages 209–218, 1987.

J. P. Williams, E. A. Bergin, P. Caselli, P. C. Myers, and R. Plume. The Ionization Fraction in Dense Molecular Gas. I. Low-Mass Cores. *ApJ*, 503:689, August 1998. doi: 10.1086/306034.

R. E. Williams. The Ionization and Thermal Equilibrium of a Gas Excited by Ultraviolet Synchrotron Radiation. *ApJ*, 147:556, February 1967. doi: 10.1086/149035.

R. E. Williams. Incorporation of density fluctuations into photoionization calculations. *ApJ*, 392:99–105, June 1992. doi: 10.1086/171409.

T. L. Wilson and R. Rood. Abundances in the Interstellar Medium. *ARA&A*, 32:191–226, 1994. doi: 10.1146/annurev.aa.32.090194.001203.

D. M. Worrall, H. Tananbaum, P. Giommi, and G. Zamorani. X-ray studies of quasars with the Einstein Observatory. IV - X-ray dependence on radio emission. *ApJ*, 313:596–606, February 1987. doi: 10.1086/164999.

S. A. Wrathmall, A. Gusdorf, and D. R. Flower. The excitation of molecular hydrogen by atomic hydrogen in astrophysical media. *MNRAS*, 382:133–138, November 2007. doi: 10.1111/j.1365-2966.2007.12420.x.

Y. Xu and R. McCray. Energy degradation of fast electrons in hydrogen gas. *ApJ*, 375:190–201, July 1991. doi: 10.1086/170180.

D. G. York, M. Meneguzzi, and T. P. Snow. Upper limits for interstellar boron and beryllium abundances toward zeta Ophiuchi. *ApJ*, 255:524–526, April 1982. doi: 10.1086/159852.

G. Zamorani, J. P. Henry, T. Maccacaro, H. Tananbaum, A. Soltan, Y. Avni, J. Liebert, J. Stocke, P. A. Strittmatter, R. J. Weymann, M. G. Smith, and J. J. Condon. X-ray studies of quasars with the Einstein Observatory. II. *ApJ*, 245:357–374, April 1981. doi: 10.1086/158815.

W. Zheng, G. A. Kriss, R. C. Telfer, J. P. Grimes, and A. F. Davidsen. A Composite HST Spectrum of Quasars. *ApJ*, 475:469, February 1997. doi: 10.1086/303560.

J. Zsargó and S. R. Federman. Nonthermal Chemistry in Diffuse Clouds with Low Molecular Abundances. *ApJ*, 589:319–337, May 2003. doi: 10.1086/374617.