Atomic Data Unleashed: A Compact Database of Helium Triplet X-Ray Line Emissivities

David P. Huenemoerder, John C. Houck, Michael A. Nowak,
Norbert S. Schulz, John E. Davis

March 28, 2011

1 Motivation

The X-ray emission from highly-ionized He-like species has long been known to be an important spectral diagnostic (Gabriel & Jordan, 1969). The He-like triplet line ratios can be sensitive to temperature, density, and photoexcitation (Blumenthal, Drake & Tucker, 1972). The quality of the atomic data is crucial to physical interpretations of these lines. Recent improvements in some of the data (Chen et al., 2006) have been motivated by our ability to resolve these lines in a variety of cosmic sources (other than the Sun) with the X-ray spectrometers on $Chandra$ and $XMM-Newton$ and have led to improved diagnostic capability (Smith et al., 2009).

In Figure 1 we reproduce the energy level diagram from Gabriel & Jordan (1969). The salient feature of the He-like transitions is the presence of a metastable level, the $n = 2$, $^3S_1$ term. This has a relatively long lifetime since the electric dipole transition to $n = 1$ is forbidden. Hence, other processes, such as collisional excitation can de-populate the level, if density is high enough. Or, if there is significant ultraviolet flux, the level can be depopulated by photoexcitation. Both of these processes populate the ^3P levels which can then radiatively decay.

The spectral signature of the He-like triplets is shown in Figure 2, reproduced from Wolfson et al. (1983), along with a line ratio plot showing their diagnostic power.

Historically, the lines are denoted by the labels w (resonance line, upper level $1s2p\,^1P_1$), x (intercombination line, upper level $1s2p\,^3P_2$), y (intercombination

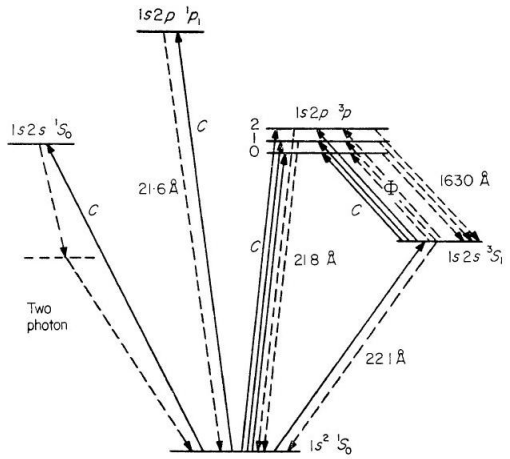

Figure 1: Energy level and term diagram for He-like ions showing the transitions between and among levels $n = 2$ and $n = 1$. The wavelengths shown are for O VII. "C" and "Φ" denote collisional and radiative excitation transitions, respectively. The resonance line (w) is at $21.6\,\text{Å}$, the intercombination doublet (x and y) is at $21.8\,\text{Å}$, and the forbidden line (z) is at $22.1\,\text{Å}$, as labeled. (From Gabriel & Jordan (1969)).

line, upper level $1s2p\,^3P_1$), and z (forbidden line, upper level $1s2s\,^3P_1$). They are also commonly referred to as r, i, and f, for resonance, intercombination, and forbidden, respectively. For a recent review, see Porquet, Dubau & Grosso (2010).

While these spectral features are very important astrophysically, their density and temperature dependence has *until now not been integrated into any publicly available, scriptable, extensible data analysis and modeling system.* There are various codes which will compute the emissivities, there are tabulations and graphs in the literature, there are digital databases of low-density emissivities, but there is no software implementation one can incorporate, for example, density as a parameter of the model for evaluation and fitting X-ray spectra. Here we remedy that situation by providing a stand-alone database of He-like coefficients for evaluating the line emissivity as a function of temperature and density, as well as a high-level spectral model for use in the ISIS spectral analysis application (Houck & Denicola, 2000; Noble & Nowak, 2008). We also provide an analogous database and interfaces for photoexcitation, in which the important parameter is UV flux density instead of electron number density. In the following we will describe the data and implementation and also provide detailed examples of use.

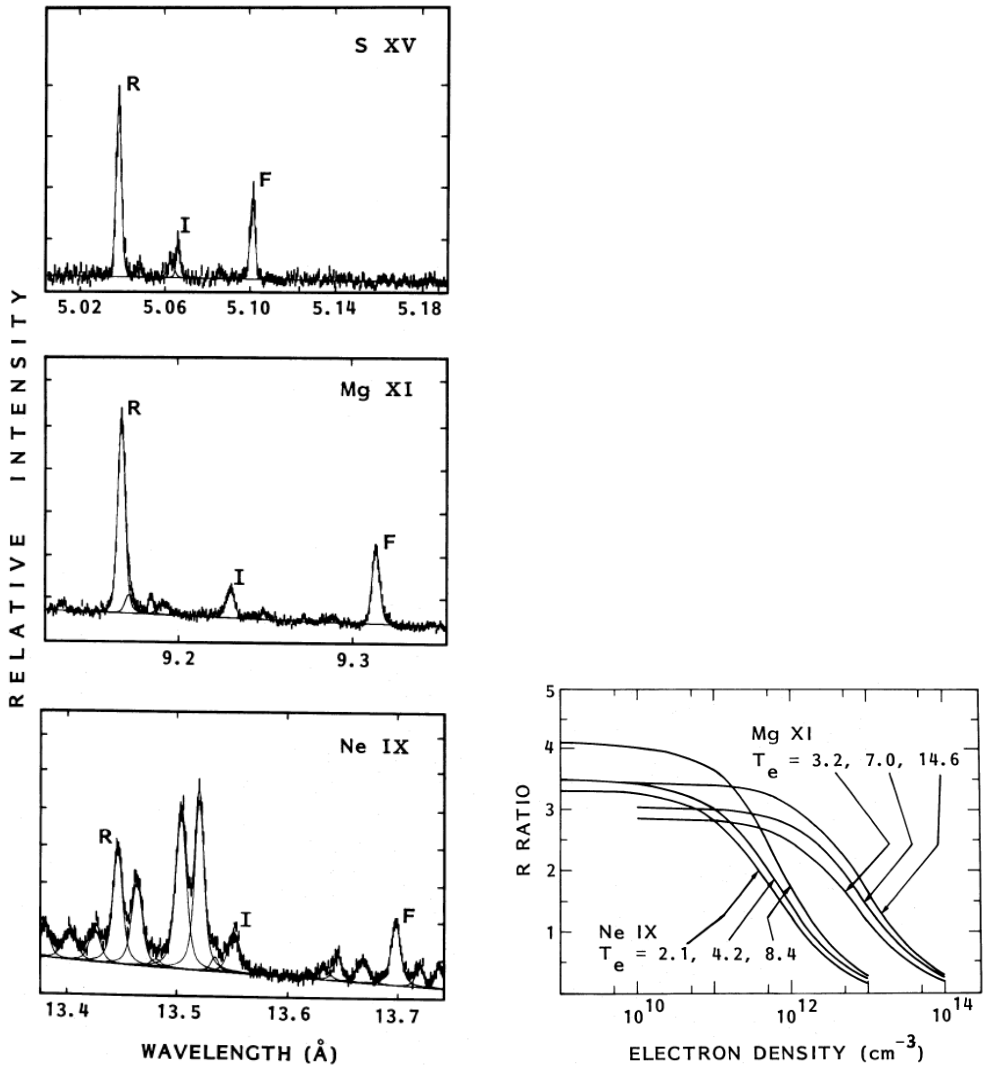

Figure 2: Solar spectra of three He-like triplet regions. The lines are marked as "R" for resonance, "I" for intercombination (a doublet), and "F" for forbidden (these are now usually denoted in lower case). On the right are the theoretical trends in the "R" ratio, defined as the ratio of the forbidden to intercombination line fluxes (not to be confused with the "R" label in the spectrum plots) as a function of density for different temperatures. (From Wolfson et al. (1983)).

2 Source Data Compilations

Two comprehensive collections of atomic data are *Chianti* (Dere et al., 1997, 2009) and the *ATOMDB*[1]. The former is a collection of atomic data and a suite of *IDL* software for using the data to compute emissivities (among other quantities). The latter includes the "Astrophysical Plasma Emission Database" (APED) and output from the "Astrophysical Plasma Emission Emission Code" (APEC; Smith & Brickhouse (2008); Smith et al. (2001)).

With the *Chianti IDL* scripts, one can compute emissivities for a specified temperature and density (or for a specified radiation field) whereas APED (versions 1.3.2 and 2.0) is a low-density-only database. While APEC can be used to compute emissivities as a function of temperature and density, the code is not publicly available.[2]

To provide the fundamental He-triplet lines' emissivity as a function of temperature and density, we have used both *Chianti* and APED data.

For the *Chianti* run (version 6; Dere et al. (2009)), we used a temperature spacing of 0.05 dex, density spacing of 0.2 dex, and computed emissivities for a set of astrophysically interesting elements. We also used *Chianti* to compute emissivities for a range of UV radiation fields for the same ions.

For APED, we have generated a custom density-dependent database using the pre-released APEC, kindly made available to us for testing by the authors and have verified the low density values against the APED 2.0 release. We used the same temperature-density grid as for *Chianti*. With this version of APEC, however, we have no way to specify a UV radiation field and so have parameterized only the dependence upon electron density.

3 Parameterization & Packaging

To provide a compact form of the He-triplet emissivities which is independent of both *Chianti* and APEC, we have fit the emissivities parametrically and tabulated the coefficients. Specifically, we fit the density dependence of the forbidden (z) and intercombination lines (x, y) emissivities at each temperature grid point for

[1] An Atomic Database, available from <http://www.atomdb.org/>

[2] APEC is being prepared for public release. Until then, custom runs are possible through private communication with the code authors at their discretion. We have generated custom databases with a pre-release version kindly provided by the code authors.

each ion. We have based our parameterization on the double-exponential functional form used by Smith et al (2002)[3]:

$$\epsilon_{jk}(n_e; T_k) = C_k \left(a_{0jk} + a_{1jk}\, e^{-n_e/a_{2jk}} + a_{3jk}\, e^{-n_e/a_{4jk}}\right). \quad (1)$$

Here we have introduced a normalization term which is the density-independent sum of the triplet emissivities:

$$C_k = \sum_j \epsilon_{jk}(n_i \sim 0, T_k), \quad (2)$$

where j enumerates w, x, y, and z, the components of the triplet, and k references a particular temperature. The emissivity of the resonance line, w, can be derived from the normalization and evaluation of the x, y, and z terms (the normalization does not vary significantly with density). The units of the emissivity, ϵ, are $[\mathrm{photons\,cm^3\,s^{-1}}]$, such that the photon emission rate per unit volume is given by $n_e\, n_{ion}\, \epsilon(T, n_e)$. Figure 3 shows an example for the fit to C VI for a temperature of 1 MK.

The coefficients are stored in a FITS binary table in which there is a single extension for each ion. In each table, there is one row per temperature, and the coefficients for the density dependence of each triplet component are stored in array columns. Table 1 lists the block structure and Table 2 lists the columns structure within a block. The (zero-based) array indices on columns `ax`, `ay`, and `az`, map directly to the integer subscripts on the a coefficients in Equation 1, and `norm` corresponds to C_k.

The same format and parameterization is used for both the number-density table and the photon-density (U_ν $[\mathrm{ergs\,cm^{-3}\,Hz^{-1}}]$) table. For the number-density dependence, we constructed two tables, one using APEC and one for *Chianti*.

4 Use in ISIS

The "Interactive Spectral Interpretation System" (ISIS; Houck & Denicola (2000); Noble & Nowak (2008)) provides interfaces to an atomic database, in particular the APED. ISIS includes both high-level access, such as generation of synthetic spectra from a plasma model, as well as low-level access to fundamental atomic

[3]Memo posted at <http://cxc.harvard.edu/atomdb/features_density.html>, "Density sensitive X-ray lines between 1.2 –31 Å" (<http://cxc.harvard.edu/atomdb/features/denHETG.ps>).

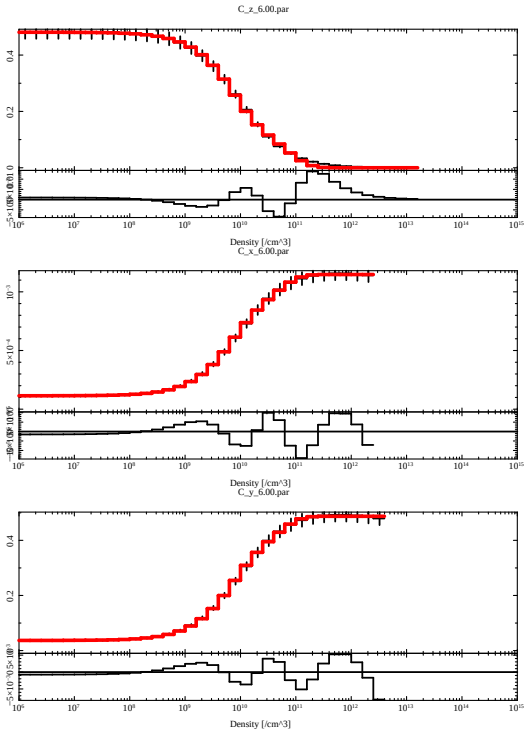

Figure 3: Example fit of the C VI emissivities of z (top), x (middle), and y (bottom) for $\log T = 6.00$. The red curve is the fit to the underlying black histogram, and below are the residuals.

Table 1: FITS file Block Structure

No.	Name	Type	Dimensions
1	PRIMARY	Null	
2	he_coefs_C	Table	5 cols x 56 rows
3	he_coefs_N	Table	5 cols x 56 rows
4	he_coefs_O	Table	5 cols x 56 rows
5	he_coefs_Ne	Table	5 cols x 56 rows
6	he_coefs_Mg	Table	5 cols x 56 rows
7	he_coefs_Si	Table	5 cols x 56 rows
8	he_coefs_S	Table	5 cols x 56 rows
9	he_coefs_Al	Table	5 cols x 56 rows
10	he_coefs_Ar	Table	5 cols x 54 rows
11	he_coefs_Ca	Table	5 cols x 51 rows

Note: the block order is arbitrary, and the number of rows may vary with block. Not all ions listed are required to be present.

Table 2: FITS file Column Structure

No.	Name	Unit	Type	Comment
1	temp	K	Real8	plasma electron temperature
2	norm	phot*cm^3/s	Real8	sum of w,x,y,z emissivity for n=0.
3	ax[5]		Real8(5)	ex(n)/norm=ax0+ax1*exp(-n/ax2)+ax3*exp(-n/ax4)
4	ay[5]		Real8(5)	ey(n)/norm=ay0+ay1*exp(-n/ay2)+ay3*exp(-n/ay4)
5	az[5]		Real8(5)	ez(n)/norm=az0+az1*exp(-n/az2)+az3*exp(-n/az4)

Note: the column order is arbitrary.

data or emissivities by element and ion as functions of temperature, and in principle, density. However, APED only contains low-density values, and ISIS does not compute rates. ISIS does, however, provide the ability to modify the emissivity of any line in the database with a custom, user-defined function. We use that feature in conjunction with our He-triplet coefficients table to provide density dependent emissivities for the astrophysically important He-like triplet lines. In this way, we can use density directly as a parameter of plasma models and as a fitting parameter.

The ISIS script, `he_modifier.sl`, provides functions to read the coefficients table, to evaluate emissivities of the He-like triplet lines, and to define an emissivity modifier which can be used in conjunction with the atomic database and plasma modeling infrastructure. The user can specify whether to use the collisional excitation or photexcitation database. The user can also specify the data source for the normalization, either from the coefficients table, or from the loaded plasma database, since there are some significant differences between *Chianti* and APEC emissivities.

The function reference is provided in the on-line documentation (see § 6, or the ISIS interactive help). Some detailed examples are provided in the Appendices.

5 Stand-alone Use

In order to demonstrate the contents of the FITS table with low-level direct access, we give an example of evaluation of emissivity vs. density of the Ne IX forbidden ("z") line at $\log T = 6.8$. Here we use the scripting language, *S-Lang* through its interactive shell, `slsh`, and we use the `cfitsio` module to read the table. This would be similar in other scripting languages (IDL, python, matlab) which have

array arithmetic, structures, and FITS table access.

```
slsh> require("fits");
slsh> t = fits_read_table(
              "he_modifier_chianti.fits[he_coefs_Ne]");
slsh> k = where( log10( t.temp ) >= 6.8 )[0];
slsh> n = 10^[9.0:14.0:0.1];
slsh> z = t.norm[k] * (t.az[k,0] + t.az[k,1]*exp(-n/t.az[k,2])
        + t.az[k,3]*exp(-n/t.az[k,4]) );
slsh> fp = fopen("zne9.tbl", "w");
slsh> for (i=0; i<length(n);i++)
            ()=fprintf(fp, "%S %S\n", n[i], z[i]);
slsh> () = fclose(fp);
```

You can read and visualize the data with your favorite plotting program. As a minimal example, we assume you have "grace"[4] installed, and can do `gracebat zne9.tbl` to produce `zne9.ps` (shown in Figure 4).

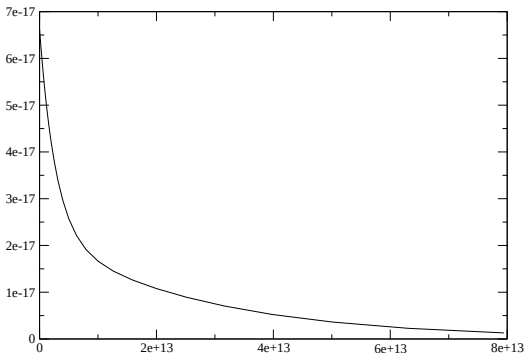

Figure 4: Example plot of emissivity vs. density output from the script above, as an example of direct file access and evaluation.

[4] `http://plasma-gate.weizmann.ac.il/Grace/`

6 Resources

The He-modifier table, ISIS script script, and examples are available from
<http://space.mit.edu/cxc/analysis/he_modifier/>
For more information on ISIS, see
<http://space.mit.edu/cxc/isis/>.
For the APED atomic database:
<http://www.atomdb.org/>.
For the *Chianti* data and IDL scripts:
<http://chianti.nrl.navy.mil/chianti.html>.

Acknowledgments Support for this work was provided by the National Aeronautics and Space Administration, Goddard Space Flight Center award NNX10AD41G to MIT through the Astrophysical Data Analysis Program. We also thank Randall Smith and Adam Foster for a pre-release version of APEC and for their help in production of a density-dependent emissivity database.

References

Blumenthal, G. R., Drake, G. W. F., & Tucker, W. H., 1972, ApJ, 172, 205

Chen, G. X., Smith, R. K., Kirby, K., Brickhouse, N. S., & Wargelin, B. J., 2006, Phys. Rev. A, 74, 042709

Dere, K. P., Landi, E., Mason, H. E., Monsignori Fossi, B. C., & Young, P. R., 1997, A&AS, 125, 149

Dere, K. P., Landi, E., Young, P. R., Del Zanna, G., Landini, M., & Mason, H. E., 2009, A&A, 498, 915

Gabriel, A. H., & Jordan, C., 1969, MNRAS, 145, 241

Houck, J. C., & Denicola, L. A., 2000, in Astronomical Data Analysis Software and Systems IX, ed. N. Manset, C. Veillet, & D. Crabtree, Vol. 216, 591

Miller, N. A., Cassinelli, J. P., Waldron, W. L., MacFarlane, J. J., & Cohen, D. H., 2002, ApJ, 577, 951

Noble, M. S., & Nowak, M. A., 2008, PASP, 120, 821

Porquet, D., Dubau, J., & Grosso, N., 2010, Space Sci. Rev., 148–+

Smith, R., & Brickhouse, N. S., 2008, in Bulletin of the American Astronomical Society, Vol. 40, 186

Smith, R. K., Brickhouse, N. S., Liedahl, D. A., & Raymond, J. C., 2001, ApJ, 556, L91

Smith, R. K., Chen, G., Kirby, K., & Brickhouse, N. S., 2009, ApJ, 700, 679

Waldron, W. L., & Cassinelli, J. P., 2001, ApJ, 548, L45

Wolfson, C. J., Leibacher, J. W., Doyle, J. G., & Phillips, K. J. H., 1983, ApJ, 269, 319

A Extended Examples

A.1 Tutorial Plasma Model ISIS Script

Here we provide a detailed example, including output, of a session using the He-modifier to evaluate density-dependent, two-temperature APED plasma models. Comments are prefixed with "%", and output will be indented (and may be omitted or truncated for clarity).

```
plasma( aped ) ;  % First load the plasma database:

% Load the modifier script; this also initializes APED database.
require ("he_modifier");

% When loaded, the program tries to find a default file and initialize
% data.  It can be initialized explicitly as:
he_modifier_init( ; collex = "path_to/he_modifier_chianti.fits" );

% Define a single-temperature plasma model named "xaped":
create_aped_fun ("xaped", default_plasma_state) ;

% Define two He-modifiers, one with all default elements, on with only
% Ne, Si, and S:
create_he_modifier ("He_a");
create_he_modifier ("He_b", "Ne Si S");

% Specify a fit function, as a sum of two components.
% Use a different He-modifier on each:
fit_fun ("xaped(1, He_a(1)) + xaped(2, He_b(1))");

% A parameter listing shows the modifier terms along with standard
%   model parameters:
list_par;
  xaped(1, He_a(1)) + xaped(2, He_b(1))
   idx  param               tie-to  freeze    value    min    max
    1  He_a(1).log10_C_ndens    0     0          10      7     16
    2  He_a(1).log10_N_ndens    0     0          11      7     16
    3  He_a(1).log10_O_ndens    0     0          11      7     16
    4  He_a(1).log10_Ne_ndens   0     0          12      7     16
    5  He_a(1).log10_Mg_ndens   0     0          13      7     16
    6  He_a(1).log10_Al_ndens   0     0          13      7     16
    7  He_a(1).log10_Si_ndens   0     0          14      7     16
    8  He_a(1).log10_S_ndens    0     0          14      7     16
    9  He_a(1).log10_Ar_ndens   0     0          14      7     16
   10  He_a(1).log10_Ca_ndens   0     0          14      7     16
```

```
   11  xaped(1).norm            0     1           1      0      0
   12  xaped(1).temperature     0     1       1e+07      0      0
   13  xaped(1).density         0     1           1      0      0
   14  xaped(1).vturb           0     1           0      0      0
   15  xaped(1).redshift        0     1           0      0      0
   16  xaped(1).metal_abund     0     1           1      0      0
   17  He_b(1).log10_Ne_ndens   0     0          12      7     16
   18  He_b(1).log10_Si_ndens   0     0          14      7     16
   19  He_b(1).log10_S_ndens    0     0          14      7     16
   20  xaped(2).norm            0     1           1      0      0
   21  xaped(2).temperature     0     1       1e+07      0      0
   22  xaped(2).density         0     1           1      0      0
   23  xaped(2).vturb           0     1           0      0      0
   24  xaped(2).redshift        0     1           0      0      0
   25  xaped(2).metal_abund     0     1           1      0      0

% Define a grid, change some parameters, and evaluate some models:
 (l, h) = linear_grid (1,23,20000);

set_par( "xaped(1).temperature",  2.e6);
set_par( "xaped(2).temperature",  4.e6);

set_par( "He_a(1).log10_O_ndens",  13);
set_par( "He_a(1).log10_Ne_ndens", 13);
set_par( "He_b(1).log10_Ne_ndens", 13);

use_thermal_profile; % provide some line breadth for better view
f = eval_fun (l, h); % evaluate for current parameters.

set_par( "He_a(1).log10_O_ndens",  10);
set_par( "He_a(1).log10_Ne_ndens", 10);
set_par( "He_b(1).log10_Ne_ndens", 10);

f2 = eval_fun (l, h);  % evaluate for changed parameters

% Open a plot device, with 2 panels; configure the plot settings:
pw2 = open_plot("/xwin", 1, 2 ) ;
resize(16, 1.4);     set_line_width(3);     charsize( 1.33 ) ;
ylin; xlin;    xlabel("wavelength"); ylabel( "flux" ) ;
s = line_label_default_style;
s.char_height = 2.0 ;
set_frame_line_width(3); charsize( 1.4 ); set_line_width( 5 ) ;
_pgscf(2);  % select nicer pgplot fonts.

% Plot the models in the Ne IX region for 2 densities;
```

```
xrange( 13.42, 13.75 );   hplot(l,h,f, 2);  ohplot( l, h, f2, 1 ) ;
plot_group( brightest( 3, where( wl(13.4, 13.75) ) ),1, s );

% In the lower panel, do the same for the O VII region:
xrange( 21.545, 22.19); hplot(l,h,f, 2);  ohplot( l, h, f2, 1 ) ;
plot_group( brightest( 3, where( wl(21.5, 22.2) ) ), 1, s );
```

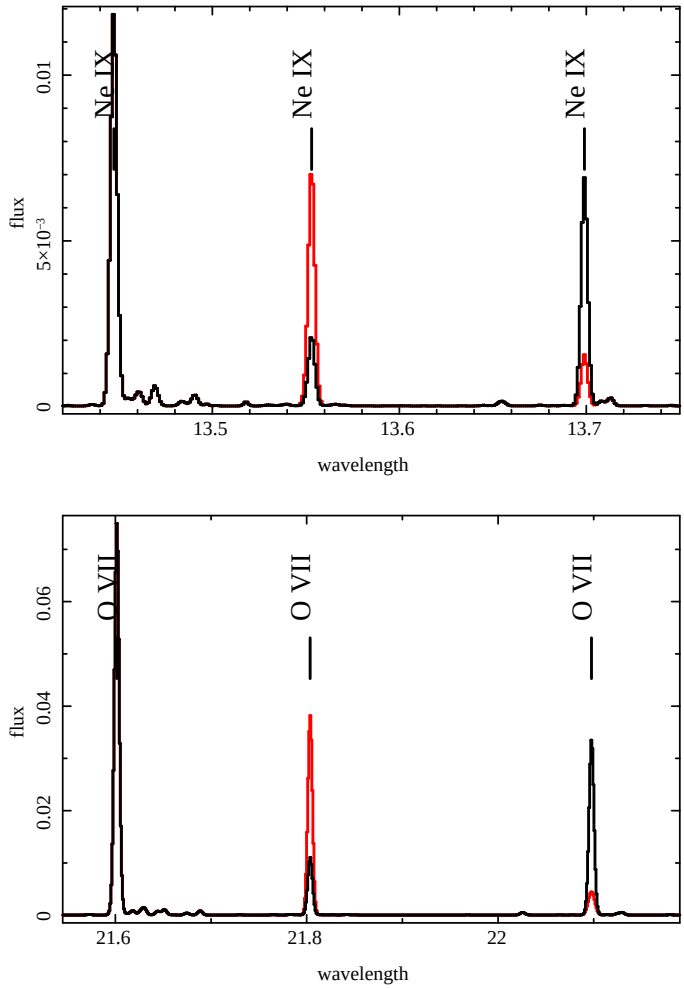

Figure 5: This shows the model spectra for two densities for Ne IX (top) and O VII (bottom). In each case, the black curves are for low density and the red are high density.

A.2 Tutorial Plasma Model with Photoexcitation ISIS Script

This is a detailed example of fitting a photoexcitation model to the O VII triplet of δ Ori (*Chandra* Observation ID 639), in order to determine the relative radius of formation. For details on this object, see Miller et al. (2002). See Waldron & Cassinelli (2001) for discussion of the physics of photoexcitation of triplets in O-stars.

We assume that basic procedures have been followed to load the spectrum and responses into ISIS.

To define the photoexcitaion model, we need the UV photon energy density which excites O VII from the forbidden level to the intercombination level. For O VII, this means at the wavelength of about 1640 Å. (Incidentally, this information is available from the APED in ISIS via: `page_group(where(trans(O, 7, [5,6], 2)));`) To get the energy density, we must rely on some external models and of the object. For the properties of δ Ori of $T_{\rm eff} = 33000$, $\log g = 3.4$, we use the TLUSTY[5] model for $T_{\rm eff} = 32500$, $\log g = 3.50$, converting H_ν via $U_\nu = (16\pi/c)\, H_\nu$, to get a value of $2.05 \times 10^{-12}\,{\rm ergs\,cm^{-3}\,Hz^{-1}}$. We will use this as a frozen parameter, then fit for the relative radius.

Comments start with a "%", and output is indented (and may be omitted or truncated for clarity).

```
plasma( aped ) ; % First load the plasma database:

% Load the modifier script; this also initializes APED database.
require ("he_modifier");

% When loaded, the program tries to find a default file and initialize
% data.  It can be initialized explicitly for photoexcitation as:
he_modifier_init(; photoex="he_photoex_modifier_chianti.fits");

% Set up a fit function using an isothermal APED plasma
% with photoexcitation; only include O VII in the parameters:
create_aped_fun( "delori", default_plasma_state ) ;
create_he_modifier( "photoex", "O"; photoex ) ;
fit_fun( "delori( 1, photoex(1; db_norm) )" ) ;

use_thermal_profile;   % we want line profile broadening enabled.
unu_o = 2.046e-12 ;    % define the UV energy density at 1640A

% Establish initial model parameters and their limits; set the temperature
% to the value of the maximum of O VII emissivity and make it frozen
% (alternatively, we could leave it free and obtain a "G-ratio" value):
```

[5] Available from
<http://nova.astro.umd.edu/Tlusty2002/tlusty-frames-BS06.html>

```
%
set_par( "delori(1).norm",        0.002, 0, 5.e-5, 0.01  ) ;
set_par( "delori(1).temperature", 2.e6, 1 ) ;
set_par( "delori(1).vturb",       1000, 0, 500, 2000 ) ;
set_par( "delori(1).redshift",    -1.e-4, 0, -0.001, 0.001 ) ;

set_par( "photoex(1).O_log10_unu", log10( unu_o ), 1 );
set_par( "photoex(1).O_r", 4, 0, 1, 14 ); %   r/r_star

% Assume data are loaded into histogram index h:
group_data( h, 5);         % bin up a bit
xnotice( h, 21.0, 22.5 ) ; % limit fitting to the O VII region

set_fit_method( "subplex" ) ;  % choose a fit method
set_fit_statistic( "cash" ) ;  % choose a statistic

() = fit_counts;     % perform the fit
list_free;           % examine resulting parameters

 delori( 1, photoex(1; db_norm) )
  idx  param           tie-to  freeze         value    min    max
   2  photoex(1).O_r       0     0          8.076768      1     14
   3  delori(1).norm       0     0      0.002320241  5e-05   0.01
   6  delori(1).vturb      0     0         1046.358    500   2000
   7  delori(1).redshift   0     0    -0.0001579913 -0.001  0.001

% set ranges, plot counts and model, mark line ids:
xrange( 21.0, 22.5 ); yrange;
rplot_counts( h ) ;
plot_group( where( trans( O, 7, [2,6,7], 1 ) ), 14 );

% Now compute confidence contours of the line width vs radius:
rg = conf_grid( "photoex(1).O_r",  1,   14,   32 );
vg = conf_grid( "delori(1).vturb", 800, 1300, 32 );
conf_rv = conf_map_counts( rg, vg ) ;

% plot the contour map:
xlabel( latex2pg( "R/R_*(O VII)") );
ylabel( latex2pg( "v_{turb}(O VII) [km/s]") );
xrange(1, 14);
yrange( 800, 1300);
plot_conf( conf_rv ) ;

% since confidence maps can take time to compute, save the map:
save_conf( conf_rv, "photoex_example_delori_confmap.fits");
```

Figure 6 shows the resulting plots of the counts spectrum and the contour map. Note that the actual value R-ratio is implicit. It can be obtained from the model as follows:

```
f = triplet_line_em( O, 2.e06,
               unu_o * W_dilution(get_par("photoex(1).O_r"))
             ; photoex, db_norm );
( f.z / (f.x+f.y) )[0,0];
0.114
```

Or it could be obtained by fitting the O VII parametrically (e.g., with a sum of Gaussians).

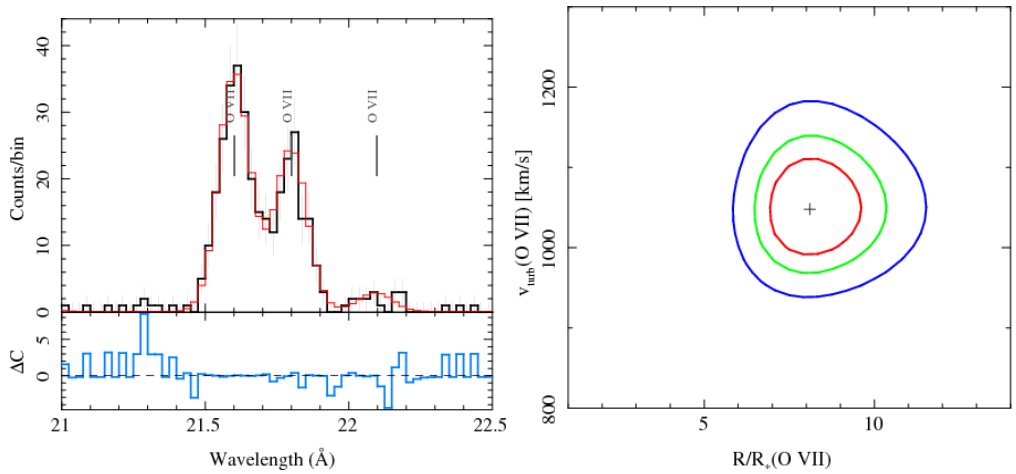

Figure 6: Photoexcitation example from Section A.2. On the left is the counts spectrum (black), model (red) and residuals. On the right are the confidence contours of the relative radius of formation against the line width. Both have significant detections.

A.3 $G - R$ Contours, Interpolating Measured $G - R$ for T, n_e

This is a detailed example which demonstrates evaluation of the G and R ratios from the database, and interpolation of measured values in the density-temperature grid. Only simple linear interpolation is used and we ignore uncertainties. An improved method would use bilinear interpolation and account for measurement uncertainties. The resulting plot is shown in Figure 7.

```
% Define a contour labeling function using pgplot primitives.
% (Note: this alters the coordinate system for the displayed plot.)
%
define c_label( z0, lev, dn, nmin, x, y)
{
    % x,y are coordinate axes for z0[iy,ix];
    % lev[] is array of contour levels to label
    % dn, nmin as as per pgplot functions:
    %       INTVAL (spacing), MININT (min cells)
    %
    variable pinf = get_plot_options;
    variable x1, x2, y1, y2 ;
    (x1, x2, y1, y2 ) = _pgqwin ;
    variable l = where( ( x >= pinf.xmin ) and ( x <= pinf.xmax ) );
    variable k = where( ( y >= pinf.ymin ) and ( y <= pinf.ymax ) );
    variable z = z0[ k, l ] ;
    variable s = array_shape( z ) ;
    variable tr = [0.0, 1.0, 0.0, 0.0, 0.0, 1.0 ] ;
    _pgswin( 1.5, s[1] + 0.5, 1.5, s[0] + 0.5 );
    variable i ;
    foreach i ( lev )
      _pgconl( z, 0, s[0]-1, 0, s[1]-1, i, tr,
               sprintf("%0.2f",i), dn, nmin);
    set_plot_options( pinf ) ;
    _pgswin( x1, x2, y1, y2 ) ;
}

% Initialize the data:
variable ht_data = "./dist/he_modifier_data/he_modifier_apec.fits";
he_modifier_init( ; collex = ht_data );

% Define temperature and density grids, then
% use them to evaluate the Ne IX emissivities:
variable tgrid = 10^[ 5.5 : 8.5  : 0.05 ] ;
variable dgrid = 10^[ 6.0 : 16.0 : 0.1 ] ;

% Evaluate theoretical emissivities
variable v = triplet_line_em( Ne, tgrid, dgrid );
```

```
% Compute the theoretical G and R ratios:
variable g = ( v.z + v.x + v.y ) / v.w ;
variable r = v.z / ( v.x + v.y ) ;

% Plot the G,R contours vs. T and n_e:
xrange( 6.2, 7.2 ) ;
yrange( 11, 13.0 ) ;
title("");
xlabel( "Log T" ) ;
ylabel( latex2pg( "Log n_e" ) );

variable g_levels = [ 0.1 : 1.2 : 0.05 ];
variable r_levels = [ 0.2: 3.8: 0.2 ] ;

color(1);
plot_contour( g, 1, log10(tgrid), log10(dgrid), g_levels );
oplot_contour( r, r_levels ) ;

% Assume some measured G, R, then find T_e, n_e:
variable gmeas = 0.92 ;
variable rmeas = 0.80 ;

% since G countours are nearly vertical, pick any n_e, say 1.e12:
% (NOTE: interpol requires ascending x arrays)
variable l = where( dgrid >= 1.e12)[0];
variable t0 = interpol( gmeas, reverse(g[l,*]), reverse(tgrid) );

% Now given T, interpolate for n:
l = where( tgrid >= t0)[0] ;
variable n0 = interpol( rmeas, reverse(r[*,l]), reverse(dgrid) );

% Mark the found point:
pointstyle( 25 ) ;
connect_points(0);
oplot( log10(t0), log10(n0), 2 ) ;

% Overplot the contour labels:
% (done last because the utility function alters the coordinate system).
_pgsci(15); _pgsch( 1 ); _pgslw( 1 );
c_label( g, g_levels, 25, 4, log10(tgrid), log10(dgrid) );
c_label( r, r_levels, 35, 4, log10(tgrid), log10(dgrid) );
```

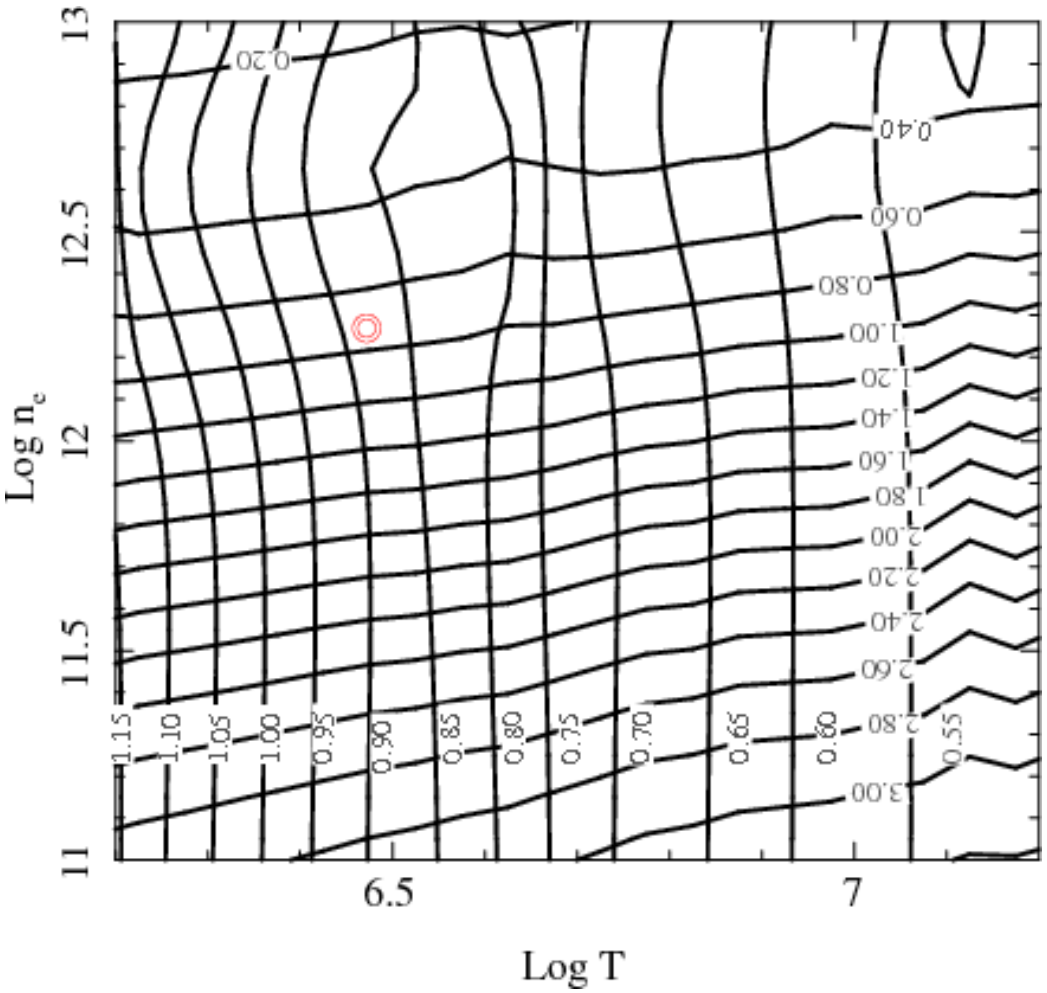

Figure 7: G-R countours for Ne IX, as a function of number density and temperature. The nearly vertical lines are contours of constant G, and the nearly horizontal are of constant R. Values are given by the labels on our near each contour.

Plots in the more common one-dimensional view can be made with the commands below, whose results are shown in Figure 8.

```
xrange( 6.2, 7.2 ) ;   xlabel( "Log T" ) ;
yrange; ylabel("G-ratio");
l = where( dgrid >= 1.e12)[0];
connect_points(1); pointstyle(-1);
plot( log10(tgrid), g[l,*] );     % theoretical
pointstyle(25); oplot( log10(t0), gmeas, 1) ;  % "measured"

xrange(10, 14) ;   xlabel( latex2pg( "Log n_e") ) ;
yrange; ylabel("R-ratio");
l = where( tgrid >= t0)[0] ;
connect_points(1); pointstyle(-1);
plot( log10(dgrid), r[*,l] );      % theoretical
pointstyle(25); oplot( log10(n0), rmeas, 1) ;   % "measured"
```

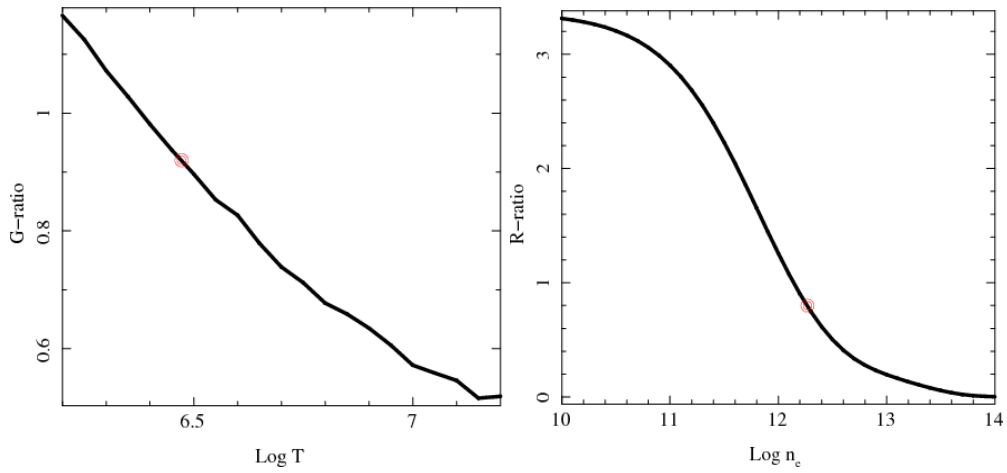

Figure 8: Example of $G(T)$ and $R(T)$ for Ne IX. Solid lines are the theoretical curves and the circle is the interpolated measurement.

B FITS Table Header Listings

Below is the full listing of the FITS file's first binary table extension header for the *Chianti*-based coefficients file. The extension name follows the pattern of "he_coefs_<element>", where "element" is the standard (case-sensitive) name from the periodic table of elements. The file will contain one or more such extensions.

```
XTENSION= 'BINTABLE' / binary table extension
BITPIX  =                    8 / 8-bit bytes
NAXIS   =                    2 / 2-dimensional binary table
NAXIS1  =                  136 / width of table in bytes
NAXIS2  =                   56 / number of rows in table
PCOUNT  =                    0 / size of special data area
GCOUNT  =                    1 / one data group (required keyword)
TFIELDS =                    5 / number of fields in each row
TTYPE1  = 'temp    ' / plasma electron temperature
TFORM1  = 'D       ' / data format of field: 8-byte DOUBLE
TTYPE2  = 'norm    ' / sum of w,x,y,z emissivity for n=0.
TFORM2  = 'D       ' / data format of field: 8-byte DOUBLE
TTYPE3  = 'ax      ' / ex(n)/norm=ax0+ax1*exp(-n/ax2)+ax3*exp(-n/ax4)
TFORM3  = '5D      ' / data format of field: 8-byte DOUBLE
TTYPE4  = 'ay      ' / ey(n)/norm=ay0+ay1*exp(-n/ay2)+ay3*exp(-n/ay4)
TFORM4  = '5D      ' / data format of field: 8-byte DOUBLE
TTYPE5  = 'az      ' / ez(n)/norm=az0+az1*exp(-n/az2)+az3*exp(-n/az4)
TFORM5  = '5D      ' / data format of field: 8-byte DOUBLE
EXTNAME = 'he_coefs_C' / name of this binary table extension
TDIM3   = '(5)     '
TDIM4   = '(5)     '
TDIM5   = '(5)     '
ATOM_NUM=                    6 / Atomic Number
TUNIT1  = 'K       ' / degrees Kelvin
TUNIT2  = 'phot*cm^3/s' / emissivity
COMMENT ------------------------------------------------------------------------
COMMENT Parameterization of emissivities of He-like triplets.
COMMENT The functional form of the parameterized emissivity, e, at
COMMENT temperature T  as a function of density is given by
COMMENT  $e(n) = norm * ( a0 + a1 * exp( -n/a2 )  +  a3 * exp( -n/a4 ) )$
COMMENT where $norm$ is in [phot*cm^3/s] and
COMMENT $n$ is the electron density in [cm^{-3}]
COMMENT This file tabulates norm, a0, a1, a2, a3, and a4 for the triplet
COMMENT components, x,y, and z as a function of element and temperature.
COMMENT The normalization term is the sum of w,x,y, and z
COMMENT emissivities at the low density limit. (w is the resonance line,
COMMENT x & y are intercombination lines, and z the forbidden line).
COMMENT The norm has negligible density dependence.
```

```
COMMENT ------------------------------------------------------------------------
COMMENT Atomic data source:  CHIANTI 6.0.1
COMMENT Atomic data ref.: Dere et al. (2009)  A&A 498, 915
COMMENT Atomic data URL:  <http://chianti.nrl.navy.mil/chianti.html>
HISTORY dens_fit_common 1.2.0
HISTORY dph@space.mit.edu
DATE    = '2010-09-14T17:33:22' / file creation date (YYYY-MM-DDThh:mm:ss UT)
CHECKSUM= 'WDMaWBJUWBJZWBJZ' / HDU checksum updated 2010-09-14T17:33:22
DATASUM = '874669351' / data unit checksum updated 2010-09-14T17:33:22
```

The FITS file's first binary table extension header for the APEC-based coefficients file is similar to the above. Differences are in the source information:

```
COMMENT Atomic data source:  APED 2.0.0.beta
COMMENT Atomic data ref.: Smith et al (2001)  ApJ 556, L91
COMMENT Atomic data URL:  <http://www.atomdb.org/>
```

The photoexcitation coefficients file is similar, with appropriate changes for photon energy density instead of number density:

```
TTYPE3  = 'ax      ' / ex(u)/norm=ax0+ax1*exp(-u/ax2)+ax3*exp(-u/ax4)
TTYPE4  = 'ay      ' / ey(u)/norm=ay0+ay1*exp(-u/ay2)+ay3*exp(-u/ay4)
TTYPE5  = 'az      ' / ez(u)/norm=az0+az1*exp(-u/az2)+az3*exp(-u/az4)
COMMENT ------------------------------------------------------------------------
COMMENT Parameterization of emissivities of He-like triplets.
COMMENT The functional form of the parameterized emissivity, e, at
COMMENT temperature T as a function of energy density,
COMMENT U(nu) [erg/cm^3/Hz] is given by
COMMENT  $e(u) = norm * ( a0 + a1 * exp( -u/a2 )  +  a3 * exp( -u/a4 ) )$
COMMENT where $norm$ is in [phot*cm^3/s] and
COMMENT $u$ is the energy density in [ergs cm^{-3} Hz^{-1}]
COMMENT ------------------------------------------------------------------------
COMMENT Atomic data source:  CHIANTI 6.0.1
COMMENT Atomic data ref.: Dere et al. (2009)  A&A 498, 915
COMMENT Atomic data URL:  <http://chianti.nrl.navy.mil/chianti.html>
```