CHIANTI

An Atomic Database for Spectroscopic Diagnostics of Astrophysical Plasmas

USER GUIDE for version 9 – 18 Mar 2019

Written by Giulio Del Zanna
with contributions from the CHIANTI team
Please send comments to g.del-zanna (at) damtp.cam.ac.uk

Contents

1 Introduction

CHIANTI is a collaborative project involving researchers based at the University of Cambridge (UK), the NASA Goddard Space Flight Center (USA), the George Mason University (GMU, USA) and the University of Michigan (USA).

1.1 What is CHIANTI

The CHIANTI package consists of a critically evaluated set of atomic data (energy levels, wavelengths, radiative transition probabilities and excitation data) for a large number of ions of astrophysical interest. It also includes a number of ancillary data and a suite of Interactive Data Language (IDL) programs to calculate optically thin synthetic spectra and to perform spectral analysis and plasma diagnostics.
Plasma emission codes have long been used to study UV and X-ray spectral lines emitted from solar or stellar atmospheres. A comparison of the theoretical line intensities with the observed intensities allows a determination of the physical parameters for the plasma (cf. the Living Review by Del Zanna and Mason, 2018).

The CHIANTI database has been used extensively by the astrophysical and solar communities to analyse emission line spectra from astrophysical sources.
The CHIANTI package is freely available at

- the main CHIANTI homepage: http://www.chiantidatabase.org/
- SolarSoft, a programming and data analysis environment for the solar physics community. http://www.lmsal.com/solarsoft/

A version of the CHIANTI programs written in Python (http://python.org, named ChiantiPy developed and maintained by K. Dere can be found at:
http://chiantipy.sourceforge.nethttp://chiantipy.sourceforge.net
Python is a modern, dynamic programming language that can be run in interactive command-line sessions within either the native Python or IPython (http://ipython.orghttp://ipython.org) environments. IPython should be preferred, as it provides a powerful, interactive Python shell.

1.2 How to acknowledge CHIANTI

The mainteinance and development of the CHIANTI database is dependent on continued funding which is generally available if we can demonstrate that the CHIANTI database is of use to astrophysical research. If you use CHIANTI, we ask that you acknowledge it appropriately in any publications:
Write in the text of any publication the reference to the CHIANTI paper associated with the particular VERSION you have used:

- 1 - Dere et al., 1997, AASS, 125, 149 http://adsabs.harvard.edu/abs/1997A%26AS..125..149D
- 2 - Landi et al. 1999, AASS, 135, 339 http://adsabs.harvard.edu/abs/1999A%26AS..135..339
- 3 - Dere et al., 2001, ApJSS, 134, 331 http://adsabs.harvard.edu/abs/2001ApJS..134..331D
- 4 - Young et al., 2002, ApJS, 144, 135 http://adsabs.harvard.edu/abs/2003ApJS..144..135Y
- 5 - Landi et al., 2005, ApJS, 162, 261 http://adsabs.harvard.edu/abs/2006ApJS..162..261L
- 6 - Dere et al., 2009, A&A, 498, 915 http://adsabs.harvard.edu/abs/2009A%26A...498..915D
- 7 - Landi et al., 2012, ApJSS, 744, 99 http://adsabs.harvard.edu/abs/2012ApJ...744...99L
- 7.1 - Landi et al., 2013, 763, 86 http://adsabs.harvard.edu/abs/2013ApJ...763...86L
- 8.0 - Del Zanna et al., 2015, A&A 582, A56 https://ui.adsabs.harvard.edu/#abs/2015A&A...5
- 9.0 - Dere et al., 2019, ApJ, in press http://adsabs.harvard.edu/abs/2019arXiv190205019D

We would appreciate if you also write in the acknowledgements of any publication the following:

CHIANTI is a collaborative project involving the University of Cambridge (UK), the NASA Goddard Space Flight Center (USA), the George Mason University (GMU, USA) and the University of Michigan (USA).
If a detail work on a particular ion is done, it would be appropriate to also refer to the original publication. References can be found at the end of each data file or on the WWW.
CHIANTI data are included into other databases. It would be appropriate to make that clear to the users so they can trace back the results they use to the original calculations.
Users should be aware of what is included in the database, of the approximations applied, and of the atomic data used. The CHIANTI results should not be blindly considered valid in all cases. For example, the CHIANTI predicted emissivities should not be used when considering temperatures outside of the validity ranges.
Any contributions and suggestions to CHIANTI are welcomed. We would appreciate a short description of how you employ CHIANTI.

1.3 Important caveats and limitations

As with any atomic data package, CHIANTI has been developed to suit some specific applications in astrophysics, and users should read the CHIANTI papers and the documentation to find out the ranges of applicability of the package.
Currently, some of the main assumptions and limitations of the data and programs are:

- Level excitations occur via collisional excitation by electrons and protons, and photoexcitation from black-body or user-defined radiation fields.

- Electrons and protons have Maxwellian distribution functions. Indeed CHIANTI data include Maxwellian-averaged electron and proton collision strengths. However, it is possible to study the effects of particle distributions that are linear combinations of Maxwellians of different temperatures.

- Electrons and protons have the same temperature.

- The plasma ionization is dominated by collisions (i.e. no photo-ionization is included).

- Atomic processes affecting the ionisation state of an element can be separated from those affecting the level balance within an ion.

  A correction to the level populations due to ionization and recombination is included, but it is only valid up to densities above which metastable level populations begin to be non-negligible.

- The plasma is in a steady state.

- All lines are optically thin.

- Line emissivities are reliable only in some (extended) temperature and density ranges. The ranges of temperatures at which the original rates were calculated are normally listed in the files.

- The current ion fractions that are provided within CHIANTI have been calculated assuming equilibrium and at zero density.

- DEM files are provided only as samples of DEM curves for different physical conditions in the solar atmosphere: the intensities they generate may differ from observed values.

1.4 A short history of the package

CHIANTI is maintained by a small team, and developments have depended on obtaining funding and managing CHIANTI among other projects.

The CHIANTI project was originally set up by Dr. Ken Dere of the Naval Research Laboratory (Washington, USA), Dr. Helen Mason of the Department of Applied Mathematics and Theoretical Physics at the University of Cambridge (UK), and Dr. Brunella Monsignori-Fossi of the Arcetri Astrophysical Observatory (Florence, Italy). Former students of Dr. Monsignori-Fossi (Dr. Enrico Landi) and Dr. Mason (Dr. Peter Young) helped in the creation of the database. The sad and unexpected death of Dr. Monsignori-Fossi in January 1995, led to Prof. Massimo Landini, a close associate of Dr. Monsignori-Fossi, becoming a new CHIANTI representative (University of Florence). Prof. Landini retired in 2010.

Additional collaborations have involved Dr. Dave Pike of the Rutherford Appleton Laboratory (RAL), who has written CHIANTI routines to run within the environment of the SOHO/CDS software (and within SolarSoft), and with Dr. Gordon Bromage, Dr. Barbara Bromage and her former student Dr. Giulio Del Zanna of the University of Central Lancashire.

Dr. Enrico Landi, now at the University of Michigan (USA) Dr. Peter Young, now at NASA Goddard Space Flight Center (USA), and Dr. Giulio Del Zanna, now at Cambridge (UK), have continued to be active collaborators in the CHIANTI project. Ken Dere and Helen Mason are retired but still contribute to the project.

1. The first version of the CHIANTI database was released in 1996 and is described in Dere et al. (1997).

   Young et al. (1998) used the CHIANTI database for a detailed comparison with observed EUV solar spectra to assess the diagnostic accuracy of the two data sets.

2. Version 2.0 (Landi et al. 1999) was released in April 1999. This Version adds atomic data for many of the so called minor ions (Na, P, Cl, K, Ti, Cr, Mn, Co, and Zn), not included in the first version. Because the astrophysical abundances of these elements are relatively low, only the strongest lines of these elements are observed. The addition of the minor ions is an important step in our goal to understand astrophsical spectra in detail. In addition, Version 2.0 extends the beryllium-like sequence, updates some of the data in Version 1, and provides an IDL procedure to calculate the continuum.

3. Version 3.0 of the CHIANTI database was released in September 2000 (Dere et al., 2001). In this version the database has been extended to wavelengths shorter than 50Å by including atomic data for the hydrogen and helium isoelectronic sequences, inner-shell transitions and satellite lines and several other ions. In addition, some of the ions already present in the database have been updated and extended with new atomic data from published calculations. The inclusion of the satellites has required a significant modification to the manner in which the spectra have been calculated with CHIANTI. Consequently, a new version of the IDL software has been produced.

   In November 2000 we have released a whole new CHIANTI package under SolarSoft.

4. Version 4 of the CHIANTI database, released in Sept. 2002 (Young et al., 2002). The major changes are the inclusion of proton excitation data, principally for ground configuration levels which are close in energy, and of photoexcitation.

   The fitting procedure for excitation data, both electrons and protons, has been extended to allow 9 point spline fits in addition to the previous 5 point spline fits. This allows higher quality fits to data from close-coupling calculations where resonances can lead to significant structure in the thermally-averaged collision strengths.

   With the addition of H I, He I and N I, the first neutral species have been added to CHIANTI.

   Many existing ion data-sets have been updated, in particular most ions of the nitrogen and beryllium isoelectronic sequences. Also, new ions have been added, including Ar IV, Fe VI and Ni XXI.

   The continuum routines have been re-written, including a new relativistic free-free continuum, a new free-bound, and a new two-photon continuum. New software has been written.

5. Version 5 of the CHIANTI database, released in August 2005, included ionization and recombination as mechanisms for populating excited levels, photoexcitation from any user-defined radiation field, software for taking into account non-Maxwellian velocity distributions, and new data for a large number of ions (both for new entries or for ions already in the database). Changes in the data consist of the inclusion of high-energy configurations of Fe XVII to XXIII to predict a large number of lines in the X-rays, n=3 configurations in the N-like and O-like sequences to predict a large number of lines at UV wavelengths, and new data for very important ions in the EUV wavelength range such as Fe IX, Fe XII and Fe XV. The software has been speeded considerably and a few more user-friendly features have been added.

6. CHIANTI version 6, released in July 2009, includes a complete database of ionization and recombination coefficients for the first 30 elements of the periodic table, as well as a new ionization equilibrium. We also provide IDL programs to read the rates.

   Major features (Dere et al. 2009, A&A, 498, 915) in CHIANTI 6 are:

   - inclusion of ionization and recombination rates
   - new ionization equilibrium calculation

- improved calculation of He-like line intensities (X-rays)
- new data for Be-like and F-like sequences (EUV)
- new data for Fe XVII to Fe XXIII (X-rays)
- new data for n=3 to n=3 N-like and O-like transitions (UV)
- some additional IDL programs:

7. Version 7 of the CHIANTI database, released in Sept 2011, includes an update for a range of ions, mostly important for the EUV and the UV.

8. CHIANTI v. 8.0, released in September 2015, includes several new entries and revises the atomic data for many ions. In particular, for ions of the He-like, Li-like, B-like, Ne-like and Na-like isoelectronic sequences, plus several important coronal iron and nickel ions. In addition, the method for computing the differential emission measure has been changed.

   Until v.8, the format of the CHIANTI files has remained the same (aside from an update in the collisional files, to allow nine-point splines). The changes that are introduced in version 8 are minor, in the sense that they only affect the format of the energy and the electron excitation files. The format of these files is different, but still fixed, i.e. the same for all the ions.

   The energy files have the same information, although the format is changed and some redundant information has been removed.

   For new ions, we have chosen not to fit the electron excitation data, but rather to provide the original rates, still in the Burgess and Tully (1992) scaled domain. For each transition, a scaled temperature is given.

   We have modified the IDL software that reads the new files. There is no backward compatibility with previous versions of CHIANTI.

   Finally, we have modified the DEM inversion program.

9. CHIANTI v. 9.0, released in March 2019, includes several new atomic data for the satellite lines at X-ray wavelengths, and a different way in which the satellite line intensities are calculated. In particular, we improve the modelling by explicitly including autoionization and dielectronic recombination processes in the calculation of level populations for several ions. Atomic data for a several ions have also been added/improved. The CHIANTI DEM program has been modified and a program to calculate SDO/AIA responses added.

Minor releases of the database and the software normally include fixes and might occur a few times per year.

1.5 How to keep updated on CHIANTI developments

- Read the CHIANTI NEWS page on the WWW.

- Read the HISTORY (software) and README (database) files in the distribution. Any news and changes are logged in these files. The first one has the details of all the software changes, while the second one describes the changes to the database. These files can be found directly in the distribution or via links on the WWW pages.
- We have a google group for comments and questions:

  `https://groups.google.com/forum/#!forum/chianti`

2 What is new in version 9

CHIANTI v. 9.0 includes several new atomic data for the satellite lines at X-ray wavelengths, and a different way in which the satellite line intensities are calculated.
The top-level IDL routines are largely unchanged, but several modifications have been applied in the low-level routines.
For ions that do not have autoionisation levels in CHIANTI, the same procedure is adopted for the line intensity calculations.
For the ions that have autoionisation levels in CHIANTI, a two-ion model is setup and solved. This new method has allowed the inclusion of level-resolved radiative-recombination rates, now included with a new CHIANTI format file, with extension '.rrlvl'
The '.reclvl' files, which have recombination rates that include cascading effects as calculated by Gu (2003) have been retained in some cases and used to apply a correction factor to the level population, as described in CHIANTI v.5.
Note that either a reclvl file or a rrlvl is available for an ion, but not both. More details are available below.
CHIANTI v. 9.0 also has a new routine to calculate SDO AIA responses and has a revised CHIANTI_DEM routine.

3 The database structure

The atomic data will continue to be updated regularly as new data are calculated or measured in the laboratory.
It is intended that these atomic data can be accessed and transfered into users own analysis programs, for more sophisticated applications.

3.1 Directory structure and atomic data file contents

The database has a tree structure, with the top directory designated with the IDL system variable `!xuvtop` (and named `dbase` within SolarSoft):

```
dbase/


In the top directory are the following files:

        README_CHIANTI   with the description of the current version.
        VERSION          with the version number.

Then, there is a series of subdirectories, one for each element present in the databa
Each element has  a subdirectory  for each ion.
The filename prefix for each ion follows spectroscopic notation.
For example, for He, we have He I and He II subdirectories:

        he/
           he_1/
           he_2/

Then, we have a series of ancillary data that are contained in
various subdirectories:

        masterlist/
                         has the list of the ions currently present
                         in the database
        abundance/       with elemental abundance files.
        continuum/       contains files for the continuum calculations.
        dem/             has DEM files.
        ioneq/           contains ionization fraction files.
        ip/              has ionization potentials.
        ancillary_data/instrument_responses/     with effective areas.
```

There are five primary ASCII files for each ion subdirectory. For example, for Fe XIV we have: energy levels: `fe_14.elvlc`; radiative data and wavelengths: `fe_14.wgfa`; electron excitation data: `fe_14.scups`; and optionally proton excitation data: `fe_14.psplups`).

3.1.1 Energies

`fe_14.elvlc` Specifies the energy levels in cm^{-1} and Rydbergs. It includes both experimental data and theoretical values of the levels energies. The energy levels are obtained from NIST. Where necessary, these are supplemented by other laboratory and theoretical values.
Each column in the files has:

1. Index of the fine structure energy levels. This index applies to all files for this ion. Levels can be arranged in any order, although they often follow the observed or theoretical energy ordering.

2. Designation of the configuration of the level. The usual configuration format is, e.g., 3s2.3p2(2P).3d. That is, orbitals are separated by a . and parent terms are placed in brackets. However, separating orbitals by white space is also present, e.g., 3s2 3p2(2P) 3d.

3. a label string, which can be used to attach a label to a level. An example is for Fe II for which the strings are used for multiplets in the same configuration which have the same LSJ labels.

Table 1: The format of the data in the energy files

Col.	Format	ID	Comment
1	i7	LVL	Level index
2	a30	CONF	Configuration description
3	a5	LABEL	Level label string
4	i5	2S+1	Spin multiplicity
5	a5	L	Orbital angular momentum
6	f5.1	J	Total angular momentum
7	f15.3	E_o	Observed energy (cm^{-1})
8	f15.3	E_b	'Best-guess' theoretical energy (cm^{-1})

4. Integer value of 2S+1, S=spin in standard usage

5. Symbol for angular momentum L, i.e. S for L=0, P for L=1, D for L=2, etc.

6. Total angular momentum J

7. observed energy in cm^{-1} (if there is no observed value, it is set to -1).

8. theoretical energy in cm^{-1} (usually this is the energy from the scattering calculation, but not necessarily). If an observed energy does not exist, then the CHIANTI software uses the theoretical energy for that level. The CHIANTI theoretical energies are not necessarily the same as those of the scattering calculations. In some cases, theoretical 'best guess' energies are provided. These are normally obtained by linear interpolation of the ab-initio level energies with the few experimental energies. The wavelength file has the corresponding wavelengths. Typical uncertainties of the ab-initio wavelengths are a up to several Å, while those obtained from the 'best guess' energies are estimated to be around 1 Å.

9. There might be further columns but these are not read by the software.

The energy file contains columns with a fixed number of characters, and the data entries are terminated by a line containing only a -1. All subsequent lines are considered to be comments.

The elvlc file is read by the routine read_elvlc.pro, which has been modified to take into account the new format of version 8. In order to maintain compatibility with the previous version of the routine, read_elvlc can be called in the identical manner to the old routine, i.e.,

```
IDL> read_elvlc, filename, l1,term, $
conf,ss,ll,jj,ecm,eryd,ecmth,erydth,ref
```

However, read_elvlc can also be called with:

```
IDL> read_elvlc, filename, elvlcstr=elvlstr
```

where elvlcstr is an IDL structure containing the data. Note that elvlcstr has the tag elvlcstr.data.energy which contains the best guess energy for a level.

3.1.2 Wavelengths, A-values and oscillator strengths

`fe_14.wgfa` Contains the wavelengths, gf and A values of the transitions and the indices initial and final level corresponding to the indices of the levels as given in the `fe_14.elvlc` file. Wavelengths calculated from the theoretical energies are of an indeterminate accuracy and their values are presented as negative values of the calculated wavelength. The ‘observed’ wavelengths in these files are based on the experimental energies and should be the best available.

The radiative data are taken from published literature and where necessary, supplemented by new calculations.

Each column in the files has:

1. index of the lower energy level, format=i5 (consistent with the ordering in the .elvlc file)

2. index of the upper energy level, format=i5

3. wavelength in Angstroms, format=f15. If the wavelength does not connect 2 observed energy levels, the wavelength is given as a negative number. Generally wavelengths are given to three decimal places, with exceptions for very short and very long wavelength transitions. Two-photon transitions are given a zero wavelength. Also, the corrections to the inner-shell levels above the ionization limit due to autoionization are included as radiationless transitions to the ground state.

4. gf value (weighted oscillator strength), format=e15

5. *A*-value, format=e15. Generally this will be the radiative decay rate (or A-value) in units of s-1. It can also be the autoionization rate or the two-photon decay rate, both in units of s-1.

6. In some cases additional columns have extra information on the transition. These are not read by the software.

7. comments at the end are free format.

The main routine for reading the wgfa file is read_wgfa2.pro, which is called as:

```
IDL> read_wgfa2, filename, lvl1, lvl2, wvl, gf, a_value, wgfaref
```

where the outputs lvl1, lvl2, wvl, gf and a_value are each 1D arrays containing the five data columns from the file. Note that for those transitions that have both a radiative decay rate and an autoionization rate, then the transition will be represented twice in the output arrays. In preparation for input to the level population solving routine, the rates are summed within the routine setup_ion.pro. The output wgfaref is a string array containing the comment string at the bottom of the data-file. There is also a routine that reads the wgfa file into a structure:

```
IDL> read_wgfa_str, filename, wgfastr, wgfaref
```

3.1.3 Electron collisional rates

`fe_14.scups`.

The SCUPS files replace the SPLUPS files in previous versions of the database. The format is very similar. For the new additions to version 8, this file contains the temperatures and effective collision strengths, scaled according to Burgess and Tully (1992). In addition, the collision strengths at scaled temperatures of 0 and 1 are given. The value at threshold is extrapolated, while the value at scaled temperature equal to 1 is either extrapolated or obtained from the high-energy limit. For the other ions that have not been modified in version 8, the previous spline fits to the scaled effective collision strengths are retained. Only the format of the files is changed.

For each transition, there are three lines in the file. The first line in the SCUPS files contains the information about the transition, the second the BT92-scaled temperatures, and the third the BT92-scaled effective collision strengths, as described in Table 3.1.3. Note that even in the cases when a single temperature array was present, with the new format each transition will have a different scaled temperature array.

Table 2: The format of the data in the SCUPS files

Col.	Format	ID	Comment
1	i7	L1	Lower level of transition (integer)
2	i7	L2	Upper level of transition (integer)
3	e12.3	DE	Energy of transition, Rydberg (float)
4	e12.3	GF	Oscillator strength (float)
5	e12.3	LIM	High-temperature limit value (float)
6	i5	NT	Number of scaled temperatures
7	i3	T_TYPE	BT92 Transition type (integer)
8	e12.3	C_VAL	BT92 scaling parameter (float)
	e12.3	SCT	Scaled temperatures (2nd line)
	e12.3	SCUPS	Scaled effective collision strengths (3rd line)

The comments in the file are at the end, bracketed by two lines containing only a -1.

The SCUPS file is read into an IDL structure as follows:

```
IDL> read_scups, splfile, splstr
```

where SPLSTR has two tags called INFO and DATA that are both structures. The tags for SPLSTR.INFO are listed in Table 6.2.

The tags for SPLSTR.DATA are listed in Table 3.1.3. The size of the TEMP and UPS arrays will be set to the maximum number of temperatures (NT_MAX) in the dataset. If e.g. NT_MAX=20 and a particular transition only has upsilons defined for 10 temperatures, then TEMP[0:9] and UPS[0:9] will contain these values, and TEMP[10:19] and UPS[10:19] will be set to the missing value, defined in UPSSTR.INFO.MISSING.

Table 3: The tags for SPLSTR.INFO

Tag	Data	Type
ION_NAME	Ion name (CHIANTI format)	String*1
ION_Z	Atomic number	Integer*1
ION_N	Spectroscopic number	Integer*1
ION_ROMAN	Ion name (Roman numerals)	String*1
ION_LATEX	Ion name (latex format)	String*1
ION_LATEX_ALT	Ion name (alternative latex format)	String*1
COMMENTS	File comments	String array
CHIANTI_VER	Version number	String*1
TIME_STAMP	Time file was made	String*1
FILENAME	Filename (including path)	String*1
MISSING	Value for missing data	Float*1
NTRANS	Number of transitions	Long*1

Table 4: The tags for SPLSTR.DATA

Tag	Data	Type
LVL1	Lower level index	Integer*1
LVL2	Upper level index	Integer*1
DE	Energy (Rydberg)	Float*1
GF	Oscillator strength	Float*1
LIM	High-temperature limit	Float*1
T_TYPE	BT92 Transition	Integer*1
C_UPS	BT92 Scaling parameter	Float*1
NSPL	Number of temperatures	Integer*1
STEMP	Scaled temperature values	Float*NT_MAX
SPL	Scaled effective collision strength values	Float*NT_MAX

3.1.4 Proton rates

`fe_14.psplups` contains the spline fits to the proton collision rate coefficients in units of cm^3 s^{-1}. The columns of the file are given in Table 3.1.4. "ID" refers to the tag of the IDL structure when the file is read with the routine `read_splups`.

Table 5: The format of the data in the PSPLUPS files.

Col.	Format	ID	Comment
1	i3	LVL1	Lower level of transition (integer)
2	i3	LVL2	Upper level of transition (integer)
3	i3	T_TYPE	Transition type (integer)
4	e10.3	GF	Oscillator strength (float)
5	e10.3	DE	Energy of transition, Rydberg (float)
6	e10.3	C_UPS	BT92 scaling parameter (float)
7	e10.3	SPL	Spline values for rate coefficients (float array)

3.1.5 Autoionization files

E.g. `fe_23.auto`. The autoionization file is new for CHIANTI version 9. If the file is present, it contains the autoionization rates with the indexing of the levels in the recombining and recombined ion.

The autoionization rate is the rate of decay of an atomic level through autoionization to a bound level. It is used for the calculation of satellite lines, i.e. to calculate the branching ratios and to calculate the dielectronic recombination rate by means of the principle of detailed-balance.

The files have three data columns and one free-format column:

1. level index of the recombining ion, format=i7. This contains the level index of the singly-ionized level in the the recombining ion corresponding to the CHIANTI .elvlc file. For simple ions such as the He-like, it is 1, the index of the ground state.

2. The level index of the energetically-higher autoionising level, format=i7. The level indices are defined in the CHIANTI .elvlc file of the recombined ion, which is in the same directory as the `.auto` file.

3. The autoionization rate in s^{-1}, format=i7.

4. This is free format. After the data columns, there can be a free-format string giving the transition information for the transition. This is used to aid reading the file by eye, and the transition information is not read by the software.

The end of the data entries is marked by a line containing only -1. Comments are then entered in a free format.

3.1.6 Other files

`fe_14.fblvl` contains data used for the free-bound calculation. Namely, the statistical weights for each nl shell, and the energy levels in cm^{-1}.

3.1.7 Corrections due to ionization/recombination

For some ions CHIANTI has level-resolved ionization and recombination rates and these are stored in .CILVL and .RECLVL files, respectively. These files have the same format, which is given in Table 3.1.7. For each transition, there are two lines of data. The first gives the log temperature at which the rates are tabulated, and the second gives the rate coefficient (cm^3 s^{-1}). *For a single ion, the temperatures must be the same in both the .CILVL and .RECLVL files.*

The level index for the starting level in these transitions is *always* 1. This means that the rates into and out of the excited levels are treated as if they go into the ground level of the ion under consideration.

The rates are read simultaneously with a single routine:

```
IDL> read_ionrec,ionname,rec_rate,ci_rate,temp_ionrec,luprec,lupci,status,rec_ref,ci_
```

where `rec_rate` and `ci_rate` are 2D arrays giving the rates as a function of temperature for each transition. `temp_ionrec` is the temperature array, `luprec` and `lupci` are arrays giving the final level for each transition.

Table 6: The format of the data in the CILVL and RECLVL files.

Col.	Format	Comment
1	i3	Atomic number of element (integer)
2	i3	Spectroscopic number of ion (integer)
3	i3	Starting level of transition, which is always 1 (integer)
4	i3	Upper level of transition (integer)
5	e10.3	Either log T (row 1) or rate coefficent (row 2) (float)

The recombination rates normally include cascading effects and are used to obtain a correction to the level populations as decribed in CHAINTI v.5. Note that several approximations are present as follows.

The basic assumption is that the ion that receives population from ionization and recombination does not have any metastable levels. Only the ground level has significant population. In this scenario, the population of any excited level, i, is given by

$$n_i = \frac{\alpha_{\mathrm{in},i}}{\alpha_{\mathrm{out},i}} \tag{1}$$

where α is the total rate for all atomic processes that send population into or out of the level. To add a new atomic process, we can simply modify α to include these processes.

For recombination and ionization, only the "in" rates are affected and they become

$$\alpha^*_{\mathrm{in},i} = \alpha_{\mathrm{in},i} + N_{\mathrm{e}} R_i / f_0 \tag{2}$$

where f_0 is the fractional population of the ion of interest and R_i is the combined ionization and recombination rate into level i and is given by

$$R_j = f_- Q_j(T) + f_+ A_j(T) \tag{3}$$

where Q is the ionization rate, A is the recombination rate, and f_- and f_+ are the fractional populations of the ionizing and recombining ions, respectively.
The "corrected" level population is then

$$n^*_i = \xi n_i \tag{4}$$

where

$$\xi = 1 + \frac{N_{\mathrm{e}} R_i / f_0}{\alpha_{\mathrm{out},i}} \tag{5}$$

The CHIANTI routine that performs this correction is `correct_pops.pro`, which is called from `pop_solver.pro`.
For the ions for which this method is applied to (mostly the H-like sequence), the assumption that there are no metastable levels is correct up to densities of about 10^{13} cm^{-3}. This limit is rarely reached in hot coronal plasmas.
For **reclvl** files (recombination), extrapolation is performed by the IDL routine `ci_rec_interp`. The rates are set to zero *below* the temperature range. *Above* the temperature range a linear extrapolation of the two highest temperature points is performed. However, if the extrapolated point has a higher value than the rate at the highest temperature point, then it is set to this latter value (radiative recombination rates should decrease with temperature, so this should not happen).
For **cilvl** files (ionization), extrapolation is performed by the IDL routine `ci_rec_interp`. The rates are set to zero *above* the temperature range. *Below* the temperature range a linear extrapolation of the two lowest temperature points is performed.
(Note that extrapolations are applied to the log values of the temperatures and rates.)

3.1.8 New (v.9) level-resolved radiative recombination rates

For the ions that have autoionisation levels in CHIANTI, a two-ion model is setup and solved. This new method has allowed the inclusion of level-resolved radiative-recombination rates, now included with a new CHIANTI format file, with extension '.rrlvl'
These are rates for the process of an electron being captured to produce a bound state of the recombined ion. This serves as a population process for the bound state. The atomic data have been produced by N.R.Badnell, see [1].
Unlike the earlier treatment of recombination (through the `reclvl` files), the rates are directly incorporated into the level balance equations by using the two-ion model. The only issue is that these rates do not include all the possible cascading effect from high levels that are not included in the CHIANTI model.

The effects of the recombination not included in this way are added in the two-ion model by a ground-to-ground recombination rate. This rate is defined as the total radiative recombination rate (obtained with the routine `recomb_rate` with the `/radiative` keyword set) *minus* the sum of the individual level-resolved rates (this sum needs to omit the "1–1" transition from the `rrlvl` file).
Note that the total radiative recombination rates are actually consistent as they were obtained from the same data sets.
For **rrlvl** files (recombination), extrapolation is performed within the IDL routine `ch_load_2ion_rates`.
For temperatures below the data range, the rate is set to be the rate at the lowest temperature point of the range. For temperatures above the range, the rate is set to the rate at the highest temperature point of the range. [Author note: this over-estimates the rate. It is better to use a linear extrapolation.]
The `rrlvl` file has a free format.
For each transition there are two lines of data. The first gives the temperatures at which the rates are tabulated, and the second line gives the rates. Line 1 format:

1. atomic number: The atomic number of the ion. Not read by the software.
2. spectroscopic number. The spectroscopic number of the ion. For example, 22 corresponds to XXII. Not read by the software.
3. index of the initial state. This is the level index of the recombining ion and thus should match the correct level index of the recombining ion model.
4. index of the final state. The index of the final level of the transition in the recombined ion.
5. Columns 5 onwards temperatures The temperatures at which the rates are tabulated. Note that, although each transition has its own temperature array, in practice the software assumes all transitions have the same temperature array

The format of the second line is the same, except for Columns 5 onwards which has the direct radiative recombination rate coefficients in units of cm^3 s^{-1}, defined for the temperatures given on line 1. The comments section begins with a -1 on a single line. The comments are in free format. The comments section is then closed with a -1 on a single line.
The rrlvl files are read with `read_rrlvl.pro`. As an example, the calling sequence for O VI is:

```
IDL> zion2filename,8,6,fname
IDL> rrdata = read_rrlvl( fname, status )
```

The output rrdata is a structure with various tags, described in the header of the routine. The optional output status is an integer with value 1 if the file was successfully found and read and 0 otherwise. Although a temperature array is provided for each transition, the output temperature array is only 1D. Therefore there is an implicit assumption that **the temperature array is the same for all transitions. The temperature array returned by read_rrlvl actually corresponds to the first transition of the reclvl file.**

3.1.9 Final comments

The basic structure of the files is to put the data at the beginning of the file followed by comments.
The comments must be enclosed at the end of the file between two lines containing a single '-1'
The original sources are documented in each data file, where also additional and detailed comments written by the CHIANTI member that assessed that particular ion can be found. You can have direct access to the references via the WWW pages.

3.2 Additional ancillary data

Some additional data files are needed in various calculations. The software allows the selection of these files, from either a 'standard' selection provided within the database, or by using user defined files that are included in the current working directory, provided they have the proper file extension. For example, it is possible to create a user defined 'myfile.dem'. If the file is in the working directory, then the file will automatically be appended to the list of available DEMs from the CHIANTI database. In other cases, it is possible to select the file by using a widget that allows the user to change directory.
Any user-defined file must have the same format as those already provided (also including a 'comment' section at the end of the file)

The list of the ions present in the database

A `!xuvtop/masterlist/masterlist.ions` file keeps the current list of all the ions in the database. This list is used as default by many routines (for example those that calculate line intensities).
In some cases, it is possible to instead use a user-defined list of ions, to speed the calculation, or to directly supply the routines with a list of ions, via the SNGL_ION keyword.

Elemental abundances

Files with various elemental abundances are provided in the directory `!xuvtop/abund/`
Element abundances are in the usual dex notation (Log_{10} values, relative to H, that has a Log_{10} value of 12).
Options are available within the routines to choose different elemental abundances. User-defined abundance files can also be used, and should have a `.abund` file extension.
Be aware that any element missing in the elemental abundance file will also be missing in any output created by any software that reads the elemental abundance file.
There is a great deal of controversy over the variation of the elemental abundances in the solar and stellar atmospheres. See the reviews of Meyer (1985, 1993), Widing and Feldman (1992), Mason (1992), Raymond et al. (2001). Also, it should be kept in mind that different analyses can lead to very different results. For example, the ionisation balance, the selection of lines, and the spectroscopic method used can each account for a variation of a factor of

two or more in the derived element abundances (see Del Zanna et al., 2002 and references therein).

Ionisation Fractions

Files giving collisional ionization equilibria are provided in the `!xuvtop/ioneq` directory. User defined ionisation files should have a `.ioneq` file extension. The ionisation fractions have been taken from the tabulated values in the published literature (e.g. Arnaud & Raymond, 1992; Arnaud & Rothenflug, 1985; Mazzotta et al., 1998).
We highly recommend that users use the latest, default CHIANTI ionization equilibrium tables in chianti.ioneq.
Be aware that the chianti.ioneq file has ion fractions calculated over a temperature grid of log T=0.05, different than the previous ones.
Most CHIANTI software uses the temperatures in these files as a base for the calculations. For example, if DEM(T) values are supplied, they are first interpolated at the temperatures in the ionization fraction, and the calculations are done at those temperatures.
Be aware that any ion missing in a ionisation fraction file will also be missing in any output created by the software.
Any line missing a temperature overlap with the chosen ionisation fraction would have zero emissivity and will not be output by the software.
Be aware that large differences between different tabulations are present, and that large uncertainties are associated with these calculations. It should be noted that the ionisation equilibrium plays a major role not only in the derivation of the DEM, but also in that one of the elemental abundances. In this respect, it is important to be aware of the fact that a number of ions, in particular those of the Li and Na isoelectronic sequence, present anomalous behaviour (see Del Zanna et al., 2002, and references therein).

Differential Emission Measure

Files specifying various standard differential emission measures (DEM) distributions for different solar features are provided in the *!xuvtop/dem* directory. Additional files for stellar atmospheres will also soon be added. Each file contains the Log_{10} T and Log_{10} DEM values in two columns, ordered with increasing temperature.
User-defined DEM files should have a `.dem` file extension and must have the same format and ordering of the files provided.
Be aware that any line missing a temperature overlap between the ion fraction and the chosen DEM distribution would have zero emissivity and will not be output by the software.
The emission measure distribution in the solar atmosphere is a complex issue. Starting with the pioneering work by Pottasch (1964), spectra in the UV wavelength range have been used to determine the distribution of material as a function of temperature, following various methods. More details can be found in Section 9.

Other files

Other files are in other miscellaneous directories. For example:
`!xuvtop+'/ip/chianti.ip'` has the ionization potentials for all the ions;
`!xuvtop+'/continuum/'` contains data used by the routines that calculate the continuum. For example, gffgu.dat contains the free-free gaunt factors of Sutherland (1998).

4 The Software structure

A number of Interactive Data Language (IDL) procedures are also provided as part of the CHIANTI package. These include routines to read the various CHIANTI database files, calculate level populations, line intensities, and temperature dependent and density dependent line intensity ratios.
Most of our efforts have gone into developing well-documented user-friendly IDL routines that meet readily apparent needs. We welcome contributions to the software.
CHIANTI has been run mainly on Sun, Dec Unix workstations and on PCs with Linux. CHIANTI also runs (with some small limitations) under Windows NT and in VMS. Please report to us any problems you might find.
All the IDL routines have been documented with extensive headers giving detailed descriptions and examples. Please read them carefully.

The CHIANTI routines can be grouped into three classes:

- Low-level routines, that for example read the files in the database, or perform the level population calculations. These are not described here **and should not be used directly by the users**.
- High-level routines, that perform more complex operations and can be called from the command line. These routines usually output arrays or structures, and optionally produce plots, postscript output or ascii files. Most of them have a long list of options, commanded via KEYWORDS. Please read the headers.
- Higher-level widget-type routines. These routines are more user-friendly, and are complementary of the above class. These routines call low-level or high-level routines to perform the calculations.

The CHIANTI routines are organised in a tree structure. The main level contains some high-level procedures and the **HISTORY file, where all modifications to the software are logged**.

4.1 Short description of the CHIANTI software

Now, a description of some high-level routines that are present within the CHIANTI software tree is given.

- `ch_synthetic.pro`, calculates (without any abundance factor) line intensities or $G(T)$, and outputs a **line intensities CHIANTI structure** (see Sect.10 for details). This IDL structure can be **saved and later restored** in various ways, for example using `ch_write_fits` and `ch_read_fits`.

- `ch_line_list.pro` takes the **line intensities CHIANTI structure** and creates ascii or latex files with lists of line identifications and intensities.

- `make_chianti_spec.pro` This program creates the **CHIANTI SPECTRUM structure** (read Sect. 11 for details), that contains the synthetic spectrum, created by multiplying by the abundance factor the line intensities and adding the continuum (optional).

- Finally, a multi-purpose widget `ch_ss.pro` has been written. It includes all the above features.

Other routines that previously were only available within SolarSoft are also included now. The users therefore now have various different routines to choose from.
We have kept the older high-level routines, so the user can still use them as before (with slight modifications/additions of keywords). We have updated them and re-written as wrapper routines (essentially that call the newly-written routines)

4.2 How to find help

For the first two classes of routines, by simply typing the name of the routine, a description of how to call the routines, with examples, is printed. For example,

```
 IDL > temperature_ratios
 IDL > temperature_ratios,ion,wmin,wmax,Log10(tempmin),Log10(tempmax),$
 IDL > temperature,ratio,description,$
 IDL >  [pressure= ,density= , psfile= , outfile= ]
 IDL >
 IDL > i.e.:
 IDL > temperature_ratios,'c_5',40.,50.,5.,7.,temp,rat,desc
```

In any case the best way to understand what a routine does and how it works is to **read the header documentation** with e.g.:

```
 IDL > xdoc,'ch_synthetic'
 IDL > doc_library,'ch_synthetic'
```

Another way to quickly see the keywords of a routine is to use:

```
 IDL > chkarg,'temperature_ratios
 .....
---> Call: pro temperature_ratios,ions,wmin,wmax,tempmin,tempmax,$
temperature,ratio,description,$
density=density,psfile=psfile, $
outfile=outfile,noprot=noprot, $
radtemp=radtemp,rphot=rphot,photons=photons, $
ioneq_file=ioneq_file, abund_file=abund_file, $
VERBOSE=VERBOSE
```

Table 7: List of main-level routines - part A

Synthetic spectra	
`ch_ss`	Multi-purpose widget to calculate line intensities, create synthetic spectra adding the continuum, tables and various outputs.
`synthetic`	Calculates a synthetic spectrum. Outputs arrays.
`synthetic_plot`	Plots the spectrum created by synthetic and interactively identify lines
`isothermal`	Calculates a synthetic spectrum with an isothermal approximation. Outputs arrays.
`make_chianti_spec`	Creates a synthetic spectrum. Works with structures.
Line intensities	
`ascii_wvl_dem`	Creates an ascii file with a list of line identifications and intensities.
`latex_wvl_dem`	Creates a latex file with a list of line identifications and intensities.
`ch_synthetic`	Multi-purpose routine that calculates line intensities (without any abundance factor), and outputs an IDL structure.
`ch_line_list`	Multi-purpose routine that creates ascii and latex files with lists of line identifications and intensities. Takes as input the structure created by CH_SYNTHETIC.
Line emissivities	
`emiss_calc`	To compute the emissivities of all lines of a specified ion over given ranges of temperature and density.
`gofnt`	Calculates the contribution functions G(T)
`g_of_t`	To compute the G(T) of selected lines.
Density-sensitive line ratios	
`dens_plotter`	A widget routine to allow the analysis of density sensitive ratios.
`density_ratios`	Plots the variation of line intensities with electron density
`chianti_ne`	A widget to calculate and plot density sensitive line ratios.
`plot_chianti_ne`	Plots density sensitive ratios saved from CHIANTI_NE
Temperature-sensitive line ratios	
`temp_plotter`	A widget routine to allow the analysis of temperature sensitive ratios.
`temperature_ratios`	Plots the variation of line intensities with electron temperature .
`chianti_te`	A widget to calculate and plot temperature sensitive line ratios.
`plot_chianti_te`	Plots temperature sensitive ratios saved from CHIANTI_TE
Continuum	
`freefree`	calculates the free-free (bremsstrahlung) continuum.
`freebound`	calculates the free-bound continuum.
`two_photon`	calculates the two-photon continuum.

Table 8: List of main-level routines - part B

Level populations and details	
`plot_populations`	plots the level populations
`pop_plot`	To plot n_jA_ji/N_e values as a function of N_e.
`show_pops`	To display populations of significant levels in a CHIANTI ion model.
`pop_processes`	Outputs to the screen the contributions of the different physical processes to the population of the specified level within an ion.
`level_lifetime`	Calculates the lifetime of a level using the radiative decay rates stored in the CHIANTI .wgfa file.
`metastable_levels`	This routine returns a list of levels in an ion that are metastable.
`plot_config_energies`	This routine graphically displays the configuration energies for an ion.
Excitation rates	
`rate_coeff`	This routine allows you to extract electron rate coefficients, both for individual transitions and the complete array of transitions. The rate coefficient is returned in units cm^3 s-1.
`spl2ups`	Converts the CHIANTI spline fits to upsilons (thermally-averaged collision strengths) for a specified ion, transition and temperature
Ionization and recombination rates	
`ioniz_cross`	Returns the cross section for electron collisional ionization
`ioniz_rate`	returns the ionization rate coefficien
`recomb_rate`	returns the recombination rate coefficient
`make_ioneq_all`	Calculates ion fractions in ioniztion equilibrium
Miscellaneous	
`rad_loss`	Calculates the radiative losses
`max_temp`	Calculates temperature at max ionisation ratio for an ion.
`plot_ioneq`	Plots the ionisation equilibrium values for an element.
`chianti_dem`	Calculates the Differential Emission Measure DEM(T) using the CHIANTI database, from a given set of observed lines.
`plot_dem`	To plot differential emission measure (DEM) values
`integral_calc`	To compute the atomic data integral for use in column or volume emission measure work.
`ch_read_fits`	Read standard CHIANTI FITS binary table data containing the output from CH_SYNTHETIC and outputs a CHIANTI line intensities structure.
`ch_write_fits`	Write standard FITS binary table data from CHIANTI line intensities structure.

```
                DENS_PLOTTER
                TEMP_PLOTTER
                ------------
                     /\
                     ||
                     ||          CHIANTI_NE    DENSITY_RATIOS
     G_OF_T     RATIO_PLOTTER   CHIANTI_TE  TEMPERATURE_RATIOS
     ------------------------------------------------------
                                    /\
                                    ||
                                -----------
                                EMISS_CALC
                                -----------
                                    /\
                                    ||
                                ----------    | PLOT_POPULATIONS
                                POP_SOLVER -->| POP_PLOT
                                ----------    | SHOW_POPS
=================================================================

                                ----------
                                POP_SOLVER
                                ----------
                                     |
                                     V
                            ------------------------------
                                CH_SYNTHETIC
                            ------------------------------
                                 |      |          |     |
                                 |      |          |     V
 FREEFREE FREEBOUND TWO_PHOTON   |      V          |   GOFNT
 -----------------------------   |  CH_LINE_LIST   |   -----
                |                | --------------  |
                |                |            |    V
                V                |            |   -------------
        -----------------        |            |-> ASCII_WVL_DEM
        MAKE_CHIANTI_SPEC        |            |-> LATEX_WVL_DEM
        -----------------        |            |   -------------
            |       |            |            |
            |       V            |            |
            |  --------------    |            |
            |    SYNTHETIC   <---|            |
            |    ISOTHERMAL  <---|            |
            |  --------------    |            |
            |                    |            |
            V                    V            V
          -------------------------------------
                       CH_SS
          -------------------------------------
```

Figure 1: Schematic flow chart showing the main links between the high-level routines. POP_SOLVER is the core low-level routine that is called by most CHIANTI routines. Note that many CHIANTI routines are now wrapper routines. For example, now DENSITY_RATIOS calls EMISS_CALC that in turn calls POP_SOLVER. SYNTHETIC and ISOTHERMAL are now wrapper routines that call CH_SYNTHETIC and MAKE_CHIANTI_SPEC.

5 The CHIANTI distribution and installation

CHIANTI is currently distributed in two ways:

1. as an independent package within SolarSoft, a programming and data analysis environment for the solar physics community. See

   `http://www.lmsal.com/solarsoft/`

   for details on how to download and install the package. The database and the software are organised in a self-contained package

   ```
   $SSW/packages/chianti/
   ```

   with the following tree structure:

   ```
   dbase/    (database)
   doc/      (documentation, in particular the USER GUIDE)
   idl/      (IDL software)
   setup/    (supplementary setup files)
   ```

   The Solarsoft website explains how to set up a "mirror" so that the contents of the package are automatically kept up-to-date, and this is highly recommended. Note, however, that updates to the CHIANTI software are fairly infrequent between major releases, and usually only for bug-fixes.

   All modifications to the software are logged in the `$SSW/packages/chianti/idl/HISTORY` file.

   Modifications to the database are much less frequent. They are described in the `$SSW/packages/chianti/dbase/README_CHIANTI` file.

   We send an e-mail to the CHIANTI user group every time we make a minor release of the database available.

   Note that the contents of the SolarSoft package change on a frequent timescale normally to fix bugs caused by the use of new IDL releases.

   We recommend that you use CHIANTI within the SolarSoft framework and that you setup in your site a mirror in order to have automatic upgrades. It is easy to follow the instructions to download and setup the package.

2. On the WWW, as tar files, via the CHIANTI download page:

   `http://www.chiantidatabase.org/chianti_download.html`

   Currently, the data and the software are distributed in two separate tar files. The tar files have a similar tree structure as the SolarSoft distribution.

The IDL procedures file includes `idl/gen/`, a copy of the `$SSW/gen/` routines. This is because some routines of the `$SSW/gen/` directory are needed to run some of the CHIANTI programs.

CHIANTI is a package, in the sense that database and progams are to be used together.

The current version of the database must be used with the current version of the programs. Backward compatibility does not always apply.

5.1 Installing CHIANTI

For the CHIANTI software to work correctly, the following need to be set:

- Access to the CHIANTI IDL routines. The IDL !PATH should contain the paths to the directories where the CHIANTI IDL procedures are.
- Access to the CHIANTI atomic database and ancillary data. This is done by defining the system variable !xuvtop, that should point to the CHIANTI atomic database top directory.
- The following IDL system variables need to be defined:
  - `!xuvtop` the top directory for the atomic database
  - `!ioneq_file` the default ionization equilibrium file
  - `!abund_file` the default elemental abundance file
  - `!BCOLOR` , `!ASPECT`

5.1.1 Installing CHIANTI within SolarSoft

If you are using SolarSoft you should have the setup already organised so as to have the path of the CHIANTI IDL procedures added to IDL_PATH, the `!xuvtop` and the other IDL system variables defined. This is done automatically by using

```
 unix> setssw chianti
 unix> sswidl
or
 unix> sswidl
 IDL > ssw_packages,/chianti
```

After this, you will be able to run the CHIANTI routines.

5.1.2 Installing CHIANTI independently as a stand-alone

Users of previous versions, please NOTICE:
The procedures to install CHIANTI have changed in small but important ways.

Download the CHIANTI files

Download the CHIANTI data tar file (e.g. CHIANTI_7.0_data.tar.gz) and the CHIANTI IDL procedures tar file (e.g. CHIANTI_7.0_pro.tar.gz) and put the tar files into a directory (for example, /data1/chianti/dbase for the data and /data1/chianti/ for the software) and then do the following:

```
unix> gunzip [file_name].tar.gz
unix> tar xvf [file_name].tar
```

This will copy all the CHIANTI data files into /data1/chianti/dbase and create the /data1/chianti/idl and /data1/chianti/doc/ directories.

Define the IDL paths and the system variables

There are two ways of doing the above. The first is to define the system variables within IDL, the second is outside IDL. We suggest the first option. Once IDL is started, there are three steps:

```
unix > idl
```

1. add to the IDL PATH the path of where the CHIANTI IDL routines are:

   ```
   Unix: IDL> !PATH = '+/data1/chianti/idl:'+!PATH
   Windows: IDL> !PATH = '+C:\data1\chianti\idl;'+!PATH
   VMS: IDL> !PATH = '+/data1/chianti/idl,'+!PATH
   ```

2. `IDL> !PATH = EXPAND_PATH(!PATH)`

   The '+' and the EXPAND_PATH are needed since the IDL routines are organised into subdirectories. The second option involves writing (UNIX) the following statement in your /.cshrc (or /.login) file:

   ```
   setenv IDL_PATH /usr/local/rsi/idl_4/lib:+/data1/chianti/idl
   ```

   (assuming you have the main IDL directory in /usr/local/rsi/idl_4).

3.
   ```
   Unix:  IDL> use_chianti, '/data1/chianti/dbase'
   Windows: IDL> use_chianti, 'C:\data1\chianti\dbase'
   ```

After following the above steps, it will be possible to run the CHIANTI routines from any directory. `use_chianti` also allows you to set your default abundance and ionization equilibria files with the abund and ioneq keywords.
Previous CHIANTI users should check the note below.
We suggest that you add the three above calls to your IDL_STARTUP file (say /.idl_startup). If this file does not exist then it should be created. In UNIX, this can be done if you add the following line to your `.login` file:

```
 setenv IDL_STARTUP ~/.idl_startup
```

(Note that the changes to the .login file mean that you should do a `source ~/.login` before running IDL).
Alternatively, you can write the three statements above in a file, say `start_chianti.pro`:

```
 !PATH = '+/data1/chianti/idl:'+!PATH
 !PATH = EXPAND_PATH(!PATH)
 use_chianti, '/data1/chianti/dbase'
 END
```

and run

```
 IDL> .r start_chianti
```

Note to previous CHIANTI users:

If you had already defined the CHIANTI system variables before entering IDL or in your IDL STARTUP file you should remove those definitions.
Alternatively, instead of using `use_chianti, '/data1/chianti/dbase'`, you have to make sure you have in your IDL STARTUP file something like this:

```
 !PATH = '+/data1/chianti/idl:'+!PATH
 !PATH = EXPAND_PATH(!PATH)
defsysv,'!xuvtop', '/data1/chianti/dbase'
defsysv,'!ioneq_file','/data1/chianti/dbase/ioneq/mazzotta_etal.ioneq'
defsysv,'!abund_file','/data1/chianti/dbase/abundance/cosmic.abund'
defsysv,'!BCOLOR',0
defsysv,'!ASPECT',1.0
```

6 Theory and definitions - implementation within the CHIANTI IDL software

6.1 Optically thin emission lines

For a review on *Spectroscopic Diagnostics in the EUV for Solar and Stellar Plasmas* see e.g. [2].
The intensity $I(\lambda_{ij})$, of an optically thin spectral line of wavelength λ_{ij} (frequency $\nu_{ij} = \frac{c}{h\lambda_{ij}}$) is

$$I(\lambda_{ij}) = \frac{h\nu_{ij}}{4\pi} \int N_j \; A_{ji} \; dh \quad [\text{ergs cm}^{-2} \text{ s}^{-1} \text{ sr}^{-1}] \tag{6}$$

where i, j are the lower and upper levels, A_{ji} is the spontaneous transition probability, N_j is the number density of the upper level j of the emitting ion and h is the line of sight through the emitting plasma. In low density plasmas the collisional excitation processes are generally faster than ionization and recombination timescales, therefore the collisional excitation is dominant over ionization and recombination in populating the excited states. This allows the low-lying level populations to be treated separately from the ionization and recombination processes.
For allowed transitions we have $N_j(X^{+m})A_{ji} \sim N_e$. The population of the level j can be expressed as:

$$N_j(X^{+m}) = \frac{N_j(X^{+m})}{N(X^{+m})} \, \frac{N(X^{+m})}{N(X)} \, \frac{N(X)}{N(H)} \frac{N(H)}{N_e} \, N_e \tag{7}$$

- $N(X^{+m})/N(X)$ is the ionization ratio of the ion X^{+m} relative to the total number density of element X (contained in the files in the `ioneq/` directory);
- $Ab(X) = N(X)/N(H)$ is the elemental abundance relative to Hydrogen (contained in the files in the `abundance/` directory);
- $N(H)/N_e$ is the Hydrogen density relative to the free electron density. Often assumed to be equal to 0.83, as hydrogen and helium are usually completely ionised for hot optically thin plasmas.

  See the routine PROTON_DENS described in Sect. 6.3.1 for details on how to calculate $N(H)/N_e$.
- The fraction $n_{\rm i} = \frac{N_j(X^{+m})}{N(X^{+m})}$ of ions $\rm X^{+m}$ lying in the state j is determined within CHIANTI by solving the statistical equilibrium equations for a number of low-lying levels of the ion including all the important collisional and radiative excitation and de-excitation mechanisms.

In the ‘standard model’ for interpreting line intensities there are three fundamental assumptions that serve to simplify the problem considerably:

1. the plasma is in a steady state;

2. atomic processes affecting the ionisation state of an element can be separated from those affecting the level balance within an ion;

3. all lines are optically thin.

The atomic data contained in the CHIANTI database are particularly suited to the analysis of emission lines via this model, and the following discussion outlines this approach. No attempt is made to discuss non-equilibrium conditions.

With the first of the assumptions, the population of ions lying in a given state is constant and so the number of ions leaving this state per unit time must exactly balance the number arriving into that state. If we denote the number of transitions leaving the state i to a state j taking place per unit time per unit volume by $\alpha_{\rm ij}$, then steady state implies

$$N_{\rm i}\sum_{\rm j\neq i}\alpha_{\rm ij} = \sum_{\rm j\neq i} N_{\rm j}\alpha_{\rm ji}. \tag{8}$$

Setting

$$\alpha_{\rm ii} = -\sum_{\rm j\neq i}\alpha_{\rm ij} \tag{9}$$

we have

$$\sum_{\rm j} N_{\rm j}\alpha_{\rm ji} = 0 \tag{10}$$

for each state i and, as the coefficients $\alpha_{\rm ji}$ are independent of the state populations, we have a set of linear equations to solve for the $N_{\rm i}$.

Now our second assumption means that the processes that affect the ionisation state of the plasma do not affect the quantity $n_{\rm i}$. Eq. 10 thus becomes

$$\sum_{\rm j} n_{\rm j}\alpha_{\rm ji} = 0 \tag{11}$$

where the $\alpha_{\rm ji}$ only include those processes that affect the level balance of the ion.

For the basic CHIANTI model these processes are simply electron and proton excitation and de-excitation, and the generalised radiative decay:

$$\alpha_{ij} = N_{\rm e}C^e_{ij} + N_{\rm p}C^p_{ij} + \mathcal{A}_{ij} \tag{12}$$

where C^e_{ij} is the electron excitation–de-excitation rate, C^p_{ij} is the proton excitation–de-excitation rate, $N_{\rm p}$ is the proton density, $\mathcal{A}_{ij}$ is the generalized radiative decay rate, that includes $A_{\rm ij}$, the radiative decay rate which is zero for i<j (the ‘A-values’ are contained in the CHIANTI `.wgfa` files), and the photoexcitation and stimulated emission.

C^e_{ij} is given by:

$$C^e_{\rm ij} = q_{\rm ij} \quad {\rm i} < {\rm j} \tag{13}$$

$$C^e_{\rm ij} = \frac{\omega_{\rm j}}{\omega_{\rm i}}\exp\left(\frac{\Delta E}{kT}\right) q_{\rm ji} \quad {\rm i} > {\rm j} \tag{14}$$

where ω_i is the statistical weight of level i, k is Boltzmann's constant, T the electron temperature, and q_ij the electron excitation rate coefficient which is given by:

$$q_\mathrm{ij} = 2.172 \times 10^{-8} \left(\frac{I_\infty}{kT}\right)^{1/2} \exp\left(-\frac{\Delta E}{kT}\right) \frac{\Upsilon_\mathrm{ij}}{\omega_\mathrm{i}} \qquad [\mathrm{cm}^3\ \mathrm{s}^{-1}] \tag{15}$$

where I_∞ is the Rydberg energy (13.61 eV), and Υ_ij is the thermally-averaged collision strength for the i → j excitation. The Υ_ij are derived from the scaled data in the CHIANTI `.splups` files.
The solution of Eq. 11 is performed by the CHIANTI routine `pop_solver.pro`, which gives the population of the levels within an ion, normalised.
The level populations for a given ion can be calculated and displayed with `plot_populations.pro` (but also see `pop_plot.pro`).

6.2 Electron excitation rates

The electron excitation rates are obtained with the use of the read_scups routine, which reads data from `.scups` files into an IDL structure (the previous version read data into several individual arrays). The call is

```
IDL> read_scups, splfile, splstr
```

where SPLSTR has two tags called INFO and DATA that are both structures. The tags for SPLSTR.INFO are listed in Table 6.2.

Table 9: The tags for SPLSTR.INFO

Tag	Data	Type
ION_NAME	Ion name (CHIANTI format)	String*1
ION_Z	Atomic number	Integer*1
ION_N	Spectroscopic number	Integer*1
ION_ROMAN	Ion name (Roman numerals)	String*1
ION_LATEX	Ion name (latex format)	String*1
ION_LATEX_ALT	Ion name (alternative latex format)	String*1
COMMENTS	File comments	String array
CHIANTI_VER	Version number	String*1
TIME_STAMP	Time file was made	String*1
FILENAME	Filename (including path)	String*1
MISSING	Value for missing data	Float*1
NTRANS	Number of transitions	Long*1

The actual effective collision strengths are obtained by calling

```
descale_all, temp, splstr, index, ups
```

where the temperature array `temp` is specified. The rates are normally used within the level population solver `pop_solver.pro`

```
IDL> cc=rate_coeff('fe_13',1e6)
```

calculates the complete rate coefficient array for Fe XIII. The rate coefficient is returned in units cm^3 s-1. There are 749 levels in the CHIANTI model for Fe XIII and so cc is an array of size 749 x 749. To calculate the rate coefficient for a particular transition, do:

```
IDL> cc=rate_coeff('fe_13',1e6,trans=[1,20])
 Wavelength:             202.04
 Exc. rate coeff:     6.986e-09
 De-exc. rate coeff:  4.746e-09
```

Only the excitation rate coefficient is returned in cc. To return the de-excitation coefficient, set trans=[20,1].

6.3 Proton excitation rates

For each ion for which proton rates are available, an additional file is required in the database to contain the fits to the rate coefficients. The file has the suffix `.psplups`, and is analogous to the `.splups` file for the electron fits. However, the proton file contains fits to the proton *rate coefficients*, whereas the electron file contains fits to the effective collision strengths (upsilons). All of the proton transitions included in CHIANTI are forbidden transitions taking place between levels within the same configuration. Many of the transitions required 9-point splines in order to provide adequate fits. One example:

```
  2  3  2 0.000e+00 1.027e-02 8.000e+02-2.150e-13 1.052e-13 4.397e-12 \
2.232e-11 5.389e-11 8.708e-11 1.014e-10 7.658e-11-2.805e-12
```

(The \ indicates a break in the line.) Note that the Z and spectroscopic number are not given for each ion, in contrast to the electron files. The rest of the line is the same as that for the electron files, except in this case there are 9 spline values as a 9-point spline was fitted to the data. The data can be read by the routine `read_splups.pro` which loads the data into an IDL structure. The call is

```
      IDL> read_splups, filename, splstr, splref , /prot
```

The structure `splstr` has the following tags

```
      .lvl1   lower level index
      .lvl2   upper level index
      .t_type transition type
      .gf     gf value
      .de     Delta-E for transition (rydbergs)
      .c_ups  the scaling parameter
      .nspl   number of spline points
      .spl    Vector of length 9, containing spline points
```

The proton rates can be obtained from the spline fits by using

```
      descale_all, temp, splstr, index, ups
```

where `temp` can be an array of temperatures.
By default, all routines will include proton rates in the calculation of the ion level balance. A keyword `/NOPROT` can be used to switch off the proton rates.

6.3.1 The proton-to-electron ratio

To include the proton rates in CHIANTI it is necessary to know the proton number density N_p. This quantity is usually expressed in terms of the ratio relative to the electron density. For a standard solar plasma this is a constant for temperatures beyond log T = 4.6 with a value around 0.85. Thus one option is to simply fix the ratio as a constant.
As we want CHIANTI to be applicable for low temperature plasmas, however, we have decided to explicitly calculate the ratio making use of the ion balance and abundance files, that uniquely determine N_p. The relevant routine is PROTON_DENS.PRO, which is called as, e.g.,

```
IDL> print, proton_dens(6.0)
      0.84860524
```

where 6.0 is the log T value, and the output is $N_\mathrm{p}/N_\mathrm{e}$. The routine can also be used to return the ratio of neutral hydrogen+protons to electrons:

```
IDL> print, proton_dens(6.0,/hydrogen)
      0.84860545
```

which is only very slightly different since hydrogen is almost fully ionized in the corona.
N_p is calculated from the ion balance and element abundance files contained in CHIANTI through the following expression

$$R(T) = \frac{N_\mathrm{p}}{N_\mathrm{e}} = \frac{\mathrm{Ab(H)}F(\mathrm{H}^+, T)}{\sum_{i=1}^{n}\sum_{Z=1}^{i} ZF(\mathrm{A}_i^{+Z}, T)\mathrm{Ab}(\mathrm{A}_i)} \tag{16}$$

where Ab is the element abundance, A_i is the ith element (i.e., A_1=H, A_2=He, etc.), Z is the charge on the ion, $F(\mathrm{A}_i^{+Z}, T)$ is the fraction of ions of element A_i in the form A_i^{+Z} at temperature T.
The ion fractions contained in CHIANTI are tabulated over the range 4.0 $\leq$ log T $\leq$ 8.0. Above and below these values, we set $R(T)$ to the values for $\log T = 8.0$ and $\log T = 4.0$, respectively.
The use of this routine has some side effects. Some routines for which the ratio may have some effects in the results don't require you to select the ion balance or abundance files. E.g., DENSITY_RATIOS.PRO does not require the user to select these files, however, at low temperatures one may see significant changes take place in line ratios on account of the change in the proton-to-electron ratio. We deal with this effect by using the default `!abund\_file` and `!ioneq\_file` files to derive the proton-to-electron ratio, but allowing the files to be directly specified by the user through keywords if he/she needs to do this.

6.4 How to obtain all the main rates for an ion at once

```
IDL> input=ch_setup_ion('o_6',rphot=rphot,radtemp=radtemp,noprot=noprot, $
                 ioneq_file=ioneq_file,abund_file=abund_file, $
                 noionrec=noionrec)
```

The call loads the atomic data for the ion O VI into the structure `input`. All the keywords are optional inputs.

The default call to `ch_setup_ion` results in all of the ion's data files being loaded into the output structure. The core data-sets of energy levels, *A*-values, and electron collision strengths are always loaded. The secondary data-sets of proton rates and level-resolved ionization and recombination rates are loaded as long as the data files exist for the ions. They can be switched off using the /NOPROT and /NOIONREC keywords, respectively.

Photon excitation (and stimulated emission) do not have atomic data files since the rates depend on the *A*-values. They are switched on by the user by specifying RPHOT, the distance from the emitting source center in source radius units. The blackbody radiation temperature is set with the input RADTEMP.

For proton rates, `pop_solver` needs to know the proton density. This is computed self-consistently from the electron temperature (assumed to be the same as the proton temperature), and ion fraction file and the element abundance file. The optional inputs IONEQ_FILE and ABUND_FILE are used to specify these files. If not set, then the default files !IONEQ_FILE and !ABUND_FILE are used.

6.5 The level population solver

The solution of Eq. 11 is performed by the CHIANTI routine `pop_solver.pro`, which gives the population of the levels within an ion, normalised.

A simple call to the routine could be:

```
IDL> input=ch_setup_ion('o_6')
IDL> pop_solver, input, 5e5, 1e9, pop
```

The first call loads the atomic data for the ion O VI into the structure `input`. The call to `pop_solver` then uses the atomic data to compute the populations (`pop`) at the specified temperature (5×10^5 K) and electron number density (10^9 cm^{-3}).

The output `pop` is simply an array containing the populations for all of the ion's levels.

As of CHIANTI 9, new routines were added that take the atomic data from `input` and reformat them into rate matrices that form the matrix solved by `pop_solver`. They are `ch_load_ion_rates` and `ch_load_2ion_rates`. These routines take the atomic data that was read by `ch_setup_ion` and put them in matrices to be used by `pop_solver`. A similar single-ion matrix building was carried out by the earlier version of `pop_solver`. Now `pop_solver` calls out to the load rate routines.

For most ions, the procedure is simply:

```
IDL> t=[1e5,2e5,3e5]
IDL> rates=ch_load_ion_rates('o_4',t)
```

where “t” is the temperature array for which the rates are computed. The result is a structure with the following tags:

```
N_LEVELS        LONG                 204
AA              DOUBLE    Array[204, 204]
AAX             DOUBLE    Array[204, 204]
PPR             DOUBLE    Array[3, 204, 204]
QQ              DOUBLE    Array[3, 204, 204]
TEMP            FLOAT     Array[3]
ION_DATA        STRUCT    -> <Anonymous> Array[1]
MULT            FLOAT     Array[204]
SUM_MWL_COEFFS  DOUBLE    Array[3]
SUMTST          INT              0
```

The rate matrices are: radiative decay rates (AA), photoexcitation and stimulated emission rates (AAX), proton rate coefficients (PPR), and electron rate coefficients (QQ). The ION_DATA structure is simply the atomic data structure returned by `ch_setup_ion`. If `pop_solver` sees that the tag `ion_data.autostr` exists, then it means that the ion has autoionization data (i.e., the `.auto` file exists). This requires a special two-ion atomic model whereby the CHIANTI model for the next ionization stage is added to the ion model. An example of how this is done for O VI is as follows:

```
IDL> rates1=ch_load_ion_rates('o_6',t)
IDL> rates2=ch_load_ion_rates('o_7',t)
IDL> rates=ch_load_2ion_rates(rates1,rates2)
```

The structure `rates` is then used for creating the atomic rates matrix used by `pop_solver`. Note that this structure has an expanded set of tags:

```
N_LEVELS        LONG                 972
AA              DOUBLE    Array[972, 972]
QQ              DOUBLE    Array[3, 972, 972]
AAX             DOUBLE    Array[972, 972]
PPR             DOUBLE    Array[3, 972, 972]
IONIZ           DOUBLE    Array[3, 972, 972]
RR              DOUBLE    Array[3, 972, 972]
AI              DOUBLE    Array[972, 972]
DC              DOUBLE    Array[3, 972, 972]
DR              DOUBLE    Array[3, 972, 972]
```

In particular: level-resolved ionization rate coefficients (IONIZ), level-resolved radiative recombination rate coefficients (RR), autoionization rates (AI), dielectronic capture rate coefficients (DC), and level-resolved dielectronic recombination rate coefficients (DR). As of CHIANTI 9, the IONIZ and DR matrices only contain data for transitions between the two ground states.

If you compare the `rates1` and `rates` structures then you will see they have 923 and 972 levels, respectively (as of CHIANTI 9). This is because the 49 levels of the O VII CHIANTI model have been added to the O VI model. The level populations are calculated by pop_solver for all 972 levels, but the array is then truncated to 923 levels so that only the O VI populations are returned. If you would like to see the populations for all 972 levels, then use the /all_levels keyword input to pop_solver.

6.6 The two-ion model developed for version 9

The main processes that form the doubly excited (autoionizing) states s are inner-shell excitation of one electron in the *lower* ionization stage Z^{+r}, and the dielectronic capture (DC) of a free electron by the *higher* (ionization stage) ion Z^{r+1} in the state k. The lower ion Z^{+r} in the *autoionizing* state s can then autoionize (releasing a free electron) to any of the states k of the ion Z^{r+1}, or produce a radiative transition into any bound state f of the recombined ion. The intensity of the satellite line, resulting from the decay to a final bound level f of the lower ion Z^{+r}, is

$$I_{sf} = N_s A_{sf} \tag{17}$$

where A_{sf} is the radiative decay rate from level s to the final level f, and N_s is the population of the autoionizing state.

In previous versions of CHIANTI, and for the ions without autoionizing states, the level populations p are obtained by solving the rate equations:

$$(A + N_{\rm e}\,(C^e + C^p) + P)\,p = b\ , \tag{18}$$

where b is a vector set to zeros except for the first element which is 1. The most important matrices are those for the spontaneous decay processes (A), and for the collisional excitation/de-excitation due to electron impact (C^e). Additional matrices for photo-excitation (P) and proton excitation (C^p) and their de-excitation processes are also included.

The population of the autoionizing state due to dielectronic capture involves the solution of rate equations where both the recombined and recombining ions are included. Thus, in CHIANTI V.9 we have modified the IDL codes in order to solve for the level populations of all the levels of the lower (recombined) and higher (recombining) ion simultaneously. In order to do this, we have extended the matrices A, C^e, C^p and P to include all the levels of the lower ion (including the autoionizing levels) and the bound levels of the higher ion. This means that the rates connecting the bound levels within each of the ions are included in the same way as in the previous versions, but we now have included the rates for the autoionizing levels and those connecting the two ions. This framework naturally takes into account density effects on the satellite lines related to the populations of the metastable levels in the recombining ions.

The population of the autoionizing state due to inner-shell excitation is calculated in a similar way as for the bound levels, including the inner-shell impact excitation rate $N_{\rm e}\,C^e_{is}$ in the rate equation. The rate coefficient C^e_{is} is retrieved from the scaled effective collision strengths stored in the *.scups* file. The de-excitation rate due to a collision by a free electron is included as in the case of the bound levels, but this term is usually negligible.

The decay rate due to autoionization, $A^{\rm auto}_{sk}$, of the doubly-excited state s to the state k is included in the same way as the matrix A. The autoionization rates are stored in the *.auto* file. Note that in the case of the satellites of He-like ions, all autoionizations go to the ground state, where most of the population is. For the recombining ions with metastable levels, dielectronic capture from populated levels can occur and thus it is included, together with autoionization rates to those levels.

Dielectronic capture is effectively a population process for the autoionizing states s, proportional to the free electron density N_e and the population N_k of the recombining ion in its state k involved in the capture. We have therefore included this populating process between states k and s as in the matrix C^e of collisional excitation. The rate coefficient C^{dc}_{ks} for the capture of the free electron by the ion Z^{r+1} in the state k into a doubly-excited state s of the lower ion Z^{+r} is obtained from the autoionization rate applying the principle of detailed balance:

$$C^{\mathrm{dc}}_{ks} = \frac{h^3}{(2\pi m k T)^{3/2}} \frac{g_s}{2g_k} A^{\mathrm{auto}}_{sk} \exp\left(-\frac{E_s - E_k}{kT}\right) \tag{19}$$

This is valid as long as the electrons have a Maxwellian distribution. To relate the populations of the lower and higher ions we also need to include ionization and recombination rates. We recall that currently CHIANTI includes collisional ionization (CI) rates between the ground states of the ions calculated by [3], and total radiative recombination (RR) and dielectronic recombination (DR) rates, also between the ground states of the ions. The recombination rates are mainly those calculated by N.R.Badnell and colleagues. The total DR rates from the ground state of the recombining ion were obtained by Badnell by summing up the contributions of all the autoionizing states. The total DR rate needs a correction, to avoid double counting the DR rates. As we have now included autoionizing levels in the model, for consistency we need to calculate the total DR due to the levels included in the model and originating from the ground level of the higher ion, and subtract this quantity from the total DR rate. The remaining rate is added in the matrix as a term connecting the two ground states.

The calculation of the total DR rate due to the autoionizing levels included in the model is not trivial, as in principle each autoionizing state could decay to another autoionizing level, as well as decay to a bound level f' of the recombined ion, or autoionize to a level of the recombining ion. We neglect the first process for two reasons. First, we note that the other two are the main ones, although we note that in some cases we do have some autoionizing levels that mainly decay to other autoionizing levels. The second reason is that the total DR rates have been calculated neglecting this cascading process. The total rate is therefore calculated as

$$c \sum_s \sum_k \frac{g_s}{2g_k} \, A^{\mathrm{auto}}_{sk} \, e^{-\frac{E_s - E_k}{kT}} \, \frac{\sum_{f'<s} A_{sf'}}{\sum_k A^{auto}_{sk} + \sum_{f'<s} A_{sf'}} \tag{20}$$

where the constant $c = h^3 \, (2\pi m k T)^{-3/2}$

The CI rates are included as they are available in CHIANTI, i.e. connecting the ground states. The RR rates are now included in two different ways. For most ions, the total RR rates from the ground state of the recombining ion are included in the matrix to connect to the ground state of the recombined ion.

For some ions, we have now introduced the level-resolved RR rates as calculated by N. Badnell. They are included with a new file, with the extension `.rrlvl`, with a format similar to that of the previous `.reclvl` files. We have included these level-resolved rates into the matrix. They typically increase the populations of the lower levels by 10% or so, but for higher levels can be the only populating process. To avoid double counting, as in the case of the DR rates, we sum the total RR of the level-resolved rates and subtract this value from the total RR. Any residual total RR is added as a rate connecting the two ground states.

Contribution to level population due to cascades from bound levels is therefore now naturally included in the model ions, although for many ions most of the RR occurs into high-lying levels that are not currently included in the model. For many ions we have therefore retained the corrections due to these cascading effects as in the previous CHIANTI versions. Prior to CHIANTI 9 the radiative recombination rates were implemented as a post-processing step once the level population equations of the standard CHIANTI model had been solved, as described in [6]. These RR rates are stored in the `.reclvl` files. For the important Fe XVII–XXIII ions, the rates are from the calculation of [4] and the rate into a level includes both the direct radiative recombination rate and the indirect recombinations that come from cascading from higher levels, *including* the autoionization levels populated by dielectronic capture. The effects of these rates on the level populations is approximated with a correction, after the matrix is inverted, as in previous CHIANTI versions.

Note that a model CHIANTI ion can only have either direct level-resolved RR rates as in the `.rrlvl` files or the level-resolved RR including cascades as in the `.reclvl` files. Also note that the `.reclvl` files can only have recombination from the ground state, while the `.rrlvl` files can in principle include recombination from excited states. Finally, note that the CHIANTI programs assume that all the rates in these two files are on the same temperature grid, although the files in principle could have different temperature grids for each transitions.

After the matrices are populated and the populations obtained, we then normalise the populations of the lower ion so the total is one, as in the case of ions without autoionizing states. Note that the relative population of the two ions as obtained solving the two-ion rate equations can sometimes be different than what is obtained by assuming that the ion populations are all in the ground state (which is what is used in CHIANTI to calculate the relative ion charge state distributions). However, this effect is small and is not considered in this version 9.

Finally, we note that the new v.9 IDL codes are compatible with earlier CHIANTI v.8 data files, but the earlier CHIANTI v.8 IDL programs should not be used with the new v.9 format files.

6.7 Contribution functions

We rewrite the intensity as:

$$I(\lambda_{ij}) = \int Ab(X) C(T, \lambda_{ij}, N_e) N_e N_H dh \tag{21}$$

where the function

$$C(T, \lambda_{ij}, N_e) = \frac{h\nu_{ij}}{4\pi} \frac{A_{ji}}{N_e} \frac{N_j(X^{+m})}{N(X^{+m})} \frac{N(X^{+m})}{N(X)} \quad [\text{ergs cm}^{+3}\ \text{s}^{-1}], \tag{22}$$

called the *contribution function*, contains all of the relevant atomic physics parameters and is strongly peaked in temperature.

`gofnt.pro` calculates these contribution functions (see also `g_of_t.pro` for a slightly different way of calculating contribution functions).
Please note that in the literature there are various definitions of *contribution functions.* Aside from having values in in either photons or ergs, sometime the factor $\frac{1}{4\pi}$ is not included. Sometimes a value of 0.83 for $N(H)/N_e$ is assumed and included. Sometimes the element abundance factor is also included. Any of the above (or any other) variations also affect the definition of a line intensity in terms of the contribution function and the DEM. In the following we will refer to the functions $C(T, \lambda_{ij}, N_e)$ and $G(T, \lambda_{ij}, Ab(X), N_e) = Ab(X)\ C(T, \lambda_{ij}, N_e)$ (i.e. the contribution function that contains the abundance factor).
If we define, assuming that is a single-value function of the temperature, the differential emission measure $DEM(T)$ function as

$$DEM(T) = \ N_e N_H \frac{dh}{dT} \ \ [\mathrm{cm}^{-5}\mathrm{K}^{-1}] \tag{23}$$

the intensity can be rewritten, assuming that the abundance is constant along the line of sight:

$$I(\lambda_{ij}) = Ab(X) \int_T \ C(T, \lambda_{ij}, N_e)\ DEM(T)\ dT \quad [\mathrm{ergs\ cm}^{-2}\ \mathrm{s}^{-1}\ \mathrm{sr}^{-1}] \tag{24}$$

The DEM gives an indication of the amount of plasma along the line of sight that is emitting the radiation observed and has a temperature between T and $T + dT$.
The IDL routine `chianti_dem.pro` described in Sect. 9 calculates the Differential Emission Measure DEM(T) using the CHIANTI database, from a given set of observed lines.
Routines such as `ch_synthetic.pro` (see Sect. 7.1) calculate line intensities **without the abundance factor**, that is only included at a later stage.
In the isothermal approximation, all plasma is assumed to be at a single temperature (T_o) and the intensity becomes:

$$I(\lambda_{ij}) = C(T_o, \lambda_{ij}, N_e) Ab(X) EM_h \tag{25}$$

where we have defined the **column emission measure**

$$EM_h = \int N_e N_H dh \quad [\mathrm{cm}^{-5}] \tag{26}$$

`ch_synthetic.pro` in the isothermal approximation calculates $I = C(T_o, \lambda_{ij}, N_e) \int N_e N_H dh$, while `isothermal.pro` and `ch_ss.pro` (see examples in Sect. 7.1) can be used to create synthetic spectra (with the abundance factor).
It is also possible to calculate intensities and spectra with a multi-temperature model, by providing an array of T_o, EM_h values.
Please note that in the literature many different definitions of Differential Emission Measures, Emission Measures and approximations can be found (see Del Zanna et al., 2002 for some clarifications).

6.8 The stellar case - irradiances and volume DEM

In the stellar case, or in the case of irradiance observations of the Sun as a star, the theoretical flux at Earth of an optically thin spectral line is:

$$F(\lambda_{ij}) = \frac{1}{d^2} \int_V Ab(X)\; C(N_\mathrm{e}, T, \lambda_{ij})\; N_\mathrm{e} N_\mathrm{H}\; dV \quad [\mathrm{ergs\ cm^{-2}\ s^{-1}}] \tag{27}$$

where $C(N_\mathrm{e}, T, \lambda_{ij})$ has the same expression as above, d is the star's distance, dV is the volume element, and V is the entire source volume. A **volume** Differential Emission Measures DEM is often defined:

$$DEM(T) = \ N_\mathrm{e} N_\mathrm{H} \frac{dV}{dT} \quad [\mathrm{cm^{-3} K^{-1}}] \tag{28}$$

together with a corresponding **volume** emission measure EM_V:

$$EM_V = \int N_e N_H dV \quad [\mathrm{cm^{-3}}] \tag{29}$$

At the moment CHIANTI does not include volume emission measures. In the near future we will modify the software and the definition of the DEM in order to include volume emission measures.

However, any **volume** Differential Emission Measures can be rescaled to **column** $DEMs$ and used within the software to produce synthetic spectra for stellar coronae. One way of doing this is to assume spherical symmetry, and that the emitting region is a layer dh distributed over the entire star's disk, i.e. $dV = 4\pi R_*^2 dh$ (R_* is the star's radius). If the star's radius and distance are known, a **volume** DEM can be scaled with the factor $\frac{4\pi R_*^2}{d^2}$ to obtain a **column** DEM.

If this is used, the outputs will have flux units, i.e. $ergscm^{-2}s^{-1}$ (or $photonscm^{-2}s^{-1}$) and **not** $ergscm^{-2}s^{-1}sr^{-1}$.

An example of scaled DEM is provided in the file AU_Mic.dem, in the CHIANTI distribution.

Column $DEMs$ and EMs are assumed when the spectra are folded with effective areas (see Sect. 7.1). The effective areas are assumed to have units of $countsphotons^{-1}cm^{+2}$, so the output units of the spectra will be $countss^{-1}pixel^{-1}$.

Also note that corrections to **interstellar absorption** are not presently included in CHIANTI.

6.9 Definitions for emission measure analysis

Going back to Eq. 7, we write

$$N_\mathrm{i} = 0.83\, F(T)\, Ab(\mathrm{X})\, N_\mathrm{e}\, n_\mathrm{i}, \tag{30}$$

where $F(T)$ is the ionisation fraction (independent of N_e in current ion balance calculations), Ab(X) the abundance of the element relative to hydrogen, and the ratio of hydrogen to free electrons has been taken as 0.83, as hydrogen and helium are completely ionised for temperatures $T > 10^4$ K.

The *emissivity* of the emission line resulting from a j-to-i radiative decay is defined as

$$\epsilon_{\rm ij} = \Delta E \, N_{\rm j} \, A_{\rm ji} \tag{31}$$

and has units of erg cm^{-3} s^{-1}. Often the alternative notation ϵ_λ will be used where λ is the wavelength of the emitted radiation in Angstroms (Å), and $\lambda = 1.986 \times 10^{-8}/\Delta E$ for ΔE in ergs. We will also define the *ion emissivity* as

$$\varepsilon_{\rm ij} = \Delta E \, n_{\rm j} \, A_{\rm ji}. \tag{32}$$

In order to relate the emissivity to the actual observed intensity of a line, we make use of the third assumption, which tells us that the intensity is proportional to the emissivity of the plasma, and so

$$P_\lambda = \int \epsilon_\lambda \, {\rm d}V, \tag{33}$$

where P_λ is the power in an observed line (units: erg s^{-1}), and dV is a volume of plasma with temperature T and density $N_{\rm e}$.
Expanding ϵ_λ using Eqs 30 and 31 gives

$$P_\lambda = 0.83 \, \Delta E \, Ab({\rm X}) \int F(T) \, n_{\rm j} A_{\rm ji} N_{\rm e} \, {\rm d}V. \tag{34}$$

An important feature of emission measure studies is to isolate those lines for which $n_{\rm j} A_{\rm ji} \sim N_{\rm e}$. By analysing only such lines, we are essentially separating the determination of the emission measure from the determination of the plasma density. If the lines all had different density dependencies, then it would be necessary to determine the density variation with temperature before finding the emission measure. If the $n_{\rm j} A_{\rm ji} \sim N_{\rm e}$ relation is assumed then we write

$$P_\lambda = \Delta E \, Ab({\rm X}) \int G_\lambda(T) \, N_{\rm e}^2 \, {\rm d}V \tag{35}$$

where

$$G_\lambda(T) = 0.83 \, F(T) \, \frac{n_{\rm j} A_{\rm ji}}{N_{\rm e}} \tag{36}$$

which is the so-called 'G-of-T' function.
On account of the ionisation fraction $F(T)$ this function is sharply peaked, and a common approximation (e.g., [7], [5]) is to assume that $G(T)$ has a constant value over a narrow temperature interval around $G(T_{\rm max})$, where $T_{\rm max}$ is the temperature of maximum ionisation for the ion. Here we will use the temperature of maximum emission or $T_{\rm mem}$ which is the temperature at which G_λ has its maximum. Defining

$$G_{\lambda,0}(T) = \begin{cases} C_\lambda & |\log T - \log T_{\rm mem}| < 0.15 \\ 0 & |\log T - \log T_{\rm mem}| > 0.15 \end{cases} \tag{37}$$

we require that

$$\int G_\lambda(T) dT = \int G_{\lambda,0}(T) dT \tag{38}$$

so

$$C_\lambda = \frac{\int G_\lambda(T) dT}{T_{\rm mem}(10^{0.15} - 10^{-0.15})}. \tag{39}$$

Our expression for P_λ thus becomes

$$P_\lambda = \Delta E \, Ab(\mathrm{X}) \, C_\lambda \, EM(V) \tag{40}$$

where

$$EM(V) = \sum_{\mathrm{i}} \left(\int_{V_\mathrm{i}} N_\mathrm{e}^2 \mathrm{d}V \right) \tag{41}$$

is the *volume emission measure.* Each volume V_i contains plasma with temperatures such that $|\log T - \log T_\mathrm{mem}| < 0.15$, and the sum over i is required in case there are distinct regions along the line of sight that satisfy this condition on T.

Now, solar emission lines are often measured as intensity (or radiance), I, with units typically of erg $\mathrm{cm}^{-2}\,\mathrm{sr}^{-1}\,\mathrm{s}^{-1}$. This quantity is related to P_λ by

$$P_\lambda = 4\pi \int I \mathrm{d}A \tag{42}$$

where dA is the projected area of the emitting element. One thus relates the observed intensity to an emission measure by

$$4\pi I = \Delta E \, Ab(\mathrm{X}) \, C_\lambda \, EM(s) \tag{43}$$

where $EM(s)$ is the *column emission measure*, where s is the line-of-sight depth of the emitting region.

Stellar emission lines are measured in flux (or irradiance), E, with units typically of erg $\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. E is related to P_λ by

$$P_\lambda = 4\pi d^2 E \tag{44}$$

where d is the distance to the object. The observed flux is then related to the emission measure by

$$E = \frac{1}{4\pi d^2} \Delta E \, Ab(\mathrm{X}) \, C_\lambda \, EM(V). \tag{45}$$

If one treats the emitting region as a uniform, spherical shell of thickness h then dV = $4\pi R^2$dh (R the distance from the star centre of the shell; typically $R = R_*$, the radius of the star) and so the expression for E becomes

$$E = \frac{1}{2} \frac{R_*^2}{d^2} \Delta E \, Ab(\mathrm{X}) \, C_\lambda \, EM(h). \tag{46}$$

where $EM(h)$ is the *emission measure over height.* The factor 1/2 denotes that half the photons from the shell are emitted towards the stellar surface and so are destroyed. Jordan and co-workers (see, e.g., [5]) utilise this definition and an assumption of spherical symmetry to deduce energy balance relations in solar and stellar atmospheres.

6.10 Non-Maxwellian particle distributions

Within CHIANTI the assumption of Maxwellian electron and proton distributions is implicit through the storage of Maxwellian-averaged electron and proton collision strengths in the .SPLUPS and .PSPLUPS data files. To model emission from plasmas with general, non-Maxwellian particle distributions would require integrations of the original collision strengths with the new particle distributions, and this is outside of the scope of the CHIANTI database. However, if the particle distributions can be expressed as a linear combination of Maxwellians of different temperatures, i.e.,

$$f(E;a_i) = \sum_i a_i f_M(E,T_i) \tag{47}$$

where the Maxwellian function $f_M(E,T_i)$ is given by

$$f_M(E,T_i) = 2\left(\frac{E}{\pi}\right)^{1/2}\left(\frac{1}{kT}\right)^{3/2}\exp\left(-\frac{E}{kT}\right) \tag{48}$$

then such distributions can be modelled in a straightforward manner within the CHIANTI framework.

The generalized electron excitation rate coefficient for the transition j to k and for the particle distribution f of electron velocities is given by

$$\begin{aligned}
\mathcal{C}_{jk} &= \int_{E_{jk}}^{\infty} Q_{jk} v f(E;a_i)\mathrm{d}E && (49)\\
&= \sum_i a_i \int_{E_{jk}}^{\infty} Q_{jk} v f_{\mathrm{M}}(E,T_i)\mathrm{d}E && (50)\\
&= \sum_i a_i C_{jk}(T_i) && (51)
\end{aligned}$$

where E_{jk} is the threshold energy for the transition, Q_{jk} is the collision cross section, E ($= m_{\mathrm{e}}v^2/2$, m_{e} the electron mass) is the energy of the incoming electron, and $C_{jk}(T_i)$ is the electron excitation rate coefficient for a Maxwellian particle distribution of temperature T_i (see, e.g., Burgess & Tully 1992).

The matrix $\mathcal{C}_{jk}$ replaces the usual Maxwellian-derived rate coefficient (C_{jk}) in the level balance equations solved by the CHIANTI software. The software routines for calculating emissivities and level populations have been modified to allow input of the non-Maxwellian parameters a_i through the keyword SUM_MWL_COEFFS. The temperatures T_i are specified through the standard temperature input to the routines. The temperatures are assumed to apply to both proton and electron distributions.

This prescription for treating non-Maxwellian distributions is not compatible with the treatment of ionization and recombination since an equilibrium ionization balance described by a single temperature is required for these processes. In such cases the ionization and recombination processes described in Sect. 3.1.7 are switched off when calculating the level populations if the a_i coefficients are specified in CHIANTI.

See Sect. 7.8 for an example.

6.11 Photoexcitation and Stimulated Emission

Within CHIANTI, we presently model the Photoexcitation and Stimulated Emission by assuming a a blackbody radiation field of temperature T_*. The generalized photon rate coefficient in this case is:

$$\mathcal{A}_{ij} = \begin{cases} W(R) A_{ji} \frac{\omega_j}{\omega_i} \frac{1}{\exp(\Delta E/kT_*)-1} & i < j \\ \\ A_{ji} \left[1 + W(R) \frac{1}{\exp(\Delta E/kT_*)-1} \right] & i > j \end{cases} \tag{52}$$

where A_{ji} is the radiative decay rate and $W(R)$ is the radiation dilution factor which accounts for the weakening of the radiation field at distances R from the source center.

We also assume an uniform (no limb brightening/darkening) spherical source with radius R_*:

$$W = \frac{1}{2} \left[1 - \left(1 - \frac{1}{r^2} \right)^{1/2} \right] \tag{53}$$

where

$$r = \frac{R}{R_*} \tag{54}$$

It is important to remember the assumptions in our formalism for radiation processes. For a given ion, only very specific wavelengths in the radiation continuum will affect the ion's level balance. If there are significant deviations from a blackbody spectrum at any of these wavelengths (perhaps due to a deep absorption line) then CHIANTI does not model the ion entirely correctly.

Examples of specific uses of the extra radiation processes include modeling of coronal emission lines above the surface of the Sun and other cool stars when the coronal electron density falls to low enough values that electron collisions lose their potency.

For the Sun, photoexcitation is very important for the infrared coronal lines. Photoexcitation is also important for modelling nebular ions that are irradiated by a hot star, such as in planetary nebulae, symbiotic stars and Wolf-Rayet stars.

6.11.1 Implementation of Photoexcitation and Stimulated Emission

No additions or modifications to CHIANTI data files are required for photoexcitation and stimulated emission as their rates are entirely determined from the radiative decay rates, level separation energies, and statistical weights – information already contained in CHIANTI. It is only necessary to specify the radiation field temperature and the dilution factor. These are specified as inputs to the IDL procedures through the new keywords RPHOT and RADTEMP. RPHOT specifies $r = \frac{R}{R_*}$, while RADTEMP gives the blackbody radiation temperature in K.

By default, photoexcitation and stimulated emission are not included in the level balance equations unless the keywords are set.

6.12 Photoexcitation by arbitrary radiation fields

Version 4 of CHIANTI introduced the possibility of including photoexcitation and stimulated emission through an external blackbody radiation field into the level balance equations. With version 5 the software has been modified to allow an arbitrary, user-defined radiation field to be specified.
The user must create an IDL routine that calculates the energy density per unit wavelength, U_λ, as a function of wavelength. The photoexcitation rate for a transition $i \to j$ is related to U_λ by the expression

$$P_{ij} = A_{ji} W(R) \frac{\omega_j}{\omega_i} \frac{\lambda^5}{8\pi hc} U_\lambda \tag{55}$$

where $W(R)$ is the dilution factor defined as in Young et al. (2003), A_{ji} is the Einstein coefficient for spontaneous radiation from j to i, ω_j and ω_i are the statistical weights of levels j and i. For example, U_λ for a blackbody of temperature, T, is given by

$$U_\lambda^{\rm bb} = \frac{8\pi hc}{\lambda^5} \frac{1}{\exp(hc/\lambda kT) - 1} \tag{56}$$

thus giving the photoexcitation rate for a blackbody of

$$P_{ij}^{\rm bb} = A_{ji} W(R) \frac{\omega_j}{\omega_i} \frac{1}{\exp(E/kT) - 1} \tag{57}$$

For reference we note that the energy density is related to the specific intensity, I_λ, by

$$I_\lambda = \frac{c}{4\pi} U_\lambda. \tag{58}$$

The user-defined radiation field function is implemented through a keyword RADFUNC='user_function, a, b' in the CHIANTI IDL routines SHOW_POPS and EMISS_CALC. The optional coefficients a and b can be used to modify the radiation field, e.g., by specifying a relative velocity between the radiation field and incident ion.
See Sect. 7.7 for an example.

6.13 Ionization and recombination

In Version 5 of CHIANTI, we have included ionization and recombination into level populations. The CHIANTI model for ionization and recombination assumes that the plasma can be described under the Coronal Model Approximation, where the total population of the excited levels of an ion is negligible compared to the population of the ground level. In this case, recombination and ionization processes can be included in a relatively straightforward way, since they can be treated as a correction to the case where populations are calculated neglecting them.
To illustrate this method, we will consider the simplified atomic model of an ion X^{+q} with abundance n_q composed of the ground level and one excited level only. In case ionization and recombination contributions to level populations are negligible, the relative population of the upper level is obtained by solving the equation:

$$N_g N_e C_{g\to i} = N_i A_{i\to g} \qquad \Longrightarrow \qquad \left(\frac{N_i}{N_g}\right)_{ion/rec} = \frac{N_e C_{g\to i}}{A_{i\to g}} \tag{59}$$

where $C_{g\to i}$ is the collisional excitation rate and $A_{i\to g}$ is the Einstein coefficient for spontaneous radiative decay. Collisional de-excitation is neglected in the coronal model approximation. In case ionization and recombination provide significant contribution, Equation 59 needs to be modified to include the rate coefficients for ionization (α_{ion}) and recombination (α_{rec}):

$$N_g N_e (n_q C_{g\to i} + n_{q-1}\alpha_{ion} + n_{q+1}\alpha_{rec}) = N_i A_{i\to g} n_q \tag{60}$$

where n_{q-1}, n_q, n_{q+1} are the ion fractions for the ions X^{q-1}, X^q and X^{q+1}, respectively. The population of the excited level can then be expressed as

$$\left(\frac{N_i}{N_g}\right)_{ion/rec} = \left(\frac{N_i}{N_g}\right)_{no\ ion/rec} \times \aleph \tag{61}$$

where the correction $\aleph$ is given by

$$\aleph = 1 + \frac{n_{q-1}\alpha_{ion} + n_{q+1}\alpha_{rec}}{n_q C_{g\to i}} \tag{62}$$

The correction $\aleph$ is temperature sensitive and can be large when the collisional excitation rate is small or when the abundance of the ion q is much smaller than the abundances of the adjacent ions. The correction due to ionization and recombination can have significant effects on intensities of observed X-ray lines.

The only limitation of this approach lies in the breakdown of the coronal model approximation at high densities for a few ions. This occurs at densities above which metastable level populations begin to be non-negligible, compared to the ground state (cf. Landi et al. 2005). The CHIANTI software has been modified to allow calculation of the correction factor $\aleph$ for the ions for which α_{ion} and α_{rec} are provided.

The inclusion of ionization and recombination effects in level population has required some more changes. New files have been created (.CILVL and .RECLVL) to store the ionization and recombination rates necessary for this process (see Sect. 3.1.7 for details).

A new routine (READ_IONREC.PRO) has been created to read these files and store their data in the input to the routine POP_SOLVER.PRO. This latter routine has been modified to include the correction to the level populations. In case the .CILVL and .RECLVL files are not available, a flag is set in the programs and these processes are ignored. The impact of this new process on the running time is negligible.

However, the introduction of ionization and recombination effects on level population has had a side effect. In previous versions of CHIANTI, the contribution to the intensity of spectral lines from levels below the ionization potential due to cascades from levels above the ionization potential was taken into account in the "dielectronic" .WGFA files, which included radiative transitions from the former, populated by cascades from the latter. For the ions for which the complete .RECLVL and .CILVL files are now available (Fe XVII to Fe XXIV), cascades from levels above ionization are now taken into account directly, so that the cascade contribution calculated by the "dielectronic" .WGFA files is not anymore necessary. To avoid double-counting this contribution, the transitions from levels below the ionization threshold in the "dielectronic" .WGFA files have been given a null wavelength, so they can be removed from the spectrum without having to change the way the "dielectronic" level population are handled.

6.14 Continuum calculations

An IDL routine to include the two photon continuum has been added to CHIANTI, while the free-bound and free-free continuum (bremsstrahlung) routines have been revised. See Young et. al. (2002) for more details.
Note that the output units of the continuum routines are by default 10^{-40} ergs sr^{-1} s^{-1} Å^{-1} per unit emission measure $\int N_e N_H dh$.
On the other hand, the SolarSoft routine CONFLX outputs a continuum in photons s^{-1} Å^{-1} assuming an emission measure $\int N_e^2 dh = 10^{50}$.

6.14.1 Two photon continuum

The two-photon continuum is calculated with `two_photon.pro`.

Transitions in hydrogen-sequence ions

The first excited level ($2s\ ^2S_{1/2}$) of the hydrogen iso-electronic sequence ions can decay only by means of forbidden magnetic dipole and two-photon transitions. The importance of the competing magnetic dipole transition increases with Z but for nickel ($Z = 28$), the two-photon transition rate is roughly 5 times that of the magnetic dipole rate.
The spectral emissivity (erg cm^{-3} s^{-1} sr^{-1} Å^{-1}) for optically-thin two-photon emission at wavelength λ is given by:

$$\frac{d\epsilon_{i,j}}{d\lambda} = \frac{hc}{4\pi\lambda} A_{ji} N_j(X^{+m}) \phi(\lambda_0/\lambda) \tag{63}$$

where $A_{j,i}$ (sec^{-1}) is the Einstein spontaneous emission coefficient (A value); $N_j(X^{+m})$ is the number density of the level j of the ion X^{+m}; ϕ is the spectral distribution function; and λ_0 is the wavelength corresponding to the energy difference between the excited and ground level.

Two-photon continuum transitions in helium-sequence ions

For the helium iso-electronic sequence, the second excited level ($1s2s\ ^1S_0$) decays through a forbidden magnetic dipole and two-photon transitions.

6.14.2 Bremsstrahlung

The bremsstrahlung emission is calculated with `freefree.pro`. This routine has been rewritten ex-novo. It now includes the Itoh et al. (2000) and Sutherland (1998) gaunt factors.
Itoh et al. (2000) have provided an analytical fitting formula for the relativistic thermal bremsstrahlung gaunt factors, and this is now added to CHIANTI. The fitting formula is valid for the ranges $6.0 \leq \log T \leq 8.5$ and $-4.0 \leq \log(hc/k\lambda T) \leq 1.0$. For temperatures below $\log T = 6.0$ we retain the non-relativistic Gaunt factors of Sutherland (1998) for computing the continuum. The condition $\log(hc/k\lambda T) \leq 1.0$ results in some of the low wavelength points being inaccurately represented by the Itoh et al. fitting formula. For these wavelengths the Gaunt factors of Sutherland (1998) are used to compute the continuum level. The relativistic free-free continuum is almost identical to the non-relativistic continuum at low temperatures. At $T = 1 \times 10^8$ K (the maximum temperature permitted by the ion balance calculations contained in CHIANTI) the relativistic continuum is around 1% higher near the peak of the distribution.

6.14.3 Free-bound continuum

The free-bound continuum emission is calculated with `freebound.pro`. This routine has been rewritten. The new routine uses the the Karzas and Latter (1961) approximation to the photoionization cross-sections and calculates free-bound gaunt factors for levels n=1-6. Additional data files have been created for this purpose. For example, free-bound radiation produced by recombination of an electron onto C IV to produce C III will use the data in the `c\_3.fblvl` file.

7 Some examples on how to use the software

In what follows we review the main points about the new software. We hope you find it useful and enjoy using it !

7.1 Calculating line intensities.

For an user-friendly, widget-based approach the best option is to use CH_SS:

```
IDL >ch_ss
```

This widget allows the user to calculate synthetic spectra in two basic steps. Basically, you follow the various widgets from top left to lower right to set the desired parameters. First calculate the line intensities. These values can be saved for later use. Next, specify further parameters such as the elemental abundances and instrumental spectral resolution and then calculate and plot the spectrum. These values can also be saved for later use. The HELP buttons in the widget provide short descriptions of the required information. More details are given below.
Alternatively, for e.g. **background jobs**, the routine CH_SYNTHETIC can be used.
`ch_synthetic.pro` calculates line intensities assuming constant pressure or density (or a

model T,N), **without the abundance factor**. One of the reasons why element abundances are not included in the line intensities calculation is so that it is easier for the user to see how modifying abundances affects their spectra in e.g. `ch_ss.pro`. The calling sequence is:

```
IDL> ch_synthetic, wmin, wmax, output=output, err_msg=err_msg, msg=msg, $
      pressure=pressure, density=density, $
      model_file=model_file, all=all,sngl_ion=sngl_ion, $
      photons=photons,  masterlist=masterlist,  $
      save_file=save_file , verbose=verbose,$
      logt_isothermal=logt_isothermal,  logem_isothermal=logem_isothermal,$
      goft=goft, ioneq_name=ioneq_name, dem_name=dem_name, $
      noprot=noprot, rphot=rphot, radtemp=radtemp, progress=progress
```

The routine has many KEYWORDS and the reader is referred to the routine header for full details. The important parameters that must be set are:

- `Wmin, Wmax:   minimum  maximum of the desired wavelength range in Angstroms`
- The (Te,Ne) model for the calculation:

```
      Pressure:  pressure in emitting region (Pe,  cm^-3 K).
                 Only a single value is accepted, and the calculation is
                 performed at constant pressure.

      Density:   density in emitting region (Ne, cm^-3).
                 Only a single value is accepted, and the calculation is
                 performed at constant  density, unless LOGT_ISOTHERMAL is
                 defined. In this case, DENSITY can be an array of values, but
                 has to have the same number of elements as LOGT_ISOTHERMAL.


      model_file    Full path of the (Te,Ne) file if defined.
                    This file should have two columns, one with the Te (K)
                    values, and one with the Ne (cm^-3) values. If these
                    values are not sorted in ascending order of Te, the
                    routine does sort them.
```

- IONEQ_NAME: The ionization fraction file to be used. The program will prompt the user to select one if not defined.
- OUTPUT: The name of the structure containing the line intensities and details.

The line intensities are calculated either in the isothermal approximation, in which case the following has to be defined:

```
LOGT_ISOTHERMAL:   Array of logarithmic temperatures.
LOGEM_ISOTHERMAL:  Array of logarithmic emission measures (0 by default).
```

or by folding the $G(T)$ with a differential emission measure DEM contained in the file specified by DEM_NAME. The program will prompt the user to select one if not defined.
Example:

```
IDL> ch_synthetic, 10,20., output=str , pressure=1.e+15,$
     ioneq_name=!ioneq_file,$
        dem_name=concat_dir(concat_dir(!xuvtop,'dem'),'flare.dem'),$
        /photons, /noprot, /all, sngl_ion=['fe_17','fe_18']
```

Creates an output structure `str` that contains the line intensities of only Fe XVII and Fe XVIII in the 10–20 Å range calculated at constant pressure of 10^{15}, with the default ionization balance (defined by `!ioneq_file`) and the DEM values in `flare.dem` in the standard CHIANTI distribution (if not supplied these files can be selected with a widget). Line intensities are in photons cm^{-2} s^{-1} sr^{-1} (KEYWORD photons), the proton rates are not included (KEYWORD noprot), and all the lines in the database (KEYWORD all) are included (also the lines with only theoretical energy levels).
You can see the contents of the structure with e.g.

```
IDL> help, str,/st
IDL> help, str.lines[0],/st
```

The last command shows the first structure associated with the first spectral line.

7.2 Saving, restoring and exporting the CHIANTI line intensities structure

The CHIANTI line intensities structure can be saved and later restored from the command line in various ways. We suggest two:

1. as IDL binary files using the SolarSoft routines:

   ```
   IDL> savegen, file='ch_int_10_20_fe.genx', struct=str
   IDL> restgen, file='ch_int_10_20_fe.genx', struct=str
   ```

   to save and restore the IDL structure `str` in the file `ch_int_10_20_fe.genx`.

   Please note that we discourage the use of e.g.:

   ```
   IDL> save, file='output.save', output
   IDL> restore, file='output.save'
   ```

   since IDL save files generated with later versions of IDL are usually not readable with earlier versions.

2. as FITS binary tables, that can be easily exported and read by different platforms. We have written two IDL routines:

```
IDL> ch_write_fits, str, 'output.fits'
IDL> ch_read_fits,'output.fits', str
```

to save and restore the IDL structure `str` in the FITS file `output.fits`. Aside from an introductory HEADER, the contents of the IDL structure are converted into two binary tables. Extensive comments are added.

In either case, the structure saved in the `.genx` and `.fits` files can be restored via CH_SS to later create a spectrum.

7.3 Create a latex or ascii file with all the line details

For an user-friendly approach the best option is to use CH_SS:

```
 IDL> ch_ss
```

Alternatively, if you have already calculated a line intensity structure (as shown above), you can use CH_LINE_LIST. This program creates a latex or an ascii file of predicted spectral line intensities and wavelengths corresponding to selected parameters.
The routine has many KEYWORDS. Please read the header for details. The calling sequence is:

```
 IDL> ch_line_list, transitions, outname, latex=latex, ascii=ascii, $
      wmin=wmin,wmax=wmax,$
      SPECTRUM=SPECTRUM, abundfile=abundfile, min_abund=min_abund, $
      minI=minI,photons=photons,kev=kev, $
      all=all,no_sort=no_sort, sngl_ion=sngl_ion
```

Example:

```
IDL> restgen, file='ch_int_10_20_fe.genx', struct=tran
IDL> ch_line_list, tran, 'ch_line_list.tex', /latex,$
     abundfile=concat_dir(concat_dir(!xuvtop,'abundance'),'cosmic.abund'),$
     mini=1e13
```

This creates a latex file `ch_line_list.tex` where only lines with an intensity greater than 10^{13} (KEYWORD mini) are included, and the `allen.abund` file in the standard CHIANTI distribution is used (if not supplied it can be selected with a widget).
Then, you have to latex the file three times, and optionally `xdvi` it:

```
unix> latex ch_line_list
unix> latex ch_line_list
unix> latex ch_line_list
unix> xdvi ch_line_list
```

Table 10: *Line List*

Ion	λ (Å)	Transition	T$_{\rm max}$	Int
Fe XVII	12.1227	2p^6 ^{1}S$_0$ - 2p^5 4d ^{1}P$_1$	6.9	1.11e+14
Fe XVII	12.2639	2p^6 ^{1}S$_0$ - 2p^5 4d ^{3}D$_1$	6.9	9.81e+13
Fe XVII	13.8231	2p^6 ^{1}S$_0$ - 2s 2p^6 3p ^{1}P$_1$	6.9	6.25e+13
Fe XVIII	13.9540	2s^2 2p^5 ^{2}P$_{3/2}$ - 2p^4 (^{1}S) 3d ^{2}D$_{5/2}$	6.9	2.06e+13
Fe XVIII	14.1519	2s^2 2p^5 ^{2}P$_{3/2}$ - 2p^4 (^{1}D) 3d ^{2}D$_{3/2}$	6.9	1.35e+13

If you do not have it already, you will need the package longtable.sty that is distributed as part of `ftp://cam.ctan.org/tex-archive/macros/latex/required/tools.tar.gz`
You will obtain a table like the one shown in Table 10.
Alternatively, you can also create a latex file with a list of line identifications and intensities using the wrapper routine LATEX_WVL_DEM:

```
 IDL > latex_wvl_dem,100.,200., pressure=1.e+15,mini=1.
```

However, `latex_wvl_dem` calls `ch_synthetic` and `ch_line_list`, and if you want to modify some of the parameters of `ch_line_list` (e.g. the minimum intensity) you will have to redo the calculation which will take some time. Windows will pop up so that you can select the abundance, the ionization equilibrium and the differential emission measure files. This will create by default a file linelist.tex in the user's working directory, by default.
To **create an ascii file with the line details** you can follow a similar approach, i.e.:

```
IDL> restgen, file='ch_int_10_20_fe.genx', struct=tran
IDL> ch_line_list, tran, 'ch_line_list.ascii', /ascii,$
     abundfile=concat_dir(concat_dir(!xuvtop,'abundance'),'allen.abund'),$
     mini=1e13
```

Alternatively, you can also use the wrapper routine

```
 IDL > ascii_wvl_dem,100.,200.,pressure=1.e+15,mini=1.
```

However, `ascii_wvl_dem` calls `ch_synthetic` and `ch_line_list`, and if you want to modify some of the parameters of `ch_line_list` (e.g. the minimum intensity) you will have to redo the calculation which will take some time.

7.4 Calculating continuum intensities

For example, to calculate the free-free, free-bound and two-photon continuum at a temperature of 5×10^6 K, for wavelengths at 1 Å intervals between 1 and 50 Å:

```
freefree,5.e+6,findgen(50)+1.,ff
freebound,5.e+6,findgen(50)+1.,fb
two_photon,5.e+6,findgen(50)+1.,tp
```

```
window,0
plot,findgen(50)+1.,ff+fb+tp,xtit='Wavelength (A)'
oplot, findgen(50)+1.,ff,line=2
oplot, findgen(50)+1.,fb,line=3
oplot, findgen(50)+1.,tp,line=4
```

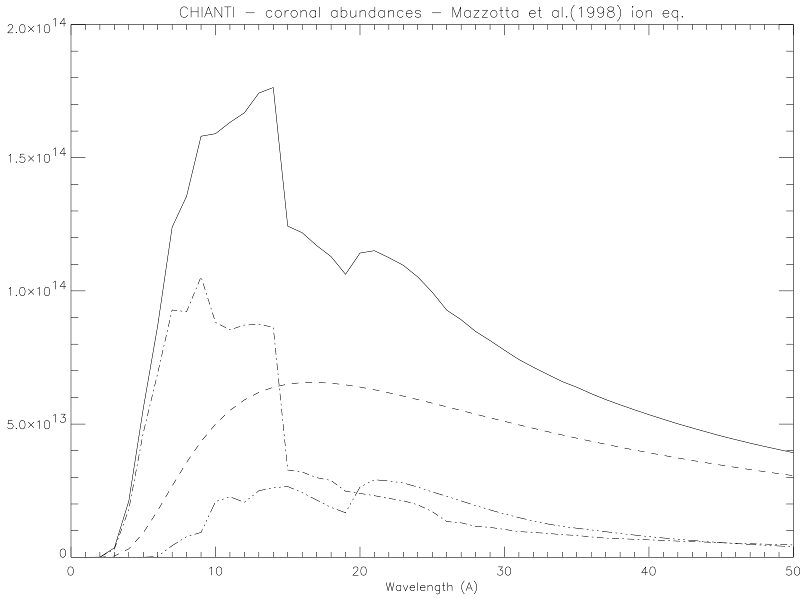


Figure 2: Continuum in the 1-50 Å range.

Note that the intensities are in units of 10^{-40} ergs cm^3 s^{-1} sr^{-1} Å^{-1} per unit emission measure $\int N_{\rm H} N_{\rm e} dh$ (cm^{-5}).
If DEM values are passed to the routines (via the keyword DEM_INT), it is assumed that they are given as $N_{\rm H} N_{\rm e} dh/dT$. The units are 10^{-40} ergs cm^{-2} s^{-1} sr^{-1} Å^{-1} in this case.

7.5 Creating a synthetic spectrum with the continuum

The structure created by CH_SYNTHETIC can be restored via CH_SS to create a spectrum. Alternatively, it can be used as an input to the program MAKE_CHIANTI_SPEC. This program creates the **CHIANTI SPECTRUM structure** (read Sect. 11 for details), an OUTPUT structure similar to the structure created by CH_SYNTHETIC, with some additional tags. The calling sequence is:

```
IDL> make_chianti_spec, TRANSITIONS, LAMBDA, OUTPUT, BIN_SIZE=BIN_SIZE,  $
        INSTR_FWHM=INSTR_FWHM,  BINSIZE=BINSIZE, $
        WRANGE=WRANGE, ALL=ALL, continuum=continuum, $
        ABUND_NAME=ABUND_NAME, MIN_ABUND=MIN_ABUND, $
        photons=photons, file_effarea=file_effarea, $
        err_msg=err_msg,  verbose=verbose
```

The routine has many keywords and options. Please read Sect. ?? for details.

```
IDL> restgen, file='ch_int_10_20_fe.genx', struct=tran
IDL> make_chianti_spec, tran, lambda, struct,/CONTINUUM, $
     BIN_SIZE=0.01, instr_fwhm=0.1,  WRANGE=[10.,19.],$
      abund_name=concat_dir(concat_dir(!xuvtop,'abundance'),'cosmic.abund')
```

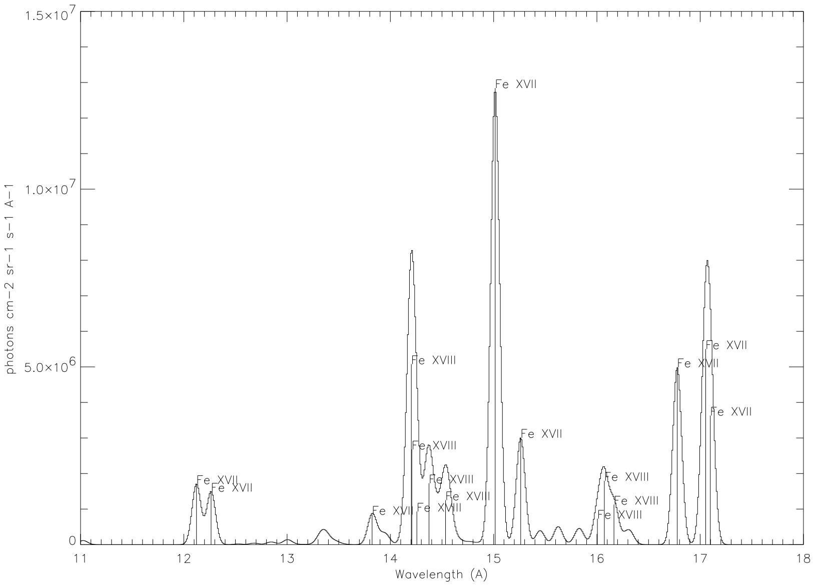


Figure 3: Synthetic spectrum created by MAKE_CHIANTI_SPEC.

Some Caveats:

You may find that the calculation is slow. This is usually due to the continuum calculation. In general, **it is advisable not to calculate spectra over large wavelength ranges**. In any case you can speed up the continuum calculation by reducing the numbers of elements, using the KEYWORD MIN_ABUND.
To see the contents of the structure:

```
IDL> help, struct,/st
IDL> help, struct.lines[0],/st
```

While to show the spectrum and the main contributing lines:

```
IDL> window,0 & plot,struct.lambda,struct.spectrum
      for i=0,n_elements(struct.lines) -1 do $
     if struct.lines[i].peak gt 7e5 then $
   xyouts, struct.lines[i].wvl,  struct.lines[i].peak, struct.lines[i].snote
```

It may be useful to save the SPECTRUM structure, that can be later inspected with the widget CH_SS:

```
IDL> savegen, file='ch_spectrum_10_20_fe.genx', struct=struct
IDL> ch_write_fits, struct, 'ch_spectrum_10_20_fe.fits'
```

Alternatively, the wrapper routine SYNTHETIC (see Fig 1) can also be used to calculate CHIANTI line intensities. For example:

```
 IDL > synthetic, 150., 200., 1., pressure=1.e+15, wvl, spectrum, list_wvl, list_iden
```

will create a synthetic spectrum with a resolution of 1 Å between 150 and 200 Å for a specified set of abundances and differential emission measure at a constant pressure of 1.e+15 (N_e T cm^{-3} K). The output arrays wvl, spectrum contain the wavelengths and the intensities (in erg cm^{-2} s^{-1} sr^{-1} $Å^{-1}$ by default). The output arrays ist_wvl, list_ident contain the list of wavelengths and descriptions of the lines that made up the spectrum.
Windows will pop up so that the user can select the abundance file, the ionization equilibrium and the differential emission measure. A spectrum is created by convolving with a Gaussian profile with a FWHM of 1 Å. If the /CONTINUUM keyword had been set, then the continuum would also have been calculated and added to the spectrum. To plot the spectrum and interactively identify lines:

```
 IDL > synthetic_plot, wvl, spectrum, list_wvl, list_ident, 2.
```

by clicking the left mouse button, a list of predicted lines within 2 Å of the selected wavelength will be printed out along with their predicted intensity. Clicking the right mouse button will exit the procedure.

7.5.1 Create a spectrum in the isothermal approximation

For an user-friendly approach the best option is to use CH_SS:

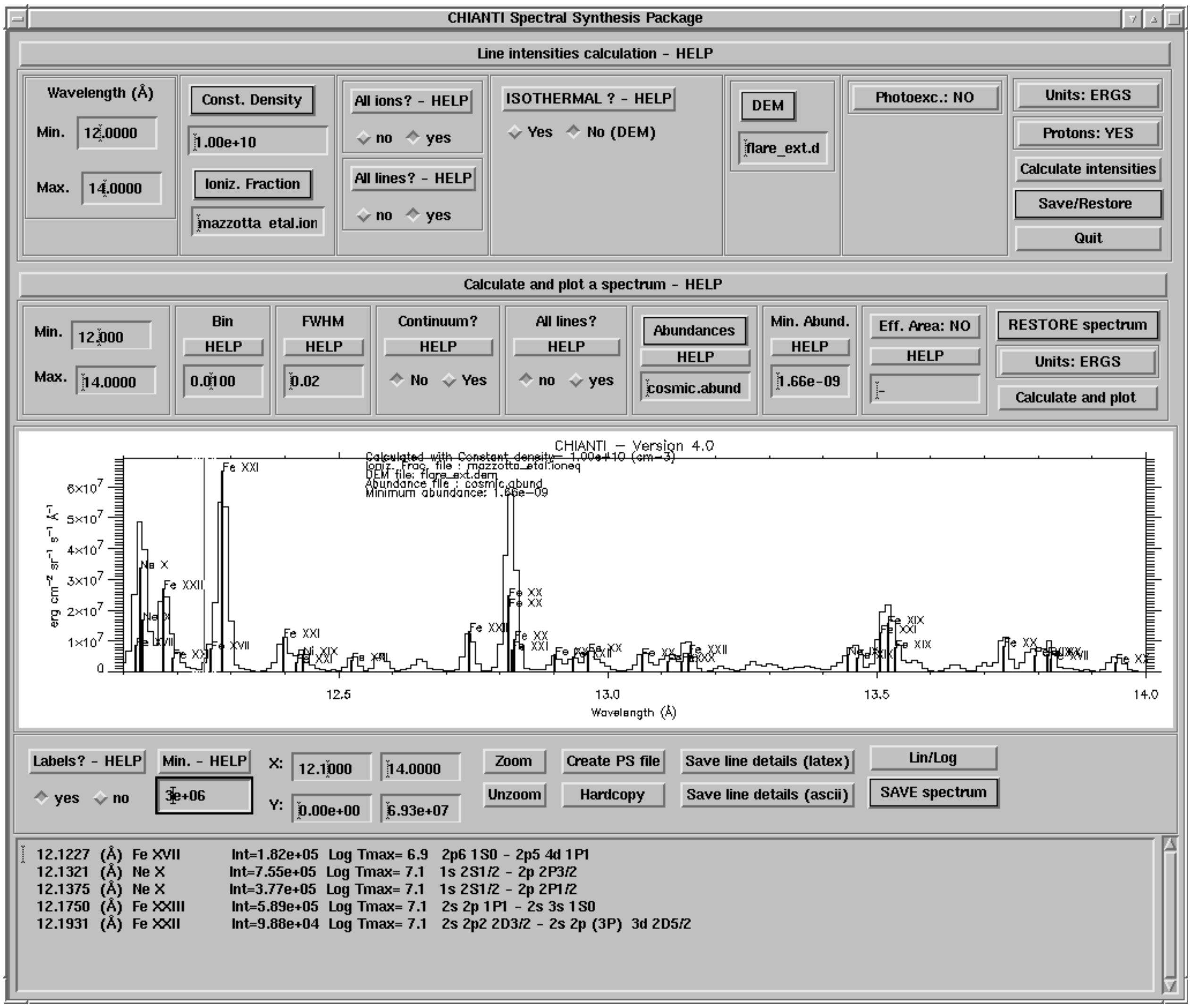
```
 IDL >ch_ss
```

Alternatively:

```
 IDL > isothermal, 150., 200., 1., [1.e6], wvl, spectrum,$
       list_wvl, list_ident, edensity=1.e9,$
      ioneq_name=!xuvtop+'/ioneq/mazzotta_etal.ioneq',$
       abund_name=!xuvtop+'/abundance/cosmic.abund'


 IDL>  synthetic_plot, wvl, spectrum, list_wvl, list_ident, 1.
```

calculates an isothermal synthetic spectrum with a resolution of 1 Å between 100 and 200 Å for a specified set of abundances and differential emission measure at a constant density N_e = 10^9 cm^{-3}. The output arrays wvl, spectrum contain the wavelengths and the intensities (in erg cm^{-2} s^{-1} sr^{-1} $Å^{-1}$ by default). The output arrays ist_wvl, list_ident contain the list of wavelengths and descriptions of the lines that made up the spectrum. `synthetic_plot` can then be used to view the spectrum.
Note: `isothermal` now is a wrapper routine that calls `ch_synthetic` . It has particular features. Please read the header documentation.



Wavelength (Å) Min. 12.0000 Max. 14.0000

1.00e+10

mazzotta etal.ion

All ions? ◇ no ◆ yes

All lines? ◇ no ◆ yes

ISOTHERMAL ? ◇ Yes ◆ No (DEM)

flare_ext.d

Min. 12.000 Max. 14.0000

Bin 0.0100

FWHM 0.02

Continuum? ◆ No ◇ Yes

All lines? ◆ no ◇ yes

cosmic.abund

1.66e-09

```
Calculated with Constant density= 1.00e+10 (cm-3)
Ioniz. Frac. file : mazzotta_etal.ioneq
DEM file: flare_ext.dem
Abundance file : cosmic.abund
Minimum abundance: 1.66e-09
```

Labels? ◆ yes ◇ no

Min. 3e+06

X: 12.1000 14.0000

Y: 0.00e+00 6.93e+07

```
12.1227  (Å)  Fe XVII     Int=1.82e+05  Log Tmax= 6.9   2p6 1S0 - 2p5 4d 1P1
12.1321  (Å)  Ne X        Int=7.55e+05  Log Tmax= 7.1   1s 2S1/2 - 2p 2P3/2
12.1375  (Å)  Ne X        Int=3.77e+05  Log Tmax= 7.1   1s 2S1/2 - 2p 2P1/2
12.1750  (Å)  Fe XXIII    Int=5.89e+05  Log Tmax= 7.1   2s 2p 1P1 - 2s 3s 1S0
12.1931  (Å)  Fe XXII     Int=9.88e+04  Log Tmax= 7.1   2s 2p2 2D3/2 - 2s 2p (3P) 3d 2D5/2
```

Figure 4: The multi-purpose widget CH_SS

7.6 The user-friendly multi-purpose widget ch_ss.pro

CH_SS is an user-friendly multi-purpose (see Fig 1) widget that allows the calculation of line intensities (calling CH_SYNTHETIC) and of a synthetic spectrum (calling MAKE_CHIANTI_SPEC) by merging line intensities and continua. The parameters can be interactively set, and the results visually inspected. Line intensities can be saved and restored in various ways. The results can also be stored in various ways, ranging from output plots to tables of line details (using CH_LINE_LIST) or save files.
CH_SS replaces the CHIANTI_SS procedure. The calling sequence is:

```
IDL> ch_ss, font=font
```

Note that if the widget appears too large you can change the font. The widget is organised into four Sections:

7.6.1 SECTION 1 - The Calculation of the CHIANTI line intensities.

This can be done in two ways:
1-Restore a save file with the CHIANTI line intensities already calculated. This is done with the RESTORE button. `.genx` and `.fits` files can be restored.
2-Calculate CHIANTI line intensities with a call to CH_SYNTHETIC.
In this case, A series of parameters must be set:

- - Minimum and maximum wavelengths in Angstroms
- - The model used for the calculation. Three are the options: 1) a constant density (cm^{-3}) 2) a constant pressure (cm^{-}3 K) 3) a general (Te,Ne) model. In this case, a file will be read. This file should have two columns, one with the Te (K) values, and one with the Ne (cm^{-3}) values.
- - The ionization fraction file to be used. "*.ioneq" files can be selected from either the CHIANTI database, the working directory or selected via a widget.
- - All ions ? If set to yes (default), then all the ions present in the database will be included.

  If set to no, then it is possible to select a list of ions with a widget
- - All lines ? If set to no (default), only the lines for which there are observed energy levels are included

  If set to yes, also the lines that do not have corresponding observed energy levels are included. In this case, the wavelengths are calculated from the theoretical energy levels, and might not be very accurate.
- - Isothermal ? If set to no (default), a DEM file must be selected. "*.dem" files (i.e. files with a .dem extension) can be selected from either the CHIANTI database, the working directory or selected via a widget.

  If set to yes, then the user is requested to enter one or more temperatures (as logarithmic values - Log T) and correspondent column emission measures EM logarithmic values. NOTE: if more than one value is entered, then the sequence must be separated by commas (e.g.: 6.0, 6.5, 7.), and both Log T and Log EM must have the same number of values
- - Photoexcitation ? If set to yes, you have to define: Trad: The blackbody radiation field temperature R/Ro: Distance from the centre of the star in stellar radius units
- - Units: Photons or Ergs
- - Protons: If set to Yes, the proton data are used to calculate the level population

Once all the parameters have been defined, the user should click on the "Calculate intensities" button to start the calculation (which calls CH_SYNTHETIC).

Once the calculation is finished, an IDL structure is loaded into memory. It is then possible to save it for later use by clicking on the "SAVE" button.
Once the IDL structure with the line intensities is in the memory, it is then possible to calculate and plot a spectrum (SECTION 2).

7.6.2 SECTION 2 - calculation of a synthetic spectrum

This section controls the parameters that are needed to fold the line intensities and the continua into a synthetic spectrum. These parameters are used by MAKE_CHIANTI_SPEC. Before this is done, a set of line intensities MUST be in the program memory. This is done either by calculating the intensities or by restoring a save file with previously calculated values (SECTION 1). Setting the parameters:

- -Minimum and maximum wavelengths.
- -spectrum bin size in Angstroms in Angstroms. Disallowed if an Effective area file is used.
- -instrumental FWHM: Setting this to a non-zero value broadens each of the spectral lines with a Gaussian of the specified FWHM (in Angstroms) so mimicking the effects of instrumental broadening.
- -continuum: Add continua to the binned spectrum: free-free, free-bound and two-photon. Please note that the continuum calculation takes some time and you may want to define a minimum abundance value to speed the calculations.
- - All lines ? If set to no (default), only the lines for which there are observed energy levels are included. If set to yes, the "unobserved lines" will be added, but only if they are present in the structure.
- -elemental abundances: "*.abund" files (i.e. files with a .abund extension) can be selected either from the CHIANTI database, the working directory, or via a widget.
- -select a minimum abundance value: If set not null, only the lines of those elements which have an abundance greater than the value set are selected. Also, the continuum is calculated only for those elements which have an abundance greater than the value set. This can significantly speed up the calculations. By default, the minimum value in the selected abundance file is used.
- Eff. Area: (Yes/No): If you want to fold the spectrum with an effective area. If set to Yes, you are requested to choose an input ascii file with two columns, the wavelength and the effective area values (cm^2). The spectrum is multiplied with these values. the wavelenghts in the file (that might not be linear) are used to create the spectrum. Note that this option only works well if a sufficient number of bins is given. The line intensities contributing to each bin are summed, and subsequently convolved with a gaussian of full-width-half-maximum FWHM, if FWHM is not set = 0. Please note that the convolution might not work if a small number of bins is defined.

Also note that to have the correct output units (counts s-1 bin-1) the appropriately scaled DEM (or EM) values must be provided.

After this, by clicking on the ”Calculate and plot” button the program calculates and plots the synthetic spectrum.
Once the spectrum is displayed, it is then possible to view the details of the lines by clicking with the mouse in the plot window, and to perform various operations by clicking on the buttons in SECTION 3

7.6.3 SECTION 3 - selection of parameters for plotting and output

This Section allows the user to select a few parameters for the plotting, and to create different types of OUTPUT.

- Labels ? : Setting this to yes plots a vertical line for each spectral line in the spectrum, and also writes a label above the strongest lines indicating the ion from which the line arises.
- Min.: Only lines which have an intensity greater than the value set here will be listed and, if requested, labelled and selected for inclusion in the various outputs. Setting the value=0. will result in all lines being listed and written in the outputs.
- X,Y, XOOM, UNZOOM: It si possible to select a region of the spectrum, by zooming with the use of the mouse or by setting the X,Y ranges.

  NOTE that only the line details and portion of the spectrum shown will be output.
- LINEAR/LOG To plot the spectrum in linear or log scale
- Create PS file: A postscript file is created.
- Hardcopy: the postscript file ”idl.ps” is created and sent to the default printer.
- Save Line details (latex): The details of the lines shown in the plot will be saved in a latex file.
- Save Line details (ascii): The details of the lines shown in the plot will be saved in an ascii file.
- Save Spectrum (ascii): The X,Y values of the spectrum are saved in an ascii file.
- Save Spectrum (IDL/FITS): The details of all the lines and the arrays of the X,Y values of the spectrum are saved into an IDL or FITS file.

Finally, SECTION 4 is a text information window, where various messages are printed.
Clicking the cursor on any part of the displayed spectrum will give a listing of the lines within a range of Angtroms of that wavelength. Text information on the lines is printed.

7.7 Photoexcitation from any user-provided radiation field

Sect. 6.12 described how photoexcitation can be modelled by using any radiation field, and Sect. 2.2 of the v.5 CHIANTI paper (Landi et al. 2005) described one example for modeling Doppler dimming of the O VI 1032,1038 Å multiplet . The IDL function used for this modeling is given below.

```
FUNCTION o6_lines, lambda, a

;   Vernazza & Reeves (1978) give the quiet Sun O VI 1032 flux to be
;   305.28 erg/cm2/sr/s. The 1038 line is blended with C II, so I take
;   it to be half of the 1032 line. I assume the FWHMs of the lines are
;   0.2 angstroms.
;
;   A   Velocity (km/s) relative to emitting ions of the structure emitting
;       the radiation field. A positive velocity implies a redshift.

IF n_elements(a) EQ 0 THEN a=0.

siz=size(lambda)
spectrum=dblarr(siz[1],siz[2])

cc=2.998d5   ; speed of light, km/s

p1=305.28/0.2
p2=p1/2.

c1=1031.914
c1=c1+ (a/cc * c1)
;
c2=1037.615
c2=c2+ (a/cc * c2)
w=0.2/2.35

i=where(abs(lambda-c1) LE 6.*w)
IF i[0] NE -1 THEN spectrum[i]=p1*exp(-(lambda[i]-c1)^2/2./w^2)*4.*!pi/2.998d10

i=where(abs(lambda-c2) LE 6.*w)
IF i[0] NE -1 THEN spectrum[i]=spectrum[i]+p2*exp(-(lambda[i]-c2)^2/2./w^2)*4.*!
pi/2.998d10

return,spectrum
END
```

This function can then be used in show_pops or emiss_calc as follows:

```
IDL> show_pops,8,6,radfunc='o6_lines, 20',rphot=1.1
```

where 'o6_lines, 20' indicates that the velocity A is set to 20 km/s. A zero velocity can be set simply by using radfunc='o6_lines'. RPHOT specifies the distance from the centre of the star in stellar radius units.
The effects of many different velocities can be studied by doing, e.g.,

```
v=findgen(11)*10.
for i=0,10 do begin
  radfunc_string='o6_lines, '+trim(v[i])
  show_pops,8,6,radfunc=radfunc_string,rphot=1.1
endfor
```

Up to 2 input parameters are allowed for radfunc and are specified by, e.g., radfunc='radfunc, a, b'.
Currently the RADFUNC= keyword is only available for the routines show_pops and emiss_calc.
Another example of radfunc is a blackbody

```
function udens_bb, lambda

t=6d3    ; temperature of Sun, 6000 K
ee=1.439d8/lambda/t

result=8.*!pi*1.986d-8/lambda^2*(1d8^3)/lambda^3/((exp(ee)-1))
return,result

END
```

which is specified to show_pops as

```
IDL> show_pops,8,6,radfunc='udens_bb',rphot=1.1
```

The user should verify that this gives the same results as using the standard CHIANTI inputs

```
IDL> show_pops,8,6,radtemp=6000.,rphot=1.1
```

7.8 Non-maxwellian distribution of electron velocities

The following commands reproduce the numbers in Table 3 of the v.5 CHIANTI paper (Landi et al. 2005). Basically, we want to study the effects of non-Maxwellian distributions on two key line ratios of O VI, involving the strong lines at 1032 Å, 173 Å and 150 Å. We consider a distribution comprised of two Maxwellians at log $T = 5.5$ and log $T = 6.0$, with the coefficients [a1,a2]=[0.75,0.25].

```
IDL> em=emiss_calc(8,6,temp=[5.5,6.0],sum_mwl_coeff=[0.75,0.25],dens=9.0)
IDL> em150=em[40].em
IDL> em173=em[43].em
IDL> em1032=em[77].em
IDL> print,em150/em1032,em173/em1032
     0.030072876
     0.043500302
```

The effects of non-Maxwellians on level populations can be demonstrated with the show_pops routine, e.g.,

```
IDL> show_pops,8,6,lev=-8

Log10 density:      10.0
Log10 temperature:   5.5

  1           1s2.2s 2S1/2  1.00e-00
  2           1s2.2p 2P1/2  2.30e-07
  3           1s2.2p 2P3/2  4.52e-07
  4           1s2.3s 2S1/2  5.07e-11
  5           1s2.3p 2P1/2  7.26e-12
  6           1s2.3p 2P3/2  1.44e-11
  7           1s2.3d 2D3/2  5.30e-12
  8           1s2.3d 2D5/2  7.96e-12

IDL> show_pops,8,6,lev=-8,temp=[5.5,6.0],sum_mwl_coeffs=[0.75,0.25]

Log10 density:      10.0
Using a sum of Maxwellians

  1           1s2.2s 2S1/2  1.00e-00
  2           1s2.2p 2P1/2  2.26e-07
  3           1s2.2p 2P3/2  4.43e-07
  4           1s2.3s 2S1/2  8.93e-11
  5           1s2.3p 2P1/2  1.60e-11
  6           1s2.3p 2P3/2  3.16e-11
  7           1s2.3d 2D3/2  1.13e-11
  8           1s2.3d 2D5/2  1.69e-11
```

where it can be seen that the n=3 level populations are enhanced by factors 2 by the high temperature component to the distribution.

7.9 Looking at level populations

There are several routines available to plot and obtain the level populations.

7.9.1 plot_populations

To plot the populations of the first 4 levels of Si III as a function of density at a temperature of 3 x 10^4 K:

```
 IDL > plot_populations,'si_3',3.e+4,4
```

Note: this routine has been significantly improved in CHIANTI version 9, and now has many options.

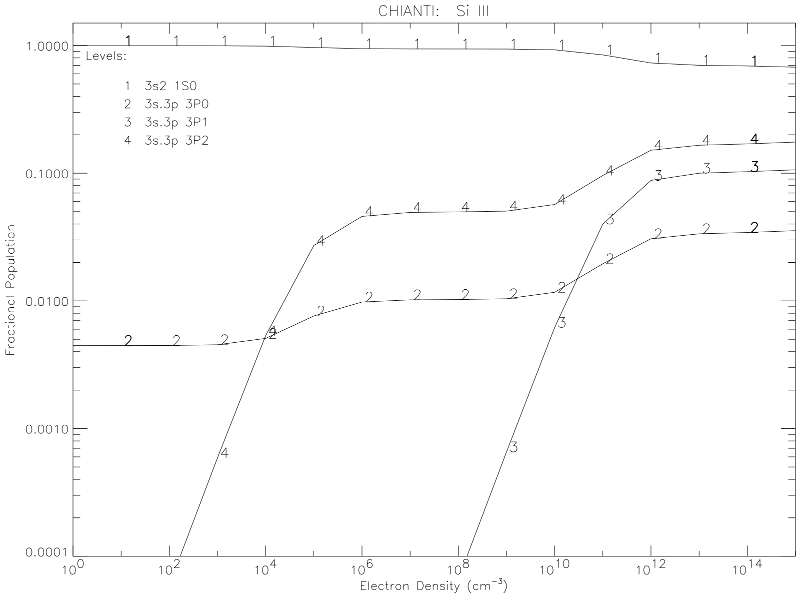


Figure 5: Output plot of PLOT_POPULATIONS

Optionally, output files can be created.

7.9.2 ch_pops routine

```
IDL> pop=ch_pops('fe_13')
```

This prints the populations of the most populous levels to the screen, while information for all levels is output to the pop structure. The tags of the output are:

```
   DENS            DOUBLE       1.0000000e+10
   TEMP            DOUBLE           1778279.4
   LEVEL           STRUCT    -> <Anonymous> Array[749]
   RADTEMP         FLOAT          -1.00000
   RPHOT           FLOAT          -1.00000
   PROTON          STRING    'yes'
   VERSION         STRING    'CHIANTI 8.0.2'
   DATE            STRING    'Wed Jun 28 15:36:26 2017'
   SUM_MWL         INT               0
   SUM_MWL_COEFFS  FLOAT          -1.00000
```

The tag `level` is a structure array, with the tags:

```
   INDEX            INT               1
   TERM             STRING    '3s2 3p2 3P0'
   POP              DOUBLE         0.11817674
```

The level population is given in the tag `pop` and it is given as the population relative to the population of the ion as a whole, so summing the `pop` values over all levels gives 1.
By default, populations are calculated at an electron number density, $N_{\rm e}$, of 10^{10} cm^{-3} and the $T_{\rm max}$ of the ion (computed with `ch_tmax`) is used for the temperature. Keywords exist to switch off proton rates (`/noprot`) and level-resolved ionization and recombination rates (`/noionrec`) where these are available for the ion. Photon excitation and non-Maxwellian distributions can be included using the standard CHIANTI keywords (see CHIANTI User Guide).
Note that `ch_pops` is a wrapper for the older routine `show_pops`, which is itself a wrapper for calling `pop_solver`.

7.9.3 get_populations

```
IDL> get_populations,'fe_13', 1e6, output='fe_13_pop'
```

This routine also returns the populations and has several keywords.

7.9.4 pop_plot

`pop_plot.pro` plots the values of

$$10^{20}\,\Delta E\,n_{\rm j}A_{\rm ji}/N_{\rm e} \tag{64}$$

against $N_{\rm e}$. As discussed in Sect. 6, if we only study lines in the emission measure analysis for which this quantity is independent of density, then the derived emission measure is independent of the plasma density.

Example: For Fe XIII, select a line/blend from lines in the range 200 to 205 Å

```
    pop_plot,26,13,wrange=[200,205]
```

Note how no single line shows zero density dependence, and so care should be taking in using Fe XIII in emission measure analyses. Compare with Fe XVI:

```
    pop_plot,26,13,wrange=[330,370]
```

where both the 335 and 360 lines are OK.

7.9.5 Looking at the processes that populate each level

To assess the contributions of the different physical processes to the population of a specified level within an ion, use POP_PROCESSES.

```
 IDL> pop_processes,'fe_13',lev=4

Level:   3s2.3p2 1D2

Log10 Temperature:   6.2
Log10 Density:      10.0

Population leaving level 4
  rad. decay:     1.60e+01     42.50%
  e de-exc:       3.29e-01      0.87%
  e exc:          2.10e+01     55.88%
  p de-exc:       2.44e-01      0.65%
  p exc:          3.75e-02      0.10%
  stim. emiss:    0.00e+00      0.00%
  photoexc:       0.00e+00      0.00%
                  --------
           TOTAL  3.77e+01


Population entering level 4
  rad. decay:     3.50e+01     92.98%
  e de-exc:       3.38e-02      0.09%
  e exc:          1.47e+00      3.92%
  p de-exc:       2.81e-03      0.01%
  p exc:          1.13e+00      3.01%
  stim. emiss:    0.00e+00      0.00%
  photoexc:       0.00e+00      0.00%
                  --------
           TOTAL  3.77e+01
```

which shows that the level population is dominated by electron excitation and cascading into the level, and by radiative decay out of the level.
Note that the rates for each physical process are multiplied by the population of originating level (this results in the totals for entering and leaving the level to balance).

7.10 Level lifetime

```
IDL> level_lifetime,'fe_13',4,lifetime
Level:     3s2.3p2 1S0
Lifetime (seconds):  9.098e-04
```

This routine calculates the lifetime of a level using the radiative decay rates stored in the CHIANTI .wgfa file. The 2nd input is the CHIANTI level index for the level you are interested in (see the CHIANTI .elvlc file). The level lifetime is returned in seconds.

7.11 Metastable levels

```
IDL> metastable_levels, 'fe_13', meta
Metastable levels are:
 1 3s2.3p2 3P0
 2 3s2.3p2 3P1
 3 3s2.3p2 3P2
 4 3s2.3p2 1D2
 5 3s2.3p2 1S0
 18 3s2.3p3d 3F4
```

This routine returns a list of levels in an ion that are metastable. 'meta' is an array with N elements, where N is the number of levels in the CHIANTI ion model. It contains values of either 0 or 1, where 1 denotes that the level is metastable.
Metastable levels are identified as those for which the largest radiative decay rate is ¡ 105 s^{-1}. This value can be modified by specifying the keyword cutoff=

7.12 Level energies

```
IDL> plot_config_energies, 'o_5'
```

This routine graphically displays the configuration energies for e.g. O V. For each configuration a box is drawn whose lower and upper edges represent the minimum and maximum energies, respectively, of the levels belonging to that configuration.

7.13 Searching for a line

If you want to list the lines within one ion around some wavelengths, you can use WHICH_LINE. For example,

```
IDL> which_line,'o_6',1032

     Wavelength   i   j  Lower level           Upper level             A-value
*      1031.914   1   3  1s2.2s 2S1/2        - 1s2.2p 2P3/2           4.28e+08
       1037.615   1   2  1s2.2s 2S1/2        - 1s2.2p 2P1/2           4.21e+08
```

Prints a list of atomic transitions and wavelengths for lines from O VI within 1% of the input wavelength (1032 Å).

7.14 Ionization and recombination

```
IDL> cross = ioniz_cross(ion,energy [, z=, ion= ])
```

cross is returned as the ionization cross section (cm^2) as a function of the specified ion and energy in (eV). The process of direct ionization and excitation-autoionization are both included. For example,

```
cross = ioniz_cross('h_1',[14.,20.])
```

returns the cross section for electron collisional ionization of H I to H II. Note that if the energies are below the ionization potential, 13.6 eV for H I, then a cross section of zero is returned for those energies.

```
IDL> rate = ioniz_rate(ion, temperature [, z= , ion =])
```

rate is returned as the ionization rate coefficient (cm^3 s^{-1}) as a function of the specified ion and temperature (K). The processes of direct ionization and excitation-autoionization are both included. For example,

```
IDL> rate = ioniz_rate('fe_14',[1.e+6, 2.e+6])
or
IDL> rate = ioniz_rate('',[1.e+6, 2.e+6], z=26, ion=14)
```

returns the rate coefficient for the ionization of Fe XIV to Fe XV at 2 temperatures. The processes of radiative and dielectronic recombination are both included.

```
IDL> rate = recomb_rate(ion, temperature [,z=, ion=])
```

rate is returned as the recombination rate coefficient (cm^3 s^{-1}) as a function of the specified ion and temperature (K). For example,

```
IDL> rate = recomb_rate('fe_14',[1.e+6, 2.e+6])
```

returns the rate coefficient for the recombination of Fe XIV to Fe XIII at 2 temperatures As an example, Fig 6 shows the ionization and recombination rate coefficients for Fe IX ('fe_9').

```
IDL> make_ioneq_all, temperature, outname='new.ioneq'
```

a file 'new.ioneq' is created in the users working directory containing the ionization equilibrium calculated at the specfied temperatures for all elements from H through Zn. Users can specify a different file name but the .ioneq extension is generally expected by the CHIANTI software.

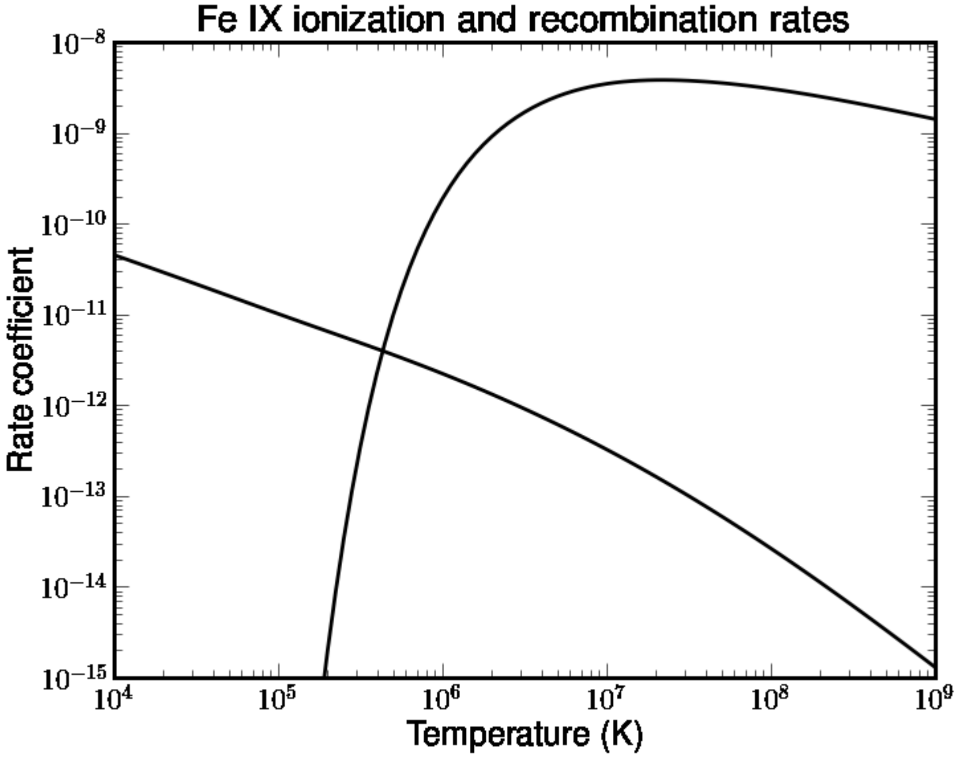


Figure 6: Ionization and recombination rate coefficients for Fe IX

7.15 Looking at the different ionisation equilibria

If you are interested to see the differences between the various ionisation equilibria for e.g. Mg, you can use:

```
IDL > plot_ioneq,'Mg'
```

You will be able to select one of the files, and optionally create a postscript file of the plot. If you are only interested in e.g. the Mg VIII, Mg IX, Mg X ions, you can type:

```
IDL > plot_ioneq,'Mg', ion=[8,10]
```

If, instead, you are interested in obtaining the temperature at the maximum ionisation fraction for e.g. Mg X, you can use:

```
IDL > print, max_temp ('Mg X')
```

You will be asked to select an ionisation equilibrium file.

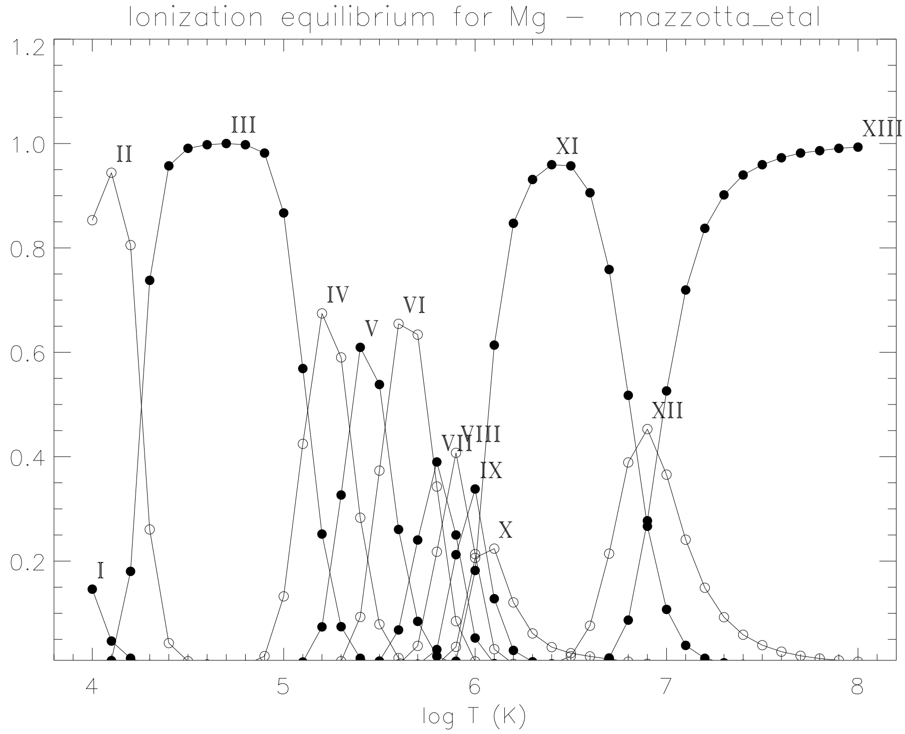


Figure 7: Output plot of PLOT_IONEQ

7.16 Density and temperature diagnostics from line ratios

Spectroscopic diagnostic line ratios in the UV wavelength range have been used extensively to determine the electron density and temperature in the solar atmosphere (cf Dere and Mason, 1981, Gabriel and Mason, 1982, Mason, 1991, Mason and Monsignori Fossi, 1994, [2]). The theoretical intensity ratios from individual ion species provide a measurement of electron density which is independent of any assumptions about the volume of the emitting region. This is of particular importance in the transition region and coronal structures. The electron density (which determines the electron pressure) is an essential parameter in the study of energy transfer mechanisms. The routines that can be used are described below.

7.16.1 The DENS_PLOTTER and TEMP_PLOTTER widgets

Both `temp_plotter.pro` and `dens_plotter.pro` call a widget-based routine (`ratio_plotter`, via the keywords /temp and /dens) that allows the thorough investigation of density or temperature sensitive ratios. Observed line intensities can be input for line ratios, and densities or temperatures derived. They allow inclusion of proton rates and photoexcitation. The calling sequence is simple:

```
IDL > dens_plotter,'o_5'
```

to study O V.

```
IDL > temp_plotter,'c_4'
```

to study C IV.
Alternatively, you can use the command-line routines, DENSITY_RATIOS and TEMPERATURE_RATIOS. They also allow inclusion of proton rates and photoexcitation via KEYWORDS.

7.16.2 The DENSITY_RATIOS procedure

The routine DENSITY_RATIOS plots the variation of line intensities with electron density, allowing density diagnostics to be studied. As an example, we can look for density sensitive line ratios of O V in the 1000 to 1500 Å wavelength region for densities between 10^8 and 10^{13} cm^{-3}:

```
IDL > density_ratios,'o_5',1000.,1500.,8.,13.,den,rat,desc
```

two windows will open and plot the relative intensities of a few O V lines. To choose the ratio of 1371.294 to 1218.393 Å line, select first the 1371.294 Å line. Another widget will appear to select the denominator. Select the 1218.393 Å line. This will chose the ratio of 1371.294 to 1218.393 which will be plotted in a new window. Values of the density and intensity ratio will be put into the variables den and rat and desc will contain a descriptive string.

```
IDL >  print, desc
IDL > CHIANTI V. 4.0 O V 1371.2939 ()/1218.3929 ()  T = 2.51e+05 (K)
```

The DENSITY_RATIOS procedure also allows to calculate the ratio at user-defined value of constant temperature. Blends are accounted for via a selection of lines.

7.16.3 The TEMPERATURE_RATIOS procedure

To calculate temperature sensitive line ratios of C IV for lines between 100 and 1600 Å for temperatures between 10^4 and 10^6 K:

```
IDL >   temperature_ratios,'c_4',100.,1600.,4.,6.,temp,rat,desc
```

As with density_ratios, a widget will appear that will allow you to select the numerator. Select the 384.175 and 384.190 Å lines as these will typically be blended in most spectrographs. Select the 1550.775 Å line for the denominator. The ratio of (384.175 + 384.190 Å) to the 1550.775 Å line as a function of temperature will be plotted and stored in the variables rat and temp, respectively. The TEMPERATURE_RATIOS procedure also allows to calculate the ratio at user-defined values of either constant pressure or constant density.

```
IDL > print, desc
IDL > CHIANTI V. 4.0 C IV 384.1750+384.1900 ()/1550.7750 () Ne = 1.00e+10 (cm!e-3!n)
```

7.16.4 The CHIANTI_NE and CHIANTI_TE widgets

These routines are now obsolete. Please use `dens_plotter` and `temp_plotter`.

7.16.5 Calculating temperatures by using different ions

Note: If you are interested in **determining an isothermal temperature by using the ratio of lines emitted by different ions (and/or elements)**, then a possible way is to first calculte the contribution functions of the lines you are interested, and then calculate their ratio. Note, however, that such determinations can be very inaccurate, since they depend on the ionisation equilibrium chosen (and eventually on the element abundance).

7.17 Calculating contribution functions

To calculate the contribution function (in units $erg\ cm^3 s^{-1} sr^{-1}$ by default) vs. temperature at a specified abundance, ionization equilibrium and pressure or density for the Fe XXIV line at 255.1 Å:

```
 IDL >  gofnt,'fe_24',250.,260.,temperature,g,desc
```

temperature, g are the arrays with the temperatures and the G(T) values. It is possible to calculate the G(T) at either constant electron density or pressure, via the KEYWORDS DENSITY or PRESSURE.
The KEYWORDS ABUND_NAME, IONEQ_NAME allow to run the routine in the background, giving names of the abundance and ionization fractions files.
The routine GOFNT allows the user to select a number of lines. If this is done, then the total sum of the G(T)'s of the selected lines is returned and plotted.
Optional outputs can be created. The default units are $erg\ cm^3 s^{-1} sr^{-1}$, unless the KEYWORD /PHOTONS is set, in which case the units are $photons\ cm^3 s^{-1} sr^{-1}$.

7.17.1 g_of_t.pro

Eq. 36 gives the definition of the *contribution* function as calculated by the `g_of_t` routine. In it's default setting `g_of_t.pro` actually calculates:

$$\Delta E\ G_\lambda(T) = 0.83\,\Delta E\ F(T)\,\frac{n_{\rm j} A_{\rm ji}}{N_{\rm e}}$$

which is more useful when considering blends of lines at different wavelengths. The ΔE can be 'disabled' with the `/no_de` keyword. It is also useful to multiply the above function by the element abundance, and this is accomplished with the `/abund` keyword. The output function is tabulated over $4.0 \leq \log T \leq 8.0$ at 0.1 dex intervals. For smaller intervals, see the `ion_interp` routine.

Examples:

```
    result=g_of_t(26,13,dens=9.)
    result=g_of_t(26,13,wrange=[200,205],/abund)
    result=g_of_t(26,13,/no_de)
```

One can also use this routine to derive the $T_{\rm mem}$ of the emission line, by way of the `ion_interp.pro` routine, e.g.,

```
    result=g_of_t(26,13,dens=9.)
    ion_interp,t,result,ti,g_ti,10
    print,ti(where(g_ti eq max(g_ti)))
```

`result` is tabulated at 0.1 dex intervals in temperature. `ion_interp` interpolates `result` and in this case gives it at 0.01 dex intervals.

7.18 Calculating emissivities

`emiss_calc.pro` calculates the ion emissivity (Eq. 32) for all transitions within the CHIANTI model of the ion. The returned data is in the form of a structure. The default is to calculate emissivities for temperatures $T_{\rm max}$ and log $T_{\rm max} \pm 0.15$, and densities log $N_{\rm e} = 8.0, 8.5, 9.0, ..., 12.0$.

Example:

```
    emiss=emiss_calc(26,13)
```

7.18.1 emiss_select.pro

Allows the selection of lines/blends from the emiss structure created by `emiss_calc.pro`. This routine is useful if you want to access the emissivities of lines directly, e.g.,

```
emiss=emiss_calc(26,13)
em202=emiss_select(emiss,wra=[200,205],sel_ind=sel_ind)
```

In this example, calling emiss_select yields a widget that allows one to select a line/blend from the 200–205 Å range. The emissivities of this line blend will be contained in `em202`, while the `emiss` index/indices of this line/blend will be contained in `sel_ind`.

7.19 Calculating radiative losses

A procedure ('RAD_LOSS') calculates the total radiative loss rate as a function of temperature for specified set of abundances and/or ionization equilibria:

```
 IDL > rad_loss,temperature,loss_rate
```

7.20 Emission measure analysis

7.20.1 integral_calc.pro

This routine calculates C_λ, defined in Eq. 39. It displays both this value and the values of $\Delta E\ C_\lambda$ and $4\pi/\Delta E\ C_\lambda$. For lines for which $n_{\rm j} A_{\rm ji} \sim N_{\rm e}$, C_λ is insensitive to $N_{\rm e}$, but for other lines $N_{\rm e}$ should be specified. Note that for blended lines only $\sum \Delta E\ C_\lambda$ and $4\pi/\sum \Delta E\ C_\lambda$ are output. The routine also outputs the $T_{\rm mem}$ of the lines, accurate to 0.02 dex.

Example: Work out C_λ for the Fe XIII lines between 200 and 205 Å at a density of 10^9 cm^{-3}.

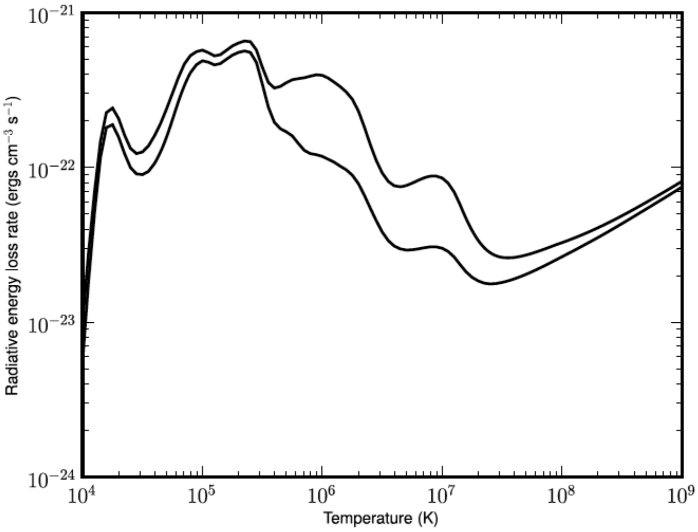


Figure 8: Radiative loss rate for coronal abundances (upper curve) and photospheric abundances (lower curve). From CHIANTI v.6, output plot of RAD_LOSS

```
    integral_calc,26,13,wrange=[200,205],dens=9.
```

From Eq. 43, an observed line intensity of 100 erg cm^{-2}s^{-1}sr^{-1} for the 202.044 line implies a column emission measure of $EM(s) = 100 \times 1.614 \times 10^{20}/Ab(Fe)$, where 1.614×10^{20} is taken from `4pi/DE*C_lambda` column of the output.
For Fe XIV, one can do:

```
    integral_calc,26,14,wrange=[210,220],dens=9.
```

and so to get the same column emission measure for Fe XIV λ211.32, an intensity of $100 \times 1.614 \times 10^{20}/2.280 \times 10^{20} = 70.8$ erg cm^{-2}s^{-1}sr^{-1} is required, where 2.280×10^{20} is the value of `4pi/DE*C_lambda` for Fe XIV λ211.32.

8 SDO/AIA temperature responses

We have added to version 9 a simple IDL program so that users can calculate the SDO AIA temperature responses of the six coronal channels using the latest understanding of the decrease in the efficiency of the various broad-band filters (as available within SSW), and the latest CHIANTI data.
Users can also calculate the responses using different parameters (e.g., densities, chemical abundances) to see how they affect them. For example, to calculate the responses for 22 May 2010 with constant pressure:

```
AIA_RESP= ch_aia_resp('20100522', pressure=1e15, $
      abund_name=!xuvtop+'/abundance/sun_coronal_1992_feldman_ext.abund',$
      ioneq_name=!xuvtop+'/ioneq/chianti.ioneq')
```

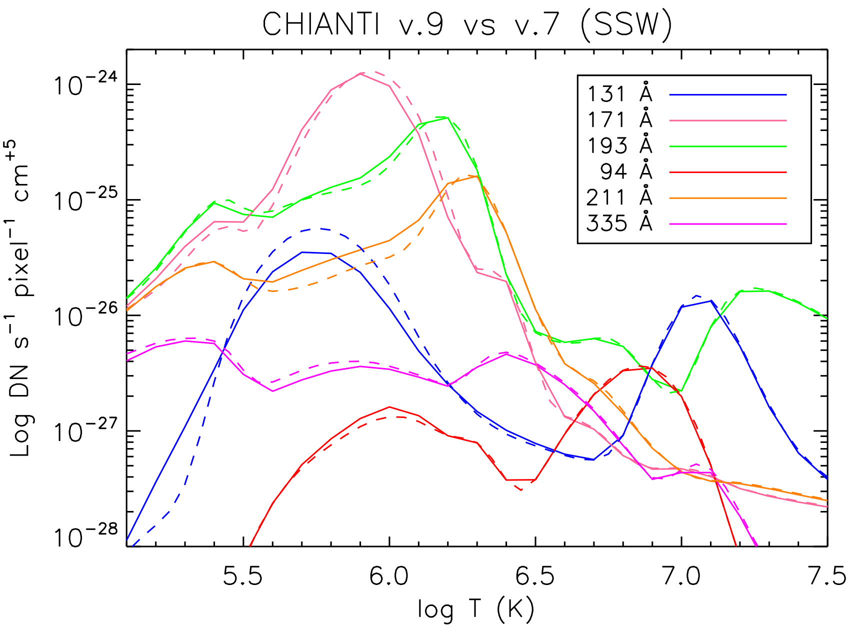


Figure 9: The AIA temperature responses as calculated with the CHIANTI Version 9 ionization equilibrium and atomic data (full lines), compared to the default AIA responses available within SolarSoft (dashed lines), which were calculated with CHIANTI v.7.

The routine first obtains the time-dependent AIA effective areas from the AIA SSW, then calculates CHIANTI isothermal spectra between 10^4 and 10^8 K and between 25 and 900 Angstroms, at a 0.1 resolution, either at constant density or constant pressure. Note that due to the off-band AIA sensitivity discovered after launch and ntroduced in 2013 within the AIA SSW effective areas, significant emission in some bands comes from wavelengths above 450 Angstroms. This emission was not originally included. Also note that the CHIANTI model for these longer wavelengths is not very accurate, epsecially when considering the continuum emission. The isothermal spectra are then multiplied by the effective areas and integrated over wavelength to produce the temperature responses.
Figure 9 shows the AIA temperature responses as calculated with the CHIANTI Version 9 ionization equilibrium and atomic data, compared to the default AIA responses available within SolarSoft, which were calculated with CHIANTI v.7. The parameters (chemical abundances and constant pressure) are the same for both. The differences in the 131 and 171 Å channel responses mainly arise due to the changes in the ionization fraction of Fe VIII and Fe IX. The other differences are mainly due to updates of the collisional excitation for each ion data-set.

9 The calculation of the DEM

The current set of programs have been substantially modified in March 2014 for version 8. They replace the previous version, retaining most of the features, and allowing different inversion methods to be used.

A further significant revision was introduced for version 9. By default, the CHIANTI_DEM program was using the subroutine *xrt_dem_iterative2.pro* [?]. This routine, widely used in solar physics and available within SolarSoft, is based on the robust chi-square fitting program *mpfit.pro*. Within this subroutine, the differential emission measure (DEM) is modelled assuming a spline, with a fixed selection of the nodes. As it turns out that the DEM solutions are quite sensitive to the choice of nodes. We have replaced the *xrt_dem_iterative2.pro* routine with a new subroutine, called MPFIT_DEM, where we allow for the definition of the number and location of the spline nodes. We also simplified substantially the code and introduced the option to input minimum and maximum limits to the spline values, which are passed to *mpfit.pro*. These limits are useful when a spectral line intensity is not measured but an upper limit can be defined. Various other options are available, as described in the header of the programs.

The user should read the documentation within the routine for more detailed examples.

Given a set of observed spectral intensities, the problem is to invert a system of integral equations like the previous one. The procedure CHIANTI_DEM solves the system and calculates the $DEM(T)$.

The inversion problem itself is not simple and requires some assumptions about the nature of the solution. A series of workshops was sponsored in 1990/91 to study differential emission measure techniques (Harrison and Thompson, 1992). It was found that most codes eventually gave consistent results, but that the DEM derived depends rather critically on the methods used to constrain the solution and the errors in the observed intensities and atomic data.

It is advisable to select a number of well resolved, unblended lines which are not density sensitive, emitted by various elements over a wide temperature interval. Appropriate values of the pressure (or density) and the elemental abundances must be chosen according to the region of the Sun being observed. The pressure value can be obtained once the values of the temperature and the density are estimated. To estimate the electron density the procedure CHIANTI_NE can be used. The temperature can be estimated for example using the procedure CHIANTI_TE.

The contribution functions $C(T, \lambda_{ij}, N_e)$ can be calculated using CHIANTI_DEM either at constant pressure or at constant electron density. It is also possible to vary the elemental abundances before starting the fit to deduce the DEM.

Many papers have been written on solar elemental abundances. A possible approach in determining elemental abundances is to use the detailed shape of the DEM distribution for ions from the same element and apply an iterative procedure to normalize the curves for different elements.

The CHIANTI_DEM procedure

The main IDL routine which has been written to perform a differential emission measure analysis of EUV spectra using the CHIANTI atomic database is CHIANTI_DEM. The resulting DEM may then be used by other procedures such as CH_SS to calculate a synthetic spectrum.

The main inputs required by CHIANTI_DEM are :

- **the file with the observed line intensities**. It can be selected using a widget-type browse from within CHIANTI_DEM or using the optional keyword FILE_INPUT='myfilename' . It must contain 5 columns of unformatted data (separated by at least one space). The 5 fields are:

1) the observed wavelength λ_{obs} [Å].

2) The observed intensity I_{obs}.

3) The corresponding uncertainty σ_{obs} on the intensity.

Note: by default the intensity units are $ergs\ cm^{-2}s^{-1}sr^{-1}$. However, via the /phot keyword, they can be $phot\ cm^{-2}s^{-1}sr^{-1}$, or with both the /phot and /arcsec keywords, they can be $phot\ cm^{-2}s^{-1}arcsec^{-1}$.

4) The value of $\delta\lambda$ [Å]. All the theoretical lines that may have contributed to the observed lines, i.e. that have a theoretical wavelength λ_{theo} in a $\lambda_{obs} \pm \delta\lambda$ range will be searched for. This value should correspond to the spectral resolution of the instrument at that wavelength.

5) The identification, written as a string of up to 20 characters. For example:

171.114 4811.0 1443.0 0.25 Fe IX

174.604 4005.0 1202.0 0.25 Fe X

180.448 3877.0 1163.0 0.25 Fe XI bl Fe X

195.149 3443.0 1033.0 0.25 Fe XII

- **the constant pressure** N_eT $[cm^{-3}K]$ **or the constant density** N_e $[cm^{-3}]$, for which the contribution functions will be calculated, passed to the routine as a keyword.

- the **ionization equilibrium file**, selected using a widget.

- **the elemental abundances file**. A selection of files are already stored in the CHIANTI package, but user-defined files in the working directory can also be used. Any **.abund* file present in the CHIANTI database **or** in the working directory can be selected through a widget from within CHIANTI_DEM. The selected file can also be edited, to change the abundances of any element.

- **An output file name** must also be supplied via a keyword (e.g. OUTPUT= 'active_region'). Various files will be generated by CHIANTI_DEM having file names created by adding suffixes to the output file name.

Once the file with the observed fluxes is read, another IDL procedure, GET_CONTRIBUTIONS, is called by CHIANTI_DEM in order to calculate the contribution functions $C(T, \lambda_{ij}, N_e)$ at the given constant density or pressure.
GET_CONTRIBUTIONS searches the CHIANTI database for all the lines that may have contributed to the observed lines, i.e. that have a CHIANTI wavelength λ_{theo} in a $\lambda_{obs} \pm \delta\lambda$ interval.
Then, for each theoretical line selected, it calculates the $C(T)$ values for the temperature grid of the ionization equilibrium file. **Note: for older files, log(T)= 4.0 - 8.0 in steps of log(T) = 0.1, while the most recent files have steps of log(T) = 0.05.** .

If a constant pressure is selected, for each ion the contribution function is calculated at an electron density N_e equal to the ratio of the pressure and the temperature of maximum ionization fraction.
The $C(T)$ values are stored by GET_CONTRIBUTIONS in an ascii output file *output.contributions* which can be used later, if required, to re-calculate the DEM, changing various parameters (e.g. the abundances), without having to recalculate the ontribution functions.
test.contributions:
The first three lines contain the abundance file, the ionization equilibrium file names, and the constant value of the pressure or the density adopted. Each subsequent line contains the observed wavelength λ_{obs}, the theoretical one λ_{theo}, the element and ionization stage, the $C(T)$ values and the specification of the transition.
The observed lines with no CHIANTI counterparts are automatically excluded. If this happens, you might consider starting again with a larger $\delta\lambda$, to see if there are theoretical lines in the vicinity of the observed one.
Then you are asked to select an *.abund file present in the CHIANTI database or in the working directory, and eventually edit it, if you want to change some abundances.
The $G(T)$ are calculated, multiplying each theoretical line by the abundance factor. The theoretical lines contributing to each blend are sorted by intensity and then their G(T) can be plotted if the keyword PLOT_GT was activated. It is recommended to do this the first time, to check if there are some observed lines which are heavily blended with lines of other elements. It might be better to exclude such lines in a second run.
The $G(T)$ for each blend are then summed and plotted, and the calculation of the DEM starts. We have several options, listed below.
The action of CHIANTI_DEM is controlled via the following keywords.

- FILE_INPUT: optional; if not set, you are prompted to select the observation file using a widget-type search.
- ARCSEC: optional set this if the intensities are specified in units per arcsec^{-2}.

  The default units are $ergs\ cm^{-2}s^{-1}sr^{-1}$.
- PHOT: optional; set this if the intensities are specified in units per steradians^{-1}.

  The default units are $ergs\ cm^{-2}s^{-1}sr^{-1}$.
- OUTPUT : required; the name for the output. Suffixes will be added to this name when creating the various outputs.
- FILE_GT: if **not** set, the routine GET_CONTRIBUTIONS is called. **Either** the pressure or the density must be set in this case.

  If set, it has to specify the name of the file previously created by GET_CONTRIBUTIONS, where all the contribution functions $C(T)$ are stored.
- PRESSURE: the value of the pressure (Ne T). Required if you do **not** already have the contribution functions C(T) (i.e. if you **do not** set FILE_GT). **Either** the pressure or the density must be set in this case.

- DENSITY : the value of the electron density (Ne). Required if you do NOT already have the contribution functions C(T). **Either** the pressure or the density must be set in this case.

- CUT_GT: optional; if set, only those theoretical lines that have a $MAX(C(T))$ greater than the value set, are kept; it is useful to set this value in order to reduce the number of lines in the file where the $C(T)$ are stored. If not set, a default value of 10^{-30} is adopted.

- N_MATCHES: optional; in the unlikely event that more than 100 (default value for N_MATCHES) theoretical lines corresponding to an observed line are found, the routine stops. In this case, you have to start again setting N_MATCHES equal to a greater number.

- PLOT_GT: optional; if set, plots of the $G(T)$ of the theoretical lines contributing to each observed line not excluded are created. It is possible to change the scale and create postscript files of these plots, interactively.

- EXCLUDE_OBS_WVL: optional; if set, has to be an array that specifies the wavelengths of the lines that you want to exclude from the fit. Note that even if you set this keyword and run GET_CONTRIBUTIONS all the theoretical lines found corresponding to all the lines in the input file are written in the $C(T)$ file. It is only in the fit that the lines are excluded.

- DEM_FILE: optional; if set you are prompted to choose a DEM file to be used initially, instead of the default constant value of 10^{22}. You can either choose one of the files in the CHIANTI database or any you have in the working directory. A plot of the DEM is created. The values in the file are marked as crosses, the mesh points are marked with triangles.

- QUIET: optional. Set to avoid various messages and the details of the result.

There are also some actions controlled via the keyboard.
When you are asked for an answer (**[y/N]**) *yes* or *no* you should either type in **y** or **n**. The capital letter in **[y/N]** means that the default choice is **n** which is what you get if you simply hit the **return key**. In case you have **[Y/n]**, hitting the **return key** is the same as choosing **y** .

9.1 Default option: MPFIT_DEM

By default, the routine runs the routine MPFIT_DEM. A series of input parameters can be given:

```
MIN_logT:   minimum temperature (log T) for the DEM
             calculation
MAX_logT:   maximum temperature (log T) for the DEM
             calculation
```

```
 DT_logT:      temperature step (log T) for the DEM
              calculation (default=0.1)
```

The G(T) are resampled using the IDL routine INTERPOL onto the chosen temperature grid.
IMPORTANT NOTE: As the line intensities are calculated with a sum, their values depend somewhat on the temperature step. A value of 0.05 in log T (the default) or less is recommended.
Optional (but recommened) input:

```
SPL_LOGT:     logt T values for the spline nodes. Note
              that the number should not be larger than the number of
              input lines -1.

SPL_LOGDEM:   log DEM input values. If not defined, a
              value of 20 will be used.

MIN_LIMITS
MAX_LIMITS:   minimum and maximum limits for
              the log DEM values at the spline temperatures.
```

Note: if the SPL_LOGT values are not given as input, the program will select an equally-spaced number of spline nodes within the given minimum and maximum temperatures.
Note: if no limits are given, by default the routine applies a minimum value for the log DEM=17. Users should apply some limits as sometimes the routine will not return a result.
Vaious outputs are produced.

```
  1) output+'_xrt_dem.dem': the DEM as a CHIANTI ascii file.
  2) output+'.out'         : an ascii file with the predicted intensities
  3) output+'_xrt_dem.save': an IDL save file with:

  input: the input structure used for the fitting
  logT_out: the log T
  log_dem_out: the log DEM
  log_dem_mciter: (optional) the results of the Monte Carlo runs
```

Users can restore the save files and re-run the MPFIT_DEM fitting routine later on independently of CHIANTI_DEM.
EXAMPLE:

```
IDL> chianti_dem,output='test_obs',file_input='test_obs',$
   pressure=1.e15,cut_gt=1e-30,/plot_gt
```

```
IDL> chianti_dem,output='test',file_input='test_obs',$
file_gt='test_obs.contributions', min_logt=5.5, max_logt=6.6
```

test.general: Is the file where general information is stored. The abundance file, the ionization equilibrium file and the pressure used are written at the beginning. Then there is one line for each observed line, with the identification present in the input file, the observed wavelength λ_{obs}, the observed flux I_{obs}, the calculated flux I_{theo} , the error on the flux σ_{obs}, the value $(\frac{I_{theo}-I_{obs}}{\sigma_{obs}})^2$ and finally the value of $\frac{I_{theo}}{I_{obs}}$.
After this line, there is one line for each theoretical line contributing to the blend, with the identification, the theoretical wavelength λ_{theo}, the configuration and terms, and the contribution (as a percentage) of each line in the blend to I_{theo}.
Users can restore the save files and re-run the XRT_DEM_ITER_NOWIDGET later on independently of CHIANTI_DEM.
To plot the results of the XRT_DEM:

```
restore, file='test_xrt_dem.save',/ver

      x_min=min(logT_out)
      x_max=max(logT_out)
      y_min=min(log_dem_out)
      y_max=max(log_dem_out)

     plot,  logT_out, log_dem_out,$
            xr=[x_min,x_max],yr=[y_min,y_max],$
            xstyle=1,xtitle = ' log Teff [ !eo!nK ]',$
            ytitle ='log DEM [ cm!S!E-5 !NK!S!E-1!N ] ',$
            title='XRT DEM INVERSION TECHNIQUE'

; over-plot the  observed/expected ratio * DEM at the effective temperature:

n_obs=n_elements(obs_int)

for iobs=0,n_obs-1 do begin

         point=spline(logT_out, log_dem_out), alog10(t_eff[iobs]))

         oplot, alog10(t_eff[iobs]), alog10(obs_int[iobs]/exp_int[iobs]* 10.^point),

         xyouts, alog10(t_eff[iobs]), alog10(obs_int[iobs]/exp_int[iobs]*$
                                              10.^point[iobs]), $
                ' '+strtrim(obs_id[iobs],2), charsize=0.8, Orientation=90

endfor
```

```
; plot the  observed/expected ratio * DEM at the temperature of the maximum of the G(

      plot,  logT_out, log_dem_out,$
             xr=[x_min,x_max],yr=[y_min,y_max],$
             xstyle=1,xtitle = ' log Tmax [ !eo!nK ]',$
             ytitle ='log DEM [ cm!S!E-5 !NK!S!E-1!N ] ',$
             title='XRT DEM INVERSION TECHNIQUE'

for iobs=0,n_obs-1 do begin

         point[iobs]=spline(logT_out, log_dem_out), temp_max_tot_contr[iobs])

         oplot, [temp_max_tot_contr[iobs]], [alog10(obs_int[iobs]/exp_int[iobs]* 10.^

         xyouts, temp_max_tot_contr[iobs], alog10(obs_int[iobs]/exp_int[iobs]*$
                                             10.^point[iobs]), $
                 ' '+strtrim(obs_id[iobs],2), charsize=0.8, Orientation=90

endfor

; if you have run the  XRT_DEM Monte Carlo simulations and want to plot them
; again, do:

restore, file='test_xrt_dem.save',/ver
sz=size(log_dem_mciter)
MC_iter=sz[2]

plot, logT_out,log_dem_out, psym=10,th=th, col=0,$
       xr=[x_min,x_max],yr=[y_min,y_max],$
             xstyle=1,xtitle = ' log T [ !eo!nK ]',$
             ytitle ='log DEM [ cm!S!E-5 !NK!S!E-1!N ] ',ystyle=1,$
      title='XRT DEM INVERSION TECHNIQUE'

  for ii=0, MC_iter-1 do oplot, logT_out, log_dem_mciter[*,ii], th=th, col=100, psym=
  oplot, logT_out,log_dem_out, psym=10,th=th, col=0
```

9.2 DATA2DEM_REG

As an option, the routine can run DATA2DEM_REG, a routine that recovers the DEM using a GSVD approach, detailed in Hannah & Kontar A&A 539, A146 2012. Users must download the IDL suite of routines, found in `http://www.astro.gla.ac.uk/~iain/demreg/` and add

them to the IDL path. A series of keywords are passed to the code, see the header of DATA2DEM_REG.
A series of input parameters can change the result (DEM), especially the number of temperatures and the temperature range.

```
  INPUT:

  DO_DEMREG: keyword to ask the routine to run
             DATA2DEM_REG.
  DEMREG_LOGT_MIN: minimum temperature (log T) for the
                   DEM calculation
  DEMREG_LOGT_MAX: maximum temperature (log T) for the
                   DEM calculation
  NT_DEMREG: Number of temperatures  for the DEM
             calculation (default=20). Note that this
             number must be larger than the number of
             input lines+1.

  OUTPUT:

   1) output+'_demreg.dem': the DEM as a CHIANTI ascii
   file.

   2) output+'_demreg.save': an IDL save file with
   REG: a structure containing all the input and output
   results of the routine.
```

```
EXAMPLE:

               IDL> chianti_dem,output='test',file_input='test_obs',$
               file_gt='test.contributions',/do_demreg,demreg_logt_min=5.5,$
               demreg_logt_max=6.6, nt_demreg=20
```

Users can restore the save file and re-run DATA2DEM_REG later on independently of CHIANTI_DEM. To plot the regularized DEM and both vertical and horizontal errors:

```
restore, file='test_demreg.save'

     x_min=min(reg.logt)
      x_max=max(reg.logt)
      y_min=2d19
```

```
y_max=2d23

ploterr,reg.logt,reg.dem_pos,reg.elogt_pos,reg.edem_pos,$
        /nohat,errcolor=9,  xr=[x_min,x_max],yr=[y_min,y_max],$
       xstyle=17,ystyle=17,/ylog, title='Regularized DEM', $
        xtitle='log!D10!N T',ytitle='DEM(T) [cm!U-5!N K!U-1!N]'
```

9.3 MCMC_DEM

As an option, the routine can run the the PINTofALE command-line function MCMC_DEM(), which runs a Markov-Chain Monte-Carlo algorithm on a set of line fluxes and returns an estimate of the DEM. Users should download the package and add the routines to the IDL path. For more information, see http://hea-www.harvard.edu/PINTofALE/ Note: the MCMC_DEM() has many keywords. Only some are passed to this routine.

```
  INPUT:

  DO_MCMC: keyword to ask the routine to run MCMC

  MCMC_LOGT_MIN: minimum temperature (log T) for the
                   DEM calculation
  MCMC_LOGT_MAX: maximum temperature (log T) for the
                   DEM calculation
  MCMC_LOGT_STEP:  temperature step (log T) for the DEM
               calculation (default=0.1)

  OUTPUT:

  1) output+'_mcmc.dem': the DEM as a CHIANTI ascii file.

  2) output+'mcmc.save' an IDL save file with all the
  input and output results, keyword parameters, etc.

Example:

 chianti_dem,output='test',file_input='test_obs',$
              file_gt='test.contributions', /do_mcmc, mcmc_logt_step=0.1,$
              mcmc_logt_max=6.6, mcmc_logt_min=5.5
```

Users can restore the save file and re-run MCMC_DEM later on independently of CHIANTI_DEM. To plot the DEM and the observed/expected ratio * DEM at the effective temperature:

```
x_min=min(logt_grid)
     x_max=max(logt_grid)
     y_min=min(alog10(dem_out))
     y_max=max(alog10(dem_out))

plot,  logt_grid, alog10(dem_out),$
               xr=[x_min,x_max],yr=[y_min,y_max],$
               xstyle=1,xtitle = ' log Teff [ !eo!nK ]',$
               ytitle ='log DEM [ cm!S!E-5 !NK!S!E-1!N ] ',$
               title='MCMC DEM INVERSION TECHNIQUE', psym=10


        for iobs=0,n_elements(obs_int)-1 do begin         &$
            point=spline(logt_grid, alog10(dem_out), alog10(t_eff[iobs]))  &$
         oplot, [alog10(t_eff[iobs])], [alog10(obs_int[iobs]/exp_int[iobs]* 10.^point
   xyouts, alog10(t_eff[iobs]), alog10(obs_int[iobs]/exp_int[iobs]*10.^point), $
                  ' '+strtrim(obs_id[iobs],2), charsize=0.8, Orientation=90 &$
end
```

9.4 Examples

You must specify the output file name and the value of the pressure (or the density). The input file name is optional.

```
IDL > CHIANTI_DEM,OUTPUT='test',FILE_INPUT='test_obs',PRESSURE=1e16,/PLOT_GT
```

Select the ionization equilibrium file (e.g. Arnaud & Raymond). If there are no problems about N_MATCHES, the routine will select the lines having $max(C(T)) \geq 10^{-30}$ and write the $C(T)$ values to the file *test.contributions*.
Then you'll be asked to select an abundance file and if you want to edit it. Pick up the Feldman abundances. Then the $G(T)$ are calculated, multiplying each theoretical line by the abundance factor, sorted (within each blend) by their $max(G(T))$ value, and plotted (see Fig. 10).
It is recommended that you check the plots at least once, to see if there are some observed lines that it might be better to exclude in a second run, for example because they are blends. Also check if your identifications are consistent with the lines found in the CHIANTI database.
The G(T) for each blend are then summed, and plotted (see Fig. 11).
At the end of the fit, the files *test.dem, test.general* are created.
Have a close look at these outputs, and check if there are emission lines not well represented by the fit or with no theoretical counterparts.
You can use the routine a second time, excluding some of the lines, and/or varying some of the fitting parameters.

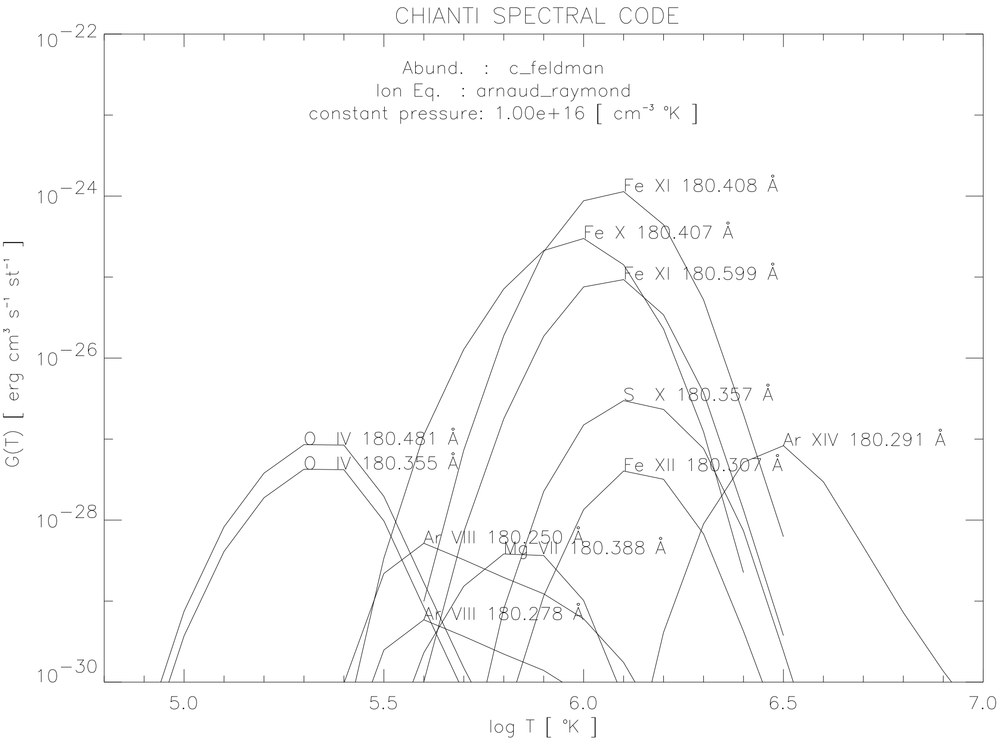


Figure 10: One of the G(T) plot of the test case

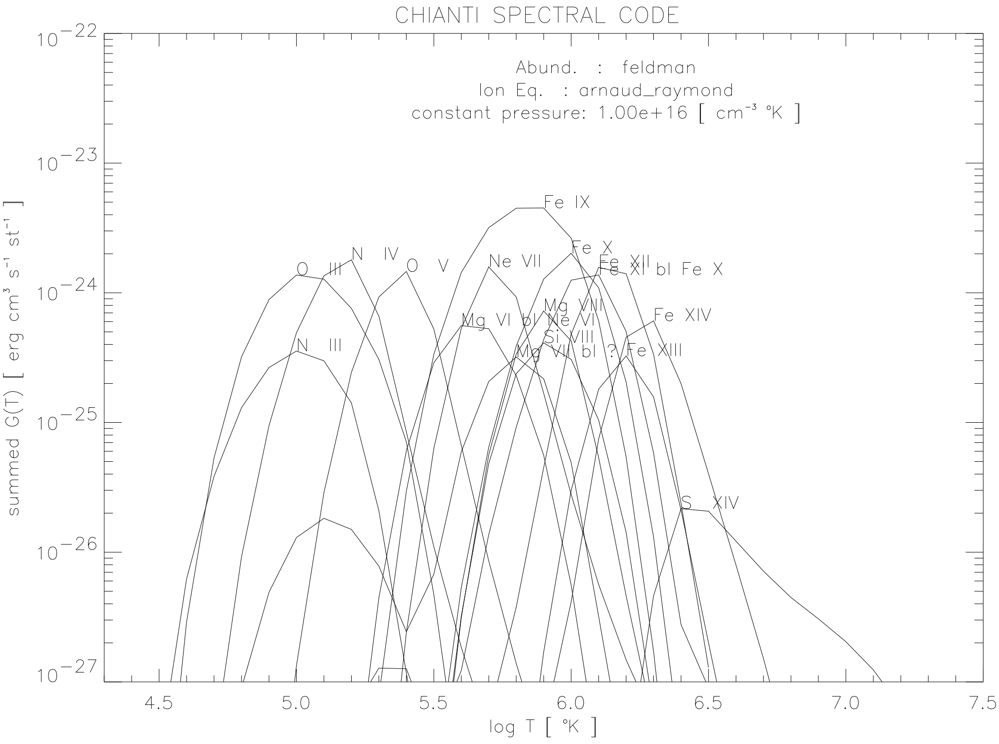


Figure 11: The summed-G(T) plot of the test case: *test_gt.ps*

9.5 Some final remarks

This package is mostly intended to be a quick method to obtain a DEM which can then be used to calculate a synthetic spectrum, to be compared with the observed data.
Try to give as input lines covering a broad range in temperatures, and that are not density sensitive.
If the resulting DEM does not give a good fit to the data, it might be a good idea to start again calculating the $G(T)$ with different abundances or to check the effect of blends.
Try fifferent end points at lower and higher temperatures, where usually there are no constraints (no observed lines).
Consider the possible effect on the DEM of different structures along the line of sight. It is important to realise that the DEM gives an indication of the amount of plasma at different temperatures *along the line of sight*, assuming *constant* density or pressure. It is not therefore possible to infer direct information about the variation of the temperature with height from this function. The inclusion of density-sensitive lines in the fit may also cause problems.

10 The CHIANTI line intensities structure

The tags of the line intensities structure are:

```
.lines     A structure containing information about the lines.
           Its size is the number of lines in the spectrum. The
           tags are:

           .iz      The atomic number of the elements (e.g., 26=Fe)

           .ion     The ionisation stage (e.g., 13=XIII)

           .snote   The identification of the ion (e.g., 'Fe XXIV d')

           .ident   The identification of the transition, configuration
                     and terms in text form.

           .ident_latex
                    The identification of the transition, configuration
                     and terms in latex form.

           .lvl1    The lower level of the transition (see .elvlc
                    file for ion)

           .lvl2    The upper level for transition.

           .tmax    The temperature of maximum emission of the line.
```

```
                   If the G(T) are output, tmax is the maximum of G(T).

                   If the isothermal approximation is used  tmax=0.

                   If a DEM is used,  tmax is the maximum of the
                   emissivity that includes the product of the ion
                   fraction and the DEM.
                   Rounded to nearest 0.1

           .wvl    Wavelength of the transition, in Angstroms.

           .flag   A flag, =-1 if the line has only theoretical energy
                   levels. Otherwise flag=0.

           .int    Intensity of line (erg/cm2/s/sr or phot/cm2/s/sr),
                   divided by the element abundance (exclusive with .goft).

           .goft   The G(T) of the line (optional /exclusive with .int).


.ioneq_name     The ion balance file used (full path).
.ioneq_logt        The Log10 T values associated.
.ioneq_ref      The references.

.dem_name       The differential emission measure file eventually  used
                (full path).
.dem            The Log10 DEM values
.dem_logt          The Log10 T values associated.
.dem_ref        The references.

.model_name    A string indicating the model used:

             1- Constant density
             2- Constant pressure
             3- Function (Te,Ne)

.model_file    Full path of the (Te,Ne) file if defined. Null string otherwise

.model_ne    the Ne value(s).

              - a scalar if 'Constant density' is selected.
              - an array if 'Function' is selected.
              - 0. if constant pressure is selected.
```

```
.model_pe    the Pe value.

              - a scalar if constant pressure is selected.
              - 0. if 'Constant density' is selected.
              - an array=density*temperature if 'Function' is selected.

.model_te    the Te values if 'Function' is selected. Otherwise 0.

.wvl_units  The wavelength units.

.wvl_limits    The wavelength limits specified by the user.

.int_units  The intensity units (erg/cm2/s/sr or phot/cm2/s/sr) if
           intensities are calculated, otherwise the G(T) units
           (erg cm3/s/sr or phot cm3 /s/sr)

.logt_isothermal
           The Log10(T) values used.

.logem_isothermal
           The Log10(EM) values used.

.date      The date and time when the structure was created.

.version   The version number of the CHIANTI database used.

.add_protons
           A flag (0/1) to indicate whether proton data were used (1)
           or not (0) to calculate the level population.

.photoexcitation
           A flag (0/1) to indicate if photoexcitation was included (1)
           or not (0).

.radtemp
          The blackbody radiation field temperature used (if
          photoexcitation was included).

.rphot
       Distance from the centre of the star in stellar radius units
       (if photoexcitation was included).
```

11 The CHIANTI spectrum structure

The spectrum structure output of MAKE_CHIANTI_SPEC has the following ADDITIONAL tags (compared to the tags of the CHIANTI line intensities structure created by CH_SYNTHETIC):

```
 LAMBDA      The array of X-values
 SPECTRUM    The array of Y-values
 UNITS       The units of LAMBDA, SPECTRUM
 WRANGE      The wavelength range
 INSTR_FWHM  The Instrumental FWHM
 BIN_SIZE    Width of the Bins  (fixed) in angstroms
 ABUND_NAME  The CHIANTI abundance file name
 ABUND       The abundance values
 MIN_ABUND   The minimum abundance value used
 ABUND_REF   The references
 CONTINUUM   The values of the continuum (if calculated)
 FILE_EFFAREA The Effective Area File used (optional)
 EFFAREA       The array of effective area values
               (optional - same size of LAMBDA)

.LINES      An array of structures, for all the lines used
            to calculate the SPECTRUM.
            The tags are the same as those created by
            CH_SYNTHETIC, plus
   .PEAK    The peak intensity of the line in the spectrum
            (approx. value)
```

12 References

References

[1] N. R. Badnell. Radiative Recombination Data for Modeling Dynamic Finite-Density Plasmas. *ApJS*, 167:334–342, December 2006.

[2] G. Del Zanna and H. E. Mason. Xuv spectroscopy. *Living Reviews in Solar Physics*, 15, 2018.

[3] K. P. Dere. Ionization rate coefficients for the elements hydrogen through zinc. *A&A*, 466:771–792, May 2007.

[4] M. F. Gu. Indirect X-Ray Line-Formation Processes in Iron L-Shell Ions. *ApJ*, 582:1241–1250, January 2003.

[5] C. Jordan and R. Wilson. *ASSL Vol. 27: Physics of the Solar Corona*, page 219, 1971.

[6] E. Landi, G. Del Zanna, P. R. Young, K. P. Dere, H. E. Mason, and M. Landini. CHIANTI-An Atomic Database for Emission Lines. VII. New Data for X-Rays and Other Improvements. *ApJS*, 162:261–280, January 2006.

[7] S.R̃. Pottasch. On the interpretation of the solar ultraviolet emission line spectrum. *Space Sci. Rev.*, 3:816, 1964.