

# Chapter 3

## New Developments for Data Handling and Production within ADAS

### 3.1 Methods for Handling Data Entry

#### 3.1.1 Bundling and Unbundling of Levels and Parents

In recent years the volume of fundamental atomic data output from large collision cross-section and atomic structure calculations has increased the requirements for computer storage space while placing a premium on methods for easing its use and also for allowing a physical focus to be maintained. For example, ADAS data format files *adf09* contain state selective dielectronic recombination coefficients for all terms up to principal quantum shell 7 supplemented with n-shell totals up to principal quantum shell  $n=1000$  (see for example Badnell et al.(1993)). Given that the main interest in a particular study may well be in transitions involving highly populated low lying metastable states it seems sensible to provide some way of focusing attention on the transition in question, but at the same time, attempting to maintain the effects of the more sophisticated treatment of highly excited levels provided by these calculations.

The *condensation* methods of *collisional-radiative* theory, see for example Summers (1994), Summers & Hooper (1983), Dickson (1993) and sec.2.2.2, allow computation of metastable level populations by projecting on to them the influence of quasi-static equilibrium calculations of highly excited levels. As shown in chapter 2, and by the above authors, this technique can be generalised to arbitrary high and low level sets which, in effect, are bundled together. This allows compression of large data collections to more manageable sizes. Levels can be bundled in different ways. For example, grouping together according to the recombining parent ion on which they were built, by the spin system to which they belong or by levels of the same term. These techniques can be beneficial to users in handling large quantities of data when studying the excited populations of complex ions.

However, bundling without a physical basis is unrealistic and any method must pay great attention to the atomic processes contributing to spectral emission. Summers & Dickson (1992) pointed out the conflicting requirements of a full theoretical description of recombination in plasmas. Evidently, as is obvious from the example above, very high n-shells can contribute to the total effective recombination coefficient. This is principally due to dielectronic recombination and its efficiency in populating very highly excited states. At such high quantum shells the effects of fields and other particles cannot be ignored (Burgess & Summers (1969)). A proper theoretical treatment of dielectronic recombination therefore requires detailed calculation for these highly excited levels. When dealing with the population structure of an ion these results can only be included using many level handling techniques such as *'bundle-n'* and *'matrix condensation'* (Summers (1974)). However, sophisticated treatment of emitting states is essential for spectroscopic studies and these transitions generally involve low lying levels labelled by an explicit coupling scheme. Here, *'bundle-n'* is not appropriate but it may be useful to perform some grouping of minor populations especially if they have little influence on the spectra being studied.

The key issue here is awareness of the physical model. Whether the study is of dielectronic recombination coefficients or population distributions of low lying levels any bundling method must maintain consistency between the requirements of the atomic structure models, the data handling and prediction of the dominant emission.

Bundling need not be confined to levels. In ionisation and recombination reactions it is proper to distinguish metastable states as well as ground states as potential initial states for recombination and final states for ionisation. Parent states of the recombining ion, which drive the population structure calculations, should be identified and included explicitly in any ionisation balance. These parents could be bundled together in order to focus attention on the parent ground state. This might be useful if there was a wish to ignore the complexities of metastables, or if full account was being taken of J-resolved metastable components in the input dataset but LS term resolution was required.

Evidently during transient events in dynamic plasmas the dominant emission from a heavy impurity ion, such as molybdenum, is principally that due to the highest ionisation stages of the ion (Badnell et al.(1996)). In such a case it is computationally unnecessary to attempt to model the ground and metastable population distributions for every ionisation stage. It is more appropriate to concentrate on the stages that are observed and to bundle together lower charge states. In addition, it may be beneficial to bundle together terms of the same principle quantum shell which are likely to be ionised simultaneously, for example those which have roughly similar ionisation potentials.

Resolution of the dominant populations is an important consideration. For light ions the metastables are LS terms, whilst for highly ionised ions the excitation energy separations between J components are such that they may be true levels. Our work, developed in the fusion context at JET, takes account of LS resolved metastable states and parents. Consequently derived quantities such as ionisation balance are treated in LS coupling. However, in the astrophysical context J-resolved datasets are sought even for light species. This situation is motivated by the available spectral resolution in the two regimes. The spectral resolution of the Normal Incidence Spectrometer on the CDS instrument is  $\sim 0.08\text{\AA}$  depending on the wavelength (Domingo, Fleck & Poland(1992)). This compares favourably with KT7 the VUV spectrometer at JET which has a spectral resolution of between 0.5 and  $4\text{\AA}$  also dependent on the wavelength (Maggi(1996)). Therefore CDS will actually see J-resolved sub-components of emission lines and they may require appropriate data to assist them.

Generation of such data is another concern. However, resolution of all transitions between J-components does not necessarily imply the need for a full J-resolved treatment of ionisation balance. Some awareness of consistency between fundamental and derived data is desirable here. Generation of LS coupled data collections is often performed by reassembling the J-resolved counterparts. The LS resolved datasets are then consistent with the J-resolved ones and duplication of effort in the atomic collision calculations can be avoided. The reassembling is done by bundling together J-resolved levels of the same term.

Bundling in this way can be of great value, to fusion plasmas, by making use of high quality calculations of effective collision strengths in the J-resolved form which are produced primarily for astrophysical use. Computer codes to calculate these, by transforming the reactance matrices obtained from LS coupling or including relativistic terms in the Hamiltonian before solution of the time independent Schrodinger equation, are now widespread. The first method referred to is that used in the JA-JOM code of Saraph(1972) & Saraph(1978), and the second method is that of Scott & Burke (1980) which is now possible with RMATRIX1 (Berrington, Eissner & Norrington (1995)).

In one direction then the capability exists to take a dataset and alter its resolution to the benefit of another application. However, there are broader applications and issues to be considered. The CDS and SUMER communities wish to have J-resolved data but would it be useful to have an approximate reverse ‘*unbundling*’ routine to take advantage of calculations targeted at JET and the fusion community? If so, how accurate would the technique be compared to other calculations, in producing J-resolved effective collision strengths? How can the bundling methods outlined be extended to coping with J-resolved metastable parents or ionisation stage populations?

Essentially the answer to the first question is yes. The Abingdon atomic data assessment meeting (Lang et al.(1994)) recommended producing J-resolved data for all ions of all species. Clearly this is an enormous computational task. However, ‘fill in’ data, especially if the technique is proven to be fairly accurate, would be very welcome in astrophysical circles. Approximate data accurate to within 30% say

could be used as effectively for spectroscopic diagnostics as **R**-matrix data which normally has comparable errors. Within ADAS ‘top up’ Born approximation data (Summers(1992)) has already proven to be in surprisingly good agreement with **R**-matrix data (Brooks et al.(1997)).

Apart from the ability to react to the observational data from CDS and SUMER, unbundling of terms to produce level data allows us to have a ‘quick look’ at new data and see whether a J-resolved treatment is necessary. Rather than blindly mass produce atomic data for everything, we can make our studies more specifically targeted towards the astrophysics and fusion communities’ needs, and, as a consequence, connect more closely with experimental results.

As for the third question, coping with J-resolved parents when the data are available is not a serious problem. An alteration to the information conventions in the ADAS specific ion file should allow the use of not only J-resolved parents but also arbitrary ones. This has already been implemented in the ADAS codes (for a full description of the specific ion file (adf04) see Summers, Brooks et al.(1996)). A detection routine flags the parent type (LS, LSJ or arbitrary) to the other routines and some minor modifications to the data gathering codes deal with the alteration. A more serious concern is in resolving the initial recombining J-parent or the final ionisation target in the first place.

Bundling together ionisation stages to focus on the more important populations has not been included as yet. However, it does not present a serious computational problem and can also be used to enable ‘unresolved’ ionisation stage population data to be reconstructed from the metastable resolved counterpart.

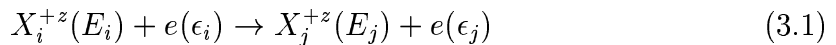
As a first step towards implementing such ideas as data entry utilities within ADAS, two codes have been developed. ADAS209 is a general purpose level bundling program which assists in controlled grouping of effective collision strengths for levels and/or parents (provided parent information is available in the adf04 files). ADAS210 is a general purpose level unbundling program which does the reverse i.e. it assists in splitting effective collision strengths for levels and parents.

Our approach has an advantage over an automatic splitting and grouping routine in that it introduces a level of flexibility which would otherwise be unavailable.

This flexible bundling and unbundling ability allows us to target specific studies by controlled expansion and contraction of data collections and by partially merging resolution levels to the degree we require. In this way we can take advantage of the best available data while reducing the computational load. In addition, bundling and unbundling of the parent metastable states helps us to retain control of the resolution employed in derived calculations and hence to achieve our objective of consistency. As with the level calculations it makes it easier for us to switch between coupling schemes and to ‘work at different resolution levels’.

### Background Theory

Consider again the electron impact excitation reaction defined in chapter 2.



If we now consider a transition  $I$  to  $J$  in LS coupling, then the ‘*LS bundled*’ effective collision strength is a summation over the effective collision strengths between  $J$ -resolved fine structure levels, such that,

$$\Upsilon_{IJ} = \sum_{i \in I} \sum_{j \in J} \Upsilon_{ij} \quad (3.2)$$

Spontaneous emission coefficients (Einstein A-values) corresponding to the new ‘*bundled*’ transitions are given by,

$$A_{J \rightarrow I} = \sum_{i \in I} \sum_{j \in J} \omega_j A_{j \rightarrow i} / \omega_J \quad (3.3)$$

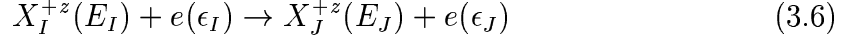
where,

$$\omega_J = \sum_{j \in J} \omega_j \quad (3.4)$$

The new bundled parent energies, relative to the lowest level of the recombining ion, are calculated in the same way as for the excitation energies of each of the excited levels of the  $X^{+z}$  ion, such that,

$$E_{J \rightarrow I} = \sum_{i \in I} \sum_{j \in J} \omega_j E_{j \rightarrow i} / \omega_J - E_{shift} \quad (3.5)$$

where  $E_{shift}$  denotes the relative change in energy between the original and bundled ground states of the  $X^{+z}$  ion. The statistical weights are summed as for the levels above. Now consider the reaction



this time between two excited terms  $I$  and  $J$ , of an ion  $X^{+z}$ , of an element  $X$ . In this case,  $E_I$  ( $E_J$ ) is the excitation energy of level  $I$  ( $J$ ), and  $\epsilon_I$  ( $\epsilon_J$ ) is the incident (scattered) electron energy. If we have high precision calculation data for the effective collision strength associated with this transition, we can obtain an ‘*LSJ resolved*’ effective collision strength corresponding to the fine structure transition,  $i \in I$  to  $j \in J$ , in intermediate coupling. We should note that while a proper representation of the angular momentum of the terms of the ion,  $X^{+z}$ , is possible in LS coupling (for low charge ions) or jj-coupling (for high charge ions), this is not true of intermediate coupling. The Hamiltonian of the system is not nearly diagonal and so off-axis contributions to the eigenvalues must be included to specify the level energies (Cowan (1981)). However, it is still possible to obtain an approximation of the angular momentum characteristics of the J-resolved levels, since the total angular momentum,  $J$ , is still a good quantum number. This can only serve as a label since the true wavefunction will be a mixture of pure coupling functions and it will be assigned by choosing the eigenvector whose contribution dominates. Notwithstanding that, we find LSJ a convenient choice for intermediate coupling in our circumstances, and will use hereafter.

Returning to our problem. We can use two methods to calculate the J-resolved effective collision strength. Firstly, we may already have lower quality data at LSJ resolution. If so, we can bundle this data and then form a ratio with our new data to use as a premultiplier to our lower quality data i.e. we normalise the J proportions of our lower quality data to that of our higher quality data. Therefore,

$$\Upsilon_{ij}^{Jn} = \Upsilon_{ij}^{Jo} \Upsilon_{IJ}^{LS} / \Upsilon_{IJ}^{BLS} \quad (3.7)$$

where  $\Upsilon_{ij}^{Jn}$  denotes the new LSJ effective collision strength,  $\Upsilon_{ij}^{Jo}$  denotes the lower quality LSJ value,  $\Upsilon_{IJ}^{LS}$  is the high quality LS resolved value and  $\Upsilon_{IJ}^{BLS}$  is the bundled

lower quality value and is given by,

$$\Upsilon_{IJ}^{BLS} = \sum_{i \in I} \sum_{j \in J} \Upsilon_{ij}^{Jo} \quad (3.8)$$

The spontaneous emission coefficients (Einstein A-values), can be calculated in a similar manner such that,

$$A_{j \rightarrow i}^{Jn} = A_{j \rightarrow i}^{Jo} (A_{J \rightarrow I}^{LS} / A_{J \rightarrow I}^{BLS}) \quad (3.9)$$

where,

$$A_{J \rightarrow I}^{BLS} = \sum_{i \in I} \sum_{j \in J} A_{j \rightarrow i} \omega_j^{Jo} / \omega_J^{BLS} \quad (3.10)$$

with  $Jo$  and  $Jn$  referring to the lower quality J-resolved data and the new data respectively, and, superscript LS referring to the higher quality LS resolved data. In this case, the statistical weights have been included with similar markings, and

$$\omega_J^{BLS} = \sum_{j \in J} \omega_j^{Jo} \quad (3.11)$$

Secondly, if we do not have any LSJ data for a particular transition, it is appropriate to use a statistical splitting method similar to that of Saraph, Seaton & Shemming (1969). The A-values are zeroed in this case and the upsilons are given by,

$$\begin{aligned} & \Upsilon^{(2S+1)L_J, 2S'+1 L_{J'}} \\ &= \sum_J \sum_{J'} \left( \frac{2J+1}{(2L+1)(2S+1)} \right) \left( \frac{2J'+1}{(2L'+1)(2S'+1)} \right) \Upsilon^{(2S+1)L, 2S'+1 L'} \end{aligned} \quad (3.12)$$

To unbundle the parent metastables we require an additional LSJ resolved dataset for the recombining ion. If available, the parent energies and statistical weights pass directly from it. The energies are always adjusted to account for the energy shift relative to the lowest level of the ion.

### 3.1.2 Overview of ADAS209 (General Level Bundling)

#### Summary of Program and Processing Linkages

ADAS interactive programs make use of an IDL widget based interface to FORTRAN algorithms and subroutines. The schematic of the ADAS209 code (Fig. 3.1) shows

the linkages between the two processes. ADAS209 has a standard three screen structure, file selection, processing of user choices and output of results. These screens correspond to the three boxes down the left of the diagram. Communications between the two processes take place via a bidirectional UNIX pipe set up by IDL as an interface buffer from the standard input/output units of FORTRAN to variable free logical units located by IDL. The IDL procedure spawns the FORTRAN process and then performs repeated checks on its status and on the interactions between them. This is in order to prevent any ‘hanging’ of the program due to *‘unsynchronised communication’*.

After selecting ADAS209 from the main system menu the standard ADAS dataset entry widget appears (see fig.3.2). Here a specific ion file of type adf04 is entered either from the central ADAS database or alternatively from the users’ own space. The data root is displayed in a text widget which becomes editable after selecting *‘Edit Path Name’* with the mouse cursor. Data files and subdirectories contained within the directory entered in this field, are displayed in a data file selection window below. Scroll bars appear if the number of files exceeds the window size. A data file is selected by highlighting with the mouse cursor. The name of the chosen file is displayed in the selection window above, and once it has been read in, the comments appended to the dataset can be viewed by clicking on the *‘Browse Comments’* button with the mouse. All adf04 type files contain information about the data they hold and how, and by whom, it was prepared. ADAS209 itself generates information about the processing options chosen by the user and appends it to the tail of its own output file which is of type adf04 also (see fig.3.6). Clicking on *‘Done’* instructs the code to proceed to the next window.

The next ‘pop-up’ is the *‘Processing Options’* widget (see fig.3.3). This displays information from the selected adf04 file and allows the user to enter their bundling allocations. At the top left of the screen are the processing option switches, level or parent bundling. Below this the adf04 file name is displayed for information followed by a *‘Browse Comments’* button which repeats the facility of the previous window. Below this, the ion and charge and the nuclear charge of the atom, are also displayed. A transcript of the level list obtained directly from the adf04 file, is presented with



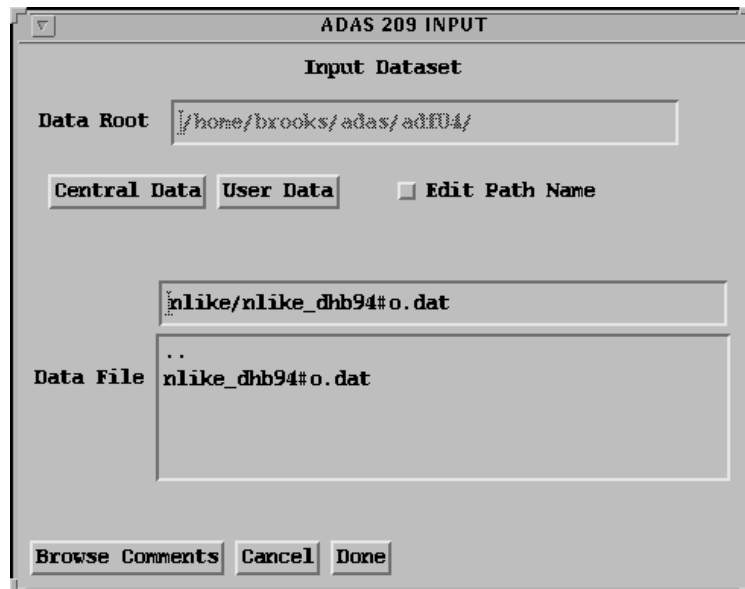


Figure 3.2: ADAS 209 input options - example widget screen. See text for explanation of use.

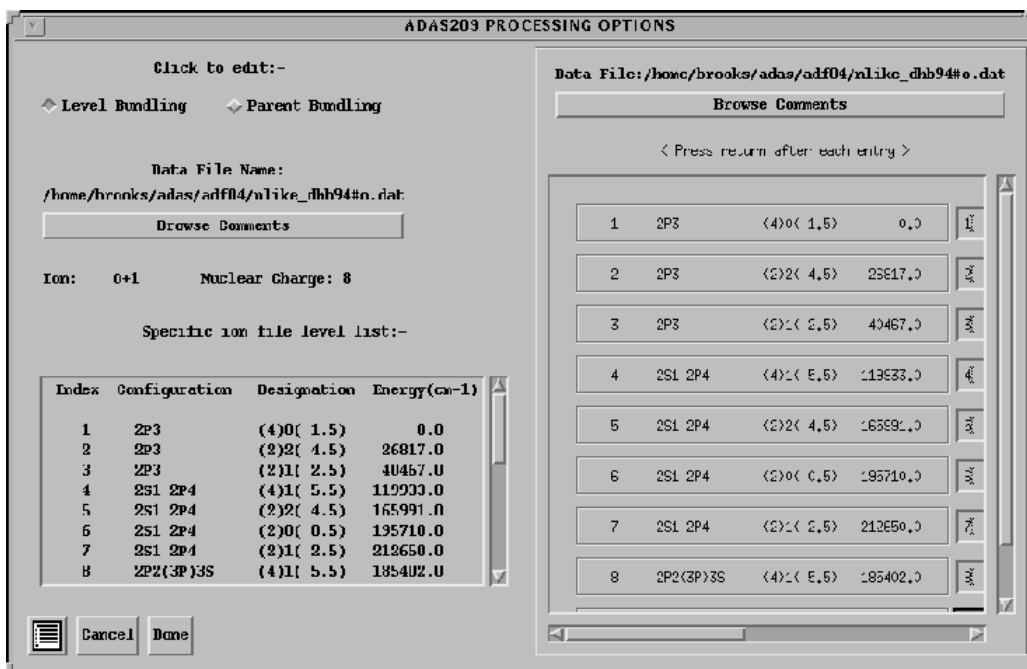


Figure 3.3: ADAS 209 processing options - example widget screen. See text for explanation of use.

a moveable slider, to its right, which allows quick scanning. For ADAS209 the level list is only present for supplementary information. In ADAS210 it is essential to the processing choices. Below this list are the ‘*Cancel*’ and ‘*Done*’ buttons. The bitmap icon to the left of the ‘*Cancel*’ button is an automatic return to ADAS series 2 menu button. On the right hand side of the screen the data file name and ‘*browse comments*’ button are again present. Below them are multiple columns containing the entire level list from the adf04 source file. Each level is displayed with its ordered list index, the configuration, the designation and the level energy. The designation is the usual  $(2S+1)L(J)$  quantum numbers of the Russel-Saunders specification for the level. The energy is displayed in  $cm^{-1}$ . To the right of each level is an editable text entry box. A decision is made here as to which levels to group together and a group index number is entered in each box along side each of the chosen levels. Thus levels to be bundled together are given the same group index. If no entry is made, it implies that the level is to be retained as an individual and it is allocated a default group value of zero.

If the Parent bundling option is chosen, the window changes to a display of information about the recombining ion parent metastables in the dataset. The new window is almost identical in appearance to that which is displayed for the level list(see fig.3.3). However, this time each parent is shown with an index, its term in brackets, its statistical weight and finally its energy relative to the lowest level of the recombined ion (in  $cm^{-1}$ ). Thus, in the case of a single parent it is just the ionisation potential. This information is present on the top line of the adf04 data set from which it is obtained. Such data are present only in the advanced adf04 specification. An example is given in fig.3.6. Up to four parent entries are allowed. There is only one, the default, if the advanced adf04 file specification is not being used. Each parent is displayed with its parent energy and must belong to one of three strictly formatted resolution types. These are ‘*term parents*’ with code ‘ $\langle 2S+1 \rangle L$ ’, ‘*level parents*’ with code ‘ $\langle 2S+1 \rangle L \langle 2J+1 \rangle$ ’ and unidentified parents with code ‘ $\langle statistical\ weight \rangle$ ’. The latter occur when mixed symmetry metastable parents are combined for example when the configuration interaction is strong. The recombining ion parent metastables

can be bundled with no direct influence on the standard adf04 file as it contains transition probability and electron collisional excitation coefficients for transitions only between the low level set of a single ionisation stage. However, in advanced population structure studies it is essential to do this properly in order to assemble effective coefficients for calculating stage to stage quantities correctly. In the advanced specification adf04 files, effective zeta parameters assign proportions of ionisation from each of the levels to each of the identified parents. Practical operation of the widget is as for the level bundling. It is not essential to select '*Parent Bundling*' but if this option is ignored a default ( $^1S$ ) is entered and this may be incorrect. In these circumstances it is advisable to check the top line of the output adf04 file.

**N.B.** It is essential to press return after each entry in order to register selections with the widget event handling routines. Also, grouping together levels with the value zero clashes with the default options and results in no bundling taking place.

After completing the processing options, clicking on '*Done*' proceeds to the *Output Options* window (see fig.3.4). This window simply allows naming and placing of the output file. Once again the source data file name is displayed accompanied by a '*Browse Comments*' button. Below this is an 'activation' button labelled '*File Output*'. Clicking on this button sensitises the other buttons in the widget. Any filename and directory can be entered in the '*File Name :*' editable text box, but the usual ADAS convention is to store the data in a '*bundle.pass*' file. This is the default file which appears on clicking the '*Default File Name*' button. The code attempts to place this file in the directory '/home/user/adas/pass', so users must create a directory '/pass' to enable use of this facility. The '*Replace*' button only comes into effect when the user already has a file of the same name. Activating '*Replace*' overwrites the older file, while not activating it produces a pop-up warning if the file already exists. Clicking on '*Done*' completes the program and returns the user to the ADAS system menu.

The levels/parents are bundled according to the entries given and information about the options chosen is written in the output adf04 file. The adf04 file produced is fully structured and can immediately enter the ADAS database. It follows the advanced adf04 formatting if the source file was of this specification. The code is

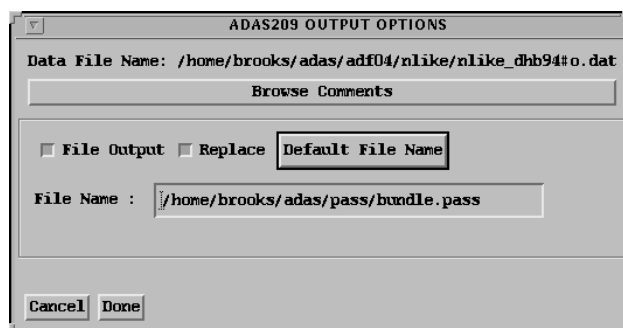


Figure 3.4: ADAS 209 output options - example widget screen

arranged to allow bundling of any levels. However, this can lead to a number of technical complexities. The principal problem occurs if the energies of the levels of a bundle occur on either side of the energy of a target level for spontaneous emission. In these circumstances the output adf04 file should be checked and ADAS209 issues a warning. The code outputs two divisions of the transition with reversed upper and lower indices: firstly, bundled transitions from the target level to lower energy levels of the group; secondly, bundled transitions from higher energy levels, of the group, to the target level. Due to the symmetry of  $\Upsilon$  these can simply be added, but the spontaneous emission coefficients will be incorrect in the latter case. This is due to wrong assignment of the ‘true’ bundled upper statistical weight. Therefore, these should be deleted.

## Example of Use

As an example, consider the LSJ resolved dataset for *Be-like*  $O^{+4}$  shown in fig.3.5. The file contains effective collision strengths calculated from the fit parameters to **R**-matrix data provided by Kato, Lang & Berrington (1990). The object in this example was to produce an LS resolved dataset using ADAS209. Therefore, all LSJ levels have been bundled together to form LS terms e.g.  $2s2p^3P_{0,1,2} \rightarrow 2s2p^3P$ . The output file is shown in fig.3.6.

The convention is to preserve the configuration label of the lowest level contributing to the bundle. In addition, a \* is added at the start of the designation to indicate this. At the tail of each output file the original indexing of levels is shown and below this the grouping selections are displayed. For parent metastables, the original terms are given along with the selection vector. Information about the program, the source file, and the date that the file was produced are also given. Since, as in this example, we would normally be bundling a J-resolved file to produce an LS resolved one, a header is printed with this additional information. The producer id is generally entered by hand.

Note that the parent metastable on the top line is ( $^1S$ ). This is the default which is entered if no further information is available and is incorrect in this case. It is essential to check the parent configurations in an output dataset when the full parent options in the processing codes ADAS209 or ADAS210 are not used.

A further example is shown in fig.3.7. This shows both input and output datasets. The original file contains **R**-matrix data, in LS coupling, from McLaughlin & Bell (1994) for *N-like*  $O^{+1}$ . There are four parent metastable terms for this ion which are  $1s^22s^22p^2^3P$ ,  $^1D$ ,  $^1S$  and  $1s^22s2p^3^5S$ . The file is laid out according to the advanced adf04 specification. Thus, to the right of each level of the list are final parent indices and effective zeta parameters. The parent indices are the numbers in brackets and refer to the index of the final parent metastable state after ionisation from the level. The adjacent numbers give the proportions of the total electron count available, which go to particular parents. The last part of the transition list of the file contains radiative and dielectronic recombination data with the code 'R' in the first column. These data are bundled consistently with the parents in like manner.

```

O + 4      8      5      918657.
1 2S2(1S0) (1)0( 0.0)      0.0
2 2S2P(3P0) (3)1( 0.0) 81942.5
3 2S2P(3P1) (3)1( 1.0) 82078.6
4 2S2P(3P2) (3)1( 2.0) 82385.3
5 2S2P(1P1) (1)1( 1.0) 158797.7
6 2P2(3P0) (3)1( 0.0) 213462.5
7 2P2(3P1) (3)1( 1.0) 213618.2
8 2P2(3P2) (3)1( 2.0) 213887.0
9 2P2(1D2) (1)2( 2.0) 231721.4
10 2P2(1S0) (1)0( 0.0) 287910.3
11 2S3S(3S1) (3)0( 1.0) 546972.7
12 2S3S(1S0) (1)0( 0.0) 561276.4
13 2S3P(1P1) (1)1( 1.0) 580824.9
14 2S3P(3P0) (3)1( 0.0) 582806.4
15 2S3P(3P1) (3)1( 1.0) 582843.1
16 2S3P(3P2) (3)1( 2.0) 582920.3
17 2S3D(3D1) (3)2( 1.0) 600748.9
18 2S3D(3D2) (3)2( 2.0) 600758.9
19 2S3D(3D3) (3)2( 3.0) 600779.2
20 2S3D(1D2) (1)2( 2.0) 612615.6
-1
5.00 3      2.50+04 5.00+04 1.25+05 2.50+05 5.00+05 1.25+06 2.50+06 5.00+06
2 1 1.00-30 6.79-02 5.99-02 4.99-02 4.23-02 3.40-02 2.27-02 1.52-02 9.18-03
3 1 2.17+03 2.04-01 1.80-01 1.50-01 1.27-01 1.02-01 6.81-02 4.55-02 2.75-02
...
19 17 1.00-30 1.55+00 1.57+00 1.25+00 9.33-01 6.54-01 3.85-01 2.52-01 1.62-01
19 18 1.00-30 3.30+00 3.26+00 2.90+00 2.41+00 1.85+00 1.17+00 7.79-01 4.93-01
-1
-1 -1
C-----
C
C The effective collision strengths are calculated using the fit
...
C this comparison (above) is unaffected.
C
C D.H.Brooks 16.08.95
C-----

```

Figure 3.5: ADAS 209 example input dataset. This is of adf04 type as are fig.3.6 and fig.3.7 and a description of the format is given in Summers et al.(1996)

```

O+ 4 8 5 918657.0(1S)
1 2S2(1S0) *(1)0( 0.0) 0.0 {1}1.000
2 2S2P(3P0) *(3)1( 4.0) 82233.9 {1}1.000
3 2S2P(1P1) *(1)1( 1.0) 158797.7 {1}1.000
4 2P2(3P0) *(3)1( 4.0) 213750.2 {1}1.000
5 2P2(1D2) *(1)2( 2.0) 231721.4 {1}1.000
6 2P2(1S0) *(1)0( 0.0) 287910.3 {1}1.000
7 2S3S(3S1) *(3)0( 1.0) 546972.7 {1}1.000
8 2S3S(1S0) *(1)0( 0.0) 561276.4 {1}1.000
9 2S3P(1P1) *(1)1( 1.0) 580824.9 {1}1.000
10 2S3P(3P0) *(3)1( 4.0) 582881.9 {1}1.000
11 2S3D(3D1) *(3)2( 7.0) 600766.4 {1}1.000
12 2S3D(1D2) *(1)2( 2.0) 612615.6 {1}1.000
-1
5.00 3 2.50+04 5.00+04 1.25+05 2.50+05 5.00+05 1.25+06 2.50+06 5.00+06
2 1 7.23+02 6.12-01 5.39-01 4.49-01 3.80-01 3.06-01 2.05-01 1.37-01 8.26-02
3 1 2.86+09 2.63+00 2.67+00 2.81+00 3.02+00 3.36+00 4.03+00 4.68+00 5.44+00

11 4 3.00-30 2.13+00 1.85+00 1.33+00 9.09-01 5.53-01 2.55-01 1.38-01 7.50-02
11 10 3.00-30 5.25+01 5.70+01 8.07+01 1.06+02 1.32+02 1.61+02 1.77+02 1.89+02
-1
-1 -1
C-----
C
C File generated by compression of a J-resolved file
C
C Program: ADAS209
C
C Source file: /export/home/adas/adas/adf04/belike/belike_dhbj95#o.dat
C
C Original level indexing:
C 1 2 3 4 5 6 7 8 9 10
C 11 12 13 14 15 16 17 18 19 20
C Selection Vector:
C 1 2 2 2 3 4 4 4 5 6
C 7 8 9 10 10 10 11 11 11 12
C
C Original parent metastables:
C (1S)
C Parent bundling vector:
C 0
C Insert producer id here: D.H.Brooks
C Date: 19/03/96
C-----

```

Figure 3.6: ADAS 209 example output dataset

```

O+ 1 8 2 283244.0(3P) 303310.(1D) 326222.(1S) 343357.(5S)
 1 2P3 (4)0( 1.5) 0.0 {1}3.00 {4}1.250
 2 2P3 (2)2( 4.5) 26817.0 {1}2.250 {2}0.622 {3}0.128
 3 2P3 (2)1( 2.5) 40467.0 {1}2.250 {2}0.622 {3}0.128
 4 2S1 2P4 (4)1( 5.5) 119933.0 {1}1.500 {4}2.500
...
23 2P2(3P)3D (2)3( 6.5) 232889.0 {1}1.00
-1
2.00 3 8.00+03 2.00+04 4.00+04 8.00+04 2.00+05 4.00+05 8.00+05
2 1 9.70-05 1.37+00 1.40+00 1.46+00 1.53+00 1.58+00 1.51+00 1.27+00
3 1 4.80-02 4.10-01 4.31-01 4.61-01 4.97-01 5.28-01 5.05-01 4.15-01
...
10 7 1.00-30 4.44-03 5.19-03 5.65-03 6.02-03 6.27-03 6.17-03 5.69-03
R 1 +1 5.58-13 3.99-13 2.59-13 1.90-13 1.40-13 9.13-14 6.30-14
...
R 21 +4 1.00-30 1.00-30 1.00-30 1.00-30 1.00-30 1.00-30 1.00-30
-1
-1 -1
C-----
C LS-resolved effective collision strengths for OII
C-----
C This file contains new effective collision strength data for all transitions ...
C ...Radiative & Dielectronic recombination data from wjd92#o.dat unchanged.
C DH Brooks 22.11.94
C-----

```

```

O+ 1 8 2 305437.7(3P)
 1 2P3 (4)0( 1.5) 0.0 {1}4.250
 2 2P3 (2)2( 4.5) 26817.0 {1}3.000
...
22 2P2(3P)3D (2)2( 4.5) 234434.0 {1}1.000
23 2P2(3P)3D (2)3( 6.5) 232889.0 {1}1.000
-1
2.00 3 8.00+03 2.00+04 4.00+04 8.00+04 2.00+05 4.00+05 8.00+05
2 1 9.70-05 1.37+00 1.40+00 1.46+00 1.53+00 1.58+00 1.51+00 1.27+00
...
10 7 1.00-30 4.44-03 5.19-03 5.65-03 6.02-03 6.27-03 6.17-03 5.69-03
-1
-1 -1
C-----
C File generated by compression of a J-resolved file
C Program: ADAS209
C Source file: /home/brooks/adas/adf04/nlike/nlike_dhb94#o.dat
C Original level indexing:
C 1 2 3 4 5 6 7 8 9 10
C 11 12 13 14 15 16 17 18 19 20
C 21 22 23
C Selection Vector:
C 0 0 0 0 0 0 0 0 0 0
C 0 0 0 0 0 0 0 0 0 0
C 0 0 0
C Original parent metastables:
C (3P) (1D) (1S) (5S)
C Parent bundling vector:
C 1 1 1 1
C Insert producer id here:D.H.Brooks
C Date: 19/03/96
C-----

```

Figure 3.7: Output example of bundled parents

In this case, the aim was to show parent bundling, therefore, all the parents have been grouped together to obtain only an effective ground state,  $^3P$ . The first line of the output file contains the resolution type label discussed above. This will include a weight parameter for LSJ and arbitrary resolution types. However, in this case the parent metastables are LS terms. The labelling convention for bundling parents is to preserve the ‘resolution type’. Therefore, in this example, since the aim was to group together all parent metastables to obtain only an effective ground state, the output bundled parent label is an LS term  $^3P$ . The effective ionisation potential of the bundled parents is given for all resolution types. Finally, the selection details are presented at the tail of the file, as above.

### 3.1.3 Overview of ADAS210 (General Level Unbundling)

#### Summary of Program and Processing Linkages

A schematic of the linkages between the FORTRAN and IDL processes, for the ADAS210 code, is shown in fig. 3.8. ADAS210 presents a very similar appearance to ADAS209. It has the standard ADAS three screen structure, file selections, processing of user selections and output of results. However, there are additional requirements for this code. Since a template dataset is required to provide the approximate J-resolved split up fractions, there are two input datasets, namely, the one to unbundle, and the template. In addition, to unbundle the parent metastables, a further J-resolved template dataset relating to the parent ion is required. A facility within ADAS210 seeks to locate this automatically. The code checks the user’s own space and if no data is available the central ADAS database. For example, if the template file *copss#be/copss#be\_ss#o4j.dat* is selected then the code searches for *copss#li/copss#li\_ss#o5j.dat*. It first looks in */home/<username>/adas/adf04* and afterwards in */home/adas/adas/adf04/*.

In the diagram, the three boxes of solid outline to the left correspond to the three screens. The dashed box on the left represents the search mechanism. It is possible to omit the parent template file, provided no attempt is subsequently made to unbundle the parents. The dashed box on the right represents the optional reading of this file.

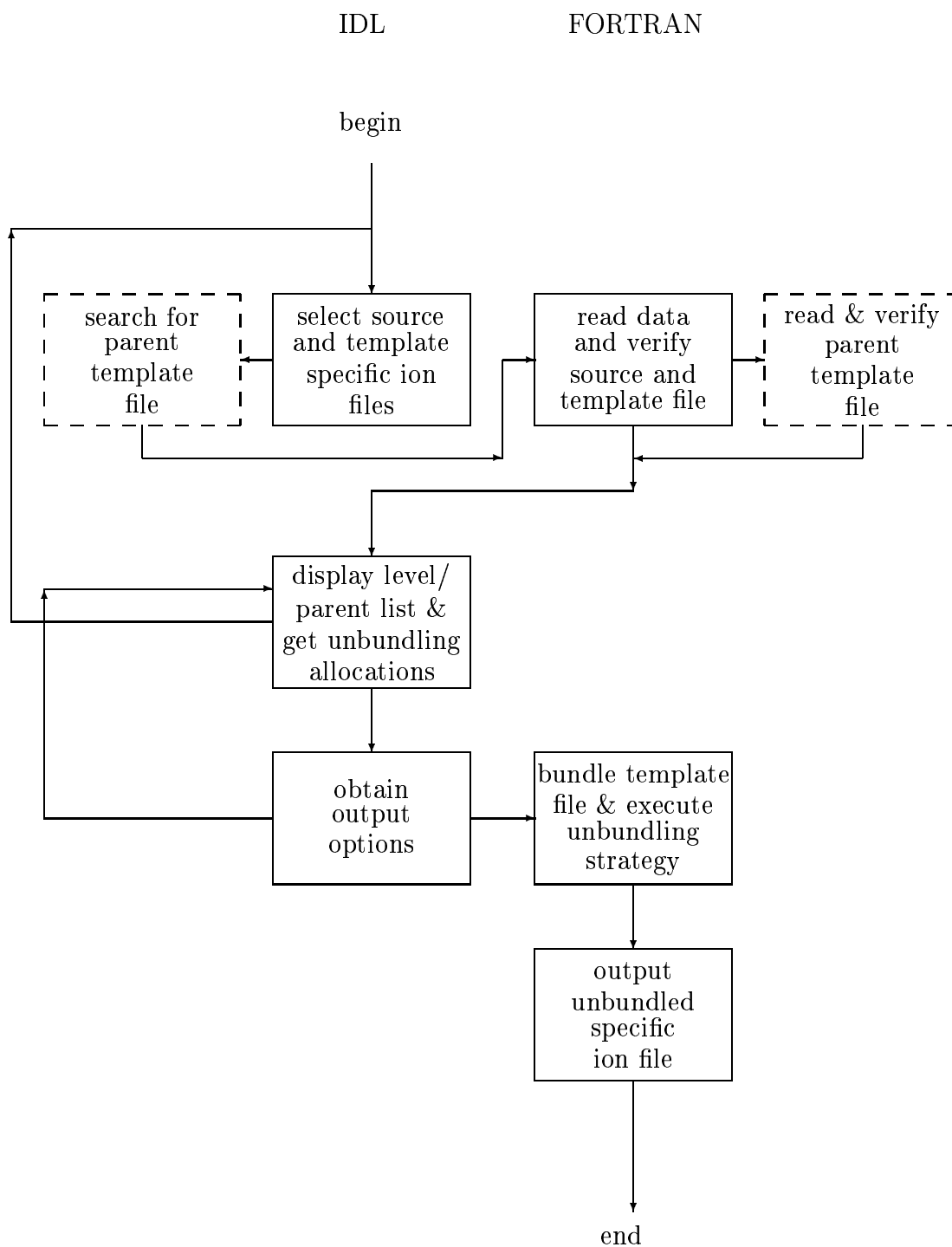


Figure 3.8: Schematic of ADAS210 program and processing linkages. The left hand boxes correspond to the IDL user interface windows. The right hand boxes correspond to the FORTRAN processes. The dashed boxes represent the parent template search.

Since we require to select two datasets, the input options screen for ADAS210 contains two data entry widgets. Each follows the standard individual widget used in ADAS209, except that a *'browse comments'* button appears for each dataset (see fig.3.9).

The upper widget is used for entry of the specific ion file to be unbundled. Access to the central ADAS database or individual user spaces follows the same operation as described for ADAS209. The *'Browse Comments'* button below the selection list refers to this dataset. The lower widget is used for entering the J-resolved dataset to be used as the template for splitting the file. Source file collections for this normally come from mass produced high resolution SUPERSTRUCTURE files, hence the information line *'Select Superstructure file for Template'*. Such files are present in the central adf04 directory in subdirectories according to isoelectronic sequence (e.g. *copss#b*). Intermediate coupling files have the termination *'j.dat'*. It is not essential to use files produced by the SUPERSTRUCTURE code but the file naming conventions must be strictly adhered to for the search mechanism to locate the parent template correctly. After selecting the files, clicking on *'Done'* proceeds to start the automatic parent template search and if the parent file is located the program proceeds without interruption. A 'pop-up' warning will appear if the search fails, but it is possible to continue regardless.

The *'Processing Options'* widget is similar in appearance to that of ADAS209. The options are to unbundle the levels or the parents. The same information as in ADAS209 is displayed in the left side window. The *'Browse Comments'* button here refers to the file the user wishes to unbundle, as does the level list below it. The right side window is different from ADAS209. Firstly, the *'Browse Comments'* button now refers to the template file. In the case of level unbundling, the multiple columns are filled with the level information from the template file. As with ADAS209, group indices are entered in the editable text boxes next to each level. In this case the index of the level to unbundle is entered in the boxes adjacent to each of the J-resolved levels to be used for the split up fractions (see example screen). No entry implies that the level is to be left alone. Fig.3.10 is an example of the parent unbundling screen,

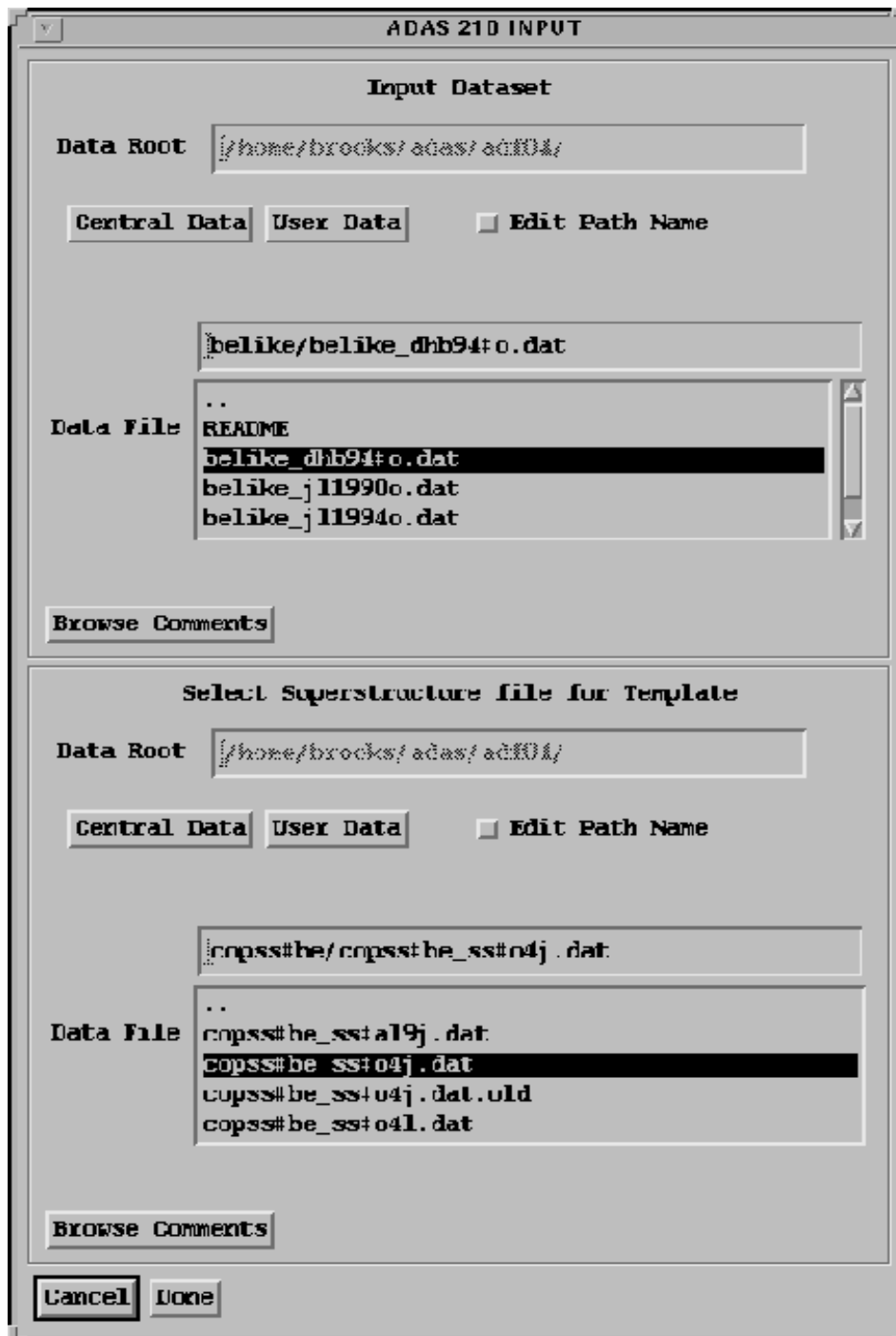


Figure 3.9: ADAS 210 input options - example widget screen. See text for explanation of use.

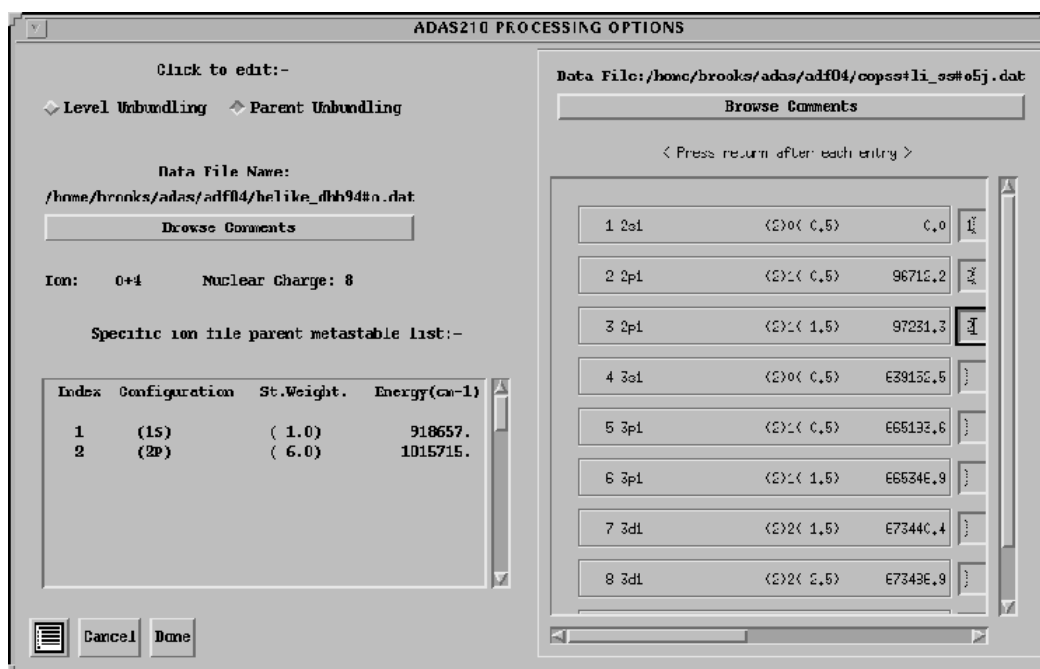


Figure 3.10: ADAS 210 processing options - example widget screen. See text for explanation of use.

and the difference applies here too. The multiple columns contain the level information from the parent template dataset rather than the input file, as was the case in ADAS209. The example screen also shows that the parent level list is transcribed directly from the parent template file. Therefore, the information presented is exactly the same as in the level list i.e. index, configuration, designation and energy. **N.B.** Expanding the number of parent metastables to greater than five is not supported by the ADAS structure. If the user attempts to do this a ‘pop-up’ warning will appear.

The final screen displayed is for entry of the ‘*Output Options*’. This widget is the same as in ADAS209 (see fig.3.4). There is no graphical display so only naming and placing of the output file is required. In this case the default output file name is *unbundle.pass*. The code attempts to place this in the directory */home/<username>/adas/pass* so it is essential to create this directory in advance.

The levels/parents are unbundled according to the processing entries and information about the options chosen is written in the output adf04 file. This file is suitable for entry to other ADAS routines such as ADAS205 etc, and is of the advanced formatting type if the source adf04 file was too. **N.B.** In some cases it may appear that transitions have disappeared from the file on output. In fact, the SUPERSTRUCTURE code does not contain forbidden transitions, such as  $J = 0 \rightarrow 0$  or  $\Delta J = 2$ , with very small cross-sections. ADAS210 deliberately splits terms according to the data available in the high resolution file so, in this case, it omits these transitions.

### Examples of Use

The example in fig.3.11 shows unbundling of an LS resolved dataset for *Be-like O<sup>+4</sup>* to produce an LSJ resolved ‘*unbundle.pass*’ file. The input file was created by bundling an LSJ resolved one. In fact it is the output file from the first *Example of Use* for ADAS209 (see fig.3.6). Notice the mix of configuration types in the level list. Some are preserved from the input file e.g.  $2s^2 \ ^1S$  and others are transcribed from the template dataset e.g.  $2s12p1$ . The latter type are the ones that have been unbundled. At the tail of the file, information is given about the selections made. The *Source file* is the input dataset and the *Template file* is the high resolution SUPERSTRUCTURE dataset. The *Template File indexing* and the *Source File Assignment* vectors provide

```

O +4      8      5 918657.0(1S)
 1 2S2(1S0) (1)0( 0.0) 0.0
 2 2s1 2p1 (3)1( 0.0) 82544.5
 3 2s1 2p1 (3)1( 1.0) 82698.1
 4 2s1 2p1 (3)1( 2.0) 83007.2
 5 2S2P(1P1) (1)1( 1.0) 158797.7
 6 2p2 (3)1( 0.0) 215600.0
 7 2p2 (3)1( 1.0) 215755.1
 8 2p2 (3)1( 2.0) 216059.3
 9 2P2(1D2) (1)2( 2.0) 231721.4
10 2P2(1S0) (1)0( 0.0) 287910.3
11 2S3S(3S1) (3)0( 1.0) 546972.7
12 2S3S(1S0) (1)0( 0.0) 561276.4
13 2S3P(1P1) (1)1( 1.0) 580824.9
14 2s1 3p1 (3)1( 0.0) 582531.6
15 2s1 3p1 (3)1( 1.0) 582569.6
16 2s1 3p1 (3)1( 2.0) 582640.4
17 2s1 3d1 (3)2( 1.0) 601247.4
18 2s1 3d1 (3)2( 2.0) 601262.9
19 2s1 3d1 (3)2( 3.0) 601286.1
20 2S3D(1D2) (1)2( 2.0) 612615.6
-1
5.00 3      2.50+04 5.00+04 1.25+05 2.50+05 5.00+05 1.25+06 2.50+06 5.00+06
 3 1 2.17+03 6.12-01 5.39-01 4.49-01 3.80-01 3.06-01 2.05-01 1.37-01 8.26-02
 5 1 2.86+09 2.63+00 2.67+00 2.81+00 3.02+00 3.36+00 4.03+00 4.68+00 5.44+00
...
18 15 2.26-30 1.31+01 1.42+01 2.01+01 2.65+01 3.30+01 4.02+01 4.42+01 4.72+01
18 16 7.44-31 4.38+00 4.75+00 6.74+00 8.84+00 1.10+01 1.34+01 1.47+01 1.57+01
19 16 3.00-30 2.45+01 2.67+01 3.77+01 4.95+01 6.17+01 7.52+01 8.28+01 8.83+01
-1
-1 -1
C-----
C File generated by expansion of an LS-resolved file
C Program: ADAS210
C
C Source file: /disk2/brooks/adas/adf04/belike/belike_dhbl95#o.dat
C Template file:
C /disk2/brooks/adas/adf04/copss#be/copss#be_ss#o4j.dat
C
C Template File indexing:
C 1 2 3 4 5 6 7 8 9 10
C 11 12 13 14 15 16 17 18 19 20
C 21 22 23 24 25 26 27 28 29 30
C 31 32 33 34 35 36 37 38 39 40
C 41 42 43 44 45 46
C Source File Assignment:
C 0 2 2 2 0 4 4 4 0 0
C 0 0 0 10 10 10 11 11 11 0
C 0 0 0 0 0 0 0 0 0 0
C 0 0 0 0 0 0 0 0 0 0
C 0 0 0 0 0 0
C Original parent metastables:
C Parent unbundling vector:
C
C Insert producer id here:D.H.Brooks
C Date: 08/02/96
C-----

```

Figure 3.11: ADAS 210 example output dataset. This is of adf04 type as is fig.3.12 and a description of the format is given in Summers et al.(1996)

a complete record of the unbundling selections made. If any selections were made, the *Original parent metastables* and the *Parent unbundling vector* are printed. If an LSJ parent level is selected then the configuration and statistical weight are printed on the top line. Note that in this case the statistical weight is equal to  $2J+1$ . Finally, the identifier of the user should be entered by hand and the current date is printed for completeness.

Next we illustrate the use of ADAS209 and ADAS210 together to vary the resolution levels in a particular study. The object is to identify which transitions dominate. We have used the same dataset for  $O^{+4}$ , at LS resolution, as in the earlier example for ADAS210. In this case, since we might expect the metastable level  $2s2p\ ^3P$  to have a significant influence, we have unbundled it to determine the possible effects of J-resolution. In addition, we have bundled all the levels above  $2p^2\ ^3P$ , to see whether we can avoid treating them in any greater detail.

The layout of the file is the same as before. Note that the bundled level 6 retains the configuration specification of the lowest level and that the bundled statistical weight is now very large (25.0). From the file we can see that electrons captured into level 6 will find it relatively easy to decay by spontaneous emission, or electron collisions, to the levels  $2s2p\ ^3P_1$  and  $2s2p\ ^1P_1$ , from which they have an intercombination and electric dipole pathway to the ground level, respectively. In LS coupling, the levels that are significantly populated are the ground ( $2s^2\ ^1S$ ) and metastable ( $2s2p\ ^3P$ ), but at a first glance, in LSJ we might suggest that, due to their inability to decay to the ground level,  $2s2p\ ^3P_0$  and  $2s2p\ ^3P_2$  are likely to become the dominant levels of the metastable.

Finally, it is interesting to compare the results of the unbundling code ADAS210 with a high quality LSJ resolved calculation. From the ‘Example of use’ in ADAS210, we can compare our `unbundle.pass` file with the original LSJ data. Fig.3.13 shows a comparison between the unbundled effective collision strengths and the original ones for some transitions whose spectrum lines are identified as useful for studies by CDS and SUMER. In most cases they will be used for differential emission measure analysis of the quiet sun but the  $2s2p\ ^3P_1-2s^2\ ^1S_0$  transition is useful for dynamic studies and observations of sunspot oscillations since it is sensitive to the electron

```

O+ 4 8 5 918657.0(1S)
1 2S2(1S0) (1) 0(0.0) 0.0 {1}1.000
2 2s1 2p1 (3) 1(0.0) 82544.5 {1}1.000
3 2s1 2p1 (3) 1(1.0) 82698.1 {1}1.000
4 2s1 2p1 (3) 1(2.0) 83007.2 {1}1.000
5 2S2P(1P1) (1) 1(1.0) 158797.7 {1}1.000
6 2P2(3P0) *(3) 1(25.0) 483048.0 {1}1.000
-1
5.00 3 2.50+04 5.00+04 1.25+05 2.50+05 5.00+05 1.25+06 2.50+06 5.00+06
3 1 2.17+03 6.12-01 5.39-01 4.49-01 3.80-01 3.06-01 2.05-01 1.37-01 8.26-02
5 1 2.86+09 2.63+00 2.67+00 2.81+00 3.02+00 3.36+00 4.03+00 4.68+00 5.44+00
6 1 1.72+09 8.78-01 8.46-01 7.90-01 7.52-01 7.44-01 7.86-01 8.57-01 9.49-01
...
6 3 4.63+08 1.77+01 1.75+01 1.75+01 1.85+01 2.00+01 2.30+01 2.54+01 2.76+01
6 4 1.00+09 3.74+00 3.57+00 3.23+00 2.96+00 2.85+00 3.13+00 3.67+00 4.39+00
6 5 4.62+09 1.54+01 1.49+01 1.48+01 1.54+01 1.68+01 1.94+01 2.16+01 2.38+01
-1
-1 -1
-----
C File generated by compression of a J-resolved file
C
C Program: ADAS209
C Source file: /disk2/brooks/adas/pass/unbundle.pass
C
C Original level indexing:
C 1 2 3 4 5 6 7 8 9 10
C 11 12 13 14
C Selection Vector:
C 0 0 0 0 0 1 1 1 1 1
C 1 1 1 1
C
C Original parent metastables:
C (1S)
C Parent bundling vector:
C 0
C Insert producer id here:D.H.Brooks
C Date: 13/02/96
-----

```

Figure 3.12: Example of Merged Resolution Levels

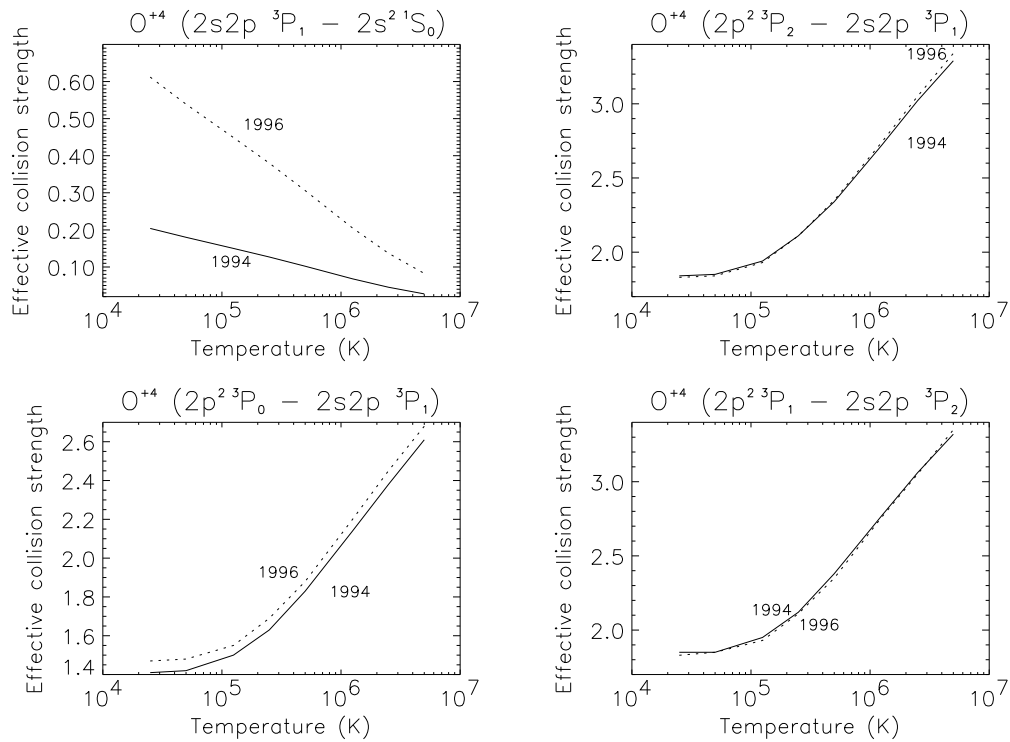


Figure 3.13: Comparison of Effective Collision Strengths produced by ADAS210 to higher quality calculation. The solid lines labelled 1994 are the higher quality data in this case from Kato et al.(1990). The dotted lines labelled 1996 are the results using the unbundling code ADAS210.

density due to its low radiative transition probability. In fact, it is this transition which is the least accurate of the examples shown. However, this is exactly as we would expect. Spin changing transitions are forbidden in LS coupling so the SUPERSTRUCTURE code does not contain this transition. Hence, using data output from it as the template will not allow us to unfold such transitions. However, for a light ion such as oxygen, the configuration interaction between the  $2s2p \ ^1P$  and  $2s2p \ ^3P$  levels is such that the  $2s2p \ ^3P_1 - 2s^2 \ ^1S_0$  transition has a slight dipole contribution. Therefore, SUPERSTRUCTURE will ‘fill in’ approximately with less accurate values roughly following this effect. In this case it mimics this behaviour by using the  $2s2p \ ^1P_1 - 2s^2 \ ^1S_0$  electric dipole transition data. We can ‘see’ this if we return to

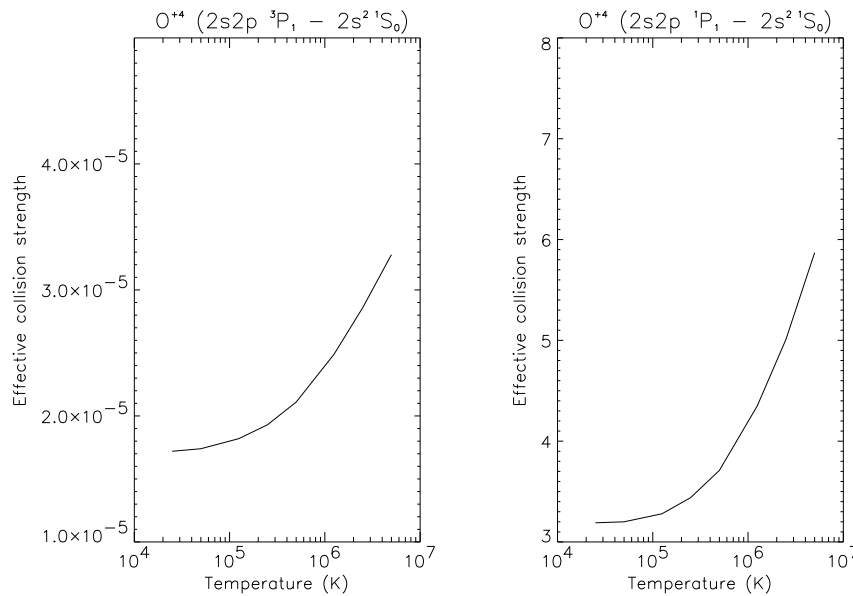


Figure 3.14: Original SUPERSTRUCTURE data

the SUPERSTRUCTURE dataset. Fig.3.14 shows the effective collision strengths for these two transitions plotted against temperature. The data comes from the file *copss#be\_ss#o4j.dat*. For a spin changing transition we would expect the value of  $\Upsilon$  to fall off at high temperature. Instead notice the broadly similar behaviour for both transitions imposed by SUPERSTRUCTURE. The upturn at high temperatures is typical of dipole transitions and behaves like  $d \ln(4\epsilon_i^2 / \Delta E_{ij})$ , where  $d = 4\omega_i f_{ij} / \Delta E_{ij}$ ,  $\epsilon_i$  is the incident electron energy,  $\Delta E_{ij}$  is the transition energy,  $\omega_i$  is the statistical weight of the level  $i$  and  $f_{ij}$  is the optical oscillator strength (Henry (1981)). The unbundled effective collision strength for this transition only reproduces the original data to within a factor of 3 (see top left graph in fig.3.13).

The largest discrepancy for the three other transitions presented in fig.3.13 is for the  $2p^2 \ ^3P_0 - 2s2p \ ^3P_1$  transition at low temperature ( $\sim 3 \cdot 10^4 K$ ) (see bottom left graph in fig.3.13). Here the disparity is still only 4.2%, and in the other two transitions (see top right and bottom right graphs) it is less than 2% for all temperatures. These are all magnetic dipole transitions since the total orbital angular momentum does

not change. We can conclude from this that, as expected, using the SUPERSTRUCTURE data is a good method for unbundling electric dipole allowed transitions but inappropriate for unbundling spin changing ones.

### 3.1.4 Atomic Data Verification

#### Introduction

Appropriate handling of large quantities of data is essential for general studies and for focusing attention on dominant transitions, but there are many circumstances in which it is necessary to assess and check the validity of individual atomic collision cross-sections in greater detail. In procuring data from the general literature it is often a source of annoyance to find transcription errors and inconsistencies between data generation and publication. Simple methods for determining data accuracy, comparing with previous calculations and uncovering errors can be extremely useful.

For example, it is important when assessing new data to ensure that any extrapolation procedures used, correctly describe the high energy behaviour of the cross-sections. One method of doing this is to remove the energy dependence completely by plotting a comparison ratio between the data and an adjustable expected approximate form. Such a ratio is sometimes referred to as the ‘*reduced  $\Omega$* ’, where  $\Omega$  refers to the collision strength. A ‘*reduced  $\Omega$* ’ plot of this type is generally slowly varying and any unusual resonance or bad point appearance should be more easily detectable. Henry (1981) discussed the limiting behaviour of cross-sections for different transition types and through his work we can obtain reasonable approximate forms.

Another problem is that at high energies the number of partial waves which have an influence on the calculation of the total cross-section increases. This happens through their contribution to the individual partial cross-sections in the expansion of the total cross-section. A lack of convergence of the expansion results and the total cross-section falls off too rapidly at high energy if insufficient partial cross-sections are included in the calculation.

An example of each of these two effects was given by Burgess & Tully (1992). Firstly, they looked at the  $\Upsilon$  data of Pradhan, Norcross & Hummer (1981). These

authors failed to reproduce the expected high energy limit point as they did not ensure that the coefficient of the dominant logarithmic term was correctly related to the  $gf$  value (see eq. 8 in Burgess & Tully (1992) and also the expression for the limiting behaviour used for electric dipole transitions below). Secondly, they mention the paper by Christensen, Norcross & Pradhan (1985) who corrected the lack of convergence in their distorted wave calculation, of the cross-section for the  $Fe^{+14}(3s^2\ ^1S - 3s3d\ ^1D)$  transition, produced by only including partial waves up to  $l = 11$ . After including partial waves up to  $l = 50$  their results changed from ‘dropping off’ at high energy and instead tended more accurately to the Born limit calculated by Burgess & Tully (1992).

Finally, it is usual when modelling the excited populations of ions in astrophysical and laboratory plasmas to assume that the electron distribution is Maxwellian. In such circumstances the excitation and de-excitation rate coefficients are expressible as functions of the Maxwell averaged collision strength  $\Upsilon$  (also referred to as the ‘*effective collision strength*’ or ‘*rate parameter*’, see chap.2, eq. 2.26). Therefore, it is convenient, at this point, to calculate  $\Upsilon$  and to perform a check on its validity.

ADAS series 1 implements techniques to tentatively verify cross-sections and rate coefficients. It also provides a mechanism to flexibly enter data, in the variety of forms present in the literature, into a standard archiving structure for future entry to the central ADAS database. ADAS101 deals with entry and validation of electron impact excitation cross-sections and display and conversion to effective collision strengths. It also includes an ADAS graph editor for quick removal of bad points and immediate updated processing of alterations by the user. ADAS102 similarly allows validation of the calculated  $\Upsilon$ s and was recently included in ADAS.

Opinions differ on the best way to verify atomic data individually so ADAS101 has been set up to allow the use of different analysis methods. Presently, two have been included. The first is the type of analysis used at JET (the ‘*ADAS option*’) and the second is an implementation of the technique developed by Burgess & Tully (1992) (the ‘*Burgess option*’). The interactive capabilities of IDL prove very appealing when confronted with problems associated with quick plot adjustments and immediate updating of interpolated splines.

## Background Theory

It is only possible for the electron impact excitation reaction (2.24) to take place if  $\epsilon_i \geq \Delta E_{ij}$ , where  $\Delta E_{ij} = E_j - E_i$ . Therefore, we can introduce a ‘*threshold parameter*’,  $X = \epsilon_i/\Delta E_{ij}$  which ranges from 1, at the excitation energy threshold, to  $\infty$ . The reaction, (3.1), is described by a cross-section which is expressible either in terms of the incident electron energy,  $\epsilon_i$ , the scattered electron energy,  $\epsilon_i - \Delta E_{ij}$ , or the threshold parameter.

Cross-section data can be obtained from the literature in many forms. ADAS internally converts all types into collision strengths,  $\Omega_{ij}$ , tabulated as a function of  $X$  using the relationships defined in sec.2.2.1. In subsequent ADAS routines, rate coefficients are stored as effective collision strengths. It is common also in the literature to provide the atomic transition probabilities in the form of the ‘*line strength*’, the ‘*absorption oscillator strength*’ or the ‘*spontaneous emission coefficient*’. These are related by

$$A_{j \rightarrow i} = \frac{\alpha^4 c}{2a_0} \left( \frac{\Delta E_{ij}}{I_H} \right)^2 \frac{\omega_i}{\omega_j} f_{i \rightarrow j} = \frac{\alpha^4 c}{6a_0} \left( \frac{\Delta E_{ij}}{I_H} \right)^3 \frac{1}{\omega_j} S_{ij} \quad (3.13)$$

where  $\alpha^4 c/2a_0 = 8.0323 \times 10^9 s^{-1}$ . ADAS accepts each of these forms. For our comparison graph we can identify three types of transition for which we have simple approximate forms which show the correct asymptotic behaviour. These are dipole, non-dipole and spin change, and the functions we use are,

1. dipole:  $\Omega_{ij} \approx F_3 S_{ij} \ln(X + F_2)$
2. non-dipole:  $\Omega_{ij} \approx F_3 + F_2/X$
3. spin change:  $\Omega_{ij} \approx F_3/(X + F_2)^2$

where  $F_2$  and  $F_3$  are constants. For the dipole case  $F_3 = 4/3$ . A plot of the ratio of our data to our approximate form, should therefore have a value  $\sim 1$ . It is convenient also to map our threshold parameter directly onto the range  $[0,1]$  by introducing a ‘*reduced X*’ such that  $X_r = 1 - 1/X$ .

The theory behind the method of Burgess & Tully (1992) is described in great detail in their paper. However, the reduced comparison plots and transition type high

energy behaviours are broadly similar to those discussed above. Reduced collision strengths and energies are calculated using transition type specific scaling rules and the asymptotic behaviours are expressed as functions of electron energy. The main difference is the introduction of an adjustable parameter  $C$  which allows a more accurate least squares fit to the simplified data to be obtained essentially by rescaling in the  $x$ -direction. The authors' idea was to provide data compaction by reducing the data to five points obtained by a representative spline fit. They also wished to extend this technique to calculation and storage of Upsilon's (see the ADAS manual for details of this). Here we are only concerned with the fitting of  $\Omega_{ij}$  and computation of  $\Upsilon_{ij}$ .

Burgess & Tully (1992) distinguish four transition types which also correspond to those identified in ADAS. The fourth, not mentioned so far, is referred to as '*semi-forbidden*' or '*other*' and is associated with very small oscillator strengths. The limiting behaviours are

1. electric dipole:  $\Omega_{ij} \approx \text{const.} \ln(E)$
2. non electric dipole, non exchange:  $\Omega_{ij} \approx \text{const.}$
3. exchange:  $\Omega_{ij} \approx \text{const.}/E^2$

And, the scaling rules are,

**Type 1 : Dipole**

$$x = 1 - \frac{\ln C}{\ln(E_j/E_{ij} + C)}$$

$$y(x) = \frac{\Omega_{ij}}{\ln(E_j/E_{ij} + e)}$$

such that  $x \in [0, 1]$  for  $E_j \in [0, \infty]$ ,  $y(0) = \Omega_{ij}(0)$  and  $y(1) = 4\omega_i f_{ij}/E_{ij}$ .

**Type 2 : Non-Dipole**

$$x = \frac{\left(\frac{E_j}{E_{ij}}\right)}{\left(\frac{E_j}{E_{ij}} + C\right)}$$

$$y(x) = \Omega_{ij}$$

such that  $y(0) = \Omega_{ij}(0)$ . Burgess & Tully(1992) also point out that  $y(1)$  can be obtained through Born approximation.

**Type 3 : Exchange**

$$y(x) = \left( \frac{E_j}{E_{ij}} + 1 \right)^2 \Omega_{ij} \quad (3.14)$$

here  $x$  is as for type 2 and  $y(0)$  again equals  $\Omega_{ij}(0)$  and  $y(1)$  is difficult to obtain.

**Type 4 : Other**

$$y(x) = \frac{\Omega_{ij}}{\left( \frac{E_j}{E_{ij}} + C \right)} \quad (3.15)$$

here  $x$  is as for type 1 and  $y(0) = \Omega_{ij}(0)/\ln C$  and  $y(1)$  is as for type 1.

Adjustment of  $C$  will alter the approximate form and re-evaluation of the least squares fit is necessary. In the case of type 1,  $C > 1$ , while in types 2,3 & 4  $C > 0$ .

### 3.1.5 Overview of ADAS101 (Electron Impact Excitation Cross Section)

#### Summary of Program and Processing Linkages

A schematic diagram of the linkages between the FORTRAN and IDL processes, in ADAS101, is shown below. Due to the complexity of the code and the number of potential pathways through it, this diagram appears quite intricate. However, it does follow the standard ADAS sequential structure discussed previously. The boxes down the left hand side of the diagram represent each of the IDL interactive screens. The first is the archive option selection screen which replaces the usual input screen although it is broadly similar in appearance and use. The second is the processing options screen, which acts as the primary data entry utility. The third is the output options screen, at which point selection of an analysis option must be made. The fourth is the interactive editable graph screen, which replaces the usual plot screen, and finally, a display plot of the calculated upsilons. The dashed boxes in the diagram show the alternative path taken when the Burgess analysis is chosen.

The input options screen has the same data file entry widgets as discussed previously for ADAS209 and ADAS210. However, its functionality and purpose are slightly different and it has an additional feature in that data file selection can be over-ridden. The main point is that users enter data, analyse it, and decide how to store, maintain



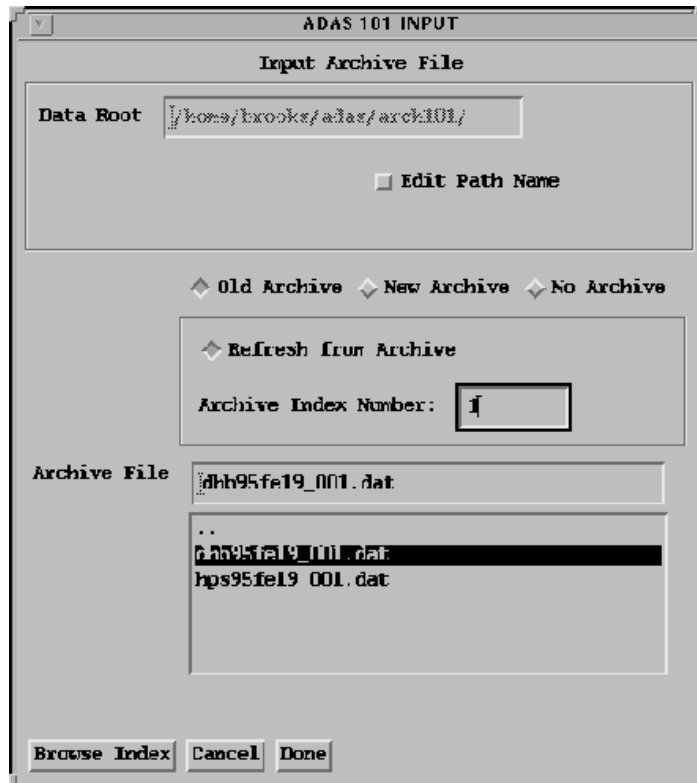


Figure 3.16: ADAS 101 input options - example widget screen. See text for explanation of use.

and update their individual databases. ADAS101 provides an archiving facility to deal with this need but there is no central ADAS database for this routine. There are three archiving options to choose from (see fig.3.16), '*Old Archive*', '*New Archive*' or '*No Archive*'.

It is possible to reload and re-analyse data from an old archive. Archive files created by ADAS101 consist of sequentially indexed transition data, and recovery of data for a particular case is made by reference to its '*Archive Index Number*'. Selecting '*Old Archive*' allows this and will sensitise the framed area and selection list below. An archive file is then chosen from the selection list display window. In this particular case it will operate in the usual manner. Once a file is selected the '*Browse Index*' button becomes active and it is important to use this to obtain the correct index number for the transition being analysed. Clicking on '*Refresh from Archive*' makes the '*Archive Index Number:*' entry box editable. The index number of the data to load is entered here. If refreshing from an old archive the defaults in subsequent windows will come from this file.

When entering new data, it is possible to save it to an old archive file. In this case, the user would follow the same procedure as above but avoiding the '*Refresh from Archive*' button. Archive files can be stored and created anywhere by using the '*Edit Path Name*' button. This performs in the same manner as for ADAS209 and ADAS210. However, the usual ADAS practice is to make a directory '*/home/<username>/adas/arch101*' and create files with the naming convention '*<user initials><year><element><charge+1>\_<no.>.dat*', e.g. '*dhb95fe19\_001.dat*'.

Clicking on '*New Archive*' will start a fresh archive file. The file is created by entering a name in the 'selected file' window which appears above the selection list. This field becomes editable for this particular case. It is also possible to completely ignore archiving and analyse a transition perhaps only keeping a printout file as record. To do this select '*No Archive*', at which point nothing more can be done on this screen. Clicking on '*Done*' will advance to the '*Processing Options*' screen.

**N.B.** It should be noted that no actual archiving is done until the final output screen, but all the selections made here will determine how the code deals with the archiving later. Also, there is no data replacement or substitution capability in

ADAS101; any new analysis, which requires archiving, is appended to the existing data contained within the archive file and given a new index number.

The next screen is the ‘*Processing Options*’ one. All the data about the transition for analysis is entered here. If the refresh from an old archive option was selected then everything on this screen will be obtained from the chosen file. In fig.3.17 the values shown are for the  $Fe^{+18} 2p^4 \ ^3P - 2p^5 \ ^3P$  transition. The data were taken from Mann(1983) and were recovered from an old archive. The ‘*Data File Name:*’ and ‘*Browse Index*’ button are displayed again. Below this are editable text boxes for entering the ‘*Nuclear Charge*’, ‘*Ion Charge*’ and ‘*transition type*’. The latter correspond to the types specified earlier. In the case of dipole transitions the ‘*Transition Probability*’ must be entered. This can be in one of three forms; the line strength (S), the absorption oscillator strength (f) or the Einstein A-value (A). Clicking on the buttons here will calculate the correct values for the other parameters. The details of the lower and upper levels of the transition are then entered. These are the chosen lower and upper level ‘*Index*’, ‘*Statistical Weight*’ and ‘*Level Energy*’, followed by the units that are being used for the energies. The latter are altered by clicking ‘*Select Energy Form:*’.

Finally, the input cross-sections, energies and output temperatures for the transition are entered by means of the ADAS table editor (for a complete description see Summers et al.(1996)). This is activated by clicking on ‘*Edit Table*’ with the mouse. Clicking on ‘*Default Temperature Values*’ will insert a standard set of temperatures, between  $2.5 \times 10^4 K$  and  $5.0 \times 10^6 K$ . Confirmation is required here and is made by responding to a ‘pop-up’ prompt which asks whether to ‘*Confirm Overwrite Values with Defaults*’. If an old archive is being refreshed then the table will already be filled with the values from the archive. The units used for the energies, cross-sections and temperatures are also displayed on the screen. When working with the table editor data can be entered using a wide array of units i.e. Kelvin, eV, scaled and reduced temperatures, Omega, scaled omega, *exc.* :  $\pi a_0^2$  and *dexc.* :  $\pi a_0^2$  for cross-sections and  $E_j k^2$ ,  $E_i k^2$ ,  $E_j (k/z_0)^2$ , X and  $E_j (k/z_{eff})^2$  for energies. This is to ease the problems of unit variation between different data sources. However, if a quantity of data is refreshed from the archive it will always appear in the form of the collision strength  $\Omega_{ij}$

**ADAS101: PROCESSING OPTIONS**

Data File Name: /home/brooks/adas/arch101/dhb95fe19 001.dat

Browse Index

Nuclear Charge: 26.00      Ion Charge: 18.00

Transition Type: 1      Transition Probability: 5.40e+10

Form:  S  f  A

Lower Level	Upper Level
Index: 1	Index: 4
Stat. Wt.: 9.0	Stat. Wt.: 9.0
Level Energy: 0.347841	Level Energy: 8.705277

Select Energy Form:  cm -1     Rydbergs

INDEX	Input Energy	Input X-sect	Output Temp.
1	1.0100E+00	1.2804E+00	3.6100E+05
2	1.2710E+00	1.3030E+00	7.2200E+05
3	1.5990E+00	1.3304E+00	1.8050E+06
4	2.0120E+00	1.3626E+00	3.6100E+06

Energy units: eV. X-section units: Omega. Temperature units: Kelvin

Edit Table

Default Temperature Values

Edit the processing options data and press Done to proceed

Cancel Done

Figure 3.17: ADAS 101 processing options - example widget screen. See text for explanation of use.

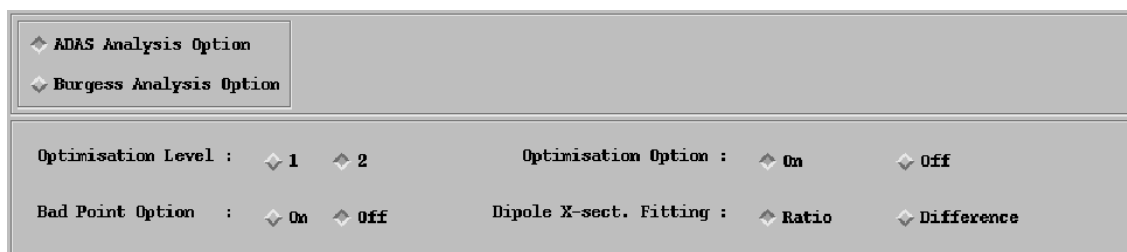


Figure 3.18: ADAS 101 output options - example segment of widget screen. See text for explanation of use.

and the threshold parameter,  $X$ . When all the editing and data entry is completed clicking on ‘*Done*’ will proceed.

The ‘*Output Options*’ screen enables selection of the analysis method and also allows control over the graph scalings, positioning of output files etc. An example diagram of the screen is available in the ADAS July 1995 bulletin. A segment of it is shown in fig.3.18. As with the previous displays the ‘*Data File Name:*’ is presented along with a ‘*Browse Index*’ button. Below this are the actual analysis option buttons discussed above, ‘*ADAS Analysis Option*’ and ‘*Burgess Analysis Option*’, see fig.3.18. In fig.3.18 we have chosen the ADAS option. Below these buttons is a ‘switching widget base’ which offers different options depending on the analysis method selected. In practice this area will be blank when the output screen first appears. There is a practical reason for this. When switching between analysis methods on old archive data processed by one technique, ADAS101 has no way of knowing what options

would be best suited for the data using the alternative method. Therefore, it is best to set default values and these are, obviously, not suited to all transition types. However, wildly differing results can be obtained using different options and analysis methods, so it was felt best to force selection of a method to ensure consideration of the various options rather than allowing the user to blindly accept the defaults.

The four buttons for the ADAS option are '*Optimisation Level:*', '*Bad Point Option:*', '*Optimisation Option:*' and '*Dipole X-sect. Fitting:*'. A full explanation of these is given in the ADAS manual (Summers (1994)). However, briefly, the '*Optimisation Option*' should generally be switched on with 1 or 2 chosen as the '*Optimisation Level*'. The '*Optimisation Level*' uses either the first data point, or the first and last data points, to set an approximating form for the data. The '*Bad Point Option*' should be switched on when there are only a few points or the data includes unrealistic values. This option will help to avoid overshooting of the interpolating spline. The '*Dipole X-sect. Fitting*' determines whether the spline is fitted to the difference, or the ratio, of the input data and the approximate form. In the case of the Burgess Option, the only active entry is an initial estimate of the adjustable C-parameter. A second button '*Insert Non-dipole Point at Infinity*' is not in use at this time.

The actual output screen then presents various graphical and text optional selection facilities. For example, there is a '*Graphical Output*' button which allows entry of a working '*Graph Title*' and default scaling ranges for the scaled omega and final epsilon plots. An '*Explicit Scaling*' button, associated with this, is used for entering minimum and maximum values for the x and y ranges of the editable omega and final epsilon plots. A printout of both the subsequent graphs can be obtained by using an '*Enable Hard Copy*' option and entering a '*File Name*', usually '*graph.ps*'. If the file already exists then ADAS101 issues a warning and an option to '*Replace*' the file is given. The output device can be chosen by clicking on a member from a small selection list. ADAS101 supports three choices at present, *Post-script*, *HP-PCL* and *HP-GL*. A hard copy of the output is also available and is printed in a similar form to how it appears in any archive file. This operates using a '*Text Output*' button similar to that which appears in ADAS209 and ADAS210. Once all the selections are complete, clicking on '*Done*' will process the data using the chosen analysis option

and bring up the editable graph screen.

The example in fig.3.19 is of the Burgess analysis graph editor. It is broadly similar to and encompasses all the features of the ADAS graph editor, but it also has some additional specific amenities. The common features are '*Move Point*', '*Delete Pt.*', '*Add X-pt.*' , '*Add Any Pt*' and '*Insert Point by Value:*'. While the Burgess option also allows movement of the 5 knot points of the Burgess spline, '*Move Knot*', constant display of the R.M.S. error, and immediate adjustment of the scalable C-parameter, '*Adjust C-value*'. In fig.3.19, the diamonds represent the original data and the dashed line is a best fit line drawn through them. The five knot points appear as crosses on the vertical lines and the solid line is the spline through the five knots. In this example, the C-value chosen is not particularly accurate, so that the data points and spline are visually separated on the graph. In the ADAS graph editor, the data points are also represented by crosses and the spline is a solid line. In this case there are no knot points. The editor has five different modes of operation. In general, the left hand mouse button is used for selecting and the right hand one for exiting.

*'Move a point mode:'* This is activated by selecting the '*Move Point*' button on the widget display. The overall widget is desensitised and the plot area becomes active. A data point is then selected with the left mouse button and is dragged around by holding this button down. To place the point in a new position, the mouse button is released. This process can be repeated any number of times. Pressing the right hand mouse button exits from this mode.

*'Delete point mode:'* Clicking on '*Delete Pt.*' will activate this mode. Once again, the plot area becomes editable and a point can be removed by selecting it with the left mouse button. The right mouse button is the exit button here also.

*'Add x-point mode:'* This can be done in two ways. Either, selecting '*Add X-Pt.*' on the display, or, entering values in the boxes provided. To add a point with the mouse we simply decide where to place it and click the left mouse button. Clicking the right hand one will again exit. To add explicitly, we enter the coordinates in the '*X-Val:*' and '*Y-Val:*' boxes and then select '*Click to Insert*'. The precision of this method can be greatly improved by noting the on screen x and y coordinates, which are displayed when moving a point, and making use of the '*Refresh*' button. When

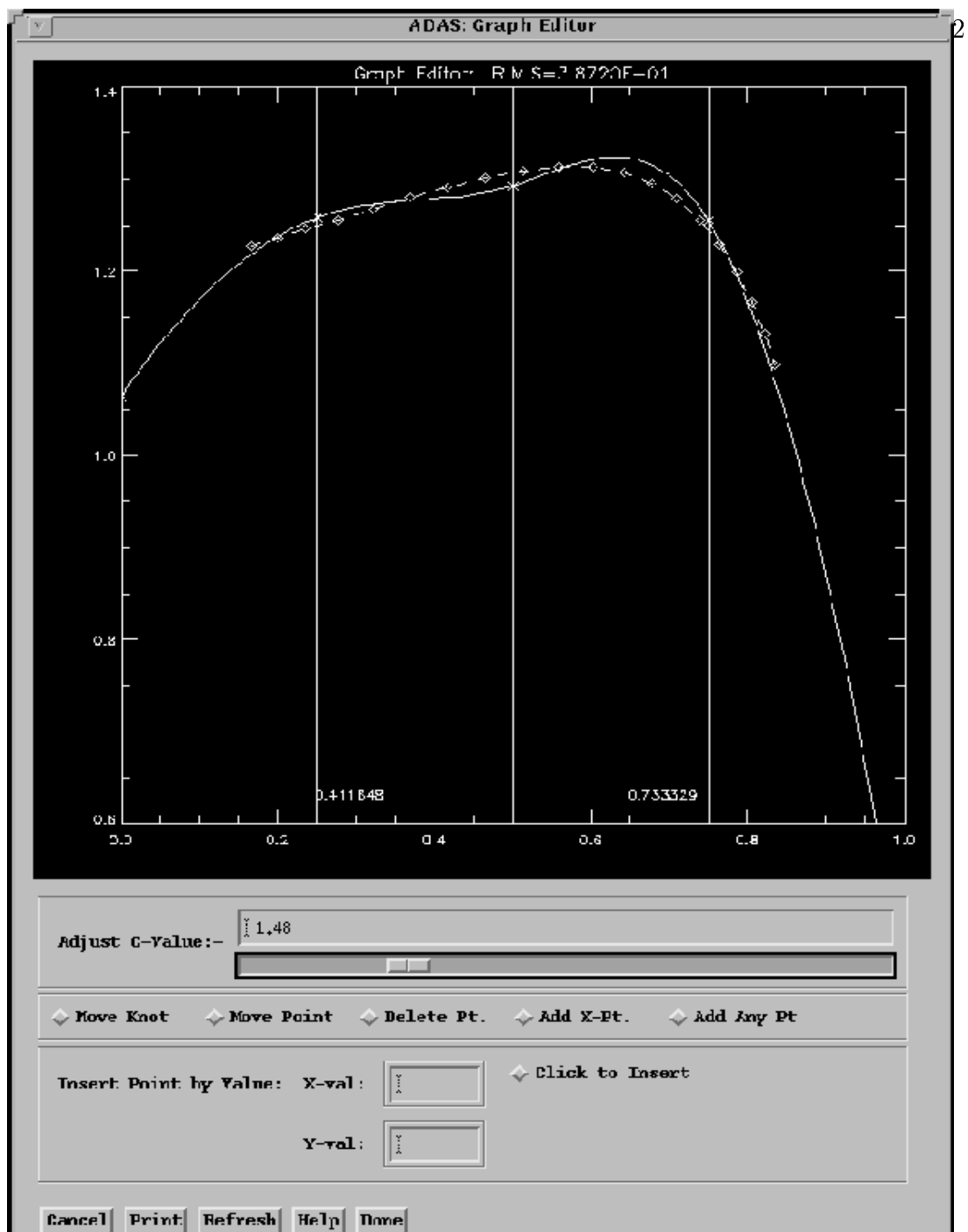


Figure 3.19: ADAS graph editor - example of Burgess option widget screen showing data from the  $Fe^{+18}(2p^4\ ^3P - 2p^5\ ^3P)$  transition. See text for explanation of use. The diamonds are the data points, the dashed line is a spline fit to them. The stars are the Burgess knot points and the solid line is the spline fit through them. The x and y positions of the cursor are given on the screen also.

selected, *Refresh* restores the graph by replotting the original unaltered data. This is helpful if any errors are made during editing. Therefore, moving the point to its best position, recording the displayed x and y values and refreshing the plot so the code doesn't record them, allows entry of significantly more accurate values.

The above two methods will add the point between the nearest two x-values to maintain the x vector as monotonically increasing. However, the following method will add a point anywhere. This is physically unreasonable, but is added for completeness.

*Add anywhere mode:* This is activated by clicking on *Add Anywhere* with the mouse cursor. It is a slightly more complex mode than the others. First, we select the point we wish to add after with the left hand mouse button. Then, we click the left mouse button again to place the points. The points are then inserted after the one originally chosen, and this will continue for unlimited points until we press the right hand button. Therefore, no matter where the point is placed, it will be inserted after the original point, and splines, plot lines etc. will pass through it in that order. Theoretically a circle could be drawn in this way. To add more data we simply choose a second starting point. In this mode the right hand button must be pressed twice to exit; once to stop adding after a particular point and then again to exit completely.

*Move a Burgess knot mode:* This mode is executed by clicking on *Move Knot* and it is performed in the same manner as with the *Move a point* mode. However, it is unique to the Burgess analysis option. In this way repositioning and visual adjustment of the five spline knot points calculated by this method is possible. Only vertical motion of the points is allowed so that they remain at the five x-coordinates specified by Burgess & Tully (1992). Once again selection is with the left button and exit is by the right one.

When moving, adding or deleting a point it is the actual data that is altered. Therefore, if any changes are made, the program will re-process the approximate forms and fits. Hence, moving a point and then clicking *Done* will cause the *Output Options* screen to re-appear, rather than continuing forward as it usually does. Alteration of the five Burgess knot points does not have this effect as their technique allows for a final visual check on the spline.

As mentioned above, the other feature unique to the Burgess option is the ability

to re-adjust the scalable parameter  $C$ . A widget slider is available, *'Adjust C-value'*, which allows quick and easy adjustment without the laborious task of continually cancelling and reprocessing. When the slider is moved the value is recorded, the data is reprocessed and the new R.M.S. is computed and immediately updated. The graph is also replotted and this makes any analysis much easier. Finally, there are some extra buttons along the bottom row of the screen. The *'Cancel'* and *'Done'* buttons have their usual meaning and the *'Refresh'* button has already been discussed. The *'Print'* button writes the edited plot into a file suitable for printing. This is only possible if an output device was selected on the *'Output Options'* screen. Finally, it was decided that although the graph editor was fairly straight forward to use, it could appear quite daunting at first sight and problems could be generated easily by moving too fast and getting out of step between the buttons and the IDL processing. Therefore, it was thought desirable to have some information available to clearly explain its use. Clicking on *'Help'* displays instructions on which buttons to press etc. similar to those given above. The *'Help'* screen can be left up as a display while using the graph editor. In addition to this there is a reminder present on the screen, in the position of the title, which explains how to start each process e.g. in *'Delete point mode'* it will display *'left button selects point to delete'*.

The final screen is a plot of the calculated upsilons. It is displayed with information about the time, date, ADAS release and file used. The graph itself is plotted against  $\log_{10}(T_e(K))$ . Once again a printout of the graph can be obtained by clicking on *'Print'*. Finally, there is an option to archive the analysed data. Clicking on *'Archive'* will do this no matter which options were chosen on the first screen. The original options only open old or create new files and prepare their format but do not actually write anything to them. This is because we often wish to try reprocessing several times before deciding on the best fit. However, clicking on *'Archive'* when originally choosing *'No Archive'* is a contradiction and ADAS101 will 'pop-up' a warning accordingly. However, the data will be archived anyway, in this case, and the default is to create a file *'archive.dat'* in the users' *'arch101'* directory.

## Example of Archive File

Fig.3.20 shows an example of the archive file produced by ADAS101. Both the ‘*ADAS analysis*’ and the ‘*Burgess analysis*’ options have been used to create the two differing output formats. The data shown is again for the  $Fe^{+18}(2p^4\ ^3P - 2p^5\ ^3P)$  transition. The first line of the file identifies the index number of the data block and also the analysis option chosen. ‘*A*’ corresponds to the ‘*ADAS analysis*’ and ‘*B*’ to the ‘*Burgess analysis*’. Between these, some information about the transition and also the date are given. In fact, this area is filled with the graph title from the ADAS101 run. Lower down in fig.3.20 the same information is repeated. This is the start of the Burgess data block where the index number is increased and the analysis code updated. Data blocks in the archive file end with the terminator ‘-1’ in the first column.

The data archived using the ‘*ADAS analysis*’ is fairly self explanatory and more details are given in the ADAS manual (Summers (1994)). In the case of the ‘*Burgess analysis*’ the data block has been set up to maintain the precise formatting output from Burgess’s own codes, except that the data has been shifted along by one space to allow the ADAS search routines to identify index numbers, and that all the data is merged in one file rather than two separate ones for  $\Omega$  and  $\Upsilon$ . Due to this alteration it was thought unnecessary to repeat the information on statistical weights etc. Also, ADAS has two routines (101 & 102) to do the fitting of  $\Omega$  and  $\Upsilon$ , so the final  $\Upsilon$ s are output in the archive file but the Burgess spline fit is not. These would appear in the archive file for ADAS102. The FORTRAN and IDL routines which output this data were coded up from the original BBC BASIC program provided by Alan Burgess. Below the information header, the  $\Omega$  data is displayed. Below this is the *ion charge* then the *nuclear charge*. Underneath these are the lower level *energy*, in  $cm^{-1}$ , and *statistical weight* followed on each line by those for the upper level. Next is the *transition energy*, the *transition type*, the *gf* value and the adjustable *C-parameter*. Following this are the five y-coordinates of the spline knot points, two parameters which indicate the range of the upsilons and the lower and upper level indices. The producer program is then printed as a reference and finally the number of cross-sections followed by the *reduced energies* and the actual  $\Omega$  values. After this the  $\Upsilon$  information, truncated from the Burgess file, is given. Essentially this is just

```

I 1 FE+18 2P4(3P)-2P5(3P) MANN(1983) 26/2/93          A
NUCLEAR CHARGE = 26.00 ION CHARGE = 18.00 EFFECTIVE ION CHARGE = 19.00
LOWER INDEX = 1          UPPER INDEX = 4
LOWER ENERGY = 0.347841  UPPER ENERGY = 8.705277
LOWER WEIGHT = 9.0       UPPER WEIGHT = 9.0
ACOEFF = 5.4000E+10  S = 3.1095E-01  FABS = 9.6251E-02  ENERGY DIFFERENCE = 8.357436
XSECT TYPE = 1  XSECT OPT.FACTORS, FXC2 = 6.9706E+00  FXC3 = 1.9825E+00  IXOPS = 1  IBPTS = 0
IFPTS = 2  IDIFF = 0
X          OMEGA      APPROX. OMEGA      DIFFERENCE
1.0100E+00  1.2804E+00      1.2804E+00      2.2204E-16
...
7.9472E+01  2.7491E+00      2.7491E+00      -1.6645E-07
TE(K)      GAMMA      APPROX. GAM      EX. RATE      DEXC. RATE      GBARF
3.6100E+05  1.3027E+00      1.3001E+00      5.3754E-11      2.0791E-09      8.3365E-02
...
3.6100E+09  4.5342E+00      4.5342E+00      7.2341E-11      7.2367E-11      2.9017E-01
-1
I 2 FE+18 2P4(3P)-2P5(3P) MANN(1983)08/03/96          B
OMEGA
18.00
26.00
  38171.080  9.0
  955292.288  9.0
  917121.208
1
0.8663E+00
1.68000
0.117E+01  0.127E+01  0.131E+01  0.116E+01  0.415E+00
0.00E+00  0.32E+00
1 4
ADAS101
20
0.101E+01  0.128E+01
...
0.795E+02  0.275E+01
UPSILON
13
0.3610E+06  0.3641E-01
...
0.3610E+10  0.3220E+00
-1
C-----
C INDEX CODE ION  TRANSITION  SOURCE  DATE
C 1      A  FE+18 2P4(3P)-2P5(3P)  MANN(1983)  26/02/93
C 2      B  FE+18 2P4(3P)-2P5(3P)  MANN(1983)  08/03/96
C-----

```

Figure 3.20: ADAS 101 example archive file. See text for a complete description.

the number of points followed by the temperatures, in Kelvin, and the calculated  $\Upsilon$ s.

## 3.2 Methods for Derived Data Production

### 3.2.1 Ionisation Balance

#### Introduction

In sec.2.2.3 the applications of derived data to plasma spectroscopy and research were outlined. Central to these applications was the use of the line emission contribution function otherwise known as the  $G(T_e, N_e)$ . As can be seen from eq. 2.64, equilibrium abundance fractions for each of the ionisation stages normalised to the total abundance of the element in question contribute to the make up of the  $G(T_e, N_e)$  function. The importance of calculating these can also be seen from sec.2.2.3. It is usual in astrophysics to assume the plasma under investigation is in equilibrium. Therefore, for derived data to be of widespread use, these fractions should be computed as accurately as possible. A discussion of the validity of this assumption is deferred until later in this chapter.

We require values of these fractions for all minor species contributing to the radiation emission from the solar atmosphere. A complete study must include processes such as dielectronic recombination in detail which can shift the temperature dependence of the peak of the ionisation stages. Also, it is important to evaluate the effect of finite plasma density and the influence of metastable states. Usually when describing the ionisation state of ions of an element only the abundances of complete ionisation stages are considered. However, plasma dynamic and atomic relaxation timescales are such that metastables can accumulate large populations and that they may evolve on long timescales (see sec.2.2.2 and 2.1.2). In these cases, the metastables are important to population calculations, contributing substantially to refined dielectronic recombination and ionisation computations. The make up of the  $G(T_e, N_e)$  function can be significantly influenced also, as can the total radiative power functions, since

they contain the ion fraction in their representation, (eq. 3.16).

$$P_{tot} = \sum_{z=0}^{z_0} \sum_{\sigma=1}^{m_z} P^z \frac{N_{\sigma}^z}{N_{tot}} \quad (3.16)$$

where  $P^z$  can be separated into radiated power coefficient contributions from each of the individual metastables. Consequently composite radiative power loss functions for the solar atmosphere (Summers & McWhirter(1979)) may be altered as a result. Calculations of ionisation balance are common in astrophysics. The calculations of Summers(1974) used a full collisional-radiative model to produce effective ionisation and recombination coefficients dependent on density and temperature. With improvements in fundamental atomic data generation, calculations must be revised. The results of Arnaud & Rothenflug (1985) are currently the most widely used but do not include metastable states or the effects of finite density plasma. Exclusion of collisional-radiative effects has been justified for solar coronal densities by suggesting that zero density calculations only are necessary since its electron density is low  $\sim 5.4 \times 10^8 \text{ cm}^{-3}$ . In reality, this is not so. In particular, it is clear that the collisional dielectronic recombination coefficients will be significantly altered even at low density. This occurs through their influence on the individual dielectronic recombination coefficients (see chap.2.2.1). Also, when moving to the chromosphere, the low temperature enhancement of the collisional dielectronic coefficients, due to the influence of three body recombination at higher density, can play a more significant role.

Summers & McWhirter (1979) discussed the possibility of including metastables in a collisional-radiative model, while Summers & Hooper (1983) developed a method to incorporate them consistently. Kato, Lang & Berrington (1990), while not including recombination processes, showed that spectral line intensities were altered by up to 60% when the effects of metastable states were included. In the fusion context, Dickson (1993) performed a study of metastable resolved ionisation balance and its influence on derived data. In so doing, he found that metastables contribute substantially to calculations of radiated power.

We wish to make an improvement in the precision of our ionisation balance calculations by introducing the ability to treat metastable states explicitly. In doing this, we add a degree of complexity to the population distribution calculation. However,

it is ultimately of interest for studying the role of significantly populated levels in different plasma environments and essential for realistic dynamic studies. Therefore, a method to incorporate metastable states in the ionisation balance eqs. 3.18 is developed below. The first ‘in-depth’ study, directed at astrophysical plasmas, including also the influence of finite density plasma, is presented in chap.4 and also in Brooks et al.(1997).

## Background Theory

For an element  $X$ , evolving with time in a thermal plasma, with nuclear charge  $z_0$ , the population of the ion  $X^{+z}$  is denoted by  $N^z$ , where  $z \in [0, z_0]$ . To evaluate  $N^z$  we construct an equation describing its time evolution due to populating and depopulating collisional and radiative processes. Therefore,

$$\begin{aligned} \frac{d}{dt}N^z &= N_e S_{CD}^{z-1 \rightarrow z} N^{z-1} \\ &- \left[ N_e S_{CD}^{z \rightarrow z+1} + N_e \alpha_{CD}^{z \rightarrow z-1} + N_H C_{CD}^{z \rightarrow z-1} \right] N^z \\ &+ N_e \alpha_{CD}^{z+1 \rightarrow z} N^{z+1} + N_H C_{CD}^{z+1 \rightarrow z} N^{z+1} \end{aligned} \quad (3.17)$$

where the  $S_{CD}$ ’s,  $\alpha_{CD}$ ’s and  $C_{CD}$ ’s are the collisional dielectronic coefficients for ionisation, recombination and charge-exchange recombination respectively (see chap.2).

In equilibrium ionisation balance the time derivative in equation 3.17 is set to zero and the stage populations can be calculated by solution of the matrix equation,

$$N_e \begin{bmatrix} \ddots & & & 0 & & 0 & 0 \\ \ddots & & & \alpha_{CD}^{z \rightarrow z-1} + \left(\frac{N_H}{N_e}\right) C_{CD}^{z \rightarrow z-1} & & 0 & 0 \\ S_{CD}^{z-1 \rightarrow z} & - \left( S_{CD}^{z \rightarrow z+1} + \alpha_{CD}^{z \rightarrow z-1} + \left(\frac{N_H}{N_e}\right) C_{CD}^{z \rightarrow z-1} \right) & \ddots & \ddots & 0 & \ddots & \ddots \\ 0 & & S_{CD}^{z \rightarrow z+1} & & \ddots & \ddots & \ddots \\ 0 & & 0 & & \ddots & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \\ N_z \\ N_{z+1} \\ \vdots \end{bmatrix} = 0 \quad (3.18)$$

and subsequent normalisation to the total population,

$$N_{tot} = \sum_{z=0}^{z_0} N^z \quad (3.19)$$

where  $N_{tot}$  is the total number density of all ions of the element  $X$ . From Eq.3.18 and 3.19 we can obtain the equilibrium fractional abundances,  $N^z/N_{tot}$  at a grid of temperature and density points. This is the usual *unresolved* or *stage to stage* picture and assumes ionisation and recombination takes place only from and to the ground states of each ion.

To include metastable states, we partition the population of each stage  $N^z$  into ground and metastable levels denoted by  $N_\sigma^z$  where  $\sigma \in [1, m_z]$ . Eq. 3.17 then becomes,

$$\begin{aligned} \frac{d}{dt}N_\sigma^z = & N_e \sum_{\rho=1}^{m_{z-1}} S_{CD,\rho \rightarrow \sigma}^{z-1 \rightarrow z} N_\rho^{z-1} + N_\tau^{z+1} \left( N_e \sum_{\tau=1}^{m_{z+1}} \alpha_{CD,\tau \rightarrow \sigma}^{z+1 \rightarrow z} + N_H \sum_{\tau=1}^{m_{z+1}} C_{CD,\tau \rightarrow \sigma}^{z+1 \rightarrow z} \right) \\ & + N_\sigma^z \left( N_e \sum_{\sigma'=1; \sigma' \neq \sigma}^{m_z} Q_{CD,\sigma' \rightarrow \sigma}^{z \rightarrow z} + N_e \sum_{\sigma'=1; \sigma' \neq \sigma}^{m_z} X_{CD,\sigma' \rightarrow \sigma}^{z \rightarrow z} \right) \\ & - N_\sigma^z \left( N_e \sum_{\tau=1}^{m_{z+1}} S_{CD,\sigma \rightarrow \tau}^{z \rightarrow z+1} + N_e \sum_{\rho=1}^{m_{z-1}} \alpha_{CD,\sigma \rightarrow \rho}^{z \rightarrow z-1} + N_H \sum_{\rho=1}^{m_{z-1}} C_{CD,\sigma \rightarrow \rho}^{z \rightarrow z-1} \right. \\ & \left. + N_e \sum_{\sigma'=1; \sigma' \neq \sigma}^{m_z} Q_{CD,\sigma \rightarrow \sigma'}^{z \rightarrow z} + N_e \sum_{\sigma'=1; \sigma' \neq \sigma}^{m_z} X_{CD,\sigma \rightarrow \sigma'}^{z \rightarrow z} \right) \end{aligned} \quad (3.20)$$

which describes the population evolution of each metastable  $\sigma$  of each ionisation stage  $z$ . The symbols  $\rho$  and  $\tau$  index the metastables of the  $z-1$  and  $z+1$  ions respectively i.e.  $\rho \in [1, m_{z-1}]$  and  $\tau \in [1, m_{z+1}]$ . Also,  $\sigma' \in [1, m_z]$  indexes the metastables of the  $z$ th ionisation stage, with  $\sigma' \neq \sigma$ . The collisional radiative coefficients now correspond to those of the *metastable resolved* case. Two new coefficients are introduced which link metastables of the same stage; contributions to populating and depopulating processes via metastable cross-coupling reactions,  $Q_{CD}$ , and recombining ion parent to parent metastable cross coupling reactions,  $X_{CD}$ . It is convenient to separate them because of their different origin in calculation.  $X_{CD}$  is that part of the cross-coupling which occurs through capture+autoionisation and is best associated with dielectronic recombination. In this case the  $X_{CD}^{z \rightarrow z}$  coefficients refer to the parents of the  $z-1$  stage. The left hand side of Eq. 3.18 then becomes,

$$N_e \begin{bmatrix} \ddots & & & & 0 & & 0 & 0 \\ \ddots & & & & \mathbf{A}_{CD}^{z \rightarrow z-1} + \left(\frac{N_H}{N_e}\right) \mathbf{C}_{CD}^{z \rightarrow z-1} & & 0 & 0 \\ \mathbf{S}_{CD}^{z-1 \rightarrow z} - \left(\mathbf{S}_{CD}^z + \mathbf{A}_{CD}^z + \left(\frac{N_H}{N_e}\right) \mathbf{C}_{CD}^z + \mathbf{Q}_{CD}^{z \rightarrow z} + \mathbf{X}_{CD}^{z \rightarrow z}\right) & & \ddots & & 0 & & \ddots & 0 \\ 0 & & & & \mathbf{S}_{CD}^{z \rightarrow z+1} & & \ddots & \ddots \\ 0 & & & & 0 & & \ddots & \ddots \end{bmatrix} \begin{bmatrix} \vdots \\ \vdots \\ N_z \\ N_{z+1} \\ \vdots \end{bmatrix} \quad (3.21)$$

where each bold faced letter in eq. 3.21 is an individual matrix describing reactions between the ground and metastable levels of the ionisation stages involved. Therefore, the ionisation coefficient matrix is given by,

$$\mathbf{S}_{CD}^{z-1 \rightarrow z} = \begin{bmatrix} \cdot & & \vdots & & \vdots & & \cdot \\ \cdots & S_{CD, \rho-1 \rightarrow \sigma-1}^{z-1 \rightarrow z} & S_{CD, \rho-1 \rightarrow \sigma}^{z-1 \rightarrow z} & \cdots & & & \\ \cdots & S_{CD, \rho \rightarrow \sigma-1}^{z-1 \rightarrow z} & S_{CD, \rho \rightarrow \sigma}^{z-1 \rightarrow z} & \cdots & & & \\ \cdot & & \vdots & & \vdots & & \cdot \end{bmatrix} \quad (3.22)$$

and the  $\mathbf{S}_{CD}^z$  matrix contains the summed values of the ionisation coefficients corresponding to ionisation out of the stage  $z$ , such that,

$$\mathbf{S}_{CD}^z = \begin{bmatrix} \cdot & & \vdots & & \vdots & & \cdot \\ \cdots & \sum_{\tau=1}^{m_z+1} S_{CD, \sigma-1 \rightarrow \tau}^{z \rightarrow z+1} & & 0 & & \cdots & \\ \cdots & 0 & & \sum_{\tau=1}^{m_z+1} S_{CD, \sigma \rightarrow \tau}^{z \rightarrow z+1} & & \cdots & \\ \cdot & & \vdots & & \vdots & & \cdot \end{bmatrix} \quad (3.23)$$

The recombination coefficient is given by,

$$\mathbf{A}_{CD}^{z \rightarrow z-1} = \begin{bmatrix} \cdot & & \vdots & & \vdots & & \cdot \\ \cdots & \alpha_{CD, \sigma-1 \rightarrow \rho-1}^{z \rightarrow z-1} & \alpha_{CD, \sigma \rightarrow \rho-1}^{z \rightarrow z-1} & \cdots & & & \\ \cdots & \alpha_{CD, \sigma-1 \rightarrow \rho}^{z \rightarrow z-1} & \alpha_{CD, \sigma \rightarrow \rho}^{z \rightarrow z-1} & \cdots & & & \\ \cdot & & \vdots & & \vdots & & \cdot \end{bmatrix} \quad (3.24)$$

and it also has a corresponding matrix which contains the summed components of

recombination out of the stage  $z$ , viz.

$$\mathbf{A}_{CD}^z = \begin{bmatrix} \cdot & \vdots & \vdots & \cdot \\ \cdots & \sum_{\rho=1}^{m_z-1} \alpha_{CD,\sigma-1 \rightarrow \rho}^{z \rightarrow z-1} & 0 & \cdots \\ \cdots & 0 & \sum_{\rho=1}^{m_z-1} \alpha_{CD,\sigma \rightarrow \rho}^{z \rightarrow z-1} & \cdots \\ \cdot & \vdots & \vdots & \cdot \end{bmatrix} \quad (3.25)$$

A similar definition applies for the  $C_{CD}$ 's.

For completeness, we should note also that the matrix containing the metastable cross-coupling coefficients also includes the summation of components populating the metastables of stage  $z$  by processes that depopulate those same metastables i.e. a transition out of a metastable of stage  $z$  will move into a metastable of stage  $z$ . Similarly, the matrix containing the parent to parent metastable cross-coupling coefficients includes the summation of components populating the parent metastables of stage  $z-1$  due to processes which depopulate those same parent metastables.

$$\mathbf{Q}_{CD}^{z \rightarrow z} = \begin{bmatrix} \cdot & \vdots & \vdots & \cdot \\ \cdots & -\sum_{\sigma'=1; \sigma' \neq \sigma}^{m_z} Q_{CD,\sigma' \rightarrow \sigma-1}^{z \rightarrow z} & Q_{CD,\sigma-1 \rightarrow \sigma}^{z \rightarrow z} & \cdots \\ \cdots & Q_{CD,\sigma \rightarrow \sigma-1}^{z \rightarrow z} & -\sum_{\sigma'=1; \sigma' \neq \sigma}^{m_z} Q_{CD,\sigma' \rightarrow \sigma}^{z \rightarrow z} & \cdots \\ \cdot & \vdots & \vdots & \cdot \end{bmatrix} \quad (3.26)$$

and

$$\mathbf{X}_{CD}^{z \rightarrow z} = \begin{bmatrix} \cdot & \vdots & \vdots & \cdot \\ \cdots & -\sum_{\sigma'=1; \sigma' \neq \sigma}^{m_z} X_{CD,\sigma' \rightarrow \sigma-1}^{z \rightarrow z} & X_{CD,\sigma-1 \rightarrow \sigma}^{z \rightarrow z} & \cdots \\ \cdots & X_{CD,\sigma \rightarrow \sigma-1}^{z \rightarrow z} & -\sum_{\sigma'=1; \sigma' \neq \sigma}^{m_z} X_{CD,\sigma' \rightarrow \sigma}^{z \rightarrow z} & \cdots \\ \cdot & \vdots & \vdots & \cdot \end{bmatrix} \quad (3.27)$$

Finally, to obtain the equilibrium metastable fractional abundances the normalisation (eq. 3.19) now becomes

$$N_{tot} = \sum_{z=0}^{z_0} \sum_{\sigma=1}^{m_z} N_{\sigma}^z \quad (3.28)$$

## Outline of Solution Method

We employ a progressive elimination algorithm to solve the tridiagonal matrix eq. 3.18. However, there is a practical problem. For a particular  $N^z$  we have,

$$N^z = N^{z+1} \left[ \alpha_{CD}^{z+1 \rightarrow z} + \left( \frac{N_H}{N_e} \right) C_{CD}^{z+1 \rightarrow z} \right] / S_{CD}^{z \rightarrow z+1} \quad (3.29)$$

which has the ionisation coefficient as the denominator. McWhirter & Summers (1985) give the following expression for the ionisation coefficient,

$$S_{CD}^{z \rightarrow z+1} = 16\sqrt{\pi}\alpha ca_0^2 \frac{(kT_e)^{1/2}}{(z+1)^4 I_H^{1/2}} \nu_p^4 F \exp\left(-\frac{(z+1)^2 I_H}{kT_e \nu_p^2}\right) \quad (3.30)$$

where  $\nu_p^z$  is the effective principal quantum number of the level p, of the ionisation stage z, from which the reaction takes place and  $F$  is a factor approximately equal to one. This equation includes an exponential dependence on  $-T_e^{-1}$  which results in a rapid ‘drop-off’ of the ionisation coefficient at low temperature. As a result of these extremely small values, it is possible to obtain *overflows* during the numerical solution of eq. 3.18. This generally occurs when attempting to form the population ratio of ionisation stages in 3.29 using the abundance of an ionisation stage formed at a radically different temperature. This problem would appear in reverse if the alternative elimination route was chosen. Therefore, to circumvent this problem the algorithm employs a ‘double pass’ technique. By simultaneously solving towards the predicted dominant ionisation stage, from the fully ionised and neutral stages, we can recover the population densities by back substitution in eq. 3.29 once the dominant stage’s actual value is known.

An equivalent algorithm is used for the *metastable resolved* case (eq. 3.21) but matrix manipulations are used to replace the usual arithmetic operations. A ‘first guess’ at the target dominant stage is made by a simple comparison of the effective ionisation and recombination coefficients. Selection is made at the ionisation stage where these coefficients are deemed to be most similar subject to the constraint that the stage is not too close to a region of rapid population ‘drop-off’. On solution of eq. 3.21 the equilibrium fractional abundances are checked and, if the dominant stage is wrong, the algorithm is re-run with the correct one.

## ADAS405 and scripts for observational sequences

The above solution method for eq. 3.20 has been implemented in ADAS series 4, as ADAS405 (Summers et al.(1996)). ADAS405 computes the equilibrium population distribution amongst the ionisation stages of an element, in both the metastable resolved and unresolved approximations. In addition, it calculates the radiated power function,  $P_{tot}$  (see eq. 3.16), and also the contribution function,  $G(T_e, N_e)$  (see eq. 2.64) for a temperature and density model. As such, it provides a complete capability for exploring the temperature and density dependence of these quantities as well as the ionisation balance and the influence of metastables.

ADAS405 draws from collections of the collisional-radiative coefficients required to solve eq. 3.21. These are stored in adas data format files *adf11*. Selection of these is made to be consistent with the intention of the calculation, e.g it is not necessary to include metastable cross-coupling data if the computation is to be unresolved.

Computation of the line emission functions,  $G(T_e, N_e)$ , is controlled by a user specified '*script file*'. This allows us to explore emission function ratios between the specified lines, and also between lines originating from different ions of the same element. A sample script file showing the OV and OVI lines associated with the suggested SOHO observing sequence of McWhirter (PTCOR in Harrison & Fludra (1995) also known as the Blue Book) to measure the coronal temperature versus height, is given in fig.3.21. The script specifies the element (SPECIES), the number of lines (NLINE) and the number of line ratios (NRATIO), for the study. ADAS405 accesses photon emissivity coefficients from the ADAS database (see chap.2) and the names of the files for the study are entered below NFILE. The coefficients are stored in ADAS datafiles of type *adf15*. NFILE itself is the number of files being accessed. ILINE indexes the number of lines, NCOMP is the number of component parts of the line, IZION is the charge to which the components belong and ICOMP indexes the components. IMET indexes the the metastables of the ion to which the component is attached and INDPH is the selection index of the component photon emissivity coefficient in the *adf15* file. The letter 'E', to the right of this index, identifies the coefficient as being driven by electron collisions or charge exchange if it is replaced by an 'H'. The file in the list, to which the index INDPH refers, is given by IFILE.



The TITLE field is usually filled with the ion, in spectroscopic notation, followed by the wavelength of the emitting line. IRATIO indexes the number of line ratios in the *script* while ILINE and JLINE give the line indices of the numerator and denominator of the ratio to be formed. There is an arbitrary information field at the bottom of each *script* which is for the convenience of the user.

Users should create their own script file suitable for analysing the spectral emission from whichever ion is of interest to them. The expectation is that the lines and line ratios present in the script file will reflect those chosen for a particular observing sequence using SOHO-CDS or SOHO-SUMER. Therefore, a direct comparison can be made between the predicted emission and the actual observations, once the observing study has been run.

### Validity of Equilibrium Assumption

It has been usual, in astrophysics, to assume that steady state ionisation balance is a reasonable approximation when investigating radiation from the outer atmosphere of the Sun. However, this hypothesis is unsound when looking at dynamic activity, such as flares, and may also be invalid for the quiet Sun.

The steady state assumption is only valid if the atomic relaxation timescales, for the ground and metastable states, are short enough to allow the plasma to reach equilibrium. We can make a simple estimate of this timescale by looking at the summation, over all ionisation stages of an element, of the reciprocal of the sum of the collisional dielectronic recombination and ionisation coefficients. We call this  $\tau_{transient}$  and define it as

$$\tau_{transient} = \sum_{z=0}^{z_0} \frac{1}{N_e(S_{CD}^{z \rightarrow z+1}(T_e) + \alpha_{CD}^{z+1 \rightarrow z}(T_e))} \quad (3.31)$$

Hence, we can obtain an estimate for the minimum amount of time required for a particular species to relax into equilibrium in a plasma of given electron temperature and density. In this simple approximation, we have assumed that the rate coefficients are independent of density. This is a reasonable assumption since they are generally only slowly varying.

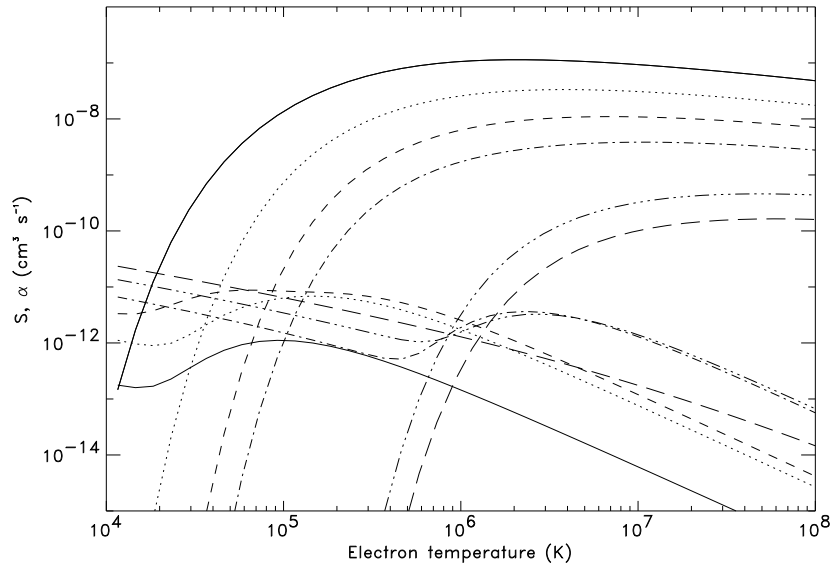


Figure 3.22: Collisional-Dielectronic Recombination and Ionisation coefficients for carbon ions,  $N_e = 10^{10} \text{ cm}^{-3}$

Fig. 3.22 shows the collisional dielectronic recombination and ionisation coefficients for each ionisation stage of carbon, plotted over a range of electron temperatures at an electron density of  $10^{10} \text{ cm}^{-3}$ . The data was taken from Abel-Van-Maanen (1982) Recombination and ionisation processes balance where  $S_{CD}^{i \rightarrow i+1} = \alpha_{CD}^{i+1 \rightarrow i}$  i.e. where each line of the same style crosses. McWhirter (1960) treated the minimum  $\tau_{transient}$  as being independent of the recombination coefficient and, therefore, controlled by the ionisation coefficient for the highest ionisation stage present in the plasma; since this would provide the largest term in the summation expansion. However, if we wish to include the possibility that the plasma may be recombining, we use eq. 3.31. Fig. 3.23 shows the variation with temperature of  $N_e \tau_{transient}$ . The behaviour of the curve indicates the temperature regions in which carbon is ionising or recombining in the plasma. For a fixed electron density, the time taken to relax into equilibrium diminishes with increasing temperature as the element ionises from the neutral stage through to the He-like one. Then, as the He-like stage dominates

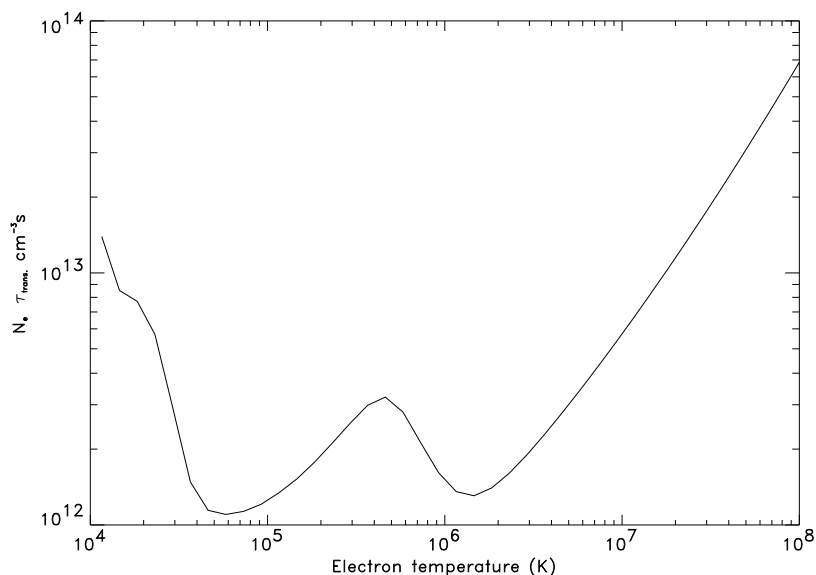


Figure 3.23: Function  $N_e \tau_{transient}$  for Carbon

and the plasma begins to recombine, the time increases with temperature until it is hot enough to strip open the closed  $n = 1$  shell. Thereafter, the timescale drops until the ion is fully stripped when recombination again dominates. A similar dip and rise would be expected at the Ne-like stage, for an element of nuclear charge  $\geq 10$ , in order to open the  $n = 2$  shell. Dividing by typical electron densities for the solar atmosphere, allows us to identify approximate relaxation timescales for the different temperature/density regimes.

The same type of analysis can be made for other ions and fig.3.24 shows a general plot of  $N_e \tau_{transient}$  against  $T_e$  for Beryllium, Carbon, Nitrogen, Oxygen and Neon. Broadly similar behaviour can be seen for each element. However, the region, at which the effects of the He-like stage set in, shifts to a higher temperature for elements of higher nuclear charge. This would be reflected in the width of the He-like stage, in the ionisation balance, at correlated electron temperatures.

For a particular plasma environment, we expect *transient* or *nonequilibrium* effects to be important if the time to pass through a temperature scale height is shorter

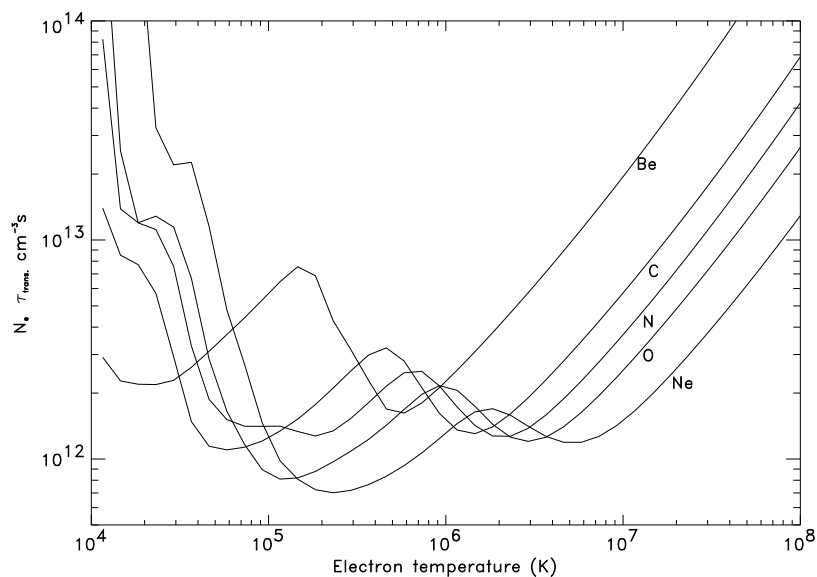


Figure 3.24:  $N_e \tau_{transient}$  for some prominent radiators in the solar atmosphere

than  $\tau_{transient}$  (Noci, Spadaro, Zappala & Antiochos (1989)). If this is true, the left hand side of eq. 3.20 cannot be set to zero and the problem becomes more complex, requiring a suitable choice of boundary conditions to solve the differential equations. Noci et al.(1989) did just this. They evolved carbon ions through a temporal variation of plasma electron density, temperature and velocity computed by a numerical hydrodynamic model of a steady siphon flow along a coronal magnetic loop. They computed the population distribution amongst the ionisation stages and their results show that, even for plasma flows of only a few  $\text{km s}^{-1}$ , nonequilibrium effects can make an appreciable difference.

It should be noted that these differences are not necessarily better able to resolve the discrepancies between theoretical models and spectroscopic observations. Spadaro, Noci, Zappala & Antiochos (1990) continued the work of Noci et al.(1989), computing UV emission lines for carbon from their siphon flow model. This was an attempt to explain the persistent red shifts observed in some transition region lines (Doschek, Bohlin & Feldman (1976)). Then, Spadaro, Antiochos & Mariska (1991)

	$N_e(cm^{-3})$	$T_e(K)$	$\tau_{transient}(s)$
upper chromosphere	$3.7 \times 10^{10}$	$2 \times 10^4$	$\sim 160$
transition region	$1.3 \times 10^{10} \rightarrow 1.3 \times 10^9$	$5 \times 10^4 \rightarrow 5 \times 10^5$	$\sim 75 \rightarrow 2300$
corona	$5.4 \times 10^8$	$1 \times 10^6$	$\sim 2800$

Table 3.1: Carbon relaxation times for the solar atmosphere

extended the research to include oxygen and investigated an asymmetrically heated magnetic loop, with and without the assumption of ionisation equilibrium. Generally they found that nonequilibrium tended to alter and smooth out the ion density distribution through the steep temperature gradients in the loop. Also, they found that the calculated line widths were changed and that the doppler shifts were significantly reduced; by greater than a factor three for OVI ( $2p^2P_{3/2} \rightarrow 2s^2S_{1/2}$ ) $\lambda 1031.9\text{\AA}$ . However, in the case of Spadaro et al.(1990), the main results predicted that blue shifts would dominate the emission lines. Also, in Spadaro et al.(1991), they conclude that the nonequilibrium results were less able to account for the observed red shifts than the equilibrium ones.

Transient effects should certainly play a role in violent events with sudden energy releases and large temperature gradients, but, it is an open question whether the ‘quiet Sun’ is in a constantly dynamic state. It is worthwhile to take representative solar parameters and use them to investigate quiet sun relaxation timescales. As an illustration consider table 3.1. This shows approximate values of  $\tau_{transient}$ , for carbon, from the chromosphere through the transition region to the corona. The solar parameters were taken from Gabriel (1976) and the suggested boundaries and regions from Athay (1981). Clearly the effect of the He-like recombination phase plays a role in lengthening the relaxation time in the transition region, despite an initial downward trend faster than the density drop trying to offset it. In the diffuse corona the density effect eventually wins causing long relaxation times even though  $N_e\tau_{transient}$  has entered its final rising phase. We can conclude that fluctuations and dynamics on timescales shorter than those shown would suggest that carbon ions should be treated as being out of equilibrium for the plasma in question. Similar

results would be expected for the other species shown in fig. 3.24. Recent evidence from SOHO-CDS & SOHO-SUMER suggests that timescales, for quiet sun activity, may indeed be shorter than those given (see chap.5).