ADAS-EU R(10)SC03

ADAS-EU

ADAS for fusion in Europe

Grant: 224607

Hugh Summers, Martin O'Mullane, Francisco Guzman and Luis Menchero

Scientific progress report 3

7 July 2010

Workpackages : 1-3, 2-2, 3-1, 3-2, 3-4, 6-2, 6-3, 13-1, 16-2, 18-1, 18-2, 18-3, 19-1, 22-2-1, 23-2-1, 22-2-2,
Category : PP

Scientific progress report 3

Hugh Summers, Martin O'Mullane, Francisco Guzman and Luis Menchero

Department of Physics, University of Strathclyde, Glasgow, UK

Abstract: *The report reviews scientific task completion for project months 13-18*

Contents

Preface

This scientific report is the third of a series of six such reports, deliverable under the ADAS-EU project, which summarise the scientific achievements of the project over the preceding six months.

H P Summers
7 July 2010

Chapter 1

Overview and milestones SCI11, SCI24, SCI51, SCI52, STS1 [+ SCI12, SCI13, SCI41, SCI43, SCI44]

Milestone SCI11 concerns baseline heavy species generation and complex heavy species spectral management. This has been completed. Details are in the heavy species theme document PUBL_3 (see Appendix A). Minimal release of baseline data for elements Xe, Sn, Ag and W was scheduled for month 16. Data for these elements have been generated and are in the ADAS data archives. Selections of these data have been released to interested parties. Full release will be with ADAS version 3.1 due in August 2010 (month 20) as increased ADAS user capacity at member sites is made available. The full heavy species baseline capacity, now available for any element, if run to completion up to radon and released *in extenso*, would add ~ 15 Gbytes to the database.

Milestone SCI24 concerns the NEW-CHEAP analysis package. This has been deferred (see the forward planning discussion in report REVIEW1 section 1.3).

Milestones SCI51 and SCI52, concerning H_2 database and vibronic population modelling, are not met at this stage. The data base transfer is progressing, but as advised in report SCIENCE2, requires substantially more work than originally expected. The original ADAS-EU plan was for the transfer and maintenance under ADAS of a stable H_2 vibronic model of Fantz and Wunderlich. The orginators have resumed work on their model and its continued development and are maintaining their proprietary interest in it. Thus it is no longer available and suitable for transfer into ADAS. ADAS-EU staff (Dr. Guzman, guided by Professor's Summers and Janev) are building a pedagogical/ application ADAS collisional-radiative model tuned to ADAS data formats and spectral analysis capabilities. This model will draw the new ADAS molecular data formats with updated content through Professor Janev. The delay in PDRA setup and the consequential one-year shift in Professor Janev's ADAS-EU appointment has put a one year delay on milestones SCI51 and SCI52. Validation and completion is expected in the 2011 work period of Professor Janev for ADAS-EU - probably at EFDA-JET in June 2011.

Milestone STS1 concerns on-site support activity examination and adjustments at FZ Juelich and IPP Garching. This has been completed for FZ Juelich. The seven month delay in the appointment of Dr. Menchero at IPP Garching means that the on-site appraisal for IPP Garching is moved back to ~ month 26.

Milestones SCI12 and SCI13 have been met substantially in advance of schedule (months 22 and 25 respectively). The superstage compression method has been put in place although it is noted that the effectiveness of the method with quite aggresive bundling is so high that it has not been necessary to release the dynamic partition capability at this stage, even though the developmental work has been done for it. The superstaging for tungsten has been taken on-board enthusiastically by the plasma modelling community, such that superstaging in now incorporated in the main 2-D transport codes such as EDGE2D and SOLPS. A number of studies and conference papers in the last year examine its implications and effectiveness. The mass ionisation rate production at level 1 for heavy elements is operational and released in ADAS. This leaves just milesone SCI14 for BBGP (scheduled for month 41) remaining in the heavy species theme 1. In fact this is already in test operation, so theme 1 is nearly complete. The primary document scientific/technical document PUBL3 (see Appendix A) is now nearly in its final version and milestone SCI14 part of the write-up (section 3.2) is already in place within it.

It has been possible to move ahead strongly with the special feature (theme 4) work packages. Milestones SCI41 and SCI43 (scheduled for months 28 and 35) are completed. The full interactive special feature software system comprises two parts, called AFG and FFS. AFG was made available with ADAS release version 3.0 in Sept. 2009 - see bulletin item 4 (Appendix B[1]). FFS is now under undergoing heavy testing and the complete system including the fitting is working well and robustly. Only milestone SCI42 remains to be done. Completion and readiness for full release is expected ~ month 22. The scientific/technical write-up, document PUBL2 is well advanced. It is available for preview and its contents pages are attached as Appendix B[2].

Chapter 2

Work package reports

2.1 Work package 1-3

Central ADAS baseline data for Kr, Xe, Sn, Ag and W has been generated. It will be released into the public domain at the next ADAS release version 3.1. [2].

2.2 Work package 2-2

Work package ”Managing spectral complexity” - task 2 (automatic sizing) has been completed (see Appendix A section 2.4).

2.3 Work package 3-1

Work package ”Fitting to impurity transport codes” - task 1 (superstage compression code) has been completed and released (see Appendix A section 4).

2.4 Work package 3-2

Work package ”Fitting to impurity transport codes” - task 2 (automatic partition creation) has been completed and released for the natural partition. The dynamic partition capability has been developeed but the priority for release has been reduced as discussed in Chapter 1, paragraph 5 above (see also Appendix A section 4.1).

2.5 Work package 3-4

Work package ”Fitting to impurity transport codes” - task 3 (exploitation in plasma models) has progressed very rapidly and successfully as discussed in Chapter 1, paragraph 5 above.

2.6 Work package 6-2, 6.3

The NEW-CHEAP workpackage tasks have been deferred as mentioned in Chapter 1, paragraph 2 above (and discussed in more detail in report REVIEW1 section 1.3).

2.7 Work package 13-1

Work package ”Embed Zeeman feature primitives and display for special feature handler” has been completed ahead of schedule. See 16-2 below and comments in Chapter 1.

2.8 Work package 16-2

Work package ”Unified special feature handling - enbed special feature objects in pedagogical parametric active display” has been completed ahead of schedule and released - see comments in Chapter 1. Further details are in Appendix B section 1, item 4 and PUBL2 - see Appendix B section 2).

2.9 Work package 18-1, 18-2, 18-3, 19-1

The workpackages have been revised and rescheduled (see chapter 1 paragraph 3 above).

2.10 Work package 22-2-1, 22-2-2

Update training completed for PDRA1 (see report SETUP2). Update training for PDRA2 will take place in month 26 (due to seven month delay in PDRA appointment).

2.11 Work package 23-2-1, 23-2-2

Local review completed for PDRA1 at FZ Juelich (see report SETUP2). Local review for PDRA2 at IPP Garching will take place in Spring 2011 (due to seven month delay in PDRA appointment).

2.12 Work package 26-1-3

The work package task comprises the completion of this report.

2.13 Supplementary work package completion

As discussed in Chapter 1, paragraph 5, workpackage tasks 3-1, 3-2, 3-3 and 4-1 have been completed in advance. As discussed in Chapter 1, paragraph 6, workpackage tasks 13-1, 13-3, 15-1, 16-1 and 16-2 have been completed in advance.

2.14 Work package S2

Expected completion Jan 2011. Programmatic connection of Mons-Hainaut and ADAS structure modelling codes has now been made. ADAS postprocessing to adf04 data sets of Mons-Hainaut WII data has been successful.

2.15 Work package S3

Expected contract signing Jul. 2010 and completion Jan. 2012.

2.16 Work package S5

Expected completion Jan 2011. First data transfer (ion impact excitation cross-section data) to ADAS has taken place

2.17 Work package S8

Expected completion Apr. 2011.

-

Appendix A

ADAS-EU Theme 1 supplementary material for the report

[1] ADAS-EU/REPORTS_PUBL/PUBL_3/ pages 1-6

-

ADAS-EU R(09)01

ADAS-EU
ADAS for fusion in Europe

Grant: 224607

Hugh Summers, Adam Foster, Stuart Loch, Martin O'Mullane
and Allan Whiteford

Heavy species in fusion plasma modelling and spectral analysis

25 May 2010

Workpackages :
Category :

Heavy species in fusion plasma modelling and spectral analysis

Hugh Summers, Adam Foster, Stuart Loch, Martin O'Mullane
and Allan Whiteford

Department of Physics, University of Strathclyde, Glasgow, UK

Abstract: *The derived data for usual application in fusion, from the perspective of light elements, are hugely unwieldy for heavy elements, preventing the immediacy and handleability to which ADAS aspires for the experimental diagnostic analyst. So additional work must be done within ADAS in the direction of spectral synthesis for the spectroscopist and in the direction of enhanced condensation for the plasma computational modeller. The purpose of this article is to put all these steps in the hands of the ADAS user who wishes to become expert and, for the more application oriented user of ADAS, to explain what is available now for heavy species in ADAS, how to access it and use it correctly.*

Contents

Appendix B

ADAS-EU Theme 4 supplementary material for the report

[1] ADAS_release_v3.0_bulletin_25sep09
[2] ADAS-EU/REPORTS_PUBL/PUBL_2/ pages 1-6

-

Sep25-09 **ADAS Bulletin**

Items:

1. ADAS release v3.0
2. An integrated heavy species modelling capability for ADAS
3. ADAS Communications and Reports
4. The ADAS Feature Generator and ADAS605
5. Code and data updates in release v3.0.

1. This new bulletin has the usual long list (more than seventy this time) of code updates, but some of these entries are in fact large new extensions of the ADAS capabilities – especially C15/C16 and C41/C42 – and justify the fact that this release takes us to v3.00. Thus C15/16 brings in the first part of the whole special feature handling development, 'ADAS Feature Generation' - AFG. It will be another year before the second part, concerning optimised fitting, is fully in place along side this first pedagogical part. More on AFG later. The other very large addition is the comprehensive heavy species handling.

2. Let me turn first to the heavy species for fusion plasma modelling and spectral analysis. I can only give a brief outline here and hopefully catch your interest. We are providing a complete capability for creating a baseline of atomic data for arbitrary heavy species together with all the derived data required for plasma modelling and spectral analysis. Most of the principles and ideas behind the approach have been signalled over the last two years at the ADAS Workshop. The complexity and potentially overwhelming size (levels, transitions etc.) of heavy atoms force us to focus on 'resolution' levels in the establishment of a baseline of atomic modelling data. The ADAS implementation deals with three resolution levels of increasing precision, but also increasing demand on computer resources - 'configuration average' (*ca*), 'LS coupled' (*ls*) and 'intermediate coupled' (*ic*). In configuration average, it is possible to execute calculations with many configurations included (*cl*) as well as smaller calculations with fewer configurations (*ca*) matching the number of configurations which can be included in *ic* or *ls*. *ic* calculations for heavy species ions may be severely limited by computer resources. So in the ADAS database you can expect to see data sets with names like *cl#<elsym><z_ion>.dat*, *ca#<elsymb><z_ion>.dat*, *ls#<elsymb><ion charge>.dat* and *ic#<elsymb><z_ion>.dat*. We use the fact that *cl* results can 'top-up' *ic* calculations (for example of power) maintaining accuracy and reliability of global properties. Already you will be familiar with the ADAS use of year numbers. Often these represent the year a data set was prepared, but in systematic semi-automatic method data production, the year number can correspond to the year of introduction of a method. Here, for heavy species mass production, we have chosen rather arbitrarily the historic year number '40'. This will allow us to use '41', '42' etc. for uplifts of the heavy species datasets in the future without causing confusion. So look for adf04 datasets like

/home/adas/adas/adf04/coparf#<iz0>/arf40_<resolution>#<elsymb><z_ion>.dat

and adf15 datasets like

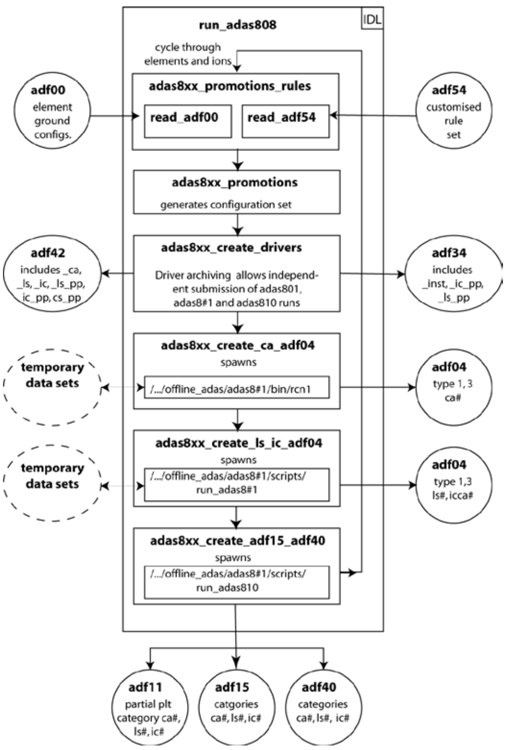
/home/adas/adas/adf15/pec40#<elsymb>/pec40#<elsymb>_<resolution>#<elsymb><z_ion>.dat

and so on, where *iz0* is the element nuclear charge, *elsymb* is the element symbol and *z_ion* is the ion charge. In a general sense, the ADAS objectives for heavy elements are the same as for light elements, namely fundamental adf04 data sets containing atomic structure, transition probability and collisional data suitable for population calculations, recombination (adf08, adf09 or parametric approximations) and ionization (adf07, adf23 or parametric approximations) for ionization state calculations. Then from these, the production of the derived data formats adf11, adf15 and adf40 for application must be enabled. The difference for heavy elements is the sheer scale of the problem and the need to automate and regulate the size according to computer resources, hopefully without too much loss of accuracy.

To automate the atomic structure calculations, we have introduced promotion rules for all ions of every element, data format adf54 (with similar adf55 and adf56 formats for recombination and ionization respectively). Also we have provided an optimization method for regulating these promotional rule data sets to available computer resources. Then at the IDL command line, we have provided procedures to execute the various production steps in a hands on manner and also integrated procedures and scripts for complete automation (online or offline) of the process. The following schematic shows the pattern. Similar schematics describe offline and batch processing. Again, somewhat similar schematics describe promotion rule optimization, recombination and ionization.

All of the above means quite a few new ADAS data formats, many new IDL and FORTRAN procedures and subroutines along with various shell scripts. Describing all of these is outside the scope of this bulletin. However there is a report "Heavy species in fusion plasma modelling and spectral analysis" which hopefully explains all. Unfortunately, with all its appendices, it is around 250 pages long. I shall be making this report available in /docs and through the ADAS web site shortly after the 2009 ADAS Workshop/ADAS-EU Course.

Heavy species unfortunately means very large amounts of data. I am afraid the ADAS data bases are going to increase dramatically. It is probably a useful point of discussion at the ADAS Workshop if the ADAS releases should continue to include everything automatically or whether some sites may wish just a subset. To give time to adjust, this release we are putting up data for a few new elements. For the heavy species ionization part (see D20 below), we have put in data for Mg, Si, Ar, W to assist both fusion and astrophysics and and are holding off on others. For the heavy species excitation and line power part (see D21 below) we have put up data for Ar and W. (I note that Kr, Ag, Sn and Xe are already in our ADAS development space and any other element is relatively easy to generate). But, such new data requires exposure to multiple users before we commit to too much.



3. This brings me conveniently to the issue of ADAS Communications and ADAS Reports. It has been apparent for some time that studies carried out in reponse to user engagement with the ADAS team, technical notes associated with ADAS developments and user manuals for items in ADAS or EXTENDED-ADAS should be more visible, more plentiful and more organized. We plan that these

documents will have a standard appearance, similar to the ADAS website first page and be accessible under Notes. They will be labelled as

	"ADAS-Communication	ADAS-CM(09)01"
or		
	"ADAS-Report	ADAS-R(09)01"

and they will include a standard disclaimer/permissions for us/copyright notice. Implementation commences with this release v3.0

4. The ADAS Feature Generator is the brain child of Christopher Nicholas and Allan Whiteford and is the first part of their integrated handling and fitting of special spectroscopic features. AFG provides an applications programming interface (API) to existing ADAS feature generation routines. The interface provides a common way to access and control the underlying routines in ADAS. AFG imparts enough information about the special features such that the ADAS605 GUI-based program can generate a custom control panel for each of the currently supported features and should easily accommodate future inclusions, without the need for further GUI development.

Upon selecting ADAS605 from the series 6 menu, you are presented with a simple input screen (Fig. 1) with a dropbox allowing selection of the feature of interest. A short description of the currently selected feature is given in the textbox below the dropbox.

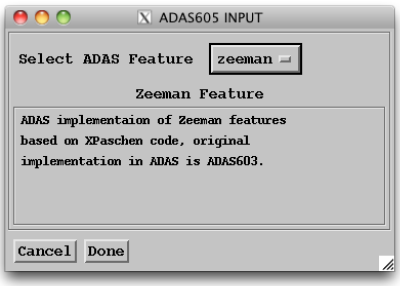


Fig. 1: ADAS605 input screen: feature selection.

The processing screen (Fig. 2) is split into two main segments; the left hand side is consistently the same regardless of the feature selected - it is a graphical display area, the right hand side is comprised of a set of control widgets to alter the special feature parameters and will therefore adapt to the particular feature selected from the input screen.

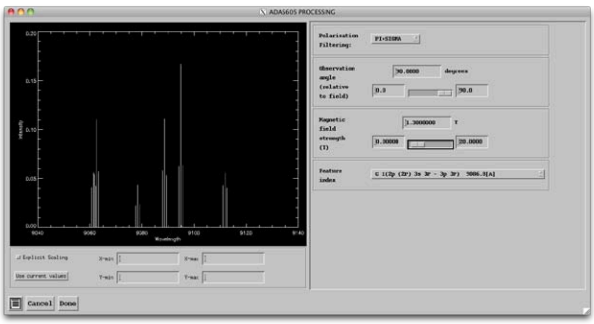


Fig. 2: ADAS605 processing screen: allows interactive manipulation of chosen feature via custom control widgets in right hand panel, with graphical output in left panel.

The plot window will update (in most cases in real-time) in response to changes of feature parameters and will re-scale the plot automatically. It may be desirable to keep a specific, fixed scale as parameters are altered and, in this case, the `explicit scaling' check-box should be checked (which will activate the X-Y min/max textboxes). The `use current values' button will auto-fill these textboxes with the current X-Y min/max values.

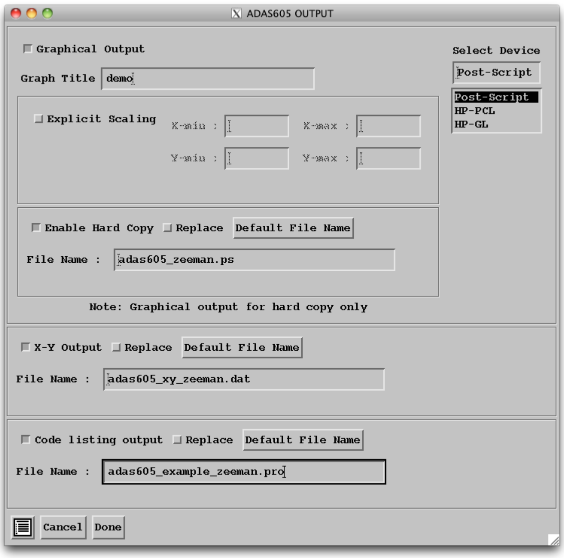


Fig. 3: ADAS605 output screen: option to produce three types of output - graphic, X-Y plot data and finally, output of IDL source code that will recreate the feature as seen in the interactive window.

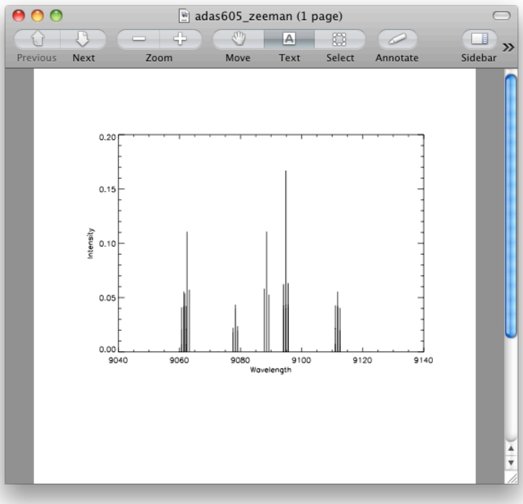


Fig. 4: ADAS605 `graphical output'.

The ADAS605 output screen (Fig. 3) follows the usual format i.e. a set of optional output types, each with the familiar `replace', `default file' and `file names', checkbox, button and textbox respectively for specifying the output file. The output options available are `graphical output' - saving the plot window as a graphic (postscript in the example Fig. 4), `X-Y output' - the plot data in a plain text file as co-ordinate pairs (Fig. 5) and finally, `code listing output' - AFG will auto-generate the appropriate IDL source code (including in-line comments) to generate the feature using the API directly, rather than via the GUI (Fig. 6). It is envisaged that this template source will serve as an entry point to most users looking to utilize AFG in their own codes. As can be seen from the output screenshot, AFG is an object-oriented program and therefore utilises the IDL object syntax. However, the program can also be utilised through a wrapper program that operates in a familiar, procedural, fashion.

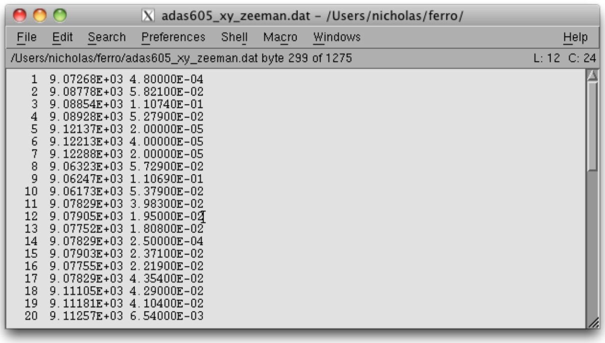


Fig. 5: ADAS605 `X-Y output'.

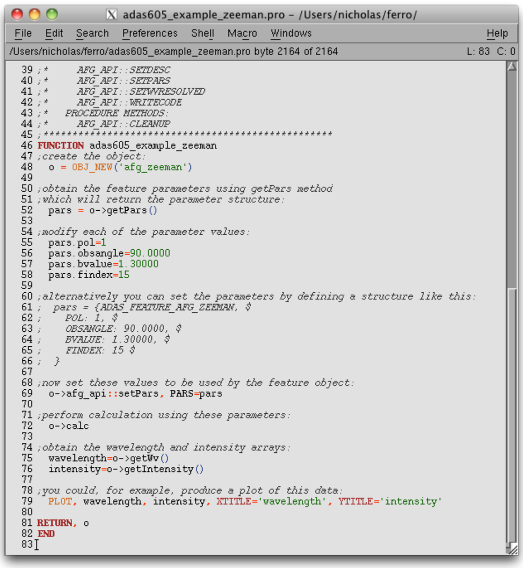


Fig. 6: ADAS605 `code listing output'.

It is very easy to obtain a plot similar to that shown in the ADAS605 interaction, from the command line. Staying with the Zeeman feature example, an interaction with the program may be as follows (command line compilation statements removed):

IDL> *pars = afg('zeeman', /parameters)*
IDL> *pars.bvalue = 1.3*
IDL> *pars.findex = 15*
IDL> *result = afg('zeeman', calculate=pars)*
IDL> *plot, result.wv, result.intensity, /nodata*
IDL> *for i=0, n_elements(result.wv)-1 do $*
IDL>   *oplot, [result.wv[i], result.wv[i]], [0.0,result.intensity[i]]*

which produces the output you see in Fig. 7.

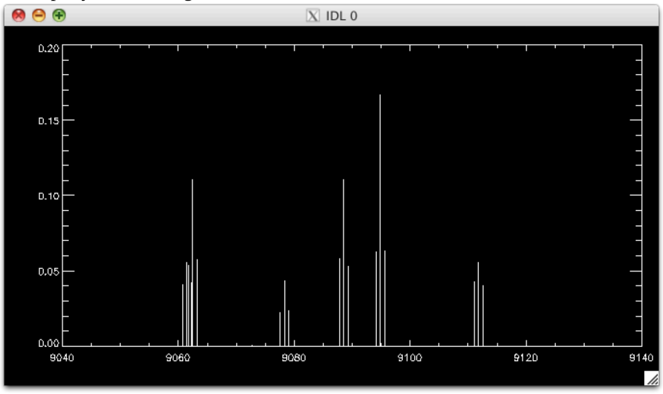


Fig. 7: Plot as produced by command line input example.

In order to understand the above interaction with AFG more fully, it is better to consider a more thorough interaction with the system.

First query AFG for a list the currently available features:

IDL> *print, afg(/list)*
stark zeeman hdlike

It is then possible to request a description of one of the listed features:

IDL> *desc = afg('zeeman', /description)*
IDL> *help, desc, /str*
** Structure <a0c409c>, 3 tags, length=1060, data length=1060, refs=1:
   NAME   STRING   'Zeeman Feature'
   TEXT   STRING   'ADAS implementaion of Zeeman features base'...
   PARAMETERS   STRUCT   -> <Anonymous> Array[1]

Now examining `parameters' specifically:

IDL> *help, desc.parameters, /str*
** Structure <a0c2c14>, 4 tags, length=1036, data length=1036, refs=2:
   POL   STRUCT   -> <Anonymous> Array[1]
   OBSANGLE   STRUCT   -> <Anonymous> Array[1]
   BVALUE   STRUCT   -> <Anonymous> Array[1]
   FINDEX   STRUCT   -> <Anonymous> Array[1]

The user is able to examine a particular parameter; picking `bvalue' as an example:

```
IDL> help, desc.parameters.bvalue,/str
** Structure <a0c2fec>, 8 tags, length=60, data length=60, refs=2:
   DESC            STRING    'Magnetic field strength (T)'
   TYPE            STRING    'float'
   UNITS           STRING    'T'
   MIN             FLOAT          0.00000
   MAX             FLOAT          20.0000
   DISPTYPE        STRING    'continuous'
   LOG             INT              0
   ALTERSLIMITS    INT              0
```

We can see that this gives the user a more complete description of what the parameter is, the expected data type, units, upper & lower limits for the parameter value, whether the feature has a logarithmic dependence on it and whether changing it can potentially alter the limits of other parameters for that feature.

To get the parameters themselves,

```
IDL> pars = afg('zeeman', /parameters)
IDL> help, pars, /str
** Structure <a0c5944>, 4 tags, length=16, data length=12, refs=1:
   POL             INT              1
   OBSANGLE        FLOAT          90.0000
   BVALUE          FLOAT          2.50000
   FINDEX          INT             15
```

So, we can see that `pars' in the original plotting example was just a structure of the parameter values. The values in this structure were altered and then simply passed in via the keyword `calculate' to have AFG evaluate the feature and the result plotted.

5. The list of code and data updates in v3.00 follows:

Corrections and updates to code (ADAS v2.13 to ADAS v3.00)

C.1 An IDL routine to write adf42 datasets, with input data in form of adf42 fulldata structure, has been added: *write_adf42.pro*.

C.2 Fixed bug in ADAS405 where a lack of the implicit SAVE feature in g77 caused problems with running resolved datasets.

C.3 Fixed bug in ADAS405 where LRSPEC wasn't being set to FALSE by default by g77.

C.4 Increased NDMET to 5 in ADAS412 *xcoef.for*

- Do not print warning messages in the middle of the ADF20 file in ADAS412
- Write a final '1' back to IDL just before exiting ADAS412 to allow files etc. to be flushed before IDL frees up the pipe.

C.5 Added dcnhne to return solar NH/NE abundancies.

C.6 Removed warning from *xxpars.for* which complained about parent being missing and 1S being forced. Also added extra check that ndmet was high enough for file being read.

C.7 Added *run_adas412*

C.8 Allowed for more than 32,768 transitions in ADAS812

C.9 Made *pers_f.c* slightly more robust (catch for failed entry lookups and blank names)

C.10 Modified ADAS405 to write ADF16 files with parameters ordered as specified in the documentation and as read by ADAS507. ADAS406 had the same fault and is also updated.

C.11 Initialise LRSPEC variable in ADAS405 and also pass further communication between IDL and Fortran to allow for buffers to be flushed etc.

C.12 Initialise LRSPEC variable in ADAS409.

C.13 Corrected documentation of *fortran/adaslib/maths/xxbasa.for*

C.14 Corrected faulty logic in *run_adas208* which meant that projection was always turned on. (Note: has minimal implications to anything processed using *run_adas208*).

C.15 Added AFG (ADAS Feature Generation) library to access ADAS special features in a uniform way. Initial features included are from ADAS603 (Zeeman) and ADAS305 (Stark).

C.16 Added ADAS605 - a graphical front-end for AFG allowing pedagogical exploration of AFG features.

C.17 Add *adas_writefile.pro*, a companion routine to *adas_readfile.pro* to write, or append, a string array to an output ascii file.

C.18 Corrected *fortran/adas8xx/adas801/ifg/orbital.for*
fortran/adas8xx/adas801/ifg/a04filter.for
offline_adas/adas8#1/adas801/ifg/orbital.for
offline_adas/adas8#1/adas801/ifgpp/a04filter.for
updating length of string to 296 to allow 36 orbitals.

C.19 Updated *fortran/adas8xx/adas801/ifg/ifgpp.for* to allow for 36 orbitals. Also updated to match offline version, with adjustable string formatting.
fortran/adas8xx/adas801/ifg/termread.for
fortran/adas8xx/adas801/ifg/levelread.for
Brought into line with offline version: carry Sval of level in unused part of label.

C.20 Updated *fortran/adas8xx/adas801/include/ifg.h*
fortran/adas8xx/adas801/include/ifgpp.h
fortran/adas8xx/adas801/include/rcg.h
offline_adas/adas8#1/adas801/include/ifg.h
offline_adas/adas8#1/adas801/include/rcg.h
offline_adas/adas8#1/adas801/include/ifgpp.h
offline_adas/adas8#1/adas801/include/rcn.h

with new dimensions for heavy species runs. Introduced ksjk, dimension

of nsjk, to *rcn.h*
Updated:
fortran/adas8xx/adas801/rcg/cafcdolp.for
fortran/adas8xx/adas801/rcg/calcfc.for
fortran/adas8xx/adas801/rcg/calcv.for
fortran/adas8xx/adas801/rcg/cpl37.for
fortran/adas8xx/adas801/rcg/elecden.for
fortran/adas8xx/adas801/rcg/energy.for
fortran/adas8xx/adas801/rcg/mlew.for
fortran/adas8xx/adas801/rcg/mupole.for
fortran/adas8xx/adas801/rcg/rceinp.for
fortran/adas8xx/adas801/rcg/sprin.for
fortran/adas8xx/adas801/rcg/sprn37.for
fortran/adas8xx/adas801/rcg/sprnadd.for
offline_adas/adas8#1/adas801/rcg/cafcdolp.for
offline_adas/adas8#1/adas801/rcg/calcfc.for
offline_adas/adas8#1/adas801/rcg/calcv.for
offline_adas/adas8#1/adas801/rcg/cpl37.for
offline_adas/adas8#1/adas801/rcg/elecden.for
offline_adas/adas8#1/adas801/rcg/energy.for
offline_adas/adas8#1/adas801/rcg/mlew.for
offline_adas/adas8#1/adas801/rcg/mupole.for
offline_adas/adas8#1/adas801/rcg/rceinp.for
offline_adas/adas8#1/adas801/rcg/sprin.for
offline_adas/adas8#1/adas801/rcg/sprn37.for
fortran/adas8xx/adas801/rcg/sprnadd.for
re-specifying dimension of nsjk as ksjk.

C.21 Updated *fortran/adas8xx/adas810/hapecf.for*
Changed ndpec 1500 -> 380000 (required for heavy species calculations)
Changed ndtrn 20000 -> 380000
Removed reference to unused subroutine *hawinf* in comments

C.22 Updated *fortran/adas4xx/adas408/d8eval.for*
No longer using wrong shell energy in case B Dielectronic
Recombination Power (POWDRC)

C.23 Corrected *fortran/adaslib/atomic/xxcftr.for*
offline_adas/adas8#1/adaslib/xxcftr.for
Fixed standard form check to include small 'h'
Allowed output strings longer than 19 characters to be handled -
output is put in first 19 spaces (previously blank string was
returned if output string not 18 or 19 characters long)

C.24 In offline series - not in the interactive or ADAS libraries:
Updated *fortran/adas2xx/adaslib/b8getp.for*
offline_adas/adas8#1/adas2xx/adaslib/b8getp.for
Increase number of levels to 3500
Updated *fortran/adas2xx/adaslib/bxcoef.for*
offline_adas/adas8#1/adas2xx/adaslib/bxcoef.for
Increase number of levels to 3500
Increase number of transitions to 380000
Incorporate changes from offline version 1.5:
- Change first dimension of ipla and zpla to 2*ndmet
to accommodate changes in *xxdata_04.for* permitting
Jpj parents.

C.25 Updated *fortran/adaslib/xxdata/xxdata_04.for* and
offline_adas/adas8#1/read_adf/xxdata_04.for

Increased length of cline to allow for 36 orbitals
Fixed bug in detecting highest level present.

C.26 Updated *fortran/adaslib/utility/xxpars.for*
Removed large amounts of commented out code.

C.27 Updated *fortran/adaslib/maths/xxminv.for*
offline_adas/adas8#1/adaslib/xxminv.for
Increased nlmax to 3501

C.28 Updated *offline_adas/adas8#1/adas810/adas810_offline.for*
Increased ndlev from 2800 to 3500, increased ndqdn from 6 to 8
Updated *offline_adas/adas8#1/adas810/Makefile_810* to remove reference to unused subroutine *hawinf.o*

C.29 Updated *offline_adas/adas8#1/adas810/hapecf.for*
Change ndpec 1500 -> 380000 (required for heavy species calculations)
Removed reference to unused subroutine *hawinf* in comments
Extended length of producer string 20->30 to match online version

C.30 Updated *offline_adas/adas8#1/adas2xx/b8scom.for* to match online version:
- Te values for S-line splining may not be the same so set lsetx to TRUE before call to xxsple.
- Set unused values in redscef and redlscom to 0.0.

Updated *offline_adas/adas8#1/adas2xx/b8splt.for* to match online version:
- The check to avoid integrating over zeros in the input can result in no valid points. This causes xxsple an out of bounds error in xxsple. Add a check to avoid the call in this case.

Updated *offline_adas/adas8#1/adas2xx/r8necip.for* to match online version:
- Removed mainframe listing information beyond column 72

Updated *offline_adas/adas8#1/adaslib/bxttyp.for* to match online version:
- Made the routine accept that transition codes of '1', '2' and '3' as well as ' ' correspond to electron impact excitation.

Updated *offline_adas/adas8#1/adaslib/xxfrmt_trm.for* to match online version
- Removed unused integer i4 to keep in line with online version

Updated *offline_adas/adas8#1/adaslib/xxname.for* to match online version:
- Allow for USERIDs > 8 characters (now set to 20).
- Changed test on REALNAME to reflect changes in underlying C code.
- Also moved removal of last character to after 'Who produced this file' is possibly set.
- Add on CHAR(0) to username as C style string terminator rather than '\0'

Updated *offline_adas/adas8#1/adaslib/xxwcmt_15.for* to match online version:
- Removed large numbers of unused variables.

Updated *offline_adas/adas8#1/adaslib/xxwcmt_40.for* to match online version:
- Increased producer string to 30 characters.

Updated *offline_adas/adas8#1/adaslib/xxpars.for* to match online version:
- Copied online version to offline, implementing:
- Removed warning to i4unit aboutlack of parent and 1S being forced
- Added check that ndmet is high enough.
- Added capital letters to comments.
- Removed write to unit 0 inadvertently added with last update.
- Removed large amounts of commented out code.

C.31 Added the option to return an error message if calculation fails in *adas603_get_hdlike*.

C.32 Added top level perl directory.

C.33 Added *atomic.pm* giving perl-implementations of:
xxesym, *xxelem* and *xxeiz0*

C.34 Corrected check of whether wavelength is in range in *hawvrg.for* (online and offline)

C.35 Corrected Te, Ne ordering in call to *d8wzcd.for* in *d8out1.for*

C.36 Fixed bug where the routines:
fortran/adas3xx/adas314/cether.for
fortran/adas3xx/adas314/cewr11.for
fortran/adas3xx/adas314/cewr12.for
fortran/adas3xx/adaslib/cxbms.for
fortran/adas3xx/adaslib/cxchrg.for
fortran/adas3xx/adaslib/cxcrdg.for
fortran/adas3xx/adaslib/cxcrip.for
fortran/adas3xx/adaslib/cxcrps.for
fortran/adas3xx/adaslib/cxdata.for
fortran/adas3xx/adaslib/cxeiqp.for
fortran/adas3xx/adaslib/cxextr.for
fortran/adas3xx/adaslib/cxfrac.for
fortran/adas3xx/adaslib/cxgfil.for
fortran/adas3xx/adaslib/cxghnl.for
fortran/adas3xx/adaslib/cxhyde.for
fortran/adas3xx/adaslib/cxlthe.for
fortran/adas3xx/adaslib/cxmrdg.for
were supposed to be included in the adas3xx static library but weren't.

C.37 Updated *idl/adas4xx/adas416/adas416.pro*:
- calculation and display of child partition fractional abundances has been turned off to stop code hanging. The fortran code was subsequently debugged and this change is reversed.

C.38 Updated *idl/adaslib/read_adf/read_adf00.pro* to allow for neutral ions to be handled with new z_ion, z_nuc keywords.

C.39 Added *write_adf54.pro* and *read_adf54.pro*

C.40 Updated *idl/adas8xx/adaslib/adas8xx_check_cowan_charge_state.pro*:
- Updated comments and removed tabs.
- Added *idl/adas8xx/adaslib/adas8xx_cowan_string_check.pro*

C.41 Updated *idl/adas8xx/adaslib/adas8xx_check_cowan_charge_state.pro*:
Updated *idl/adas8xx/adaslib/adas8xx_cowan_string_check.pro*:
- Added lun_verb keyword for diagnostic output to file.

Updated *idl/adas8xx/adaslib/adas8xx_create_adf15_adf40.pro*:
- first commented version
- added donotrun keyword
- changed inputs to z0_nuc and z_ion from z0,z1
- moved list of files into files structure
- added ca_only keyword

Updated *idl/adas8xx/adaslib/adas8xx_create_ca_adf04.pro*:
- Added call to *adas8xx_cowan_string_check*
- Modified temporary filenames to make unique
- Modified to return 'exit_status'. Also quits rather than crashes if Cowan run has failed
- Put ,/sh on spawning final copying to archive
- Changed z0, z1 inputs to z0_nuc, z_ion
- Moved plasma conditions into plasma structure

- Added exitstatus output
- Added lun_verb for diagnostic output to file
- Added cowan_scale_factors to allow custom adjustments to Slater parameters

Updated *idl/adas8xx/adaslib/adas8xx_create_drivers.pro*:

- First commented version.
- File names are now provided as imputs, not generated within this routine
- Moved plasma conditions into plasma structure
- Added support for z0_nuc and z_ion inputs instead of z0, z1.

Updated *idl/adas8xx/adaslib/adas8xx_create_drivers.pro*:

- First commented version.
- File names are now provided as inputs, not generated within this routine
- Moved plasma conditions into plasma structure
- Added support for z0_nuc and z_ion inputs instead of z0, z1.

Updated *idl/adas8xx/adaslib/adas8xx_create_ls_ic_adf04.pro*:

- First commented version
- added donotrun keyword
- files are now provided in files structure

Updated *idl/adas8xx/adaslib/adas8xx_promotion_rules.pro*:

- Major rewrite: now calls *read_adf54.pro* to obtain promotion rules instead of using hardcoded versions. Also return data in prom_rules structure.
- Now uses z0_nuc and z_ion instead of z0, z1 inputs.

Updated *idl/adas8xx/adaslib/adas8xx_promotions.pro*:

- Added lonarr for config, term and level count vector
- Improved input/output descriptors.
- Correction to logic for rare gas omitted closed-shell detection
- Further correction to logic for Cowan effective z for adf34 driver. Now use zc=z1 for z0<19.
- Correction to preamble text for fill_par
- Replaced variable name z0 by z0_nuc, z1 by z_ion and zc by zc_cow to avoid confusion.
- Introduced rules structure as a keyword parameter
- Changed rules structure to cope with both single element fields and vectors
- Data now returned in promotion_results structure.

C.42 Promotion rules optimisation codes added:

idl/adas8xx/adaslib/adas8xx_opt_check_configuration_match.pro
idl/adas8xx/adaslib/adas8xx_opt_check_parity.pro
idl/adas8xx/adaslib/adas8xx_opt_check_valid_promotion_set.pro
idl/adas8xx/adaslib/adas8xx_opt_control_expand_promotions.pro
idl/adas8xx/adaslib/adas8xx_opt_expand_levels.pro
idl/adas8xx/adaslib/adas8xx_opt_get_total_line_power.pro
idl/adas8xx/adaslib/adas8xx_opt_initialise_rules.pro
idl/adas8xx/adaslib/adas8xx_opt_make_adf11.pro
idl/adas8xx/adaslib/adas8xx_opt_prep_make_adf11.pro
idl/adas8xx/adaslib/adas8xx_opt_promotions_control.pro
idl/adas8xx/adaslib/adas8xx_opt_promotions_run_ca.pro
idl/adas8xx/adaslib/adas8xx_opt_wrapper.pro
idl/adas8xx/adaslib/adas8xx_plasma_defaults.pro

C.43 Added *offline_adas/adas8#4*:

- Scripts for generating inputs for, and then running adas808, adas801 and adas810 offline using the promotions rules structure (ADF54 files).
- Added *offline_adas.adas8#4/run_optimise_promotion_rules.sh*, for calculating the optimal set of configurations to use.
- Added *offline_adas.adas8#4/run_adas808.sh* for using these configurations to run adas8#1

C.44 Updated *idl/adas8xx/adas808/run_adas808.pro*. Substantial rewrite to accommodate heavy species:

- Inclusion of ca_only keyword
- Changed to use adf54 file for promotion rules
- Use long integers for term and level counts
- Added Cowan scale factors keyword
- Added year, verbose, donotrun keywords
- Use theta structure to supply custom plasma conditions

C.45 Updated *offline_adas/adas8#1/adaslib/xxwcmt_15.for*
offline_adas/adas8#1/adas8xx/hapecf.for
fortran/adas8xx/adas810/hapecf.for
fortran/adaslib/xxdata/xxwcmt_15.for
Increased length of ctrans string from 29 to 35.

C.46 Updated *offline_adas/adas8#4/run_adas808.sh*
offline_adas/adas8#4/run_optimise_promotion_rules.sh
Added -idl switch, updated comments.

C.47 A new offline code, adas8#2, has been added to calculate ionisation data with the configuration average distorted wave (CADW) method. This work is in association with Stuart Loch and the Auburn University group.

C.48 Added *idl/adaslib/atomic/tev_alf_s.pro*

C.49 Added *idl/adaslib/write_adf/write_adf11.pro*

C.50 Fixed bug in indexing of last element of xa() array in *adas809/h9ntqd.for*. Only affects running on g77 and other compilers which don't auto-initialise numbers to zero.

C.51 Removed reference to UTC_IS_FLOAT from *cw_adas809_proc*, replaced by call to *num_chk* routine.

C.52 Add *write_adf21.pro* routine which operates on fulldata structure.

C.53 Initialise LRSPEC variable in ADAS406.

C.54 The element in the fulldata structure listing the number of datapoints in adf02 datasets (IEA) did not have the correct type (or information).

C.55 Addition of *xxlvals.pro* and *xxorbs.pro* to canonically specify *l* values (s, p, d...) and orbital specifications (1s, 2s, 2p...).

C.56 Updated *cfg2occ.pro* and *config_orbital_energies.pro* to use *xxlvals* (also *xxorbs* for *cfg2occ*).

C.57 Add *r8waveh.pro*, a companion routine to *r8ah.pro*, to return the wavelength of an n-n' transition.

C.58 *write_adf07.pro* correctly writes the ionisation potential for each block rather than assuming it is the same for all of them.

C.59 Add an IDL routine for bundle-n population calculations. The core low level routine, *cgbnhs.pro*, has been added. A higher level routine, *adas3xx_bn.pro*, which performs the 4 coupled runs to separate and assemble the populations and effective rates is the most likely way *cgbnhs.pro* will be used.

C.60 *run_adas406.pro* crashed if relying on the default behaviour when no initial fractional abundance was set.

C.61 Add an optional multiplet (n-n') A-value output to the hydrogenic routine *r8ah.pro*.

C.62 Clarify meaning of the input variable fraction in *read_adf21.pro*. It is the fraction composition of the target plasma, not the full, half and third energy make-up of the beam.

- Extend the use of /nocheck to turn off most on-screen warning messages. For embedding in other codes the '% READ_ADF21: Assume fraction is constant for all requested parameters' warning will no longer be seen.

C.63 Add utility routine, xxslna.for, to return the length of the largest non-blank string in a string array.

C.64 The length of the configuration string in adf04 datasets is no longer fixed at 18 characters; however full advantage has not been taken of this improvement. Some enhancements to *xxdata_04.for* (in libadaslib) are made without changing the interface.

- The fulldata structure from *read_adf04.pro* (and *xxdata_04.pro*) now returns the full length of the (non-blank part) configuration string.
- write_adf04.pro writes the full user supplied length rather than 18 characters as before. For aesthetic reasons a minimum space of 18 characters will be used in the output for shorter configurations.
- The *filter04.x* command will also not truncate the length of the configuration string.
- Note there is still a limit - a valid adf04 configuration must be larger than 5 and less than 90 characters in length. Hopefully these limits will be sufficient for all purposes.

C.65 Increase the number of levels and transition that ADAS208 can use - up from 150 to 1000 for levels and from 5500 to 1000000.

- adas208 should work with most central adf04 data. However increasing the number of levels will make the code run slower. This compromise balances speed and utility. It may fail on some of the larger heavy species data but adas810 is the preferred way of processing these data.

C.66 Fixed logic in *adas_setup.ksh* for bash users who hadn't manually set an ADASUSER environment variable.

C.67 The eigenvalue/eigenvector routine *xxeign.for* (based on EISPACK routines from netlib) did not normalise the returned eigenvectors. Subsequent use of the eigenvectors in adas406 resulted in numerical instability which is reduced/eliminated if normalized ones are used.

C.68 A new version of *xxdata_09.for*, and an accompanying *xxdata_09.pro*, now returns a more extensive set of data and is compatible with the data added during the DR Project.

C.69 *xxprs3.for* mistakenly used the user, rather than central, ADAS to find its adf00 file. The algorithm for filling the left-out shells in the configuration has been made more robust. An IDL version, xxprs3.pro, is now provided.

C.70 A new utility IDL routine, *occ2cfg.pro*, converts an occupation vector to a standard (or Eissner) configuration. Note that the *cfg2cow.pro* routine should be used to give configurations valid for Cowan/ADAS801 input files.

C.71 adas807 has been re-factored to remove obsolete IDL calls, to use better named supplementary routines (now prefixed with adas807_) and to use central ADAS routines for reading adf04 and adf09 data. This will allow the larger and more complex datasets resulting from the GCR and DR projects to be used. Metastables up to Ar-like are now permitted (Ne-like was the previous limit).

- A command line, *run_adas807.pro*, method is also added.

C.72 The workflow of adas212 has been changed to mirror that of adas211. The output is now a set of augmented R-lines for inclusion in the adf04 file. The option not to supplement any existing R-lines has been removed. The program is still driven by an adf18/a09_a04 driver file but a complete adf04 file is not written. The latest routines to read adf04 and adf09 files are used. This change requires that any existing adas212 defaults file (in the user's adas/defaults/ directory) be removed.

- A command line, *run_adas212.pro*, method is also added.

C.73 For symmetry and completeness a *run_adas211.pro* has also been added.

C.74 Including projection in adas208 population calculations in rare cases leads to situations where minor differences in temperature could result in numerical instability leading to NANs in the recombination photon emissivity coefficients. The problem occurred in mapping the projection data at an early point in the code. The input data to the interpolation routine was not as conditioned as well as it should have been. This has now been fixed. A welcome benefit of this fix is that it has allowed the removal of a long standing ad-hoc filtering routine on the output data. There will be minor numerical differences between the old and fixed versions but the shape of the recombination PECs, for low Te, is now much more believable.

C.75 Do not write warning of 'missing class name in file' to screen when reading adf11 data. The volume of warning quickly overwhelms the user.

C.76 Trim the size of the data in the adf17 dataset produce by adas204 which could occasionally turn the files to binary from plain text.

C.77 Add *preview_natural_partition.pro* to the series 4 IDL library. This routine plots the natural partition for any element and returns the partition in a form suitable for inclusion in the adas416 script files.

C.78 Increased version number to 3.00.

Corrections and updates to data (ADAS v2.13 to ADAS v3.00)

D.1 Add adf15, in the low level metastable unresolved picture, for Li-like Cr^{+21} and Na-like Cu^{+18} :

adf15/transport/transport_llu#cr21ic.dat
adf15/transport/transport_llu#cu18ic.dat

The source adf04 for Cu^{+18} is extracted from Sampson data and is

adf04/copsm#na/copsm#na_sm#cu18.dat

D.2 Remove n=4 data from *adf01/qcx#he0/qcx#he0_old#ne10.dat* because is was zero at all energies.

- Produce an adf12 daaset based on this adf01 as
  adf12/qef93#he/qef93#he_old#ne10.dat

D.3 Minor modifications to existing adf04 datasets:

- Made energy level list more standard in
  adf04/adas#18/helike_adwl01#ar16.dat
  adf04/helike/helike_kvih93he.dat
- Remove excessive white space in
  adf04/belike/belike_nrb05#fe22.dat
- Fix comments to conform to adf04 specification
  adf04/coppm#li/coppm#li_pm#si11j.dat

D.4 Replace *adf11/prc89/prc89_cr.dat* because data from stages 19 to 24 was missing. This may have occurred when transferring the dataset in the distant past.

D.5 Added adf54 directory and files for tungsten and carbon

adf54/promotion_rules_c_adf54.dat
adf54/promotion_rules_w_adf54.dat

D.6 Effective emissivity coefficients for CX emission driven B, C, N and O. For unknown reasons the adf12 data was not produced when the adf01 cross sections were made.

D.7 Update ionisation potentials of magnesium, silicon and iron in *adf00/* using NIST data. Add term resolved dataset for iron (*fe_ls.dat*).

D.8 Minor editing changes to *adf04/copmm#5/* files to make them valid adf04 datasets. The numerical data has not changed but it was not possible to read them with *xxdata_04*.

D.9 Add specific ion data for Mg and Fe from CHIANTI v6. These data are converted to adf04 format, run through the *filter04.x* program to remove levels above the ionisation potential and are e-ordered in increasing energy. The configuration labeling and numerical data is that of the original CHIANTI data. A naming convention is adopted: the data are stored as

adf04/copch#12/chv6_ic#mg<z1>.dat
adf04/copch#26/chv6_ic#fe<z1>.dat

where ch represents CHIANTI,v6 the version used and <z1> is the ion charge of interest.

D.10 The neutral stage was missing for Ni in *adf04/copmm#28/. ls#ni0.dat* has been added.

D.11 Add in location of adf07 data and switch on ionization supplementation from this external file in the adf25 driver for neutral H. Note that this does not affect any derived H data in the central database: the hydrogen GCR data was not generated in the same way as the impurities.

D.12 Remove adf09 dataset: *nrbmb00#he/mb00#he_cu27ls12.dat* since it contained no data.

D.13 The metastable resolved adf00 dataset for Ar (*ar_ls.dat*) had incorrect configurations (one too many electrons) for Ar^{+6} and one of the Ar^{+4} metastables. Note that the energies in the dataset were correct.

D.14 Add a third block to H ionisation rates in *szd93#h_h.dat*. The same Bell et al data is used but the temperature range is larger (5-20keV).

D.15 Some partial cross sections at high nl (10,9 and 9,8) were set to zero in the adf01 file

qcx#h0/qcx#h0_en2_kvi#c6.dat.

These have been corrected to the difference between the shell total and the sum of the other partial cross sections.

D.16 Added B-like Si file *blike_lgy08#si9.dat*, produced by Guiyun Liang.

D.17 Ion impact ionisation from n=2,3,4,5 levels of Hydrogen was in error. The convoluted history and new recommended data are given in an ADAS communication note, available from the website:

http://www.adas.ac.uk/notes/adas_cm09-01.pdf

There are consequences for beam stopping, emission and excited population datasets.

- the new recommended data are stored at the end of the existing adf02 dataset:
  adf02/sia#h/sia#h_j99#h.dat.
- new beam stopping coefficients in
  adf21/bms98#h/bms98#h_h1.dat
  adf21/bms98#h_fast/bms98#h_fast_h1.dat
- new beam emission coefficients
  adf22/bme98#h/bme98#h_h1.dat

- new excited level populations

adf22/bmp98#h/bmp98#h_2_h1.dat
adf22/bmp98#h/bmp98#h_3_h1.dat
adf22/bmp98#h/bmp98#h_4_h1.dat

Although the data were in error this is not a case in which the existing central ADAS data (same file name but with 97) should be replaced. The published cross sections, on which the adf02 data was based, was corrected in an unsatisfactory way in the literature.

D.18 Add adf04 data for ArI and ArII.

The ArII is an update from Don Griffin (J. Phys. B, 48, (2007), p4537) and is added alongside the existing dataset as:

adf04/cllike/cllike_dcg07#ar1.dat

The neutral system is based on Cowan (adas801) supplemented by cross-section data from Dasgupta (Phys Rev A61, 012703 and Phys Rev A65, 039905(E)).

Photon emissivity coefficients for diagnostically useful lines:

adf15/transport/transport_llu#ar0.dat
adf15/transport/transport_llu#ar1.dat

More details are in the ADAS communication note

http://www.adas.ac.uk/notes/adas_cm08-01.pdf

D.19 Ionisation rates for Si by K Dere (Astron. Astrophys., 466 (2007), p771) are added as adf07 data

ionelec_dere07#si.dat.

Archiving data by producer is well established in election excitation (adf04) but this is the first single producer dataset in adf07. More can be added if they are found to be useful.

D.20 Heavy Species Project : Part I - Ionisation.

Ionisation data is generated with the configuration average distorted wave (CADW) code from Auburn University.

To avoid overwhelming ADAS we have generated data for four elements of interest: Ar and W for fusion, Mg and Si for astrophysics.

The driver files are stored in isonuclear directories with 09 as the tag year.

adf32/cadw#12/ca09_mg0.data
ca09_mg1.dat
.
.
adf32/cadw#74/ca09_w73.data

The drivers are automatically produced guided by a set of promotion rules -

adf56/large_arf09.dat.

'arf' stands for Adam Foster who produced these rules.

Individual ionisation rate (adf23) dataset are analogously named:

adf23/cadw#12/ca09_mg0.data
ca09_mg1.dat
.
.
adf23/cadw#74/ca09_w73.data

From the individual adf23 data an adf07 dataset is produced for each element and is stored in adf07:

adf07/cadw/ca09_mg.dat
ca09_si.dat
ca09_ar.dat
ca09_w.dat

D.21 Heavy Species Project : Part II - Excitation and line power.

The *offline_adas/adas8#1* set of codes have been developed to generate baseline data for heavy species. This ADAS release has the first data from this effort. We have restricted it to two elements - argon and tungsten to pilot the process.

There is a significant quantity of data; 2.1Gb mostly made up of specific ion data (adf04). These are stored in
adf04/coparf#18/
adf04/coparf#74/
where 'arf' denotes Adam Foster (http://www.adas.ac.uk/theses/foster_thesis.pdf).

The nominal year 40 tag is used to identify the baseline data.

Four different resolutions are stored for each ion, eg Ar^{+11}
arf40_ca#ar11.dat
arf40_cl#ar11.dat
arf40_ic#ar11.dat
arf40_ls#ar11.dat
which represent,
ca : configuration average
ic : intermediate coupling for configurations in ca
ls : Russell-Saunders coupling for configurations in ca
cl : large set of configuration average

Driver datasets for Cowan/adas801 are archived in
adf34/heavy_species/argon
adf34/heavy_species/tungsten
which use the configurations in the *ca/ls/ic* collection.

Photon emissivities for each resolution are archived as, again for the Ar^{+11} example,
adf15/pec40#ar/pec40#ar_ca#ar11.dat
pec40#ar_cl#ar11.dat
pec40#ar_ls#ar11.dat
pec40#ar_ic#ar11.dat

Feature emissivities for three spectral regions

1.0A - 10.0A	for 128 pixels
10.0A - 100.0A	128
1.0A - 10000.0A	512

are archived in adf40 collection
adf40/fpec40#ar/fpec40#ar_ca#ar11.dat
fpec40#ar_cl#ar11.dat
fpec40#ar_ls#ar11.dat
fpec40#ar_ic#ar11.dat

These have been selected to give the widest applicability but the data tailored to particular instruments would be preferred for serious application. See the adas810 code.

Total radiated power is calculated from the specific ion (adf04) data with the adas810 (online or offline) population code. The difference between P(cl)-P(ca) give the power from the missing configurations and is added to P(ic) to form the total. Note that the ls set are not used but are included for completeness.

Each ionisation stage give rise to a partial adf11/plt dataset and these are archived.

adf11/plt_partial/plt40_partial_ar/
adf11/plt_partial/plt40_partial_w/
Note the parent directory does not have a year number. Future uplift in data quality will be stored under a different year tag.

Again, using the Ar11+ example, we store
plt40_ca#ar11.dat
plt40_cl#ar11.dat
plt40_ls#ar11.dat
plt40_ic#ar11.dat

These are assembled in the final iso-nuclear, and familiar adf11/plt, dataset for
adf11/plt40/plt40_ar.dat
adf11/plt40/plt40_w.dat

HPS
1 Oct. 2009

-

ADAS-EU R(10)PU02

ADAS-EU
ADAS for fusion in Europe

Grant: 224607

Christopher Nicholas, Hugh Summers and Martin O'Mullane

PUBL2: Special features and spectral analysis for fusion plasmas

22 Jun 2010

Workpackages : 26-6-1
Category : DRAFT

PUBL2: Special features and spectral analysis for fusion plasmas

Christopher Nicholas, Hugh Summers and Martin O'Mullane

Department of Physics, University of Strathclyde, Glasgow, UK

Abstract: *In the magnetic confinement fusion domain, spectral analysis is a principal tool for establishing behaviour and performance characteristics of devices. It can assist, inter alia, in determining, with spatial and temporal resolution, key parameters such as electron and plasma ion temperatures and densities, radiated power, impurity transport, impurity concentrations, internal magnetic and electric fields. The spectrometer complement on devices such as JET, AUG and LHD is very large, spanning wavelength ranges from x-ray to infra-red. Similar instrumentation will again be present on the next step in the world fusion program, ITER, currently under construction in France. Spectral analysis, at its most powerful, uses related spectral lines and features for diagnostic inference. In this thesis, such groupings are called special features. Their sensitivity is determined by and enabled by the response of the emitting atoms to their physical environment. In broad concept, this thesis is concerned with special features and their diagnostic exploitation. To achieve this end, the thesis reviews special features and focusses on a number of representative types. It explores and expands the atomic physics link so that the special feature may be realised as a mathematical/computational construct for use in spectral fitting. In implementing this, it works closely with the Atomic Data and Analysis Structure, ADAS, project and its databases. The thesis seeks to empower special feature analysis by implementing generalised computational structures — AFG (ADAS feature generator), FFS (Framework for Feature Synthesis) etc. These allow both a pedagogical insight into the capabilities of each special feature as well as practical execution of optimised spectral fitting and plasma parameter extraction. The methods, based on object oriented programming, are universal including aspects such as self-generating graphical user interfaces and an algbra of parametric feature creation. Demonstration of these methods is on selected JET spectra and 'shot' sequences. It is hoped that the work of this thesis will provide an advanced tool to match the spectral instrumentation capabilities from current fusion devices through to ITER. The implementation will be made available to the fusion community in an ADAS release in due course.*

Contents