AmesPAHdbIDLSuite Guide

- Introduction:

  The *AmesPAHdbIDLSuite*, a suite of object classes utilizing IDL*'s object programming capabilities developed and maintained by Dr. C. Boersma, is provided here that can be used to work with a downloaded database XML-file, offline on your own machine.

  The IDL object classes significantly extend the tools available online. Reading and parsing of the XML-file is automated and operations such as plotting to the screen and/or a file; shifting, convolving, co-adding, fitting astronomical spectra, and applying an emission model have all been fully implemented. Regarding PAH emission models, the suite handles the full temperature cascade, with in addition, the ability to use stellar models for the incident radiation field and optionally use it to do a full convolution of the emergent spectrum. Furthermore, tools for searching and, utilizing IDL object graphics, viewing rotatable 3D molecular structures are also available.

  This document accompanies the version of the suite that is dated *August 27, 2015*.

  NB This guide only applies to versions of the *AmesPAHdbIDLSuite* dated post *February 14, 2015*, as the suite was rewritten to be more consistent and implement some performance enhancements, which are exposed through a new API (application programming interface).

  *IDL is a registered trademark of Exelis Visual Information Solutions

  back to index

- Obtaining the AmesPAHdbIDLSuite:

  The *AmesPAHdbIDLSuite* can be freely downloaded from the download section of the NASA Ames PAH IR Spectroscopic Database website as a gzipped tar-archive.

  back to index

- Installation:

  The *AmesPAHdbIDLSuite* has been tested with IDL versions 7.0 through 8.5. Spectroscopic fitting requires IDL version 8.0 or higher due to the use of !NULL in the implementation of the NNLS-algorithm.

  1. Unpack the *AmesPAHdbIDLSuite*.

     ```
     tar xvaf AmesPAHdbIDLSuite.tgz
     ```

  2. Place the unpacked files in a directory accessible by IDL or update the IDL !PATH-variable accordingly.

     ```
     !PATH = !PATH + ":/path/to/AmesPAHdbIDLSuite/"
     ```

  3. Obtain the XML-version(s) of either/or both the theoretical and experimental database from the download section of the NASA Ames PAH IR Spectroscopic Database website and unpack the gzipped-file(s).

     ```
     gunzip pahdb-full-theoretical-v2.00.xml.gz
     ```

  4. Place the unpacked file somewhere where IDL can access it and/or set either or both of the system and IDL AMESPAHDEFAULTDB environment variables to a default database file you wish to use.

     IDL:

     ```
     DEFSYSV,'!AMESPAHDEFAULTDB',"/path/to/default/xml-file"
     ```

     Bash:

     ```
     export AMESPAHDEFAULTDB='/path/to/default/xml-file'
     ```

     NB On some systems these variables are case-sensitive.

  5. Run 'example.pro' from the 'examples'-directory to verify correct installation.

  Complementary information can be found in the README and INSTALL files accompanying the download.

  back to index

- Getting started:

  ### Main interface

  Interaction with the NASA Ames PAH IR Spectroscopic Database is organized around the 'AmesPAHdbIDLSuite'-object, which is created as shown below.

  ```
  pahdb = OBJ_NEW('AmesPAHdbIDLSuite')
  ```

  Alternatively, the 'AmesPAHdbIDLSuite'-object can be created through the implicit object creation method as shown below.

```
pahdb = AmesPAHdbIDLSuite()
```

In case the system/IDL variable for the default database is not set, or one wishes to load another version or type of database, the 'Filename'-keyword can be specified.

```
pahdb = OBJ_NEW('AmesPAHdbIDLSuite', Filename='/path/to/xml-file')
```

By default the *AmesPAHdbIDLSuite* will cache the parsed database XML-file for faster subsequent access. However, this behavior can be disabled by setting the 'Cache'-keyword to '0'.

```
pahdb = OBJ_NEW('AmesPAHdbIDLSuite', Cache=0)
```

When parsing the database XML-file, the *AmesPAHdbIDLSuite* will try and validate its content against a URL-linked Schema. However, validation can be disabled by setting the 'Check'-keyword to '0'. This can be useful when not having an active internet connection.

```
pahdb = OBJ_NEW('AmesPAHdbIDLSuite', Check=0)
```

It is possible to combine the different keywords.

```
pahdb = OBJ_NEW('AmesPAHdbIDLSuite', Filename='path/to/xml-file', Cache=0, Check=0)
```

When finished with the *AmesPAHdbIDLSuite* the object should be destroyed.

```
OBJ_DESTROY,pahdb
```

back to index

Accessing the molecular PAH data

The molecular PAH data is accessed through their associated unique identifier (UID). However, a search-interface is available that allows retrieval of these UIDs through simplified queries. The syntax is the same as used on the NASA Ames PAH IR Spectroscopic Database website.

```
uids = pahdb->Search("c<=20 neutral n=2 neutral")
```

The molecular PAH data contained in the database XML-files consists of four main parts:

1. Molecular properties, .e.g, molecular weight, zero point energy, total energy, etc.
2. Fundamental vibrational transitions
3. Molecular geometric data
4. Raw laboratory spectra for a subset of molecules in the experimental database

```
pahdb = OBJ_NEW('AmesPAHdbIDLSuite')

uids = pahdb->Search("c<=20 neutral n=2 neutral")

pahs = pahdb->getSpeciesByUID(uids)

transitions = pahdb->getTransitionsByUID(uids)

geometries = pahdb->getGeometryByUID(uids)

laboratory = pahdb->getLaboratoryByUID(uids)

... work ...

OBJ_DESTROY,[laboratory, $
             geometries, $
             transitions, $
             pahs, $
             pahdb]
```

Alternatively, one can access these data components through the 'AmesPAHdbIDLSuite_Species'-object.

```
pahdb = OBJ_NEW('AmesPAHdbIDLSuite')

pahs = pahdb->getSpeciesByUID( $
       pahdb->Search("c<=20 neutral n=2 neutral"))

transitions = pahs->transitions()

geometries = pahs->geometry()

laboratory = pahs->laboratory()

... work ...

OBJ_DESTROY,[laboratory, $
             geometries, $
             transitions, $
             pahs, $
             pahdb]
```

back to index

Working with molecular PAH properties

The 'AmesPAHdbIDLSuite_Species'-object exposes the molecular PAH properties.

```
pahs = pahdb->getSpeciesByUID( $
       pahdb->Search("c<=20 neutral n=2 neutral"))
```

The 'AmesPAHdbIDLSuite_Species'-object's 'Print'-method will print out the associated molecular properties for each PAH species.

```
pahs->Print
```

Optionally, the 'Str'-keyword can be given to the 'Print'-method, which will return the molecular PAH properties for each species as an array of strings.

```
pahs->Print,Str=Str
```

back to index

Working with fundamental vibrational transitions

The 'AmesPAHdbIDLSuite_Transitions'-object exposes the fundamental vibrational transitions.

```
transitions = pahdb->getTransitionsByUID( $
              pahdb->Search("c<=20 neutral n=2 neutral"))
```

The 'AmesPAHdbIDLSuite_Transitions-object's 'Print'-method will print out the associated fundamental vibrational transitions for each PAH species.

```
transitions->Print
```

Optionally, the 'Str'-keyword can be given to the 'Print'-method, which will return the associated fundamental vibrational transitions for each PAH species as a single, concatenated string.

```
transitions->Print,Str=Str
```

The 'AmesPAHdbIDLSuite_Transitions'-object's 'Plot'-method will display the fundamental vibrational transitions in a 'stick'-plot. The transitions of each PAH species will be presented in a different color.

```
transitions->Plot
```

Optionally, the 'Wavelength', 'Stick', 'Oplot', 'Legend', and 'Color'-keywords can be given to the 'Plot'-method to control the abscissa, stick representation, overplotting, legend and color, respectively. Through IDL's keyword inheritance mechanism additional keywords accepted by IDL's 'PLOT'-procedure can be passed.

```
transitions->Plot,/Wavelength,Color=5,XRANGE=[2.5,15],/XSTYLE
```

The 'AmesPAHdbIDLSuite_Transitions'-object's 'Write'-method will write the fundamental vibrational transitions to file. The transitions of each PAH species will be written to a separate text (.txt) file, where each filename will have the PAH species UID embedded. Optionally, a prefix can be given that will be prepended to the filename.

```
transitions->Write,'myPrefix'
```

back to index

Working with molecular geometric data

The 'AmesPAHdbIDLSuite_Geometry'-object exposes the molecular geometric data.

```
geometry = pahdb->getGeometryByUID( $
           pahdb->Search("c<=20 neutral n=2 neutral"))
```

The 'AmesPAHdbIDLSuite_Geometry'-object provides both the 'Plot' and 'Structure'-methods to output the chemical structure of the PAH with provided UID. The first method uses IDL's PLOT-procedures to create the output, while the latter IDL object graphics.

```
; UID 18 = coronene (C24H12)

void = geometry->Plot(18)

img = geometry->Structure(18)
```

The 'Plot'-method will accept the 'Scale', 'Thick', and 'RGB'- keywords to control the scale at which the atoms are to be drawn, the thickness of the bonds, and if colors should be outputted as decomposed RGB, respectively. In addition, the 'NoErase' and 'Position'-keywords control display erasing prior plotting and the plot position in normalized coordinates, respectively. When the 'Resolution'-keyword is set, the plot will be rendered to IDL's Z-buffer device at given resolution and outputted as an image and returned to the caller. When the 'Save'-keyword is also set, a PNG-image will be generated as well, which will have the UID of the PAH molecule under consideration embedded in the filename. Lastly, the 'Angle'-keyword controls rotation of the structure around the *z*-axis.

```
; UID 18 = coronene (C24H12)

img = geometry->Plot(18, Scale=2.0, Thick=5.0, $
                         Resolution=[1200, 1200], /Save)
```

The 'Structure'-method will accept the 'Background' and 'Frame'-keywords to control background color and display of a wireframe. The 'Resolution'-keyword controls the dimension of the 3D-rendered output image. The 'Save'-keyword can be specified to save the generated image as a PNG-image as well, which will have the UID of the PAH molecule under consideration embedded in the filename. If in addition the 'Transparent'-keyword is specified as a color-triplet array, that color will be set to transparent in the generated PNG-file. The 'Axis' and 'Angle'-keywords can control axis and angle of rotation, respectively. With the 'View'-keyword set the generated structure will be displayed using IDL's XOBJVIEW routine. Lastly, when the 'Obj'-keyword is set, the generated IDLgrModel-object is returned to the caller.

```
; UID 18 = coronene (C24H12)

img = geometry->Structure(18, Background=[0, 255, 0], Resolution=[600, 600], $
                              /Save, Transparent=[0, 255, 0], $
                              /View)
```

In addition, the 'AmesPAHdbIDLSuite_Geometry'-object provides the 'Mass', 'Rings', and 'Area'-methods that return the calculated mass based on atomic masses, the number of 3-8 membered rings, and total surface area of the PAH molecules under consideration.

```
masses = geometry->Mass()

rings = geometry->Rings()

areas = geometry->Area()
```

Lastly, the 'Inertia'-method provides the moment of inertia matrices, which are diagonalized with the 'Diagonalize'-method, for the PAH molecules under consideration. The latter method can be particular useful to ensure proper alignment of the structure with the view before calling the 'Plot' or 'Structure'-methods.

```
matrix = geometry->Inertia()

geometry->Diagonalize
```

back to index

Working with raw laboratory spectra

The 'AmesPAHdbIDLSuite_Laboratory'-object exposes available raw laboratory spectra when an experimental database XML-file is loaded.

```
laboratory = pahdb->getLaboratoryByUID( $
             pahdb->Search("c<=20 neutral n=2 neutral"))
```

The 'AmesPAHdbIDLSuite_Laboratory-object's 'Plot'-method will display the raw laboratory spectra. The spectrum of each PAH species will be presented in a different color.

```
laboratory->Plot
```

Optionally, the 'Wavelength', 'Oplot', and 'Color'-keywords can be given to the 'Plot'-method to control the abscissa, overplotting, and color, respectively. Through IDL's keyword inheritance mechanism additional keywords accepted by IDL's 'PLOT'-procedure can be passed.

```
laboratory->Plot,/Wavelength,Color=5,XRANGE=[2.5,15],/XSTYLE
```

The 'AmesPAHdbIDLSuite_Laboratory-object's 'Write'-method will write the raw laboratory spectrum to file. The spectrum of each PAH species will be written to a separate text (.txt) file, where each filename will have the PAH species UID embedded. Optionally, a prefix can be given that will be prepended to the filename.

```
laboratory->Write,'myPrefix'
```

back to index

Working outside the *AmesPAHdbIDLSuite* object classes

The *AmesPAHdbIDLSuite* object classes all provide the 'Get'-method that allows extraction of the object's internal data into an anonymous IDL struct.

```
transitions_s = transitions->Get()
```

This allows, for example, more control over the presentation of the data.

```
PLOT,transitions_s.data.frequency,1D5*transitions_s.data.intensity, $
     XTITLE=transitions_s.units.x.str, $
     YTITLE='integrated intensity [cm!U-2!N mol!U-1!N]', $
     PSYM=10
```

Subsequently this anonymous structure can be manipulated and even be set to the object, which will try altering its internal state to reflect that of the manipulated data.

```
transitions->Set,transitions_s
```

In addition, the 'Set'-method accepts keywords to alter its internal state.

```
transitions->Set,Data=transitions_s.data
```

Lastly, it is also possible to create new *AmesPAHdbIDLSuite*-objects initialized with an appropriate anonymous structure or through keywords.

```
transitions = OBJ_NEW('AmesPAHdbIDLSuite_Transitions', transitions_s)

transitions = OBJ_NEW('AmesPAHdbIDLSuite_Transitions', Type=transitions_s.type, $
                                                       Version=transitions_s.version, $
                                                       PAHdb=pahdb->Pointer(), $
                                                       Data=transitions_s.data, $
                                                       Uids=transitions_s.uids, $
                                                       Model=transitions_s.model, $
                                                       Units=transitions_s.units)
```

back to index

- Creating PAH spectra:

The (theoretical) database XML-files provide fundamental vibrational transitions at 0-Kelvin. To compare these with observations, they need to be transformed into a spectral density, i.e., spectra. When not dealing with absorption at 0-Kelvin, an emission model is required as well.

Emission models

The *AmesPAHdbIDLSuite* offers three PAH emission models. With increasing complexity they are the 'FixedTemperature', 'CalculatedTemperature', and 'Cascade' model. The first simply multiplies a blackbody at fixed given temperature with the integrated cross-section of each vibrational transition. The second first calculates the maximum attained temperature from the provided input and subsequently multiplies a blackbody at that fixed temperature with the integrated cross-section of each vibrational transition. The third averages the total emission over the entire cooling cascade (time).

Emission models are handled by the 'AmesPAHdbIDLSuite_Transitions'-object. The 'FixedTemperature'-model simply takes a temperature, in Kelvin, and, in their simplest form, both the 'CalculatedTemperature' and

'Cascade' models take an energy, in erg.

```
transitions->FixedTemperature,600D

transitions->CalculatedTemperature,6D*1.603D-12; 6 eV

transitions->Cascade,6D*1.603D-12; 6 eV
```

Both the 'CalculatedTemperature' and 'Cascade'-methods accept the 'Approximate', 'Star', 'StellarModel', and 'ISRF'-keywords. With the 'Approximate'-keyword specified, calculations are performed using the PAH emission model from Bakes et al. (2001a, b). When the 'Star'-keyword is set, a stellar blackbody at the provided temperature is used to calculate the average energy absorbed by each PAH utilizing the PAH absorption cross-sections from Draine & Li (2007). In case the 'StellarModel'-keyword is provided as well, the input is considered to be a full-blown, for example, Kurucz stellar atmosphere model. The 'AmesPAHdbIDLSuite_CREATE_KURUCZ_STELLARMODEL_S' helper routine is provided to assist with molding the model data into the proper input format. Lastly, with the 'ISRF'-keyword set, the interstellar radiation field from Mathis et al. (1983) is used to calculate the average energy absorbed by each PAH.

```
transitions->CalculatedTemperature,17D3,/Star

transitions->Cascade,/Approximate,/ISRF

FTAB_EXT,'ckp00_17000.fits',[1,10],angstroms,flem,EXT=1

transitions->Cascade,AmesPAHdbIDLSuite_CREATE_KURUCZ_STELLARMODEL_S(angstroms, flem), $
                     /Star, $
                     /StellarModel
```

The 'Cascade'-method also accepts the 'Convolve'-keyword. When set and combined with either the 'Star', optionally with the 'StellarModel'-keyword, or 'ISRF'-keyword, will instead of calculating the average absorbed photon energy for each PAH, convolve the PAH emission with the entire radiation field.

```
transitions->Cascade,17D3,/Star,/Convolve
```

NB This is computationally expense.

The 'AmesPAHdbIDLSuite_Transitions'-object's 'Shift'-method can be used to redshift the fundamental transitions to simulate anharmonic effects.

```
transitions->Shift,-15D
```

NB Red-shifting the fundamental vibrational transitions should be done before applying one of the three emission models described above.

Line profiles

Line profiles are handled by the 'AmesPAHdbIDLSuite_Transitions'-object and it provides three profiles; Lorentzian, Gaussian and Drude. Convolution with the keyword chosen line profile is achieved through the 'AmesPAHdbIDLSuite_Transitions'-object's 'Convolve'-method, which will return the convolved spectrum in the form of an 'AmesPAHdbIDLSuite_Spectrum'-object.

```
spectrum = transitions->Convolve(/Drude)
```

Optionally, the 'Convolve'-method accepts the 'FWHM', 'Grid', 'NPoints', and 'XRange'-keywords, which control the full-width-at-half-maximum of the selected line profile (in cm^{-1}), convolution onto a specified grid, the number of resolution elements in the generated spectrum, and the frequency range (in cm^{-1}) of the spectrum.

```
spectrum = transitions->Convolve(/Drude, FWHM=20D, Grid=myGrid)
```

The 'AmesPAHdbIDLSuite_Spectrum'-object exposes the convolved spectra and provides the 'Plot', and 'Write'-methods. The 'Plot'-method will display the spectrum of each PAH species in a different color. The 'Write'-method will write all spectra to a single text (.txt) file. Optionally, a prefix can be given that will be prepended to the filename.

```
spectrum->Plot

spectrum->Write,'myPrefix''
```

Optionally, the 'Wavelength', 'Stick', 'Oplot', 'Legend', and 'Color'-keywords can be given to the 'Plot'-method to control abscissa, stick representation, overplotting, legend and color, respectively. Through IDL's keyword inheritance mechanism additional keywords accepted by IDL's 'PLOT'-procedure can be passed.

```
spectrum->Plot,/Wavelength,XRANGE=[2.5,15],/XSTYLE
```

back to index

- Spectroscopic database fitting:

Dealing with astronomical observations

Astronomical observations can be handled by the 'AmesPAHdbIDLSuite_Observation'-object, which is able to read text, *ISO*-SWS, and *Spitzer*-IRS-files. A convenience routine is available to manage units; AmesPAHdbIDLSuite_CREATE_OBSERVATION_UNITS_S.

```
observation = OBJ_NEW('AmesPAHdbIDLSuite_Observation', 'myObservationFile', $
              Units=AmesPAHdbIDLSuite_CREATE_OBSERVATION_UNITS_S())
```

In addition, an 'AmesPAHdbIDLSuite_Observation'-object can be created using keyword-initializers.

```
observation = OBJ_NEW('AmesPAHdbIDLSuite_Observation', X=frequency, $
                                                       Y=intensity, $
                                                       ErrY=ystdev, $
              Units=AmesPAHdbIDLSuite_CREATE_OBSERVATION_UNITS_S())
```

The 'AmesPAHdbIDLSuite_Observation'-object exposes the observation and provides the 'Plot', and 'Write'-methods for output. The 'Plot'-method will display the observation and accepts the 'Oplot', and 'Color'-keywords to control overplotting and color, respectively. Through IDL's keyword inheritance mechanism additional keywords accepted by IDL's 'PLOT'-procedure can be passed.

```
observation->Plot,XRANGE=[2.5,15],/XSTYLE
```

The 'Write'-method will write the observation to a single text (.txt) file. Optionally, a prefix can be given that will be prepended to the filename.

```
observation->Write,'myPrefix'
```

NB Text-files can have up to five columns, organized as follows. Column 1: abscissa, Column 2: ordinate, Column 3: continuum, Column 4: uncertainty in ordinate, and Column 5: uncertainty in abscissa.

The 'AmesPAHdbIDLSuite_Observation'-object's 'Rebin', 'AbscissaUnitsTo', and 'SetGridRange'-methods can rebin the observation onto a specified grid or, with the 'Uniform'-Keyword set, onto a uniform created grid with specified sampling; convert the units associated with the abscissa; and change the grid range, respectively.

```
observation->Rebin,myGrid

observation->Rebin,5D,/Uniform

observation->AbsciccaUnitsTo

observation->SetGridRange,500,2000
```

Database fitting

Spectroscopic database fitting is handled by the 'AmesPAHdbIDLSuite_Spectrum'-object and it either accepts an 'AmesPAHdbIDLSuite_Observation'-object, or a simple array of ordinates with an optional array of ordinate uncertainties. Whether ordinate uncertainties are provided or not, the 'AmesPAHdbIDLSuite_Spectrum'-object's 'Fit'-method will perform a non-negative least-chi-square or non-negative least-square fit and return an 'AmesPAHdbIDLSuite_Fitted_Spectrum'-object.

```
fit = spectrum->Fit(intensity, uncertainty)
```

The 'AmesPAHdbIDLSuite_Fitted_Spectrum'-object exposes the fit and provides the 'Plot', and 'Write'-methods for output. The 'Plot'-method accepts the 'Residual', 'Size', 'Charge', and 'Composition'-keywords, which selectively display the residual of the fit, or either the size, charge and compositional breakdown. Without these keywords the fit itself is displayed.

```
fit->Plot,/Charge
```

Optionally, the 'Wavelength', 'Stick', 'Oplot', 'Legend', and 'Color'-keywords can be given to the 'Plot'-method to control the abscissa, stick representation, overplotting, legend and color, respectively. Through IDL's keyword inheritance mechanism additional keywords accepted by IDL's 'PLOT'-procedure can be passed.

```
fit->Plot,/Size,/Wavelength,XTITLE=[2.5,15],/XSTYLE
```

The 'AmesPAHdbIDLSuite_Fitted_Spectrum'-object's 'Write'-method will write the fit to a single text (.txt) file. Optionally, a prefix can be given that will be prepended to the filename.

```
fit->Write,'myPrefix'
```

The 'AmesPAHdbIDLSuite_Fitted_Spectrum'-object's 'GetClasses', and 'GetBreakdown'-methods return the fit broken down by charge, size, and composition, where the first provides the spectrum for each component and the latter its relative contribution.

```
classes = fit->getClasses()

breakdown = fit->getBreakdown()
```

Optionally the 'Small' keyword can be set, which controls the small cutoff size in number of carbon atoms.

```
classes = fit->getClasses(Small=20L)
```

The 'GetBreakdown'-method also accepts the 'Flux'-keyword, which controls whether the relative breakdown should be reported based on fitted weight or integrated flux.

```
breakdown = fit->getBreakdown(/Flux)
```

back to index

- Browser:

  Utilizing IDL's widget programming capabilities, a simple graphical user interface (GUI) is available to explore the contents of a database XML-file. The GUI provides an overview of the embedded PAH species and the fundamental vibrational transitions - both tabulated and graphically, and 3D-rotatable chemical structure of the selected species. A database XML-file can be loaded through the 'File→Open' menu-option.

  ```
  browser = OBJ_NEW('AmesPAHdbIDLSuite_Browser')
  ```

  NB The 'AmesPAHdbIDLSuite_Browser'-object does not require to be explicitly destroyed, as it calls OBJ_DESTROY upon window close.

  back to index

- A complete example:

  A complete example of fitting an observation is shown below. It is an adaption of the 'fit_a_spectrum.pro'-procedure found in the examples-directory of the *AmesPAHdbIDLSuite*, and is followed by a detailed, line-by-line, explanation of the code.

```
1  observation = OBJ_NEW('AmesPAHdbIDLSuite_Observation', 'myFile', $
2                        Units=AmesPAHdbIDLSuite_CREATE_OBSERVATION_UNITS_S())
3
4  observation->AbscissaUnitsTo,1
5
6  observation->Rebin,5D,/Uniform
7
8  pahdb = OBJ_NEW('AmesPAHdbIDLSuite')
9
01 transitions = pahdb->getTransitionsByUID( $
11             pahdb->Search("mg=0 o=0 fe=0 si=0 chx=0 ch2=0 c>20"))
21
31 transitions->FixedTemperature,600D
41
51 transitions->Shift,-15D
61
71 spectrum = transitions->Convolve(/Lorentzian, $
81                                  Grid=observation->getGrid(), $
91                                  FWHM=20D)
02
12 fit = spectrum->Fit(observation)
22
32 OBJ_DESTROY,spectrum
42
52 fit->Plot,/Wavelength
62
72 fit->Plot,/Wavelength,/Residual
82
92 fit->Plot,/Wavelength,/Charge
03
13 transitions->Intersect, $
23        fit->getUIDs()
33
43 spectrum = transitions->Convolve(/Lorentzian, $
53                               FWHM=20D, $
63                               XRange=1D4/[20D, 3D])
73
83 coadded = spectrum->CoAdd(Weights=fit->getWeights())
93
04 coadded->Plot
14
24 OBJ_DESTROY,[coadded, spectrum, fit, transitions, pahdb, observation]
```

lines 1-2: An observation is read from 'myFile' and the 'AmesPAHdbIDLSuite_CREATE_OBSERVATION_UNITS_S'-helper function is called to associate units.

line 4: Observation abscissa units are converted to wavenumber.

line 6: The observation is rebinned onto a uniform grid spaced 5 wavenumbers.

line 8: The default NASA Ames PAH IR Spectroscopic Database XML-file is loaded

line 10: The fundamental vibrational transitions from a subset of PAHs are retrieved.

line 13: A FixedTemperature emission model at 600 Kelvin is applied.

line 15: The fundamental vibrational transitions are redshifted 15 wavenumbers.

line 17: The fundamental vibrational transitions are convolved with Lorentzian profiles having a full-width-at-half-maximum of 20 cm^{-1} onto the observational grid.

line 21: The observation is fitted with the PAH emission spectra.

line 23: Cleanup of 'spectrum'.

line 25-29: Display several aspects of the fit.

line 31: The transitions are intersected with the PAH species in the fit.

line 34: The fundamental vibrational transitions are again convolved with Lorentzian profiles having a full-width-at-half-maximum of 20 cm^{-1}, but now onto a generated grid from 3-10 micron.

line 38: The individual PAH spectra are added using weights retrieved from the fit.

line 40: The coadded spectrum is displayed, revealing the entire 3-20 micron, predicted, PAH spectrum.

line 42: Cleanup.

back to index

- FAQ:

- **Why doesn't the code in this guide work with my installation?**

  You are probably using a version of the AmesPAHdbIDLSuite dated prior February 14, 2015. Please update to a newer version.

- **Does the *AmesPAHdbIDLSuite* work with GDL?**

  *The AmesPAHdbIDLSuite is **not** compatible with GDL, the open source alternative to IDL, as it currently lacks the framework to parse XML-files.*

- **How does the *AmesPAHdbIDLSuite* deal with color output?**

  The AmesPAHdbIDLSuite 'Plot'-methods create and load a color table containing 15 colors. When outputting to screen, the device will be set into discrete graphics mode. Upon completion, both the color table and the display mode will be restored to their initial settings.

- **Where have the C-routines gone?**

  The C-code has been removed from the AmesPAHdbIDLSuite for three reasons. First, The NNLS implementation in the suite now makes use of IDL's LA_LEAST_SQUARES-routine, which is implemented using LAPACK and therefore very fast. Second, it appears that IDL's READS-routine has seen some significant speed improvements in recent IDL releases such that the CONVERT_ASCII C-implementation has become obsolete. Third and last, making the C-routines compile and reliable across multiple platforms has proven to be difficult.

back to index