Spectral Feature Synthesis

Chris Nicholas
Andrew Meigs
Hugh Summers
Allan Whiteford

ADAS 2008

Overview

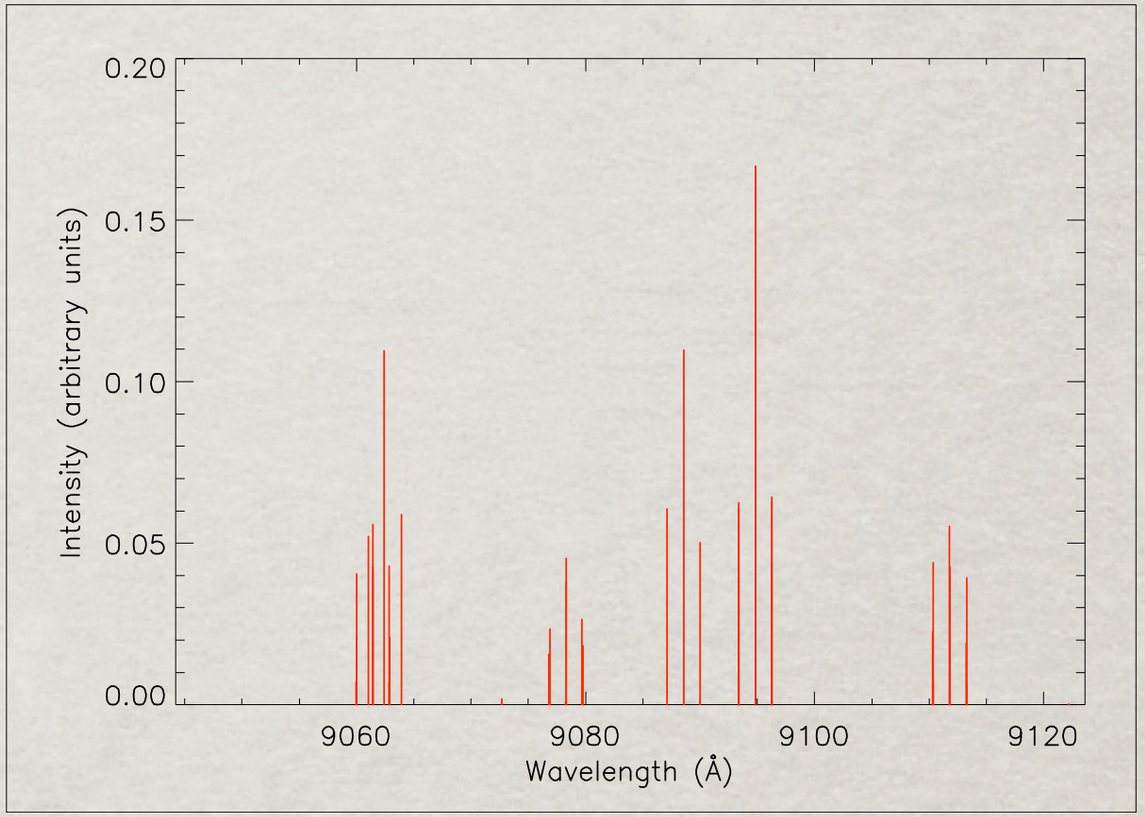
- ADAS Feature Generation (AFG)
  - Easy access to ADAS special feature models.
  - Provides common access point to the ADAS special feature codes.
  - Consistent interface when utilising each of the models.
  - Graphical exploration tool allows auto generated example code.
- Framework for Feature Synthesis (FFS)
  - Managed data structure for modelling complex spectra, using a modular approach.
  - Provides a simple language for defining combination of features.
  - Handles parameter attributes for numerical fitting.
  - Model definition language allows for coupling of parameters.

ADAS Feature Generation (AFG) API

- Currently, the supported models include:
  - Heavy species envelope emission
  - Motional Stark multiplet
  - Zeeman / Paschen Back
- Awaiting completion / inclusion:
  - He-like soft x-ray resonance and satellite lines
  - Balmer series / series limit
- Modular system allows plug-in of future ADAS special features.

Example AFG Plot

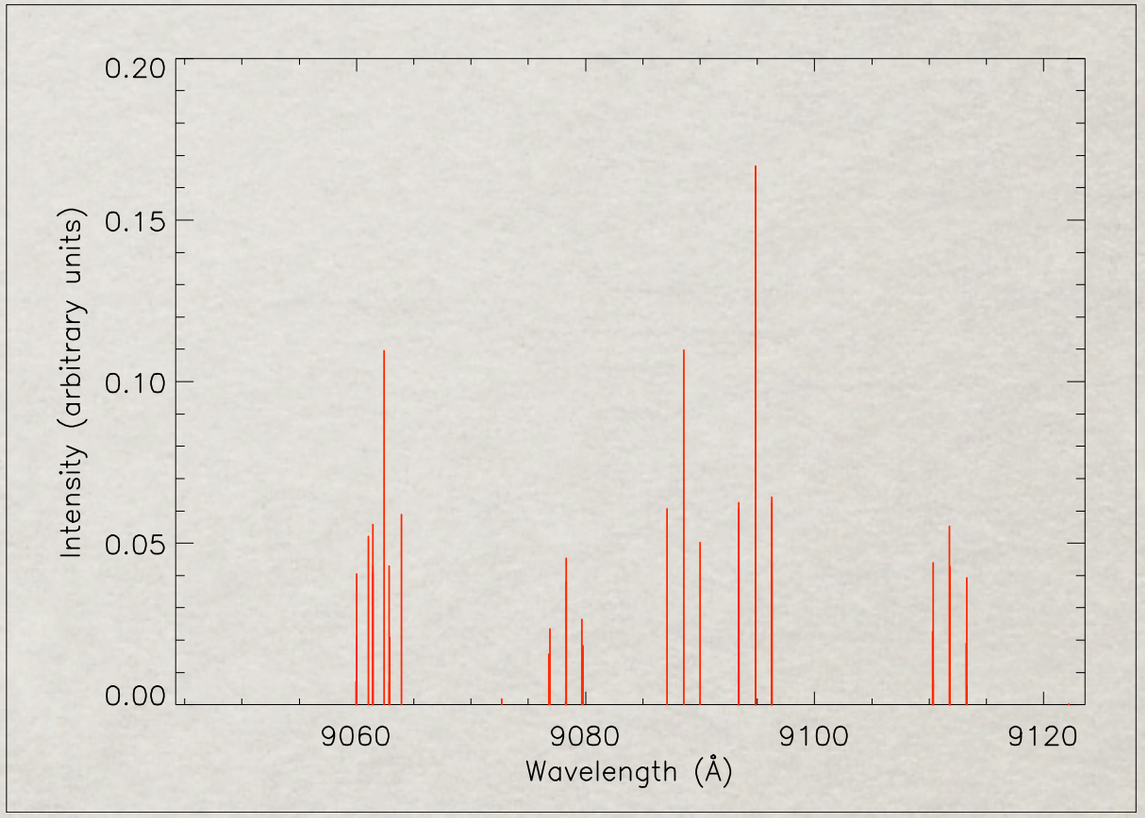
```
o = obj_new('afg_zeeman')
pars = o->getPars()

pars.pol=1
pars.obsangle=90.0
pars.bvalue=2.5
pars.findex=15

isok = o->setPars(PARS=pars)

res = o->getcalc()

plot,res.wv,res.intensity,/nodata

for i=0, n_elements(res.wv)-1 do $
        oplot, [res.wv[i], res.wv[i]], [0.0, res.intensity[i]]
```



Note: above code can be auto-generated from the GUI.

Example AFG Plot

```
pars=afg('zeeman',/parameters)

pars.pol=1
pars.obsangle=90.0
pars.bvalue=2.5
pars.findex=15

res=afg('zeeman',calculate=pars)

plot,res.wv,res.intensity,/nodata

for i=0,n_elements(res.wv)-1 do $
    oplot,[res.wv[i],res.wv[i]],[0,res.intensity[i]]
```

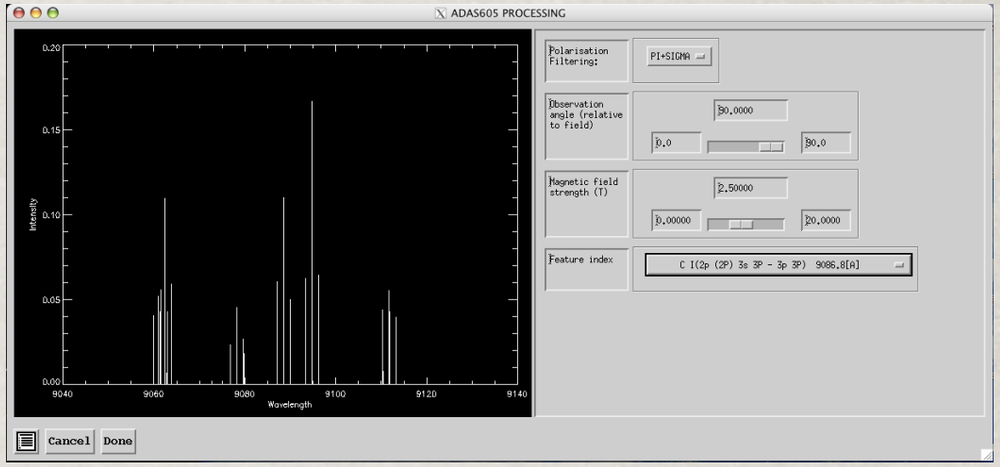


Example AFG Query

```
o = obj_new('zeeman')
desc = o->getDesc()
```

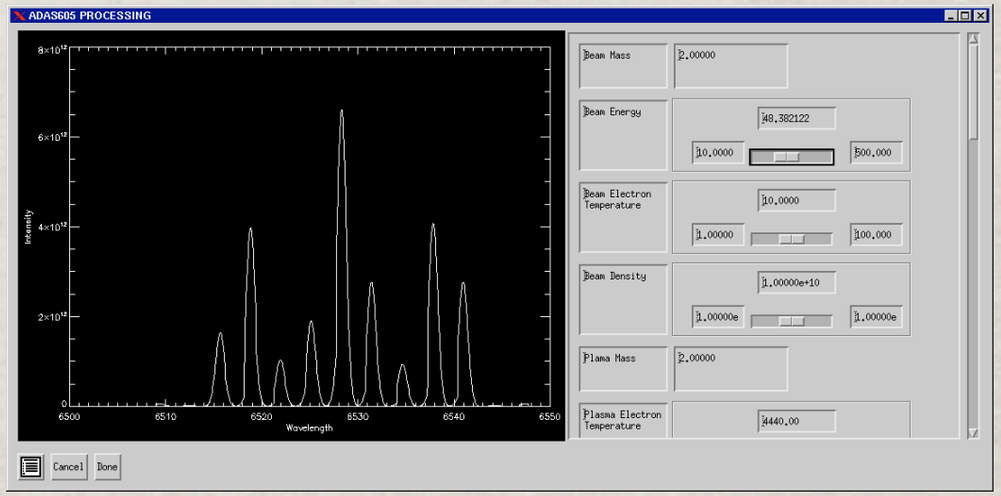
```
help, desc, /str
```

```
NAME            STRING    'Zeeman Feature'
PARAMETERS      STRUCT    -> <Anonymous> Array[1]
```




```
help, desc.parameters, /str
```

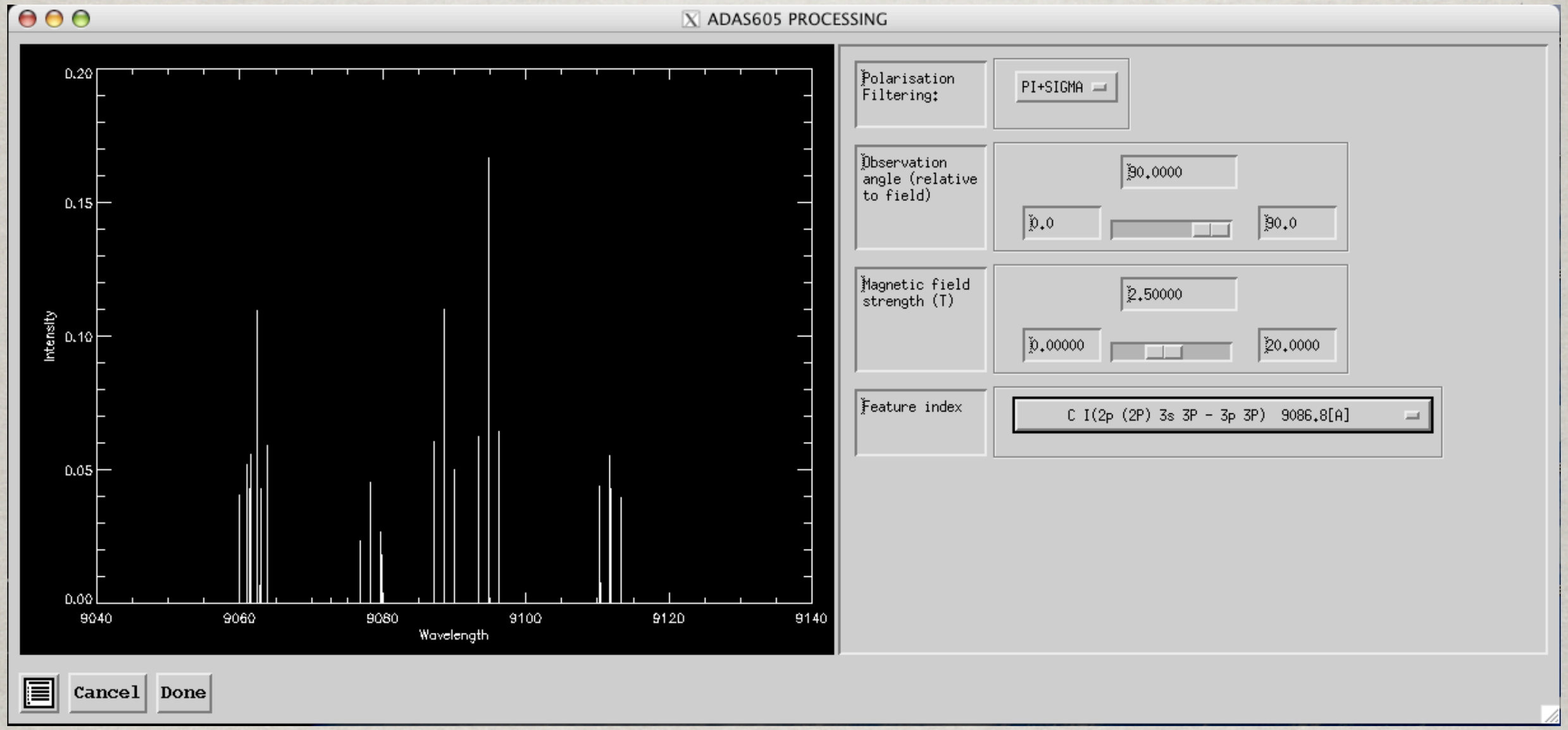
```
POL             STRUCT    -> <Anonymous> Array[1]
OBSANGLE        STRUCT    -> <Anonymous> Array[1]
BVALUE          STRUCT    -> <Anonymous> Array[1]
FINDEX          STRUCT    -> <Anonymous> Array[1]
```




```
help, desc.parameters.obsangle, /str
```

```
DESC            STRING    'Observation angle (relative to field)'
TYPE            STRING    'float'
UNITS           STRING    ‘degrees'
MIN             STRING    '0.0'
MAX             STRING    '90.0'
DISPTYPE        STRING    'continuous'
```

ADAS605 Processing screen



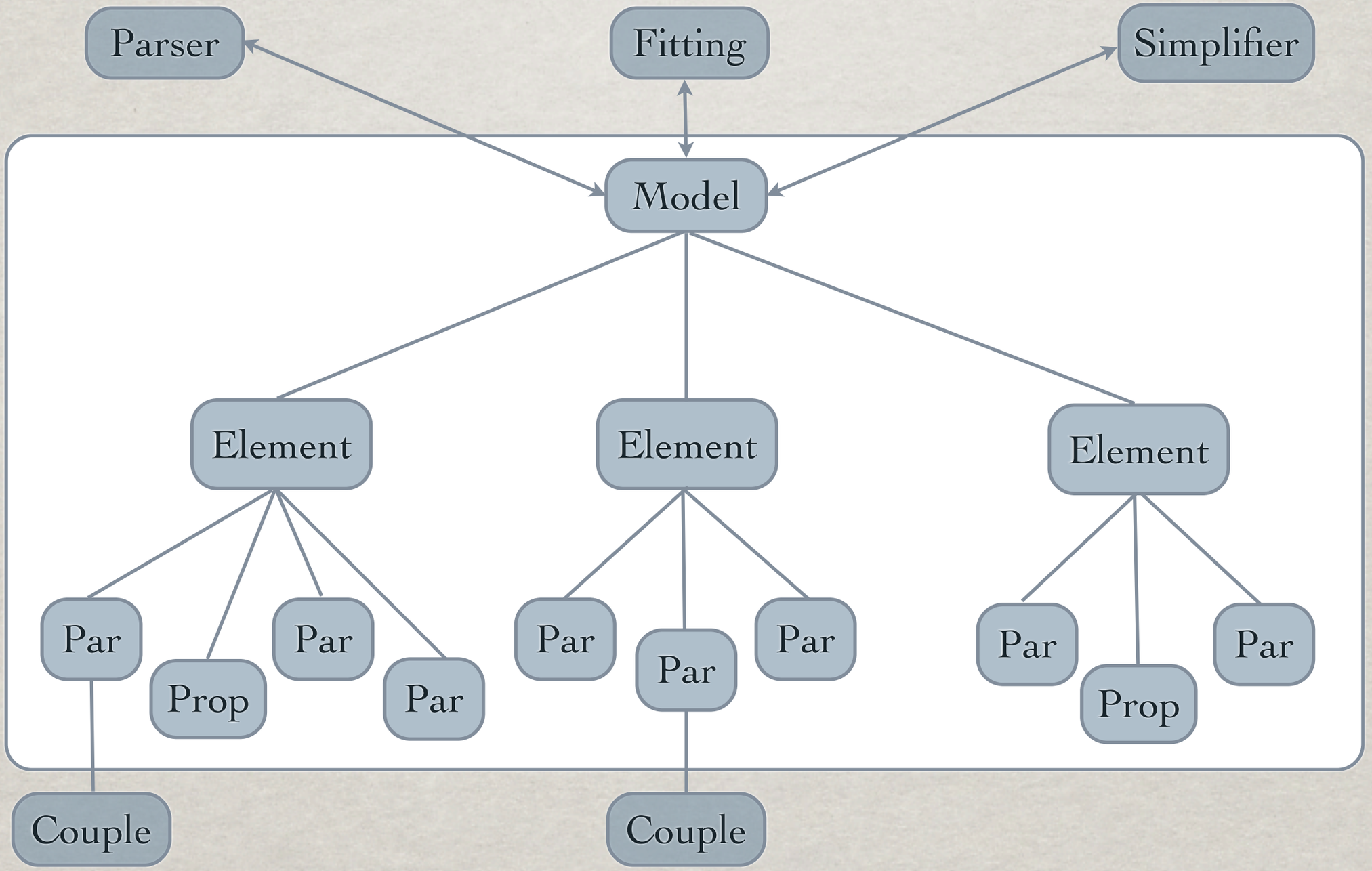
Spectral Feature Synthesis

Christopher Nicholas

Overview

- ADAS Feature Generation (AFG)
  - Easy access to ADAS special feature models.
  - Provides common access point to the ADAS special feature codes.
  - Consistent interface when utilising each of the models.
  - Graphical exploration tool allows auto generated example code.
- Framework for Feature Synthesis (FFS)
  - Managed data structure for modelling complex spectra, using a modular approach.
  - Provides a simple language for defining combination of features.
  - Handles parameter attributes for numerical fitting.
  - Model definition language allows for coupling of parameters.

DIAGRAM OF FFS MODULES



FFS Model Definition Syntax

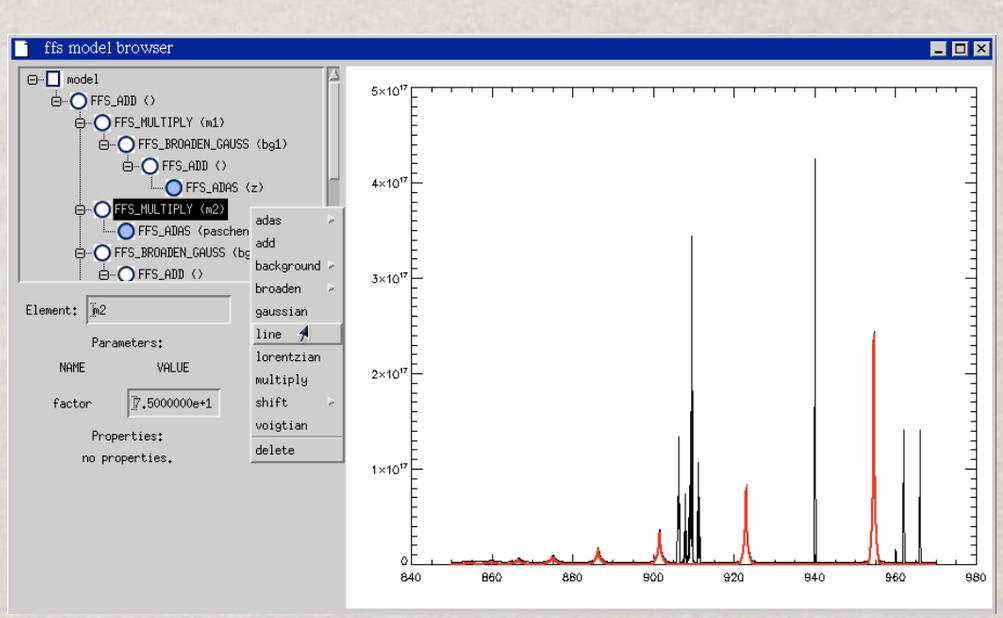
- An example model definition:

```
(model modelname
  (+
    (*
      (broaden_gauss
        (+
          (adas-zeeman z)
        )
      bg1)
    m1)
    (* (adas-paschen_archived paschen) m2)
    (broaden_gauss
      (+
        (line l1)
        (line l2)
        (line l3)
        (line l4)
        (line l5)
        (line l6)
      )
    bg2)
    (background-linear back)
  )
)
```

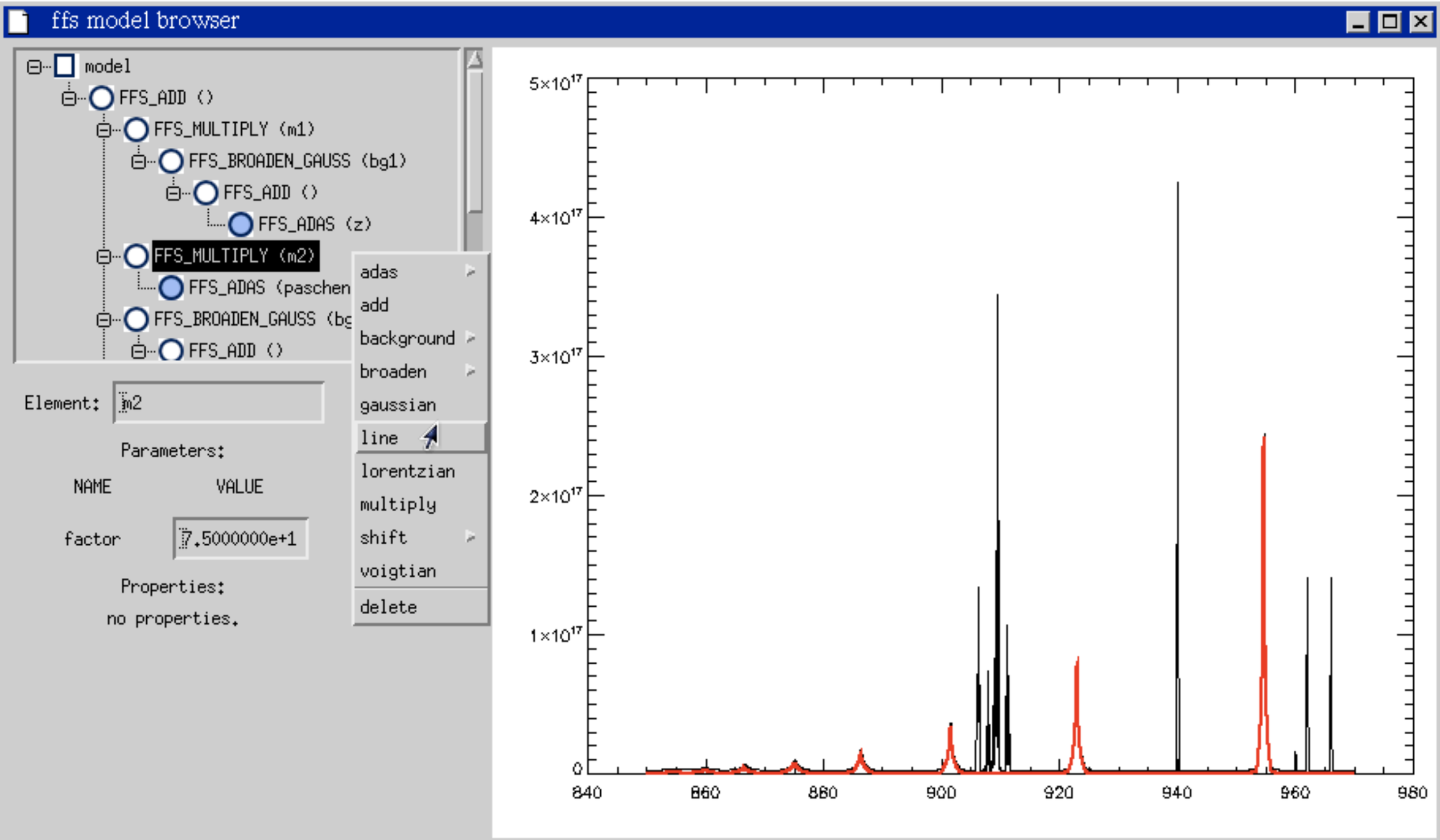
Format is specified as follows:

(elementclass[-optinput] operands name)

where the operands are further element expressions (optional) and 'optinput' allows for an additional string to be passed to that particular feature code.



FFS Model Browser (test version)



Model Definition: Parameter Values

- Setting parameter values:

(setval bg1.fwhm 0.1)

(setval z.obsangle 90.000000)

(setval z.bvalue 4.0000000)

(setval z.pol 1)

(setval z.findex 15)

(setval m1.factor 9.0e16)

(setval paschen.te 1.7e4)

(setval paschen.dens 2.0e13)

(setval paschen.filename /home/nicholas/afg/examples/paschen.dat)

(setval m2.factor 7.5e19)

(setval bg2.fwhm 0.2)

(setval l1.pos 940.0)

(setval l1.intensity 9.0e16)

....

....

Format is specified as follows:
(setval *elementname.parname value*)

Model Definition: Parameter Coupling and Limits

- Setting parameter coupling:
  - Line 'l2' is to be coupled such that it's intensity is twice that of line 'l1':

    (couple l2.intensity (* l1.intensity 2.0))

    Format is specified as follows:
    (couple *elementname.partocouplename expression*)
    The expressions are of the form:
    (*operator* operands*)*
    the operators are arithmetic (+, -, *, /, ^) and the operands are numeric values or further model parameters.

- Setting parameter limits:
  - (setmin l1.intensity 0.0)
  - (setlimits l1.intensity 0.0 60.0)

    Format is specified as follows:
    (setmin *elementname.parname value*)
    or
    (setlimits *elementname.parname min max)*

- Note that special features coming from AFG have limits imposed automatically using the description structure supplied by AFG.

Model 'Simplification'

- Takes input model definition and provides a more optimal representation of the model.
- Can provide more efficient calculation of model spectra.
- Can enable use of analytical expressions for model partial derivatives with respect to the parameters.
- Opaque to the user - 'simplified' model used for evaluation and partial derivatives, but linked back to original user specified model.
  - This means for an arbitrarily complex model the code implicitly/explicitly does the necessary maths to determine what the analytic partial derivative is.

ELEMENT COMBINATIONS

Consider a Gaussian broadening function:

$$
\begin{aligned}
&B_g\left\{f\{\dots\}, w_g\right\}(x) \\
&= \left[G\{w_g\} * f\{\dots\}\right](x) \\
&= \int_{-\infty}^{+\infty} G\left\{w_g\right\}(x - x') f\{\dots\}(x') dx'
\end{aligned}
$$

$$G\{w_g\}(x) = \frac{C}{\sqrt{\pi} w_g} \exp\left(-\frac{C^2 x^2}{w_g^2}\right)$$

$$I_g\{\lambda_0, \phi, w_g\}(\lambda) = \frac{C\phi}{\sqrt{\pi} w_g} \exp\left(-\frac{C^2 (\lambda - \lambda_0)^2}{w_g^2}\right)$$

$$
\begin{aligned}
\frac{\partial}{\partial \lambda_0} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda) &= \frac{2C^2 (\lambda - \lambda_0)}{w_g^2} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda) \\
\frac{\partial}{\partial w_g} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda) &= \frac{1}{w_g}\left(2C^2 \frac{(\lambda - \lambda_0)^2}{w_g^2} - 1\right) I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda) \\
\frac{\partial}{\partial \phi} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda) &= \frac{1}{\phi} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda)
\end{aligned}
$$

$$C = 2\sqrt{\ln 2}$$

Apply the gaussian broadener to a gaussian line and we get another gaussian:

$$
\begin{aligned}
I_{\mathrm{g(new)}}\{\lambda_{0(\mathrm{new})}, \phi_{(\mathrm{new})}, w_{g(\mathrm{new})}\}(\lambda) &= B_g\left\{I_g\left\{\lambda_0, \phi, w_{g1}\right\}, w_{g2}\right\}(\lambda) \\
&= \frac{C}{\sqrt{\pi}\sqrt{w_{g_1}^2 + w_{g_2}^2}} \exp\left(\frac{-C^2 (\lambda - \lambda_0)^2}{w_{g_1}^2 + w_{g_2}^2}\right)
\end{aligned}
$$

ELEMENT COMBINATIONS

Consider a Gaussian broadening function:

$$
\begin{aligned}
&B_g\left\{f\{\dots\}, w_g\right\}(x) \\
&= \left[G\{w_g\} * f\{\dots\}\right](x) \\
&= \int_{-\infty}^{+\infty} G\left\{w_g\right\}(x - x') f\{\dots\}(x') dx'
\end{aligned}
$$

$$G\{w_g\}(x) = \frac{C}{\sqrt{\pi} w_g} \exp\left(-\frac{C^2 x^2}{w_g^2}\right)$$

$$I_g\{\lambda_0, \phi, w_g\}(\lambda) = \frac{C\phi}{\sqrt{\pi} w_g} \exp\left(-\frac{C^2(\lambda - \lambda_0)^2}{w_g^2}\right)$$

$$\frac{\partial}{\partial \lambda_0} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda) = \frac{2C^2(\lambda - \lambda_0)}{w_g^2} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda)$$

$$\frac{\partial}{\partial w_g} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda) = \frac{1}{w_g}\left(2C^2\frac{(\lambda - \lambda_0)^2}{w_g^2} - 1\right) I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda)$$

$$\frac{\partial}{\partial \phi} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda) = \frac{1}{\phi} I_g\left\{\lambda_0, \phi, w_g\right\}(\lambda)$$

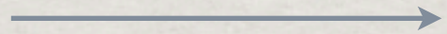
$$C = 2\sqrt{\ln 2}$$

Apply the gaussian broadener to a gaussian line and we get another gaussian:

$$
\begin{aligned}
&I_{\mathrm{g(new)}}\{\lambda_{0(\mathrm{new})}, \phi_{(\mathrm{new})}, w_{g(\mathrm{new})}\}(\lambda) = B_g\left\{I_g\left\{\lambda_0, \phi, w_{g1}\right\}, w_{g2}\right\}(\lambda) \\
&= \frac{C}{\sqrt{\pi}\sqrt{w_{g_1}^2 + w_{g_2}^2}} \exp\left(\frac{-C^2(\lambda - \lambda_0)^2}{w_{g_1}^2 + w_{g_2}^2}\right)
\end{aligned}
$$

FFS 'Simplification' Example

```
(model original
   (+
      (broaden_gauss
         (+
            (gaussian g1)
            (lorentzian l1)
            (broaden_lorentz
               (+
                  (line theline)
                  (adas-zeeman az)
               )
            blorentz)
         )
      bgauss)
   )
)
```



```
(model simplified
   (+
      (gaussian new_gauss)
      (voigtian new_voigt)
      (voigtian new_voigt)
      (broaden_voigt
         (+
            (adas-zeeman new_az)
         )
       new_bvoigt)
   )
)
```

Internally, expressions such as:

(couple new_gauss (^ (+ (^ (* bgauss.fwhm 1.0) 2) (^ g1.fwhm 2)) 0.5)))

are formed to couple the parameters back to the original parameter set.

Summary

- AFG provides easy, common interface to ADAS special feature models.
- FFS will provide unified approach to modelling arbitrarily complex spectra.
- ADAS Feature Generation (AFG) program will be in the next release of ADAS.
- GUI to AFG will also be available as ADAS 605.
- FFS still has some time to reach completion, expected sometime next year.