User Macros in PAN

Rob Dimeo

This document describes the new functionality in PAN in which the user can type in customized macros in IDL syntax for fitting their data.

<u>The user macro interface</u>

Once data is loaded into PAN then you can select the `user macro function` selection from the `select function` drop-down menu as shown in figure 1. Note that you can define and use as many fit macros as you wish for your overall fit function.
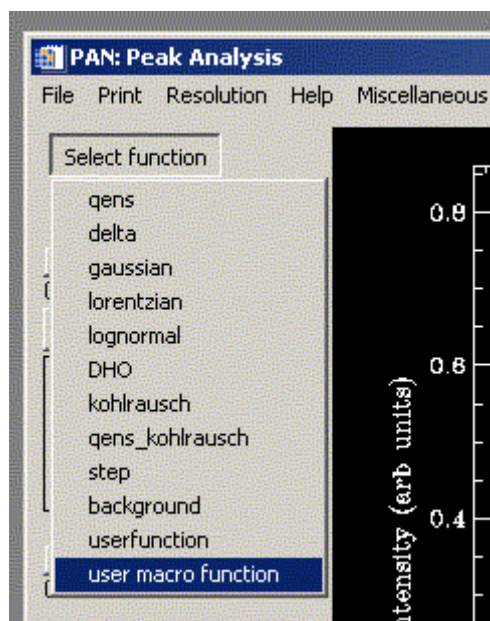

Fig. 1 User macro function selection from the select-function menu.

The macro entry interface is then launched as shown in figure 2. A default function is loaded in automatically in order to illustrate the required syntax for the user macro. In the default case the macro defines a Gaussian plus a flat background. You can save your custom macros or you can restore previously-saved macros using the buttons at the bottom of the macro entry interface. Note that if a resolution function is loaded into PAN then the default behavior is to fit the result of the convolution of your fit function with the resolution function to the data.
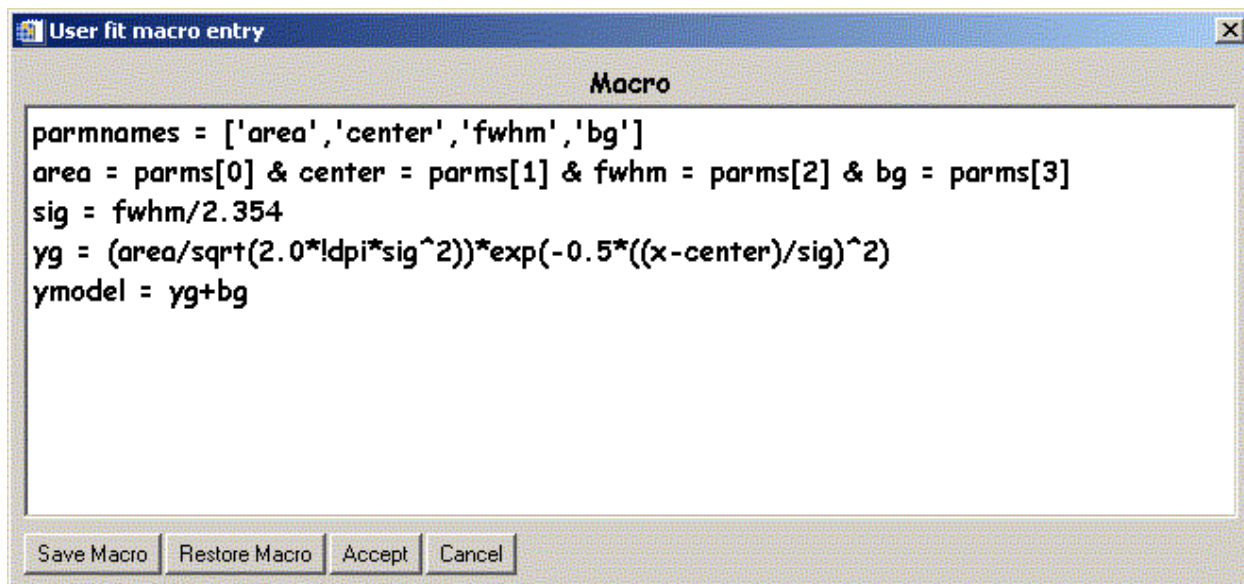
Fig. **2** The user macro function entry interface.

Macro Requirements

The requirements for the macros are as follows:

1. The macro <u>must begin</u> with a string array defining the parameter names. You can be as creative as you like with the names but the array must adhere to IDL syntax.   The parameter names for the Gaussian are defined as follows:

   ```
   parmnames = ['area','center','fwhm','bg']
   ```

2. The parameter values are stored in a vector called `parms` and must be referenced as such. The parameter values must be stored in the same order as the parameter names in the parmnames array, or,

   ```
   parms = [area,center,fwhm,bg]
   ```

   In IDL syntax this is the same thing as:

   ```
   area = parms[0]
   center = parms[1]
   fwhm = parms[2]
   bg = parms[3]
   ```

3. The independent variable is always `x`.

4. The output for the function which does not include δ-functions must be a variable named `ymodel`. This is the last line in the code shown in fig. 2. See point 6 below for how to incorporate δ-functions into your fit function.
5. The code must be written like an IDL macro. This means that you have access to all of the high level functionality of IDL but you are limited to a single executable line within a control loop. Examples of macros are shown in the next section.
6. When using δ-functions in your fit function the resolution function must be loaded in. Furthermore the δ-functions are specified by an array of areas and centers using the variable name `delta-params`. For example, to represent the following equation,

$$y_{\text{deltas}} = a_1 \delta(x - c_1) + a_2 \delta(x - c_2),$$

you would enter the following

```
delta_params = [[a₁,c₁],[a₂,c₂]].
```

Macro Examples

The syntax for the macros is standard IDL syntax. Therefore you may want to consult the IDL help for more information on the functionality available.

Sloping line (simple):

$$y_{\text{model}} = b + m \cdot x$$

```
parmnames = ['offset','slope']
offset = parms[0] & slope = parms[1]
ymodel = offset+slope*x
```

Note that you could have also written this more compactly as:

```
parmnames = ['offset','slope']
ymodel = parms[0]+parms[1]*x
```

Quasielastic lineshape (with Gaussian resolution):

$$y_{\text{model}} = \text{eisf} \bullet \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{1}{2}\left(\frac{x - x_o}{\sigma}\right)^2\right) + (1 - \text{eisf}) \bullet \frac{\Gamma}{\pi} \frac{1}{(x - x_o)^2 + \Gamma^2}$$

```
parmnames = ['amp','eisf','gcenter','gfwhm','lfwhm']
amp = parms[0] & eisf = parms[1] & center = parms[2] & gfwhm = parms[3]
```

```
lfwhm = parms[4]
sig = gfwhm/2.354
gamma = lfwhm/2.0
yg = (1.0/sqrt(2.0*!dpi*sig^2))*exp(-0.5*((x-center)/sig)^2)
yl = (gamma/!pi)/((x-center)^2+gamma^2)
ymodel = amp*(eisf*yg+(1.0-eisf)*yl)
```

Diffusion within a plane (somewhat more complicated):

$$y_{model} = \frac{1}{\pi} \frac{1}{x^2 + B^2} \bullet K$$

$$K = 1 + \sum_{n=1}^{\infty} \frac{1}{2n+1} \cos[(n+1)\varphi], \quad \tan \varphi = \frac{x}{B}$$

```
parmnames = ['amp','width']
amp = parms[0] & b = parms[1]
nx = n_elements(x)
nn = 500
nindex = 1.0+findgen(nn)
phi = atan(x/b)
ux = 1+bytarr(nx)
t1 = (1.0/(2.0*nindex+1.0))#ux
t2 = (cos((nindex+1.0)#phi))
k = 1.0+total(t1*t2,1)
ymodel = amp*((k/!pi)/(x^2+b^2))
```

QENS lineshape (with resolution function):

$$y_{model} = area\left[eisf \bullet \delta(x) + (1 - eisf) \bullet \frac{\Gamma}{\pi} \frac{1}{x^2 + \Gamma^2} \right] \otimes R(x)$$

```
parmnames = ['area','eisf','fwhm','bg']
area = parms[0] & eisf = parms[1] & fwhm = parms[2]
bg = parms[3]
gamma = 0.5*fwhm
ylor = area*(1.0-eisf)*(gamma/!dpi)/(x^2+gamma^2)+bg
delta_params = [[area*eisf,0.0]]
ymodel = ylor
```