
Subject: gaussfit

Posted by [Frank Molster](#) on Tue, 25 Jun 1996 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi everyone,

I have encountered a small problem, I hope somebody can help me.
I am not an experienced user but to me it seems that the keyword NTERMS
in gaussfit doesn't work. I've tried several possibilities but nothing
works. If I'm correct the next command should work, shouldn't it?

```
IDL> yfit = gaussfit(x,y,A,nterms=4)
% Compiled module: GAUSSFIT.
% Procedure/function called with too many parameters: GAUSSFIT.
% Execution halted at: $MAIN$
```

It works correctly without the 'nterms=4' option.
Is this a (known) bug or do I something wrong?
Can anybody help me?

Thanks a lot,

Frank Molster

--

	Frank Molster	% Internet: frankm@astro.uva.nl
	Astronomical Institute	% Phone: (+31 20\020) 5257470
	University of Amsterdam	% FAX: (+31 20\020) 5257484
	Kruislaan 403	%
	1098 SJ Amsterdam	%

Subject: Re: gaussfit

Posted by [rivers](#) on Wed, 03 May 2000 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

In article <8en0oi\$cq6\$1@nnrp1.deja.com>, loeh@my-deja.com writes:
> does anyone has routines for fitting multiple gaussian
> to a data set as it appears in energy spectra?
> (2 or more gaussians which overlap - convolution of gaussians)

I have a pretty comprehensive routine for fitting Gaussians to energy spectra.
It uses MPFITFUN by Craig B. Markwardt for the actual fitting.
My routines is very flexible in terms of what parameters are to be fitted, and
which are to be kept fixed. I am appending some of the documentation from the

procedure.

You can get it at

ftp://cars.uchicago.edu/pub/epics/idl_mca.tar

Mark Rivers

Here is some of the documentation for the routine.

```
; In general a Gaussian peak has 3 adjustable parameters: position
; (or energy), sigma (or FWHM), and amplitude (or area). For many
; applications, however, not all of these parameters should be
; adjustable during the fit. For example, in XRF analysis the energy of
; the peaks is known, and should not be optimized. However, the overall
; energy calibration coefficients for the entire spectrum, which relate
; channel number to energy, might well be optimized during the fit.
; Similarly, the FWHM of XRF peaks are not independent, but rather
; typically follow a predictable detector response function:
;   FWHM = A + B*sqrt(energy)
; Finally, even the amplitude of an XRF peak might not be a free
; parameter, since, for example one might want to constrain the K-beta
; peak to be a fixed fraction of the K-alpha. Such constraints allow
; one to fit overlapping K-alpha/K-beta peaks with much better accuracy.
;
; This procedure is designed to be very flexible in terms of which
; parameters are fixed and which ones are optimized. The constraints are
; communicated via the Fit and Peaks structures.
;
; The energy of each channel is assumed to obey the relation:
;   energy = energy_offset + (channel * energy_slope)
;
; These parameters control the fit for peaks whose energy is fixed,
; rather than being a fit parameter.
; If Fit.energy_flag is 1 then these energy calibration coefficients
; will be optimized during the fitting process. If it is 0 then these
; energy calibration coefficients are assumed to be correct and are not
; optimized. Not optimizing the energy calibration coefficients can
; both speed up the fitting process and lead to more stable results when
; fitting small peaks. This function does a sanity check and will not
; optimize these energy calibration coefficients unless at least 2 peaks
; have their .energy_flag field set to 0, so that they use these global
; calibration coefficients.
;
; The FWHM of the peaks is assumed to obey the relation:
;   fwhm = fwhm_offset + (fwhm_slope * sqrt(energy))
; These parameters control the fit for peaks whose FWHM is neither fixed
; nor a fit parameter.
```

If `Fit.fwhm_flag` is 1 then these coefficients will be optimized during the fitting process. If it is 0 then the specified coefficients are assumed to be correct and are not optimized. Not optimizing the FWHM coefficients can both speed up the fitting process and lead to more stable results when fitting very small peaks. This function does a sanity check and will not optimize these FWHM calibration coefficients unless at least 2 peaks have their `.fwhm_flag` field set to 0, so that they use these global calibration coefficients.

This function also optimizes the following parameters:

- The amplitudes of all peaks whose `.ampl_factor` field is 0
- The energies of all peaks whose `.energy_flag` field is 1
- The FWHM of all peaks whose `.fwhm_flag` field is 1

The parameter which is the minimized during the fitting process is χ^2 , defined as:

$$\chi^2 = \sum_i \frac{(y_{\text{obs}}[i] - y_{\text{pred}}[i])^2}{\text{sigma}[i]}$$

where `y_obs[i]` is the observed counts in channel `i`, `y_pred` is the predicted counts in channel `i`, and `sigma[i]` is the standard deviation of `y_obs[i]`.

This function assumes that:

$$\text{sigma}[i] = y_{\text{obs}}[i] ** \chi_{\text{exponent}}$$

e.g. that the standard deviation in each channel is equal to the counts in the channel to some power. For photon counting spectra where Poisson statistics apply `chi_exponent=0.5`, and this is the default. Setting `chi_exponent=0.` will set all of the `sigma[i]` values to 1., and the fit would then be minimizing the sum of the squares of the residuals. This should tend to result in a better fit for the large peaks in a spectrum and a poorer fit for the smaller peaks. Setting `chi_exponent=1.0` will result in a minimization of the sum of the squares of the relative error in each channel. This should tend to weight the fit more strongly toward the small peaks.

If `.ampl_factor` for a peak is 0., then the amplitude of the peak is a fit parameter. If the `amplitude_factor` is non-zero then the amplitude of this peak is not a fit parameter, but rather is constrained to be equal to the amplitude of the last previous peak in the array which had an amplitude factor of zero, times the `amplitude_factor`. This can be used, for instance, fit K-alpha and K-beta x-ray lines when the alpha/beta ratio is known, and one wants to add this known constraint to the fitting process.

```
; For example:
; peaks = replicate({mca_peak}, 3)
; ; Fe Ka is the "reference" peak
; peaks[0].initial_energy=6.40 & peaks[0].ampl_factor=0.0
; ; Si-Ka escape peak is 3% of Fe Ka at 4.66 keV
; peaks[1].initial_energy=4.66 & peaks[1].ampl_factor=0.03
; ; Fe-Kb is 23% of Fe Ka
; peaks[2].initial_energy=7.06 & peaks[2].ampl_factor=0.23
; In this example the amplitude of the Fe-Ka peak will be fitted, but the
; amplitudes of the escape peak and the Fe-Kb peak are constrained to
; be fixed fractions of the Fe-Ka peak. The reference peak is always the
; closest preceding peak in the array for which ampl_factor is 0.
;
```
