Subject: Gaussfit problem
Posted by wlandsman on Thu, 03 Jul 2014 17:50:54 GMT
View Forum Message <> Reply to Message

Below I give example code showing a problem with gaussfit.pro when NTERMS=4 (fitting a Gaussian plus a constant). The code is exactly the same as in the Gaussfit documentation, except the input data has a sigma width of 5 instead of 2. The result of the fit can be seen at
http://idlastro.gsfc.nasa.gov/ftp/old/gaussfit.png

What happens is that gaussfit.pro first subtracts the mean from the input data, so that the mean of the data is zero. It then tries to determine if the data is an emission or an absorption feature, by comparing the absolute value of the maximum with the absolute value of the data minimum. But if the Gaussian is an emission feature but not strongly peaked, the minimum occurring at the end of the data range can have a larger absolute value than the real peak. So the gaussfit initial guess is that the data is an absorption feature with a centroid at the very end of the range, and this yields a nonsense result.

My gaussfit.pro call was actually from gauss2dfit.pro.which requires the user to state whether it is an absorption or emission feature. (The /NEGATIVE keyword is used for an absorption, otherwise a peak is fit). But the information about whether the user is fitting a peak or absorption feature is not passed onto gaussfit.pro

I'd suggest that gaussfit include a keyword stating whether it is an emission or absorption feature, and gauss2dfit be modified to call gaussfit.pro with this keyword. If the keyword is not supplied then gaussfit reverts to its current behavior.

And yes I know I should be using Craig Markwardt's mpfit2dpeak
http://cow.physics.wisc.edu/~craigm/idl/fitting.html
but I am debugging someone else's code. --Wayne

```
pro test
;Modify the test code for gaussfit.pro to show problem with data that is not
;strongly peaked.
  n = 101

x = (FINDGEN(n)-(n/2))/4

; Define the coefficients.

a = [4.0, 1.0, 5.0, 1.0]

print, 'Expect: ', a
```

```
z = (x - a[1])/a[2] ; Gaussian variable

seed = 123321 ; Pick a starting seed value
y = 0.4*RANDOMN(seed, n)
y = y + a[3] + a[0]*exp(-z^2/2)
nterms = 4
 yfit = GAUSSFIT(x, y, coeff, NTERMS=nterms)

print, 'Result: ', coeff[0:nterms-1]

; Plot the original data and the fitted curve:

p1 = PLOT(x, y, TITLE='nterms='+STRTRIM(nterms,2), $

LAYOUT=[4,1,nterms-2], CURRENT=(nterms gt 3), $

DIMENSIONS=[800,200], MARGIN=[0.1,0.2,0.1,0.2])

p2 = PLOT(x, yfit, THICK=2, /OVERPLOT)
return
end
```