Subject: Re: Voigt funcion in mpfit
Posted by Markus Schmassmann on Wed, 25 Oct 2017 13:37:18 GMT
View Forum Message <> Reply to Message

On 10/25/2017 01:15 PM, abc wrote:
> HI, I have set of data of an absorption spectra and I want to fit the voigt profile using mpfit. The function I am using is http://www.heliodocs.com/php/xdoc_print.php?file=$SSW/yohkoh /ucon/idl/metcalf/voigt.pro
> But I am getting error as
> GDL> res=mpfit('voigt',p0,functargs=fa)
> % VOIGT: function VOIGT takes 2 params: 'Result = VOIGT(A,U)'
> % Execution halted at: MPFIT_CALL_FUNC_EXTRA 1396 /home/abc/Documents/gdl/pro/pro/mpfit.pro
> % MPFIT_CALL 1439 /home/abc/Documents/gdl/pro/pro/mpfit.pro
> % MPFIT 3185 /home/abc/Documents/gdl/pro/pro/mpfit.pro
> % $MAIN$
> Kindly help
>
I recommend looking at the documentation in MPFITFUN.pro & VOIGT.pro & https://en.wikipedia.org/wiki/Voigt_profile#Relation_to_Voig t_profile

I'll probably end up with something like this, but check the derivatives
before using AUTODERIVATIVE=0 or PARINFO[i].MPSIDE = 3.
You can do this manually or using
; PARINFO[i].MPSIDE = 3 ; Enable explicit derivatives
; PARINFO[i].MPDERIV_DEBUG = 1 ; Enable derivative debugging mode
; PARINFO[i].MPDERIV_RELTOL = ?? ; Relative tolerance for comparison
; PARINFO[i].MPDERIV_ABSTOL = ?? ; Absolute tolerance for comparison
for more details see the documentation in MPFIT.pro

I hope this works, beyond compiling I have not debuged it.

Good luck, Markus

```
function voigt_wraper, x, p, dp
    voigt, p[1]/(sqrt(2)*p[0]),x/(sqrt(2)*p[0]),h,f
    if n_params() gt 2 then begin
       requested=dp
       dp = make_array(n_elements(x), n_elements(p), value=x[0]*0)
       if requested[0] ne 0 then $
          dp[*,0]=-p[2]*(p[1]*f-x*h)/(sqrt(2*!dpi)*p[0]^3)
       if requested[1] ne 0 then $
          dp[*,1]=-p[2]*( (2*p[1]-h)/(sqrt(2*!dpi)*p[0]^2) $
                -(p[1]^2*h+2*x*f*p[0]-x^2*h)/p[0]^3 )
       if requested[2] then ne 0 dp[*,2]=-1/(sqrt(2*!dpi)*p[0])*h
       if requested[3] then ne 0 dp[*,3]=1d
    endif
```

```
    return, p[3] -p[2]/(sqrt(2*!dpi)*p[0])*h
end


p0=[sigma0,gamma0,norm,cont]
p_out = MPFITFUN('voigt_wraper', x, y, err, p0)
```