
Subject: Re: gaussfit

Posted by [Walid](#) on Mon, 01 Jul 1996 07:00:00 GMT

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

Frank Molster wrote:

>
> 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 % |

> |
Hi,

I tried to respond earlier to this gaussfit message but something must be wrong with my news server because it never showed up! Anyway, Robert Moss is correct in stating that the new version of gaussfit was never shipped with the latest version (4.01) of IDL. I contacted RSI about this and they gave me the new version, which I am attaching. Also, if you are running IDL on a PC under windows, there is another bug which may rear its ugly head while running gaussfit, which calls curvefit, which calls NR_MACHAR(), a numerical recipes routine that returns machine specific information. This routine hangs the system. The (user-supplied) workaround is to compile a new routine, say, macharr(), and rename nr_machar() in the routine curvefit to macharr(). Macharr()

was also given to me by RSI, and I am also attaching it. Hope this helps. Let me add that RSI informed me that this was still an "open" bug which is being given top priority.

Also, I should state that making the above changes worked MOST of the time, but when I tried using gauss2dfit, which calls gaussfit, and used the /TILT option, IDL hangs for my particular data set. If anyone knows how to work around this, please let me know. I will try it on a SUN station and see if it hangs there as well. If it does, I'll let RSI know, and hopefully they can do something about it.

Walid

atia@wam.umd.edu

; \$Id: gaussfit.pro,v 1.3 1995/09/21 17:03:17 dave Exp \$

```
PRO GAUSS_FUNCT,X,A,F,PDER
; NAME:
; GAUSS_FUNCT
;
; PURPOSE:
; EVALUATE THE SUM OF A GAUSSIAN AND A 2ND ORDER POLYNOMIAL
; AND OPTIONALLY RETURN THE VALUE OF IT'S PARTIAL DERIVATIVES.
; NORMALLY, THIS FUNCTION IS USED BY CURVEFIT TO FIT THE
; SUM OF A LINE AND A VARYING BACKGROUND TO ACTUAL DATA.
;
; CATEGORY:
; E2 - CURVE AND SURFACE FITTING.
; CALLING SEQUENCE:
; FUNCT,X,A,F,PDER
; INPUTS:
; X = VALUES OF INDEPENDENT VARIABLE.
; A = PARAMETERS OF EQUATION DESCRIBED BELOW.
; OUTPUTS:
; F = VALUE OF FUNCTION AT EACH X(I).
;
; OPTIONAL OUTPUT PARAMETERS:
; PDER = (N_ELEMENTS(X),6) ARRAY CONTAINING THE
; PARTIAL DERIVATIVES. P(I,J) = DERIVATIVE
; AT ITH POINT W/RESPECT TO JTH PARAMETER.
; COMMON BLOCKS:
; NONE.
; SIDE EFFECTS:
; NONE.
; RESTRICTIONS:
; NONE.
; PROCEDURE:
```

```

; F = A(0)*EXP(-Z^2/2) + A(3) + A(4)*X + A(5)*X^2
; Z = (X-A(1))/A(2)
; Elements beyond A(2) are optional.
; MODIFICATION HISTORY:
; WRITTEN, DMS, RSI, SEPT, 1982.
; Modified, DMS, Oct 1990. Avoids divide by 0 if A(2) is 0.
; Added to Gauss_fit, when the variable function name to
; Curve_fit was implemented. DMS, Nov, 1990.
;
n = n_elements(a)
ON_ERROR,2 ;Return to caller if an error occurs
if a(2) ne 0.0 then begin
    Z = (X-A(1))/A(2) ;GET Z
    EZ = EXP(-Z^2/2.) ;GAUSSIAN PART
endif else begin
    z = 100.
    ez = 0.0
endelse

case n of
3: F = A(0)*EZ
4: F = A(0)*EZ + A(3)
5: F = A(0)*EZ + A(3) + A(4)*X
6: F = A(0)*EZ + A(3) + A(4)*X + A(5)*X^2 ;FUNCTIONS.
ENDCASE

IF N_PARAMS(0) LE 3 THEN RETURN ;NEED PARTIAL?
;
PDER = FLTARR(N_ELEMENTS(X),n) ;YES, MAKE ARRAY.
PDER(0,0) = EZ ;COMPUTE PARTIALS
if a(2) ne 0. then PDER(0,1) = A(0) * EZ * Z/A(2)
PDER(0,2) = PDER(*,1) * Z
if n gt 3 then PDER(*,3) = 1.
if n gt 4 then PDER(0,4) = X
if n gt 5 then PDER(0,5) = X^2
RETURN
END

```

Function Gaussfit, x, y, a, NTERMS=nt

```

;+
; NAME:
; GAUSSFIT
;
; PURPOSE:
; Fit the equation y=f(x) where:
;
;

```

```

; F(x) = A0*EXP(-z^2/2) + A3 + A4*x + A5*x^2
; and
; z=(x-A1)/A2
;
; A0 = height of exp, A1 = center of exp, A2 = sigma (the width).
; A3 = constant term, A4 = linear term, A5 = quadratic term.
; Terms A3, A4, and A5 are optional.
; The parameters A0, A1, A2, A3 are estimated and then CURVEFIT is
; called.
;
; CATEGORY:
; ?? - fitting
;
; CALLING SEQUENCE:
; Result = GAUSSFIT(X, Y [, A])
;
; INPUTS:
; X: The independent variable. X must be a vector.
; Y: The dependent variable. Y must have the same number of points
; as X.
; KEYWORD INPUTS:
; NTERMS = Set NTERMS to 3 to compute the fit: F(x) = A0*EXP(-z^2/2).
; Set it to 4 to fit: F(x) = A0*EXP(-z^2/2) + A3
; Set it to 5 to fit: F(x) = A0*EXP(-z^2/2) + A3 + A4*x
;
; OUTPUTS:
; The fitted function is returned.
;
; OPTIONAL OUTPUT PARAMETERS:
; A: The coefficients of the fit. A is a three to six
; element vector as described under PURPOSE.
;
; COMMON BLOCKS:
; None.
;
; SIDE EFFECTS:
; None.
;
; RESTRICTIONS:
; The peak or minimum of the Gaussian must be the largest
; or smallest point in the Y vector.
;
; PROCEDURE:
; If the (MAX-AVG) of Y is larger than (AVG-MIN) then it is assumed
; that the line is an emission line, otherwise it is assumed there
; is an absorption line. The estimated center is the MAX or MIN
; element. The height is (MAX-AVG) or (AVG-MIN) respectively.
; The width is found by searching out from the extrema until

```

```

; a point is found less than the 1/e value.
;
; MODIFICATION HISTORY:
; DMS, RSI, Dec, 1983.
; DMS, RSI, Jun, 1995, Added NTERMS keyword. Result is now float if
;   Y is not double.
;-
;
on_error,2          ;Return to caller if an error occurs
if n_elements(nt) eq 0 then nt = 6
if nt lt 3 or nt gt 6 then $
    message,'NTERMS must have values from 3 to 6.'
n = n_elements(y) ;# of points.
s = size(y)
c = poly_fit(x,y,1,yf) ;Fit a straight line
yd = y - yf
if s(s(0)+1) ne 5 then begin ;If Y is not double, use float

    yd = float(yd) & c = float(c) & endif

ymax=max(yd) & xmax=x(!c) & imax=!c ;x,y and subscript of extrema
ymin=min(yd) & xmin=x(!c) & imin=!c

if abs(ymax) gt abs(ymin) then i0=imax else i0=imin ;emiss or absorp?
i0 = i0 > 1 < (n-2) ;never take edges
dy=yd(i0) ;diff between extreme and mean
del = dy/exp(1.) ;1/e value
i=0
while ((i0+i+1) lt n) and $ ;guess at 1/2 width.
    ((i0-i) gt 0) and $
    (abs(yd(i0+i)) gt abs(del)) and $
    (abs(yd(i0-i)) gt abs(del)) do i=i+1
a = [yd(i0), x(i0), abs(x(i0)-x(i0+i))]
if nt gt 3 then a = [a, c(0)] ;estimates
if nt gt 4 then a = [a, c(1)]
if nt gt 5 then a = [a, 0.]
!c=0 ;reset cursor for plotting
return,curvefit(x,y,replicate(1.,n),a,sigmaa, $
    function_name = "GAUSS_FUNCT") ;call curvefit
end

function MACHARR, DOUBLE=double

; Values were obtained by executing Numerical Recipes
; example program XMACHAR (calls NR routine MACHAR).
if keyword_set(double) then $
    return, {beta:2L, it:64L, irnd:3L, ngrd:1L, $

```

```
machep:-52L, negep:-53L, iexp:11L, $
minexp:-1022L, maxexp:1024L, $
eps: 0.222044604925031D-015, $
epsneg:0.111022302462516D-015, $
xmin:0.222507385850720D-307, $
xmax:0.179769313486232D+39 } $
else $
  return, {beta:2L, it:64L, irnd:3L, ngrd:1L, $
    machep:-23L, negep:-24L, iexp:8L, $
    minexp:-126L, maxexp:128L, $
    eps: 0.119209E-06, epsneg:0.596046E-07, $
    xmin:0.117549E-37,xmax:0.340282E+39 }
end
```

File Attachments

- 1) [gaussfit.pro](#), downloaded 140 times
 - 2) [Macharr.pro](#), downloaded 102 times
-