
Subject: Problem with gaussfit in IDL/v5.5
Posted by [thompson](#) on Fri, 16 Nov 2001 17:14:12 GMT
[View Forum Message](#) <> [Reply to Message](#)

We found a problem with the procedure gaussfit which is supplied with IDL/v5.5. It appears that somebody at RSI tried to improve the algorithm for generating the initial guess, but didn't quite complete the job. I've informed RSI, and decided to also inform the newsgroup. The simple fix is to change the line

```
if nt gt 5 then a = [a, 0.]
```

to instead read

```
if nt gt 5 then a = [a, c[2]]
```

The complete routine with the fix is below.

William Thompson

```
; $Id: gaussfit.pro,v 1.13 2001/02/21 00:17:32 chris Exp $
;
; Copyright (c) 1982-2001, Research Systems, Inc. All rights reserved.
; Unauthorized reproduction prohibited.
;
; 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.
;
PRO GAUSS_FUNCT,X,A,F,PDER
  COMPILE_OPT idl2, hidden
  ON_ERROR,2 ;Return to caller if an error occurs
  n = n_elements(a)
  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] = EZ ;COMPUTE PARTIALS
  if a[2] ne 0. then PDER[* ,1] = A[0] * EZ * Z/A[2]
  PDER[* ,2] = PDER[* ,1] * Z
  if n gt 3 then PDER[* ,3] = 1.
  if n gt 4 then PDER[* ,4] = X
  if n gt 5 then PDER[* ,5] = X^2
  RETURN
END

```

```

;+
; 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:
; KEYWORD INPUTS:
; ESTIMATES = optional starting estimates for the parameters of the
; equation. Should contain NTERMS (6 if NTERMS is not
; provided) elements.
; 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:
; The initial estimates are either calculated by the below procedure
; or passed in by the caller. Then the function CURVEFIT is called
; to find the least-square fit of the gaussian to the data.
;
; Initial estimate calculation:
; 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.
; DMS, RSI, Added ESTIMATES keyword.
;-
;
Function Gaussfit, x, y, a, NTERMS=nt, ESTIMATES = est

```

COMPILE_OPT idl2

```

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)

```

```

if n_elements(est) eq 0 then begin ;Compute estimates?

```

```

    if (nt gt 3) then begin
        ; For a Gaussian + polynomial, we need to subtract off the polynomial
        ; in order to get good estimates.
        degree = nt - 4 ; Fit a constant, straight line, or a quadratic
        c = poly_fit(x,y,degree,yf)
        yd = y - yf
    endif else begin

```

```

; Just fitting a Gaussian. Don't need to subtract off anything.
yd = y
c = 0d
endelse

if s[s[0]+1] ne 5 then begin ;If Y is not double, use float
  yd = float(yd)
  c = float(c)
endif

;x,y and subscript of extrema
ymax=max(yd, imax)
xmax=x[imax]
ymin=min(yd, imin)
xmin=x[imin]
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, c[2]]
endif else begin
  if nt ne n_elements(est) then message, 'ESTIMATES must have NTERM elements'
  a = est
endelse

return,curvefit(x,y,replicate(1.,n),a,sigmaa, $
function_name = "GAUSS_FUNCT") ;call curvefit
end

```
