
Subject: Patch for contrib code SUM.PRO
Posted by [wagener](#) on Tue, 19 Jan 1993 23:25:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

We discovered an error in the function 'SUM' in the User-contributed library, distributed with PV-Wave 4.0, (and probably IDL as well).

The error occurs, e.g. when summing over the first dimension (DIMENSION=0) and the product of the product of the other dimensions is gt 2^{15} . The fix involved replacing a call to 'FIX' with 'LONG', I also changed the loop counters to long integers (which is probably overkill in some cases).

```
;
;
; $Id: sum.pro,v 1.1 91/03/29 13:58:25 jeffry Exp $
;
FUNCTION SUM,ARRAY,DIMENSION
;+
; NAME:
; SUM
; PURPOSE:
; Total up an array over one of its dimensions.
; CATEGORY:
; Array manipulation
; CALLING SEQUENCE:
; a=sum(array,dimension)
; INPUTS:
; array = multi-dimensional array
; dimension = Dimension to sum over (first dimension = 0)
; OPTIONAL INPUT PARAMETERS:
; None.
; KEYWORD PARAMETERS:
; None.
; OUTPUTS:
; a = array of one dimension less than 'array' containing sums
;   of the input 'array' over the 'dimension+1'th dimension.
; OPTIONAL OUTPUT PARAMETERS:
; None.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; None.
; RESTRICTIONS:
; None.
; PROCEDURE:
; Either loops over the dimension to be sum'ed or over all other
; dimensions, depending on which requires fewer iterations of
; the loop.
```

```

; MODIFICATION HISTORY:
; ??-??-??: Created by ??
; 91/03/29: jeffry Exp
; 93-1-4:  Changed some loop counters from short to longword
; integers and the use of function 'fix' to 'long' to
; fix a problem, which occurs e.g. when summing over
; the first dimension ('dimension'=0) and the product
; of other dimensions is gt 2^15
; (Richard Wagener, wagener@bnl.gov)
;-
IF (N_PARAMS() LT 2) THEN BEGIN
  PRINT, '*** Function SUM must be called with two parameters:'
  PRINT, '          ARRAY , DIMENSION'
  RETURN, ARRAY
ENDIF
;
S = SIZE(ARRAY)
N_DIM = S(0)
IF N_DIM EQ 0 THEN BEGIN
  PRINT, '*** Variable must be an array, name= ARRAY, routine SUM.'
  RETURN, ARRAY
END ELSE IF (DIMENSION GE N_DIM) OR (DIMENSION LT 0) THEN BEGIN
  PRINT, '*** Dimension out of range, name= ARRAY, routine SUM.'
  RETURN, ARRAY
END ELSE IF N_DIM EQ 1 THEN BEGIN ;Trivial case, equivalent to TOTAL.
  F = TOTAL(ARRAY)
  RETURN, F
ENDIF
;
S2 = S(1+WHERE(INDGEN(N_DIM) NE FIX(DIMENSION)));Set up array for output variable.
F = MAKE_ARRAY(DIMENSION=S2, TYPE=S(S(0)+1))
;
; Calculate product of sizes of dimensions lower than, equal to, and higher
; than DIMENSION (NI,NJ,NK respectively).
;
NI = 1I
IF DIMENSION GT 0 THEN FOR M = 1I,DIMENSION DO NI = NI * S(M)
NJ = S(DIMENSION+1)
NK = 1I
IF DIMENSION LT N_DIM-1 THEN FOR M = DIMENSION+2I,N_DIM DO NK = NK * S(M)
;
; Set up index arrays.
;
XIK = LINDGEN( NI * NK )
XJ = LINDGEN( NJ )
NIJ = NI*NJ
;
; Choose whether it is more efficient to loop over NI and NK ...

```

```

;
;
IF NI*NK LT NJ THEN BEGIN
  FOR I = 0I,NJ-1 DO FOR K = 0I,NK-1 DO $
    F(I+NI*K) = TOTAL( ARRAY(I + NI*XJ + NIJ*K) )
;
;
; ... or over NJ.
;
;
END ELSE BEGIN
  XI = XIK MOD NI
  XK = long( XIK / NI )
  FOR J = 0I,NJ-1 DO F = F + ARRAY(XI + NI*J + NIJ*XK)
ENDELSE
RETURN,F
END
--
Tschuess ...rick... (wagener@bnl.gov)

```
