
Subject: Math bug and work around

Posted by [dudley](#) on Fri, 23 Sep 1994 04:07:00 GMT

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

Using IDL Version 3.0.0, taking the cube root of a negative number generates a floating point error, and returns the absolute value of the number. This is unfortunate since odd root stretches are both useful in bringing up low surface brightness features in background subtracted images that generally contain some negative values, and mathematically more sensible than the more conventional log stretch. Below is a work around solution to this problem called nthroot. Please let me know if you find it doing anything unexpected.

Chris Dudley

```
-----cut here-----  
FUNCTION Nthroot, data, n  
;+  
; NAME:  
;   
; nthroot  
;   
; PURPOSE:  
;   
; Take the nth root of real numbers. Avoids a bug using the ^ operator  
;   
; CATEGORY:  
;   
; math function  
;   
; CALLING SEQUENCE:  
;   
; result=nthroot(data,n)  
;   
; INPUTS:  
;   
; data: byte, integer, float, or double  
; n: byte, integer, or long  
;   
; OUTPUTS:  
;   
; result: float or double, unless error occurs in which case -1  
; 0^0 returns 1.  
;   
; RESTRICTIONS:  
;  
;
```

```

; complex, string, or structure data returns -1
;
; AUTHOR:
;
; Please report bugs in this routine to dudley@galileo.ifa.hawaii.edu
;
;-
sd = size(data)
sn = size(n)
IF sd(sd(0)+1) LT 1 OR sd(sd(0)+1) GT 5 OR sn(sn(0)+1) LT 1 OR $
  sn(sn(0)+1) GT 3 OR sn(sn(0)+2) NE 1 THEN BEGIN
    print, 'data must be byte, integer, long, float, or double.'
    print, 'n must be byte, integer, or long with only one element'
    return, -1
  ENDEF
IF sn(0) EQ 1 THEN n = n(0)
IF n NE 0 THEN sign_n = n/abs(n) ELSE BEGIN
  data = data*0.+1
  return, data
ENDELSE
odd = abs(n MOD 2)
IF (sd(0)+2) EQ 1 THEN BEGIN
  IF data NE 0 THEN sign_d = data/abs(data) ELSE return, data*0.
  IF sign_d EQ -1 AND NOT odd THEN BEGIN
    print, 'No even roots of negative numbers.'
    return, -1
  ENDEF
  return, sign_d*abs(data)^(1./n)
ENDEF
neg = where(data LT 0, count)
IF count NE 0 THEN BEGIN
  IF NOT odd THEN BEGIN
    print, 'No even roots of negative numbers.'
    return, -1
  ENDEF
  result = abs(data)^(1./n)
  result(neg) = -result(neg)
  return, result
ENDEF
return, data^(1./n)
END

```
