Subject: Re: About INTERPOLATE function.
Posted by wlandsman on Thu, 30 Jul 2015 03:55:37 GMT
View Forum Message <> Reply to Message

I think this a case of the location values being interpreted as floating point, and you are on a knife-edge when next to an NaN value

if a = [0,!values.f_nan,2,3]

then one expects interpolate(a,0) = 0 and interpolate(a,2) = 2

because the zeroth element has value of 0 and the second element has a value of 2.

But one also expects interpolate(a,0.00000000001)=NaN
and interpolate(a,1.99999999999(=NaN

because any type of interpolation/calculation involving an NaN value -- no matter how tiny -- should give NaN as an answer. If the location values are interpreted as floating point then inputing a 0 could mean a value very slightly smaller or larger. Thus IDL gives

IDL> print,interpolate(a,0),interpolate(a,2)
NaN 2.00000

though evidently with the /grid keyword the rounding works differently

My advice is -- don't use INTERPOLATE() when you have NaN values.

On Wednesday, July 29, 2015 at 9:21:04 PM UTC-4, Dae-Kyu Shin wrote:
> Example 1.
>
> a = findgen(4)
> a[1] = !values.f_nan
> print, a
> 0.000000 NaN 2.00000 3.00000
>
> print, interpolate(a, [0, 1, 2, 3])
> NaN NaN 2.00000 3.00000
>
> Example 2.
> a = transpose(a)
> print, interpolate(a, 0, [0, 1, 2, 3], /grid)
> 0.000000
> NaN
> 2.00000
> 3.00000

>
> different result~!!
> is this right output?