Subject: Re: where() and IEEE NaN and other stuff...
Posted by T Bowers on Mon, 08 Nov 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Pavel Romashkin <promashkin@cmdl.noaa.gov> wrote in message
news:3823685B.B76B91A7@cmdl.noaa.gov...
> I am not sure if trying to use WHERE on NaN is legitimate. NaN is a
> missing value by definition, and the request to WHERE those values is
> senseless. However, I'd too better like WHERE to complain than to
> quietly return some array.
OK,
for i=0, n_elements(a)-1 do $
  if a[i] EQ !Values.F_NaN then print, i
;)

> It does not seem to be a *practical* problem because NaNs don't sneak
> into your arrays unless you place them in. And if you do, then handle
> them accordingly.

Huh?? Who says I put 'em there. I have text files from an instrument
that has NaN's. IDL reads these values fine. I then wanna show the
data as a series of IDLgrSurface plots (data points and solid surface).
I wanna show it as the original data points (no interp, the data's
already on an irregular grid with the NaN's in place), the interpolated
surface with the NaN's in the surface (NaN's will show as "holes" in
the surface), and a surface with the values interpolated "through"
the NaN regions so I get a continuous surface. Since trigrid will not
interpolate thru the NaN's, I have to remove them from the data to get
this continuous surface. An option would be to eliminate my NaN's at
the source file:

read from file to dataArrayWithNaN
  read all data, NaN's included
triangulate, ...
gridDataWithNaN = trigrid(dataArrayWithNaN, MISSING=!Values.F_NaN, ...)
 ;I'll get a surface with holes where there's NaN's
If the user clicks the "Plot data without NaN's" menu item, then...
 read from file to dataArrayNoNaN
  if it's a NaN, don't read it in
 triangulate, ...
 gridDataNoNaN = trigrid(dataArrayNoNaN,...) ;I'll get a smooth surface

I'd rather just read all the data (including NaN's) once then do something
like:
dataArrayNoNaN = dataArrayWithNaN[where(dataArrayWithNaN NE !Values.F_NaN)]
as I needed it. This is alot better than 2 reads because my data spans
multiple files and collectively is definately on the high side of large.
Of course, this is just how I implemented it before I got any responses

to the original post.

Incidentally, if I have arrays data[m,n], x[m], and y[n], and say the data
array has some NaN's in it, then... (x across the top, y down the side,
data a [3,3], simplistic, but demonstrative)

```
x = [200, 201, 202]
y = [0,1,2,3]
data = [ [!Values.F_NaN, !Values.F_NaN, !Values.F_NaN], [25,20,35], $
        [15,20,18], [15,!Values.F_NaN,18] ]
```

Just for kicks, looks like this in the file...

```
  200 201 202
0 NaN NaN NaN
1  25  20  35
2  15  20  18
3  15 NaN  18
```

```
surfaceA = OBJ_NEW('IDLgrSurface', data, x, y);set data,x,y thru constructor
surfaceB = OBJ_NEW('IDLgrSurface')
surfaceB->setProperty, DATAZ=data, DATAX=x, DATAY=y ;set data,x,y thru
method
```

Ok, now surfaceA and surfaceB are set to the same data. Now if I...

```
surfaceA->GetProperty, XRange=xrangeA, YRange=yrangeA, ZRange=zrangeA
surfaceB->GetProperty, XRange=xrangeB, YRange=yrangeB, ZRange=zrangeB
```

The ranges for the arrays are different, specifically:
yrangeA ([0,3]) isn't the same as yrangeB ([1,3]).
Not exactly what I would expect. What else is the constructor doing
differently than the setProperty method?

```
IDL> help, /structure, !version
** Structure !VERSION, 5 tags, length=40:
   ARCH            STRING    'x86'
   OS              STRING    'Win32'
   OS_FAMILY       STRING    'Windows'
   RELEASE         STRING    '5.2'
   BUILD_DATE      STRING    'Oct 30 1998'
```