Subject: Re: Turning off math error checking for a code block
Posted by Liam E. Gumley on Thu, 17 Jan 2002 17:26:36 GMT
View Forum Message <> Reply to Message

Kenneth Bowman wrote:

```
>
>  I have an array x that is likely to have missing values in it, indicated by NaN's.
>  I would like to search the array for values less than x_min.  Because of the NaN's,
>  WHERE generates a floating point error, e.g.,
>
>  IDL> print, x
>       0.00000          NaN      2.00000      3.00000
>  IDL> print, where(x lt 2.0)
>             0
>  % Program caused arithmetic error: Floating illegal operand
>
>  As best I understand the interaction between !EXCEPT and CHECK_MATH,
>  in order to suppress this error message, while still checking errors elsewhere
>  in the code, I must do the following:
>
>  error       = CHECK_MATH(/PRINT)           ;If any errors have occurred, print
>  save_except = !EXCEPT                      ;Save current exception flag
>  !EXCEPT     = 0                            ;Set exception flag to 0
>  i           = WHERE(x LT x_min, ni)        ;Find all x < x_min
>  error       = CHECK_MATH()                 ;Clear accumulated error status
>  !EXCEPT     = save_except                  ;Restore exception flag
>
>  Am I making this harder than it needs to be?
```

The FINITE function returns 1 where the argument is finite, and 0 where the argument is infinite *or* NaN (see p. 134 of my book). Try the following:

```
x_min = 2.0
index = where(finite(x) eq 1, count)
if (count gt 0) then print, where(x[index] lt x_min)
```

Cheers,
Liam.
Practical IDL Programming
http://www.gumley.com/