Subject: Re: avoiding "floating illegal operand" errors with /nan keyword in mean
Posted by Paul Levine on Wed, 21 Aug 2013 02:17:41 GMT
View Forum Message <> Reply to Message

On 2013-08-21 00:26:02 +0000, wlandsman said:

> Does setting !except=0 work for you? --Wayne

Thank you for the suggestion. It does work insofar as the error message does not appear, though when I check_math I get 128, so isn't that more or less the equivalent of simply ignoring the errors?

I realized that the example code I banged out in my initial post had a couple typos; here is some double-checked code that presents the problem I am having, guaranteed to run properly.

The example data is completely contrived, though it essentially a much smaller version of the data I am dealing with (it simulates 9 "images" with 3 columns and 3 rows where I want to average every 3 images).

In example 1, my method of checking works, and the math error is avoided, because in the second iteration of the loop, the entire subset is NaN so max(finite(subset)) is zero. The only difference between examples 1 and 2 is in the second line, which sets certain elements of the array to NaN. In example 2, in the second iteration of the loop only one of the "pixels" is all NaN, which my method of checking does not catch, thus yielding the math error.

```
; Example 1
array = findgen(3,3,9)
array[*,*,3:5] = !VALUES.D_NAN
newarray = fltarr(3,3,3)
for j = 0, 2 do begin
  subset = array[*,*,j*3:j*3+2]
  if max(finite(subset)) eq 1 then begin
    newarray[0,0,j] = mean(subset, dimension=3, /nan)
  endif else begin
    newarray[0,0,j] = make_array(3,3,value=!VALUES.D_NAN)
    print, 'else case happens'
  endelse
endfor

; Example 2
array = findgen(3,3,9)
array[2,2,3:5] = !VALUES.D_NAN
```

```
newarray = fltarr(3,3,3)
for j = 0, 2 do begin
  subset = array[*,*,j*3:j*3+2]
  if max(finite(subset)) eq 1 then begin
    newarray[0,0,j] = mean(subset, dimension=3, /nan)
  endif else begin
    newarray[0,0,j] = make_array(3,3,value=!VALUES.D_NAN)
    print, 'else case happens'
  endelse
endfor
```