Subject: Re: avoiding "floating illegal operand" errors with /nan keyword in mean
Posted by Fabzi on Wed, 21 Aug 2013 21:11:45 GMT
View Forum Message <> Reply to Message

Hi Paul,

I also had the same problem:
https://groups.google.com/forum/#!topic/comp.lang.idl-pvwave /XmPXtQE6VZ0

It is not necessary to use a for loop to avoid the warnings, you just
need to write your own MEAN() where you check for cases with
TOTAL(FINITE(data), dimension) eq 0 but you have to decide if it worth
it or not...

On 08/21/2013 01:37 AM, Paul Levine wrote:
> I have an array with 2800 rows, 2800 columns, and 120 layers, where each
> layer is a month from a 10-year time series). I would like to calculate
> annual means, so I will end up with an array that is 2800 x 2800 x 10. So
>
> for j = 0, 9 do begin
> ; make one year subset of 2800x1800x12
> subset = array[*,*,j*12:j*12+12]
> newarray[0,0,j] = mean(newarray, dimension=3, /nan)
> endfor
>
> I am using the /nan keyword because there are a lot of NaNs in the
> data. As a result, I get
> % Program caused arithmetic error: Floating illegal operand
> whenever the mean function tries to calculate an average completely out
> of NaN values.
>
> I know that I could just ignore the error, because the results are what
> I want them to be, but I'm sure it would be better to figure out how to
> prevent the error.
>
> So, I tried the following method of checking to see whether I'm dealing
> with all NaNs
>
> for j = 0, 9 do begin
> ; make one year subset of 2800x1800x12
> subset = array[*,*,j*12:j*12+12]
> if max(finite(subset)) eq 1 then begin
> newarray[0,0,j] = mean(newarray, dimension=3, /nan)
> endif else begin
> newarray[0,0,j] = make_array(2800,2800,value=!VALUES.D_NAN)
> endelse
> endfor

>
> This eliminates the error for any situation where the entire subset is
> NaN. However, because the mean function is essentially calculating
> 2800x2800 means of 12 elements each, there are instances where only one
> or a few of those 12-element sets are completely NaN, which is not
> caught by my method of checking, so I still get the error message.
>
> The only way I can imagine extending my checking method would be to loop
> through every row and column of the data set, calculating each mean one
> at a time so that I can check whether or not all 12 elements are NaN.
> Of course, that would be incredibly inefficient, so I would not
> seriously entertain that idea.
>
> Is there a better solution out there, or should I just suck it up and
> live with the error message?
>
