

---

Subject: Re: Ug...Floating Illegal Operand...leave me alone!

Posted by [weitkamp](#) on Wed, 18 Jul 2001 12:12:27 GMT

[View Forum Message](#) <> [Reply to Message](#)

---

Hi,

Concerning Jeff's statement:

> zeros. (I don't think there's a way to make IDL halt at the time the  
> error occurs, as it does with other errors.)

I think that's true, but you can at least make IDL print out the error message at the moment where it occurs (instead of the default behavior of message output after program termination) by setting the system variable !EXCEPT to 2. Combining that with the print command Jeff has suggested inside the "for" loop can help track down the reason for the error message Matt gets.

Timm

--

Timm Weitkamp

European Synchrotron Radiation Facility (ESRF)

38043 Grenoble, France

Jeff Guerber <[jguerber@icesat2.gsfc.nasa.gov](mailto:jguerber@icesat2.gsfc.nasa.gov)> wrote in message  
news:<[Pine.GHP.4.32.0107172133450.433-100000@icesat2.gsfc.nasa.gov](mailto:Pine.GHP.4.32.0107172133450.433-100000@icesat2.gsfc.nasa.gov)>...

> Matt,

>

> It sounds to me like some data in your 73rd dataset is likely producing  
> a situation where both (b\*c) and d are 0.0 (or rather, some elements of  
> b\*c and d, since you imply they're arrays). They would not necessarily be  
> 0.0 in the input dataset -- if b, c, and d are themselves the result of  
> calculations, you could get zeros due to eg. underflows, in which the  
> result of a calculation is so small that the computer has to set it to  
> zero (for example, 1.e-30 \* 1.e-30).

>

> Try printing out b\*c and d before you do the division, and look for  
> zeros. (I don't think there's a way to make IDL halt at the time the  
> error occurs, as it does with other errors.) Also, try printing out a and  
> looking for the value "NaN" (a floating-point "Not-a-Number", which is  
> the result of 0./0.) Once variable (or array element) becomes a NaN, the  
> NaN just propagates through any further calculations that the variable  
> takes part in, eventually showing up on the output. (That's one of the  
> really nice things about IEEE arithmetic -- you can tell which results are  
> illegitimate.) Hope this helps,

>

> Jeff Guerber

> Raytheon ITSS / NASA GSFC code 971  
> (but I don't speak for either one)  
>  
> On 17 Jul 2001, Matt wrote:  
>  
>> Hey everyone,  
>>  
>> I'm rather new to the world of IDL...so bear with me. I've been  
>> attempting to debug a program all afternoon...but the damn "Program  
>> caused an arithmetic error: Floating illegal operand" message won't go  
>> away no matter what I do. I've narrowed the problem down to an  
>> expression in the following form:  
>>  $a = a + ((b * c) / d)$  where all the variables are of DOUBLE precision. They  
>> were originally FLOAT but I changed them all and thought it may have  
>> been the problem, but apparently it wasn't.  
>>  
>> A little more detail on the program: I'm working through about 73  
>> datasets using a for loop. The expression above is embedded in an  
>> additional for loop so as to perform some manipulations  $((b * c) / d)$  on  
>> arrays and sum all these manipulations into 'a.' The interesting  
>> thing is, I can run 1, 10, or even 72 datasets and not get the  
>> "floating illegal operand" error. Only when I run all 73 datasets at  
>> once do I receive the message when the program is complete. I'm  
>> getting results as well...I just question whether or not they're right  
>> because of the arithmetic error.  
>>  
>> I'm not sure if the Check\_math function would help out here or not. I  
>> tried using it earlier but I couldn't actually figure out how to use  
>> it properly. Can anyone help me out here at all? Either helping me  
>> diagnose this or helping me figure out how to use the check\_math  
>> function? I'm really stumped and extremely frustrated. Thanks.  
>>  
>> Matt  
>> randomguy79@yahoo.com

---