Subject: Re: How do I prevent underflow errors?
Posted by davidf on Tue, 16 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Phillip & Suzanne David (pdavid@earthling.net) writes:

> I have a large array of data that I'd like to plot with the contour routine.
> However, the dynamic range of the data is very large, with values as large as
> 1e36 and as small as 1e-40. I noticed that contour accepts float data, not
> double data. This data is outside the range of float data, so it needs to be
> scaled for the contour routine. I don't really care to differentiate the
> 1e-40 from 0, but would like to be able to handle values up to the 1e36. I
> was going to scale the data by the largest value (i.e.,
> PlotData=Float(Data/Max(Abs(Data)))). This puts the data in the range of -1.0
> to 1.0. This should be fine for Contour, but I get an underflow error when
> converting from double data to float data. I understand that the data will
> come out with a 0 instead of 1e-76, and don't really care. How do I get IDL
> to ignore the underflow and just convert the value?

I don't think it is possible to avoid underflow error messages
(although I would love to be proved wrong about this). It might
help to think of them not as *error* messages, but as helpful
informational messages from a concerned computer. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438 E-Mail: davidf@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: How do I prevent underflow errors?
Posted by Phillip & Suzanne on Wed, 17 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Unfortunately, our data is NOT all positive, so this doesn't work. However,
it appears that the really large numbers are due to someone putting a "huge"
value in where there were holes in the data. We're trying to get them to put
a smaller "huge" number (1e10?) in instead!

Phillip

Ethan Alpert wrote:

>
> Is your data all possitive? Have you considered contouring the log of the data?
> The contours generated by taking the log of the data will be the same if
> you set the contour intervals correctly.
>
> This should solve the problem.
>
> -ethan alpert

Subject: Re: How do I prevent underflow errors?
Posted by meron on Wed, 17 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <36CA865E.2381C0E9@uni-c.dk>, Michael Viskum <Michael.Viskum@uni-c.dk> writes:
>
>
> Craig Markwardt wrote:
>
>> Phillip & Suzanne David <pdavid@earthling.net> writes:
>>>
>>> I have a large array of data that I'd like to plot with the contour routine.
>>> However, the dynamic range of the data is very large, with values as large as
>>> 1e36 and as small as 1e-40. I noticed that contour accepts float data, not
>>> double data. This data is outside the range of float data, so it needs to be
>>> scaled for the contour routine. I don't really care to differentiate the
>>> 1e-40 from 0, but would like to be able to handle values up to the 1e36. I
>>> was going to scale the data by the largest value (i.e.,
>>> PlotData=Float(Data/Max(Abs(Data)))). This puts the data in the range of -1.0
>>> to 1.0. This should be fine for Contour, but I get an underflow error when
>>> converting from double data to float data. I understand that the data will
>>> come out with a 0 instead of 1e-76, and don't really care. How do I get IDL
>>> to ignore the underflow and just convert the value?
>>>
>>> Phillip
>>>
>>
>> Greetings,
>>
>> Try the following with check_math(), which is documented in IDL.
>>
>> dummy = check_math(1, 1)
>> .... commands with math errors are placed here ...
>> dummy = check_math(0, 0)
>>
>> Most of your messages will go away (except for maybe the last one).
>>
>> Craig

>>
>> --
>> -- ----------
>> Craig B. Markwardt, Ph.D. EMAIL: craigmnet@astrog.physics.wisc.edu
>> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
>> -- ----------
>
> Hi,
>
> Try also to have a look at the !EXCEPT system variable. It can have 3 possible
> values.
> If !EXCEPT=0 then IDL will not report any exceptions.
>
The problem with these approaches is that they eliminate all math
error messages, including some you may want to see. What you would
want is to eliminate only specific error messages, the "underflow"
ones in this case. I've been in touch with RSI on this issue,
suggesting an upgrade to check-math. Specifically I suggested adding
a keyword variable which'll enable you to specify (through a bit mask
or so) which errors you want to check and clear. They promised to
look into it but I haven't seen results yet.

Anyway, for the moment I wrote a routine which deals with this issue,
se below. As it is calling other routines from my library, one should
really copy the library to make it usable. The whole library may be
found on cars3.uchicago.edu , it is accessible through anonymous FTP.
do a CD to MIDL after logging in, it is all there.

Function FPU_fix, x, no_abs = nab

```
;+
; NAME:
; FPU_FIX
; VERSION:
; 3.0
; PURPOSE:
; Clears Floating Point Underflow errors, setting the offending values to
; zero.
; CATEGORY:
; Programming.
; CALLING SEQUENCE:
; Result = FPU_FIX( X)
; INPUTS:
;    X
; Arbitrary.
; OPTIONAL INPUT PARAMETERS:
```

```
; None.
; KEYWORD PARAMETERS:
;    /NO_ABS
; Switch.  If set, uses value instead of absolute value for comparison
; with machine minimum.  For internal use only.
; OUTPUTS:
; If the input is of any numeric type, returns the input, with the
; possible substitution of 0 for all occurences of Floating Point
; Underflow.  A non-numeric input is returned as is.
; OPTIONAL OUTPUT PARAMETERS:
; None.
; COMMON BLOCKS:
; None.
; SIDE EFFECTS:
; None.
; RESTRICTIONS:
; None.
; PROCEDURE:
; Straightforward.  Uses the system routines CHECK_MATH and MACHAR.  Also
; calls ISNUM and M_ABS from MIDL.
; MODIFICATION HISTORY:
; Created 30-AUG-1998 by Mati Meron.
;-

    on_error, 1
    fpucod = 32
    matherrs = ['Integer divided by zero','Integer overflow','','',$
  'Floating-point divide by zero','Floating-point underflow',$
  'Floating-point overflow','Floating-point operand error']

    chem = check_math()
    if Isnum(x,type = typ) and chem gt 0 then begin
 if chem eq fpucod then begin
     sinf = machar(double = Isnum(x,/double))
     if keyword_set(nab) then dum = where(x lt sinf.xmin, nuf) $
     else dum = where(M_abs(x) lt sinf.xmin, nuf)
     if nuf gt 0 then x(dum) = 0
 endif else message, matherrs(round(alog(chem)/alog(2)))
    endif

    return, x
end

Mati Meron                    | "When you argue with a fool,
meron@cars.uchicago.edu       |  chances are he is doing just the same"
```

Subject: Re: How do I prevent underflow errors?
Posted by Ethan Alpert on Wed, 17 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Phillip & Suzanne David wrote:

> I have a large array of data that I'd like to plot with the contour routine.
> However, the dynamic range of the data is very large, with values as large as
> 1e36 and as small as 1e-40. I noticed that contour accepts float data, not
> double data. This data is outside the range of float data, so it needs to be
> scaled for the contour routine. I don't really care to differentiate the
> 1e-40 from 0, but would like to be able to handle values up to the 1e36. I
> was going to scale the data by the largest value (i.e.,
> PlotData=Float(Data/Max(Abs(Data)))). This puts the data in the range of -1.0
> to 1.0. This should be fine for Contour, but I get an underflow error when
> converting from double data to float data. I understand that the data will
> come out with a 0 instead of 1e-76, and don't really care. How do I get IDL
> to ignore the underflow and just convert the value?

Is your data all possitive? Have you considered contouring the log of the data?
The contours generated by taking the log of the data will be the same if
you set the contour intervals correctly.

This should solve the problem.

-ethan alpert

>
> Phillip

--
---------- ----------
Ethan Alpert ethan@ncar.ucar.edu
---------- ----------

Subject: Re: How do I prevent underflow errors?
Posted by Michael Viskum on Wed, 17 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Craig Markwardt wrote:

> Phillip & Suzanne David <pdavid@earthling.net> writes:
>>
>> I have a large array of data that I'd like to plot with the contour routine.
>> However, the dynamic range of the data is very large, with values as large as
>> 1e36 and as small as 1e-40. I noticed that contour accepts float data, not
>> double data. This data is outside the range of float data, so it needs to be

>> scaled for the contour routine. I don't really care to differentiate the
>> 1e-40 from 0, but would like to be able to handle values up to the 1e36. I
>> was going to scale the data by the largest value (i.e.,
>> PlotData=Float(Data/Max(Abs(Data)))). This puts the data in the range of -1.0
>> to 1.0. This should be fine for Contour, but I get an underflow error when
>> converting from double data to float data. I understand that the data will
>> come out with a 0 instead of 1e-76, and don't really care. How do I get IDL
>> to ignore the underflow and just convert the value?
>>
>> Phillip
>>
>
> Greetings,
>
> Try the following with check_math(), which is documented in IDL.
>
> dummy = check_math(1, 1)
> .... commands with math errors are placed here ...
> dummy = check_math(0, 0)
>
> Most of your messages will go away (except for maybe the last one).
>
> Craig
>
> --
> ----------
> Craig B. Markwardt, Ph.D. EMAIL: craigmnet@astrog.physics.wisc.edu
> Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
> ----------

Hi,

Try also to have a look at the !EXCEPT system variable. It can have 3 possible values.
If !EXCEPT=0 then IDL will not report any exceptions.

cheers

Michael

--

Michael Viskum, Ph.D.
UNIï¿½C, Scientific Computing
Olof Palmes Alle 38
DK-8200 Aarhus N, Denmark

Phone (+45) 8937 6614 - Fax (+45) 8937 6677
Michael.Viskum@uni-c.dk
http://www.uni-c.dk

Subject: Re: How do I prevent underflow errors?
Posted by Craig Markwardt on Wed, 17 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Phillip & Suzanne David <pdavid@earthling.net> writes:
>
> I have a large array of data that I'd like to plot with the contour routine.
> However, the dynamic range of the data is very large, with values as large as
> 1e36 and as small as 1e-40. I noticed that contour accepts float data, not
> double data. This data is outside the range of float data, so it needs to be
> scaled for the contour routine. I don't really care to differentiate the
> 1e-40 from 0, but would like to be able to handle values up to the 1e36. I
> was going to scale the data by the largest value (i.e.,
> PlotData=Float(Data/Max(Abs(Data)))). This puts the data in the range of -1.0
> to 1.0. This should be fine for Contour, but I get an underflow error when
> converting from double data to float data. I understand that the data will
> come out with a 0 instead of 1e-76, and don't really care. How do I get IDL
> to ignore the underflow and just convert the value?
>
> Phillip
>

Greetings,

Try the following with check_math(), which is documented in IDL.

```
dummy = check_math(1, 1)
.... commands with math errors are placed here ...
dummy = check_math(0, 0)
```

Most of your messages will go away (except for maybe the last one).

Craig

--
-- --------------
Craig B. Markwardt, Ph.D. EMAIL: craigmnet@astrog.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------

Subject: Re: How do I prevent underflow errors?
Posted by meron on Thu, 18 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

In article <7afle2$lkg@post.gsfc.nasa.gov>, thompson@orpheus.nascom.nasa.gov (William Thompson) writes:
> meron@cars3.uchicago.edu writes:
>
> (stuff deleted)
>
>> The problem with these approaches is that they eliminate all math
>> error messages, including some you may want to see. What you would
>> want is to eliminate only specific error messages, the "underflow"
>> ones in this case. I've been in touch with RSI on this issue,
>> suggesting an upgrade to check-math. Specifically I suggested adding
>> a keyword variable which'll enable you to specify (through a bit mask
>> or so) which errors you want to check and clear. They promised to
>> look into it but I haven't seen results yet.
>
> (stuff deleted)
>
> I think we'd all like to get rid of those underflow error messages, while still
> letting other error messages through.
>
Well, I certainly would like to see them gone. They ain't nothing but
a nuisance.

Mati Meron | "When you argue with a fool,
meron@cars.uchicago.edu | chances are he is doing just the same"

Subject: Re: How do I prevent underflow errors?
Posted by thompson on Thu, 18 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

meron@cars3.uchicago.edu writes:

(stuff deleted)

> The problem with these approaches is that they eliminate all math
> error messages, including some you may want to see. What you would
> want is to eliminate only specific error messages, the "underflow"
> ones in this case. I've been in touch with RSI on this issue,
> suggesting an upgrade to check-math. Specifically I suggested adding
> a keyword variable which'll enable you to specify (through a bit mask
> or so) which errors you want to check and clear. They promised to
> look into it but I haven't seen results yet.

(stuff deleted)

I think we'd all like to get rid of those underflow error messages, while still
letting other error messages through.

Bill Thompson

Subject: Re: How do I prevent underflow errors?
Posted by Ethan Alpert on Fri, 19 Feb 1999 08:00:00 GMT
View Forum Message <> Reply to Message

Phillip & Suzanne David wrote:

> Unfortunately, our data is NOT all positive, so this doesn't work. However,
> it appears that the really large numbers are due to someone putting a "huge"
> value in where there were holes in the data. We're trying to get them to put
> a smaller "huge" number (1e10?) in instead!

Use the idl operator '<' which can be usIDL> var = [1,2,1e38,1,1e12]ed to convert
every number over
a certain threshold.

i.e:
```
IDL> var = [1,2,1e38,1,1e12]
IDL> print,var
     1.00000      2.00000  1.00000e+38      1.00000  1.00000e+12
IDL> print,var<1e10
     1.00000      2.00000  1.00000e+10      1.00000  1.00000e+10
IDL>
```

1e10 is inserted where ever the value of var is greater than 1e10.

-ethan

--
Ethan Alpert ethan@ncar.ucar.edu
