Subject: Floating underflow in a plot
Posted by Miguel on Tue, 14 Apr 2015 07:50:03 GMT
View Forum Message <> Reply to Message

Hi,

I'm trying to plot an implicit function f(x,y)=a by computing the function for many values of x and y and plotting the result.
But when the number of point (x,y) become too large, there is a problem of floating underflow.
Here is my code :

```
-------------------------------------------------------
set_plot, 'ps'
device, filename="Essai",/color, bits_per_pixel=8
loadct,13, /silent
!EXCEPT=2
close,/all
plot,[1,1],xrange=[-30,30],yrange=[-30,100],psym=3
openw,1,"bla.cat"

l=(dindgen(45)+0)/1*!PI/180
d=dindgen(1000)/10.+0
l_n=0
inc_color=0
Ro=8.5
To=220

resultat=dblarr(3,n_elements(l)*n_elements(d))
foreach i,l do begin
  foreach j,d do begin
    R=sqrt(j^2+Ro^2-2*Ro*j*cos(i))
 if (R NE 0.) then begin
    T=To*(1.00767*(R/Ro)^(0.0394)+0.00712)
    V=Ro*sin(i)*((T/R)-(To/Ro))
    resultat(0,l_n)=i
    resultat(1,l_n)=j
    resultat(2,l_n)=V
printf,1,-sin(!PI-i)*j,-cos(!PI-i)*j;,i*180/!PI,j,V
    l_n=l_n+1
 endif
  endforeach
endforeach

k=min(resultat(2,*))
while ( k LE max(resultat(2,*)) ) do begin
ind=where(resultat(2,*) GE k AND resultat(2,*) LE k+0.5)
if (ind(0) NE -1.) then begin
trajectoire=resultat(*,ind)
```

```
 oplot,-sin(!PI-trajectoire(0,*))*trajectoire(1,*),-cos(!PI-t
rajectoire(0,*))*trajectoire(1,*),psym=3,color=fix(inc_color ),NSUM=1
endif
k=k+0.5
inc_color=inc_color+1
if (inc_color EQ 256) then inc_color=0
endwhile
close,1
```

Th bla.cat file contains all the points and when I plot them, it seems fine (except for the last value of l)

What is the problem here =

Thanks
Miguek

Subject: Re: Floating underflow in a plot
Posted by Craig Markwardt on Tue, 14 Apr 2015 11:32:06 GMT
View Forum Message <> Reply to Message

On Tuesday, April 14, 2015 at 3:50:06 AM UTC-4, Miguel wrote:
> I'm trying to plot an implicit function f(x,y)=a by computing the function for many values of x and y and plotting the result.
> But when the number of point (x,y) become too large, there is a problem of floating underflow.
...
>
> Th bla.cat file contains all the points and when I plot them, it seems fine (except for the last value of l)
>
> What is the problem here =

Put aside the floating point underflow message, is there an actual problem with your result? This error indicates a numerical loss of precision but it doesn't always indicate a problem. It's like a butcher who tells you he needs to trim a little more fat than usual.

Craig

Subject: Re: Floating underflow in a plot
Posted by Heinz Stege on Tue, 14 Apr 2015 11:36:56 GMT
View Forum Message <> Reply to Message

Hi Miguel,

there is a system variable named !EXCEPT. You can change its value to
!EXCEPT=2. This makes IDL running slower. But the floating underflow
message should be accompanied by another message, which tells you the
line of the code, where the floating underflow happens.

HTH, Heinz

Subject: Re: Floating underflow in a plot
Posted by Miguel on Tue, 14 Apr 2015 15:53:39 GMT
View Forum Message <> Reply to Message

> Put aside the floating point underflow message, is there an actual problem with your result?
This error indicates a numerical loss of precision but it doesn't always indicate a problem. It's like a butcher who tells you he needs to trim a little more fat than usual.
>
> Craig

The problem is that the data are computed but some of them are not plotted with oplot,
I can plot the missing data by doing it separately (loosing the color codification) or I have to decrease the number of points in d.

Miguel

Subject: Re: Floating underflow in a plot
Posted by Miguel on Tue, 14 Apr 2015 15:55:05 GMT
View Forum Message <> Reply to Message

El martes, 14 de abril de 2015, 13:36:44 (UTC+2), Heinz Stege escribió:
> Hi Miguel,
>
> there is a system variable named !EXCEPT. You can change its value to
> !EXCEPT=2. This makes IDL running slower. But the floating underflow
> message should be accompanied by another message, which tells you the
> line of the code, where the floating underflow happens.
>
> HTH, Heinz

The !EXCEPT=2 is actually in the code (line 4) and the problem arises at the line where oplot is used.

Miguel

Subject: Re: Floating underflow in a plot
Posted by Heinz Stege on Tue, 14 Apr 2015 18:28:37 GMT
View Forum Message <> Reply to Message

On Tue, 14 Apr 2015 08:55:05 -0700 (PDT),
miguelfigueirasebastiao@gmail.com wrote:

> El martes, 14 de abril de 2015, 13:36:44 (UTC+2), Heinz Stege escribió:
>> Hi Miguel,
>>
>> there is a system variable named !EXCEPT. You can change its value to
>> !EXCEPT=2. This makes IDL running slower. But the floating underflow
>> message should be accompanied by another message, which tells you the
>> line of the code, where the floating underflow happens.
>>
>> HTH, Heinz
>
> The !EXCEPT=2 is actually in the code (line 4) and the problem arises at the line where oplot is used.
>
Oh, yes, of cause. I was too lazy to look into the code in detail.
Sorry for this.

Seems to be a very strange error. I can't explain it. However, are you
really sure, that the floating underflow error leads to missing
points, as you say in the answer to Craig's post?

You can eliminate the floating-underflow-error by skipping the points
near the position x=0.0 and y=0.0. This can be done be defining
```
  eps=(machar(/double)).eps
```
somewhere in the head of your code, and replacing the oplot command by
following lines:
```
  x=-sin(!PI-trajectoire(0,*))*trajectoire(1,*)
  y=-cos(!PI-trajectoire(0,*))*trajectoire(1,*)
  ii=where(abs(x) ge eps and abs(y) ge eps,count)
  if count ge 1 then $
    oplot,x[ii],y[ii],psym=3,color=fix(inc_color),NSUM=1
```

This should not make a visible change to your plot. However for me the
plot still looks some kind of "incomplete".

Cheers, Heinz

Subject: Re: Floating underflow in a plot
Posted by Lajos Foldy on Tue, 14 Apr 2015 20:04:16 GMT
View Forum Message <> Reply to Message

On Tuesday, April 14, 2015 at 5:55:06 PM UTC+2, miguelfigue...@gmail.com wrote:
> El martes, 14 de abril de 2015, 13:36:44 (UTC+2), Heinz Stege escribió:
>> Hi Miguel,
>>
>> there is a system variable named !EXCEPT. You can change its value to
>> !EXCEPT=2. This makes IDL running slower. But the floating underflow
>> message should be accompanied by another message, which tells you the
>> line of the code, where the floating underflow happens.
>>
>> HTH, Heinz
>
> The !EXCEPT=2 is actually in the code (line 4) and the problem arises at the line where oplot is used.
>
> Miguel

Set !EXCEPT to 0 and add "if check_math() ne 0 then stop" after the oplot line. Now IDL will stop on the underflow and you can examine the input to oplot.

regards,
Lajos

Subject: Re: Floating underflow in a plot
Posted by Miguel on Wed, 15 Apr 2015 17:58:13 GMT
View Forum Message <> Reply to Message

> Seems to be a very strange error. I can't explain it. However, are you
> really sure, that the floating underflow error leads to missing
> points, as you say in the answer to Craig's post?

The underflow problem was not the cause of the error

I found the problem : the subscript l_n for the array "resultat" can be very high so I needed this variable to have a double precision.

l_n=double(l_n) fixed the problem.

Thank you for your help ;)

Miguel

Subject: Re: Floating underflow in a plot
Posted by Jeremy Bailin on Wed, 15 Apr 2015 19:04:20 GMT
View Forum Message <> Reply to Message

On Wednesday, April 15, 2015 at 12:58:16 PM UTC-5, miguelfigue...@gmail.com wrote:
>> Seems to be a very strange error. I can't explain it. However, are you
>> really sure, that the floating underflow error leads to missing
>> points, as you say in the answer to Craig's post?
>
> The underflow problem was not the cause of the error
>
> I found the problem : the subscript l_n for the array "resultat" can be very high so I needed this variable to have a double precision.
>
> l_n=double(l_n) fixed the problem.
>
> Thank you for your help ;)
>
> Miguel

Ah, interesting. You don't really want to be using a double for something that's fundamentally an integer. But I see that you're actually using a short integer -- any integer that you don't know a priori will never go about 32000 should always be defined as a long:

l_n=0L

And anything that is really a floating-point number should not be declared as an integer:

To=220.0

-Jeremy.
