Subject: Re: 1e38 limit?
Posted by Spon on Fri, 27 Feb 2009 16:03:16 GMT
View Forum Message <> Reply to Message

On Feb 27, 3:55 pm, Paula <paulartcoe...@gmail.com> wrote:
> hello,
>
> i read float numbers from a text file, and eventually i need to
> multiply them by large numbers ~1e40.
> 'result' is the array that i read from the text file, and 'chi' is the
> array i'm analysing. i extract it with:
>
> chi = double(reform(result[3,*]))
>
> and then i do something like:
>
> chi = chi * 1e40
>
> but the routine apparently enters into an infinite loop, lots of CPU,
> RAM and HD usage, and i end up having to kill the process after a
> while. trial-and-error showed me that as long as i keep the
> multiplicative factor <= 1e38, everything runs smoothly.
>
> i tried even a stupid:
> chi = chi * 1e38
> chi = chi * 1e2
>
> but the routine still gets crazy with that.
>
> what is going on? how can i work it around?
>
> many thanks,
> paula

Hi Paula,

try
IDL> HELP,

Subject: Re: 1e38 limit?
Posted by Spon on Fri, 27 Feb 2009 16:07:32 GMT
View Forum Message <> Reply to Message

On Feb 27, 3:55 pm, Paula <paulartcoe...@gmail.com> wrote:
> hello,
>
> i read float numbers from a text file, and eventually i need to

> multiply them by large numbers ~1e40.
> 'result' is the array that i read from the text file, and 'chi' is the
> array i'm analysing. i extract it with:
>
> chi = double(reform(result[3,*]))
>
> and then i do something like:
>
> chi = chi * 1e40
>
> but the routine apparently enters into an infinite loop, lots of CPU,
> RAM and HD usage, and i end up having to kill the process after a
> while. trial-and-error showed me that as long as i keep the
> multiplicative factor <= 1e38, everything runs smoothly.
>
> i tried even a stupid:
> chi = chi * 1e38
> chi = chi * 1e2
>
> but the routine still gets crazy with that.
>
> what is going on? how can i work it around?
>
> many thanks,
> paula

I'll try that again :-\

Hi Paula,

try
IDL> Help, 1e40

Your multiplier is too large to be represented as a float, so you have
to use double precision.
What happens if you replace it with 1d40? You may still be getting
overflow, but you'll have a lot more room to play.

I find you can't read the following article often enough:
http://www.dfanning.com/math_tips/sky_is_falling.html

All the best,
Chris

Subject: Re: 1e38 limit?
Posted by paulartcoelho on Fri, 27 Feb 2009 17:00:29 GMT

View Forum Message <> Reply to Message

On Feb 27, 5:07 pm, Spon II <christoph.b...@gmail.com> wrote:
> I'll try that again :-\
>
> Hi Paula,
>
> try
> IDL> Help, 1e40
>
> Your multiplier is too large to be represented as a float, so you have
> to use double precision.
> What happens if you replace it with 1d40? You may still be getting
> overflow, but you'll have a lot more room to play.
>
> I find you can't read the following article often
enough:http://www.dfanning.com/math_tips/sky_is_falling.html
>
> All the best,
> Chris

many thanks Chris!

multiplying by 1d40 still gets me into the infinite loop, but it may
well be that i still have some 'e' to change to 'd' in other places of
the routine... and i'll go through a good read of 'sky is falling'
right now :)

p.

Subject: Re: 1e38 limit?
Posted by Chris[6] on Fri, 27 Feb 2009 17:39:42 GMT
View Forum Message <> Reply to Message

The overflow, by itself, shouldn't result in an infinite process - idl
should just complain about floating point overflow maybe.

What COULD be happening is that you have a loop in your code. You are
using a floating point value as the loop variable, and incrementing it
by a number much smaller than itself. When that happens, there isn't
enough precision to store the difference between the numbers before
and after increment, and the loop variable stays the same forever.
It's actually a floating point UNDERFLOW.

Maybe? any floating point loops?

chris

Subject: Re: 1e38 limit?
Posted by paulartcoelho on Fri, 27 Feb 2009 18:39:08 GMT
View Forum Message <> Reply to Message

i think i found it, but i'm not sure how to explain...

i have a loop (integer) that goes through a list of files, reading
data and computing stuff. i added a 'print,loop' line to the code, and
could see the routine would get stuck in one specific file, staying
there forever. thus i opened this file in IDL prompt and went through
all the steps of the routine one by one. doing the things via prompt,
i was getting an underflow message when reaching a line

pdf = exp(-0.5*chi)

because for the data inside that (only) file, the 1d40 factor i had
applied before was too large and chi ended up of the order of 1e4, and
then pdf would end up equal to 0 + underflow.

but, why running the routine automatically the code would get stuck
there, instead of giving the underflow message and moving on, i don't
know. i can guess the zero-ed array ended up producing a NaN array
somewhere later in the routine that messed the things up. to be honest
i haven't gone further as i stopped to improve the code, so that i
will test/treat the data on-the-fly and avoid the underflow in the
first place :).

anyway, i had to change a couple of 'e' to 'd' in the code indeed, as
you pointed out. as well as creating a double-precision array to read
the data from the asc files (it was float before and some numbers were
being lost without me realising it).

paula

On Feb 27, 6:39 pm, Chris <beaum...@ifa.hawaii.edu> wrote:
> The overflow, by itself, shouldn't result in an infinite process - idl
> should just complain about floating point overflow maybe.
>
> What COULD be happening is that you have a loop in your code. You are
> using a floating point value as the loop variable, and incrementing it
> by a number much smaller than itself. When that happens, there isn't
> enough precision to store the difference between the numbers before
> and after increment, and the loop variable stays the same forever.
> It's actually a floating point UNDERFLOW.
>
> Maybe? any floating point loops?
>
> chris

Subject: Re: 1e38 limit?
Posted by R.Bauer on Sat, 28 Feb 2009 10:49:17 GMT
View Forum Message <> Reply to Message

Paula schrieb:
> i think i found it, but i'm not sure how to explain...
>
> i have a loop (integer) that goes through a list of files, reading
> data and computing stuff. i added a 'print,loop' line to the code, and
> could see the routine would get stuck in one specific file, staying
> there forever. thus i opened this file in IDL prompt and went through
> all the steps of the routine one by one. doing the things via prompt,
> i was getting an underflow message when reaching a line
>
> pdf = exp(-0.5*chi)
>
> because for the data inside that (only) file, the 1d40 factor i had
> applied before was too large and chi ended up of the order of 1e4, and
> then pdf would end up equal to 0 + underflow.
>
> but, why running the routine automatically the code would get stuck
> there, instead of giving the underflow message and moving on,

because you haven't implemented an error handler.

look also on compile_opt
There is one to force the default type to become double.

cheers
Reimar
i don't
> know. i can guess the zero-ed array ended up producing a NaN array
> somewhere later in the routine that messed the things up. to be honest
> i haven't gone further as i stopped to improve the code, so that i
> will test/treat the data on-the-fly and avoid the underflow in the
> first place :).
>
>
> anyway, i had to change a couple of 'e' to 'd' in the code indeed, as
> you pointed out. as well as creating a double-precision array to read
> the data from the asc files (it was float before and some numbers were
> being lost without me realising it).
>
> paula
>
> On Feb 27, 6:39 pm, Chris <beaum...@ifa.hawaii.edu> wrote:
>> The overflow, by itself, shouldn't result in an infinite process - idl
>> should just complain about floating point overflow maybe.
>>

>> What COULD be happening is that you have a loop in your code. You are
>> using a floating point value as the loop variable, and incrementing it
>> by a number much smaller than itself. When that happens, there isn't
>> enough precision to store the difference between the numbers before
>> and after increment, and the loop variable stays the same forever.
>> It's actually a floating point UNDERFLOW.
>>
>> Maybe? any floating point loops?
>>
>> chris
>
