Subject: Re: strange behaviour of bytscl by large arrays
Posted by Lajos Foldy on Thu, 26 Apr 2012 16:51:20 GMT
View Forum Message <> Reply to Message

On Thursday, April 26, 2012 6:00:23 PM UTC+2, alx wrote:
> On 26 avr, 16:59, fawltylangu...@gmail.com wrote:
>> I think this is not a precision issue. Float can represent numbers up to 10^38 with a relative error of 10^-7. For huge values FINDGEN() creates indices with much bigger errors and this is the consequence of the current implementation, not the nature of floating point representation.
>>
>
> This is a precision issue, not *relative* but *absolute*. For n
> expressed as a floating point number and larger than its precision
> inverse, n+1 is no longer discernible from n. As you can see:
>
> IDL> print,float(10L^8+indgen(10)),FORMAT='(10Z8)'
> 5F5E100 5F5E100 5F5E100 5F5E100 5F5E100 5F5E108 5F5E108 5F5E108
> 5F5E108 5F5E108
> IDL> print,double(10L^8+indgen(10)),FORMAT='(10Z8)'
> 5F5E100 5F5E101 5F5E102 5F5E103 5F5E104 5F5E105 5F5E106 5F5E107
> 5F5E108 5F5E109
>
> Here 10L^8 is larger than 2/(machar()).eps = 16777216, and smaller
> than 2/(machar(/DOUBLE)).eps) (about 9e15).
> Creating a floating point ramp beyond 16777216 is formally possible,
> but is no sense since distinct values will be more and more spaced.
>
> alx.

I know all that. But when I write float(10l^8) I expect a floating point number with a relative error of 10^-7, a number in the range [10l^8-10, 10l^8+10]. FINDGEN()'s value is not in this range, it is far from it.

But as others wrote, the real solution is to mark FINDGEN in the docs as undefined/unsuitable for values greater than 16777216. Probably FINDGEN should print a warning, too.

regards,
Lajos