
Subject: Re: Plot bug or another "strange feature" ?
Posted by [John-David T. Smith](#) on Tue, 16 May 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Nicolas Decoster wrote:

>
> Paul van Delst wrote:
>>
>> Nicolas Decoster wrote:
>>>
>>> Hi.
>>>
>>> Does anybody notice and, eventually, know how to handle this "strange
>>> feature" of the plot procedure ?
>>>
>>> IDL> s = indgen(1000) + 1000000000
>>> IDL> plot, s, /ynozero
>>>
>>> The plot is not a clean line joining down-left and up-right corners, but
>>> an ugly stairway...
>>
>> I thought that sort of thing came about because:
>>
>> a) PLOT converts all its arguments to single precision floats. Do a
>> PRINT, FLOAT(s[0:100]),FORMAT='(e20.13)' to see what happens to the
>> float'd integers,
>>
>> b) people think they can represent numbers at the extremes of machine
>> precision exactly. Your "s" array, when converted to single precision
>> float, changes in the 8th or 9th d.p. While I think that IDL should
>> allow users to set the PLOT conversion to double precision if they want
>> (e.g. with a DOUBLE keyword or something), in general you can't expect
>> these sorts of numbers to be represented well - in IDL or any language.
>> That's not how floating point arithmetic works. All floating point
>> numbers are approximations to their actual value.
>
> I mainly agree with you. But "s" array is not a double array, it is a
> long array:
>
> IDL> help, s
> S LONG = Array[1000]
>
> I agree with you that plot converts everything to floating point number.
> But I think that it is important to see exactly what are the values of
> my integers (in fact my long integers) in a graph. And it is the same
> with double precision floating point numbers: I am not talking about the
> representation of real world numbers using floating point numbers.
> Example:

```
>
> IDL> b = [100000000.00000001d, 100000000.00000003d]
> IDL> print, b, format = '(e22.16)'
> 1.00000000000000001e+08
> 1.00000000000000003e+08
> IDL> plot, b, /ynozero
> % PLOT: Data range for axis has zero length.
> % Execution halted at: $MAIN$
>
> Yes, perhaps a keyword would be fine, here. Or better, a plot procedure
> that computes the data range according to the precision of the data.
```

Such a plot routine could easily be written by a motivated user. Consider that in order to generate drawing commands for typical display and hardcopy devices, IDL must convert your data to floating point values at some point in the process. You may disagree with where and how they perform this conversion, and therefore can write a general purpose plot wrapper which auto-detects the type of data, scales or offsets it into a range well matched to the internal plot's floating precision, and then undoes the scaling in the labelling of the tick values. This linear scaling, which could be saved, e.g., in a system variable, could then be applied to all subsequent suitably overridden plotting routines.

It seems as if your argument is equivalent to complaining that your image value which has data ranging from 1.e-12 to 2.e-12 doesn't display correctly with a straight "tv" command. RSI *could* have developed some overly-generalized notions of when and how to scale our data, which would work sometimes and sometimes not, and when it didn't, it would take heavy magic and carnal knowledge to get what you want. Thankfully they didn't.

JD

--

J.D. Smith |*| WORK: (607) 255-5842
Cornell University Dept. of Astronomy |*| (607) 255-6263
304 Space Sciences Bldg. |*| FAX: (607) 255-5875
Ithaca, NY 14853 |*|
