
Subject: Re: Slow object graphics when plotting multiple lines
Posted by [lecacheux.alain](#) on Mon, 04 Apr 2016 15:54:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Le lundi 4 avril 2016 17:09:36 UTC+2, Paul van Delst a écrit :

> Hello,
>
> On 04/04/16 10:48, Yngvar Larsen wrote:
>> Another trick, using a single array with lines separated by NaN.
>> This works both in direct and function graphics. With these two added, plus
>> the new CONNECTIVITY trick, your TEST routine prints this on my machine
>> (2012 iMac)
>
> I'm amazed at everyone's ability to discover these things! Wow.
>
> But, I think this need for various "tricks" to speed up basic plotting
> functionality is something that the IDL/Harris folks need to fix by
> making PLOT(/OVERPLOT) - with additional keywords if necessary - work
> similarly, i.e. fast.
>
> Similarly when plotting gobs of satellite data on maps, surfaces,
> contours, etc.
>
> The slowness of function graphics when plotting lots and lots of data
> makes it unusable for looking at large datasets.
>
> Other, similar, tools don't have these issues. (Yes, I mean matlab).
>
> cheers,
>
> paulv

As far as I understand, the function PLOT(/OVERPLOT) was NOT designed as being a simple overplotting routine. When you add a new curve, the entire plot is actually modified (ranges, axes, etc...) at the expense of some slowness. I guess that it is a feature from Exelis.

If you want to do a "simple" overplotting (ala OPLOT), an efficient way is to combine POSITION, CURRENT and [XYZ]RANGE keywords, instead of using OVERPLOT.

The POLYLINE trick, when many curves are needed, looks like to me somewhat faster than Matlab or Python equivalents.

alx.

alx.
