Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Lajos Foldy on Thu, 02 May 2013 17:01:56 GMT
View Forum Message <> Reply to Message

Hi David,

On Thursday, May 2, 2013 4:53:44 PM UTC+2, David Fanning wrote:
> David Fanning writes:
>
>
>
>> P.S. Another way to speed this up would be to create your own user
>
>> symbol (circle) and specify PSYM=8 in the call. That way you would avoid
>
>> going into cgSymbol each time and *remaking* the circle there. That is
>
>> probably what is taking much of the time.
>
>
>
> This is interesting. Reducing the number of points in my circles from 72
>
> to 36, made about 1 second of difference. Creating my own circle and
>
> passing PSYM=8 made about another second of difference. So, finally, I
>
> tried to get as close to the machine as possible with this code:
>
>
>
> n = 793647L
>
> lon = RANDOMU(seed,n)*360
>
> lat = RANDOMU(seed,n)*180
>
> cgLoadCT, 39
>
> cgDisplay
>
> circle = cgSymCat(3)
>
> black = cgColor('black')
>
> white = cgColor('white')
>
> tic

>
> PLOT, lon,lat, PSYM=circle, SYMSIZE=0.2, color=black, background=white
>
> toc
>
> END
>
>
>
> This took 19 seconds, compared to my original 25 seconds in a fully
>
> Coyote Graphics format. This seems to correspond pretty well to my rule
>
> of thumb that using Coyote Graphics routines adds about 20% rendering
>
> time to the normal direct graphics time. But, I'm a little bit surprised
>
> direct graphics is this slow here. Using a "dot" instead of a "circle"
>
> can be done in about 8.5 seconds. So, circle rendering seems to be a
>
> slower operation.
>

I tried this in Linux, it took 3.4 seconds through a remote X11 connection (3.2 seconds locally on a much slower machine). Maybe Windows is the limiting factor here :-)

regards,
Lajos