Subject: How to display single orbits of satellite data in function graphics?

Posted by Paul Van Delst[1] on Mon, 29 Apr 2013 23:26:44 GMT

View Forum Message <> Reply to Message

Hello,

The subject line initially read "Function graphics equivalent of PLOTS?"
but I changed it to what I really want to do.

I have an older direct graphics procedure that plots individual data
points (satellite data) on a map, where the colour of each distinct
field-of-view (FOV) is a function of the measured quantity (say,
radiance or temperature).

This is achieved by creating the global map, then looping over each
observation and plotting it on the map via PLOTS setting the colour
separately as needed for each plot. Takes about 0.5 seconds to display a
couple of orbits of data.

Standard sort of stuff IDL is used for, right?

For grins I thought I'd alter the code to do it using function graphics.
But, how does one do that? There's no equivalent of PLOTS. And besides,
plotting one point at a time in function graphics (when you have more
than a couple hundred points) takes forever (15minutes and counting
right now, for pete's sake).

To reiterate my question: How would one plot satellite tracks of
individual FOV data on a global map? E.g. a single orbit of polar
orbiter data?

It used to be a trivial thing to do in direct graphics. And the IDL help
is useless unless you want to register a nice regular image with a map
projection.

cheers,

paulv

p.s. I'm still at IDL v8.2 and I'm getting really really tired of
waiting many minutes for plots to display (that take fractions of a
second in DG). I'm hoping the latest versions of IDL have sped up
function graphics display by at least several orders of magnitude. Is
that the case?

Subject: Re: How to display single orbits of satellite data in function graphics?

Posted by David Fanning on Wed, 01 May 2013 14:49:36 GMT
View Forum Message <> Reply to Message

Paul van Delst writes:

> Sure. Once I figure out how to use COLORBAR... :o)

RTFM. ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to display single orbits of satellite data in function graphics?

Posted by David Fanning on Wed, 01 May 2013 14:52:59 GMT
View Forum Message <> Reply to Message

David Fanning writes:

>> Sure. Once I figure out how to use COLORBAR... :o)
>
> RTFM. ;-)

Oh, wait. Is that written in English? :-(

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to display single orbits of satellite data in function graphics?

Posted by Paul Van Delst[1] on Wed, 01 May 2013 16:10:28 GMT
View Forum Message <> Reply to Message

Yep - got it sorted. Wasn't too bad actually because there are a
multitude of examples in the documentation.

Still, the fact that the FG map display takes 16s to display (~750000
points) and the DG equivalent takes ~1s means that Function Graphics
can't really be used for investigating data. Makes beautiful plots for
display in presentations and articles, but not so great for day-to-day
usage.

Visitors to my cubicle all had a bit of a giggle while they were waiting
for the map to display "Why is it taking so long?"

Then, when the inevitable "can we zoom in on this-or-that region"
request came, the eyerolls began during the wait. Everyone has better
things to do than stare at the back of my head.

Is this FG speed problem fixed in a later IDL release? I haven't
bothered to go beyond v8.2 mostly because I've been using IDL less due
to this issue. The prospect of using Grads (no offense to the grads
folks! :o) is what scared me back to mapping/displaying data in IDL.

On 05/01/13 10:49, David Fanning wrote:
> Paul van Delst writes:
>
>> Sure. Once I figure out how to use COLORBAR... :o)
>
> RTFM. ;-)
>
> Cheers,
>
> David
>
>

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by David Fanning on Wed, 01 May 2013 16:16:59 GMT
View Forum Message <> Reply to Message

Paul van Delst writes:

> Is this FG speed problem fixed in a later IDL release?

My sense is that it has been improved (possibly) in a still forthcoming
release. :-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by chris_torrence@NOSPAM on Wed, 01 May 2013 21:58:42 GMT
View Forum Message <> Reply to Message

Hi all,

Well, there is both good news and bad news. The new graphics are indeed faster in IDL 8.2.x, but for your particular problem, it doesn't make much difference. The real bottleneck is just object graphics and OpenGL. Here is a reproduce case which compares both pure object graphics and new graphics:

```
n = 793647
lon = RANDOMU(seed,n)*360
lat = RANDOMU(seed,n)*180
colour = BYTSCL(RANDOMU(seed,n))
oSym = IDLgrSymbol(24, /FILLED)
oPal = IDLgrPalette()
oPal->LoadCT, 39
tic
oPlot = IDLgrPlot(lon, lat, LINESTYLE=6, $
  PALETTE=oPal, SYMBOL=oSym, VERT_COLORS=colour)
oModel = IDLgrModel()
oModel->Add, oPlot
oView = IDLgrView(VIEWPLANE_RECT=[0,0,360,180])
oView->Add, oModel
oWin = IDLgrWindow(GRAPHICS_TREE=oView)
oWin->Draw
toc

tic
p = PLOT(lon,lat,$
  SYMBOL='circle', $
  /SYM_FILLED, $
  SYM_SIZE=0.2, $
  RGB_TABLE=39, $
  VERT_COLORS=colour, $
  LINESTYLE=6)
toc
```

On my Win7 laptop, running 64-bit IDL, with hardware rendering, this takes 10.5 seconds for object graphics, and 24.0 seconds for new graphics.

So the new graphics is only off by a factor of 2 from pure object graphics. Now we can certainly try to chip away at that difference in future releases, but we're only going to be able to get it down to 10 seconds without lifting the hood on object graphics.

I think the main problem is that there are 790,000 points, each of which is a filled circle which has 25 vertices.

Fundamentally, it comes down to the difference between direct graphics, where you are just "burning" pixels into the screen, versus object graphics, where you are maintaining an object model in both memory and in the graphics card. One is fast, the other can be modified later.

Thoughts?

-Chris
ExelisVIS

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by David Fanning on Wed, 01 May 2013 22:05:54 GMT
View Forum Message <> Reply to Message

Chris Torrence writes:

> Fundamentally, it comes down to the difference between direct graphics, where you are just "burning" pixels into the screen, versus object graphics, where you are maintaining an object model in both memory and in the graphics card. One is fast, the other can be modified later.
>
> Thoughts?

Coyote Graphics? ;-)

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/

Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by chris_torrence@NOSPAM on Thu, 02 May 2013 03:33:09 GMT
View Forum Message <> Reply to Message

On Wednesday, May 1, 2013 4:05:54 PM UTC-6, David Fanning wrote:
>
>
> Coyote Graphics? ;-)
>
>

What is this "Coyote Graphics" that you speak of? :-)

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Fabzi on Thu, 02 May 2013 07:45:54 GMT
View Forum Message <> Reply to Message

On 05/02/2013 12:05 AM, David Fanning wrote:
> Chris Torrence writes:
>
>> Fundamentally, it comes down to the difference between direct graphics,
where you are just "burning" pixels into the screen, versus object
graphics,
where you are maintaining an object model in both memory and in the
graphics
card. One is fast, the other can be modified later.
>> Thoughts?
>
> Coyote Graphics? ;-)
>
> Cheers,
>
> David

Hi David,

CG will be indeed very fast to produce a DG output or a PS output but
then the bottleneck will be imagemagick which has to sort out all those
vertices to build a PNG out of it. But It don't think it will need 20
seconds...

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by David Fanning on Thu, 02 May 2013 12:09:09 GMT
View Forum Message <> Reply to Message

Fabien writes:

> CG will be indeed very fast to produce a DG output or a PS output but
> then the bottleneck will be imagemagick which has to sort out all those
> vertices to build a PNG out of it. But It don't think it will need 20
> seconds...

Yes, Coyote Graphics will not produce much of a Wow! factor when you have a handful of people crowded into your cubicle, looking over your shoulder at screen output. Their claim to fame (if the have one!) is that they are blazingly fast to render, extremely flexible in what they can do, and extraordinarily easy for most people to program. Plus, if you have a couple of seconds, even the most complicated output can be turned into PostScript, raster, and PDF output that is equal in quality to anything IDL can produce.

http://www.idlcoyote.com/cg_tips/cgwfg.php

I'm not saying I don't wish they were more beautiful on the screen. I'm saying that EXCEPT for when there are five people huddled in my cubicle, looking over my shoulder, they get the job done in a way that never makes me think "I should go get some coffee while I'm waiting."

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Fabzi on Thu, 02 May 2013 14:07:21 GMT
View Forum Message <> Reply to Message

On 05/02/2013 02:09 PM, David Fanning wrote:
> Coyote Graphics will not produce much of a Wow! factor

Hi David,

my point was not to criticize CG (I am a big fan ;), but rather to agree with Chris that plotting 790,000 filled points on the screen cannot be fast.

Actually, following code makes my machine crash:

```
n = 793647
lon = RANDOMU(seed,n)*360
lat = RANDOMU(seed,n)*180
tic
cgLoadCT, 39
cgWindow
cgPLOT, lon,lat, PSYM=16, SYMSIZE=0.2, /WINDOW
toc
```

while this:

```
n = 793647
lon = RANDOMU(seed,n)*360
lat = RANDOMU(seed,n)*180
tic
cgLoadCT, 39
cgPLOT, lon,lat, PSYM=16, SYMSIZE=0.2, OUTPUT='t.png'
toc
```

needs 66 secs to compute...

In all cases, plotting 790,000 filled points doesn't make much sense to me, but sometimes one cannot decide how many points are going to be plotted.

I have the same issue with a scatterplot I am making for a publication (~50000 points). If I save this as an eps the plot is 6Mb large!!! As a pdf it is 2.3 Mb but still too large for a publication (besides, it takes too long to render in the PDF reader). So I could decide to plot only 10% of the points. To be honest, the information on the plot doesn't get lost and is not falsified, but still for a publication you'd rather want to show "all" the data... Second option is going for PNG, but in publications I think that vector format should be the norm...

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by David Fanning on Thu, 02 May 2013 14:25:58 GMT
View Forum Message <> Reply to Message

Fabien writes:

> my point was not to criticize CG (I am a big fan ;), but rather to agree
> with Chris that plotting 790,000 filled points on the screen cannot be fast.

I know you are a fan. I wasn't writing for you, necessarily. ;-)

> Actually, following code makes my machine crash:
>
> n = 793647
> lon = RANDOMU(seed,n)*360
> lat = RANDOMU(seed,n)*180
> tic
> cgLoadCT, 39
> cgWindow
> cgPLOT, lon,lat, PSYM=16, SYMSIZE=0.2, /WINDOW
> toc

Well, it take 25 seconds on my machine. I use 72 points for the circles, though, which is probably overkill. My ears perked up when Chris mentioned 25 points the other day. And, of course, at this size, you probably couldn't tell any difference if I just made the damn things triangles! :-)

> while this:
>
> n = 793647
> lon = RANDOMU(seed,n)*360
> lat = RANDOMU(seed,n)*180
> tic
> cgLoadCT, 39
> cgPLOT, lon,lat, PSYM=16, SYMSIZE=0.2, OUTPUT='t.png'
> toc
>
> needs 66 secs to compute...

Yep. Of course, a black rectangle would have rendered faster, with the same result. :-)

> In all cases, plotting 790,000 filled points doesn't make much senseto
> me, but sometimes one cannot decide how many points are going to be
> plotted.
>
> I have the same issue with a scatterplot I am making for a publication
> (~50000 points). If I save this as an eps the plot is 6Mb large!!! As a
> pdf it is 2.3 Mb but still too large for a publication (besides, it
> takes too long to render in the PDF reader). So I could decide to plot
> only 10% of the points. To be honest, the information on the plot
> doesn't get lost and is not falsified, but still for a publication you'd
> rather want to show "all" the data... Second option is going for PNG,
> but in publications I think that vector format should be the norm...

I used to think so, but in the book publishing business the printers prefer high resolution PNG or TIFF raster files it seems to me. I like to use them because I can actually see them in my book building software.

Cheers,

David

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.

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by David Fanning on Thu, 02 May 2013 14:53:44 GMT

View Forum Message <> Reply to Message

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.

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Paul Van Delst[1] on Thu, 02 May 2013 16:41:54 GMT

View Forum Message <> Reply to Message

Hello,

On 05/01/13 17:58, Chris Torrence wrote:
> Hi all,
>
> Well, there is both good news and bad news. The new graphics are
> indeed faster in IDL 8.2.x, but for your particular problem, it
> doesn't make much difference. The real bottleneck is just object
> graphics and OpenGL. Here is a reproduce case which compares both
> pure object graphics and new graphics:

[example snipped]

> On my Win7 laptop, running 64-bit IDL, with hardware rendering, this
> takes 10.5 seconds for object graphics, and 24.0 seconds for new
> graphics.
>
> So the new graphics is only off by a factor of 2 from pure object
> graphics. Now we can certainly try to chip away at that difference in
> future releases, but we're only going to be able to get it down to 10
> seconds without lifting the hood on object graphics.
>
> I think the main problem is that there are 790,000 points, each of
> which is a filled circle which has 25 vertices.
>

> Fundamentally, it comes down to the difference between direct
> graphics, where you are just "burning" pixels into the screen, versus
> object graphics, where you are maintaining an object model in both
> memory and in the graphics card. One is fast, the other can be
> modified later.
>
> Thoughts?

Well, my initial thought is that, by definition I guess, New Graphics cannot be used for on-the-fly investigations into largish datasets.

I mean... that's really the point here, right?

I just spent about 30minutes in a colleague's cubicle while he plotted and mapped -- using direct graphics routines -- said 750K-1M points of satellite data (two sets - one for the operational result, another for experimental results) in several different ways, many times, changing ranges, data quantities, etc. We learned a lot about the data in that 30minutes. The guy couldn't type fast enough to keep up with the requests to plot the data versus wind speed, or water temperature, or land coverage fraction, or <insert quantity of choice>.

That is an impossible task for New (or Object) graphics. Last night I and another colleague did something similar and producing several maps of data took minutes (and I was doing it locally. My DG colleague was using IDL on a computer several hundred miles away!) That 10seconds seems like an eternity when you do it again and again and again...

For the data we were looking at, we could plot/map different stuff in DG several times over (limited more by person typing speed) in the time it took to create one NG plot/map.

So, basically, for largish datasets, the interesting task of actually studying the data (the main objective, no?) MUST be done using direct graphics. The decidedly uninteresting task of creating a plot suitable for publication can be done with New graphics (but doesn't have to be).

The fact that I have 1M points plotted with circles that have 25 vertices within an object model are implementation details I am not really interested in (that's why we pay $$$$ for IDL). Besides, the resulting "object model" is useless for actually manipulating the contained objects in real time due to the slow refresh/rendering when changes are made.

Nowadays satellite/weather/climate datasets are measured in terabytes. If IDL NG chokes on 1M points (which I do not consider a large number) then I would definitely suggest "lifting the hood" on object/New graphics.

cheers,

paulv

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

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by David Fanning on Thu, 02 May 2013 17:06:07 GMT

View Forum Message <> Reply to Message

fawltylanguage@gmail.com writes:

> 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 :-)

Sigh... And I was thinking only my laptop needed to be replaced. :-(

Cheers,

David
--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.idlcoyote.com/
Sepore ma de ni thue. ("Perhaps thou speakest truth.")

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Phillip Bitzer on Thu, 02 May 2013 19:26:12 GMT

View Forum Message <> Reply to Message

Just for kicks and giggles, here's another data point:

```
IDL> print, !VERSION
{ x86_64 darwin unix Mac OS X 8.2.2 Jan 23 2013      64      64}

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

% Time elapsed: 1.0270369 seconds.

n = 793647
lon = RANDOMU(seed,n)*360
lat = RANDOMU(seed,n)*180
tic
cgLoadCT, 39
cgWINDOW
circle = cgSymCat(3)
black = cgColor('black')
white = cgColor('white')
cgPLOT, lon,lat, PSYM=circle, SYMSIZE=0.2, /ADD, color=black, background=white
cgcontrol, output='test.png'
toc

% Time elapsed: 15.803132 seconds.
```

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by DavidF[1] on Thu, 02 May 2013 19:35:26 GMT
View Forum Message <> Reply to Message

OK, but (my fault, I copied the code after my tests, rather than during) circle = cgSymCat(3) is a dot. What are these numbers if you replace the 3 with 16?

Cheers,

David

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Lajos Foldy on Thu, 02 May 2013 20:06:08 GMT
View Forum Message <> Reply to Message

Hi David,

On Thursday, May 2, 2013 9:35:26 PM UTC+2, Coyote wrote:

> OK, but (my fault, I copied the code after my tests, rather than during) circle = cgSymCat(3) is a dot. What are these numbers if you replace the 3 with 16?

Linux, remote X11: 9.0 seconds. Still better than your Windows :-)

regards,
Lajos

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Phillip Bitzer on Thu, 02 May 2013 20:31:04 GMT
View Forum Message <> Reply to Message

>> OK, but (my fault, I copied the code after my tests, rather than during) circle = cgSymCat(3) is a dot. What are these numbers if you replace the 3 with 16?
>

> Linux, remote X11: 9.0 seconds. Still better than your Windows :-)
>

To screen:
% Time elapsed: 11.061576 seconds. (now larger than the Linux remote case from Lajos)

To cgWindow->PNG (using ImageMagick)
% Time elapsed: 71.683804 seconds.

For the "->PNG" case, I've noticed that the plot is "redrawn" in the cgWindow (i.e., it seems to be plotted twice).

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Jim Pendleton on Fri, 03 May 2013 02:39:10 GMT
View Forum Message <> Reply to Message

On Monday, April 29, 2013 5:26:44 PM UTC-6, Paul van Delst wrote:
> Hello,
>
>
>
> The subject line initially read "Function graphics equivalent of PLOTS?"
>
> but I changed it to what I really want to do.
>
>
>
> I have an older direct graphics procedure that plots individual data
>
> points (satellite data) on a map, where the colour of each distinct
>
> field-of-view (FOV) is a function of the measured quantity (say,
>
> radiance or temperature).
>
>
>
> This is achieved by creating the global map, then looping over each
>
> observation and plotting it on the map via PLOTS setting the colour
>
> separately as needed for each plot. Takes about 0.5 seconds to display a
>
> couple of orbits of data.
>
>
>
> Standard sort of stuff IDL is used for, right?
>
>
>
> For grins I thought I'd alter the code to do it using function graphics.
>
> But, how does one do that? There's no equivalent of PLOTS. And besides,
>
> plotting one point at a time in function graphics (when you have more
>
> than a couple hundred points) takes forever (15minutes and counting
>
> right now, for pete's sake).
>

>
>
> To reiterate my question: How would one plot satellite tracks of
>
> individual FOV data on a global map? E.g. a single orbit of polar
>
> orbiter data?
>
>
>
> It used to be a trivial thing to do in direct graphics. And the IDL help
>
> is useless unless you want to register a nice regular image with a map
>
> projection.
>
>
>
> cheers,
>
>
>
> paulv
>
>
>
> p.s. I'm still at IDL v8.2 and I'm getting really really tired of
>
> waiting many minutes for plots to display (that take fractions of a
>
> second in DG). I'm hoping the latest versions of IDL have sped up
>
> function graphics display by at least several orders of magnitude. Is
>
> that the case?

With respect to Chris' example, the data is greatly oversampled relative to the display pixels available in the window so large blobs are not really appropriate. Try simply changing the IDLgrSymbol to create a single point via "oSym = IDLgrSymbol(3)", which is more like DG PLOTS with PSYM = 3. That'll reduce both the execution time and memory use significantly.