Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Paul Van Delst[1] on Tue, 30 Apr 2013 16:24:53 GMT
View Forum Message <> Reply to Message

Just an update: left my brain-dead function graphics translation of the direct graphics program running overnight.... the plot was probably only 10% complete when I hit ^C. That's quite funny.

Back to DG I guess...

On 04/29/13 19:26, 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?

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by David Fanning on Tue, 30 Apr 2013 16:34:49 GMT
View Forum Message <> Reply to Message

Paul van Delst writes:

> Just an update: left my brain-dead function graphics translation of the
> direct graphics program running overnight.... the plot was probably only
> 10% complete when I hit ^C. That's quite funny.

So, the "couple orders of magnitude" improvement you are seeking will
bring that down to under 10 hours, at least. That's certainly doable, it
seems to me. But, I would wait for 8.2.3 before upgrading. :-)

I'm curious to see some code. Function graphics have been slow, but they
are not totally brain dead, usually. I wonder if something else is going
on here?

Unfortunately, I don't have the faintest idea how to even get started
replicating the problem. Can you suggest a test case?

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 Tue, 30 Apr 2013 16:50:03 GMT
View Forum Message <> Reply to Message

Hello,

On 04/30/13 12:34, David Fanning wrote:
> Paul van Delst writes:

>
>> Just an update: left my brain-dead function graphics translation of the
>> direct graphics program running overnight.... the plot was probably only
>> 10% complete when I hit ^C. That's quite funny.
>
> So, the "couple orders of magnitude" improvement you are seeking will
> bring that down to under 10 hours, at least. That's certainly doable, it
> seems to me. But, I would wait for 8.2.3 before upgrading. :-)
>
> I'm curious to see some code. Function graphics have been slow, but they
> are not totally brain dead, usually. I wonder if something else is going
> on here?

Here's the DG code plotting the data "var"

```
-----%<-----
lat=reform(meta.cenlat)
lon=reform(meta.cenlon)

  map_set,limit=[latmin,lonmin,latmax,lonmax],$
         title=title1
  map_continents

  colour=bytscl(var,min=minval,max=maxval)

  for i=0L,nobs-1 do begin
    plots,lon(i),lat(i),color=colour(i),$
         psym=8,symsize=symsize
  endfor
-----%<-----
```

And here is my practically line-by-line translation of the above using FG. note that I am using "PLOT()" to approximate PLOTS by simply overplotting the same point.

```
-----%<-----
   map = MAP('Mercator', $
            LIMIT=[minlat,minlon,maxlat,maxlon], $
            RGB_TABLE=3, $
            CURRENT=w)

   ; Change some grid properties.
   grid = map.MAPGRID
   grid.LINESTYLE = "dotted"
   grid.LABEL_POSITION = 0
   grid.BOX_AXES = 1
```

```
  ; Display continetnal outlines
  m1 = MAPCONTINENTS()


  lat = REFORM(metadata.cenlat)
  lon = REFORM(metadata.cenlon)

  colour = BYTSCL(variable,MIN=minval,MAX=maxval)

  map.Refresh, /Disable
  FOR i = 0L, N_ELEMENTS(variable)-1L DO BEGIN
    p = PLOT([lon[i],lon[i]],[lat[i],lat[i]],$
            SYMBOL='circle',SYM_COLOR=colour[i],$
            /SYM_FILLED,SYM_FILL_COLOR=colour[i],$
            VERT_COLORS=[colour[i],colour[i]], $
            LINESTYLE=6, $
            /OVERPLOT)
  ENDFOR
  map.Refresh
-----%<-----
```

I realise the symbol colouring is completely wrong but the above FG code was written as the initial test case. First plot the data points, then colour them accordingly.

The data vectors (variable, lat, lon) contains 131085 data points. Not that big.

I also realise my use of PLOT() above is completely wrong/bad/stupid (creating millions of FG objects no doubt) - hence my request about how to plot individual points, in this case satellite FOV measurements, on a map. The actual data is thinned, cloud-cleared, and quality controlled so creating a regular image is not an option.

There's obviously a much smarter way of doing this - before I started data mining the IDL documentation (which, with regards to examples of using the FG mapping commands that *don't* involve an image, is quite poor), I figured I'd ask the newsgroup.

cheers,

paulv

>
> Unfortunately, I don't have the faintest idea how to even get started
> replicating the problem. Can you suggest a test case?
>

> Cheers,
>
> David
>
>
>

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by David Fanning on Tue, 30 Apr 2013 16:56:08 GMT
View Forum Message <> Reply to Message

Paul van Delst writes:

> And here is my practically line-by-line translation of the above using
> FG. note that I am using "PLOT()" to approximate PLOTS by simply
> overplotting the same point.

My first thought would have been to try to plot all points at once, with
a single Plot() command, without the loop. Did you try that?

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 Mark Piper on Tue, 30 Apr 2013 18:17:00 GMT
View Forum Message <> Reply to Message

On Tuesday, April 30, 2013 10:56:08 AM UTC-6, David Fanning wrote:
>
> My first thought would have been to try to plot all points at once, with
> a single Plot() command, without the loop. Did you try that?
>

As David suggests, vectorization is the IDL Way! Here's an example:

```
IDL> x = randomu(1, n)*360
IDL> y = randomu(2, n)*180 - 90
IDL> z = floor(randomu(3, n)*256)
IDL> tic
IDL> p = plot(x, y, rgb_table=27, vert_colors=z, linestyle='none', symbol='+')
IDL> toc
% Time elapsed: 0.54600000 seconds.
```

mp

## Subject: Re: How to display single orbits of satellite data in function graphics?
## Posted by Mark Piper on Tue, 30 Apr 2013 18:18:34 GMT

View Forum Message <> Reply to Message

```
On Tuesday, April 30, 2013 12:17:00 PM UTC-6, Mark Piper wrote:
>  On Tuesday, April 30, 2013 10:56:08 AM UTC-6, David Fanning wrote:
>
>>
>
>>  My first thought would have been to try to plot all points at once, with
>
>>  a single Plot() command, without the loop. Did you try that?
>
>>
>
>
>
>  As David suggests, vectorization is the IDL Way! Here's an example:
>
>
>
>  IDL> x = randomu(1, n)*360
>
>  IDL> y = randomu(2, n)*180 - 90
>
>  IDL> z = floor(randomu(3, n)*256)
>
>  IDL> tic
>
>  IDL> p = plot(x, y, rgb_table=27, vert_colors=z, linestyle='none', symbol='+')
>
>  IDL> toc
>
>  % Time elapsed: 0.54600000 seconds.
>
>
>
>  mp
```

Ooops, n = 1e3 in my example.

mp

### Subject: Re: How to display single orbits of satellite data in function graphics?
### Posted by Mark Piper on Tue, 30 Apr 2013 18:24:49 GMT

View Forum Message <> Reply to Message

On Tuesday, April 30, 2013 10:50:03 AM UTC-6, Paul van Delst wrote:

> There's obviously a much smarter way of doing this - before I started
> data mining the IDL documentation (which, with regards to examples of
> using the FG mapping commands that *don't* involve an image, is quite
> poor), I figured I'd ask the newsgroup.

Paul,

If we collaborate to get this example working, may we please include it in the IDL documentation?

mp

### Subject: Re: How to display single orbits of satellite data in function graphics?
### Posted by Dick Jackson on Tue, 30 Apr 2013 18:50:37 GMT

View Forum Message <> Reply to Message

Hi Paul,

A little late to the party, but I can add a couple of points.

1:

If you have the points all at once, then as David says, this would work fine
(and for your 131,085 points my aging MacBook Pro with IDL 8.2.2 did this Plot()
in 3.77 seconds)

```
;;    Placing all points in one plot (no joining lines)

p = PLOT(lon, lat,$
   SYMBOL='circle', $   ; SYM_COLOR=colour,$      ; not needed
   /SYM_FILLED, $       ; SYM_FILL_COLOR=colour,$ ; not needed
   RGB_TABLE=3, $
   VERT_COLORS=colour, $
   LINESTYLE=6, $
   /OVERPLOT)
```

But if you really need to add the points one at a time (because they're not available at the start), p.SetData is your friend, allowing you to assign a new, longer set of lon/lat values as you get new data. I'm simulating this situation here, and assuming that the colours aren't known beforehand either:

```
;;    Placing first two points in a plot (no joining lines)...

p = PLOT(lon[0:1], lat[0:1],$
   RGB_TABLE=3, $
   SYMBOL='circle', $ ;SYM_COLOR=colour[0:1],$
   /SYM_FILLED, $ ;SYM_FILL_COLOR=colour[0:1],$
   VERT_COLORS=colour[0:1], $
   LINESTYLE=6, $
   /OVERPLOT)

FOR i = 1L, N_ELEMENTS(variable)-1L DO BEGIN ; Each time we get new data
   map.Refresh, /Disable
   p.SetData, lon[0:i], lat[0:i]
   p.Vert_Colors = colour[0:i]
   map.Refresh
END
```

2:

I found an oddity in the behaviour with and without refreshing: how is it that when only performing *two* operations, having it refresh automatically takes *three* times as long as doing one refresh at the end? This is with the 131,085 points (note that I'm using the handy 8.2.2 routines TIC and TOC that just start and stop a timer... somebody may have liked my old TStart and TReport routines!):

```
IDL> map.Refresh & tic & p.setdata, lon, lat & p.Vert_Colors = colour & toc
% Time elapsed: 10.551133 seconds.

IDL> map.Refresh, /Disable & tic & p.setdata, lon, lat & p.Vert_Colors = colour
& map.Refresh & toc
% Time elapsed: 3.6593549 seconds.
```

Chris? Mark? Could there be an extra redraw happening when refresh is on?

Cheers,
-Dick

Dick Jackson Software Consulting
Victoria, BC, Canada --- +1-250-220-6117
dick@d-jackson.com --- http://www.d-jackson.com

David Fanning wrote:
> Paul van Delst writes:
>
>> And here is my practically line-by-line translation of the above using
>> FG. note that I am using "PLOT()" to approximate PLOTS by simply
>> overplotting the same point.
>
> My first thought would have been to try to plot all points at once, with
> a single Plot() command, without the loop. Did you try that?
>
> Cheers,
>
> David
>
>
>

--

Cheers,
-Dick

Dick Jackson Software Consulting
Victoria, BC, Canada
www.d-jackson.com

## Subject: Re: How to display single orbits of satellite data in function graphics?
## Posted by Paul Van Delst[1] on Tue, 30 Apr 2013 20:24:43 GMT

View Forum Message <> Reply to Message

First off, regarding the loop over points using PLOT(): D'oh!!
<headslap> I was caught in a brainfade loop due to trying to recreate
the DG code.

Based on David and Mark's (and Dick's!) suggestions, this is what I did,
but now using multiple orbits (793647 points):

```
-----%<----
  map.Refresh, /Disable

  t0 = SYSTIME(1)
  p = PLOT(lon,lat,$
         SYMBOL='circle', $
         /SYM_FILLED, $
         SYM_SIZE=0.2, $
         RGB_TABLE=39, $
```

```
        VERT_COLORS=colour, $
        LINESTYLE=6, $
        /OVERPLOT)
  t1 = SYSTIME(1)
  MESSAGE, 'Plot of '+STRTRIM(N_ELEMENTS(lon),2)+$
        ' datapoints took '+STRTRIM(t1-t0,2)+' seconds', $
        /INFORMATIONAL

  t0 = SYSTIME(1)
  map.Refresh
  t1 = SYSTIME(1)
  MESSAGE, 'Refresh of map display took '+$
        STRTRIM(t1-t0,2)+' seconds', $
        /INFORMATIONAL
-----%<----
```

and this is what the various MESSAGE output tells me:

```
IDL> display_gsi_map,nnobs.depar_bc[2],nnmeta
% DISPLAY_GSI_MAP: Plot of 793647 datapoints took 1.0648339 seconds
% DISPLAY_GSI_MAP: Refresh of map display took 15.363068 seconds
```

If I do the following, commenting out all the map.refresh stuff:

```
-----%<----
;  map.Refresh, /Disable

  t0 = SYSTIME(1)
  p = PLOT(lon,lat,$
        SYMBOL='circle', $
        /SYM_FILLED, $
        SYM_SIZE=0.2, $
        RGB_TABLE=39, $
        VERT_COLORS=colour, $
        LINESTYLE=6, $
        /OVERPLOT)
  t1 = SYSTIME(1)
  MESSAGE, 'Plot of '+STRTRIM(N_ELEMENTS(lon),2)+$
        ' datapoints took '+STRTRIM(t1-t0,2)+' seconds', $
        /INFORMATIONAL

;  t0 = SYSTIME(1)
;  map.Refresh
;  t1 = SYSTIME(1)
;  MESSAGE, 'Refresh of map display took '+$
;        STRTRIM(t1-t0,2)+' seconds', $
;        /INFORMATIONAL
```

-----%<----

I get the following:

```
IDL> display_gsi_map,nnobs.depar_bc[2],nnmeta
% DISPLAY_GSI_MAP: Plot of 793647 datapoints took 31.771896 seconds
```

So it takes twice as long if you just do a straight PLOT() without use
of the refresh method?

Why?

Additionally, if I switch to another virtual desktop (on my linux
system) when I switch back to the desktop that contains the map display,
it takes 15s (I timed it 3 times) for the graphic to redisplay (from
black) and for the IDL command line to become active again.

This may be more of a system question (e.g. load, memory, etc) but i'll
ask anyway: Why?

cheers,

paulv

p.s. Onto using COLORBAR! (cowering in corner in abject fear....)

On 04/30/13 14:17, Mark Piper wrote:
> On Tuesday, April 30, 2013 10:56:08 AM UTC-6, David Fanning wrote:
>>
>> My first thought would have been to try to plot all points at once, with
>> a single Plot() command, without the loop. Did you try that?
>>
>
> As David suggests, vectorization is the IDL Way! Here's an example:
>
> IDL> x = randomu(1, n)*360
> IDL> y = randomu(2, n)*180 - 90
> IDL> z = floor(randomu(3, n)*256)
> IDL> tic
> IDL> p = plot(x, y, rgb_table=27, vert_colors=z, linestyle='none', symbol='+')
> IDL> toc
> % Time elapsed: 0.54600000 seconds.
>
> mp
>

Subject: Re: How to display single orbits of satellite data in function graphics?
Posted by Jim  Pendleton on Tue, 30 Apr 2013 20:29:24 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?

This isn't a direct reply to the thread, but one of my favorite tools in the IDL Workbench for determining performance bottlenecks is the Profiler tool. In IDL 8.2 this can be instantiated through the Window menu.

It may not answer all the questions, but it can be used to direct the questions.

In the case of overplotting a single point at a time, as in this example, the methods _ACCUMULATEXYZRANGE and GETXYZRANGE in the undocumented class _IDLITVISUALIZATION use a lot of execution time. Without even knowing what these methods do, one could guess that each time a point is added, the plot range is calculated new. This might lead one to ask, "Is there a way to set the data range a single time, given that I know my map limits and the range of valid latitudes and longitudes I intend to plot?"

If there isn't a likely answer to such a question in the help for the documented routines, it's a good indicator that Dr. Piper might be interested in hearing your concerns.

## Subject: Re: How to display single orbits of satellite data in function graphics?
## Posted by Paul Van Delst[1] on Wed, 01 May 2013 14:48:17 GMT

View Forum Message <> Reply to Message

Hi Mark,

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

cheers,

paulv

On 04/30/13 14:24, Mark Piper wrote:
>  On Tuesday, April 30, 2013 10:50:03 AM UTC-6, Paul van Delst wrote:
>
>>  There's obviously a much smarter way of doing this - before I started
>>  data mining the IDL documentation (which, with regards to examples of
>>  using the FG mapping commands that *don't* involve an image, is quite
>>  poor), I figured I'd ask the newsgroup.
>
>  Paul,
>
>  If we collaborate to get this example working, may we please include it in the IDL documentation?
>
>  mp
>