
Subject: Function Graphics Questions

Posted by [David Fanning](#) on Tue, 14 Jan 2014 17:56:18 GMT

[View Forum Message](#) <> [Reply to Message](#)

Folks,

I have a New Year's resolution to learn more about function graphics this year. I thought I would start with this multiple axis plot Chris and Matt have been talking about this week. My idea was to produce mirror of the Coyote Gallery plots, as much as I can.

I guess I've spent about three hours on this now and finally have this one program in shape to be able to compare Coyote Graphics output with the equivalent function graphics output. In doing so, I've run into some questions. Perhaps someone knows the answers.

Question 1: I have no particular objection to the PostScript output produced by function graphics commands, but is it true there is no program control over things like the thickness of the PostScript lines?

Question 2: As far as I can tell, saving the contents of a function graphics window as a PostScript file **always** creates encapsulated PostScript files. Since encapsulated Postscript files (AFAIK) always have to be in portrait mode, what is the purpose of the LANDSCAPE keyword to the window save command?

In other words, this command:

```
window.save, 'test.ps', /Landscape
```

Produces exactly the same output, as far as I can tell, as this command:

```
window.save, 'test.ps'
```

Question 3: I haven't upgraded to IDL 8.3. Can someone tell me if the bug in IDL 8.2.3 that prevents any line style except solid in PostScript output is fixed. In other words, do these commands produce a Postscript plot with a dashed line:

```
p = Plot(/test, LineStyle=2)
p.save, 'test.ps'
```

Question 4: Am I missing something obvious here. I mostly produce JPEG, PNG, and TIFF output either for my web page or for e-mailing intermediate results to colleagues. I like them to be reasonably small. For my web page, for example, I like them to be no more than 600 pixels wide. My usual way of creating such raster output is to run my code like this:

```
cgPS_Open, 'test.png'  
cgPlot, cgDemoData(1)  
cgPS_Close, Width=600
```

The equivalent in function graphics is something like this:

```
p = Plot(cgDemoData(1))  
p.save, 'test.png', width=600
```

But, this kind of output is very low resolution compared to what I've come to expect.

I find the only way I can get high quality PNG files is to produce them at full resolution, then resize them in the software I use for dealing with raster images (Photoshop, Hypersnap, etc.). Since I have ImageMagick hanging around, I find I can get what I want in IDL by doing something like this:

```
p = Plot(cgDemoData(1))  
p.save, 'test.png'  
Spawn, 'convert test.png -resize 600 test_resized.png'
```

Is there a better way to do this?

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: Function Graphics Questions

Posted by [Paul Van Delst\[1\]](#) on Tue, 14 Jan 2014 21:23:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

On 01/14/14 12:56, David Fanning wrote:

> Folks,

>

> I have a New Year's resolution to learn more about function graphics
> this year. I thought I would start with this multiple axis plot Chris
> and Matt have been talking about this week. My idea was to produce
> mirror of the Coyote Gallery plots, as much as I can.

>

> I guess I've spent about three hours on this now and finally have this

> one program in shape to be able to compare Coyote Graphics output with
> the equivalent function graphics output. In doing so, I've run into some
> questions. Perhaps someone knows the answers.

>

> Question 1: I have no particular objection to the PostScript output
> produced by function graphics commands, but is it true there is no
> program control over things like the thickness of the PostScript lines?

?

You mean you don't see any difference when you use the THICK keyword?

I see a difference in the line thickness between ps output when I do:

```
p = Plot(/test, LineStyle=2)
p.save, 'test.ps'
```

and

```
p = Plot(/test, LineStyle=2,thick=2)
p.save, 'test.ps'
```

> Question 2: As far as I can tell, saving the contents of a function
> graphics window as a PostScript file *always* creates encapsulated
> PostScript files. Since encapsulated Postscript files (AFAIK) always
> have to be in portrait mode, what is the purpose of the LANDSCAPE
> keyword to the window save command?

>

> In other words, this command:

>

> window.save, 'test.ps', /Landscape

>

> Produces exactly the same output, as far as I can tell, as this command:

>

> window.save, 'test.ps'

Dunno. I always create png output or eps for including in documents so I
don't use /landscape.

> Question 3: I haven't upgraded to IDL 8.3. Can someone tell me if the
> bug in IDL 8.2.3 that prevents any line style except solid in PostScript
> output is fixed. In other words, do these commands produce a Postscript
> plot with a dashed line:

>

```
> p = Plot(/test, LineStyle=2)
> p.save, 'test.ps'
```

IDL v8.3 produces a dashed line in the ps output.

> Question 4: Am I missing something obvious here. I mostly produce JPEG,
> PNG, and TIFF output either for my web page or for e-mailing
> intermediate results to colleagues. I like them to be reasonably small.
> For my web page, for example, I like them to be no more than 600 pixels
> wide. My usual way of creating such raster output is to run my code like
> this:

```
>  
> cgPS_Open, 'test.png'  
> cgPlot, cgDemoData(1)  
> cgPS_Close, Width=600  
>
```

> The equivalent in function graphics is something like this:

```
>  
> p = Plot(cgDemoData(1))  
> p.save, 'test.png', width=600  
>
```

> But, this kind of output is very low resolution compared to what I've
> come to expect.

I don't know if it's the same thing (you're windows, right?), but I was experiencing a problem with v8.2.2/3 on a RHEL6 system where the png output of a plot had very blocky/pixel-y lines. Exelis help replicated the problem on their CentOS systems even with v8.3. The following workaround provided by Exelis made the lines smooth again:

```
$ export IDL_DISABLE_STROKED_LINES=1
```

The issue has been reported as IDL-69024.

cheers,

paulv

```
>  
> I find the only way I can get high quality PNG files is to produce them  
> at full resolution, then resize them in the software I use for dealing  
> with raster images (Photoshop, Hypersnap, etc.). Since I have  
> ImageMagick hanging around, I find I can get what I want in IDL by doing  
> something like this:
```

```
>  
> p = Plot(cgDemoData(1))  
> p.save, 'test.png'  
> Spawn, 'convert test.png -resize 600 test_resized.png'
```

>
> Is there a better way to do this?
>
> Cheers,
>
> David
>

Subject: Re: Function Graphics Questions
Posted by [David Fanning](#) on Tue, 14 Jan 2014 21:36:02 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> You mean you don't see any difference when you use the THICK keyword?
>
> I see a difference in the line thickness between ps output when I do:
>
> p = Plot(/test, LineStyle=2)
> p.save, 'test.ps'
>
> and
>
> p = Plot(/test, LineStyle=2,thick=2)
> p.save, 'test.ps'

No, I see a difference using the thick keyword. But, I am used to writing code like this to make a line proportionately as thick in the PostScript device.

```
thick = (!D.Name EQ 'PS') ? 6 : 2
```

The difference between PostScript "thickness" and display "thickness" appears to be taken care of automatically, so that a line that is twice as thick on the display is proportionately twice as thick in PostScript. But, as far as I can tell, there is no way to do something like this:

```
thick = (!D.Name EQ 'PS') ? 10 : 2
```

And make the line *extra* thick in PostScript.

I actually like the way this works, but I just wanted to be sure there was no programmable way to control it.

With regard to pixelated PNG files, this sounds about right. I don't know how to apply the "fix" to my Windows machine, though. :-(

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: Function Graphics Questions

Posted by [David Fanning](#) on Tue, 14 Jan 2014 22:47:17 GMT

[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> Dunno. I always create png output or eps for including in documents so I
> don't use /landscape.

Do you use software that automatically resizes these gigantic PNG files that are created in function graphics? What software is it? What do you do if you want to e-mail one of these babies?

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: Function Graphics Questions

Posted by [Paul Van Delst\[1\]](#) on Thu, 16 Jan 2014 17:12:13 GMT

[View Forum Message](#) <> [Reply to Message](#)

All of my IDL-based png output is created thusly (for an imaginary FG object "p"):

```
p.save, "file.png", HEIGHT=600
```

The height value for a default sized function graphics window creates pngs that fit nicely when I attach and display them in my trac scm ticket logs. :o)

Good size for emailing too.

cheers,

paulv

On 01/14/14 17:47, David Fanning wrote:

> Paul van Delst writes:

>

>> Dunno. I always create png output or eps for including in documents so I

>> don't use /landscape.

>

> Do you use software that automatically resizes these gigantic PNG files

> that are created in function graphics? What software is it? What do you

> do if you want to e-mail one of these babies?

>

> Cheers,

>

> David

>

Subject: Re: Function Graphics Questions

Posted by [David Fanning](#) on Thu, 16 Jan 2014 17:19:52 GMT

[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> All of my IDL-based png output is created thusly (for an imaginary FG

> object "p"):

>

> p.save, "file.png", HEIGHT=600

>

> The height value for a default sized function graphics window creates

> pngs that fit nicely when I attach and display them in my trac scm

> ticket logs. :o)

>

> Good size for emailing too.

Yes, this is what I hoped to do, too. But, the output from doing this on my machine in IDL 8.2.3 is crap. Very low resolution. :-(

Maybe this has been fixed in IDL 8.3?

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: Function Graphics Questions
Posted by [Paul Van Delst\[1\]](#) on Thu, 16 Jan 2014 18:29:11 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 01/16/14 12:19, David Fanning wrote:

> Paul van Delst writes:

>

>> All of my IDL-based png output is created thusly (for an imaginary FG

>> object "p"):

>>

>> p.save, "file.png", HEIGHT=600

>>

>> The height value for a default sized function graphics window creates

>> pngs that fit nicely when I attach and display them in my trac scm

>> ticket logs. :o)

>>

>> Good size for emailing too.

>

> Yes, this is what I hoped to do, too. But, the output from doing this on

> my machine in IDL 8.2.3 is crap. Very low resolution. :-(

I'm curious - is it just the line plots that are low res? Are all the accompanying graphics (axis text) shown in high res (i.e. not blocky)?

I only had issues with the lines in my plots. All the other plot accoutrements were fine.

Subject: Re: Function Graphics Questions
Posted by [David Fanning](#) on Thu, 16 Jan 2014 18:45:20 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst writes:

> I'm curious - is it just the line plots that are low res? Are all the

> accompanying graphics (axis text) shown in high res (i.e. not blocky)?

>

> I only had issues with the lines in my plots. All the other plot

> accoutrements were fine.

Yeah, I guess it's just the lines. The plot I was working with had three

lines on it. All very low resolution. The plot annotations look OK.

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: Function Graphics Questions

Posted by [Paul Van Delst\[1\]](#) on Thu, 16 Jan 2014 18:56:47 GMT

[View Forum Message](#) <> [Reply to Message](#)

Looks like the same/similar problem then.

I would contact exelis for a windows work-around until it's fixed in v8.3.whatever. Their linux workaround is giving me smooth lines again.

FWIW (prolly not much), PDF output is always smooth (well, in my test script plot). JPG output suffers from the same PNG-y issue.

On 01/16/14 13:45, David Fanning wrote:

> Paul van Delst writes:

>

>> I'm curious - is it just the line plots that are low res? Are all the

>> accompanying graphics (axis text) shown in high res (i.e. not blocky)?

>>

>> I only had issues with the lines in my plots. All the other plot

>> accoutrements were fine.

>

> Yeah, I guess it's just the lines. The plot I was working with had three

> lines on it. All very low resolution. The plot annotations look OK.

>

> Cheers,

>

> David

>

Subject: Re: Function Graphics Questions

Posted by [chris_torrence@NOSPAM](#) on Fri, 17 Jan 2014 15:32:40 GMT

[View Forum Message](#) <> [Reply to Message](#)

On Thursday, January 16, 2014 11:45:20 AM UTC-7, David Fanning wrote:

> Paul van Delst writes:

>

>

>

>> I'm curious - is it just the line plots that are low res? Are all the

>

>> accompanying graphics (axis text) shown in high res (i.e. not blocky)?

>

>>

>

>> I only had issues with the lines in my plots. All the other plot

>

>> accoutrements were fine.

>

>

>

> Yeah, I guess it's just the lines. The plot I was working with had three

>

> lines on it. All very low resolution. The plot annotations look OK.

>

>

>

> 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.")

Hi David,

Can you post your plot with the bad lines somewhere, or email it to me?

Thanks!

-Chris

Subject: Re: Function Graphics Questions
Posted by [David Fanning](#) on Fri, 17 Jan 2014 16:53:56 GMT
[View Forum Message](#) <> [Reply to Message](#)

Chris Torrence writes:

> Can you post your plot with the bad lines somewhere, or email it to me?

Hi Chris,

Out of town at the moment and my e-mail is causing me problems. But, you can get the code that produces the image in question here:

http://www.idlcoyote.com/gallery/additional_axes_plot_fg.pro

The image it produces is called additional_axes_plot_fg.png.

Cheers,

David

--

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: <http://www.idlcoyote.com/>

Seppure ma de ni thue. ("Perhaps thos speakest truth.")

Subject: Re: Function Graphics Questions
Posted by [chris_torrence@NOSPAM](#) on Fri, 17 Jan 2014 21:14:47 GMT
[View Forum Message](#) <> [Reply to Message](#)

On Friday, January 17, 2014 9:53:56 AM UTC-7, David Fanning wrote:

> Chris Torrence writes:

>

>

>

>> Can you post your plot with the bad lines somewhere, or email it to me?

>

>

>

> Hi Chris,

>

>

>

> Out of town at the moment and my e-mail is causing me problems. But, you

>

> can get the code that produces the image in question here:

>

>
>
> http://www.idlcoyote.com/gallery/additional_axes_plot_fg.pro
>
>
>
> The image it produces is called additional_axes_plot_fg.png.
>
>
>
> 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 thos speakest truth.")

Okay, I got it all to run. You're right, it looks pretty chunky at 600 pixels. I had toyed around with adding "anti-aliasing" to the output, but I never finished the code. I'll dig it up and see if I can get it into the next release.

It would basically do the same thing - construct the output at a high resolution, then scale it down to the desired resolution and output the result. The advantage would be that it wouldn't need to write out an intermediate file or rely on imagemagick.

-Chris

Subject: Re: Function Graphics Questions
Posted by [chris_torrence@NOSPAM](#) on Tue, 18 Feb 2014 21:37:14 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi all,

I just added antialias support to the new graphics ::Save method. By default, for bitmap file formats, if your resolution is less than 300dpi, then it will use anti-aliasing to produce a smooth output result. You can use a new ANTIALIAS keyword to control the behavior.

If anyone want to try out the code, let me know.

Cheers,
Chris
