Subject: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?
Posted by jkj on Fri, 29 Nov 2013 21:01:04 GMT
View Forum Message <> Reply to Message

Hi,

The researchers I work for [San Antonio, Southwest Research Institute, Space Science & Engineering] are using object graphics for the interactivity it provides.

We work with a lot of very small-valued data (electron precipitation measurements). In certain units the mean of our data will typically be 1.e-14. We consider 1.d-25 to be a reasonable measure of zero.

There are many quirks presented by IDL when data values become very small. One such quirk is "exploding text", as seen in this random selection of line plots, two showing well-behaved text (goodExample*.png) and the other two showing exploded text (badExample*.png):

http://safaripass.com/goodExample.png

http://safaripass.com/badExample.png

http://safaripass.com/goodExample2.png

http://safaripass.com/badExample2.png

I set recompute_dimensions=2 and allow IDL to calculate the size of the text but have also tried values of '0' and '1'. It does not, however, appear that this approach (IDL internal calculation of tick text dimensions) will work reliably with our data sets. We are working with IDL 8.1, I see exactly the same behaviour in IDL 5.5, no change between those versions.

I can make "most of the plots" well-behaved "most of the time" by paying careful attention to the yrange (or whatever range is very-small-valued) of the data. For example, if the data does not cover a full decade in log space then I need to increase the min/max for the range until a full decade is represented. I also need to avoid ranges with values like "1.129e-14", choosing something like "1.e-14" or "1.1e-14". This means I can not choose arbitrary ranges for fear of text explosion... so we try then choose to sacrifice the best view of the data for well-behaved text.

By and large I can "whack the range around" and get the tick text to behave well. It seems to me that the real solution is to calculate my own character dimensions and I think I understand how to do that but simple examples with large-valued data work but when I switch to our very-small data the approaches taken with the larger data values then fail and I get the same exploded text as when character dimensions are calculated internally by IDL.

That leaves me wondering if there are good examples of explicitly computing character dimensions laying around somewhere... and it also leaves me wondering if those of us who work with very-small data values are left at the mercy of something internal to IDL that ignores the possibility of such small-valued data sets.

Any thoughts would be appreciated. If IDL is simply internally deficient with respect to very small data then I should switch to some other method of putting up text. I am able to reliably display text summaries of ranges as a 'plot title' and those 'idlgrtext' summaries have yet to behave poorly, but the tick text behaviour is really unacceptable.

Thanks,
-Kevin

Subject: Re: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?
Posted by David Fanning on Fri, 29 Nov 2013 21:16:44 GMT

View Forum Message <> Reply to Message

jkj writes:

> That leaves me wondering if there are good examples of explicitly computing character dimensions laying around somewhere... and it also leaves me wondering if those of us who work with very-small data values are left at the mercy of something internal to IDL that ignores the possibility of such small-valued data sets.
>
> Any thoughts would be appreciated. If IDL is simply internally deficient with respect to very small data then I should switch to some other method of putting up text. I am able to reliably display text summaries of ranges as a 'plot title' and those 'idlgrtext' summaries have yet to behave poorly, but the tick text behaviour is really unacceptable.

I suppose another possibility is that OpenGL is to blame. I'd be curious
to know if MatLab or R has the same problems with this data.

But, there can't be too many people here who wouldn't dismiss those
kinds of numbers as "essentially zero." What have the folks at ExelisVis
had to say?

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: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?
Posted by jkj on Fri, 29 Nov 2013 21:30:52 GMT

View Forum Message <> Reply to Message

On Friday, November 29, 2013 3:16:44 PM UTC-6, David Fanning wrote:
> jkj writes:
>
> I suppose another possibility is that OpenGL is to blame. I'd be curious

Yea, we use Linux (Solaris in the past)... another complicator.

>
> to know if MatLab or R has the same problems with this data.

no idea, never use either

> But, there can't be too many people here who wouldn't dismiss those
> kinds of numbers as "essentially zero." What have the folks at ExelisVis
> had to say?

I figured they read this newsgroup but maybe I need to formally contact them through SwRI. I remember how you considered 1.d-12 as a reasonable value for zero and apparently the IDL Community at large had no issues with that, so our data is clearly not within the norm of other IDL users. Most imaging in the past has been generated by local, custom [SDDAS] software and this is the first real push for quality imaging through IDL... this sort of difficulty can become a brick wall in that regard.

-Kevin

>
>
>
> 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: object graphics, exploding axis text, how to fix by explicitly setting

char_dimens?
Posted by jkj on Mon, 02 Dec 2013 21:05:55 GMT
View Forum Message <> Reply to Message

On Friday, November 29, 2013 3:30:52 PM UTC-6, jkj wrote:
> On Friday, November 29, 2013 3:16:44 PM UTC-6, David Fanning wrote:
>
>> jkj writes:
>
>>
>
>> I suppose another possibility is that OpenGL is to blame. I'd be curious
>
>
>

I'll update this if anything changes but there are two issues in play here: IDL version and my thumbprint!

two-dimensional plots have 'exploding text' only if IDL 5.5 is used [hedging a bit here because I'd have sworn I've seen cases with 8.2]

three-dimensional surfaces have 'exploding text' on the z-axis and in submitting a ticket for ExelisVis it became clear that there is an error in how I am implementing "layered/phantom axes" [funny how preparing material for troubleshooting assistance generally leads one to the solution!]

the first set of axes is used to build the surface but the data displayed would simply be grid values [0->200 for a 200x200 grid] and we want to display the science data, so the surface is built with indgen() arrays for x/y data and then 'phantom axes' are built with the science data... somehow in that process I am fumbling because if I turn off the science data axes there is no exploding text and with the science axes on I can see evidence of confusion in the x/y/zrange values (which, of course, can not be explicitly set for 'idlgraxis')

anyway, I'll figure out how to wipe away my thumbprints and that should resolve this... the whole thing has been a bit quirky and intermittent enough that I still won't be surprised to find another element but I definitely think I'm the culprit in all of this

Subject: Re: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?
Posted by Starbuck on Tue, 10 Dec 2013 20:35:48 GMT
View Forum Message <> Reply to Message

On Friday, November 29, 2013 2:01:04 PM UTC-7, jkj wrote:
> Hi,
>
>
>

> The researchers I work for [San Antonio, Southwest Research Institute, Space Science & Engineering] are using object graphics for the interactivity it provides.
>
>
>
> We work with a lot of very small-valued data (electron precipitation measurements). In certain units the mean of our data will typically be 1.e-14. We consider 1.d-25 to be a reasonable measure of zero.
>
>
>
> There are many quirks presented by IDL when data values become very small. One such quirk is "exploding text", as seen in this random selection of line plots, two showing well-behaved text (goodExample*.png) and the other two showing exploded text (badExample*.png):
>
>
>
> http://safaripass.com/goodExample.png
>
>
>
> http://safaripass.com/badExample.png
>
>
>
> http://safaripass.com/goodExample2.png
>
>
>
> http://safaripass.com/badExample2.png
>
>
>
> I set recompute_dimensions=2 and allow IDL to calculate the size of the text but have also tried values of '0' and '1'. It does not, however, appear that this approach (IDL internal calculation of tick text dimensions) will work reliably with our data sets. We are working with IDL 8.1, I see exactly the same behaviour in IDL 5.5, no change between those versions.
>
>
>
> I can make "most of the plots" well-behaved "most of the time" by paying careful attention to the yrange (or whatever range is very-small-valued) of the data. For example, if the data does not cover a full decade in log space then I need to increase the min/max for the range until a full decade is represented. I also need to avoid ranges with values like "1.129e-14", choosing something like "1.e-14" or "1.1e-14". This means I can not choose arbitrary ranges for fear of text explosion... so we try then choose to sacrifice the best view of the data for well-behaved text.
>
>

>
> By and large I can "whack the range around" and get the tick text to behave well. It seems to me that the real solution is to calculate my own character dimensions and I think I understand how to do that but simple examples with large-valued data work but when I switch to our very-small data the approaches taken with the larger data values then fail and I get the same exploded text as when character dimensions are calculated internally by IDL.
>
>
>
> That leaves me wondering if there are good examples of explicitly computing character dimensions laying around somewhere... and it also leaves me wondering if those of us who work with very-small data values are left at the mercy of something internal to IDL that ignores the possibility of such small-valued data sets.
>
>
>
> Any thoughts would be appreciated. If IDL is simply internally deficient with respect to very small data then I should switch to some other method of putting up text. I am able to reliably display text summaries of ranges as a 'plot title' and those 'idlgrtext' summaries have yet to behave poorly, but the tick text behaviour is really unacceptable.
>
>
>
> Thanks,
>
> -Kevin

I was able to reproduce the problem with the following code:

```
h = double(HANNING(100,100)*2.3e-13)
s = surface(h,COLOR='black', style=1, CLIP=0)
```

I have filed a bug report IDL-68998 about the issue.

-Starbuck

Subject: Re: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?

Posted by lecacheux.alain on Tue, 10 Dec 2013 22:37:35 GMT

View Forum Message <> Reply to Message

Le mardi 10 décembre 2013 21:35:48 UTC+1, Starbuck a écrit :
> On Friday, November 29, 2013 2:01:04 PM UTC-7, jkj wrote:
>
>> Hi,
>
>>

>
>>
>
>>
>
>> The researchers I work for [San Antonio, Southwest Research Institute, Space Science & Engineering] are using object graphics for the interactivity it provides.
>
>>
>
>>
>
>>
>
>> We work with a lot of very small-valued data (electron precipitation measurements). In certain units the mean of our data will typically be 1.e-14. We consider 1.d-25 to be a reasonable measure of zero.
>
>>
>
>>
>
>>
>
>> There are many quirks presented by IDL when data values become very small. One such quirk is "exploding text", as seen in this random selection of line plots, two showing well-behaved text (goodExample*.png) and the other two showing exploded text (badExample*.png):
>
>>
>
>>
>
>>
>
>> http://safaripass.com/goodExample.png
>
>>
>
>>
>
>>
>
>> http://safaripass.com/badExample.png
>
>>
>
>>
>

>>
>
>> http://safaripass.com/goodExample2.png
>
>>
>
>>
>
>>
>
>> http://safaripass.com/badExample2.png
>
>>
>
>>
>
>>
>
>> I set recompute_dimensions=2 and allow IDL to calculate the size of the text but have also tried values of '0' and '1'. It does not, however, appear that this approach (IDL internal calculation of tick text dimensions) will work reliably with our data sets. We are working with IDL 8.1, I see exactly the same behaviour in IDL 5.5, no change between those versions.
>
>>
>
>>
>
>>
>
>> I can make "most of the plots" well-behaved "most of the time" by paying careful attention to the yrange (or whatever range is very-small-valued) of the data. For example, if the data does not cover a full decade in log space then I need to increase the min/max for the range until a full decade is represented. I also need to avoid ranges with values like "1.129e-14", choosing something like "1.e-14" or "1.1e-14". This means I can not choose arbitrary ranges for fear of text explosion... so we try then choose to sacrifice the best view of the data for well-behaved text.
>
>>
>
>>
>
>>
>
>> By and large I can "whack the range around" and get the tick text to behave well. It seems to me that the real solution is to calculate my own character dimensions and I think I understand how to do that but simple examples with large-valued data work but when I switch to our very-small data the approaches taken with the larger data values then fail and I get the same exploded text as when character dimensions are calculated internally by IDL.
>

>>
>
>>
>
>>
>
>> That leaves me wondering if there are good examples of explicitly computing character dimensions laying around somewhere... and it also leaves me wondering if those of us who work with very-small data values are left at the mercy of something internal to IDL that ignores the possibility of such small-valued data sets.
>
>>
>
>>
>
>>
>
>> Any thoughts would be appreciated. If IDL is simply internally deficient with respect to very small data then I should switch to some other method of putting up text. I am able to reliably display text summaries of ranges as a 'plot title' and those 'idlgrtext' summaries have yet to behave poorly, but the tick text behaviour is really unacceptable.
>
>>
>
>>
>
>>
>
>> Thanks,
>
>>
>
>> -Kevin
>
>
>
> I was able to reproduce the problem with the following code:
>
>
>
> h = double(HANNING(100,100)*2.3e-13)
>
> s = surface(h,COLOR='black', style=1, CLIP=0)
>
>
>
> I have filed a bug report IDL-68998 about the issue.
>

>
>
> -Starbuck

No problem for me with your code on two different machines
{ x86 Win32 Windows Microsoft Windows 8.2.3 May 3 2013 32 64}
and
{ x86_64 Win32 Windows Microsoft Windows 8.2.3 May 3 2013 64 64}
running Win7 !
Maybe, as Fanning suggested, an OpenGL or video driver problem ?
alx.

Subject: Re: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?
Posted by David Fanning on Tue, 10 Dec 2013 23:14:03 GMT

View Forum Message <> Reply to Message

alx writes:

> No problem for me with your code on two different machines
> { x86 Win32 Windows Microsoft Windows 8.2.3 May 3 2013 32 64}
> and
> { x86_64 Win32 Windows Microsoft Windows 8.2.3 May 3 2013 64 64}
> running Win7 !

Humm. I see the problem on my Windows 64-bit IDL 8.2.3 with both
software and hardware rendering turned on. Strange!

You are rotating the surface, right?

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: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?
Posted by lecacheux.alain on Tue, 10 Dec 2013 23:49:02 GMT

View Forum Message <> Reply to Message

Le mercredi 11 décembre 2013 00:14:03 UTC+1, David Fanning a écrit :

> alx writes:
>
>
>
>> No problem for me with your code on two different machines
>
>> { x86 Win32 Windows Microsoft Windows 8.2.3 May 3 2013   32   64}
>
>> and
>
>> { x86_64 Win32 Windows Microsoft Windows 8.2.3 May 3 2013   64   64}
>
>> running Win7 !
>
>
>
> Humm. I see the problem on my Windows 64-bit IDL 8.2.3 with both
>
> software and hardware rendering turned on. Strange!
>
>
>
> You are rotating the surface, right?
>
>
>
> 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.")

Well. Sorry. I was looking too quickly. The problem indeed arises when the image is rotated at z very close to 0. Forget my previous message.
alx.

Subject: Re: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?
Posted by andeh on Wed, 11 Dec 2013 11:59:27 GMT
View Forum Message <> Reply to Message

On Tuesday, 10 December 2013 20:35:48 UTC, Starbuck wrote:
> On Friday, November 29, 2013 2:01:04 PM UTC-7, jkj wrote:
>
>> Hi,
>
>>
>
>>
>
>>
>
>> The researchers I work for [San Antonio, Southwest Research Institute, Space Science & Engineering] are using object graphics for the interactivity it provides.
>
>>
>
>>
>
>>
>
>> We work with a lot of very small-valued data (electron precipitation measurements). In certain units the mean of our data will typically be 1.e-14. We consider 1.d-25 to be a reasonable measure of zero.
>
>>
>
>>
>
>>
>
>> There are many quirks presented by IDL when data values become very small. One such quirk is "exploding text", as seen in this random selection of line plots, two showing well-behaved text (goodExample*.png) and the other two showing exploded text (badExample*.png):
>
>>
>
>>
>
>>
>
>> http://safaripass.com/goodExample.png
>
>>
>

>>
>
>>
>
>> http://safaripass.com/badExample.png
>
>>
>
>>
>
>>
>
>> http://safaripass.com/goodExample2.png
>
>>
>
>>
>
>>
>
>> http://safaripass.com/badExample2.png
>
>>
>
>>
>
>>
>
>> I set recompute_dimensions=2 and allow IDL to calculate the size of the text but have also tried values of '0' and '1'. It does not, however, appear that this approach (IDL internal calculation of tick text dimensions) will work reliably with our data sets. We are working with IDL 8.1, I see exactly the same behaviour in IDL 5.5, no change between those versions.
>
>>
>
>>
>
>>
>
>> I can make "most of the plots" well-behaved "most of the time" by paying careful attention to the yrange (or whatever range is very-small-valued) of the data. For example, if the data does not cover a full decade in log space then I need to increase the min/max for the range until a full decade is represented. I also need to avoid ranges with values like "1.129e-14", choosing something like "1.e-14" or "1.1e-14". This means I can not choose arbitrary ranges for fear of text explosion... so we try then choose to sacrifice the best view of the data for well-behaved text.
>
>>
>

>>
>
>>
>
>> By and large I can "whack the range around" and get the tick text to behave well. It seems to me that the real solution is to calculate my own character dimensions and I think I understand how to do that but simple examples with large-valued data work but when I switch to our very-small data the approaches taken with the larger data values then fail and I get the same exploded text as when character dimensions are calculated internally by IDL.
>
>>
>
>>
>
>>
>
>> That leaves me wondering if there are good examples of explicitly computing character dimensions laying around somewhere... and it also leaves me wondering if those of us who work with very-small data values are left at the mercy of something internal to IDL that ignores the possibility of such small-valued data sets.
>
>>
>
>>
>
>>
>
>> Any thoughts would be appreciated. If IDL is simply internally deficient with respect to very small data then I should switch to some other method of putting up text. I am able to reliably display text summaries of ranges as a 'plot title' and those 'idlgrtext' summaries have yet to behave poorly, but the tick text behaviour is really unacceptable.
>
>>
>
>>
>
>>
>
>> Thanks,
>
>>
>
>> -Kevin
>
>
>
> I was able to reproduce the problem with the following code:
>

>
>
> h = double(HANNING(100,100)*2.3e-13)
>
> s = surface(h,COLOR='black', style=1, CLIP=0)
>
>
>
> I have filed a bug report IDL-68998 about the issue.
>
>
>
> -Starbuck

I also get strange behaviour in contour.

```
IDL> c = CONTOUR(h,C_LABEL_SHOW=1,XRANGE=[60,80])
IDL> HELP, !VERSION
** Structure !VERSION, 8 tags, length=104, data length=100:
   ARCH            STRING    'x86_64'
   OS              STRING    'linux'
   OS_FAMILY       STRING    'unix'
   OS_NAME         STRING    'linux'
   RELEASE         STRING    '8.2'
   BUILD_DATE      STRING    'Apr 10 2012'
   MEMORY_BITS     INT             64
   FILE_OFFSET_BITS
                   INT             64
```

Andy

Subject: Re: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?
Posted by David Fanning on Wed, 11 Dec 2013 12:58:05 GMT
View Forum Message <> Reply to Message

AJAS writes:

> I also get strange behaviour in contour.
>
> IDL> c = CONTOUR(h,C_LABEL_SHOW=1,XRANGE=[60,80])

I do note that cgSurface and cgContour handle these two strange cases
flawlessly. ;-)

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: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?
Posted by lecacheux.alain on Wed, 11 Dec 2013 14:01:00 GMT
View Forum Message <> Reply to Message

Le mercredi 11 décembre 2013 12:59:27 UTC+1, AJAS a écrit :
> On Tuesday, 10 December 2013 20:35:48 UTC, Starbuck wrote:
>
>> On Friday, November 29, 2013 2:01:04 PM UTC-7, jkj wrote:
>
>>
>
>>> Hi,
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> The researchers I work for [San Antonio, Southwest Research Institute, Space Science & Engineering] are using object graphics for the interactivity it provides.
>
>>
>
>>>
>
>>
>

>>>
>
>>
>
>>>
>
>>
>
>>> We work with a lot of very small-valued data (electron precipitation measurements). In certain units the mean of our data will typically be 1.e-14. We consider 1.d-25 to be a reasonable measure of zero.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> There are many quirks presented by IDL when data values become very small. One such quirk is "exploding text", as seen in this random selection of line plots, two showing well-behaved text (goodExample*.png) and the other two showing exploded text (badExample*.png):
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> http://safaripass.com/goodExample.png
>
>>
>

>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> http://safaripass.com/badExample.png
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> http://safaripass.com/goodExample2.png
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> http://safaripass.com/badExample2.png
>
>>
>

>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> I set recompute_dimensions=2 and allow IDL to calculate the size of the text but have also tried values of '0' and '1'. It does not, however, appear that this approach (IDL internal calculation of tick text dimensions) will work reliably with our data sets. We are working with IDL 8.1, I see exactly the same behaviour in IDL 5.5, no change between those versions.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> I can make "most of the plots" well-behaved "most of the time" by paying careful attention to the yrange (or whatever range is very-small-valued) of the data. For example, if the data does not cover a full decade in log space then I need to increase the min/max for the range until a full decade is represented. I also need to avoid ranges with values like "1.129e-14", choosing something like "1.e-14" or "1.1e-14". This means I can not choose arbitrary ranges for fear of text explosion... so we try then choose to sacrifice the best view of the data for well-behaved text.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>

>>>
>
>>
>
>>> By and large I can "whack the range around" and get the tick text to behave well. It seems to me that the real solution is to calculate my own character dimensions and I think I understand how to do that but simple examples with large-valued data work but when I switch to our very-small data the approaches taken with the larger data values then fail and I get the same exploded text as when character dimensions are calculated internally by IDL.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> That leaves me wondering if there are good examples of explicitly computing character dimensions laying around somewhere... and it also leaves me wondering if those of us who work with very-small data values are left at the mercy of something internal to IDL that ignores the possibility of such small-valued data sets.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> Any thoughts would be appreciated. If IDL is simply internally deficient with respect to very small data then I should switch to some other method of putting up text. I am able to reliably display text summaries of ranges as a 'plot title' and those 'idlgrtext' summaries have yet to behave poorly, but the tick text behaviour is really unacceptable.
>

>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> Thanks,
>
>>
>
>>>
>
>>
>
>>> -Kevin
>
>>
>
>>
>
>>
>
>> I was able to reproduce the problem with the following code:
>
>>
>
>>
>
>>
>
>> h = double(HANNING(100,100)*2.3e-13)
>
>>
>
>> s = surface(h,COLOR='black', style=1, CLIP=0)
>
>>
>
>>
>

```
>>
>
>>  I have filed a bug report IDL-68998 about the issue.
>
>>
>
>>
>
>>
>
>>  -Starbuck
>
>
>
>  I also get strange behaviour in contour.
>
>
>
>  IDL> c = CONTOUR(h,C_LABEL_SHOW=1,XRANGE=[60,80])
>
>  IDL> HELP, !VERSION
>
>  ** Structure !VERSION, 8 tags, length=104, data length=100:
>
>     ARCH            STRING    'x86_64'
>
>     OS              STRING    'linux'
>
>     OS_FAMILY       STRING    'unix'
>
>     OS_NAME         STRING    'linux'
>
>     RELEASE         STRING    '8.2'
>
>     BUILD_DATE      STRING    'Apr 10 2012'
>
>     MEMORY_BITS     INT             64
>
>     FILE_OFFSET_BITS
>
>                     INT             64
>
>
>
>
>
>  Andy
```

It seems to be a scaling problem. With c.ASPECT_RATIO=1, the labels are well formed, then get unreadable when c.ASPECT_RATIO goes down to 0 (the default).
alx.

Subject: Re: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?

Posted by jkj on Wed, 11 Dec 2013 14:25:07 GMT

View Forum Message <> Reply to Message

On Wednesday, December 11, 2013 8:01:00 AM UTC-6, alx wrote:
> Le mercredi 11 décembre 2013 12:59:27 UTC+1, AJAS a écrit :
>
>> On Tuesday, 10 December 2013 20:35:48 UTC, Starbuck wrote:
>
>>
>
>>> On Friday, November 29, 2013 2:01:04 PM UTC-7, jkj wrote:
>
>>
>
>>>
>
>>
>
>>>> Hi,
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>

>
>>>>
>
>>
>
>>>
>
>>
>
>>>> The researchers I work for [San Antonio, Southwest Research Institute, Space Science & Engineering] are using object graphics for the interactivity it provides.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> We work with a lot of very small-valued data (electron precipitation measurements). In certain units the mean of our data will typically be 1.e-14. We consider 1.d-25 to be a reasonable measure of zero.
>
>>
>

>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> There are many quirks presented by IDL when data values become very small. One such quirk is "exploding text", as seen in this random selection of line plots, two showing well-behaved text (goodExample*.png) and the other two showing exploded text (badExample*.png):
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>

```
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> http://safaripass.com/goodExample.png
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> http://safaripass.com/badExample.png
>
```

>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> http://safaripass.com/goodExample2.png
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>

>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> http://safaripass.com/badExample2.png
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> I set recompute_dimensions=2 and allow IDL to calculate the size of the text but have also tried values of '0' and '1'. It does not, however, appear that this approach (IDL internal calculation

of tick text dimensions) will work reliably with our data sets. We are working with IDL 8.1, I see exactly the same behaviour in IDL 5.5, no change between those versions.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> I can make "most of the plots" well-behaved "most of the time" by paying careful attention to the yrange (or whatever range is very-small-valued) of the data. For example, if the data does not cover a full decade in log space then I need to increase the min/max for the range until a full decade is represented. I also need to avoid ranges with values like "1.129e-14", choosing something like "1.e-14" or "1.1e-14". This means I can not choose arbitrary ranges for fear of text explosion... so we try then choose to sacrifice the best view of the data for well-behaved text.
>
>>
>
>>>
>
>>
>
>>>>
>

>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> By and large I can "whack the range around" and get the tick text to behave well. It seems to me that the real solution is to calculate my own character dimensions and I think I understand how to do that but simple examples with large-valued data work but when I switch to our very-small data the approaches taken with the larger data values then fail and I get the same exploded text as when character dimensions are calculated internally by IDL.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>

>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> That leaves me wondering if there are good examples of explicitly computing character dimensions laying around somewhere... and it also leaves me wondering if those of us who work with very-small data values are left at the mercy of something internal to IDL that ignores the possibility of such small-valued data sets.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Any thoughts would be appreciated. If IDL is simply internally deficient with respect to very small data then I should switch to some other method of putting up text. I am able to reliably display text summaries of ranges as a 'plot title' and those 'idlgrtext' summaries have yet to

behave poorly, but the tick text behaviour is really unacceptable.
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>
>>>> Thanks,
>
>>
>
>>>
>
>>
>
>>>>
>
>>
>
>>>
>
>>
>

>>>> -Kevin
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> I was able to reproduce the problem with the following code:
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> h = double(HANNING(100,100)*2.3e-13)
>
>>
>
>>>
>
>>
>
>>> s = surface(h,COLOR='black', style=1, CLIP=0)
>
>>
>
>>>
>
>>
>

>>>
>
>>
>
>>>
>
>>
>
>>> I have filed a bug report IDL-68998 about the issue.
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>>
>
>>
>
>>> -Starbuck
>
>>
>
>>
>
>>
>
>> I also get strange behaviour in contour.
>
>>
>
>>
>
>>
>
>> IDL> c = CONTOUR(h,C_LABEL_SHOW=1,XRANGE=[60,80])
>
>>
>
>> IDL> HELP, !VERSION
>
>>
>

```
>>  ** Structure !VERSION, 8 tags, length=104, data length=100:
>
>>
>
>>    ARCH            STRING    'x86_64'
>
>>
>
>>    OS              STRING    'linux'
>
>>
>
>>    OS_FAMILY       STRING    'unix'
>
>>
>
>>    OS_NAME         STRING    'linux'
>
>>
>
>>    RELEASE         STRING    '8.2'
>
>>
>
>>    BUILD_DATE      STRING    'Apr 10 2012'
>
>>
>
>>    MEMORY_BITS     INT             64
>
>>
>
>>    FILE_OFFSET_BITS
>
>>
>
>>                    INT             64
>
>>
>
>>
>
>>
>
>>
>
>>
>
```

>> Andy
>
>
>
> It seems to be a scaling problem. With c.ASPECT_RATIO=1, the labels are well formed, then get unreadable when c.ASPECT_RATIO goes down to 0 (the default).
>
> alx.

I'm guessing all this activity relates to yesterday's bug report IDL-68998 since the example from Starbuck is exactly what was provided to me by Exelis.

Yes, this was cited as a probable bug. If there are any others out there who use such small-valued data sets then please be aware and communicate any issues to Exelis.

My guess is that something internal in IDL is involving a float where a double is required.

Subject: Re: object graphics, exploding axis text, how to fix by explicitly setting char_dimens?

Posted by jkj on Wed, 11 Dec 2013 14:28:41 GMT

View Forum Message <> Reply to Message

On Wednesday, December 11, 2013 6:58:05 AM UTC-6, David Fanning wrote:
> AJAS writes:
>
>
>
>> I also get strange behaviour in contour.
>
>>
>
>> IDL> c = CONTOUR(h,C_LABEL_SHOW=1,XRANGE=[60,80])
>
>
>
> I do note that cgSurface and cgContour handle these two strange cases
>
> flawlessly. ;-)
>

I have also experienced exploding axis text while using cgSurface, one of the reasons we wrote our own from scratch, but it was very nice to have your code to review. This is almost certainly an internal IDL issue. Your contributions are rightly well-admired and appreciated.