Subject: Questions about NG image with (log) axis
Posted by Xin Tao on Wed, 07 Aug 2013 06:38:51 GMT
View Forum Message <> Reply to Message

Hi,

I'm not sure whether I understand the image function from IDL 8 correctly. I want to plot a 2D image with x and y axis. I've observed at least two strange behaviours. Please give me some help. Thanks.

data = dist(300)

My question 1: What determines the size of the image?
If I have

x=dindgen(300)*2
y=dindgen(300)

then im=image(data, x, y, axis_style=2) will give me an elongated image.

If I have
x=dindgen(300)
y=dindgen(300)

then im=image(data, x, y, axis_style=2) will give me an square image.

This behavior of the IMAGE function is strange to me.

My question 2: This looks like a bug.

If I have

x=dlinear(10, 100, 300) ;; create an array from 10 to 100 with 300 elements
y=dindgen(300)

and

im=image(data, x, y, /xlog, axis_style=2)

the image x axis has a value from 10^20, 10^40, ..., 10^100. This is certainly wrong to me.

Am I missing something here?

Subject: Re: Questions about NG image with (log) axis
Posted by Fabzi on Wed, 07 Aug 2013 06:53:30 GMT
View Forum Message <> Reply to Message

Hi Xin Tao,

On 08/07/2013 08:38 AM, Xin Tao wrote:
> data = dist(300)
>
> My question 1: What determines the size of the image?
> If I have
>
> x=dindgen(300)*2
> y=dindgen(300)
>
> then im=image(data, x, y, axis_style=2) will give me an elongated image.
>
> If I have
> x=dindgen(300)
> y=dindgen(300)
>
> then im=image(data, x, y, axis_style=2) will give me an square image.
>
> This behavior of the IMAGE function is strange to me.

This is documented here I think:

ASPECT_RATIO
A floating point value indicating the ratio of the Y dimension to the X
dimension in data units. If this property is set to a nonzero value, the
aspect ratio will be preserved as the graphic is stretched or shrunk.
The default value is 0 for all graphics except images, meaning that the
aspect ratio is not fixed, but is allowed to change as the graphic is
stretched or shrunk.

```
IDL> y=dindgen(300)
IDL> x=dindgen(300)*2
IDL> im=image(data, x, y, axis_style=2, ASPECT_RATIO=2.)
```

is quadratic

>
> My question 2: This looks like a bug.
>
> If I have
>
> x=dlinear(10, 100, 300) ;; create an array from 10 to 100 with 300 elements
> y=dindgen(300)
>
> and
>
> im=image(data, x, y, /xlog, axis_style=2)

>
> the image x axis has a value from 10^20, 10^40, ..., 10^100. This is certainly wrong to me.

I don't know what dlinear does so I can't say anything here.

Cheers,

Fab

Subject: Re: Questions about NG image with (log) axis
Posted by Xin Tao on Wed, 07 Aug 2013 09:22:29 GMT

View Forum Message <> Reply to Message

Thank you very much, Fab.

The purpose of dlinear is to create an array (in my case) from 10 to 100 linearly with 300 elements.
It's equivalent to

x=10.0+dindgen(300)*(100-10.0)/(300-1)

Best,

Xin

On Wednesday, August 7, 2013 2:53:30 PM UTC+8, Fabien wrote:
> Hi Xin Tao,
>
>
>
> On 08/07/2013 08:38 AM, Xin Tao wrote:
>
>> data = dist(300)
>
>>
>
>> My question 1: What determines the size of the image?
>
>> If I have
>
>>
>
>> x=dindgen(300)*2
>
>> y=dindgen(300)
>
>>

>
>> then im=image(data, x, y, axis_style=2) will give me an elongated image.
>
>>
>
>> If I have
>
>> x=dindgen(300)
>
>> y=dindgen(300)
>
>>
>
>> then im=image(data, x, y, axis_style=2) will give me an square image.
>
>>
>
>> This behavior of the IMAGE function is strange to me.
>
>
>
> This is documented here I think:
>
>
>
> ASPECT_RATIO
>
> A floating point value indicating the ratio of the Y dimension to the X
>
> dimension in data units. If this property is set to a nonzero value, the
>
> aspect ratio will be preserved as the graphic is stretched or shrunk.
>
> The default value is 0 for all graphics except images, meaning that the
>
> aspect ratio is not fixed, but is allowed to change as the graphic is
>
> stretched or shrunk.
>
>
>
> IDL> y=dindgen(300)
>
> IDL> x=dindgen(300)*2
>
> IDL> im=image(data, x, y, axis_style=2, ASPECT_RATIO=2.)
>
>

>
> is quadratic
>
>
>
>>
>
>> My question 2: This looks like a bug.
>
>>
>
>> If I have
>
>>
>
>> x=dlinear(10, 100, 300) ;; create an array from 10 to 100 with 300 elements
>
>> y=dindgen(300)
>
>>
>
>> and
>
>>
>
>> im=image(data, x, y, /xlog, axis_style=2)
>
>>
>
>> the image x axis has a value from 10^20, 10^40, ..., 10^100. This is certainly wrong to me.
>
>
>
> I don't know what dlinear does so I can't say anything here.
>
>
>
> Cheers,
>
>
>
> Fab

Subject: Re: Questions about NG image with (log) axis
Posted by Fabzi on Wed, 07 Aug 2013 11:59:56 GMT
View Forum Message <> Reply to Message

On 08/07/2013 11:22 AM, Xin Tao wrote:
> the image x axis has a value from 10^20, 10^40, ..., 10^100. This is certainly wrong to me.

Yes indeed, it looks like a bug. I am not a NG specialist though.

I know it's not really constructive, but since David is not here I'll
try to replace him as good as I can:

```
x=10.0+dindgen(300)*(100-10.0)/(300-1)
y=dindgen(300)
data = dist(300)
cgImage, data, x, y, /KEEP_ASPECT_RATIO, /AXES, MARGIN=1, $
  AXKEYWORDS={XLOG:1, XRANGE:[10,100]}, /WINDOW
```

Subject: Re: Questions about NG image with (log) axis
Posted by Xin Tao on Wed, 07 Aug 2013 12:10:17 GMT

View Forum Message <> Reply to Message

Thanks Fab.

I know David's Coyote library, it's very nice. I have used it a lot in the past. However, the NG in IDL 8 looks quite promising as the future of IDL plotting tools, and that's why I'm working on that. :-)

Cheers,

Xin Tao

On Wednesday, August 7, 2013 7:59:56 PM UTC+8, Fabien wrote:
> On 08/07/2013 11:22 AM, Xin Tao wrote:
>
>> the image x axis has a value from 10^20, 10^40, ..., 10^100. This is certainly wrong to me.
>
>
>
> Yes indeed, it looks like a bug. I am not a NG specialist though.
>
>
>
> I know it's not really constructive, but since David is not here I'll
>
> try to replace him as good as I can:
>
>
>
> x=10.0+dindgen(300)*(100-10.0)/(300-1)
>

> y=dindgen(300)
>
> data = dist(300)
>
> cgImage, data, x, y, /KEEP_ASPECT_RATIO, /AXES, MARGIN=1, $
>
>   AXKEYWORDS={XLOG:1, XRANGE:[10,100]}, /WINDOW

Subject: Re: Questions about NG image with (log) axis

Posted by chris_torrence@NOSPAM on Thu, 15 Aug 2013 17:37:25 GMT

View Forum Message <> Reply to Message

On Wednesday, August 7, 2013 6:10:17 AM UTC-6, Xin Tao wrote:
> Thanks Fab.
>
>
>
> I know David's Coyote library, it's very nice. I have used it a lot in the past. However, the NG in IDL 8 looks quite promising as the future of IDL plotting tools, and that's why I'm working on that. :-)
>
>
>
> Cheers,
>
>
>
> Xin Tao
>
>
>
> On Wednesday, August 7, 2013 7:59:56 PM UTC+8, Fabien wrote:
>
>> On 08/07/2013 11:22 AM, Xin Tao wrote:
>
>>
>
>>> the image x axis has a value from 10^20, 10^40, ..., 10^100. This is certainly wrong to me.
>
>>
>
>>
>
>>
>
>> Yes indeed, it looks like a bug. I am not a NG specialist though.
>

>>
>
>>
>
>>
>
>> I know it's not really constructive, but since David is not here I'll
>
>>
>
>> try to replace him as good as I can:
>
>>
>
>>
>
>>
>
>> x=10.0+dindgen(300)*(100-10.0)/(300-1)
>
>>
>
>> y=dindgen(300)
>
>>
>
>> data = dist(300)
>
>>
>
>> cgImage, data, x, y, /KEEP_ASPECT_RATIO, /AXES, MARGIN=1, $
>
>>
>
>>   AXKEYWORDS={XLOG:1, XRANGE:[10,100]}, /WINDOW

Hi Xin Tao,

Well, I think both David's coyote library and IDL's NG are going to have problems with a log axis for an Image. When I try David's code (yes, I have it installed!) the axes "look" correct, but the image is just plotted as a normal image.

If the X axis were truly a "log" axis, then I would expect the pixels to be stretched out on the left side and get smaller as you moved to the right. For example, try the following:

```
x=10.0+dindgen(300)*(100-10.0)/(300-1)
y=dindgen(300)
data = dist(300)
```

c = contour(data, x, y, /xlog, /fill)

Notice how the contour plot is more stretched out on the left side. Is this what you want? If so, then you aren't going to be able to use an Image (either from NG or Coyote). You could use a filled contour plot, say with n_levels=50.

On the other hand, if you don't want a stretched out image, then you might want to take the log10 of your X data first, then do the i=image() and set the xtickvalues/xtickname to get the correct labels. In the meantime, I'll try to fix the Image function for IDL 8.3 so it at least gives the correct labels, even if it doesn't stretch out the pixels.

Cheers,
Chris
ExelisVIS

Subject: Re: Questions about NG image with (log) axis
Posted by David Fanning on Thu, 15 Aug 2013 17:52:44 GMT

View Forum Message <> Reply to Message

Chris Torrence writes:

> Well, I think both David's coyote library and IDL's NG are going to have problems with a log axis for an Image. When I try David's code (yes, I have it installed!) the axes "look" correct, but the image is just plotted as a normal image.

Yes, I'm not log scaling any image data. I presume the user knows how to
do that for himself! I think Paolo Grigis's routine pg_plotimage does
the logarithmic interpolation, though. You can find a version of his
code in the "public" directory of the Coyote Library distribution.

Here is an example from his documentation:

```
 loadct,5
 im=dist(128,128)
 x=findgen(128)
 y=findgen(128)
 pg_plotimage,im,x,y,xrange=[5,100], $
    yrange=[5,100],/xstyle,/ystyle,/xlog
```

That image definitely looks like the log of a dist function!

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: Questions about NG image with (log) axis
Posted by Xin Tao on Fri, 16 Aug 2013 03:28:49 GMT

View Forum Message <> Reply to Message

On Friday, August 16, 2013 1:37:25 AM UTC+8, Chris Torrence wrote:
>
> Hi Xin Tao,
>
>
>
> Well, I think both David's coyote library and IDL's NG are going to have problems with a log axis for an Image. When I try David's code (yes, I have it installed!) the axes "look" correct, but the image is just plotted as a normal image.
>
>
>
> If the X axis were truly a "log" axis, then I would expect the pixels to be stretched out on the left side and get smaller as you moved to the right. For example, try the following:
>
>
>
> x=10.0+dindgen(300)*(100-10.0)/(300-1)
>
> y=dindgen(300)
>
> data = dist(300)
>
> c = contour(data, x, y, /xlog, /fill)
>
>
>
> Notice how the contour plot is more stretched out on the left side. Is this what you want? If so, then you aren't going to be able to use an Image (either from NG or Coyote). You could use a filled contour plot, say with n_levels=50.
>
>
>
> On the other hand, if you don't want a stretched out image, then you might want to take the log10 of your X data first, then do the i=image() and set the xtickvalues/xtickname to get the correct labels. In the meantime, I'll try to fix the Image function for IDL 8.3 so it at least gives the correct labels, even if it doesn't stretch out the pixels.
>
>

>
> Cheers,
>
> Chris
>
> ExelisVIS

Hi Chris,

Yes, I want a truely "logged" x-axis. In the example I gave, it will give me a stretched image. The problem I have in my mind, though, is to plot a 2D data whose x-axis is defined logarithmically. When I plot the 2D data, I imagine I should be able to do it with: image(data, x, y, /xlog).

I understand I can fix things by using xtickname/xtickvalues stuff, but that sounds too complicated for a simple thing like this, since /xlog is a keyword that has been defined. Look forward to using IDL8.3.

Thanks,

Xin Tao