Subject: Pain with the contour() function
Posted by Helder Marchetto on Wed, 14 Jun 2017 15:09:46 GMT
View Forum Message <> Reply to Message

Hi,
I was trying to plot an image with the contour function and I banged my head against the monitor for a while. I now have the solution and I'm sharing it. Probably most people know this very well. I didn't.
I followed the example given for the colorbar() function (Example: Discrete Contour Levels with Colorbar):
http://www.harrisgeospatial.com/docs/Colorbars.html

So I generated my data with the code below and it appeared strangely shifted to the side.

```
dis = dist(688)
n_levels = 6
levels = findgen(n_levels)
ct_number = 4
ct_indices = bytscl(levels)
loadct, ct_number, rgb_table=ct, /silent
step_ct = congrid(ct[ct_indices, *], 256, 3)
dis = (n_levels-1)*dis/max(dis)
ii = contour(dis, c_value = levels, rgb_table = step_ct, rgb_indices = ct_indices, /fill, axis_style=0)
```

I then started playing around with the position and margin keywords, but had no luck. Finally it all comes down to using xRange and yRange (or xStyle=1, yStyle=1):

```
ii = contour(dis, c_value = levels, rgb_table = step_ct, rgb_indices = ct_indices, /fill, axis_style=0, xStyle=1, yStyle=1)
```

The reason is that contour() plots images as if they were plots, so it defines some axis around it and you have to make sure you're not having uncovered regions.

Well, back to work.

Cheers,
Helder

Subject: Re: Pain with the contour() function
Posted by Markus Schmassmann on Wed, 14 Jun 2017 15:45:35 GMT
View Forum Message <> Reply to Message

On 06/14/2017 05:09 PM, Helder wrote:
> I was trying to plot an image with the contour function and I banged my head against the monitor for a while. I now have the solution and I'm sharing it. Probably most people know this very well. I didn't.
> I followed the example given for the colorbar() function (Example: Discrete Contour Levels with

Colorbar):
> http://www.harrisgeospatial.com/docs/Colorbars.html
>
> So I generated my data with the code below and it appeared strangely shifted to the side.
>
> dis = dist(688)
> n_levels = 6
> levels = findgen(n_levels)
> ct_number = 4
> ct_indices = bytscl(levels)
> loadct, ct_number, rgb_table=ct, /silent
> step_ct = congrid(ct[ct_indices, *], 256, 3)
> dis = (n_levels-1)*dis/max(dis)
> ii = contour(dis, c_value = levels, rgb_table = step_ct, rgb_indices = ct_indices, /fill, axis_style=0)
>
> I then started playing around with the position and margin keywords, but had no luck. Finally it all comes down to using xRange and yRange (or xStyle=1, yStyle=1):
>
> ii = contour(dis, c_value = levels, rgb_table = step_ct, rgb_indices = ct_indices, /fill, axis_style=0, xStyle=1, yStyle=1)
>
> The reason is that contour() plots images as if they were plots, so it defines some axis around it and you have to make sure you're not having uncovered regions.
>
> Well, back to work.

another pain with contour, or any combination of raster graphics and
vector graphics elements is, that they are natively offset by half a
pixel (or whatever you want to call the data unit here):

```
 c=contour(dist(10),overplot=image(35*dist(10),dimension=[250 ,250], $
     position=[25,25,225,225],/dev))
```

the solution to that is

```
contour, dist(10), path_xy=line, path_info=info, /path_data_coord, $
     closed=0,levels=[0:6]
i1=image(35*dist(10),dimension=[250,250], $
     position=[25,25,225,225],/dev)
foreach in,info do p=plot(line[*,in.offset+[lindgen(in.n),in.type ? 0 :$
     !null]]+.5, color=255b-[40b,40b,0b]*byte(in.value),overplot=i1)
```

-- Markus

Subject: Re: Pain with the contour() function

Posted by lecacheux.alain on Thu, 15 Jun 2017 08:56:30 GMT

View Forum Message <> Reply to Message

Le mercredi 14 juin 2017 17:45:39 UTC+2, Markus Schmassmann a écrit :

```
>  On 06/14/2017 05:09 PM, Helder wrote:
>>  I was trying to plot an image with the contour function and I banged my head against the monitor for a while. I now have the solution and I'm sharing it. Probably most people know this very well. I didn't.
>>  I followed the example given for the colorbar() function (Example: Discrete Contour Levels with Colorbar):
>>  http://www.harrisgeospatial.com/docs/Colorbars.html
>>
>>  So I generated my data with the code below and it appeared strangely shifted to the side.
>>
>>  dis = dist(688)
>>  n_levels = 6
>>  levels = findgen(n_levels)
>>  ct_number = 4
>>  ct_indices = bytscl(levels)
>>  loadct, ct_number, rgb_table=ct, /silent
>>  step_ct = congrid(ct[ct_indices, *], 256, 3)
>>  dis = (n_levels-1)*dis/max(dis)
>>  ii = contour(dis, c_value = levels, rgb_table = step_ct, rgb_indices = ct_indices, /fill, axis_style=0)
>>
>>  I then started playing around with the position and margin keywords, but had no luck. Finally it all comes down to using xRange and yRange (or xStyle=1, yStyle=1):
>>
>>  ii = contour(dis, c_value = levels, rgb_table = step_ct, rgb_indices = ct_indices, /fill, axis_style=0, xStyle=1, yStyle=1)
>>
>>  The reason is that contour() plots images as if they were plots, so it defines some axis around it and you have to make sure you're not having uncovered regions.
>>
>>  Well, back to work.
>
>  another pain with contour, or any combination of raster graphics and
>  vector graphics elements is, that they are natively offset by half a
>  pixel (or whatever you want to call the data unit here):
>
>   c=contour(dist(10),overplot=image(35*dist(10),dimension=[250 ,250], $
>       position=[25,25,225,225],/dev))
>
>  the solution to that is
>
>  contour, dist(10), path_xy=line, path_info=info, /path_data_coord, $
>       closed=0,levels=[0:6]
>  i1=image(35*dist(10),dimension=[250,250], $
>       position=[25,25,225,225],/dev)
>  foreach in,info do p=plot(line[*,in.offset+[lindgen(in.n),in.type ? 0 :$
>      !null]]+.5, color=255b-[40b,40b,0b]*byte(in.value),overplot=i1)
>
>  -- Markus
```

I would rather write:

```
 c=contour(dist(10),overplot=image(35*dist(10),dimension=[250 ,250], $
     IMAGE_LOCATION=[-0.5,-0.5], $
     position=[25,25,225,225],/dev))
```

since, by convention, the image grid locates the lower left point of each pixel.

alx.

Subject: Re: Pain with the contour() function
Posted by Helder Marchetto on Thu, 15 Jun 2017 08:57:02 GMT

View Forum Message <> Reply to Message

On Wednesday, June 14, 2017 at 5:45:39 PM UTC+2, Markus Schmassmann wrote:
> On 06/14/2017 05:09 PM, Helder wrote:
>> I was trying to plot an image with the contour function and I banged my head against the monitor for a while. I now have the solution and I'm sharing it. Probably most people know this very well. I didn't.
>> I followed the example given for the colorbar() function (Example: Discrete Contour Levels with Colorbar):
>> http://www.harrisgeospatial.com/docs/Colorbars.html
>>
>> So I generated my data with the code below and it appeared strangely shifted to the side.
>>

```
>>  dis = dist(688)
>>  n_levels = 6
>>  levels = findgen(n_levels)
>>  ct_number = 4
>>  ct_indices = bytscl(levels)
>>  loadct, ct_number, rgb_table=ct, /silent
>>  step_ct = congrid(ct[ct_indices, *], 256, 3)
>>  dis = (n_levels-1)*dis/max(dis)
>>  ii = contour(dis, c_value = levels, rgb_table = step_ct, rgb_indices = ct_indices, /fill, axis_style=0)
>>
```

>> I then started playing around with the position and margin keywords, but had no luck. Finally it all comes down to using xRange and yRange (or xStyle=1, yStyle=1):
>>

```
>>  ii = contour(dis, c_value = levels, rgb_table = step_ct, rgb_indices = ct_indices, /fill, axis_style=0, xStyle=1, yStyle=1)
>>
```

```
>>  The reason is that contour() plots images as if they were plots, so it defines some axis around
it and you have to make sure you're not having uncovered regions.
>>
>>  Well, back to work.
>
>  another pain with contour, or any combination of raster graphics and
>  vector graphics elements is, that they are natively offset by half a
>  pixel (or whatever you want to call the data unit here):
>
>   c=contour(dist(10),overplot=image(35*dist(10),dimension=[250 ,250], $
>       position=[25,25,225,225],/dev))
>
>  the solution to that is
>
>  contour, dist(10), path_xy=line, path_info=info, /path_data_coord, $
>       closed=0,levels=[0:6]
>  i1=image(35*dist(10),dimension=[250,250], $
>       position=[25,25,225,225],/dev)
>  foreach in,info do p=plot(line[*,in.offset+[lindgen(in.n),in.type ? 0 :$
>       !null]]+.5, color=255b-[40b,40b,0b]*byte(in.value),overplot=i1)
>
>  -- Markus
```

Hi,
I see your problem. This seems to have been discussed here:
https://groups.google.com/d/msg/comp.lang.idl-pvwave/8mTxEIM evzE/RIplXmvm9N4J

Your solution works, but at the cost of losing the contour function managing all the contour lines at once.

Helder