Subject: Drawing vector fields with New Graphics
Posted by Gordon Farquharson on Mon, 01 Jul 2013 21:34:09 GMT
View Forum Message <> Reply to Message

Is there a way to plot (using New Graphics) two different vector fields on the same set of axes such that the vector fields have the same scaling?

Below is a minimal working program. What I want (and sort of expect) is that the v2 vectors be proportionally scaled with respect to the v1 vectors. What I get is that the v2 vectors appear larger than the v1 vectors, even though they are clearly smaller.

```
PRO test_vector

   x = [0.,1.,2.]
   y = [0.,0.,0.]
   vx = [1.,1.,1.]
   vy = [1.,1.,1.]

   v1 = vector(vx, vy, x, y, $
            XTITLE='X', YTITLE='Y', $
            XRANGE=[0.,4.], YRANGE=[0.,4.])

   v1.arrow_thick = 2
   v1.length_scale = 2

   stop

   x = [1.,2.]
   y = [1.,1.]
   vx = [-0.5,-0.5]
   vy = [-0.5,-0.5]

   v2 = vector(vx, vy, x, y, $
            /OVERPLOT)

   v2.arrow_thick = 2
   v2.length_scale = 2

END
```

Gordon

Subject: Re: Drawing vector fields with New Graphics
Posted by Phillip Bitzer on Tue, 02 Jul 2013 14:31:46 GMT
View Forum Message <> Reply to Message

This doesn't help with your immediate problem, but part of the reason v1 appears smaller is your axis scale. See what I mean with these two commands:

```
IDL> v1.yrange=[-1, 4]
IDL> v1.xrange=[-1, 4]
```

This is because of what (relative to the vector) the x,y locations refer to. The default is the middle of the arrow. (You could also try v1.data_location=0 / v2.data_location=0 to "see" the same thing.)

This note in the help scares me:
Note: The units for the U and V components are arbitrary, and are not related to the units of the X and Y coordinates. The VECTOR function will compute a default length scale and head size based upon the overall number of vectors and the average magnitude of the vectors. The LENGTH_SCALE and HEAD_SCALE properties may be used to change the default size.

Intuitively (at least to me), this should not be the behavior. Uv, Ux should mean *something* about the magnitude.

From my reading of this note, it looks like you could find the average magnitude of the vectors in v1, which sets the default scale, and then scale the others (v2, etc) accordingly relative to this value.

Subject: Re: Drawing vector fields with New Graphics
Posted by David Fanning on Tue, 02 Jul 2013 14:38:49 GMT

View Forum Message <> Reply to Message

Phillip Bitzer writes:

> This note in the help scares me:
> Note: The units for the U and V components are arbitrary, and are not related to the units of the X and Y coordinates. The VECTOR function will compute a default length scale and head size based upon the overall number of vectors and the average magnitude of the vectors. The LENGTH_SCALE and HEAD_SCALE properties may be used to change the default size.
>
> Intuitively (at least to me), this should not be the behavior. Uv, Ux should mean *something* about the magnitude.

This should scare ALL of us! Good God.

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: Drawing vector fields with New Graphics
Posted by Gordon Farquharson on Tue, 02 Jul 2013 17:54:36 GMT
View Forum Message <> Reply to Message

Hi Philip

On Tuesday, July 2, 2013 7:31:46 AM UTC-7, Phillip Bitzer wrote:
> This doesn't help with your immediate problem, but part of the reason v1 appears smaller is your axis scale. See what I mean with these two commands:
>
>
>
> IDL> v1.yrange=[-1, 4]
>
> IDL> v1.xrange=[-1, 4]
>
>
>
> This is because of what (relative to the vector) the x,y locations refer to. The default is the middle of the arrow. (You could also try v1.data_location=0 / v2.data_location=0 to "see" the same thing.)

Ah! Good point. I picked bad example vectors for the displayed axis range.

> This note in the help scares me:
>
> Note: The units for the U and V components are arbitrary, and are not related to the units of the X and Y coordinates. The VECTOR function will compute a default length scale and head size based upon the overall number of vectors and the average magnitude of the vectors. The LENGTH_SCALE and HEAD_SCALE properties may be used to change the default size.

I'm not so scared by this note. The length of a vector in the vector field doesn't have anything to do with the spatial coordinates in which the vector field is plotted. One typically scales vector lengths to make the vector field pretty (easy to read). As I understand it, this automatic scaling is what the vector routine is doing quite conveniently, so I can see that this automatic scaling is useful for a single vector field because it makes the vector field fit in the area nicely. But the second vector field should use the same scaling factor, or at least, there should be an option to reuse the scaling factor computed form the first vector field.

> From my reading of this note, it looks like you could find the average magnitude of the vectors in v1, which sets the default scale, and then scale the others (v2, etc) accordingly relative to this value.

Did you mean something like:

```
PRO test_vector

   x = [0.,1.,2.]
   y = [0.,0.,0.]
   vx = [1.,1.,1.]
   vy = [1.,1.,1.]

   vmag = mean(sqrt(vx^2 + vy^2))

   v1 = vector(vx, vy, x, y, $
            XTITLE='X', YTITLE='Y', $
            XRANGE=[-1.,4.], YRANGE=[-1.,4.])

   v1.arrow_thick = 2
   v1.length_scale = 2

   x = [1.,2.]
   y = [1.,1.]
   vx = [-0.5,-0.5]
   vy = [-0.5,-0.5]

   vx /= vmag
   vy /= vmag

   v2 = vector(vx, vy, x, y, $
            /OVERPLOT, XRANGE=[-1.,4.], YRANGE=[-1,4.])

   v2.arrow_thick = 2
   v2.length_scale = 2

END
```

This version produces vectors in v1 and v2 that are the same length! The scaling does not seemed to be maintained between calls to vector.

Exelis: is this something that can be fixed easily and released as an out of band patch? It is really frustrating and significantly reduces the usefulness of the vector routine. (I'm trying to be a good IDL citizen, and give the new graphics a chance, but I do have a copy of David's graphics book on my bookshelf...)

Gordon

Subject: Re: Drawing vector fields with New Graphics
Posted by David Fanning on Tue, 02 Jul 2013 18:10:29 GMT
View Forum Message <> Reply to Message

Gordon Farquharson writes:

>
> Hi Philip
>
> On Tuesday, July 2, 2013 7:31:46 AM UTC-7, Phillip Bitzer wrote:
>> This doesn't help with your immediate problem, but part of the reason v1 appears smaller is your axis scale. See what I mean with these two commands:
>>
>>
>>
>> IDL> v1.yrange=[-1, 4]
>>
>> IDL> v1.xrange=[-1, 4]
>>
>>
>>
>> This is because of what (relative to the vector) the x,y locations refer to. The default is the middle of the arrow. (You could also try v1.data_location=0 / v2.data_location=0 to "see" the same thing.)
>
> Ah! Good point. I picked bad example vectors for the displayed axis range.
>
>> This note in the help scares me:
>>
>> Note: The units for the U and V components are arbitrary, and are not related to the units of the X and Y coordinates. The VECTOR function will compute a default length scale and head size based upon the overall number of vectors and the average magnitude of the vectors. The LENGTH_SCALE and HEAD_SCALE properties may be used to change the default size.
>
> I'm not so scared by this note. The length of a vector in the vector field doesn't have anything to do with the spatial coordinates in which the vector field is plotted. One typically scales vector lengths to make the vector field pretty (easy to read). As I understand it, this automatic scaling is what the vector routine is doing quite conveniently, so I can see that this automatic scaling is useful for a single vector field because it makes the vector field fit in the
area nicely. But the second vector field should use the same scaling factor, or at least, there should be an option to reuse the scaling factor computed form the first vector field.
>
>> From my reading of this note, it looks like you could find the average magnitude of the vectors in v1, which sets the default scale, and then scale the others (v2, etc) accordingly relative to this value.
>
> Did you mean something like:
>
> PRO test_vector
>
>   x = [0.,1.,2.]
>   y = [0.,0.,0.]
>   vx = [1.,1.,1.]

```
>     vy = [1.,1.,1.]
>
>     vmag = mean(sqrt(vx^2 + vy^2))
>
>     v1 = vector(vx, vy, x, y, $
>               XTITLE='X', YTITLE='Y', $
>               XRANGE=[-1.,4.], YRANGE=[-1.,4.])
>
>     v1.arrow_thick = 2
>     v1.length_scale = 2
>
>     x = [1.,2.]
>     y = [1.,1.]
>     vx = [-0.5,-0.5]
>     vy = [-0.5,-0.5]
>
>     vx /= vmag
>     vy /= vmag
>
>     v2 = vector(vx, vy, x, y, $
>               /OVERPLOT, XRANGE=[-1.,4.], YRANGE=[-1,4.])
>
>     v2.arrow_thick = 2
>     v2.length_scale = 2
>
>  END
>
```

> This version produces vectors in v1 and v2 that are the same length! The scaling does not seemed to be maintained between calls to vector.

Don't you want this:

```
  v2.length_scale = vi.length_scale
```

That seems to give the right result.

> *Exelis*: is this something that can be fixed easily and released as an out of band patch? It is really frustrating and significantly reduces the usefulness of the vector routine. (I'm trying to be a good IDL citizen, and give the new graphics a chance, but I do have a copy of David's graphics book on my bookshelf...)

No one acknowledges this, of course, but I recall this line from my
favorite story in Barry Lopez's collection, River Notes:

"As dear and coyote sipped from the same tiny pool they abrogated
their agreement, and the deer contemplated the loss of the coyote
as he would the lose of a friend; for the enemy, like the friend,
made you strong."

I like to think "fixed easily and released as a patch" is a response to
similar service along these graphics lines. ;-)

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: Drawing vector fields with New Graphics
Posted by Phillip Bitzer on Tue, 02 Jul 2013 18:15:00 GMT

View Forum Message <> Reply to Message

On Tuesday, July 2, 2013 12:54:36 PM UTC-5, Gordon Farquharson wrote:

> I'm not so scared by this note. The length of a vector in the vector field doesn't have anything to do with the spatial coordinates in which the vector field is plotted. One typically scales vector lengths to make the vector field pretty (easy to read). As I understand it, this automatic scaling is what the vector routine is doing quite conveniently, so I can see that this automatic scaling is useful for a single vector field because it makes the vector field fit in the area nicely. But the second vector field should use the same scaling factor, or at least, there should be an option to reuse the scaling factor computed form the first vector field.
>

Chacun a son gout, I guess. I would like to have some sort of control of the length of the vectors, using vx, vy.

```
>
>  Did you mean something like:
>
>     vmag = mean(sqrt(vx^2 + vy^2))
>
>
>     vx /= vmag
>     vy /= vmag
>
>     v2.length_scale = 2
>
```

> This version produces vectors in v1 and v2 that are the same length! The scaling does not seemed to be maintained between calls to vector.
>

Not exactly what I was thinking. Defining v1 sets the "units" of the length_scale. You then set the "unit" to be 2 tics with v1.length_scale=2

If instead of dividing the vx, vy by the magnitude, instead try

v2.length_scale = 2/vmag

I think this is what you're looking for....at least for this example.

Subject: Re: Drawing vector fields with New Graphics
Posted by David Fanning on Tue, 02 Jul 2013 18:32:30 GMT

View Forum Message <> Reply to Message

David Fanning writes:

> Don't you want this:
>
> v2.length_scale = vi.length_scale
>
> That seems to give the right result.

Humm. Can't reproduce that result now. :-(

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: Drawing vector fields with New Graphics
Posted by David Fanning on Tue, 02 Jul 2013 18:40:58 GMT

View Forum Message <> Reply to Message

Phillip Bitzer writes:

> Chacun a son gout, I guess. I would like to have some sort of control of the length of the vectors, using vx, vy.

In cgMapVector, I choose the default length of the vector to be 1/100 of the distance of the plot in the X direction. This causes the maximum length of the vector to be no more than the default length times SQRT (2). The length is user-configurable, but has some connection to the coordinate system in any case. In cgMapVector the length is usually expressed in meters and gives you a convenient way to write a vector legend on the map.

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: Drawing vector fields with New Graphics
Posted by Gordon Farquharson on Tue, 02 Jul 2013 19:08:37 GMT

View Forum Message <> Reply to Message

On Tuesday, July 2, 2013 11:15:00 AM UTC-7, Phillip Bitzer wrote:
> On Tuesday, July 2, 2013 12:54:36 PM UTC-5, Gordon Farquharson wrote:
> Chacun a son gout, ...

I have no idea how to pronounce it, but I like it :-)

> Not exactly what I was thinking. Defining v1 sets the "units" of the length_scale. You then set the "unit" to be 2 tics with v1.length_scale=2
>
> If instead of dividing the vx, vy by the magnitude, instead try
>
> v2.length_scale = 2/vmag
>
> I think this is what you're looking for....at least for this example.

Perfect! It works. Thanks very much for the suggestion. For reference (for others), here is the updated program.

```
PRO test_vector

   x = [0.,1.,2.]
   y = [0.,0.,0.]
   vx = [1.,1.,1.]
   vy = [1.,1.,1.]
   vmag = mean(sqrt(vx^2 + vy^2))

   v1 = vector(vx, vy, x, y, $
            XTITLE='X', YTITLE='Y', $
            XRANGE=[-1.,4.], YRANGE=[-1.,4.])

   v1.arrow_thick = 2
   v1.length_scale = 2

   x = [1.,2.]
   y = [1.,1.]
   vx = [-0.5,-0.5]
   vy = [-0.5,-0.5]

   v2 = vector(vx, vy, x, y, $
            /OVERPLOT, XRANGE=[-1.,4.], YRANGE=[-1.,4.])

   v2.arrow_thick = 2
   v2.length_scale = 2. / vmag

END
```

Thanks for the help.

(Maybe Mark or Chris could still comment on whether vector could be improved to provide a more intuitive interface.)

Gordon

Subject: Re: Drawing vector fields with New Graphics
Posted by Mark Piper on Thu, 04 Jul 2013 20:20:07 GMT

View Forum Message <> Reply to Message

Chris & I started looking at this yesterday, but we didn't reach any conclusions. We're both out for the next week, but we'll take it up again when we get back.

Also, I got in a little trouble for posting code on the newsgroup (sigh). So, any updates might have to wait for a hotfix.

mp