
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 from 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', $
>               X RANGE=[-1.,4.], Y RANGE=[-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, X RANGE=[-1.,4.], Y RANGE=[-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 = v1.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 loss 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.")
