Subject: Re: Flow3 procedure and WHERE
Posted by David Fanning on Tue, 25 May 2004 00:03:05 GMT
View Forum Message <> Reply to Message

Adhara writes:

> Hello Dr.Fanning, Thanks for effort in trying to understand my
> program!However, let me clearify some points:
>
> - Yes, my velocity vectors are three dimensional. Meaning that Vx is
> the component of the velocity in the x-direction, Vy is the component
> of the velocity in the y-direction, and Vz is the component of the
> velocity in the z-direction. I expect IDL to plot the resultant of
> these three components starting at the coordiates (sx,sy,sz).

Well, then you might try making your S vectors the same size
as your V vectors and subsetting them with the same index you
are using to find all the non-zero values:

```
index = Where((vx ne 0) AND (vy ne 0) AND (vz ne 0), count)
vxx = vx[index]
sxx = sx[index]
```

Note that your arrays are floating point, so instead of
trying to find those values that *exactly* equal zero you
might have to find values that are "pretty close" to zero.

> - I think I will get them to have the same size because Vx,Vy,Vz have
> a value different to zero, all at the same time and at the same
> location. So probably FlOW3 will like it!!

May God grant you long life and fabulous wealth!
I mean, yes, maybe. :-)

> - I also have the exact location at which the vector starts, therefore
> I rather use it as input in FLOW3. My problem has been to *extract*
> these Sx, Sy, Sz coordinates at which Vx,Vy,Vz are different from
> zero. These location vectors will also have the same size as the
> velocity vectors.

OK, this sounds like a plan.

> -From the 32400 data points that I have, only 521 have data different
> from zero. The velocity data is very small, and as you suggested I am
> already using a factor to increase those values. However, this factor
> can not be greater than 20 because:
>
> % Program caused arithmetic error: Floating divide by 0

> % Program caused arithmetic error: Floating overflow
> % Program caused arithmetic error: Floating illegal operand

Humm. Hard to see how this could happen by scaling a small
value by a large value, unless you were using a HUMONGOUS
value! Probably an error in your algorithm, I think.

> - Do my comments clearify my ideas to you?

Uh, well, I'm pretty dense when it comes to this kind of thing. :-)

> I tried to do it as well in 2D as you said. However, I have a simple
> question because I am having an error according to the command
> arguments, due to the size of U and V using VELOVECT.
>
> I am using U as one dimensional vector with Vx, and V as one
> dimensional vector with Vy. What does it mean that it "must be a
> two-dimensional array".??

Well, it means the arguments must have two dimensions.
Humm, let's see, how would I explain this? Suppose
you have a three dimensional array:

```
IDL> array = Findgen(4, 3, 5)
IDL> Help, array
  ARRAY          FLOAT     = Array[4, 3, 5]
```

This maybe represents XYZ values. If I want a 2D
slice in the Z direction, I might do this:

```
IDL> slice = Reform(array[*,*,2])
IDL> Help, slice
  SLICE          FLOAT     = Array[4, 3]
```

If I want a vector (a 1D array) from the slice:

```
IDL> vector = Reform(slice[*,1])
IDL> Help, vector
  VECTOR         FLOAT     = Array[4]
```

Just for fun, I'd try something like this:

```
VELOCECT, Reform(VX[*,*,40]), Reform(VY[*,*,40])
```

Or contour those variables. I don't know. *Something*
should work.

Cheers,

David

--
David Fanning, Ph.D.
Fanning Software Consulting, Inc.
Coyote's Guide to IDL Programming: http://www.dfanning.com/
