Subject: Re: Strange array division problem Posted by cgguido on Mon, 04 May 2009 23:35:06 GMT

View Forum Message <> Reply to Message

On May 4, 4:42 pm, David Klassen <klas...@rowan.edu> wrote:

- > I'm trying to create an array by reading in 2-d data and putting into
- > planes in a 3-d array, however, some of the data is redundant---the 2-
- > d data would go into the same plane. In these cases, I want to
- > average them so I figure I can just add the data to the current plane
- > value, keeping track of how many data arrays go into each plane, then
- > just divide the final 3-d array by these counts. But I'm stuck on how
- > exactly to implement that.

>

- > My 3-d array X columns, Y rows, Z planes and the 2-d array is Z
- > columns by Y rows (so I'm "rotating" the data and "sliding" each one
- > into a column of the 3-d array---I hope that makes sense). I then
- > have a vector, xcounts, that is X elements long and as data go into
- > the columns, I increment xcounts[X].

>

- > So, when I'm done populating the 3-d array I need to divide each row
- > in each plane by the vector xcounts. Is there an easy way to do this
- > that doesn't involve me looping through all the points?

how about:

a3d /= rebin(xcounts,z,y,x)

Subject: Re: Strange array division problem
Posted by cgguido on Mon, 04 May 2009 23:52:22 GMT
View Forum Message <> Reply to Message

```
> how about:
```

>

> a3d /= rebin(xcounts,z,y,x)

oops:

a3d /= transpose(rebin(xcounts, x, y, z), [1,2,0])

depending on how you do things, you may have to switch the 'y' and 'z' and the '1' and '2'

a=indgen(5) help, transpose(rebin(a, 5, 10, 10), [2, 1, 0]) ;<Expression> INT = Array[10, 10, 5]

```
Subject: Re: Strange array division problem
Posted by David Klassen on Tue, 05 May 2009 12:44:09 GMT
View Forum Message <> Reply to Message
```

Subject: Re: Strange array division problem
Posted by David Fanning on Tue, 05 May 2009 13:07:59 GMT
View Forum Message <> Reply to Message

Dave Klassen writes:

- > This seems to do it (well, without the transpose---I assume because
- > IDL thinks of standard vectors as columns...?).

No, IDL does not think of standard vectors as columns.

Cheers,

David

P.S. I'd give that result an extra pair of eyeballs before you turn it in for a final grade. :-)

David Fanning, Ph.D.

Fanning Software Consulting, Inc.

Coyote's Guide to IDL Programming: http://www.dfanning.com/

Sepore ma de ni thui. ("Perhaps thou speakest truth.")

Subject: Re: Strange array division problem Posted by cgguido on Tue, 05 May 2009 21:23:13 GMT

View Forum Message <> Reply to Message

```
On May 5, 8:07 am, David Fanning <n...@dfanning.com> wrote:
```

- > P.S. I'd give that result an extra pair of eyeballs
- > before you turn it in for a final grade. :-)

yes yes... especially because I can get dimensions mixed up faster then I can say...uhmm... transpose.

Subject: Re: Strange array division problem Posted by David Klassen on Thu, 07 May 2009 02:03:39 GMT View Forum Message <> Reply to Message

On May 5, 9:07 am, David Fanning <n...@dfanning.com> wrote:

- > Dave Klassen writes:
- >> This seems to do it (well, without the transpose---I assume because
- >> IDL thinks of standard vectors as columns...?).

> No, IDL does not think of standard vectors as columns.

```
Hmmm...really?
IDL> a=[[2,1],[3,4]]
IDL> print,a
    2
          1
    3
          4
IDL> v=[5,6]
IDL> print,v
    5
IDL> print,a##v
```

16

39

Which leads me to my original thought.

But, then again: IDL> print, a#v

28 29

So I guess IDL figures a vector is whatever it need to be...?

Interestingly:

IDL> print,a##transpose(v)

16

39

But:

IDL> print,a#transpose(v)

% Operands of matrix multiply have incompatible dimensions: A,

<INT

Array[1, 2]>.

% Execution halted at: \$MAIN\$

So that's confusing.

- > P.S. I'd give that result an extra pair of eyeballs
- > before you turn it in for a final grade. :-)

Well, I didn't get extra eyes, but I did look long and hard, several times, at the output and it looks like what I expect.

Subject: Re: Strange array division problem
Posted by David Fanning on Thu, 07 May 2009 03:08:59 GMT
View Forum Message <> Reply to Message

Dave Klassen writes:

```
> Hmmm...really?
> IDL> a=3D[[2,1],[3,4]]
> IDL> print,a
      2
            1
>
      3
> IDL> v=3D[5,6]
> IDL> print,v
      5
> IDL> print,a##v
        16
        39
> Which leads me to my original thought.
> But, then again:
> IDL> print,a#v
        28
                 29
> So I guess IDL figures a vector is whatever it need to be...?
```

I think a thorough reading the the Dimensional Juggling and Array Concatenation tutorials should clear up any confusion:

http://www.dfanning.com/tips/rebin_magic.html http://www.dfanning.com/tips/array_concatenation.html Cheers,

David

David Fanning, Ph.D. Fanning Software Consulting, Inc. Coyote's Guide to IDL Programming: http://www.dfanning.com/ Sepore ma de ni thui. ("Perhaps thou speakest truth.")