Subject: Re: dimension problem in matrix multiplication
Posted by Craig Markwardt on Tue, 13 Apr 2004 21:58:26 GMT
View Forum Message <> Reply to Message

Benjamin Hornberger <benjamin.hornberger@stonybrook.edu> writes:

> Hi all,
>
> I have a 3-dimensional data array "image" (n_cols, n_rows, n_data) where
> cols and rows denote pixels in a 2d-image and data are different data
> values for each pixel. Now I would like to perform a matrix
> multiplication so that the data vector (n_data elements) for each pixel
> is multiplied by a 2-dimensional matrix "c".
>
> If I try result = image # c, I get an "incompatible dimensions" error
> since obviously IDL doesn't know that the 3rd dimension in "image" is
> the vector I want to multiply by the matrix, and I want that operation
> done for every pixel. Playing around with * and 0:n_data-1 subscripts
> didn't help so far.

...

Greetings, I think you won't be able to get the '#' or '##' operators
to work very nicely with 3D matrices.

It may be possible to:

1. Do it yourself with FOR loops and TOTAL(), which may not be too bad if you can slice it right. Example, you can remove your original inner loop by doing,

```
    cji = rebin(reform(C(*,I), 1, 1, n_data), n_cols, n_rows, n_data)
    result(*,*,I) = total(image*cji, 3)
```

   This probably has the fewest total loop iterations. Of course you need twice the storage of IMAGE.

2. Loop through each pixel, and compute the matrix product explicitly

```
    image_vec = reform(image(I,J,*))
    result(I,J,*) = c # image_vec
```

   Where I probably have screwed up the # vs ## thing.

Good luck,
Craig

Craig

--
 -- --------------

Craig B. Markwardt, Ph.D. EMAIL: craigmnet@REMOVEcow.physics.wisc.edu
Astrophysics, IDL, Finance, Derivatives | Remove "net" for better response
-- --------------