
Subject: [Fwd: tensor multiplication]

Posted by [Paul van Delst](#) on Thu, 18 May 2000 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

A reply I got from William Thompson that allows tensor mult. in a real compact form.

William Thompson wrote:

>

> In comp.lang.idl-pvwave you write:

>

>> Daniel Luebbert wrote:

>>>

>>> Hi,

>>>

>>> does anybody out there know an efficient and elegant way (i.e., without
>>> for-loops)

>>> to implement a tensor multiplication in IDL?

>>>

>>> What I mean is this:

>>> IDL can do a matrix multiplication, e.g. if I do

>>> c = indgen(3,4)

>>> d = indgen(4)

>>> then for

>>> help, c#d

>>> I get

>>> LONG ARRAY[3],

>>> and that's what I expect.

>>>

>>> But now, when I take one more dimension, like

>>> c = indgen(2,3,4)

>>> d = indgen(4)

>>> then

>>> help, c#d

>>> gives an error! (incompatible matrix dimensions...).

>>> What I would like to get is obviously an

>>> ARRAY[2,3]

>>>

>>> Does anybody know how?

>

<...my initial reply snipped...>

>

> I just sent a reply by mail to Daniel, suggesting

>

> result = reform(reform(c,6,4) # d, 2, 3)

Cool huh? I figured something like this could be done but my knowledge of how you can manipulate matrices and still get the same answer isn't too hot.

>
> Unfortunately, it appears that we still can't post messages to the outside
> world.
>
>> p.s. Can someone explain to me the utility/need for having both the #
>> *and* ## operator? I understand their operation but why both?
>> Convenience? Performance?
>
> I think the answer is more along the lines "historical" and "compromise".
>
> Older versions of IDL only had the # operator, and was described as the matrix
> multiplication operator. Somewhere along the line in the not too distant
> past--I think it was version 4.0--somebody decided to add in the ## operator,
> and the newer documentation says that this is the proper matrix multiplication,
> not the older #. I think this had something to do with the incorporation of
> some linear algebra routines from Numerical Recipes. I also think it may have
> come in part from a desire to make IDL more C-like--originally, IDL was
> modelled after FORTRAN.
>
> William Thompson

--

Paul van Delst Ph: (301) 763-8000 x7274
CIMSS @ NOAA/NCEP Fax: (301) 763-8545
Rm.202, 5200 Auth Rd. Email: pvandelst@ncep.noaa.gov
Camp Springs MD 20746
