
Subject: tensor multiplication

Posted by [Daniel Luebbert](#) on Wed, 17 May 2000 07:00:00 GMT

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

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 a would like to get is obviously an

```
ARRAY[2,3]
```

Does anybody know how?

Daniel

Daniel Luebbert

luebbert@slac.stanford.edu

Subject: Re: tensor multiplication

Posted by [Daniel Luebbert](#) on Thu, 18 May 2000 07:00:00 GMT

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

Hi,

ok, thanks for the various answers.

In case anybody should be interested in the future,

I'll leave here a little program which I have now written
to perform a general multiplication of a
[N_1, N_2, ..., N_n]-array with a N_n -vector.
The result is a [N_1, N_2, ... , N_(n-1)]-array:

```
function tensor_product,a,b

sa = size(a)
a_matrix = reform(a,total_product(sa[1:sa[0]-1]),sa[sa[0]])
result = a_matrix # b
result = reform(temporary(result),sa[1:sa[0]-1])

return,result
end
```

Daniel

Daniel Luebbert
Stanford Synchrotron Radiation Laboratory (SSRL)
luebbert@slac.stanford.edu

Subject: Re: tensor multiplication
Posted by [Craig Markwardt](#) on Thu, 18 May 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

Paul van Delst <pvandelst@ncep.noaa.gov> writes:

```
> 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?
>>
>> ...
>> 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 a would like to get is obviously an
```

```

>>     ARRAY[2,3]
>>
>> Does anybody know how?
>
> Great question! I have always only thought about 2D matrices, but why
> should nD be any different? Maybe when IDL says "matrix" is really means
> 2-D array?
>
> how about
>
> sz = size(c)
> e = INTARR( sz(1), sz(2) )
> FOR i = 0, sz(1) - 1 DO BEGIN
>   tmp_e = REFORM( c[i,*,*] ) # d
>   e[i,*] = TEMPORARY( tmp_e )
> ENDFOR

```

I would have done something similar. The point is that FOR loops are reasonable to use if you can do enough processing within one loop iteration. Here, "enough" is an entire matrix multiply.

Yorick, a language similar to IDL, has a cool tensor contraction operator which would fit the bill nicely, or at least make it look more compact.

```

> 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 do occasionally use ## for matrix multiply. I use # more often to expand a vector into an array; there's been discussion here before on which is faster, # or rebin.

Craig

--

```

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

```

Subject: Re: tensor multiplication
 Posted by [Paul van Delst](#) on Thu, 18 May 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

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?

```

Great question! I have always only thought about 2D matrices, but why should nD be any different? Maybe when IDL says "matrix" is really means 2-D array?

how about

```

sz = size(c)
e = INTARR( sz(1), sz(2) )
FOR i = 0, sz(1) - 1 DO BEGIN
    tmp_e = REFORM( c[i,*,*] ) # d
    e[i,*] = TEMPORARY( tmp_e )
ENDFOR

```

I know it is not a great solution (I haven't tested it, just typed it) but something like this should work. If you encapsulate it in it's own function, you would have a general tensor mult. method.

I'm sure the Gumley's, Fanning's, Markwardt's, and JD Smith's of the world will have more elegant answers.

paulv

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?

--

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

Subject: Re: tensor multiplication
Posted by [hcp](#) on Fri, 19 May 2000 07:00:00 GMT
[View Forum Message](#) <> [Reply to Message](#)

In article <onpuqjsqp.fsf@cow.physics.wisc.edu>, Craig Markwardt
<craigmnet@cow.physics.wisc.edu> writes:

|> Yorick, a language similar to IDL, has a cool tensor contraction
|> operator which would fit the bill nicely, or at least make it look
|> more compact.

That's similar as in "it's an interpreted language for scientific
data handling and prototyping" not similar as in "the syntax is
just about the same" As a language it looks very C-like -- not for the
curly-bracket haters.

Yorick is much smaller than IDL and doesn't have geographic maps and
sundry other things. However it is free software and the array syntax is
very sophisticated, the contraction operator being in some ways one of the
better features as Craig points out. It seems fairly stable and behaves well
when you try to actually achieve something with it -- this isn't true
of some similar packages.

Interested parties can point their browsers at

<ftp://ftp-icf.llnl.gov/pub/Yorick/yorick-ad.html>

Hugh

--

=====
=====

Hugh C. Pumphrey	Telephone 0131-650-6026
Department of Meteorology	FAX 0131-650-5780
The University of Edinburgh	Replace 0131 with +44-131 if outside U.K.

=====

=====
