
Subject: Re: matrix multiplication of 2 three-dimensional arrays
Posted by thomas.jagdhuber@dlr on Thu, 21 Aug 2008 10:00:05 GMT
[View Forum Message](#) <> [Reply to Message](#)

On 20 Aug., 21:53, "thomas.jagdhuber" <thomas.jagdhuber...@gmail.com> wrote:
> On 20 Aug., 18:11, Jeremy Bailin <astroco...@gmail.com> wrote:
>
>
>
>> On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhuber...@gmail.com>
>> wrote:
>
>>> Dear experts,
>
>>> I would like to matrix multiply two matrices with dimensions
>>> [3,3,1500]. means: 1500 times a matrix multiplication of 2 matrices
>>> with dimension [3,3]
>>> I could do this with a for loop over the dimension [1500] but i
>>> suppose this is not very elegant. Is there any other way to do this
>>> time-efficient.
>
>>> Best regards,
>
>>> thomas
>
>> My IDL-foo feels strong today... let's see:
>
>> If A and B are 3x3, then C = A ## B is equivalent to:
>
>> q = rebin(indgen(3),3,3)
>> p = rebin(reform(indgen(3),1,3),3,3)
>> c = total(rebin(reform(a[q,p],1,3,3),3,3,3) * reform(b[*],q,3,3,3), 2)
>
>> Adding an extra dimension on the end of that gets tricky, but I think
>> the following should work if A and B are each 3x3xNMATRIX and you want
>> C to be a 3x3xNMATRIX array where each C[*,* ,i] = A[*,* ,i] ##
>> B[*,* ,i]:
>
>> c = total(rebin(reform((a[q,p,*])[0:2,3*lindgen(3),*],1,3,3,nmatrix),
>> 3,3,3,nmatrix) * reform(b[*],q,*],
>> 3,3,3,nmatrix), 2)
>
>> Incidentally, does anyone have a better way of doing the (a[q,p,*])
>> [0:2,3*lindgen(3),*] bit of it? The problem is that if A, Q and P are
>> each 3x3 then A[Q,P] is 3x3, but if A is 3x3xN then A[Q,P,*] is 9x9xN.
>
>> If I see another reform today, I'm going to scream. ;-)

```
>  
>> -Jeremy.  
>  
> Thank you all, very much!  
> I like of course the code of jeremy most, but i think paolo's  
> suggestions is the most efficient one. But let us see, i will make  
> some investigations...  
>  
> Best regards,  
>  
> thomas
```

Here is another suggestion from a very nice colleague:

```
function matrix_multiply_3, A, B  
;---only for quadratic matrices  
  
s = size(A)  
N2a = s(2)  
N3a = s(3)  
  
s = size(B)  
N1b = s(1)  
  
C = make_array([N1b,N2a,N3a],type=s(4))  
  
for k=0,N2a-1 do begin  
  for n=0,N1b-1 do begin  
  
    C(n,k,*) = total(A(*,k,*) * B(n,*,*),1)  
  
  endfor  
endfor  
  
return, C  
  
end
```

Advantage: Without rebin it can also handle complex numbers

```
best regards,  
thomas
```
