

---

Subject: Re: vector multiplication of a colum-vectors(1col,3row) and a row-vector(3col,1row), but each vector position[col,row] is a 1000x1400 array  
Posted by [Paolo Grigis](#) on Wed, 28 Feb 2007 09:11:27 GMT

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

---

thomas.jagdhuber@dlr.de wrote:

> Hi,  
>  
> I am a rookie in programming IDL. So I try to compute a vector  
> product out of a colum-vectors(1col,3row) and a row-vector(3col,1row),  
> with the specialty that each position in the vectors is an 1000x1400  
> array.  
> vector1=[[array1],[array2],[array3]]  
> vector2=[[array1],[array2],[array3]]  
> matrix2=matrix\_multiply(vector1,vector2,/btranspose)  
> But this is not generating a 3x3 Matrix!  
>  
> Does anyone know anything??

Well, most people at least do know something...  
but maybe you're taking a Socratic stance here ;-)

I think that before asking us how to do whatever it is you want done in IDL, you should try to explain better what it is that you are trying to do in the first place (at the level of algebra, not programming language). It seems to me that you are confusing vector (cross) product with scalar product anyway... and why you want to get 9 numbers out of the 4.2 millions you start with?

Ciao,  
Paolo

>  
> Thank you very much  
>  
> Tom  
>

---

Subject: Re: vector multiplication of a colum-vectors(1col,3row) and a row-vector(3col,1row), but each vector position[col,row] is a 1000x1400 array  
Posted by [thomas.jagdhuber@dlr.](#) on Wed, 28 Feb 2007 11:17:29 GMT

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

---

On 28 Feb., 10:11, Paolo Grigis <pgri...@astro.phys.ethz.ch> wrote:

> thomas.jagdhu...@dlr.de wrote:  
>> Hi,  
>

```
>> I am a rookie in programming IDL. So I try to compute a vector
>> product out of a colum-vectors(1col,3row) and a row-vector(3col,1row),
>> with the specialty that each position in the vectors is an 1000x1400
>> array.
>> vector1=[[array1],[array2],[array3]]
>> vector2=[[array1],[array2],[array3]]
>> matrix2=matrix_multiply(vector1,vector2,/btranspose)
>> But this is not generating a 3x3 Matrix!
>
>> Does anyone know anything??
>
> Well, most people at least do know something...
> but maybe you're taking a Socratic stance here ;-))
>
> I think that before asking us how to do whatever it is you want
> done in IDL, you should try to explain better what it is that
> you are trying to do in the first place (at the level of algebra,
> not programming language). It seems to me that you are confusing
> vector (cross) product with scalar product anyway... and why you
> want to get 9 numbers out of the 4.2 millions you start with?
>
> Ciao,
> Paolo
>
>
>
>> Thank you very much
>
>> Tom
```

I just have 3 Matrices and I have to calculate the conjugate,  
transpose of this matrices and then multiply each by each so I will  
get 9 matrices

```
11* 12* 13*
21* 22* 23*
31* 32* 33*
```

and in the end I want to store this in one big Matrix of matrices.  
So I can do all this with for-loops but I thought may be there is a  
shorter and more elegant way to compute this.  
Sorry, for my incomplete explanation.

tom

---

Subject: Re: vector multiplication of a colum-vectors(1col,3row) and a  
row-vector(3col,1row), but each vector position[col,row] is a 1000x1400 array  
Posted by [Paolo Grigis](#) on Wed, 28 Feb 2007 13:35:18 GMT

thomas.jagdhuber@dlr.de wrote:

> On 28 Feb., 10:11, Paolo Grigis <pgri...@astro.phys.ethz.ch> wrote:

>> thomas.jagdhu...@dlr.de wrote:

>>> Hi,

>>> I am a rookie in programming IDL. So I try to compute a vector  
>>> product out of a colum-vectors(1col,3row) and a row-vector(3col,1row),  
>>> with the specialty that each position in the vectors is an 1000x1400  
>>> array.

>>> vector1=[[array1],[array2],[array3]]

>>> vector2=[[array1],[array2],[array3]]

>>> matrix2=matrix\_multiply(vector1,vector2,/btranspose)

>>> But this is not generating a 3x3 Matrix!

>>> Does anyone know anything??

>> Well, most people at least do know something...

>> but maybe you're taking a Socratic stance here ;-)

>>

>> I think that before asking us how to do whatever it is you want  
>> done in IDL, you should try to explain better what it is that  
>> you are trying to do in the first place (at the level of algebra,  
>> not programming language). It seems to me that you are confusing  
>> vector (cross) product with scalar product anyway... and why you  
>> want to get 9 numbers out of the 4.2 millions you start with?

>>

>> Ciao,

>> Paolo

>>

>>

>>

>>> Thank you very much

>>> Tom

>

> I just have 3 Matrices and I have to calculate the conjugate,  
> transpose of this matrices and then multiply each by each so I will  
> get 9 matrices

> 11\* 12\* 13\*

> 21\* 22\* 23\*

> 31\* 32\* 33\*

> and in the end I want to store this in one big Matrix of matrices.  
> So I can do all this with for-loops but I thought may be there is a

> shorter and more elegant way to compute this.

> Sorry, for my incomplete explanation.

>

>

> tom

Ok, then if I understand correctly, if you have just two  
3d vectors a and b, you want to compute the outer (or tensor)  
product of them in this way:

```
a=transpose([1,2,3])
b=[10,20,30]
print,a##b
```

```
10    20    30
20    40    60
30    60    90
```

Now, you just happen to have n couples of 3d vectors and want to compute the n products as above, right?

The you don't have to worry about the loops over the dimensions (is just 3 by 3 = 9 times), but you want the multiplication of the n elements to be vectorized. So if a is nx3 array and b an nx3 array, the result c should be a nx3x3 matrix given by

```
FOR i=0,2 DO FOR j=0,2 DO c[* ,i,j]=a[* ,i]*b[* ,j]
```

where c[m, \*, \*] is the m-th 3x3 matrix you want.

To see all the 3x3 matrices, you can use

```
print,transpose(c,[2,1,0])
```

Ciao,  
Paolo

>

---