Subject: Re: matrix multiplication of 2 three-dimensional arrays Posted by Juggernaut on Wed, 20 Aug 2008 13:36:27 GMT

View Forum Message <> Reply to Message

On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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

Have you searched help on product() and its dimensional keyword? This could be useful for you.

Subject: Re: matrix multiplication of 2 three-dimensional arrays Posted by pgrigis on Wed, 20 Aug 2008 14:34:06 GMT

View Forum Message <> Reply to Message

Bennett wrote:

- > On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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
- >
- > Have you searched help on product() and its dimensional keyword?
- > This could be useful for you.

It is not clear to me how "product" can be used for solving matrix multiplications.

To the original poster:

- 1) your problem is so small that I don't see any need for optimization.
- 2) however, if you really want to optimize in case that the number of matrices N should increase in the future, use loops over the 3x3 matrix

arrays and columns instead and treat the matrix elements as N-element vectors. This way, more work is done per loop for large values of N.

Ciao, Paolo

Subject: Re: matrix multiplication of 2 three-dimensional arrays Posted by thomas.jagdhuber@dlr. on Wed, 20 Aug 2008 14:35:29 GMT View Forum Message <> Reply to Message

On 20 Aug., 15:36, Bennett <juggernau...@gmail.com> wrote:
> On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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

>> reduid do this with a for loop over the dimension [1500] but if so suppose this is not very elegant. Is there any other way to do this

>> time-efficient.

>> Best regards,

>> thomas

> Have you searched help on product() and its dimensional keyword?

> This could be useful for you.

No bad idea, but product() does not work as a matrix product, as far as I understood the descripition of the function

Subject: Re: matrix multiplication of 2 three-dimensional arrays Posted by thomas.jagdhuber@dlr. on Wed, 20 Aug 2008 14:39:58 GMT View Forum Message <> Reply to Message

```
On 20 Aug., 16:34, pgri...@gmail.com wrote:
> Bennett wrote:
>> On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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
>> Have you searched help on product() and its dimensional keyword?
>> This could be useful for you.
> It is not clear to me how "product" can be used for solving
> matrix multiplications.
> To the original poster:
> 1) your problem is so small that I don't see any need for
> optimization.
> 2) however, if you really want to optimize in case that the number of
> matrices N should increase in the future, use loops over the 3x3
> matrix
> arrays and columns instead and treat the matrix elements as N-element
> vectors. This way, more work is done per loop for large values of N.
> Ciao.
> Paolo
I think i will calculate each matrix element alone by the linear
```

combination and then just use the whole vector of 1500 Values for calculating each linear combination. this should be reasonable as long as the 3x3-dimension is valid and not growing.

Subject: Re: matrix multiplication of 2 three-dimensional arrays Posted by pgrigis on Wed, 20 Aug 2008 14:51:45 GMT View Forum Message <> Reply to Message

thomas.jagdhuber wrote:

```
> On 20 Aug., 16:34, pgri...@gmail.com wrote:
>> Bennett wrote:
>>> On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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
>>> Have you searched help on product() and its dimensional keyword?
>>> This could be useful for you.
>>
>> It is not clear to me how "product" can be used for solving
>> matrix multiplications.
>>
>> To the original poster:
>> 1) your problem is so small that I don't see any need for
  optimization.
>>
>> 2) however, if you really want to optimize in case that the number of
>> matrices N should increase in the future, use loops over the 3x3
>> matrix
>> arrays and columns instead and treat the matrix elements as N-element
  vectors. This way, more work is done per loop for large values of N.
>>
>> Ciao,
>> Paolo
> I think i will calculate each matrix element alone by the linear
> combination and then just use the whole vector of 1500 Values for
> calculating each linear combination. this should be reasonable as long
> as the 3x3-dimension is valid and not growing.
Yes, that is exactly what I was suggesting. The code will look ugly
though...;-)
Ciao,
Paolo
```

Subject: Re: matrix multiplication of 2 three-dimensional arrays Posted by Juggernaut on Wed, 20 Aug 2008 16:08:06 GMT

View Forum Message <> Reply to Message

```
On Aug 20, 10:51 am, pgri...@gmail.com wrote:
> thomas.jagdhuber wrote:
>> On 20 Aug., 16:34, pgri...@gmail.com wrote:
>>> Bennett wrote:
>>> On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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
>>>> Have you searched help on product() and its dimensional keyword?
>>> This could be useful for you.
>>> It is not clear to me how "product" can be used for solving
>>> matrix multiplications.
>>> To the original poster:
>>> 1) your problem is so small that I don't see any need for
>>> optimization.
>
>>> 2) however, if you really want to optimize in case that the number of
>>> matrices N should increase in the future, use loops over the 3x3
>>> matrix
>>> arrays and columns instead and treat the matrix elements as N-element
>>> vectors. This way, more work is done per loop for large values of N.
>
>>> Ciao,
>>> Paolo
>
>> I think i will calculate each matrix element alone by the linear
>> combination and then just use the whole vector of 1500 Values for
>> calculating each linear combination, this should be reasonable as long
>> as the 3x3-dimension is valid and not growing.
> Yes, that is exactly what I was suggesting. The code will look ugly
```

```
> though...;-)
> Ciao,
> Paol
```

Way I read it was that you had a matrix of 3x3x1500 and wanted to multiply them together....not sure what it means to say 1500 times 2 matrices of 3x3 dimension unless you just mean 1500*[A]*[B] on an element by element basis where A = [[a,b,c],[d,e,f],[g,h,i]] and B =[[a2,b2,c2],[d2,e2,f2],[g2,h2,i2]]? That's how that reads to me... I don't think I quite understand what it is that you're trying to do so therein lies the confusion.

Subject: Re: matrix multiplication of 2 three-dimensional arrays Posted by Jeremy Bailin on Wed, 20 Aug 2008 16:11:55 GMT

```
View Forum Message <> Reply to Message
On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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.

Subject: Re: matrix multiplication of 2 three-dimensional arrays Posted by Juggernaut on Wed, 20 Aug 2008 16:27:29 GMT View Forum Message <> Reply to Message

```
On Aug 20, 12:11 pm, Jeremy Bailin <astroco...@gmail.com> wrote:
> On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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.
```

Nevermind I see what you're wanting... sorry for the confusion

Subject: Re: matrix multiplication of 2 three-dimensional arrays
Posted by thomas.jagdhuber@dlr. on Wed, 20 Aug 2008 19:53:16 GMT

View Forum Message <> Reply to Message

```
On 20 Aug., 18:11, Jeremy Bailin <astroco...@gmail.com> wrote:
> On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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

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.jagdhu...@gmail.com>
wrote:
> On 20 Aug., 18:11, Jeremy Bailin <astroco...@gmail.com> wrote:
>
>
>> On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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:
\Rightarrow q = rebin(indgen(3),3,3)
\rightarrow 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 collegue:
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
```

Subject: Re: matrix multiplication of 2 three-dimensional arrays Posted by Jeremy Bailin on Thu, 21 Aug 2008 13:48:31 GMT View Forum Message <> Reply to Message

```
On Aug 21, 6:00 am, "thomas.jagdhuber" <thomas.jagdhu...@gmail.com>
wrote:
> On 20 Aug., 21:53, "thomas.jagdhuber" <thomas.jagdhu...@gmail.com>
> wrote:
>
>> On 20 Aug., 18:11, Jeremy Bailin <astroco...@gmail.com> wrote:
>>> On Aug 20, 8:22 am, "thomas.jagdhuber" <thomas.jagdhu...@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 collegue:
> 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
```

Yeah, that's basically Paolo's solution. I hadn't realized that REBIN doesn't accept complex arguments - that's quite annoying!

-Jeremy.