Subject: Re: simple array math question
Posted by Craig Markwardt on Thu, 16 Jan 2003 21:30:37 GMT
View Forum Message <> Reply to Message

"Sean Raffuse" <sean@me.wustl.edu> writes:

> Hello.
>
> Here is what I believe will be a pretty simple question.
>
>>> a=[[1,2,3],[4,5,6],[7,8,9]]
>
>>> b=[1,2,3]
>
> What is the best (read, fastest) way to multiply b by each individual row of
> a? I would like to return a result of:
>
> [[1,4,9],[4,10,18],[7,14,27]]
>
> assuming my arithmatic is correct.

x = ([1,1,1]##b) * a

A more stylistically correct way to do it would be to REBIN/REFORM the
variable B into the proper dimensions.

Craig
--

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

Subject: Re: simple array math question
Posted by Heinz Stege on Fri, 17 Jan 2003 00:26:25 GMT
View Forum Message <> Reply to Message

On Thu, 16 Jan 2003 14:05:27 -0600, "Sean Raffuse" <sean@me.wustl.edu>
wrote:

>>> a=[[1,2,3],[4,5,6],[7,8,9]]
>
>>> b=[1,2,3]
>
> What is the best (read, fastest) way to multiply b by each individual row of
> a? I would like to return a result of:
>

> [[1,4,9],[4,10,18],[7,14,27]]

result=a*b(*,intarr(3))

Heinz

Subject: Re: simple array math question
Posted by Craig Markwardt on Fri, 17 Jan 2003 03:47:46 GMT
View Forum Message <> Reply to Message

Heinz Stege <reply_to_posting@arcor.de> writes:
> On Thu, 16 Jan 2003 14:05:27 -0600, "Sean Raffuse" <sean@me.wustl.edu>
> wrote:
>
>>>> a=[[1,2,3],[4,5,6],[7,8,9]]
>>
>>>> b=[1,2,3]
>>
>> What is the best (read, fastest) way to multiply b by each individual row of
>> a? I would like to return a result of:
>>
>> [[1,4,9],[4,10,18],[7,14,27]]
>
>
> result=a*b(*,intarr(3))

WOW! I've never seen that! It scares me how cool that is. :-)

Craig

--

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

Subject: Re: simple array math question
Posted by JD Smith on Fri, 17 Jan 2003 21:46:58 GMT
View Forum Message <> Reply to Message

On Thu, 16 Jan 2003 20:47:46 -0700, Craig Markwardt wrote:

> Heinz Stege <reply_to_posting@arcor.de> writes:

>> On Thu, 16 Jan 2003 14:05:27 -0600, "Sean Raffuse" <sean@me.wustl.edu>
>> wrote:
>>
>>>> > a=[[1,2,3],[4,5,6],[7,8,9]]
>>>
>>>> > b=[1,2,3]
>>>
>>> What is the best (read, fastest) way to multiply b by each individual
>>> row of a? I would like to return a result of:
>>>
>>> [[1,4,9],[4,10,18],[7,14,27]]
>>
>>
>> result=a*b(*,intarr(3))
>
> WOW! I've never seen that! It scares me how cool that is. :-)
>
> Craig

I may have to add that to the REBIN/REFORM tutorial. I'll see how fast it is first. It's definitely one of the more readeable ways to add a new trailing dimension. Doesn't work for leading or in-the-middle dimensions, as far as I can tell.

JD

Subject: Re: simple array math question
Posted by Sean Raffuse on Fri, 17 Jan 2003 22:10:49 GMT
View Forum Message <> Reply to Message

"JD Smith" <jdsmith@as.arizona.edu> wrote in message
news:pan.2003.01.17.21.46.58.749790.29842@as.arizona.edu...
> On Thu, 16 Jan 2003 20:47:46 -0700, Craig Markwardt wrote:
>
>
>> Heinz Stege <reply_to_posting@arcor.de> writes:
>>> On Thu, 16 Jan 2003 14:05:27 -0600, "Sean Raffuse" <sean@me.wustl.edu>
>>> wrote:
>>>
>>>>> >> a=[[1,2,3],[4,5,6],[7,8,9]]
>>>>>
>>>>> >> b=[1,2,3]
>>>>>
>>>> What is the best (read, fastest) way to multiply b by each individual
>>>> row of a? I would like to return a result of:
>>>>

>>>> [[1,4,9],[4,10,18],[7,14,27]]
>>>
>>>
>>> result=a*b(*,intarr(3))
>>
>> WOW! I've never seen that! It scares me how cool that is. :-)
>>
>> Craig
>
>
> I may have to add that to the REBIN/REFORM tutorial. I'll see how fast
> it is first. It's definitely one of the more readeable ways to add a new
> trailing dimension. Doesn't work for leading or in-the-middle
> dimensions, as far as I can tell.
>
> JD

Yes, it worked wonders for me. Plenty fast compared to what I was doing before. I did have to use it on some leading dimensions and so am using transpose(). Not sure if it is the optimal solution, but good enough for my needs.

Subject: Re: simple array math question
Posted by Heinz Stege on Sat, 18 Jan 2003 00:51:51 GMT
View Forum Message <> Reply to Message

On Fri, 17 Jan 2003 16:10:49 -0600, "Sean Raffuse" <sean@me.wustl.edu>
wrote:

>
> "JD Smith" <jdsmith@as.arizona.edu> wrote in message
> news:pan.2003.01.17.21.46.58.749790.29842@as.arizona.edu...
>> On Thu, 16 Jan 2003 20:47:46 -0700, Craig Markwardt wrote:
>>
>>
>>> Heinz Stege <reply_to_posting@arcor.de> writes:
>>>> On Thu, 16 Jan 2003 14:05:27 -0600, "Sean Raffuse" <sean@me.wustl.edu>
>>>> wrote:
>>>>
>>>> >>> a=[[1,2,3],[4,5,6],[7,8,9]]
>>>> >
>>>> >>> b=[1,2,3]
>>>> >
>>>> >What is the best (read, fastest) way to multiply b by each individual
>>>> >row of a? I would like to return a result of:
>>>> >
>>>> >[[1,4,9],[4,10,18],[7,14,27]]

>>>>
>>>>
>>>> result=a*b(*,intarr(3))
>>>
>>> WOW! I've never seen that! It scares me how cool that is. :-)
>>>
>>> Craig
>>
>>
>> I may have to add that to the REBIN/REFORM tutorial. I'll see how fast
>> it is first. It's definitely one of the more readeable ways to add a new
>> trailing dimension. Doesn't work for leading or in-the-middle
>> dimensions, as far as I can tell.
>>
>> JD
>
> Yes, it worked wonders for me. Plenty fast compared to what I was doing
> before. I did have to use it on some leading dimensions and so am using
> transpose(). Not sure if it is the optimal solution, but good enough for my
> needs.
>

Yes, TRANSPOSE is also my favorite for the leading dimension:

```
result=a*(transpose(b))(intarr(3),*)
```

may be, you would prefer following syntax:

```
temp=transpose(b)
result=a*temp(intarr(3),*)
```

For middle dimensions REFORM does a good job :

```
c=fltarr(2,3,4)
...
d=fltarr(2,4)
...
result=c*(reform(d,2,1,4))(*,intarr(3),*)
```

(all tested with PV-Wave)

Heinz

Subject: Re: simple array math question
Posted by Jeff Guerber on Sat, 18 Jan 2003 04:21:34 GMT
View Forum Message <> Reply to Message

On 16 Jan 2003, Craig Markwardt wrote:

> Heinz Stege <reply_to_posting@arcor.de> writes:
>> On Thu, 16 Jan 2003 14:05:27 -0600, "Sean Raffuse" <sean@me.wustl.edu>
>> wrote:
>>
>>>> > a=[[1,2,3],[4,5,6],[7,8,9]]
>>>
>>>> > b=[1,2,3]
>>>
>>> What is the best (read, fastest) way to multiply b by each individual row of
>>> a? I would like to return a result of:
>>>
>>> [[1,4,9],[4,10,18],[7,14,27]]
>>
>>
>> result=a*b(*,intarr(3))
>
> WOW! I've never seen that! It scares me how cool that is. :-)
>
> Craig

That IS way cool, but, uhhh, would someone mind explaining just what's going on? I'm not getting it. It seems to only depend on the total number of elements in the indexing array, not on its values (or even dimensionality):

```
IDL> b=[4,5,6]
IDL> print,b[*,[10,10,10]]
       4       5       6
       4       5       6
       4       5       6
IDL> print,b[*,[1000,1000]]
       4       5       6
       4       5       6
IDL> print,b[*,[[10,10,10],[10,10,10]]]
       4       5       6
       4       5       6
       4       5       6
       4       5       6
       4       5       6
       4       5       6
IDL> help,b[*,[[10,10,10],[10,10,10]]]
<Expression>    INT       = Array[3, 6]
IDL>
```

Oh, you can apply it multiple times, too:

```
IDL> print,b[*,[10,10,10,10],[10,10]]
       4       5       6
```

```
      4       5       6
      4       5       6
      4       5       6

      4       5       6
      4       5       6
      4       5       6
      4       5       6
IDL> help,b[*,[10,10,10,10],[10,10]]
<Expression>    INT       = Array[3, 4, 2]
IDL>
```

Thanks,

Jeff Guerber

Subject: Re: simple array math question
Posted by marc schellens[1] on Mon, 20 Jan 2003 06:55:50 GMT
View Forum Message <> Reply to Message

Jeff Guerber wrote:
> On 16 Jan 2003, Craig Markwardt wrote:
>
>
>> Heinz Stege <reply_to_posting@arcor.de> writes:
>>
>>> On Thu, 16 Jan 2003 14:05:27 -0600, "Sean Raffuse" <sean@me.wustl.edu>
>>> wrote:
>>>
>>>
>>>> >>a=[[1,2,3],[4,5,6],[7,8,9]]
>>>> >
>>>> >>b=[1,2,3]
>>>> >
>>>> What is the best (read, fastest) way to multiply b by each individual row of
>>>> a? I would like to return a result of:
>>>>
>>>> [[1,4,9],[4,10,18],[7,14,27]]
>>>
>>>
>>> result=a*b(*,intarr(3))
>>
>> WOW! I've never seen that! It scares me how cool that is. :-)
>>
>> Craig
>
>

> That IS way cool, but, uhhh, would someone mind explaining just what's
> going on? I'm not getting it. It seems to only depend on the total
> number of elements in the indexing array, not on its values (or even
> dimensionality):
>
> IDL> b=[4,5,6]
> IDL> print,b[*,[10,10,10]]
>   4   5   6
>   4   5   6
>   4   5   6
> IDL> print,b[*,[1000,1000]]
>   4   5   6
>   4   5   6
> IDL> print,b[*,[[10,10,10],[10,10,10]]]
>   4   5   6
>   4   5   6
>   4   5   6
>   4   5   6
>   4   5   6
>   4   5   6
> IDL> help,b[*,[[10,10,10],[10,10,10]]]
> <Expression>   INT   = Array[3, 6]
> IDL>
>
> Oh, you can apply it multiple times, too:
>
> IDL> print,b[*,[10,10,10,10],[10,10]]
>   4   5   6
>   4   5   6
>   4   5   6
>   4   5   6
>
>   4   5   6
>   4   5   6
>   4   5   6
>   4   5   6
> IDL> help,b[*,[10,10,10,10],[10,10]]
> <Expression>   INT   = Array[3, 4, 2]
> IDL>
>
If you index an array in IDL with another array,
IDL always eats it, and sets the index to the bounds of the
array.

e.g:
IDL> print,indgen(5)-1
      -1   0   1   2   3
IDL> a=indgen(3)

IDL> print,a[indgen(5)-1]
       0   0   1   2   2

so the result is:
a[0], a[0], a[1], a[2], a[2]

this works for all dimensions, so in you case
b[*,[10,10,10,10],[10,10]]

is the same as

b[*,[0,0,0,0],[0,0]]

because b is
B   INT   = Array[3]

which is in IDL the same as:
Array[3,0,0,0,0,0,0,0]

hope that helped,
marc

Subject: Re: simple array math question
Posted by JD Smith on Thu, 23 Jan 2003 16:53:33 GMT
View Forum Message <> Reply to Message

On Fri, 17 Jan 2003 14:46:58 -0700, JD Smith wrote:

> On Thu, 16 Jan 2003 20:47:46 -0700, Craig Markwardt wrote:
>
>
>> Heinz Stege <reply_to_posting@arcor.de> writes:
>>> On Thu, 16 Jan 2003 14:05:27 -0600, "Sean Raffuse" <sean@me.wustl.edu>
>>> wrote:
>>>
>>>> >> a=[[1,2,3],[4,5,6],[7,8,9]]
>>>>
>>>> >> b=[1,2,3]
>>>>
>>>> What is the best (read, fastest) way to multiply b by each individual
>>>> row of a? I would like to return a result of:
>>>>
>>>> [[1,4,9],[4,10,18],[7,14,27]]
>>>
>>>
>>> result=a*b(*,intarr(3))
>>

>> WOW! I've never seen that! It scares me how cool that is. :-)
>>
>> Craig
>
>
> I may have to add that to the REBIN/REFORM tutorial. I'll see how fast
> it is first. It's definitely one of the more readeable ways to add a
> new trailing dimension. Doesn't work for leading or in-the-middle
> dimensions, as far as I can tell.
>
>

OK, I've performed some speed tests comparing the two methods:

Expanding 1st dimension:

```
 d=findgen(j,k)

 e=rebin(reform(d,1,j,k),i,j,k,/SAMPLE)
  vs.
 e=(reform(d,1,j,k))[intarr(i),*,*]
```

Expanding middle dimension:

```
 d=findgen(i,k)

 e=rebin(reform(d,i,1,k),i,j,k,/SAMPLE)
  vs.
 e=(reform(d,i,1,k))[*,intarr(j),*]
```

Expanding last dimension:

```
 d=findgen(i,j)

 e=rebin(d,i,j,k,/SAMPLE)
  vs.
 e=d[*,*,intarr(k)]
```

I tested this for all permutations of (i,j,k)=(100,200,500)

When your arrays fit in memory, the rebin(reform) method is 2.9-3.5 times faster. Actually, it's remarkably close to 3.0 in all cases. However, when you begin to run out of memory, the intarr() method really begins to suffer, up to 25 times slower. I suspect this is because all the index arrays must be pre-computed in memory when "*" is used.

The one convenience of the slower method: you don't need to keep track

of and enter the other two dimensions. However, since REBIN/REFORM (as of v5.5) now take a single dimension argument, this problem is minimized; you can save yourself the trouble like this:

```
 dim=size(d,/dimension)
 e=rebin(reform(d,[1,dim]),[i,dim],/SAMPLE)
```

or

```
 e=rebin(d,[dim,k],/SAMPLE)
```

or

```
  e=rebin(reform(d,[dim[0],1,dim[2:*]]),[dim[0],j,dim[2:*]],/S AMPLE)
```

and this has the added advantage that you don't even need to know how many dimensions your arrays have, just where you'd like to add a dimension of some size.

JD

Subject: Re: simple array math question
Posted by Pepijn Kenter on Fri, 24 Jan 2003 12:43:30 GMT
View Forum Message <> Reply to Message

"Craig Markwardt" <craigmnet@cow.physics.wisc.edu> wrote in message news:onwul4cu8t.fsf@cow.physics.wisc.edu...
>
> Heinz Stege <reply_to_posting@arcor.de> writes:
>> On Thu, 16 Jan 2003 14:05:27 -0600, "Sean Raffuse" <sean@me.wustl.edu>
>> wrote:
>>
>>>> > a=[[1,2,3],[4,5,6],[7,8,9]]
>>>
>>>> > b=[1,2,3]
>>>
>>> What is the best (read, fastest) way to multiply b by each individual
row of
>>> a? I would like to return a result of:
>>>
>>> [[1,4,9],[4,10,18],[7,14,27]]
>>
>> result=a*b(*,intarr(3))
>
> WOW! I've never seen that! It scares me how cool that is. :-)

That is mighty cool indeed! Is that documented somewhere?

And does any IDL-wizard know a similar trick to average each row/column of
a?
i.e. to replace the following lines:

result = dblarr(3)
for i = 0, 2 do result[i] = mean(a[*,i])

I don't think it's possible, but who knows.

Pepijn Kenter.

Subject: Re: simple array math question
Posted by Jeff Guerber on Sat, 25 Jan 2003 09:12:08 GMT
View Forum Message <> Reply to Message

On Mon, 20 Jan 2003, Marc Schellens wrote:

> If you index an array in IDL with another array,
> IDL always eats it, and sets the index to the bounds of the
> array.
>
> e.g:
> IDL> print,indgen(5)-1
> -1 0 1 2 3
> IDL> a=indgen(3)
> IDL> print,a[indgen(5)-1]
> 0 0 1 2 2
>
> so the result is:
> a[0], a[0], a[1], a[2], a[2]
>
> this works for all dimensions, so in you case
> b[*,[10,10,10,10],[10,10]]
>
> is the same as
>
> b[*,[0,0,0,0],[0,0]]
>
> because b is
> B INT = Array[3]
>
> which is in IDL the same as:
> Array[3,0,0,0,0,0,0,0]
>
> hope that helped,

> marc

That did indeed, thanks Marc! I think it was mostly the trailing degenerate dimensions that I was missing. It's instructive too to play around with this with a two-dimensional b:

```
IDL> b=[[1,2,3],[4,5,6]]
IDL> print,b[*,[0,0]]
      1       2       3
      1       2       3
IDL> print,b[*,[0,1,2],[-1,0,1,2]]
      1       2       3
      4       5       6
      4       5       6

      1       2       3
      4       5       6
      4       5       6

      1       2       3
      4       5       6
      4       5       6

      1       2       3
      4       5       6
      4       5       6
IDL>
```

Jeff Guerber

Subject: Re: simple array math question
Posted by Heinz Stege on Sun, 26 Jan 2003 19:25:36 GMT
View Forum Message <> Reply to Message

On Thu, 23 Jan 2003 09:53:33 -0700, JD Smith <jdsmith@as.arizona.edu> wrote:

<snip>
>
> OK, I've performed some speed tests comparing the two methods:
>
<snip>
>
> JD

Thanks a lot for this very instructive contribution! Since the proposal of the intarr() method was from me, the NG may allow me to state this here.

REBIN(REFORM(...)) is the better alternative. This is obvious now.

Heinz
