Subject: simple array math question
Posted by Sean Raffuse on Thu, 16 Jan 2003 20:05:27 GMT
View Forum Message <> Reply to Message

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.

Thanks in advance,

Sean Raffuse

Subject: Re: simple array math question
Posted by K. Bowman on Fri, 24 Jan 2003 14:38:18 GMT
View Forum Message <> Reply to Message

In article <b0rce6$23c$1@news.surfnet.nl>,
 "Pepijn Kenter" <kenter@tpd.tno.nl> wrote:

> 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])

That's an easy one:

result1 = TOTAL(a, 1, /DOUBLE)/n1
result2 = TOTAL(a, 2, /DOUBLE)/n2

where n1 and n2 are the sizes of the first and second dimensions, respectively.

Ken Bowman

Subject: Re: simple array math question
Posted by Pepijn Kenter on Fri, 24 Jan 2003 15:50:03 GMT
View Forum Message <> Reply to Message

"Kenneth Bowman" <k-bowman@null.tamu.edu> wrote in message
news:k-bowman-4D0582.08381824012003@news.tamu.edu...
> In article <b0rce6$23c$1@news.surfnet.nl>,
> "Pepijn Kenter" <kenter@tpd.tno.nl> wrote:
>
>> 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])
>
> That's an easy one:
>
> result1 = TOTAL(a, 1, /DOUBLE)/n1
> result2 = TOTAL(a, 2, /DOUBLE)/n2
>
> where n1 and n2 are the sizes of the first and second dimensions,
> respectively.
>
> Ken Bowman

Thanks, but what I actually meant was a way where you can replace the mean
function by any, arbitrary function.
I forgot about the dimension parameter of the TOTAL function, so my example
was a bad one. Sorry.

Pepijn.

Subject: Re: simple array math question
Posted by Craig Markwardt on Fri, 24 Jan 2003 19:29:11 GMT
View Forum Message <> Reply to Message

"Pepijn Kenter" <kenter@tpd.tno.nl> writes:
> "Kenneth Bowman" <k-bowman@null.tamu.edu> wrote in message
> news:k-bowman-4D0582.08381824012003@news.tamu.edu...
>> In article <b0rce6$23c$1@news.surfnet.nl>,
>> "Pepijn Kenter" <kenter@tpd.tno.nl> wrote:
>>
>>> 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])
>>
>> That's an easy one:
>>
>> result1 = TOTAL(a, 1, /DOUBLE)/n1
>> result2 = TOTAL(a, 2, /DOUBLE)/n2
>>
>> where n1 and n2 are the sizes of the first and second dimensions,
>> respectively.
>>
>> Ken Bowman
>
> Thanks, but what I actually meant was a way where you can replace the mean
> function by any, arbitrary function.
> I forgot about the dimension parameter of the TOTAL function, so my example
> was a bad one. Sorry.

"Arbitrary function"? No. In more recent versions of IDL, quite a number of built-in math functions have the ability to specify one or more dimensions over which the function is to be applied. I have a function called CMAPPLY, which may be useful with older versions of IDL, and does many of those same things. One thing it can do is apply an arbitrary function, but it won't necessarily be speedy. JD Smith also has a DLM which did a lot of these kinds of things.

Good luck,
Craig

http://cow.physics.wisc.edu/~craigm/idl/idl.html (under Array/Set ops)

--
---------- ----------
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 Craig Markwardt on Sun, 26 Jan 2003 20:11:31 GMT
View Forum Message <> Reply to Message

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

"Better" can be defined in a lot of ways. Your solution is by far the
most readable way, in my view, to extend arrays. And it's cool
because nobody seems to have discovered it before!

Speed is only king when you need a king.

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 Mon, 27 Jan 2003 18:00:59 GMT
View Forum Message <> Reply to Message

On Sun, 26 Jan 2003 13:11:31 -0700, Craig Markwardt wrote:

> Heinz Stege <reply_to_posting@arcor.de> writes:
>>
>> 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.
>
> "Better" can be defined in a lot of ways. Your solution is by far the
> most readable way, in my view, to extend arrays. And it's cool because
> nobody seems to have discovered it before!
>
> Speed is only king when you need a king.

Well put. Ideally, IDL would have an intrinsic, readable, "inflate"
operator to do what we're all doing in roundabout, quasi-readable ways
right now. Maybe something like:

IDL> a=findgen(10,12,14)
IDL> g=a[3;7]*randomu(sd,10,12,7,14)

I.e. replicate 7 times over the third dimension.

You might even be able to make it general enough to combine with the
other indexing operators.

JD

Subject: Re: simple array math question
Posted by JD Smith on Mon, 27 Jan 2003 18:20:02 GMT
View Forum Message <> Reply to Message

On Fri, 24 Jan 2003 12:29:11 -0700, Craig Markwardt wrote:

> "Pepijn Kenter" <kenter@tpd.tno.nl> writes:
>> "Kenneth Bowman" <k-bowman@null.tamu.edu> wrote in message
>> news:k-bowman-4D0582.08381824012003@news.tamu.edu...
>>> In article <b0rce6$23c$1@news.surfnet.nl>,
>>> "Pepijn Kenter" <kenter@tpd.tno.nl> wrote:
>>>
>>>> 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])
>>>
>>> That's an easy one:
>>>
>>> result1 = TOTAL(a, 1, /DOUBLE)/n1
>>> result2 = TOTAL(a, 2, /DOUBLE)/n2
>>>
>>> where n1 and n2 are the sizes of the first and second dimensions,
>>> respectively.
>>>
>>> Ken Bowman
>>
>> Thanks, but what I actually meant was a way where you can replace the
>> mean function by any, arbitrary function. I forgot about the dimension
>> parameter of the TOTAL function, so my example was a bad one. Sorry.
>
> "Arbitrary function"? No. In more recent versions of IDL, quite a
> number of built-in math functions have the ability to specify one or
> more dimensions over which the function is to be applied. I have a
> function called CMAPPLY, which may be useful with older versions of IDL,

> and does many of those same things. One thing it can do is apply an
> arbitrary function, but it won't necessarily be speedy. JD Smith also
> has a DLM which did a lot of these kinds of things.
>

My REDUCE DLM has been entirely superseded with v5.6 by builtin methods. I think the list currently stands at:

MAX
MEAN
MEDIAN
MIN
MULTIPLY (PRODUCT)
TOTAL

And of course REFORM and REBIN (obviously).

You can use these primitives to build up the moments along any dimension... see MOMENT and moment.pro. In fact, I think just sticking an _EXTRA in all the totals in that file, and touching up the argument checking would buy you a REDUCE-capable MOMENT as well (not sure why RSI didn't do this for us). Note that sometimes you use a keyword (DIMENSION) and sometimes you use a vector argument.

What we don't have is an APL style function-as-argument mechanism, with the attendant abilities to thread and collaps function over any dimension of arbitrarily sized arrays. But IDL's argument rules are far too loose to permit this anyway.

Good luck,

JD

Subject: Re: simple array math question
Posted by Craig Markwardt on Mon, 27 Jan 2003 19:54:18 GMT
View Forum Message <> Reply to Message

JD Smith <jdsmith@as.arizona.edu> writes:

> On Sun, 26 Jan 2003 13:11:31 -0700, Craig Markwardt wrote:
>
>
>> Heinz Stege <reply_to_posting@arcor.de> writes:
>>>
>>> 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.
>>
>> "Better" can be defined in a lot of ways. Your solution is by far the
>> most readable way, in my view, to extend arrays. And it's cool because
>> nobody seems to have discovered it before!
>>
>> Speed is only king when you need a king.
>
>
> Well put. Ideally, IDL would have an intrinsic, readable, "inflate"
> operator to do what we're all doing in roundabout, quasi-readable ways
> right now. Maybe something like:
>
> IDL> a=findgen(10,12,14)
> IDL> g=a[3;7]*randomu(sd,10,12,7,14)

Yorick, a language very similar to IDL, but sadly not very popular,
has some very cool ways to do this. The first is a "pseudo-index"
which automatically introduces dimensions of length one.

a[*,*,-,*] would be a 10x12x1x14 array, the "-" introducing the extra
        dimension. Yorick has a separate meaning for the "*"
        operator but I mean it here in the normal IDL sense.

The second thing is "broadcasting," which automatically extends
dimensions of length 1. Thus, in the expression,

  a[*,*,-,*] * randomu(sd,10,12,7,14)

The 3rd dimension is automatically extended to length 7 by replicating
A. Someone has already thought about these things (Yorick's author,
David Munro), and they are very cool!

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 Pepijn Kenter on Tue, 28 Jan 2003 13:40:03 GMT
View Forum Message <> Reply to Message

Thanks all.