Subject: Re: Using two different arrays in tha same calculation
Posted by dan on Wed, 08 Jun 1994 14:42:33 GMT
View Forum Message <> Reply to Message

Newsgroup: comp.lang.idl-pvwave:
Subject: Using two different arrays in tha same calculation

ragnar@kvark.fi.uib.no (Ragnar Aas) writes :

> I have the following problem :
>
> I have two different arrays, (8) of float and (300,8) of float.
> I want to vectorize the equation and therefore I need to use both
> arrays in the same equation. For example :
>
> newarray=cos(small_array)*sin(large_array)
>
> where I want the data in small_array to be used over and over 300 times
> in this calculation.
>
> Hope somebody can help me.
>

Ragnar,

You can use the # operator (matrix multiply) to make the small (8)
array into a (300,8) array. Try this :

```
IDL> arr1 = Findgen(8)
IDL> arr2 = Findgen(300, 8)
IDL> newarr = (Replicate(1.0, 300) # Cos(arr1)) * Sin(arr2)
```

Dan Carr
Research Systems
Boulder, Colorado
dan@rsinc.com

Subject: Re: Using two different arrays in tha same calculation
Posted by thompson on Thu, 09 Jun 1994 13:20:39 GMT
View Forum Message <> Reply to Message

ragnar@kvark.fi.uib.no (Ragnar Aas) writes:

> I have the following problem :

> I have two different arrays, (8) of float and (300,8) of float.
> I want to vectorize the equation and therefore I need to use both
> arrays in the same equation. For example :

> newarray=cos(small_array)*sin(large_array)

> where I want the data in small_array to be used over and over 300 times
> in this calculation.

That's simple,

 newarray = cos(replicate(1,300)#small_array) * sin(large_array)

Bill Thompson

P.S. Another one for the FAQ?

Subject: Re: Using two different arrays in tha same calculation
Posted by landers on Thu, 09 Jun 1994 16:09:29 GMT
View Forum Message <> Reply to Message

In article <2t49er$dp5@due.uninett.no>, ragnar@kvark.fi.uib.no (Ragnar Aas) writes:
|> I have the following problem :
|>
|> I have two different arrays, (8) of float and (300,8) of float.
|> I want to vectorize the equation and therefore I need to use both
|> arrays in the same equation. For example :
|>
|> newarray=cos(small_array)*sin(large_array)
|>
|> where I want the data in small_array to be used over and over 300 times
|> in this calculation.
|>
|> Hope somebody can help me.
|>
|> Ragnar Aas

Try like this:

a = fltarr(8)
b = fltarr(300,8)

ia = lindgen(300,8) / 300L

c = a(ia) * b

Here's why.... ia is a 300,8 array of indices into the 8-element array.

So ia(*,0) = 0, ia(*,1) = 1, etc.

Then a(ia) becomes a 300,8 array that repeats the 8-element array 300 times.

Note that you'd have to play with this if the index orders were reversed (like
you were trying to operate on an 8-element and a (8,300) arrays). You'd need
to change ia to be lindgen(8,300) mod 8

;Dave

Subject: Re: Using two different arrays in tha same calculation
Posted by steinhh on Fri, 10 Jun 1994 07:44:17 GMT
View Forum Message <> Reply to Message

In article <thompson.771168039@serts.gsfc.nasa.gov>, thompson@serts.gsfc.nasa.gov (William Thompson) writes:
|> ragnar@kvark.fi.uib.no (Ragnar Aas) writes:
|>
|> >I have the following problem :
|>
|> >I have two different arrays, (8) of float and (300,8) of float.
|> >I want to vectorize the equation and therefore I need to use both
|> >arrays in the same equation. For example :
|>
|> >newarray=cos(small_array)*sin(large_array)
|>
|> >where I want the data in small_array to be used over and over 300 times
|> >in this calculation.
|>
|> That's simple,
|>
|> newarray = cos(replicate(1,300)#small_array) * sin(large_array)

It certainly does the trick, Bill, but I wouldn't think that it's more efficient
doing 8*300 cosine operations just to vectorize a multiplication of 8*300 elements :-)
I'd suggest instead:

```
newarray = replicate(1,300)#cos(small_array) * sin(large_array)
```

or (I don't know which is the more efficient - would be nice to get some feedback):

```
small_array = reform(small_array,1,8)
newarray = rebin(cos(small_array),300,8,/sample) * sin(large_array)
```

Stein Vidar

Subject: Re: Using two different arrays in tha same calculation
Posted by landers on Thu, 16 Jun 1994 13:52:30 GMT
View Forum Message <> Reply to Message

In article <Cr326y.Etw@rsinc.com>, dan@rsinc.com (Dan Carr) writes:
[snip]
|> You can use the # operator (matrix multiply) to make the small (8)
|> array into a (300,8) array. Try this :
|>
|> IDL> arr1 = Findgen(8)
|> IDL> arr2 = Findgen(300, 8)
|> IDL> newarr = (Replicate(1.0, 300) # Cos(arr1)) * Sin(arr2)
|>
|> -- -------------------
|> Dan Carr
|> Research Systems
|> Boulder, Colorado
|> dan@rsinc.com
|> -- -------------------

OK - I'll bite on this one. Now that RSI has decided to post to the group, I want to know - from the 'experts'....

Is this matrix multiply method any faster or more efficient than generating an array of indices? Which is 'better':

```
 replicate( 1.0, 300 ) # arr1
```

or

```
 arr1( Lindgen(300,8) / 300 )
```

Seems to me the second way should be more efficient - it involves one integer divide and some memory dereferencing, while the matrix method involves floating point multiplies and addition.

The second method has the added advantage that it doesn't 'mess' with your data. If you're not careful, you could convert ints to floats, etc.
Matrix multiply of any two integer types (byte, int, long) results in a long.
You also can't matrix multiply strings.

I'll grant you that the first way (#) is the easiest to remember - that's what I generally use from the command line, and it's what I recommend to people around here who need a 'quick fix' or a one-shot answer. But I am starting to use this second way in my programs, for generality and efficiency. If I'm wrong about this, let me know....

;;;; for the FAQ?

to convert an array1(M) to array2(n,M) :
 array2 = array1(Lindgen(n,M) / n)
  or array2 = replicate(1,n) # array1

to convert an array1(M) to array2(M,n) :
 array2 = array1(Lindgen(M,n) / MOD M)
  or array2 = array1 # replicate(1,n)

;Dave