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