
Subject: look, no loops!

Posted by [Robert Cannon](#) on Wed, 12 Jun 1996 07:00:00 GMT

[View Forum Message](#) <> [Reply to Message](#)

These postings about loops seem to show that the basic array manipulation routines in IDL are not quite as clear as they could be.

If you want to make a 2d array with $a(i,j) = j$ you can do it with

```
a = lindgen(nx, ny) mod nx
OR  a = lindgen(nx) # (intarr(ny) + 1)
OR  a = rebin (lindgen(nx), nx, ny)
OR  ...others?
```

But why should you have to? - You don't see many Yorick plugs on this group, but almost everything I do in IDL (with the crucial exception of widget applications) could be done better in Yorick - and it is free.

eg, the above would be
a = span(1,nx,nx)(, -:1:ny)

Yorick indices start at one. The "-:1:ny" adds an index to the 1-D array and copies the first row ny times.

You can add as many dimensions as you like; a(,,-,-) is a 4-D array with the last two dimensions equal to one.

Or, if you want the maximum along a particular dimension, you have:

```
maxx = a(max,)    a vector where each element is the max of a row
maxy = a(,max)    ...                               column
```

etc, etc for arbitrary arrays.

Likewise, you can do a generalised matrix multiplication of a pair of arrays along any dimensions by marking the dimensions you want to contract with a + sign:

```
a (+,) * b(+,) for an ordinary matrix multiplication.
```

Well, I'm very impressed by Yorick, and thought maybe someone else who was fed up with writing messy idl loops might be interested. It was written by David H. Munro at Lawrence Livermore National Laboratory. I don't know if there is a home page, but some info is at:

<http://netlib.org/env/yorick.readme>
and
<http://hpux.ee.ualberta.ca/hpux/Physics/yorick-1.0.html>

Regards, Robert Cannon

Neuroscience Research Group
Southampton UK, SO16 7PX

tel: (0)1703 594286
Fax: (0)1703 594319
rcc@hera.neuro.soton.ac.uk
