Subject: Re: For loop avoidance - getting indices of real space
Posted by Yngvar Larsen on Mon, 27 Aug 2012 10:37:34 GMT
View Forum Message <> Reply to Message

On Thursday, 23 August 2012 22:58:48 UTC+2, simu...@gmail.com wrote:

> I have read and re-read until cross-eyed this post: http://www.idlcoyote.com/tips/forloops.html
>
>
>
> And yet, I still can't quite grasp at how I can solve my for loop problem. I think it might involve the use of modulo (MOD), but I'm not sure how. My question is, how can you grab the indices (i,j,k) of a 3D array in real space, and throw them into basically 3 1D arrays that is just a list of all the cells in the "proper" order (column-major).

There is a perfectly good builtin function in IDL to do this: ARRAY_INDICES.

```
IDL> nx = 15L
IDL> ny = 10L
IDL> nz = 20L
IDL> data = randomn(seed, nx,ny,nz)
IDL> ai = array_indices(data, lindgen(nx*ny*nz))
IDL>
IDL> help, ai
AI              LONG      = Array[3, 3000]
```

If you really need dimensions [3000,3], you can add

```
IDL> ai = transpose(temporary(ai))
```

--
Yngvar