
Subject: Re: For loop avoidance - getting indices of real space

Posted by [lecacheux.alain](#) on Fri, 24 Aug 2012 09:32:05 GMT

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

Le jeudi 23 août 2012 22:58:48 UTC+2, simu...@gmail.com a écrit :

> 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).

>

>

>

> Here is an example of what I mean:

>

>

>

> pro testreader

>

>

>

> xcells=15

>

> ycells=10

>

> zcells=20

>

> ncells=xcells*ycells*zcells

>

>

>

> data=dindgen(xcells,ycells,zcells)

>

> coord=intarr(ncells,3)

>

>

>

> index=0L

>

> for k=0,zcells do begin

>

> for j=0,ycells do begin

>

> for i = 0,xcells do begin

>

```

> coord(index,0)=i
>
> coord(index,1)=j
>
> coord(index,2)=k
>
> index=index+1
>
> endfor
>
> endfor
>
> endfor
>
> end
>
>
>

```

> This is a really simple version of a complex problem I have. I have sets of different size boxes from an AMR MHD code, and I need to keep track of their indices, but I just want a list of all of the cells, not to drag around a bunch of smaller arrays or try to concatenate them into one giant sparse array (waste of space). I'm certain that someone must have had this problem before, but I can't find any other suggestions on this forum.

If I understand well your problem, a solution might be:

```

IDL> coord = [ [lindgen(xcells)#replicate(1,ycells*zcells)], $
IDL> [lindgen(ycells)#replicate(1,xcells*zcells)], $
IDL> [lindgen(zcells)#replicate(1,xcells*ycells)] ]
IDL> coord = reform(coord,ncells,3,/OVER)

```

alain.
