Subject: Re: Array indexing: what is IDL doing?
Posted by Conor on Mon, 09 Jun 2008 13:02:19 GMT
View Forum Message <> Reply to Message

On Jun 6, 5:15 pm, Jean H <jghas...@DELTHIS.ucalgary.ANDTHIS.ca>
wrote:

```
>>  cx = n_elements(wx)
>>  cy = n_elements(wy)
>>  nx = n_elements(bigarr[*,0])
>>  ind = transpose(rebin(wy,cy,cx))*nx + rebin(wx,cx,cy)
>>  res = bigarr[ind]
>
>  You don't have to use nx.
>  res = bigarr[rebin(bigx,75,10000),rebin(bigy,10000,75)]
>
>  Note that you must have the same number of elements in every subscripted
>  dimension.
>
>  Readhttp://www.dfanning.com/code_tips/asterisk.html
>
>  Jean
```

On second thought, I'm now thoroughly confused. I don't understand
this bit:

```
res = bigarr[rebin(bigx,75,10000),rebin(bigy,10000,75)]
```

obviously rebin(bigx,75,10000) and rebin(bigy,10000,75) give you two arrays with exactly opposite dimensions, so I don't understand how you can index bigarr with these two arrays. Also, I'm not even sure what IDL is doing when you index an array like that. My first thought was that this was the equivelent of:

```
res = bigarr[ [rebin(bigx,75,10000),rebin(bigy,10000,75)] ]
```

But that obviously can't be the case because since they have different dimensions, you can't concatenate the two arrays. So how does IDL pull out an array index in your above example?