Subject: Re: Array indexing: what is IDL doing?
Posted by Jean H. on Mon, 09 Jun 2008 17:12:49 GMT
View Forum Message <> Reply to Message

Conor wrote:
> 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?

Ah, I guess you are right... I was typing too fast (and the no-error
didn't warn me..)

it should have been:
IDL> help, bigarr[rebin(bigx,75,10000),transpose(rebin(bigy,10000,75))]

So, rebin(bigx,75,10000) replicates your 75 entries 10 000 times.
rebin(bigy,10000,75)) replicates the 10 000 entries 75 times. Now,
transpose(rebin(bigy,10000,75)), well, transposes it so the two

subscript now have the same dimensions. At last, for each corresponding index in both subscripts, the value of bigArr is returned.

Was i posted was working because there are the same number of elements in both subscripts... though the combination was wrong and therefore it returns the wrong result..

Sorry for the confusion!
Jean
