
Subject: Re: How to add each member of a pointarray to each member of another pointarray with arrayoperations?

Posted by [JD Smith](#) on Fri, 28 Jan 2005 16:14:01 GMT

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

On Fri, 28 Jan 2005 03:38:57 -0800, Bismarck wrote:

> Hi (again),
>
> same day, another question.
>
> I have two arrays with arbitrary sizes:
> a=[[1,0],[0,1]]
> b=[[5,5],[2,7],[5,3],[3,4]]
>
> I want an array where each member of b is added to each member of a.
> so c=make_a_wonder(a,b) should deliver
> c=[[5+1,5+0],[5+0,5+1],[2+1,7+0],[2+0,7+1],[5+1,3+0],[5+0,3+ 1],
> [3+1,4+0],[3+0,3+1]]
> I hope anyone understands my idea.
> So THX for any solutions and ideas.

How about:

```
na=n_elements(a)/2 & nb=n_elements(b)/2  
c=rebin(b,2,na*nb,/SAMPLE) + a[* ,lindgen(na*nb) mod 2]
```

A couple of notes: normally, REBIN is used to inflate dimensions of length 1, which just repeats the sub-array. Here we've used it to double (or na x) the length, so you *must* use /SAMPLE to avoid REBIN interpolating between points. To turn [[a,b],[c,d]] into [[a,b],[c,d],[a,b],[c,d],...], we use the traditional index arithmetic with MOD to wrap around. If you want an arbitrary width, instead of 2, just use:

```
sa=size(a,/DIMENSIONS)  
w=sa[0] & na=sa[1]  
nb=(size(b,/DIMENSIONS))[1]  
c=rebin(b,w,na*nb,/SAMPLE) + a[* ,lindgen(na*nb) mod w]
```

JD
