Subject: indexing
Posted by Jeremy Bailin on Thu, 02 Apr 2009 01:12:54 GMT
View Forum Message <> Reply to Message

I swear I've seen this come up within the past few months, but I can't find it, so here goes:

Let's say I have a 3D array - think of it as x,y,z if you want. I have a list of x,y pairs and I want to perform on operation on a given range in z (say z1:z2) and each x,y pair. For a simple example, let's say I just want them all up. So, if the following were my x,y pairs:
3 4
2 1
3 7
and my z range was 1:3 then I want

```
array[3,4,1]+array[3,4,2]+array[3,4,3]
+array[2,1,1]+array[2,1,2]+array[2,1,3]
+array[3,7,1]+array[3,7,2]+array[3,7,3]
```

Is there a simple representation for this? My standard solution, if pair is an npair x 2 array containing the x,y pairs, looks something like this:

```
nz=z2-z1+1
zindices=rebin(reform(z1+lindgen(nz),1,nz), npair,nz)
xindices=rebin(pair[*,0],npair,nz)
yindices=rebin(pair[*,1],npair,nz)
answer = total(array[xindices,yindices,zindices])
```

...but if nz and npair are large, generating all of those 2D index arrays is really wasteful. The following also works:

```
answer = total(array[pair[*,0],pair[*,1],z1:z2] * rebin(identity
(npair,npair,nz)))
```

but again generates 2 intermediate npair x npair x nz arrays that are wasteful if npair is large.

Any takers?

-Jeremy.