Subject: Indexing arrays with arrays
Posted by Conor on Fri, 21 Sep 2007 15:18:38 GMT
View Forum Message <> Reply to Message

A couple times I've asked about this. I sometimes would like to be able to index arrays with arrays without using for loops. For intance imagine I have:

```
data = findgen(20)

st = [5,2,10]
ed = [6,4,15]
```

I would like to do:

```
res = data[st:ed]
```

obviously, IDL doesn't allow this. The last time I posted about this, I got this response:

Well, if end_ind is always a constant offset from start_ind, it's easy just to contruct the index vector yourself:

```
 t=[end_ind[0]-start_ind[0]+1,n_elements(start_ind)]
 extract=res[reform(rebin(1#start_ind,t)
+rebin(indgen(t[0]),t),t[0]*t[1])]
```

If the start-end difference can be any variable amount, this is a problem for HISTOGRAM related to chunk indexing. See the HISTOGRAM tutorial

This was very helpful and got me started. After much pondering, I came up with this for the general case. Assume that st and ed are row arrays (as per the above example)

```
function array_index,st,ed

nst = n_elements(st)
ned = n_elements(ed)
diff = ed - st + 1

h = histogram(total(diff,/cumulative)-1,/
binsize,min=0,reverse_indices=ri)
i = ri[0:n_elements(h)-1]-ri[0]

arr = [fltarr(1,nst),transpose(diff)-1]
maxdiff = max(diff)
x = rebin(findgen(maxdiff),maxdiff,nst)/(maxdiff-1)
```

```
y = rebin(findgen(1,nst),maxdiff,nst)
int = interpolate(arr,x,y)
ref = reform(ceil(int),maxdiff*nst)
adds = ref[uniq(ref)]

return,st[i]+adds
```

Given the above example:

```
data = indgen(20)
st = [5,2,10]
ed = [6,4,15]

print,data[array_index(st,ed)]

; prints       5       6       2       3       4      10      11
12      13      14      15
```

I'm rather happy about the result. There's been a couple times when I really wish I had this available. Anyway, a couple questions:

1) Anyone see a better way to do this?
2) Anyone want to generalize this to n dimensions?
i.e. inds = array_index([[st1,ed1],[st2,ed2],[st3,ed3]])
