Subject: array index summations

Posted by H. Evans on Fri, 18 Dec 2009 11:07:23 GMT

View Forum Message <> Reply to Message

Hi,

I have a time-series data set, effectively a 3-vector (x,y,z), where the Z values are to be binned into a 2D array along X/Y.

Currently, I'm using the following, which loops over all of the data points (which could be numbered in the millions):

```
; calculate the X & Y indices into the 2D array
ix = FLOOR((x - self.xfit[0]) / self.xfit[1])
iy = FLOOR((y - self.yfit[0]) / self.yfit[1])

;    Range check the index calculations
ii = WHERE( (ix ge 0) AND (ix LT self.nx) AND $
                    (iy ge 0) AND (iy LT self.ny) )

IF ii[0] GE 0 THEN BEGIN
  ix = ix[ii]
  iy = iy[ii]
  zz =  z[ii]
  FOR i=0L,N_ELEMENTS(ix)-1 do BEGIN
    (*self.zsum) [ ix[i], iy[i]] += zz[i]
    (*self.zsum2)[ ix[i], iy[i]] += zz[i]^2
    (*self.nmap) [ ix[i], iy[i]]++
  ENDFOR
ENDIF
```

I want to get rid of the FOR loop. So I was thinking initially of something analogous to:

```
   x = INDGEN(10)
   y = INTARR(5)
   i = [0,0,0,0,0, 1,1, 2,3,4]   ; one for every element of X
   y[i] += x
```

where y would then become:

```
   y = [ 10, 11, 7, 8, 9]
```

with the assumption that IDL would do a "loop" over the indices accumulating the results in y at the appropriate indices, or rewritten in C:

```
   for (j=0; j<10; j++) {
      y[i[j]] =+ x[j]
   }
```

This, of course, doesn't happen. Another option is to use the histogram with the TOTAL function to solve the binning, but this then still implies looping over the nX*nY bins, which could also be large (~100*600). Or:

```
h = HISTOGRAM( i, reverse_ind=ri)
for j=0L,N_ELEMENTS(h)-1 DO $
   IF (h[j] GT 0) THEN $
      y[j] = TOTAL( x[  ri[ri[j]] : ri[ri[j+1]-1]  ])
```

But this still yields a FOR loop.

Also, is it possible to provide weights for the entries to the histogram, rather than just counting the number of entries in the bins. So for each bin it would return SUM(weights[i]) rather than N_ELEMENTS(i), where I is the list of indices of the data points that went into the bin (or i[j] = ri[ri[j]] : ri[ri[j+1]-1] in the above example)

Ideas?

Thanks,
Hugh