Subject: Optimizing code for faster calculation
Posted by Kenneth D on Thu, 13 Mar 2014 06:33:31 GMT
View Forum Message <> Reply to Message

I've been looking at this block of code now for... ever.

I've been editing a program created by my Adviser to reduce run time wherever possible. So far I've reduced the run time by nearly half, and I'm trying to juice any performance I can get from absolutely anywhere. My final project will use an array roughly 17,000 by 17,000. And I have to iterate through the program at least 17,000*10 times. If I'm lucky it won't take a month to process my data-sets now. This code is about all I have left to work with:

```
 exceed_subs = where(min_rmse GT rmse_threshold, counter)
 if counter GT 0 then modeled_class(exceed_subs) = "unmodeled"
```

min_rmse is an array Float[200], such as [0.347272, 0.312437, 0.360164,...]
rmse_threshold = 0.025
modeled_class is an array String[200], such as ["soil","quag","soil","grass",...]

The code find the locations where min_rmse is greater than a threshold value, and replaces those index locations in the string array (modeled_class) with "unmodeled".

This may well be the most efficient way to do this (this code will run a minimum of 17,000 times) but a look at the histograms page at Exelis:
http://www.exelisvis.com/docs/HISTOGRAM.html

shows:
For example, make the histogram of array A:

```
H = HISTOGRAM(A, REVERSE_INDICES = R)
;Set all elements of A that are in the ith bin of H to 0.
IF R[i] NE R[i+1] THEN A[R[R[I] : R[i+1]-1]] = 0

;The above is usually more efficient than the following:
bini = WHERE(A EQ i, count)
IF count NE 0 THEN A[bini] = 0
```

Which looks so similar to what I'm trying to do. I tried to implement this with no luck (maybe because strings?). Is there anything else I can do? That is, besides taking out iterations, they simply must be there to do what I need.