Subject: Re: Optimizing loops
Posted by MarioIncandenza on Tue, 08 Dec 2015 19:05:07 GMT
View Forum Message <> Reply to Message

On Wednesday, December 2, 2015 at 11:04:33 AM UTC-8, sam.t...@gmail.com wrote:
> Hello all! I am working with a large number of satellite data files. The files are quite big, but I've never had this much trouble working with them before, even while doing similar processing.
>
> The basic program structure that's the slowest is below; it also just dominates the processing power as it enters this loop. I need to do this for each file I process.
>
```
>     ; we want to calculate the percent of each satellite pixel
>     ; covered by land
>     land_perc = FLTARR(N_ELEMENTS(field1))
>     FOR j = 0, N_ELEMENTS(field1)-1 DO BEGIN
>        dist = (((land_lon - sat_lon[j])*COSD(land_lat))^2. $
>              + (land_lat - sat_lat[j])^2.)^0.5*111.12
>
>        ind14 = WHERE(dist LE 14.)
>        land14 = land_mask(ind14)
>        landy = WHERE(land14 EQ 0, landy_cnt)
>        land_perc[j] = FLOAT(landy_cnt)/FLOAT(N_ELEMENTS(land14))*100
>     ENDFOR
```
>
> If anyone has optimization suggestions, please let me know! Thanks :)

So you have these 2D arrays: LAND_LON, LAND_LAT, LAND_MASK
And these 2D (or 1D) arrays (different dims): SAT_LON, SAT_LAT
You want to calculate the fraction of land in a 14km radius from each satellite pixel. This can be done analytically using these steps:
1) Calculate binary land/water us BLAND=(LAND_MASK NE 0);
2) Project binary land mask BLAND to 1km (or 250m or whatever) rectangular grid (MAP_PROJ_INIT(), MAP_PROJ_FORWARD(),INTERPOLATE(INTERPOL());
3) Calculate fractional "land within 14km":
```
 KERNEL=sqrt((rebin(findgen(29),29,29)-14)^2+(rebin(findgen(1 ,29),29,29)-14)^2) ge 14; binary +/-14km
FLAND14_1KM = CONVOL(FLOAT(BLAND_1KM),KERNEL,/EDGE_TRUNCATE)
```
4) Project satellite LON/LAT onto 1km grid (MAP_PROJ_FORWARD);
5) Sample FLAND14_1KM at SAT_LON/SAT_LAT:
```
SAT_IX_1KM=INTERPOL(FINDGEN(N_ELEMENTS(X_1KM)),X_1KM,SAT_X_1 KM)
 SAT_IY_1KM=INTERPOL(FINDGEN(N_ELEMENTS(Y_1KM)),Y_1KM,SAT_Y_1 KM)
FLAND14_SAT = FLAND14_1KM[SAT_IX_1KM,SAT_IY_1KM]
```

This will give you a numerically robust solution in a single step for your entire granule.
NOTE 1) This is memory-intensive and if you really need a global grid at 250 meters you might not have enough memory for it.
NOTE 2) If you use a global grid, the answers near the poles and dateline will be suspect.