Subject: Astronomys` Sixth Neighbour Needs Help
Posted by touser2001 on Thu, 24 Jul 2003 20:40:40 GMT
View Forum Message <> Reply to Message

I have to first say hello to everyone!! I am new to this (and all) newsgroup(s).
I had a small IDL class and several months of on-off IDL work in making astronomy-related programs.
My latest program needs some expert help and thus I am writing this post in hope of a small discussion to enlighten me. I sincerely apologize if the post is too long!!

Basically, the program reads a 2-column n-line file (n is number of stars in field, the columns are the position of the stars, Right Ascencion and Declination which are similar to latitude and longitude). My objective is to obtain, for each star, the distance to its 6th neighbour. Simply, Right ascencsion is x and Declination is y on a xy grid.

What I am doing is:

```
k=0
while k lt u do begin                ; I know the number of stars
(lines) which is equal to "u"
 i=0              ;
 while i lt u do begin  ;
  dxki = (x(k) - x(i))   ;this variable is the RA distance between
point i
                ;and the kth point in the array

  dyki = (y(k) - y(i))   ;this variable is the Dec distance between
point i
                ;and the kth point in the array

  dki = sqrt (dxki^2 + dyki^2) ;Now I define the distance, in 2-d
space, between the two objects i and k

  D2(k,i) = dki  ;This array contains all the distance data for all
the stars
     ;The reference star is the column number (k)
  i = i+1   ;for all the values of i
 endwhile   ;
k=k+1
endwhile
```

The above loop calculates the distance from star 1 to 2, 1 to 3 etc etc and then 2 to 1, 2 to 2. I note that distance 1 to 2 is the same as 2 to 1 so a first improvement would be to say that once 1 to 2 is done it need to do 2 to 1 (and this for 3 to 1 etc) ... can I

implement that in the loop? Or any ideas as to improve this distance calculation?

The loop gives me an array whose dimension is u x u. This is a huge array when the star file is 10,000 stars... Which brings me to the second critical stage: calculating the sixth neighbour by first sorting, in ascending order, the distances in each column.
I did:

```
n=0
while n lt u do begin
D2n = D2(n,*)            ; this is so that I am sorting column by
column

 for j=0, n_elements(D2n)-2 do begin  ;
  for l=j+1, n_elements(D2n)-1 do begin ;
   if D2n[l] lt D2n[j] then begin ;This just arranges distances in
increasing order.
   temp=D2n[j]   ;
   D2n[j]=D2n[l]   ;
   D2n[l]=temp   ;So the array that I get is now the array of
distances
   endif    ;to the nearest neighbours, in order of the neighbour
number!
  endfor     ;
 endfor
D2(n,*)=D2n
n=n+1       ;
endwhile
```

Now that all columns are sorted I just go to line number six to get the sixth neighbour of each star!

I really don`t know if this is the best (quickest) method for organizing a column in ascending order... the second improvement would be to implement a different sorting algorithm. Note: the program does this for each of the 10,000 columns...

My problem is that for a field with 10,000 stars the program takes around 3 days (!) to run and it slows down all computer programs (unix system) (last time I ran the program it was killed by the system administrator since it took up too much memory).

I wish the department would hire some professional programmers!! Not that I don`t like IDL but I wouldn`t mind doing some astronomy too ;-)

Greatly appreciate and and all help!!!!!!!!!!!!
Bruno
PhD Student