Hi all,

   probably this is a knwon question with a known solution, but I
haven't figure it out yet. I have points in 3D, and I want to calculate
the distances between them, but not the distance between the same
point. This is because I want to know the total sum of the inverse of
those distances. The first thing it came to my mind was to do it with a
for loop, but of course, for large number of points (which is the
case), takes for ever (basically a nxn operation). This is the code I
did (very simple, brutal way):

```
; xp,yp,zp are the positions

xold=xp
yold=yp
zold=zp

np=n_elements(xp)

dist=fltarr(np)

for i=0,np-1 do begin

    xp[i]=0.
    esc1=where(xp ne 0,cc1)   ;removing the i-th particle
    newxp=xp[esc1]

    yp[i]=0.
    esc2=where(yp ne 0,cc2)
    newyp=yp[esc2]

    zp[i]=0.
    esc3=where(zp ne 0,cc3)
    newzp=zp[esc3]


 dist[i]=total(1./sqrt((newxp-xold(i))^2.+(newyp-yold(i))^2.+ (newzp-zold(i))^2.))

    xp[i]=xold[i]
    yp[i]=yold[i]   ; restoring the i-th particle
    zp[i]=zold[i]

endfor
```

Any idea of how to make this better? Thanks a lot!!

Andres

Subject: Re: Distances in 3D
Posted by biophys on Fri, 28 Oct 2005 01:35:38 GMT
View Forum Message <> Reply to Message

well, I don't see the point of using 3 where functions while they just
do exactly the same thing here. basically esc1=esc2=esc3, so you only
need 1 where() per loop. i guess shift function should work faster even
if you correct that redundant where functions. here's a one 1d example
for your reference.

```
x=randomu(systime(1),100)
y=x
t0=systime(1)
for i=0,99 do begin
  y[i]=total(1/abs(x[1:99]-x[0]))
  x=shift(x,-1)
endfor
print,systime(1)-t0
```

Subject: Re: Distances in 3D
Posted by panblosky on Fri, 28 Oct 2005 08:31:08 GMT
View Forum Message <> Reply to Message

Thanks, you are right. Using the shift function is faster. Thanks a
lot!!