In article
<4707b3ed-9e43-4863-977f-7fd091e62868@a4g2000prb.googlegroups.com>,
 Maxwell Peck <maxjpeck@gmail.com> wrote:

> Hi All,
>
> I have two sets of data points, inputa and inputb, inputa has many
> more data points than the other. I need to find the distance between
> each point in inputa and all the points in inputb (hopefully without
> loops). At the moment I am using parts of distance_measure.pro to
> generate the full set of distances and subsetting indexes as required.
> The code follows. I can't see an easy way of generating the indexes
> though so that only the pairs i want are calculated (i.e. not
> calculating between inputb/inputb or inputa/inputa indexes).
>
> inputa and inputb can be very very large so I don't think i can use a
> matrix approach to do it vectorially and looping seems awfully slow.
>
> An alternative approach or other suggestions would be appreciated.
>
> Regards
>
> Max
>
>
> inputa = findgen(2,10)
> inputb = findgen(2,2)
>
> t = [[inputa],[inputb]]
> m=n_elements(t)/2
> n = m*(m-1)/2
>
> ii = 0L
> index0 = LINDGEN(m - 1) + 1   ; work array
> index1 = LONARR(n, /NOZERO)
> index2 = LONARR(n, /NOZERO)
>
> for i=0,m-2 do begin
>   n1 = m - (i+1)
>   index1[ii:ii+n1-1] = i
>   index2[ii] = index0[0:n1-1] + i
>   ii += n1
> endfor
>

```
>  diff = abs(t[*,index1] - t[*,index2])
>  res = sqrt(TOTAL(diff^2, 1))
```

If I understand you right, don't you want to do this?

```
na = 10
nb = 2
xa = FINDGEN(na)
ya = FINDGEN(na)
xb = FINDGEN(nb)
yb = FINDGEN(nb)

dist = FLTARR(na, nb)

FOR i = 0, nb-1 DO BEGIN
   d = SQRT((xb[i] - xa)^2 + (yb[i] - ya)^2)
   dist[0,i] = d
ENDFOR
```

I changed the notation to make the indexing simpler.

This computes the Cartesian distance between all pairs of points in a and b.

You should loop over the smaller of na or nb.

Ken Bowman