

```

c      +-----+
c      + subroutine oibox : OI analysis step (for a selected box) +
c      +      A = B + Pg * (O - HB) / (P + D) +
c      +      A - analysis +
c      +      B - background (here use mean for simplicity) +
c      +      D - obs error co-variance matrix +
c      +      Pg - b.g. error co between ana points and obs points +
c      +      P - b.g. error co between obs points +
c      +-----+
c      subroutine oibox(ipar,nbox,lfgs,
+      idim,jdim,var,fgs,
+      lat,lon,dlat,dlon,clat,nana,latana,lonana,
+      valmean,valana,skip,valfgs,wl,coi,sdevfg,gscale,rmax)

      implicit none
c      ----- basic input -----
      integer nana
      logical lfgs,skip(nana)
      integer ipar,nbox,idim,jdim
      real lat,lon,dlat,dlon,clat
      real latana(nana),lonana(nana),valmean,valana(nana),
+      valfgs(nana),wl(nana),coi(nana,nana),
+      sdevfg,gscale,fgs(idim,jdim),rmax
c      ----- basic output -----
      real var(idim,jdim)
c      ----- work arrays -----
      integer i,j,k,iana,jana
      real varian,flat,flon,clon,dis2
      real xdist,ydist,f,r,omega,ro,dpx,dpy,t2,rmax2
      real pi

      pi = asin(1.)*2.

      if(nbox.eq.1) then
iana = idim
jana = jdim
      else if(nbox.eq.9)then
iana = 11
jana = 11 ! hard-coded
      else
write(6,*) 'nbox can only be 1 or 9'
stop
endif

      if(.not.lfgs) then ! use mean as fgs
do i=1,nana
valfgs(i) = valmean
enddo
endif
do i=1,nana
wl(i) = 0.
do j=1,nana
if(.not.skip(j)) ! only when data is good
+      wl(i) = wl(i) + (valana(j)-valfgs(j))*coi(j,i)
+      ( O - HB ) / (P+D)
enddo
enddo

varian = sdevfg*sdevfg
rmax2 = rmax*rmax

c      the basic required box size is iana x jana

      if(.not.lfgs) then ! use mean as fgs
do j=1,jana
do i=1,iana
fgs(i,j) = valmean
enddo
enddo
endif
do j=1,jana
flat = lat + (j-(jana/2+1))*dlat
clon = clat*cos(flat*pi/180.)
do i=1,iana
var(i,j) = fgs(i,j)
c      -- B --
flon = lon + (i-(iana/2+1))*dlon
do k=1,nana
dis2 = ((flat-latana(k))*clat)**2+((flon-lonana(k))*clon)**2
if(dis2.le.rmax2)
+      var(i,j) = var(i,j)+varian*exp(-dis2/gscale**2)*wl(k)
c      ---- Pg -----
enddo
enddo
enddo

      return

      end

```