Document NWPSAF-MO-TR-023

Version 1.0

12 October 2009

GOME-2 and IASI collocation

N. C. Atkinson

Met Office, Exeter, UK

GOME-2 and IASI collocation

This documentation was developed within the context of the EUMETSAT Satellite Application Facility on Numerical Weather Prediction (NWP SAF), under the Cooperation Agreement dated 1 December, 2006, between EUMETSAT and the Met Office, UK, by one or more partners within the NWP SAF. The partners in the NWP SAF are the Met Office, ECMWF, KNMI and Météo France.

Change record			
Version	Date	Author / changed by	Remarks
1.0	12.10.09	N C Atkinson	

GOME-2 and IASI collocation

Background

The GOME community would like to perform retrievals of ozone and/or trace gases simultaneously from IASI and GOME-2. Archived IASI and GOME-2 level 1B files, in native EPS format, are available but there is no readily-available method for collocating the two instruments. Noting that AAPP performs collocations for AMSU-to-IASI and AMSU-to-HIRS, this report discusses the issues for collocating IASI and GOME-2.

Scan characteristics

The scan characteristics of the two instruments are compared in the following table:

	Spots per scan	Scan time (sec)	Time per spot (sec)	Forward scan viewing time (sec)
IASI	30×4	8	0.216	6.48
GOME-2	24	6	0.1875	4.5

GOME-2 actually performs a slow forward scan and a rapid reverse scan, but we ignore the reverse scan. Also, some GOME-2 channels use longer integration times. IASI views 4 detectors simultaneously at each of the 30 scan positions. The scan directions are the same for the two instruments (left to right facing in the direction of satellite motion).

A major difference between the two instruments is that the scan speed for GOME-2 is not uniform with scan angle: it scans slower at the edges of the swath in order to give constant sampling distance on the earth's surface – approximately 80km. The along-track distance is 40km. Also, the direction of the GOME scan can be tilted in order to allow the instrument to view the polar regions.

The size of the IASI spots is 6km at nadir.

It is not clear from the documentation whether the GOME scan cycle is synchronised to the 8 second cycle of IASI, AMSU and MHS. Therefore some data samples were examined in order to try to establish this.

Reading the data

Data were obtained from the UMARF – level 1B GOME and 1C IASI in Native format. To avoid large files a time subset of about 8 minutes was used.[1]

EUMETSAT provide IDL readers for the EPS formats: see www.eumetsat.int Home -> Access to Data -> User Support -> Useful Programs & Tools.

Code for extracting the geolocation data from IASI and GOME files is shown at Appendix 1.

In Appendix 1, the entire dataset is read into memory in the "read_eps" call. If computer memory is an issue it is possible to store only the fields that are of interest (see EPS Product Reader User Manual).

[1] The subsettting process revealed a bug in the UMARF, in which the ASCII portion of the Generic Record Header was incorrectly formatted; a temporary fix was implemented. EUMETSAT subsequently fixed the bug.

Geolocation comparison

Figure 1 plots IASI spots on top of the GOME fields of view (FOV), as defined by the 'corner' latitude/longitude values in the GOME 1B file. We see that the number of IASI spots present within a GOME FOV is typically 4, but can be as large as 8 (near the centre of the scan) or as small as 2 (near the edge – e.g. FOV 3). Note the overlap of the GOME FOVs near the edges of the scan in the along-track direction – because of the increased distance from satellite to earth.

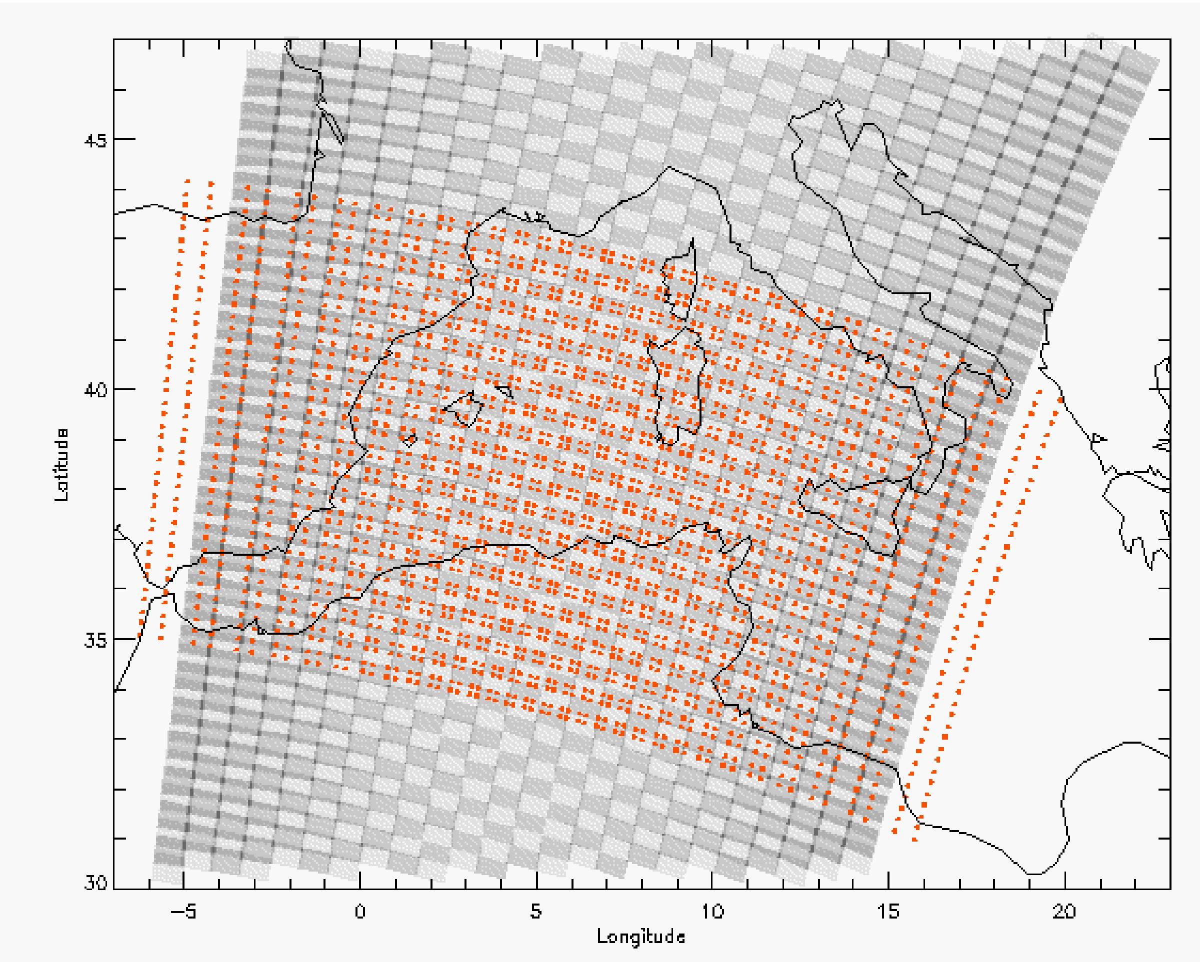

Figure 1: IASI spots (red) plotted on top of GOME fields of view (grey chequerboard pattern) over the Mediterranean. Regions where the GOME FOVS overlap are shown in a darker shade.

Figure 2 plots the time differences between each GOME scan (first spot) and the nearest IASI scan (first spot). As anticipated there is a 4-scan line (24 second) cycle. Expanding on the lowest point in the plot, there is some evidence of a drift.

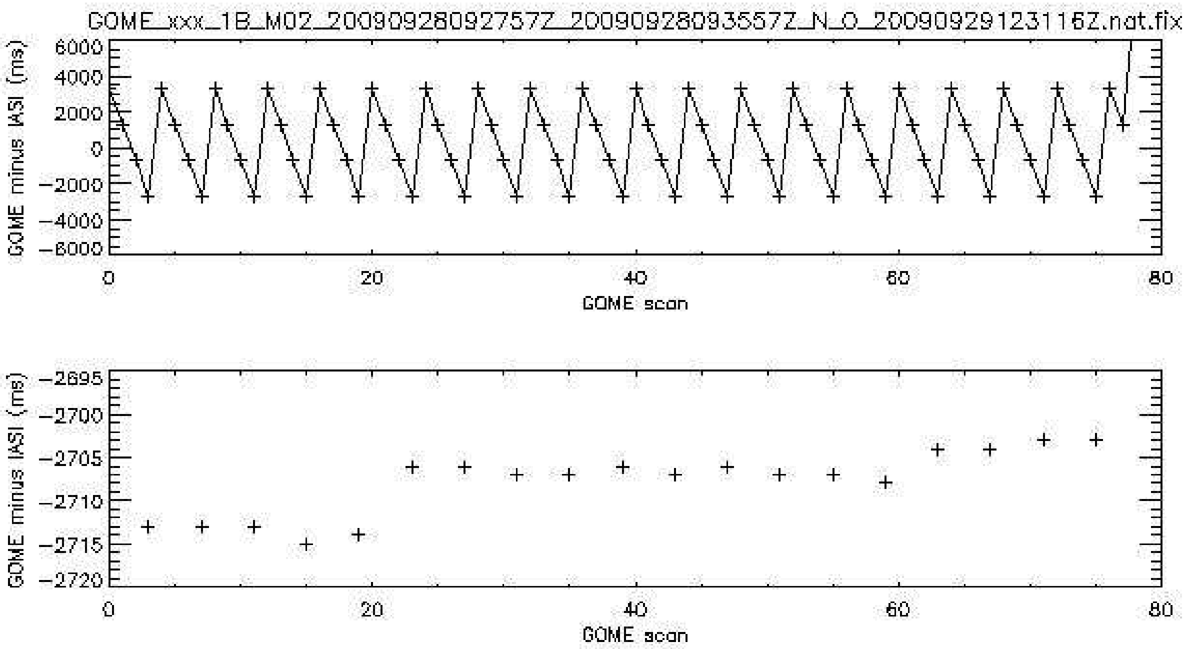

Figure 2: Time differences between each GOME scan (first spot) and the nearest IASI scan (first spot)

Further data samples were examined:

- The full orbit corresponding to Figure 2 showed a maximum GOME-AMSU time difference variability of 15ms (noting that IASI and AMSU are known to be synchronised)
- However, looking at a portion of data from the following day, the GOME-IASI time difference had shifted by 150ms.

We conclude that there is no strong evidence that the GOME and IASI scan cycles are synchronised. Even if they were, we would still have to identify the position within the 24 second cycle of each GOME scan. Therefore it is safest not to rely on scan synchronisation.

Implications for re-mapping

The AAPP mapping routine (LUTMAP) is unsuitable for use with GOME for several reasons:

- Apparent lack of synchronisation of GOME and IASI scan cycles
- GOME scan is not uniform with scan angle
- GOME has alternative viewing angle offsets (e.g. polar viewing mode)
- Adding a GOME data format reader to AAPP would be difficult

Therefore a standalone program is preferred, with the re-mapping parameters computed at run time for the particular files being analysed.

One decision that needs to be made by the user is whether to use the “nearest neighbour” IASI spot for each GOME FOV or whether to average spectra for more than one spot – potentially up to 8 spots (see Figure 1). In either case, IASI spots flagged as bad quality (based on GQisFlagQual) should be excluded.

To select matching IASI spots, three approaches could be considered:

1. Use latitude and longitude values. In the case of GOME this could either be the centre co-ordinates of the FOV (used to plot Figure 1) or (better) the co-ordinates of the four corners.
2. Use scan angle and time. This is the method used in AAPP.
3. Use time (and possibly scan angle) for a coarse estimate, then use latitude and longitude to refine the collocation.

Method 3 appears the most promising: use the mid-scan times to identify a pair of IASI scan lines which could potentially map onto each GOME scan (as in Figure 2). Then for each GOME FOV use the lat/lon values to determine the collocated IASI spots. It is prudent to convert lat/lon to x/y/z co-ordinates in order to avoid problems at the poles and at 180° longitude.

The best way of handling the data will depend to a large extent on the way that the data are to be presented to the level 2 processing suite. For example:

- We could produce a collocation file containing the IASI scan line and spot numbers to be used for each GOME scan. Then the level 2 suite would need to read in the original IASI and GOME 1B files together with the collocation file.
- We could instead save the IASI spectra themselves in an output file, possibly together with the necessary information extracted from the GOME file.

Software demonstration

The procedure ***collocate_gome_iasi.pro***, together with its associated utility subroutines, is shown in Appendix B. The procedure generates a collocation look-up table for IASI mapped to GOME, using method 3 of the previous section. The output file contains, for each GOME FOV, a list of the IASI spots that are collocated, together with the distance of each from the centre of the GOME FOV. For example:

```
   0   0   4     GOME_FOV  GOME_line  Number_of_collocations
2,1,7   6.6       IASI_detector,IASI_spot,IASI_line  Distance(km)
3,1,8  28.9
1,1,7  56.7
0,1,8  63.4
   1   0   4
2,2,7  45.7
1,1,7  48.5
3,2,8  54.6
0,1,8  55.3
   2   0   2
1,2,7  13.3
0,2,8  30.6
```

The collocations are ordered in increasing distance from the GOME FOV centre, so the user is able to limit the number of IASI spectra to be used. An IASI spot is only included if it falls within the boundary of the GOME FOV, as defined by the co-ordinates of the four corners.

In terms of computer time the largest single operation is the reading in of the GOME file. On a Met Office Linux PC it takes about 100 seconds to ingest a full-orbit (950Mb) GOME level 1b file using the "ReadProduct" method described in the EPS Product Reader User Manual. It is possible that the "ReadField" method might be quicker, as it does not store the whole product in memory, but it is not obvious from the manual how this method should be used for GOME as the data structures for GOME are more complicated than those in the AVHRR illustration. It would probably be necessary to seek the advice of EUMETSAT. The time taken to compute the collocations and write the output file to disk is of the order 10% of the time to read in the 1B files.

Conclusion

The scan characteristics of GOME-2 and IASI have been compared and prototype software demonstrated for collocating them.

The GOME users are invited to provide feedback on the extent to which this meets their requirements.

Appendix A: IDL readers for GOME and IASI

```
pro read_eps_gome,infile,times,angles,centre_lats,centre_lons,corner_lats,corner_lons

  prod = read_eps(infile,/PRODUCT)
  prod->GetProperty,MDR6=mdr6

  nscans = n_elements(*mdr6)

;The format contains 32 pixels per scan but we are only interested in the
;forward scan, points 1 to 24

  times = lonarr(24,nscans)          ;milliseconds
  angles = dblarr(24,nscans)         ;degrees
  centre_lats = dblarr(24,nscans)    ;degrees
  centre_lons = dblarr(24,nscans)    ;degrees
  corner_lats = dblarr(24,4,nscans)  ;degrees
  corner_lons = dblarr(24,4,nscans)  ;degrees

  for iscan=0,nscans-1 do begin
    mdrstruct = *((*mdr6)(iscan))
    times(*,iscan) = (mdrstruct.geo_basic.utc_time.time)(1:24)

;Note that index 1 (of 6) seems to be the one we are interested in.

    angles(*,iscan) = $
      1D-6*(mdrstruct.geo_earth_actual.scanner_angle_actual)(1:24,1)
    centre_lats(*,iscan) = $
      1D-6*(mdrstruct.geo_earth_actual.centre_actual.latitude)(1:24,1)
    centre_lons(*,iscan) = $
      1D-6*(mdrstruct.geo_earth_actual.centre_actual.longitude)(1:24,1)
    corner_lats(*,*,iscan) = $
      1D-6*(mdrstruct.geo_earth_actual.corner_actual.latitude)(1:24,*,1)
    corner_lons(*,*,iscan) = $
      1D-6*(mdrstruct.geo_earth_actual.corner_actual.longitude)(1:24,*,1)
  endfor

  obj_destroy,prod
end

pro read_eps_iasi,infile,times,angles,lats,lons,quals

  prod = read_eps(infile,/PRODUCT)
  prod->GetProperty,MDR=mdr

  nscans = n_elements(*mdr)

  times = lonarr(30,nscans)        ;milliseconds
  angles = dblarr(30,nscans)       ;degrees
  lats = dblarr(4,30,nscans)  ;degrees
  lons = dblarr(4,30,nscans)  ;degrees
  quals = bytarr(4,30,nscans)

  for iscan=0,nscans-1 do begin
    mdrstruct = *((*mdr)(iscan))

    start_time = mdrstruct.grh.record_start_time.time
    times(*,iscan) = start_time + indgen(30)*216.0       ;216.0ms per FOV
    angles(*,iscan) = (indgen(30)-14.5)*3.333333         ;fixed scan angles

    lats(*,*,iscan) = 1D-6*reform((mdrstruct.GGEOSONDLOC)(1,*,*))
    lons(*,*,iscan) = 1D-6*reform((mdrstruct.GGEOSONDLOC)(0,*,*))
    quals(*,*,iscan) = mdrstruct.GQISFLAGQUAL
  endfor
  obj_destroy,prod

end
```

Appendix B: Collocation software and utilities

```
function inside_quadrilateral,xp,yp,zp,x,y,z

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; See whether a point (or points) xp, yp, zp lies within a quadrilateral ABCD:
;
;       A  C
;       B  D
;
; Input argument x = [xA,xB,xC,xD], similarly for y, z.
;
; Returns 1 if the point is within the area of the quadrilateral, otherwise 0
;
; 07/10/2009 NCA
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

npoints = n_elements(xp)

if npoints gt 1 then result=intarr(npoints) else result=0

BA = [x(1)-x(0),y(1)-y(0),z(1)-z(0)]
DC = [x(3)-x(2),y(3)-y(2),z(3)-z(2)]
CA = [x(2)-x(0),y(2)-y(0),z(2)-z(0)]
DB = [x(3)-x(1),y(3)-y(1),z(3)-z(1)]

for i=0,npoints-1 do begin

  pA = [xp(i)-x(0),yp(i)-y(0),zp(i)-z(0)]
  pC = [xp(i)-x(2),yp(i)-y(2),zp(i)-z(2)]

  prod1 = crossp(pA,BA)
  prod2 = crossp(pC,DC)

;The scalar product of prod1 and prod2 must be <0 if point is between the lines.

  if total(prod1*prod2) lt 0 then begin   ;repeat for the other pair of lines
    pB = [xp(i)-x(1),yp(i)-y(1),zp(i)-z(1)]
    prod3 = crossp(pA,CA)
    prod4 = crossp(pB,DB)
    if total(prod3*prod4) lt 0 then result(i) = 1
  endif

endfor

return,result

end



pro latlon_to_xyz,lat,lon,x,y,z

  d2r = !pi/180

  x = cos(lon*d2r)*cos(lat*d2r)
  y = sin(lon*d2r)*cos(lat*d2r)
  z = sin(lat*d2r)

end
```

```
;***************************************************************************
; collocate_gome.pro
;
; DESCRIPTION
;    IDL procedure to create a GOME-IASI collocation index file. Format of
;    output file is as follows:
;       GOME_spot GOME_scan Number_of_collocations
;       IASI_detector,IASI_spot,IASI_scan  Distance_from centre(km)
;           [repeat above for each collocation]
;
;    where the spot and scan numbers start at 0
;
; External procedures:
;    read_eps_iasi (uses EUMETSAT-supplied EPS readers)
;    read_eps_gome
;    latlon_to_xyz
;    inside_quadrilateral
;
; COPYRIGHT
;    This software was developed within the context of the EUMETSAT Satellite
;    Application Facility on Numerical Weather Prediction (NWP SAF), under the
;    Cooperation Agreement dated 16 December 2003, between EUMETSAT and the
;    Met Office, UK, by one or more partners within the NWP SAF. The partners
;    in the NWP SAF are the Met Office, ECMWF, KNMI and MeteoFrance.
;
;    Copyright 2009, EUMETSAT, All Rights Reserved.
;
; History:
; Version  Date       Comment
;
;  1.0    07/10/2009  Original code (Nigel Atkinson)
;
;***************************************************************************

pro collocate_gome_iasi,gfile,ifile,collocfile,diagnostics=diagnostics

;-------------------------------------------------
;Read in the GOME and IASI data in EPS format
;-------------------------------------------------

print,'reading ',ifile
read_eps_iasi,ifile,itimes,iangles,ilats,ilons,quals
print,'reading ',gfile
read_eps_gome,gfile,gtimes,gangles,glats,glons,cornerlats,cornerlons

print,'processing ...'
openw,1,collocfile

ispots = n_elements(itimes(*,0))
gspots = n_elements(gtimes(*,0))

i0 = reform(itimes(ispots/2,*))      ;time of centre spot
g0 = reform(gtimes(gspots/2,*))

iscans = n_elements(i0)
gscans = n_elements(g0)

;-------------------------------------------------
;For each GOME scan, find the two IASI scans that are nearest in time.
;-------------------------------------------------

nearestIASI = intarr(2,gscans)

for gscan=0,gscans-1 do begin
  tmin = min(abs(g0(gscan)-i0), minpos)
  nearestIASI(0,gscan) = minpos
  i1 = i0
  i1(minpos) = 86500000L   ;discard it
```

```
  tmin = min(abs(g0(gscan)-i1), minpos)
  nearestIASI(1,gscan) = minpos
endfor

;-------------------------------------------------
;Convert lat/lon to xyz
;-------------------------------------------------

latlon_to_xyz,ilats,ilons,ix,iy,iz
latlon_to_xyz,glats,glons,gx,gy,gz
latlon_to_xyz,cornerlats,cornerlons,cx,cy,cz

;-------------------------------------------------
;Create arrays to hold the IASI line/spot/detector numbers
;-------------------------------------------------

ispotnum = intarr(4,30,2)
ilinenum = intarr(4,30,2)
idetnum = intarr(4,30,2)
for idet=0,3 do for i=0,1 do ispotnum(idet,*,i) = indgen(30)
for ispot=0,29 do for i=0,1 do idetnum(*,ispot,i) = indgen(4)

;-------------------------------------------------
;For each GOME spot find which IASI spots are collocated. Initially look at all
;IASI spots in the candidate scan lines. Could be made more efficient if
;necessary by using the scan angles. Look at the quality flag also.
;-------------------------------------------------

for gscan=0,gscans-1 do begin

  if keyword_set(diagnostics) then begin
    if gscan eq 0 then begin       ;plot out the first scan
      plot,[0],/nodata,xr=[min(cornerlons(*,*,0)),max(cornerlons(*,*,0))],$
        yr=[min(cornerlats(*,*,0)),max(cornerlats(*,*,0))]
      for gspot=0,gspots-1 do begin
        oplot,cornerlons(gspot,[0,1,3,2,0],0),cornerlats(gspot,[0,1,3,2,0],0)
      endfor
    endif
  endif

  xp = ix(*,*,[nearestIASI(0,gscan),nearestIASI(1,gscan)]) ;range of IASI spots
  yp = iy(*,*,[nearestIASI(0,gscan),nearestIASI(1,gscan)])
  zp = iz(*,*,[nearestIASI(0,gscan),nearestIASI(1,gscan)])
  qp = quals(*,*,[nearestIASI(0,gscan),nearestIASI(1,gscan)])

  ilinenum(*,*,0) = nearestIASI(0,gscan)
  ilinenum(*,*,1) = nearestIASI(1,gscan)

  for gspot=0,gspots-1 do begin
    x = reform(cx(gspot,*,gscan))   ;co-ordinates of GOME corners
    y = reform(cy(gspot,*,gscan))
    z = reform(cz(gspot,*,gscan))

    inside = inside_quadrilateral(xp,yp,zp,x,y,z)

    pos = where(inside gt 0 and qp eq 0, num_collocs)

;-------------------------------------------------
;order the collocations according to the distance from the centre of the FOV
;-------------------------------------------------

    if num_collocs gt 0 then begin
      dist_km = 8461.0 * sqrt((xp(pos)-gx(gspot,gscan))^2 + $
                              (yp(pos)-gy(gspot,gscan))^2 + $
                              (zp(pos)-gz(gspot,gscan))^2)
      order = sort(dist_km)
    endif
```

```
;-------------------------------------------------
;write to the output file
;-------------------------------------------------

    printf,1,gspot,gscan,num_collocs,format='(3I4)'
    for colloc=0,num_collocs-1 do begin
      i = idetnum(pos(order(colloc)))
      j = ispotnum(pos(order(colloc)))
      k = ilinenum(pos(order(colloc)))

      printf,1,strtrim(i,2) + ',' + strtrim(j,2) + ',' + strtrim(k,2) + $
        string(dist_km(order(colloc)),format='(F6.1)')
      if keyword_set(diagnostics) and gscan eq 0 then $
        oplot,[ilons(i,j,k)],[ilats(i,j,k)],psym=(gspot mod 7) + 1
    endfor
  endfor
endfor

close,1

print,'created ',collocfile

end
```