

LYRA Flatfield Software: Relative Output as Function of Pointing

IED, 17 Aug 2006

The IDL program for this purpose (see below) asks for the coordinates of the off-pointing. The solar center is assumed to be $(x,y) = (0,0)$, where the first coordinate denotes the east-west dimension, $x < 0$ being east and $x > 0$ being west; the second coordinate denotes the south-north dimension, $y < 0$ being south and $y > 0$ being north.

The program then calculates the image of the Sun on the LYRA detector plane. The image is limited by the precision aperture, diameter = 3 mm, distance = 27.75 mm, and also limited by the view-limiting aperture, diameter = 6.4 mm, distance = 27.75 + 70.00 mm. The location of the two images is calculated with the help of a tangens-formula, the resulting real image is the intersection of the two.

The program then asks for the LYRA channel (1-3, 1-4,...,3-4) and reads the measured "homogeneity" values from the LYRA Calibration at BESSY (March 2006), i.e. the geometrically distributed relative responses. The solar image on the detector plane is weighted with this homogeneity. The resulting sum is divided by the result that would be delivered by (0,0) i.e. no off-pointing. The quotient can be called "relative" or "normalized response" of the channel.

This approach uses the documents

- "LYRA Opto-Mechanical Properties", Silvio Koller 06.02.2004
- "Off-pointings for LYRA" from P2SC wiki
- "LYRA" by Hochedez et al. ASR 2006

These sources contain differing values for the view-limiting aperture, 6.4 mm diameter TBC. They also contain differing values for the detector sensitive areas, but these do not matter since the actually measured values from the BESSY calibration campaign were used.

The approach is, so far, limited to the channels *-3 and *-4, since there are no measurements for channels *-1 and *-2. Could these be determined in a similar way?

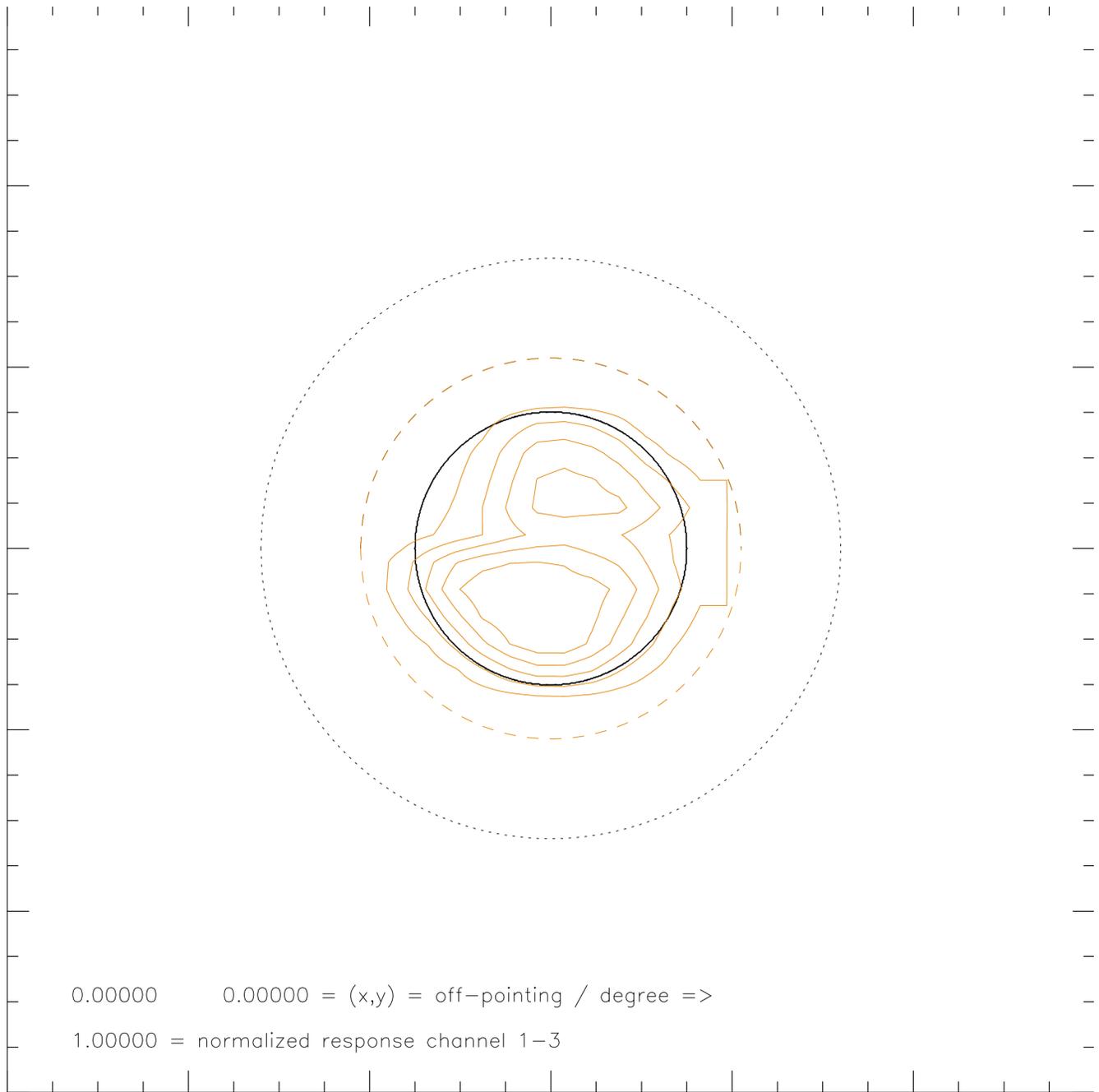
The orientation of the detector homogeneity measurements relative to their position on PROBA or their position to SWAP could not be determined. Is this possible?

One could think of several improvements to get better results for the "normalized response". But it has to be considered first if the greater exactness would be worth the effort:

- The extension of the Sun (0.5 degree) leads to a halfshadow on both sides of the border of the solar images; this is, so far, not implemented.
- The resolution used for the numerical calculation of the solar images' intersection is 0.01 mm and could be made smaller.
- The detector homogeneity measurements were taken at 0.3 mm resolution; for the numerical calculation the 13x13 value array had to be blown up 30 times this size; it was done using "congrid" with the "/interp" keyword for smoothing. Without /interp, there are different values. Which strategy is better?

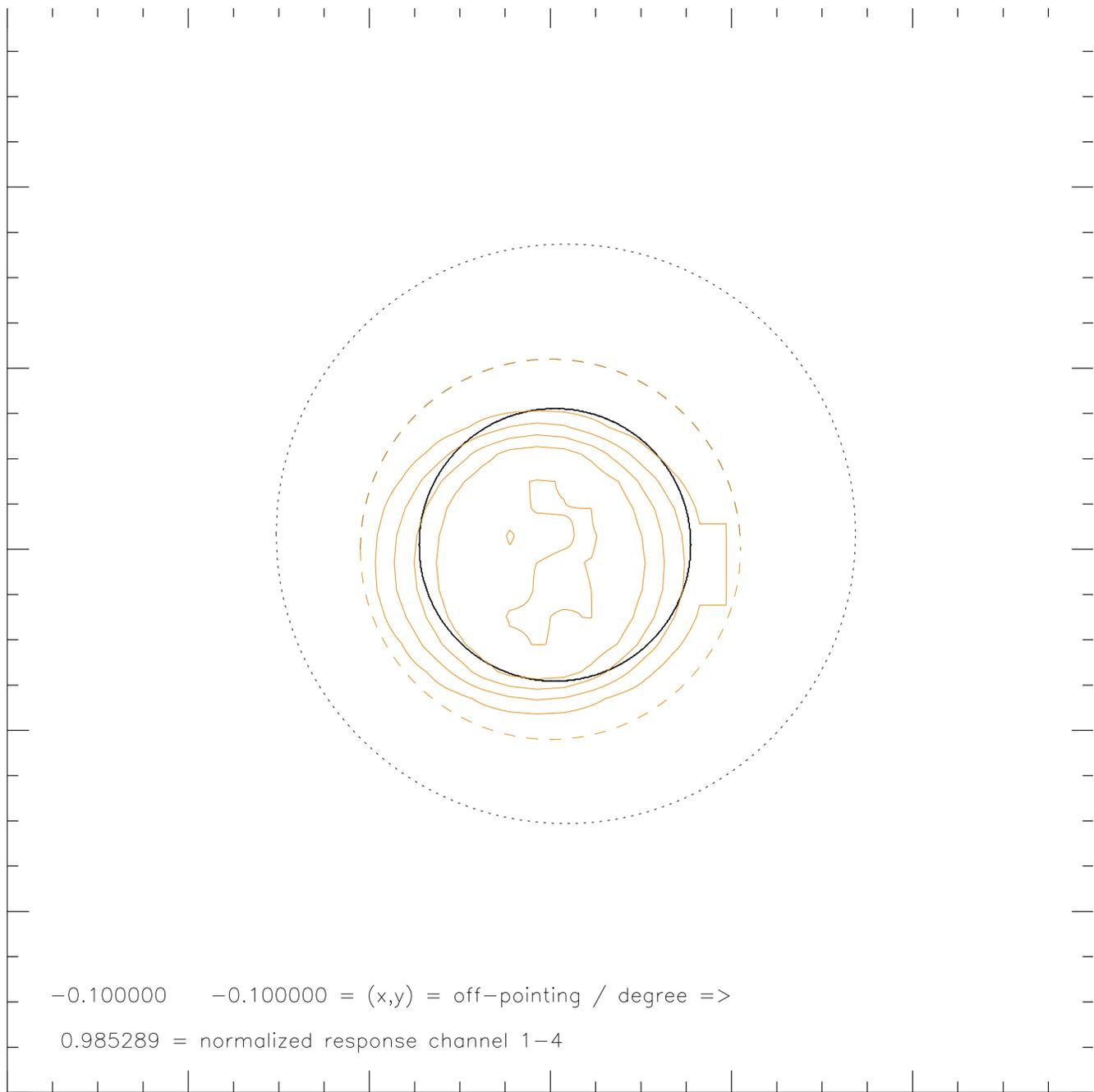
Would it make sense to produce an (x,y) -map of relative responses? What values of off-pointings (x,y) are to be expected?

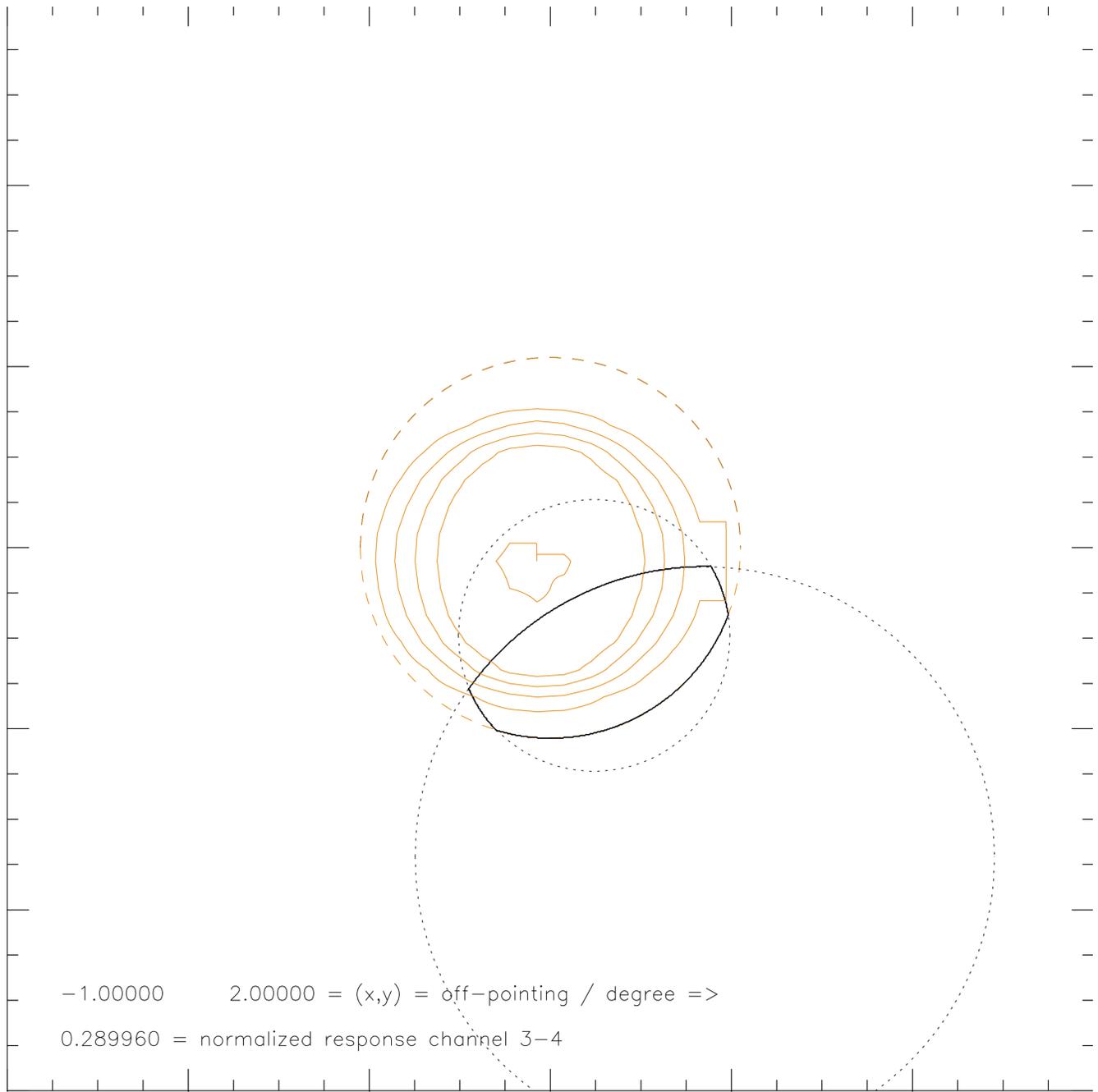
5 examples, IDL program listing: see below.

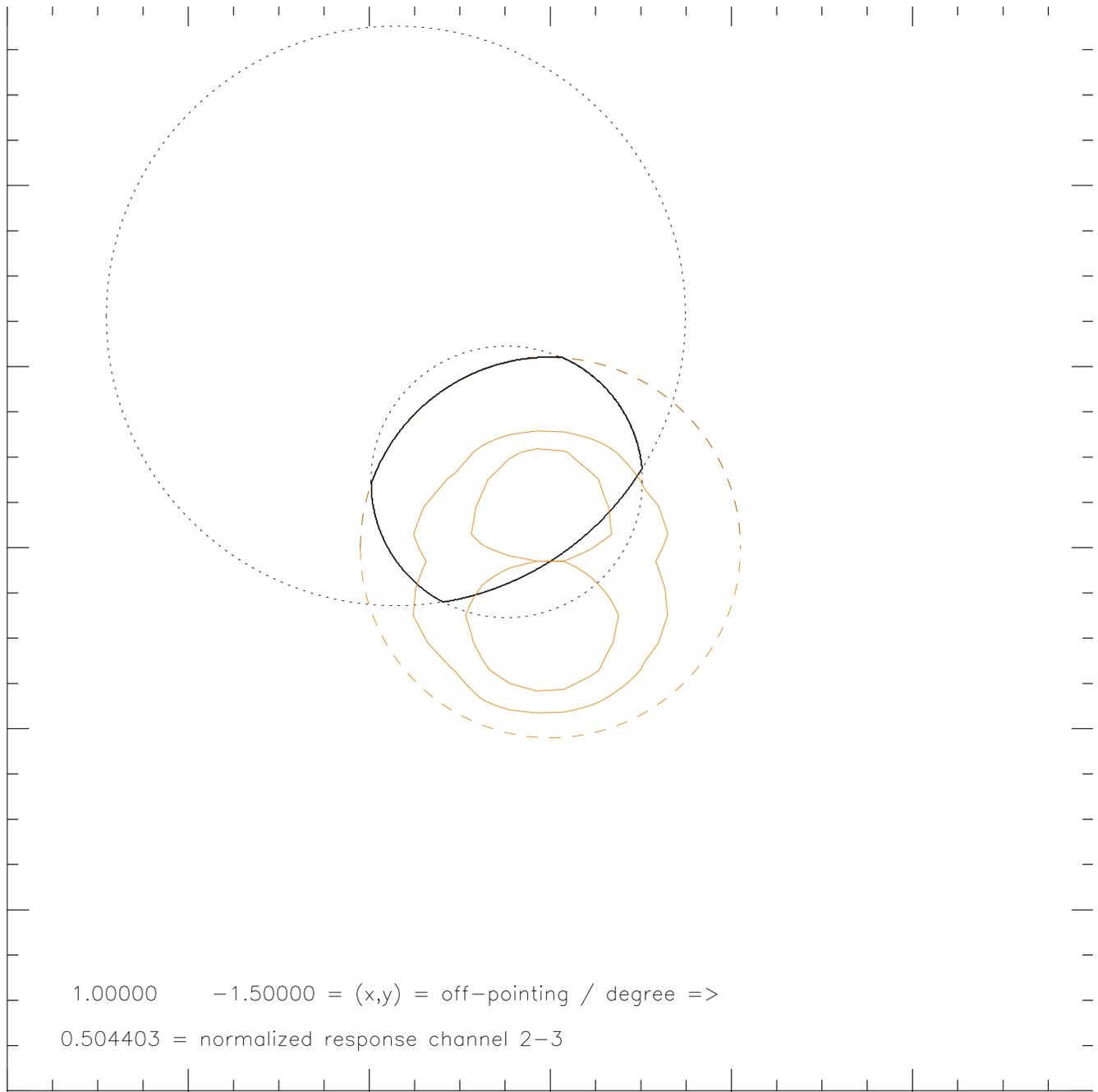


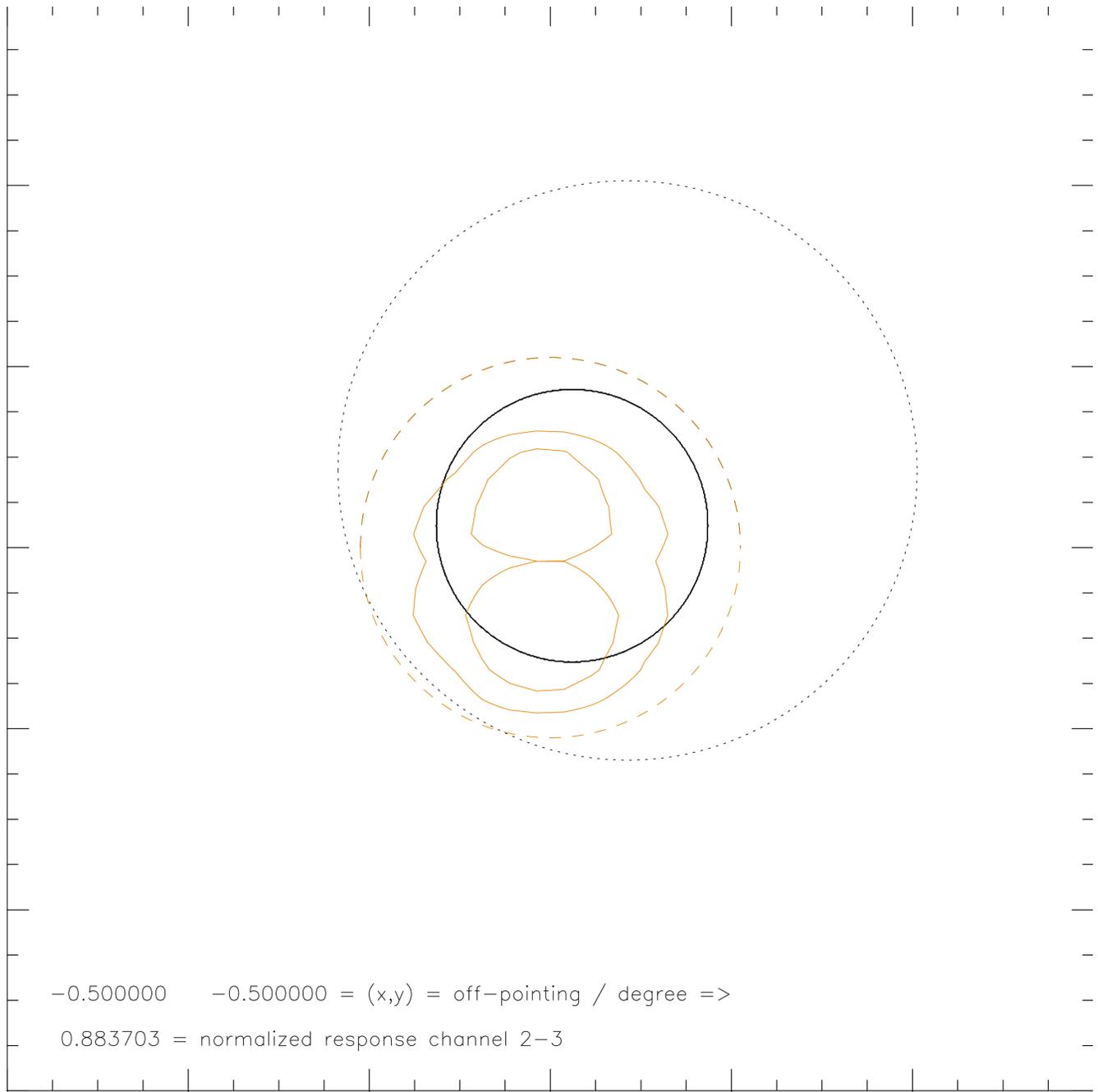
Caption for this and following figures

- 12 mm x 12 mm extension in detector plane
- red, dashed: detector limits
- red, solid: detector flatfield contour
- black, larger, dotted: image of Sun through 6.4 mm view-limiting aperture
- black, smaller, dotted: image of Sun through 3 mm precision aperture
- black, thick solid: intersection, resulting image of Sun in detector plane









```

; plot_detimff.pro
;-----
; IED 17 Aug 2006
;-----
; IDL program to calculate the image of the Sun on a LYRA detector with 4.2 mm
; diameter, given an off-pointing.
; Coordinates (x=east-west, y=south-north) of solar center = (0,0),
; coordinates of pointing in degrees east x<0, west x>0, south y<0, north y>0
; Aperture with diameter 3 mm at 27.75 mm distance,
; aperture with diameter 6.4 mm at 27.75 + 70.00 mm distance.
; Then the program asks for a LYRA channel (1-3,1-4,...,3-4), reads the
; measured homogeneity values, plots them, and calculates the normalized
; detector response.
;-----

chpr=0
loadct,3
read,'print .PS (Y=1) ? ',chpr
if (chpr eq 1) then begin
  set_plot,'ps'

device,/portrait,xoffset=1,yoffset=7,xsize=19,ysize=19,bits=8,/encapsulated,/color
endif else begin
  set_plot,'x'
  window,xsize=600,ysize=600,retain=2
endelse
xoff=0. & yoff=0.
read,'off-pointing in degree (east-west,south-north) #,# ? ',xoff,yoff
dist=2775.
dist2=dist+7000.
ximgoff=-dist*tan(xoff/180.*!pi)
yimgoff=-dist*tan(yoff/180.*!pi)
ximgoff2=-dist2*tan(xoff/180.*!pi)
yimgoff2=-dist2*tan(yoff/180.*!pi)
print,ximgoff,yimgoff
print,ximgoff2,yimgoff2
!xmin=-600 & !xmax=600 & !ymin=-600 & !ymax=600
ddet=420
rdet=ddet/2.
dpra=300
rpra=dpra/2.
dvla=640
rvla=dvla/2.
xdet=findgen(ddet+1)-rdet
ydet=sqrt(rdet*rdet-xdet*xdet)
plot,xdet,ydet,linestyle=2,charsize=0.01
loadct,4
oplot,xdet,ydet,linestyle=2,color=200
oplot,xdet,-ydet,linestyle=2,color=200
loadct,0
ximg=findgen(dpra+1)-rpra
ximg2=findgen(dvla+1)-rvla
yimg=sqrt(rpra*rpra-ximg*ximg)
yimg2=sqrt(rvla*rvla-ximg2*ximg2)
oplot,ximg+ximgoff,yimg+yimgoff,linestyle=1
oplot,ximg+ximgoff,-yimg+yimgoff,linestyle=1
oplot,ximg2+ximgoff2,yimg2+yimgoff2,linestyle=1
oplot,ximg2+ximgoff2,-yimg2+yimgoff2,linestyle=1
detimg=filtarr(1201,1201)
for i=0,ddet do begin
  xco=0>(round(xdet(i))+600)<1200

```

```

yco1=0>(round(-ydet(i))+600)<1200
yco2=0>(round(ydet(i))+600)<1200
detimg(xco,ycol:yco2)=detimg(xco,ycol:yco2)+1.
endfor
for i=0,dpra do begin
xco=0>(round(ximg(i)+ximgoff)+600)<1200
yco1=0>(round(-yimg(i)+yimgoff)+600)<1200
yco2=0>(round(yimg(i)+yimgoff)+600)<1200
detimg(xco,ycol:yco2)=detimg(xco,ycol:yco2)+1.
endfor
for i=0,dvla do begin
xco=0>(round(ximg2(i)+ximgoff2)+600)<1200
yco1=0>(round(-yimg2(i)+yimgoff2)+600)<1200
yco2=0>(round(yimg2(i)+yimgoff2)+600)<1200
detimg(xco,ycol:yco2)=detimg(xco,ycol:yco2)+1.
endfor
!xmin=0 & !xmax=0 & !ymin=0 & !ymax=0
detimg(0,*)=0
detimg(1200,*)=0
detimg(*,0)=0
detimg(*,1200)=0
detimg(where(detimg lt 3))=0.
detimg=detimg/3.
ok=''
read,'ok ? ',ok
contour,detimg,/noerase,charsize=0.01

close,1
ch=''
print,'13 = Aluminium + MSM11
print,'14 = Zr (300nm) + AXUV20D
print,'23 = Aluminium + MSM15
print,'24 = Zr (150nm) + MSM19
print,'33 = Aluminium + AXUV20B
print,'34 = Zr (300nm) + AXUV20C
read,'channel ## ? ',ch
if (ch eq '13') then norm00=48709.4
if (ch eq '14') then norm00=58950.9
if (ch eq '23') then norm00=64658.0
if (ch eq '24') then norm00=68075.2
if (ch eq '33') then norm00=56453.5
if (ch eq '34') then norm00=58632.8
;if (ch eq '13') then norm00=48450.0
;if (ch eq '14') then norm00=60330.7
;if (ch eq '23') then norm00=67168.0
;if (ch eq '24') then norm00=70798.1
;if (ch eq '33') then norm00=57246.7
;if (ch eq '34') then norm00=59933.0
ffhom=fltarr(13,13)
openr,1,'/projects/lyra/LYRA_Calibration_Data/BESSY_GI/Bessy_GI_March2006/homogenei
ty/FF3Dch'+ch+'.txt'
inline=fltarr(13)
readf,1,inline
for i=0,12 do begin
inline=fltarr(14)
readf,1,inline
ffhom(*,i)=inline(1:13)
endfor
detff=fltarr(1201,1201)
ffhom30=congrid(ffhom,390,390,/interp)
;ffhom30=congrid(ffhom,390,390)

```

```
detff(600-195,600-195)=ffhom30
loadct,4
contour,detff,/noerase,color=200,xstyle=4,ystyle=4,charsize=0.01
loadct,0
detimgff=total(detimg*detff)
print,detimgff,'=> normalized: ',detimgff/norm00
chstr=strmid(ch,0,1)+'-'+strmid(ch,1,1)
xyouts,0,100,string(xoff)+string(yoff)+' = (x,y) = off-pointing / degree =>'
xyouts,0,50,string(detimgff/norm00)+' = normalized response channel '+chstr
if (chpr eq 1) then begin
  device,/close
  ;spawn,'lpr -P lpsidc idl.ps'
  set_plot,'x'
endif

loadct,0
end
```