COG IMAGING DETECTOR USER GUIDE by Jim Hall and Ed Lent

A new capability

Marie-Anne Descalle

May 14, 2020

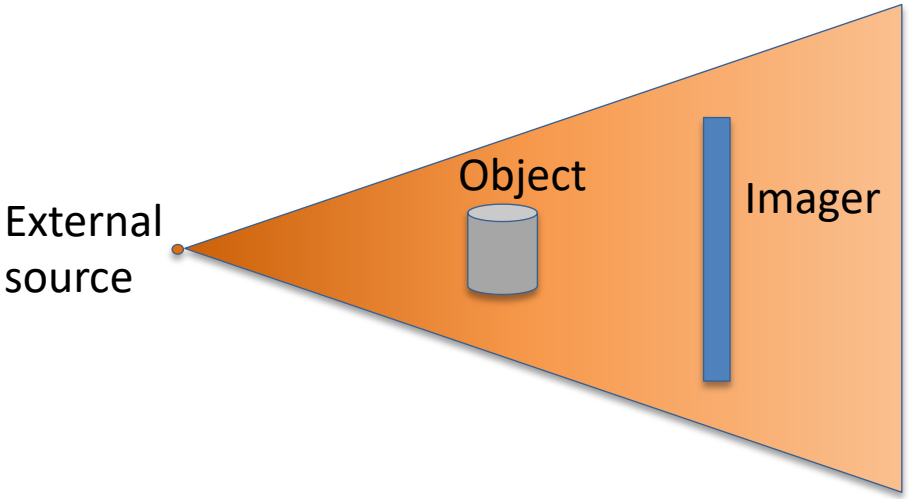
LLNL-PRES-843321

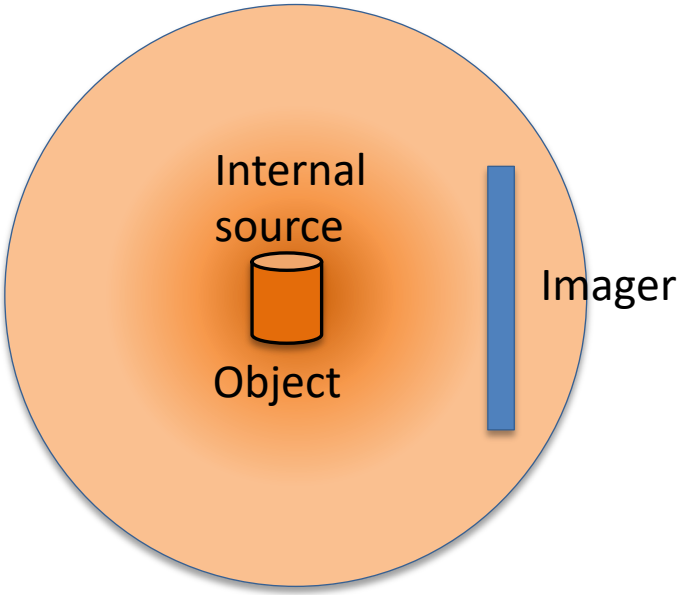
This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC

Lawrence Livermore National Laboratory

COG Imaging detector

- Code version: /usr/gapps/cogdev/COG11.3LOON.20.5.4
- Available on LC rztopaz, quartz
- Applications: Transmission and emission imaging

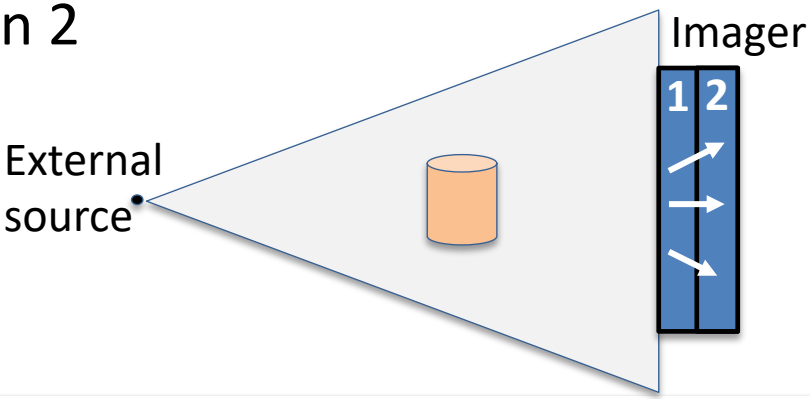


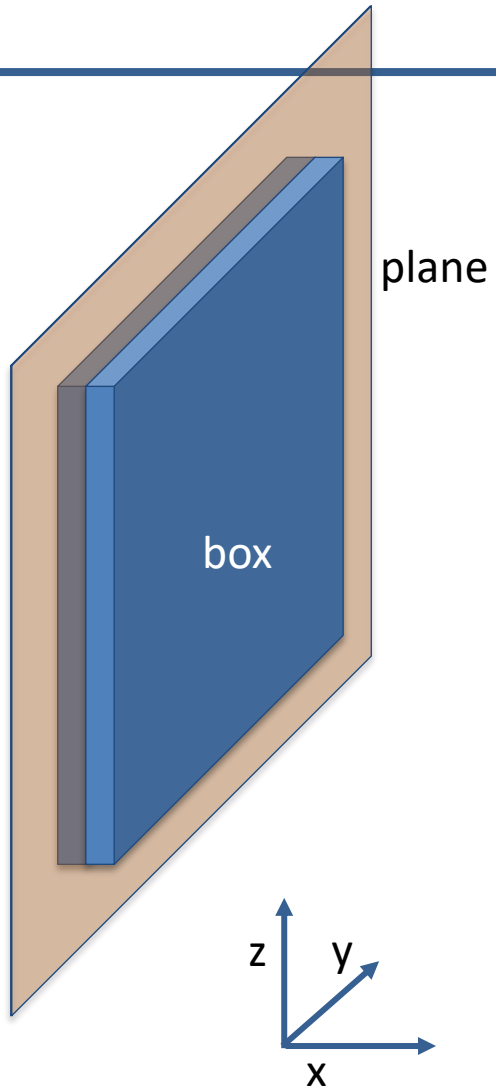


Imaging detector definition

Surface, Geometry, Assign blocks

- Define the size and shape of your imaging detector
  - Thin parallelepiped ('box' in COG)
- Define a plane splitting the detector in 2 regions
- The Imaging detector tallies particles crossing the plane from region 1 to region 2

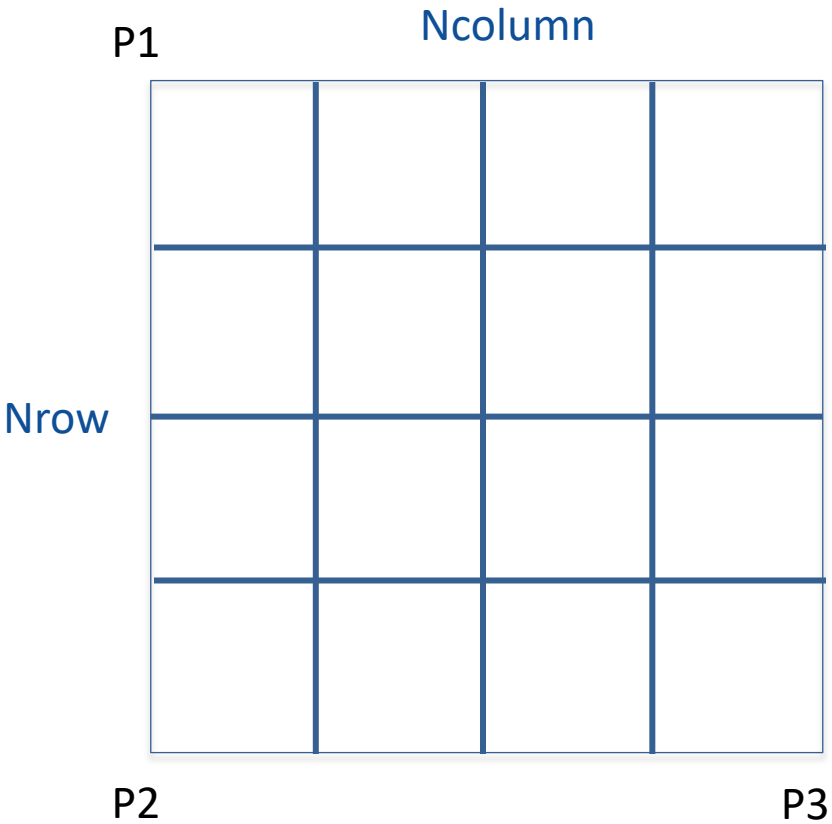




Imaging detector definition

Detector block

- Define image resolution
  - The number of pixels or mesh is defined by the number of rows and columns
- Define the detector boundaries CCW with 3 points:
  - Top left: P1 (x1, y1,z1),
  - Bottom left: P2 (x3, y2,z2)
  - Bottom right: P3 (x3, y3,z3)
  - As viewed from the source
- The code will generate the pixelated tally



Implementation

Tally definition in detector block

Detector

```
NUMBER = number
 {TITLE = "title"}
  IMAGING
    MODE dsmode
    REFGIONS r1 r2
    DETBNDRY  xtl  ytl  ztl
              xbl  ybl  zbl
              xbr  ybr  zbr
    MESHSIZE  nrows  ncols
    PARTICLE  iptype
  { E-MASK iptype   emargv1  emargv2 }
  { T-MASK    tmargv1  tmargv2 }
  { A-MASK    amargv1  amargv2 }
  { C-MASK    cmargv1  cmargv2 }
```

Implementation

Input file example

```
Neutron Radiography
basic
  neutron photon centimeters URRPT
source
  nparticles =1e7
  define position = 1 point -50 0 0
  define energy = 1 neutron  line 0.1 1.0
  define angle =1  1.0 0.0 0.0
    bin 1.0 1.0 0.8988
  increment 1.0 P=1 E=1 A=1

surfaces
  1 sphere       10       $ tr 0 0 100.0  $ Critical Radius
  2 sphere       1000.0
201 box          1.0 100.0 100.0   TR +100. 0.0 0.0
202 plane    x   +100.00 $ detector plane

geometry
  sector   1 UO2F2 -1
  sector 201 Air  -201 -202 $ detector region1
  sector 202 Air  -201 +202 $ detector region2
  fill    2
  boundary vacuum  2

mix nlib=ENDFB7R1 ptlib=PT.ENDFB7R1.BNL
mat=1   bunches ...
  mat=2 bunches    ...$ Air
assign-m 1 1 201 2 202 2
```

```
detector
number= image001
title = "neutron image at 1m 100x100 pixels"
imaging
  mode = ptcnts                    $ particle counts
  regions  11 12
  detbndry       +100.0 -50.0 +50.0
                 +100.0 -50.0  -50.0
                 +100.0 +50.0 -50.0
  meshsize  100  100          $ rows & columns
  particle neutron

  $ e-smask 100 1000
  $ c-mask  0 2
```

Documentation: Required entries

NUMBER = detector ID string keyword
- 'number' = identification string (≤ 8 char)
- IMAGING = JH Image specification keyword
- REGIONS = detector region keyword
  - 'r1' = detector region number 1
  - 'r2' = detector region number 2
- MODE = scoring mode keyword
  - 'dsmode' = detector scoring mode:
    - 'ptcnts' (particle counts [#])
    - 'sccnts' (scaled counts (USRDRF))
    - 'ptflux' (particle flux [#/cm^2])
    - 'scflux' (scaled flux (USRDRF))
    - 'ptenrg' (particle energy [MeV])
    - 'enflux' (energy flux [MeV/cm^2])
    - 'dose91' (radiation dose [mrem])
    - 'r.vs.e' (response vs energy [response/cm^2])
- DETBNDRY = detector boundary keyword
  - 'xtl, ..' = corner points (tl, bl, br)
- MESHSIZE = detector mesh keyword
  - 'nrows,..' = detector dimensions (j, k)
- PARTICLE = particle type keyword
  - 'iptype' = incident particle type

Documentation: Optional entries

TITLE = detector title keyword
  'title' = title string (≤ 80 char)

E-MASK = energy mask keyword
  'iptype' = incident particle type
  'emargv1' = lower (upper) bound
  'emargv2' = upper (lower) bound
T-MASK = time (age) mask keyword
  'tmargv1' = lower (upper) bound
  'tmargv2' = upper (lower) bound
A-MASK = angle (μ) mask keyword
  'amargv1' = lower (upper) bound
  'amargv2' = upper (lower) bound
C-MASK = collisions mask keyword
  'cmargv1' = lower (upper) bound
  'cmargv2' = upper (lower) bound

Ouput

2 files are generated

outputFile.uout

```
JH Image detector summary information
Scoring mode :  particle counts (event statistical weight) [--]
Detector size   :  1.00001E+02 X 1.00000E+02 [cm] (j X k)
Pixel format    :   100 X  100 pixels (j(rows) X k(columns))
Pixel size      :  1.00001E+00 X 1.00000E+00 [cm] (j X k)
Sensitivity     :  detector responds to neutrons (only)
Total number of source particles run     =     14072310
Number of particles scoring in detector= 6008161 ( 42.69%)
Total number of image pixels (j * k)     =        10000
Number of pixels with non-zero response=10000 (100.00%)
Average number of scores per pixel       =          600

Note: JH Image scoring is initially done on a 'per source particle' basis (i.e.
statistical weighting factors in the code are normalized to reflect the score
for one effective source particle). These scores are then multiplied by the
source increment strength ('xnors' in COG); therefore, the units of the final
results listed in the output files will be those of the response function (if
any) multiplied by the assumed units of the source increment strength.

>>>  See 'infile.udet' for tab-delimited image data (text)
```

Summary of

- detector parameters
- scores

outputFile.udet

```
1.2   2.3   3.1   4.2 …
1.1   1.1   1.1   1.1
2.1   10.5  11.2  1.8
2.5   11.6  9.2   2.3
1.1   7.0   6.1   1.2
…
```

- Text image
- 1 line per row
- Tab-separated columns

- Can be viewed in imageJ

Common errors, questions

- Detector dimensions in the surface block and the 3 points defining the detector boundaries have to be consistent

- Maximum: 5 imaging detectors per run
  - Can be used to generate images at different locations, from total signal & scattered signal, or different particles, etc

  - If there is more than 5 detectors, COG will let you know
    'ERROR: Only 5 imaging detector per job'

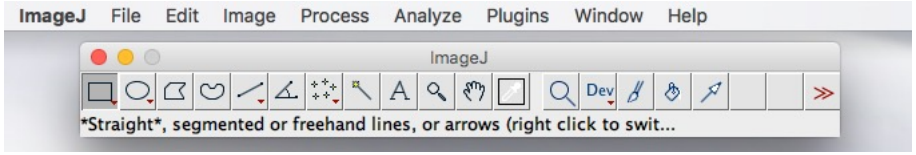
- Things to test
  - *What if regions are inverted in tally definition r1-> r2 r2-> r1?*

ImageJ

- NIH image processing tool written in Java, released under BSD2 license
- Available for OSX, linux, Windows
- References…among many
  - https://imagej.nih.gov/ij/
  - https://imagej.net/Welcome
  - https://github.com/imagej
- Download and install in Applications Folder
  - https://imagej.nih.gov/ij/download.html
- You will need extra plugins
  - 3D surface plot https://imagej.nih.gov/ij/plugins/surface-plot-3d.html
  - Contour plotter *https://imagej.nih.gov/ij/plugins/contour-plotter.html*
- …and save them in directory ImageJ/Plugins

Viewing your image with ImageJ

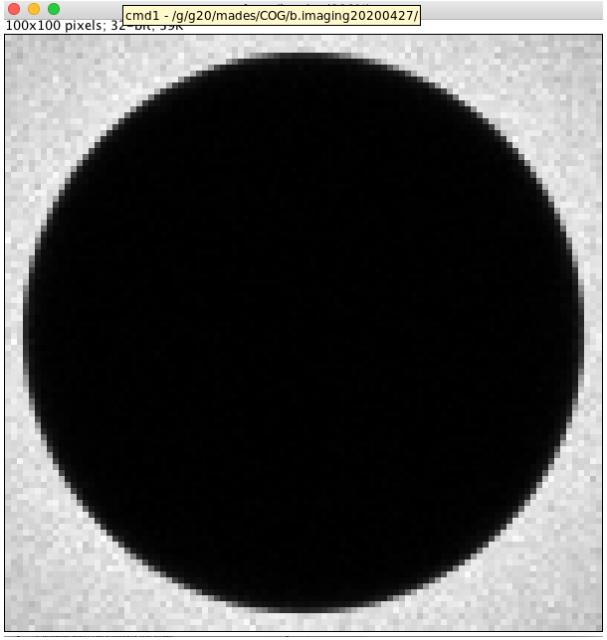
- Start ImageJ.app in Application directory



- In menu select File > Import> Text Image…
- Select COG image file with .udet extension
- Click Open
- A window should open and display the image

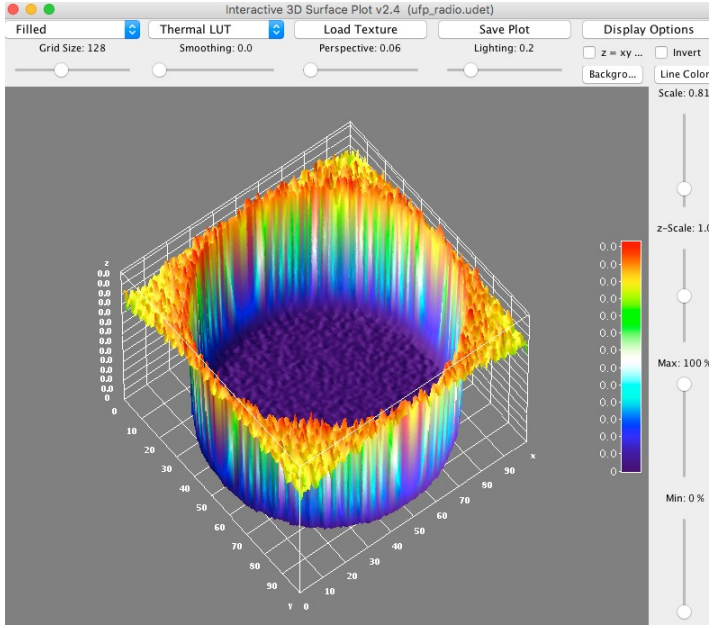
To increase the image size:

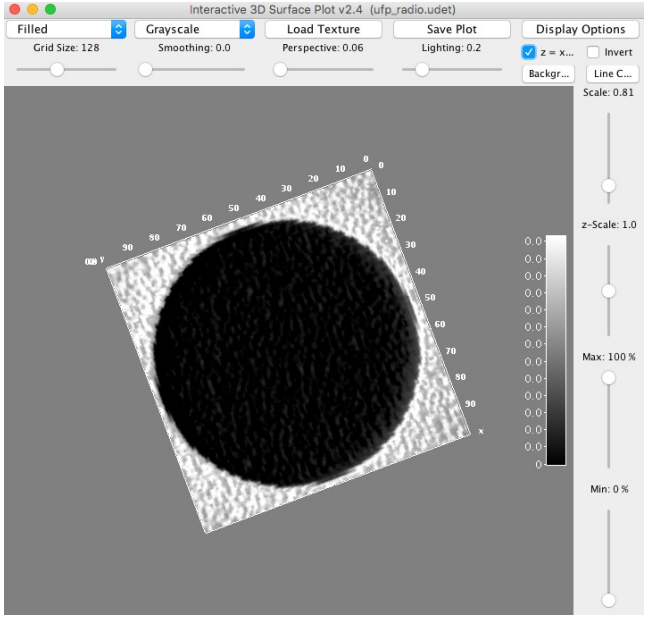
- In menu select Image > Zoom > In
- Keep zooming in until the image has the desired size



3D viewing

- Click on your image
- In menu select Plugins > 3D > Interactive 3D Surface Plot
- A window open, the image is displayed as a 3D surface, mesh, lines, or isolines
- Grab and rotate the 'plot'
- Modify plot appearance





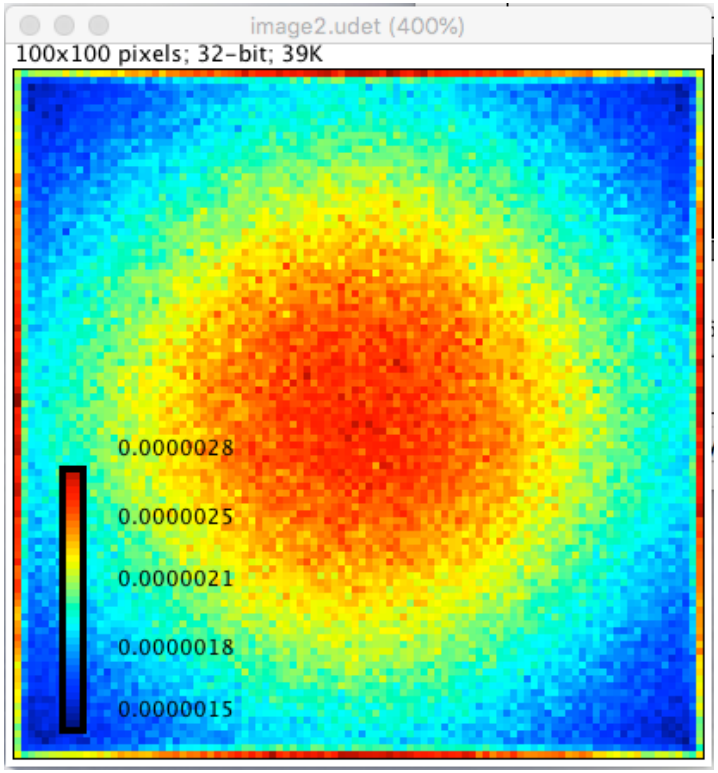
Contours

- Click on your image
- In menu select Plugins > ContourPlotter
- A window opens,
  - enter values for each level or contour
  - change color if desired
- Click button ‘Make contour plot’
- **Warning!** Work best for images with pixel values between 0 and 255.

Color map with colorbar

- **Changing the color scale**: in menu select Image > Lookup Table (last entry)> pick a color scale
- **Adding a color bar**: in menu select Analyze> Tools (last entry) > Calibration Bar…
- Then set the parameters in the window, for example:
  - Fill color = None
  - Decimal places = 7
  - Font size = 12
  - Zoom factor 0.3
- **Information on pixel value**
  - Image > show Info
  - Analyze > Histogram
    - shows Max, Min, Mean pixel values.

'Jet' colorscale



Lineout, Region of Interest

- To plot lineouts, draw a line across the image or in a region of the image
- In Menu select Analyze > Plot Profile
- ROI: draw a box around the region of interest

Color map and contour

- Wrote a basic script Readmyimage.py
- Need to input filename or detect all images
- Have a default w/o contour,
- output min-max