IPBMA. Exercise 4.

Analyzing the visibility of a nodule inside a noisy image

Analyzing the visibility of a nodule within a noisy image, using python functions. These functions will be called from the main program. The functions to be built, will be called ***createNoiseImageN(), createNoiseImageP(), insertNodule(), plotMiddleLine(), plotMiddleLineCnt(), plotCellDistribution()***, and will include the following parameters:

createNoiseImageN(N0, n, cellSize):

i) N0 → Number of incident photons/mm2
ii) n → Number of the detector cells
iii) cellSize -> Size of the edge of the detector cell in mm

Output→ Numpy array (2D), whose values will be the image's pixel values created following a Normal distribution.

createNoiseImageP(N0, n, cellSize):

i) N0 → Number of incident photons/mm2
ii) n → Number of the detector cells
iii) cellSize -> Size of the edge of the detector cell in mm

Output→ Numpy array (2D), whose values will be the image's pixel values created following a Poisson distribution.

insertNodule(img, noduleSize, noduleContrast, N0, cellSize):

i) img → Image where the nodule will be inserted
ii) noduleSize → Diameter of the nodule in mm
iii) noduleContrast → Nodule contrast in %
iv) N0 → Number of incident photons/mm2
v) cellSize -> Size of the edge of the detector cell in mm

Output→ Numpy array (2D), whose values will be the image's pixel values where the nodule was inserted.

plotMiddleLine(img, N0, cellSize):

i) img → Image from which the center line will be plotted

ii) N0 → Number of incident photons/mm2

iii) cellSize -> Size of the edge of the detector cell in mm

plotCellDistribution(img, numberOfBins):

i) img → Image from which the pixel value distribution will be plotted

ii) numberOfBins → Number of bins used to group the pixel values when building the distribution

Note.- each team of students has to bring a zip file called *lastName_Name_P4.zip*, to the following address: *pablogtahoces@gmail.com*. The subject of the e-mail, should be: IPBMA_P4. Inside the zip should be included:

- A jupyter notebook, showing how the software works (see the example).
- An html file of the notebook.
- The .py files with the python functions that were created.
- All the necessary files to verify the correct operation of the application.