IPBMA. Exercise 6

Building a basic projection radiography system. Spatial Resolution vs Noise.

Built a projection radiography system using python functions and check its degree of operation using the cube and breast phantoms (see the example of the enclosed html file). The functions will be called from the main program. The new functions to be implemented will be: ***getCoef(), insertArtifact(), detector(), detectorNoiseFullP()*** and will include the following parameters:

getCoef(fName, eE).- function that return the value of the Linear Attenuation Coefficient for a efficient energy eE.

i) fName → Filename of the data file where the data are stored.
ii) eE → Efficient Energy

Output→ Linear Attenuation Coefficient value

insertArtifact(obj, pos, sizeArtifact, mu).- function that allows the insertion inside an object *obj* of an spherical artifact of linear attenuation coefficient *mu* and diameter *sizeArtifact* in the position *pos*= [x, y, z].

i) obj → 3D array that represents the object where the artifact will be inserted.
ii) pos → List that include the (x, y, z) coordinates of the position where the artifact will be inserted.
iii) sizeArtifact → diameter of the spherical artifact.
iv) Mu → Linear attenuation coefficient of the artifact to be inserted.

Output→ Numpy array (3D), whose values represent the values of the object's linear attenuation coefficient with the artifact inserted.

detector(Image, n, n).- function that simulates a squared digital detector of n x n cells.

i) Image → Quantum image captured
ii) n → Number of cells per side of the detector. Therefore, the total number of cells in the detector will be n x n.

Output→ Numpy array (2D), whose values represent the amount of electrical charge produced by the incident radiation inside each detector cell.

detectorNoiseFullP(Image, n, n).- function that simulates a squared digital detector of n x n cells, following Poisson distribution.

i) Image → Quantum image captured

ii) n → Number of cells per side of the detector. Therefore, the total number of cells in the detector will be n x n.

Output→ Numpy array (2D), whose values represent the amount of electrical charge produced by the incident radiation inside each detector cell.

Note.- each student has to bring a zip file called *lastName_Name_P6.zip*, to the following address: pablogtahoces@gmail.com. The subject of the e-mail, should be: IPBMA_P6. Inside the zip should be included:

- A jupyter notebook, showing how the software works (see the example).
- A .py file with the python functions created.
- A pdf file explaining the results of the experiments and its coherence with the theory.
- All the necessary files to verify the correct operation of the application.

Deadline: Wednesday, January 19.