GI – Continue

1. Add Noise to the Simulation

In our experiments, two primary noise types are present: shot noise (Poisson-distributed) and thermal noise (Gaussian-distributed).

Shot Noise: This can be modeled using numpy.random.poisson. Instead of treating the beam as a uniform matrix, treat it as having Poisson-distributed intensity.

Thermal Noise: This originates in the detector and can be modeled as Gaussian noise using numpy.random.normal.

The bucket measurement, incorporating both noise types, can be represented as:

buckets = numpy.random.poisson(lambda_param) * masks * object + numpy.random.normal(mu, sigma)

Here: lambda_param is the mean intensity of the beam. mu and sigma are the mean and standard deviation of the thermal noise.

Add this to your code, ensuring all dimensions match. Investigate how changing the noise parameters (lambda_param, mu, sigma) influences the reconstruction quality.

2. Explore Object Properties and Their Effects on Reconstruction

Object Shape and Dimensions: Modify the object's shape and dimensions, then analyze how these changes impact the reconstruction results.

Object Contrast: Adjust the contrast of the object (e.g., using pixel values such as 0.9 and 1) and observe its effect on reconstruction performance.

3. Experiment with Mask Contrast

Change the contrast of the masks and study its effect on the reconstruction process.

4. Test Different Reconstruction Methods

Compare various reconstruction approaches. Some suggestions include:

Pseudoinverse:

https://numpy.org/doc/2.1/reference/generated/numpy.linalg.pinv.html

Least Squares Optimization:
https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.least_squares.html

Total Variation Regularization:

https://pylops.readthedocs.io/en/stable/gallery/plot_tvreg.html

Compare the results from these methods and evaluate their performance in terms of reconstruction accuracy and robustness to noise.

Good luck!