1 Test sufficiency of angular sampling

1. Create a $L \times L \times L$ cube of zeros, $\rho(\vec{r})$. Here, L matches the image length of your 2D projections.

2. Fill $\rho(\vec{r})$ with a centered ball of random values. This ball should have radius R_N, the approximate radius of the nanoparticle in the experimental data. Plot the central sections of $\rho(\vec{r})$: $\rho(x, y, 0)$, $\rho(x, 0, z)$, $\rho(0, y, z)$.

3. Apply a low-pass filter to the densities $\rho(\vec{r})$. To do this, Fourier transform $\rho(\vec{r}) \to \rho(\vec{k})$ and apply the low-pass filter:

$$\rho_B(\vec{k}) = \rho(\vec{k}) \exp\left(-\frac{|\vec{k}|^2}{2k_0^2}\right), \tag{1}$$

   where k_0 is approximately $R_N/2$.

4. Inverse Fourier transform $\rho_B(\vec{k}) \to \rho_B(\vec{r})$. Plot the central sections of this blurred object. Check that you get a blurred versions of the central sections in step 2.

5. We are ready to expand our densities. Do `expand`($\rho_B(\vec{k})$, quat_n) $\to$ $\widetilde{\rho}_B(j, i)$ to obtain the tomograms sampled by the list of quaternions quat_n. I will give you a number of quaternions $\{n = 4, \ldots, 10\}$. **Time how long this takes, and how much memory is used.**

6. Now, `compress`($\widetilde{\rho}_B(j, i)$, quat_n) $\to \widetilde{\rho}_B(\vec{k})$. **Time how long this takes, and how much memory is used.**

7. Compute the resolution-resolved error

$$\Delta(k) = \sqrt{\left\langle \left(|\widetilde{\rho}_B(\vec{k})| - |\rho_B(\vec{k})|\right)^2 \right\rangle_{|\vec{k}|=k}}. \tag{2}$$

   You should modify your algorithm for computing angular averages to do this.

8. Plot how $\Delta(k)$ varies with n, for the $\rho_{L\times L\times L}$ that is relevant to our problem.

2 Next steps

Cythonic versions of expand and compress. We might have to change how we deal with intermediate output when we use MPI4PY. Also, can we use openmp with Cython codes?