Technical Note on the Formalism Used in `debye3D.py`

N. Ratel-Ramond

October 2025

Abstract

This technical note summarizes the physical and mathematical formalisms implemented in the `debye3D.py` code. The purpose is to document the model assumptions, the reciprocal-space mapping, and the Debye-based computation of scattered intensity without entering into implementation details. The note covers the experimental geometry, intensity projection on the detector, the treatment of uniaxial orientation distributions, as well as extensions to include size polydispersity and paracrystalline structure factors.

1 Experimental Geometry and Q-space Mapping

The simulation reproduces a small-angle or wide-angle X-ray scattering (SAXS/WAXS) experiment in transmission geometry. The incident X-ray beam is oriented along the laboratory Y-axis. The two-dimensional detector is placed perpendicular to the beam, such that each pixel (i, j) corresponds to a reciprocal-space vector $\boldsymbol{Q} = (Q_x, Q_y, Q_z)$ defined by the scattering geometry.

For each detector pixel, the code computes the components of the scattering vector:

$$Q_x = \frac{2\pi}{\lambda} \frac{\Delta_i}{\sqrt{D^2 + \Delta_i^2 + \Delta_j^2}}, \tag{1}$$

$$Q_z = \frac{2\pi}{\lambda} \frac{\Delta_j}{\sqrt{D^2 + \Delta_i^2 + \Delta_j^2}}, \tag{2}$$

$$Q_y = \frac{2\pi}{\lambda} \frac{D - \sqrt{D^2 + \Delta_i^2 + \Delta_j^2}}{\sqrt{D^2 + \Delta_i^2 + \Delta_j^2}}, \tag{3}$$

where λ is the X-ray wavelength, D the sample–detector distance, and $\Delta_i = i\,p$ and $\Delta_j = j\,p$ are detector coordinates expressed in meters, with p the pixel size. The accessible reciprocal-space range (Q_x, Q_y, Q_z) thus corresponds to the region intercepted by the Ewald sphere projected onto the detector plane (Q_x, Q_z).

2 Scattered Intensity Computation

The total scattered intensity on the detector is computed from the atomic coordinates using the Debye scattering formalism. For a set of N atoms located at positions $\boldsymbol{r}_k$, the scattering amplitude is

$$A(\boldsymbol{Q}) = \sum_{k=1}^{N} f_k(Q) e^{i\boldsymbol{Q}\cdot\boldsymbol{r}_k}, \tag{4}$$

where $f_k(Q)$ is the X-ray atomic form factor of atom k. The scattered intensity is then

$$I(\boldsymbol{Q}) = |A(\boldsymbol{Q})|^2 = \sum_{i=1}^{N}\sum_{j=1}^{N} f_i(Q) f_j(Q) e^{i\boldsymbol{Q}\cdot(\boldsymbol{r}_i - \boldsymbol{r}_j)}. \tag{5}$$

In practice, the intensity is evaluated numerically for each $\boldsymbol{Q}$ vector corresponding to a detector pixel, and the resulting values are arranged on a 2D grid (Q_x, Q_z) for visualization. This procedure is implemented in the `compute_intensity` function.

3 Uniaxial Orientation Distribution

To account for a partially oriented ensemble of particles, the code implements a Monte Carlo averaging procedure that simulates a uniaxial orientation distribution function (ODF). The reference axis of alignment is the X-axis. Each particle orientation is generated by applying random rotations around the Y and Z axes:

$$\theta_y \sim \mathcal{N}(0, \sigma_y^2), \qquad \theta_z \sim \mathcal{N}(0, \sigma_z^2), \tag{6}$$

where σ_y and σ_z are the standard deviations (in radians) characterizing the angular spread around the alignment axis.

For each Monte Carlo sample n, the atomic positions are rotated by the corresponding Euler angles, the scattered intensity $I_n(\boldsymbol{Q})$ is computed, and the final averaged intensity is obtained as

$$\langle I(\boldsymbol{Q}) \rangle = \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} I_n(\boldsymbol{Q}). \tag{7}$$

This approach captures the effect of a uniaxial orientation distribution on the measured scattering pattern while preserving the microscopic structural detail of the input atomic model.

4 Size and Structural Effects

4.1 Monte Carlo Treatment of Size Distribution

Instead of performing an analytical integration over a continuous size distribution, the code uses a Monte Carlo sampling scheme to represent the polydispersity of particle sizes. A set of N_R particle radii $\{R_n\}$ is drawn from a probability density $P(R)$, typically Gaussian or log-normal. For each sampled radius, the scattering intensity is computed independently, yielding

$$I(\boldsymbol{Q}; R_n) = \left| \sum_{k=1}^{N} f_k(Q, R_n)\, e^{i\boldsymbol{Q}\cdot\boldsymbol{r}_k(R_n)} \right|^2 . \tag{8}$$

The ensemble-averaged intensity is then evaluated numerically as

$$\langle I(\boldsymbol{Q}) \rangle_R = \frac{1}{N_R} \sum_{n=1}^{N_R} I(\boldsymbol{Q}; R_n). \tag{9}$$

This Monte Carlo approach is flexible and easily extended to arbitrary distributions, without requiring analytical expressions of $P(R)$.

4.2 Combined Size and Orientation Distribution

The code also allows the combined treatment of size and orientation effects. For each Monte Carlo orientation sampled from the uniaxial ODF, an independent size is drawn from $P(R)$. The total ensemble-averaged intensity is then computed as

$$\langle I(\boldsymbol{Q})\rangle = \frac{1}{N_{\text{samples}}} \sum_{n=1}^{N_{\text{samples}}} I(\boldsymbol{Q}; R_n, \Omega_n), \tag{10}$$

where R_n is the randomly drawn particle size, and Ω_n represents the corresponding orientation. This formulation allows the simulation of partially aligned, polydisperse nanoparticle ensembles, as encountered in realistic thin-film or fiber systems.

4.3 Inclusion of a Structure Factor (Paracrystal Model)

To model interparticle correlations or partial positional order, a paracrystalline structure factor $S(Q)$ can be introduced. To do so, a supercell of the paracrystalline model can be created using functions available in generateparacrystalassembly.py. The debye3d code can then be used to compute the structure factor $S(Q)$.

The total scattered intensity is then expressed as

$$I_{\text{total}}(\boldsymbol{Q}) = I_{\text{form}}(\boldsymbol{Q})\, S(Q), \tag{11}$$

where $I_{\text{form}}(\boldsymbol{Q})$ is the single-particle (form-factor) contribution computed by the Debye summation.

5 Summary

The `debye3D.py` framework provides a versatile numerical implementation of the Debye scattering formalism with several physical extensions:

- A geometrically consistent mapping between detector coordinates and reciprocal space, with the incident beam along Y;
- A Debye-type calculation of the scattered intensity projected onto the (Q_x, Q_z) plane;
- A Monte Carlo treatment of uniaxial orientation distributions;

- The inclusion of size polydispersity and structure factor effects (paracrystal model).

These features make the code suitable for simulating realistic scattering patterns from complex nanostructured materials, including anisotropic and partially ordered ensembles.