Fast Debye equation calculation on graphics processing units - GPU used for calculation and fitting of nanoparticles real structure

M. Dopita[1], L. Horák[1], V. Holý[1], M. Rudolph[2]

[1]Department of Condensed Matter Physics, Faculty of Mathematics and Physics, Charles University, Ke Karlovu 5, 12116, Prague, Czech Republic,

[2]Institute of Materials Science, TU Bergakademie Freiberg, Gustav-Zeuner-Straße 5, D–09599 Freiberg, Germany

dopita@gmail.com

The Debye scattering equation, as an orientation average of summation of electromagnetic waves scattered by scattering objects, was derived by Peter Debye in 1915 [1]. The intensity scattered by isotropic ensemble of scattering objects can be expressed as

$$I(q) = \sum_i \sum_j f_i(q) f_j(q) \frac{\sin qr_{ij}}{qr_{ij}}, \quad (1)$$

where $q = 4\pi \sin\theta / \lambda$ is the magnitude of scattering vector, 2θ is the scattering angle, λ is the wavelength of used radiation, r_{ij} is the distance between scatterer centres (atoms) i and j and f_i, f_j are the atomic scattering factors of atoms i and j.

Since its derivation, because of its generality, the equation was successfully used for calculation of scattered intensity from various ensembles of isotropic atomic clusters and structures. The biggest drawback, significantly restricting its use, is its computational demandingness. The evaluation of double sum is computationally time consuming since it scales as $N(N\text{-}1)/2$, N being the number of atoms in the cluster. Various approaches and tricks were adopted in past, as binning of inter-atomic distances, etc. to overcome this drawback and speed up the calculation.

Another approach, driven by a fast development of personal computers and graphics processing units (GPU), in past years, is the use of parallel computation for the Debye equation evaluation. This concept is described and discussed in details in [2, 3] who used the NVIDIA graphics processing units with CUDA (Compute Unified Device Architecture) for calculation of Debye equation on GPU in C or C++ languages. The parallel computing approach using GPU significantly speeds up the calculation.

Using the GPU one can calculate the Debye equation for nanomaterials, nanoclusters, nano-scaled objects, clusters violating the crystallographic symmetry or disordered materials directly without any additional structural assumptions or atomic distances histogram binning. In that case the Debye equation evaluation using modern GPU is relatively fast, however significant time consuming part of the whole process is the real space atomic cluster generation and mainly passing of the atomic cluster parameters (atomic types, coordinates, occupancies and temperature factors) into GPU.

In our work we used parts of the C++/CUDA cuDebye code [3] combined with Matlab® GPU computing support for Debye equation calculation. This procedure strongly benefits from simple and flexible atomic cluster generation performed in Matlab® and mainly from fast memory access and data transfer of atomic cluster parameters (even for a big clusters) to GPU for Debye equation calculation. Fast atomic cluster generation and modification in Matlab® together with fast data transfer to GPU memory allows not only the intensity simulation for individual atomic clusters, but as well nanoparticle real structure fitting using the Debye equation. This procedure was successfully tested on non crystallographic nanoparticle clusters and relaxation effects modelling, and fitting of stacking faults, defects and the real structure parameters in Si, Ag, Au, Cu, Pt and Ir nanoparticles, and Fe@FeO, $NaYF_4$ and Ag@Ti core@shell nanoparticles.

[1] Debye, P. (1915). *Ann. Phys*., **46**, 809.

[2] Gelisio L, Azanza Ricardo CL, Leoni M, Scardi P. (2010) J Appl Cryst, **43,** 647.

[3] Rudolph, M., Motylenko, M., Rafaja, D., (2019). *IUCrJ*, **Vol. 6**, pp. 116-127.

Keywords: Debye scattering equation, x-ray scattering, nanomaterials, crystal defects

This work was financially supported by the project NanoCent - Nanomaterials Centre for Advanced Applications (Project No. CZ.02.1.01/0.0/0.0/15_003/0000485) financed by European Regional Development Fund (ERDF).