Journal of Quantitative Spectroscopy & Radiative Transfer 199 (2017) 103–110

Contents lists available at ScienceDirect

Journal of Quantitative Spectroscopy & Radiative Transfer

journal homepage: www.elsevier.com/locate/jqsrt

CELES: CUDA-accelerated simulation of electromagnetic scattering by large ensembles of spheres

Amos Egel[a,b,*], Lorenzo Pattelli[c], Giacomo Mazzamuto[c,d], Diederik S. Wiersma[c,e], Uli Lemmer[a,b]

[a] Light Technology Institute, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[b] Institute for Microstructure Technology, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[c] European Laboratory for Non-linear Spectroscopy (LENS), 50019 Sesto Fiorentino, Florence, Italy
[d] Istituto Nazionale di Ottica (CNR-INO), Via Carrara 1, 50019 Sesto Fiorentino, Florence, Italy
[e] Department of Physics, Universita di Firenze, 50019 Sesto Fiorentino, Florence, Italy

ARTICLE INFO

Article history:
Received 27 February 2017
Revised 11 May 2017
Accepted 11 May 2017
Available online 12 May 2017

Keywords:
T-matrix method
Multiple sphere scattering
Computational electrodynamics
GPU Computing
CUDA

ABSTRACT

CELES is a freely available MATLAB toolbox to simulate light scattering by many spherical particles. Aiming at high computational performance, CELES leverages block-diagonal preconditioning, a lookup-table approach to evaluate costly functions and massively parallel execution on NVIDIA graphics processing units using the CUDA computing platform. The combination of these techniques allows to efficiently address large electrodynamic problems ($>10^4$ scatterers) on inexpensive consumer hardware. In this paper, we validate near- and far-field distributions against the well-established multi-sphere *T*-matrix (MSTM) code and discuss the convergence behavior for ensembles of different sizes, including an exemplary system comprising 10^5 particles.

1. Introduction

In computer-assisted investigations of light scattering and propagation, aggregates of spheres are traditionally used to represent various types of ordered and disordered optical materials. Systems that have been modeled as multi-sphere geometries include dust [1] and soot particles [2], sand [3], white paint [4], photonic glasses [5,6], chiral structures [7], ice crystals [8], arrays of plasmonic nano-particles [9] and scattering layers in optoelectronic devices [10–12].

Whereas the individual particles show a high degree of symmetry, structure is encoded in the relative particle configuration and size distribution. In the case of dilute particle ensembles, an individual-scattering approximation can be applied, which allows for a probabilistic ray optics description in combination with the Mie solution of single sphere scattering [13–15]. On the other hand, when particles are densely packed, coherent and near-field effects become important [16–22] and a full wave-optics treatment of the multi-particle scattering problem is required.

In this paper, we focus on the simulation of dense aggregates comprising large numbers of scattering particles. These simulations are usually employed to study bulk properties of scattering media, such as slabs or half-spaces of particles with a spatial dimension that is large compared to the extent of the probing beam. When increasing the number of simulated particles, however, the computational load induced by multiple scattering grows rapidly. In order to push the limits of feasible ensemble sizes, one can either aim at more efficient algorithms, or at a better exploitation of available computer resources. Existing scattering codes for multiple spheres already offer parallel execution on computer clusters [23]. With the release of a new code named CELES, we want to add a simulation environment that makes use of the massively parallel computing capabilities offered by consumer graphics processing units (GPUs). The purpose of this paper is to introduce the software, to demonstrate the correctness of the calculated fields, and to investigate the convergence behavior of simulations involving very large numbers of scattering particles.

2. Electromagnetic scattering by *N* spheres

The *T*-matrix formalism for the simulation of electromagnetic scattering by multiple particles has been described in many publications [24,25] (for spherical particles, this formalism is also referred to as the generalized multiparticle Mie-solution). Here, the theory is briefly summarized, mainly in order to establish the notation. We consider an ensemble of *N* disjoint spheres S_i, each

* Corresponding author.
E-mail address: amos.egel@kit.edu (A. Egel).

http://dx.doi.org/10.1016/j.jqsrt.2017.05.010
0022-4073/

characterized by its center position $\boldsymbol{r}_i$, its radius R_i and complex refractive index n_i, $i = 1, \ldots, N$. The spheres are embedded inside a background medium with refractive index n_0. For simplicity, we assume that all materials are homogeneous, isotropic and non-magnetic. The particles are illuminated by a monochromatic incident field $\boldsymbol{E}_{\text{in}}(\boldsymbol{r})$ fulfilling Maxwell's equations in the absence of the scatterers. A harmonic time dependence $\exp(-\mathrm{i}\omega t)$ is implicitly understood for all fields and we define the background wavenumber $k = n_0\omega/c$ with c denoting the vacuum speed of light.

2.1. Scattering by a single sphere

In the case of electromagnetic scattering by one sphere, the T-matrix approach is equivalent to the well known Mie solution. Picking out one sphere S_i, we can write the total electric field as the sum of an incoming wave and the scattered field, which are expanded in terms of regular and outgoing spherical vector wave functions (SVWFs, see Appendix A):

$$\boldsymbol{E}(\boldsymbol{r}) = \boldsymbol{E}^i_{\text{in}}(\boldsymbol{r}) + \boldsymbol{E}^i_{\text{scat}}(\boldsymbol{r}) \tag{1}$$

with

$$\boldsymbol{E}^i_{\text{in}}(\boldsymbol{r}) = \sum_n a^i_n \boldsymbol{\Psi}^{(1)}_n(\boldsymbol{r} - \boldsymbol{r}_i) \tag{2}$$

$$\boldsymbol{E}^i_{\text{scat}}(\boldsymbol{r}) = \sum_n b^i_n \boldsymbol{\Psi}^{(3)}_n(\boldsymbol{r} - \boldsymbol{r}_i). \tag{3}$$

Here, a^i_n and b^i_n denote the SVWF coefficients of the incoming and the scattered field of the ith sphere, respectively, while n is a multi-index that subsumes the polarization $\tau = 1, 2$ and the multipole indices $l = 1, 2, \ldots$ and $m = -l, \ldots, l$. The T-matrix relates the coefficients of the incoming field to the coefficients of the scattered field:

$$b^i_n = \sum_{n'} T^i_{nn'} a^i_{n'}. \tag{4}$$

For isotropic spheres, $T^i_{nn'}$ is diagonal and does not depend on m. Explicit expressions are given in Appendix B.

2.2. Multiple scattering

In the case of multiple particles, the incoming field for each particle S_i is the sum of the initial excitation and the scattered field of all other spheres:

$$\boldsymbol{E}^i_{\text{in}}(\boldsymbol{r}) = \boldsymbol{E}_{\text{in}}(\boldsymbol{r}) + \sum_{i' \neq i} \boldsymbol{E}^{i'}_{\text{scat}}(\boldsymbol{r}) \tag{5}$$

Consequently, the incoming field coefficients are given by a contribution from the initial field plus a sum over contributions from all other particles. Whereas the former is known a priori (see Appendix C for a derivation of the initial field coefficients in the case of Gaussian beam illumination), the latter is a linear function of the scattered field coefficients of the other particles:

$$a^i_n = a^i_{\text{in},n} + \sum_{i' \neq i} \sum_{n'} W^{ii'}_{nn'} b^{i'}_{n'}. \tag{6}$$

Here, the coupling matrix W is the transposed of the SVWF translation operator A from $\boldsymbol{r}_{i'}$ to $\boldsymbol{r}_i$ (see Appendix A)

$$W^{ii'}_{nn'} = A_{n'n}(\boldsymbol{r}_i - \boldsymbol{r}_{i'}). \tag{7}$$

Eqs. (4) and (6) form a coupled system of linear equations for a^i_n and b^i_n. Eliminating a^i_n yields

$$\sum_{i',n'} M^{ii'}_{nn'} b^{i'}_{n'} = \sum_{n'} T^i_{nn'} a^i_{\text{in},n'} \tag{8}$$

with

$$M^{ii'}_{nn'} = \delta_{nn'}\delta_{ii'} - \sum_{n''} T^i_{nn''} W^{ii'}_{n''n'}. \tag{9}$$

The multiple scattering problem is thereby reduced to the solution of the linear system of Eqs. (8). When the scattered field coefficients b^i_n have been determined, all quantities of interest can be derived from them, including near and far-field distributions (see Appendix D).

3. The software

The CELES package is implemented in MATLAB, using an object oriented programming style. Code design was guided by the attempt to optimize the efficiency at the computational bottleneck (that is the solution of the linear system (8)) and following a "keep it simple" paradigm throughout the rest of the software design process.

The software is intended to simulate light scattering by large aggregates of spheres, where the ensemble of scattering targets is larger than the width of the incoming light ray. The appropriate initial excitation for the simulations is thus that of a Gaussian beam (although plane waves are implemented, too). Accordingly, the simulation output is given in terms of power reflectivity and transmittivity figures, as well as electric near field patterns and far field intensity distributions.

3.1. Installation

The CELES toolbox for the simulation of light scattering by many spherical particles is a free software distributed under the 3-Clause BSD License and can be downloaded from https://github.com/disordered-photonics/celes. In order to run simulations, the following system requirements need to be met:

- A current MATLAB installation. The code was developed and tested using MATLAB 2016b.
- A CUDA-capable NVIDIA GPU.
- A CUDA toolkit installation consistent with the GPU model and MATLAB release. Use MATLAB's `gpuDevice` command to check for the compatible toolkit version.
- A C++ compiler that MATLAB accepts for CUDA compilation. Usually, on Linux platforms the built-in GCC C++ compiler is automatically detected and used. On Windows systems with MATLAB 2016b, the MS Visual Studio 2013 compiler needs to be installed.

If the system requirements are met, an exemplary simulation can be started by running the `CELES_MAIN` script. Parameters that represent the particle configuration, the initial field as well as the numerical settings can be specified in that script following the instructions in the comments.

3.2. Computational strategy

For very large numbers of particles, the matrix $M^{ii'}_{nn'}$ is too large to be stored in the main memory. Instead, we make use of the fact that for an iterative solution of the linear system (8), only matrix-vector products are required. In the current version, the user can select between the biconjugate gradient stabilized method (BiCGSTAB) and the generalized minimal residual method (GMRES) [26]. Then, the translation coefficients $A_{n'n}(\boldsymbol{r}_i - \boldsymbol{r}_{i'})$ can be computed on the fly during each iteration step, and do not need to be stored [23]. Nonetheless, the convergence time of the iterative solver depends on the number of iterations needed to achieve some desired accuracy, and on the time that a single matrix-vector