Contents lists available at ScienceDirect

Journal of Quantitative Spectroscopy & Radiative Transfer

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

The discrete-dipole-approximation code ADDA: Capabilities and known limitations

Maxim A. Yurkin [a,b,*], Alfons G. Hoekstra [c]

[a] Institute of Chemical Kinetics and Combustion SB RAS, Institutskaya Street 3, 630090 Novosibirsk, Russian Federation
[b] Novosibirsk State University, Pirogova Street 2, 630090 Novosibirsk, Russian Federation
[c] Computational Science Research Group, Faculty of Science, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands

ARTICLE INFO

Available online 1 February 2011

Keywords:
Discrete dipole approximation
Light scattering simulation
Computer program
Very large particles

ABSTRACT

The open-source code ADDA is described, which implements the discrete dipole approximation (DDA), a method to simulate light scattering by finite 3D objects of arbitrary shape and composition. Besides standard sequential execution, ADDA can run on a multiprocessor distributed-memory system, parallelizing a *single* DDA calculation. Hence the size parameter of the scatterer is in principle limited only by total available memory and computational speed. ADDA is written in C99 and is highly portable. It provides full control over the scattering geometry (particle morphology and orientation, and incident beam) and allows one to calculate a wide variety of integral and angle-resolved scattering quantities (cross sections, the Mueller matrix, etc.). Moreover, ADDA incorporates a range of state-of-the-art DDA improvements, aimed at increasing the accuracy and computational speed of the method. We discuss both physical and computational aspects of the DDA simulations and provide a practical introduction into performing such simulations with the ADDA code. We also present several simulation results, in particular, for a sphere with size parameter 320 (100-wavelength diameter) and refractive index 1.05.

1. Introduction

The discrete dipole approximation (DDA) is a general method to calculate scattering and absorption of electromagnetic waves by particles of arbitrary geometry. In this method the volume of the scatterer is divided into small cubical subvolumes (“dipoles”). Dipole interactions are approximated based on the integral equation for the electric field [1]. Initially the DDA (sometimes referred to as the “coupled dipole approximation”) was proposed by Purcell and Pennypacker [2] replacing the scatterer by a set of point dipoles (hence the name of the technique). Although the final equations are essentially the same, derivations based on the integral equations give more mathematical insight into the approximation, while the model of point dipoles is physically clearer. For an extensive review of the DDA, including both theoretical and computational aspects, the reader is referred to [1] and references therein.

ADDA is a C implementation of the DDA developed by the authors. The development was conducted by Hoekstra and coworkers [3–6] since 1990 at the University of Amsterdam. From the very beginning the code was intended to run on a multiprocessor system or a multicore processor (parallelizing a *single* DDA simulation). The code was significantly rewritten and improved by Yurkin et al. [7], also at the University of Amsterdam. Since then the authors have been further developing the code. Originally coined “Amsterdam DDA”, the code has been

[*] Corresponding author at: Institute of Chemical Kinetics and Combustion SB RAS, Institutskaya Street 3, 630090 Novosibirsk, Russian Federation. Tel.: +7 383 333 3240; fax: +7 383 330 7350.
E-mail address: yurkin@gmail.com (M.A. Yurkin).

0022-4073/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.jqsrt.2011.01.031

officially abbreviated to ADDA to reflect the international nature of its development. ADDA is intended to be a versatile community tool, suitable for a wide variety of applications ranging from interstellar dust and atmospheric aerosols to biological cells and nanoparticles; its applicability is limited only by available computer resources.

ADDA is freely available under the terms of the GNU General Public License at http://code.google.com/p/a-dda, and has been used by the community since 2006.[1] Both the full source code and compiled executables for 32-bit Windows systems are available. The source code is easily compiled under any operating system supporting a C99 compiler and executes on any parallel system supporting the MPI (message passing interface). It is accompanied by extensive documentation, consisting of a user manual and a number of wiki pages. The former focuses on computational and physical aspects of ADDA, while the latter are devoted to more technical issues, ranging from compiling instructions to description on how to add new predefined shapes. ADDA development has gone beyond its original authors and is intended to be an open-source community effort, with (source code) contributions from members of the community.

This paper summarizes the capabilities and limitations of ADDA,[2] and is based on the corresponding manual [8]. The overall goal is to discuss physical and computational aspects of DDA simulations and to provide a practical introduction into performing such simulations using the ADDA code. The paper discusses general applicability of the code (Section 2), system requirements (Section 3), and how to specify a scattering problem, including particle orientation and incident beam (Section 4). Then different DDA formulations, as incorporated into ADDA, are discussed (Section 5) as well as scattering quantities that can be calculated (Section 6). At the end, we discuss computational aspects of the code (Section 7), present a number of sample simulations (Section 8) and a general conclusion (Section 9).

ADDA is a console application without graphical user interface. Its behavior is mostly controlled through the command line, although large sets of parameters (e.g. shape of a scatterer) are supplied through special input files. A brief description of the most important command line options is given in the relevant parts of this paper. The full list can be obtained through the built-in help system (running ADDA with "−h" flag) and from the manual [8]. Much more detailed information, in particular, formats of input and output files, is given in the above-mentioned documentation of the code.

Finally, to our knowledge, there are at least three other freely available DDA codes: DDSCAT [9], OpenDDA [10], and DDA-SI toolbox [11]. More DDA codes exist, and some of them are discussed in [12], but these are not freely available to the community. Although certain comparative comments are given in the remainder of the paper, a thorough comparison of ADDA with those other codes lies outside its scope. A detailed albeit slightly outdated comparison of ADDA, DDSCAT, and two other codes was performed by Penttila et al. [12].

2. Applicability of the DDA

2.1. General applicability

The principal advantage of the DDA is that it is completely flexible regarding the geometry of the scatterer, being limited only by the need to use a dipole size d small compared to both any structural length in the scatterer and the wavelength λ. A large number of studies devoted to the accuracy of DDA exist, e.g. [9,13–16,7,17–21]. Most of them are reviewed in [1]; here we only give a brief overview.

The rule of thumb for particles with size comparable to the wavelength is: "10 dipoles per wavelength inside the scatterer", i.e. size of one dipole is

$$d = \lambda/10|m|, \tag{1}$$

where m is the refractive index of the scatterer. That is the default for ADDA. The expected accuracy of cross sections is then several percents (for moderate m, see below). With increasing m the number of dipoles that is used to discretize the particle increases; moreover, the convergence of the iterative solver (Section 7.1) becomes slower. Additionally, the accuracy of the simulation with default dipole size deteriorates, and smaller, hence more dipoles must be used to improve it. Therefore, it is accepted that the refractive index should satisfy

$$|m-1| < 2. \tag{2}$$

Higher m can also be simulated accurately. In that case however, the required computer resources rapidly increase with m. Fortunately, state-of-the-art DDA formulations (Section 5) can alleviate this problem and render higher refractive indices accessible to DDA simulations. Note however that the application of the DDA in this large m regime is investigated much less thoroughly than for moderate refractive indices, and therefore warrants further studies.

When considering larger scatterers (volume-equivalent size parameter $x > 10$) the rule of thumb still applies. However, it does not describe well the dependence on m. When employing the rule of thumb, errors do not significantly depend on x, but do significantly increase with m [7]. However, simulation data for large scatterers is also limited; therefore, it is hard to propose any simple method to set the dipole size. The maximum reachable x and m are mostly determined by the available computer resources (Section 3).

The DDA is also applicable to particles smaller than the wavelength, e.g. nanoparticles. In some respects, it is even simpler than for larger particles, since many convergence problem for large m are not present for small scatterers. However, in this regime there is an additional requirement for d—it should allow for an adequate description of the shape of the particle. This requirement is relevant for any scatterer, but for larger scatterers it is usually automatically satisfied by Eq. (1). For instance, for a sphere (or similar

[1] See http://code.google.com/p/a-dda/wiki/Publications for a list of journal publications using ADDA.

[2] The descriptions in this paper are based on v.1.0 of the code, released in September 2010.