Systematic comparison of the discrete dipole approximation and the finite difference time domain method

Maxim A. Yurkin,[1,2,4] Alfons G. Hoekstra,[1] R. Scott Brock,[3] Jun Q. Lu,[3]

[1] *Faculty of Science, Section Computational Science, of the University of Amsterdam, Kruislaan 403, 1098 SJ, Amsterdam, The Netherlands, tel: +31 (20) 525-7530, fax: +31 (20) 525-7490*
[2] *Institute of Chemical Kinetics and Combustion, Siberian Branch of the Russian Academy of Sciences, Institutskaya 3, Novosibirsk 630090, Russia, tel: +7 (383) 333-3240, fax: +7 (383) 334-2350*
[3] *Biomedical Laser Laboratory, Department of Physics, East Carolina University, East Fifth Street, Greenville, NC 27858-4353, USA, tel: +1 (252) 328-6739, fax: +1 (252) 328-6314*
[4] *e-mail: myurkin@science.uva.nl*

Abstract

We compare the discrete dipole approximation (DDA) and the finite difference time domain (FDTD) method for simulating light scattering of spheres in a range of size parameters x up to 80 and refractive indices m up to 2. Using parallel implementations of both methods, we require them to reach a certain accuracy of scattering quantities. We show that relative performance sharply depends on m with boundary value of 1.4. DDA is faster for smaller m, while FDTD – for larger.

1 Introduction

DDA [1] and FDTD [2] are two of the most popular methods to simulate light scattering of arbitrarily shaped inhomogeneous particles. These methods have a very similar region of applicability; however, they are rarely used together. In a few papers either one method is used to validate the other or they are compared for a few scatterers [3]. We perform a new comparison, which in two respects is more extended than the previous studies. First, we cover a large range of x and m, which includes, e.g., almost the whole range of biological cells (x up to 80). Second, we pre-set the accuracy to be reached by both methods, which makes the performance results more informative.

2 Methods

As a numerical implementation of DDA we have used the ADDA computer code [4] v.0.76, which is capable of running on a cluster of computers (parallelizing a single DDA computation), allowing simulating light scattering by scatterers much larger than a wavelength. In this paper we use the default ADDA settings for dipole polarizability and iterative method (lattice dispersion relation and quasi minimal residual method respectively). The convergence criterion of the iterative solver (required relative residual norm) was set to 10^{-3}, which is larger than the default value but is enough for the accuracy required in this study (as shown by results).

The implementation used for FDTD was developed in the Biomedical Laser Laboratory at East Carolina University [5], based on the methods described by Yang and Liou [2] with numerical dispersion correction [6]. The implementation is written in standard Fortran90 and uses the MPI standard for communications, allowing it to run on a variety of platforms. The incident field used was an approximate Gaussian pulse with an average wavelength equal to the wavelength of interest. Berringer's perfectly matching layer (PML) boundary condition was used to terminate the lattice. To determine the convergence, multiple simulations are carried out, each simulating a time period longer than the previous. The time periods are in increments of the time it takes the incident pulse to travel once across the scattering particle. When the difference in results for two simulations is negligible, or when the differences start to oscillate, the result is said to have converged.

We simulate scattering by spheres with different x and $m = m' + im''$, m'' is fixed at 1.5×10^{-5}. For each sphere we calculated the extinction cross section Q_{ext}, asymmetry parameter g, and Mueller matrix in one

Table 1. Performance results of DDA vs. FDTD for spheres with different x and m'.[a]

m'	x	Time, s		dpl[b]		Used RAM, GB		Iterations[c]	
		DDA	FDTD	DDA	FDTD	DDA	FDTD	DDA	FDTD
1.02	10	1.1	0.6	15	12	0.15	0.02	2	275
	20	11	4.1	20	14	1.4	0.13	4	509
	30	24	17	17	13	2.9	0.28	4	651
	40	78	384	18	22	7.1	2.3	5	1398
	60	453	7026	20	32	30	20	7	4004
	80	691	(40580)	16	(32)	40	(47)	9	(5239)
1.08	10	0.7	2.1	10	18	0.07	0.06	6	453
	20	1.9	25	10	19	0.22	0.30	12	1005
	30	8.7	207	10	19	0.79	0.84	18	2531
	40	19	388	10	20	1.4	2.1	25	1928
	60	31	1196	6.7	18	1.4	4.7	49	2509
	80	129	12215	6.3	22	2.9	18.7	84	4009
1.2	10	0.9	3.2	10	18	0.07	0.07	20	671
	20	3.2	58	7.5	20	0.15	0.44	57	1589
	30	8.7	645	6.7	24	0.22	2.09	146	3321
	40	106	740	7.5	18	0.79	2.09	384	3837
	60	1832	35998	8.4	25	2.9	15.9	1404	13762
1.4	10	4	2.5	15	10	0.15	0.03	78	1047
	20	896	3203	25	37	2.9	3.4	687	10333
	30	7256	3791	17	23	2.9	2.8	5671	11013
	40	10517	(47410)	18	(32)	7.1	(15.7)	2752	(21580)
1.7	10	185	5.5	25	8	0.61	0.03	900	2323
	20	22030	998	35	18	7.1	0.82	5814	13101
	30	(185170)	47293	(37)	30	(25)	10	(12005)	39751
2	10	1261	32	40	11	1.4	0.07	2468	7481
	20	(252370)	6416	(60)	20	(30)	1.7	(14067)	30693

[a] Parentheses indicate that computational method failed to achieve required accuracy for this x and m'.
[b] Number of dipoles or grid cells per incident wavelength.
[c] Number of the iterations and time steps during time marching for DDA and FDTD respectively.

scattering plane (polar angle θ changes from 0° to 180° in steps of 0.25°). From the whole Mueller matrix we analyze only the S_{11} element and the linear polarization $P = -S_{12}/S_{21}$. The spherical symmetry of the problem is used to calculate the Mueller matrix using the result for only one incident polarization [4]. This accelerates the simulation almost twice compared to the general shapes, both for DDA and FDTD. In this study we fix the accuracy required by both methods. We take the crudest discretization that satisfies both of the following: the relative error (RE) of Q_{ext} less than 1%, and the root mean square (RMS) RE of S_{11} less than 25%. All simulations were performed on the Lemieux cluster using 16 nodes (each has 4 Alpha EV6.8 1 GHz processors and 4 GB RAM, http://www.psc.edu/machines/tcs/).

3 Results and discussion

Results of the performance comparison of DDA and FDTD are shown in Table 1. The total computational time describes overall performance. It is determined by two factors: the number of cells in the computational grid and the number of iterations or time steps. The former depends on x and dpl (number of grid cells per wavelength) and determines the memory consumption. Values of dpl can not be directly compared between both methods because the typical values for DDA [1] are twice as small as for FDTD [2]. The same applies to the iteration count in an even greater extent. For some problems one of the methods failed to reach the prescribed accuracy for the given hardware. Results of these simulations are shown in parenthesis.

Table 2. Same as Table 1 but for accuracy results.

m'	x	RE(Q_{ext})		RMSRE(S_{11})		RE(g)		RMSE(P)	
		DDA	FDTD	DDA	FDTD	DDA	FDTD	DDA	FDTD
1.02	10	2.5×10^{-3}	4.3×10^{-3}	0.20	0.17	1.6×10^{-4}	3.6×10^{-4}	0.039	0.043
	20	1.4×10^{-4}	9.3×10^{-4}	0.17	0.22	1.6×10^{-5}	6.9×10^{-5}	0.088	0.095
	30	5.2×10^{-5}	7.9×10^{-3}	0.13	0.22	1.5×10^{-5}	5.3×10^{-5}	0.037	0.10
	40	8×10^{-6}	3.3×10^{-3}	0.19	0.21	4×10^{-6}	1.6×10^{-5}	0.064	0.074
	60	1.6×10^{-4}	5.9×10^{-3}	0.25	0.20	1×10^{-6}	4×10^{-6}	0.071	0.048
	80	1.2×10^{-4}	(4.3×10^{-3})	0.25	(0.33)	3×10^{-6}	(2×10^{-6})	0.074	(0.12)
1.08	10	2.5×10^{-4}	5.5×10^{-3}	0.15	0.064	6.4×10^{-5}	1.2×10^{-4}	0.074	0.024
	20	5.8×10^{-5}	1.0×10^{-2}	0.17	0.063	3.6×10^{-4}	5.2×10^{-5}	0.097	0.061
	30	3.8×10^{-4}	9.3×10^{-3}	0.10	0.054	1.3×10^{-4}	6×10^{-6}	0.062	0.033
	40	2.8×10^{-4}	9.5×10^{-3}	0.083	0.053	5.1×10^{-5}	8.2×10^{-5}	0.11	0.045
	60	2.2×10^{-3}	8.3×10^{-3}	0.16	0.072	2.7×10^{-4}	4.7×10^{-4}	0.14	0.062
	80	3.8×10^{-3}	8.7×10^{-3}	0.13	0.071	9.6×10^{-5}	1.1×10^{-3}	0.13	0.054
1.2	10	7.1×10^{-4}	7.6×10^{-3}	0.073	0.024	6.2×10^{-4}	3.6×10^{-4}	0.059	0.022
	20	5.4×10^{-3}	9.3×10^{-3}	0.13	0.037	3.3×10^{-4}	3.4×10^{-3}	0.11	0.029
	30	2.5×10^{-3}	7.8×10^{-3}	0.16	0.075	3.4×10^{-4}	1.4×10^{-3}	0.14	0.069
	40	3.9×10^{-3}	9.1×10^{-3}	0.19	0.25	1.2×10^{-3}	1.0×10^{-2}	0.15	0.23
	60	2.3×10^{-3}	6.0×10^{-3}	0.13	0.25	1.2×10^{-3}	1.3×10^{-3}	0.14	0.23
1.4	10	7.0×10^{-3}	8.9×10^{-3}	0.13	0.14	8.2×10^{-3}	4.6×10^{-2}	0.059	0.093
	20	9.7×10^{-3}	9.8×10^{-3}	0.23	0.17	1.3×10^{-2}	2.7×10^{-2}	0.095	0.15
	30	7.4×10^{-3}	8.2×10^{-3}	0.24	0.19	5.6×10^{-3}	4.6×10^{-3}	0.24	0.19
	40	7.1×10^{-3}	(1.5×10^{-2})	0.15	(0.24)	7.3×10^{-5}	(2.7×10^{-3})	0.13	(0.097)
1.7	10	5.2×10^{-4}	8.0×10^{-3}	0.12	0.22	3.4×10^{-2}	9.6×10^{-2}	0.097	0.13
	20	1.0×10^{-2}	8.0×10^{-3}	0.12	0.24	1.2×10^{-2}	1.8×10^{-2}	0.086	0.21
	30	(2.0×10^{-2})	1.1×10^{-2}	(0.14)	0.12	(1.5×10^{-2})	1.0×10^{-2}	(0.12)	0.095
2	10	4.7×10^{-3}	8.3×10^{-3}	0.16	0.16	5.1×10^{-3}	2.3×10^{-2}	0.11	0.17
	20	(2.6×10^{-2})	8.3×10^{-3}	(0.086)	0.14	(5.0×10^{-3})	3.1×10^{-2}	(0.098)	0.11

Naturally, both methods require larger computational time for larger x just because the number of grid cells scale cubically with x, if dpl is kept constant. Apart from that, the behavior of the methods is quite different. Dpl required by DDA to reach the prescribed accuracy do not systematically depend on x, except for $m' = 1.7$ and 2. However, dpl does depend on m' – it increases both when m' increases over 1.4 and approaches the unity. The latter is partly artificial because $S_{11}(\theta)$ for soft spheres has very sharp maxima, the position of which depends on the exact shape of the particle. Using the methodology described elsewhere [7] we determined that shape errors constitute 90% of RMSRE of S_{11} for $m' = 1.02$, $x = 20$, and dpl = 10 (data not shown). The number of iterations for DDA is relatively small and only moderately increases with x for $m' = 1.02$ and 1.08. However, for larger m' it rapidly increases both with m' and x. For $m' = 1.7$ and 2 this combines with increasing dpl leading to the sharp increase in computational time.

The behavior of dpl for FDTD is oscillating on the whole range of x and m' studied. On the contrary, the number of time steps increase systematically with both x and m', which is expected. The dependences of the FDTD performance on x and m' are less interdependent than that of DDA. Comparing the overall performance of two methods, one can see that for small m′ and large x DDA is an order of magnitude faster than FDTD, and for large m' vice versa. The boundary value of m' is about 1.4, for which both methods are comparable. They are also comparable for small values of both m' and x. Memory requirements of the two methods are generally similar. However, they naturally correlate with computational time – in most cases the faster method is also less memory consuming.

Accuracy results for several scattering quantities are shown in Table 2. For $m' \geq 1.4$ errors of both Q_{ext} and S_{11} are close to the required values (0.01 and 0.25 respectively) for both DDA and FDTD. However, for smaller m' DDA has relatively small errors of Q_{ext} while FDTD has smaller errors of S_{11}. In other words, performance of DDA is limited by S_{11} (because of the shape errors as discussed above), while performance of FDTD is limited by Q_{ext}. DDA results in several times smaller errors of g, which is correlated with smaller errors of Q_{ext}, and FDTD – in smaller errors of P. We can, therefore, conclude that DDA is generally more accurate for integral scattering quantities while FDTD – for angle-resolved ones. However, that only means that general interrelation between DDA and FDTD as a function of m' may slightly change depending on the certain scattering quantities that are calculated.

4 Conclusion

A systematic comparison of DDA and FDTD for a range of x up to 80 and m' up to 2, using state-of-the-art parallel implementations of both methods, was performed requiring a certain accuracy of the simulated scattering quantities. DDA is an order of magnitude faster for $m' \leq 1.2$ and $x > 30$, while for $m' \geq 1.7$ FDTD is faster by the same extent. $m' = 1.4$ is a boundary value, for which both methods perform comparably. Although these conclusions depend slightly on particular scattering quantity and on the implementations of both methods, they will not change principally unless a major improvement of one of the method is made. For instance, improving iterative solver and/or preconditioning of the DDA would improve the DDA performance for larger m. For the FDTD, a "safe" set of PML parameters were chosen; fine tuning these parameters could lead to a thinner PML and increase performance especially for the larger problem sizes. Also the FDTD code is designed to use memory conservatively; relaxing the memory restrictions would allow faster simulation times at the expense of additional memory use.

The current study is far from being complete, since we do not vary the imaginary part of the refractive index, which is known to significantly influence the performance of the methods. This should be a topic of a future work.

Acknowledgments

J.Q.Lu acknowledges the support of National Institute of Health (grant 1R15GM70798-01) and Teragrid for supercomputer time allocations. M.A.Y. acknowledges support of the Russian Science Support Foundation through the grant "Best PhD-students of Russian Academy of Sciences" and of the administration of the Novosibirsk region.

References

[1] M. A. Yurkin and A. G. Hoekstra, "The discrete dipole approximation: an overview and recent developments," JQSRT (2007), doi:10.1016/j.jqsrt.2007.01.034.

[2] P. Yang and K. N. Liou, "Finite difference time domain method for light scattering by nonspherical and inhomogeneous particles," in *Light Scattering by Nonspherical Particles, Theory, Measurements, and Applications,* M. I. Mishchenko, J. W. Hovenier, and L. D. Travis, eds. (Academic Press, New York, 2000), pp. 173-221.

[3] T. Wriedt and U. Comberg, "Comparison of computational scattering methods," JQSRT **60**, 411-423 (1998).

[4] M. A. Yurkin, V. P. Maltsev, and A. G. Hoekstra, "The discrete dipole approximation for simulation of light scattering by particles much larger than the wavelength," JQSRT (2007), doi:10.1016/j.jqsrt.2007.01.033.

[5] R. S. Brock, X. Hu, P. Yang, and J. Q. Lu, "Evaluation of a parallel FDTD code and application to modeling of light scattering by deformed red blood cells," Opt. Expr. **13**, 5279-5292 (2005).

[6] R. S. Brock and J. Q. Lu, "Numerical dispersion correction in a parallel FDTD code for the modeling of light scattering by biologic cells," to be submitted to Appl. Opt.

[7] M. A. Yurkin, V. P. Maltsev, and A. G. Hoekstra, "Convergence of the discrete dipole approximation. II. An extrapolation technique to increase the accuracy," J. Opt. Soc. Am. A **23**, 2592-2601 (2006).