International Journal of Modern Physics C., Vol.**6**, No.5 (1995), pp.663-679.

COUPLED DIPOLE SIMULATIONS OF ELASTIC LIGHT SCATTERING ON PARALLEL SYSTEMS

A.G. HOEKSTRA and P.M.A. SLOOT
Parallel Scientific Computing and Simulation Group, University of Amsterdam, Kruislaan 403, 1098 SJ Amsterdam, the Netherlands, tel. (+31)20-5257463, email: {alfons, peterslo}@fwi.uva.nl, http://www.fwi.uva.nl/fwi/research/vg4/pwrs/

Received
Revised

The Coupled Dipole method is used to simulate Elastic Light Scattering from arbitrary shaped particles. To facilitate simulation of relative large particles, such as human white blood cells, the number of dipoles required for the simulation is approximately 10^5 to 10^6. In order to carry out such simulations, very powerful computers are necessary. We have designed a parallel version of the Coupled Dipole method, and have implemented it on a distributed memory parallel computer, a Parsytec PowerXplorer, containing 32 PowerPC-601 processors. The efficiency of the parallel implementation is investigated for simulations of model particles. Scattering by a sphere, modelled with 33552 dipoles, is simulated and compared with analytical Mie theory. Finally the suitability of the Coupled Dipole method to simulate Elastic Light Scattering from larger particles, such as white blood cells, is discussed.

Keywords: Elastic Light Scattering; Parallel Computing; Coupled Dipole method; Performance Evaluation

1. Introduction

Elastic light scattering (ELS) from arbitrary particles has many important applications. Examples are ELS from human white blood cells,[e.g. 1,2,3] from interstellar and interplanetary dust particles,[e.g. 4, 5, 6] from soot particles in combustion flames,[e.g. 7,8] or from airborne particles.[e.g. 9, 10] In many cases these particles are not highly symmetrical (e.g. ellipsoidal or spherical), preventing separation of variables in the Maxwell equations and subsequent analytical solution of the ELS problem. Nor is their surface smooth enough to exploit the powerful T-matrix method.[11] Moreover, many of these particles also fall outside the range of approximation theories of ELS, such as Rayleigh-Debey-Gans theory or anomalous diffraction.[see e.g. 12] Yet, the need to calculate ELS from these particles definitely exists. For instance, if one has to verify models of particles, solely on the basis of ELS information, as was the case for the interstellar dust particles,[4] or if one has to define an optimal scattering experiment to detect subtle changes in particle morphology, as is the case in our Flowcytometric experiments on human white blood cells.[1] More examples can be found in Shuerman.[13]

This observation, the need to calculate ELS from arbitrary particles, prompted much research to develop methods that support numerical solutions of the ELS problem. One

such method is the Coupled Dipole (CD) method,[14] which is equivalent to the VIEF method,[15,16] and which has recently been reviewed by Flatau and Draine.[17] The original CD method is due to Purcell and Pennypacker.[14] Afterwards many authors have contributed to the method.[18,19,20,21,22,23]

The CD method treats an arbitrary particle as a collection of coupled induced dipoles. The electric field on each dipole, due to an external field and the fields radiated by all other dipoles, must be calculated. Once the electric field on the dipoles is known, the scattered field is calculated by summing the contributions of all dipoles in the far field region. The main computational problem in the CD method is the calculation of the electric fields on the dipoles, resulting in a system of $3N$ equations with $3N$ unknowns, N being the number of dipoles. In our application, simulation of ELS from human white blood cells, N becomes large ($O(10^4)$ to $O(10^6)$), implying that the computation time gets very high. Therefore, to keep the computation times acceptable, an efficient method to solve linear system must be implemented on a very powerful computer.[24,25]

We have implemented a Conjugate Gradient method, suited for the linear system from the CD method, on a parallel MIMD computer.[3,25] Here we report on an implementation of the CD method, using this parallel kernel, on a 32 node Parsytec PowerXplorer MIMD computer programmed using PVM. After a short introduction of the CD method, we will describe the parallelization of the CD method and will give performance results of the implementation, focusing on parallel efficiency and total execution time as a function of the model parameters. Finally we will present results of simulations of systems consisting of up to 33.000 dipoles.

The performance measurements should answer two questions. First, is the parallel CD method suitable for massively parallel processing, using a very large number of processors, and secondly, extrapolating the results to the largest MPP systems, is it possible to carry out our desired simulation of ELS from randomly oriented white blood cells?

2. The Coupled Dipole Method

Consider an arbitrary particle, located at the origin of a Cartesian co-ordinate system. The particle is illuminated by a monochromatic electromagnetic field $\mathbf{E}^0(\mathbf{r})$. The wavelength is λ, and we assume that the incident field is travelling in the positive z direction. Our task is to calculate the scattered electric field $\mathbf{E}^s(\mathbf{r})$ in the full solid angle around the particle, for an incident field polarised in the x direction, and an incident field polarised in the y direction. With these scattered fields the complete scattering matrix $\mathbf{S}$ of the particle can be calculated.[12]

The CD method divides the particle into N sub volumes. The size of a sub volume, d, must be small enough to ensure that its response to an electromagnetic field is the response of an ideal induced dipole. Recommended values in the literature range from $\lambda/20 < d < \lambda/10$.[26] The field at $\mathbf{r}_i$ radiated by a dipole located at $\mathbf{r}_j$ is[27]

$$\mathbf{E}(\mathbf{r}_i) = \frac{1}{4\pi\varepsilon_0}\left(k^2(\mathbf{n}\times\mathbf{p})\times\mathbf{n}\frac{\mathbf{e}^{ikr}}{r} + \left[3\mathbf{n}(\mathbf{n}\cdot\mathbf{p}) - \mathbf{p}\right]\left(\frac{1}{r^3} - \frac{ik}{r^2}\right)\mathbf{e}^{ikr}\right); \qquad (2.1)$$

k is the wave number, defined by $k = 2\pi/\lambda$; $\mathbf{p}$ is the induced dipole moment; $\mathbf{n}$ is the direction vector defined by

$$\mathbf{n} = \mathbf{r} / r\,, \text{ and } \mathbf{r} = \mathbf{r}_i - \mathbf{r}_j \;;\; r = |\mathbf{r}|.$$

The induced dipole moment is assumed to depend linearly on the electric field on the dipole:

$$\mathbf{p} = \Gamma\,\mathbf{E}\,, \qquad (2.2)$$

Γ is the polarizability tensor. We will assume an isotropic polarizability: $\Gamma = \gamma\mathbf{I}$, with $\mathbf{I}$ the identity matrix and γ a scalar polarizability. This means that we take spherical sub volumes to build the original particle. The field at $\mathbf{r}_i$ radiated by a dipole located at $\mathbf{r}_j$, with an isotropic polarizability γ_j, can now be written as

$$\mathbf{E}(\mathbf{r}_i) = \gamma_j\,\mathbf{F}_{ij}\mathbf{E}_j\;, \qquad (2.3)$$

with the functional $\mathbf{F}_{ij}$ defined by Eq. (2.1) and (2.2).

The electric field on dipole i ($1 \le i \le N$), due to the external field $\mathbf{E}^0(\mathbf{r})$ and the field radiated by all other dipoles, is

$$\mathbf{E}(\mathbf{r}_i) = \mathbf{E}^0(\mathbf{r}_i) + \sum_{j\neq i}^{\mathbf{N}} \gamma_j\,\mathbf{F}_{ij}\mathbf{E}_j\;,\; 1 \le \mathrm{i} \le \mathrm{N}. \qquad (2.4)$$

The summation in Eq. (2.4) runs over all dipoles, except dipole i. This term, the so-called eigenterm, is neglected in the original CD method. As was shown by several authors,[e.g. 15,28] this term is required to satisfy energy conservation in the scattering process. However, neglecting this term has a very small effect on the calculated scattering matrix.[28] Therefore, the eigenterm will be neglected in the sequel of this paper. To simplify even more, we assume that the particle is homogeneous, i.e. $\gamma_i = \gamma$, for all values of i.

Eq. (2.4) defines a set of $3N$ equations for the $3N$ unknowns ($\mathbf{E}_x(\mathbf{r}_i)$, $\mathbf{E}_y(\mathbf{r}_i)$, $\mathbf{E}_z(\mathbf{r}_i)$). These equations can be reformulated as a matrix equation $\mathbf{Ax} = \mathbf{b}$, with

$$\mathbf{x} = \begin{pmatrix} \mathbf{E}(\mathbf{r}_1) \\ \vdots \\ \mathbf{E}(\mathbf{r}_N) \end{pmatrix}, \quad \mathbf{b} = \begin{pmatrix} \mathbf{E}^0(\mathbf{r}_1) \\ \vdots \\ \mathbf{E}^0(\mathbf{r}_N) \end{pmatrix}, \text{ and}$$

$$\mathbf{A} = \begin{pmatrix} \mathbf{I} & -\gamma\mathbf{F}_{12} & \cdots & & -\gamma\mathbf{F}_{1N} \\ -\gamma\mathbf{F}_{21} & \mathbf{I} & & & \vdots \\ \vdots & & \ddots & \mathbf{I} & -\gamma\mathbf{F}_{N-1,N} \\ -\gamma\mathbf{F}_{N1} & & \cdots & -\gamma\mathbf{F}_{N,N-1} & \mathbf{I} \end{pmatrix}. \tag{2.5}$$

The matrix $\mathbf{A}$ is the $3N\times3N$ interaction matrix. The diagonal elements are unity. In the presence of an eigenterm, only these diagonal elements are changed. It is obvious from the definition of $\mathbf{F}_{ij}$, that $\mathbf{F}_{ij} = \mathbf{F}_{ji}$. Therefore, the interaction matrix is symmetric. Even for non-homogeneous particles the interaction matrix can be made symmetric by solving the equations for $\mathbf{p}(\mathbf{r}_i)$ instead of $\mathbf{E}(\mathbf{r}_i)$. This property of the interaction matrix will be used in the implementation of the CD method.

After solving the matrix equation, the scattered electric field $\mathbf{E}^s$ is calculated by summing the fields, radiated by the dipoles, at the observation point $\mathbf{r}_{obs}$:

$$\mathbf{E}^s(\mathbf{r}_{obs}) = \gamma \sum_{i=1}^{N} \mathbf{F}_{obs,i}\mathbf{E}(\mathbf{r}_i) \ . \tag{2.6}$$

The dipoles are placed on a cubic grid with grid spacing d. The diameter of the spherical dipoles is equal to the grid spacing d. The scalar polarizability is calculated using the Claussius-Mossotti relation[27]

$$\gamma = \frac{4\pi\varepsilon_0}{3n}\frac{m^2-1}{m^2+2} \ , \tag{2.7}$$

with m the relative refractive index of the particle and n the number of dipoles per unit volume, i.e.

$$n = d^{-3}.$$

Both the position of the dipoles and the calculation of the polarizability can be modified to improve the model.[see e.g. 17,18,23]

Fig. 1 gives an estimate of the number of dipoles needed to describe a compact particle, as a function of the size parameter α[1], with d equal to $\lambda/20$, $\lambda/10$, and $\lambda/5$. Even for modest size parameters the number of dipoles is $O(10^4)$ or larger. In our

1 The size parameter is defined as $\alpha = 2\pi r/\lambda$, with r the radius of the particle

specific application, simulation of ELS from human white blood cells, α is in the range of 20 to 60. This means that the number of dipoles should be larger than 100.000.

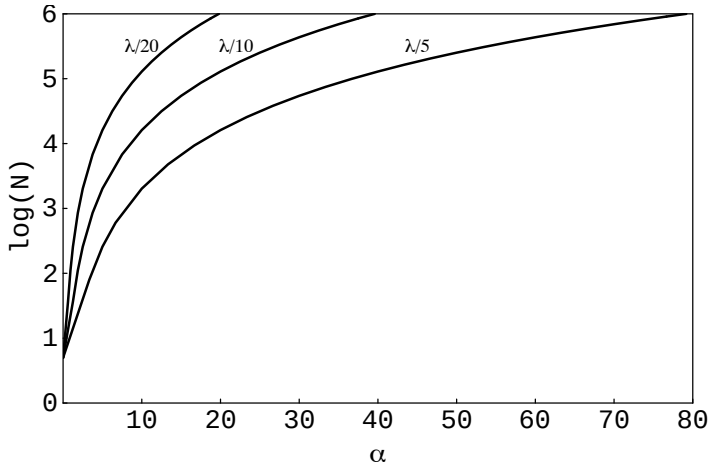


Fig. 1. Estimation of the number of dipoles needed to model a compact particle.

3. Parallel Implementation

3.1. Numerical considerations

The CD method consists of three parts. First an initialisation, in which the dipole positions, the incident field, and the other model parameters are specified. Second the calculation of the field at the dipoles and third the calculation of the scattered field.

Calculation of the electric field on the dipoles, Eq. (2.4), is the computational most expensive part of the CD method. From a numerical point of view, this calculation boils down to solving a very large system of linear equations $\mathbf{Ax} = \mathbf{b}$, with $\mathbf{A}$ a $n \times n$ complex symmetric matrix, $\mathbf{b}$ a known complex vector and $\mathbf{x}$ the unknown complex vector. Generally speaking linear systems are solved by means of direct or iterative methods.[29] In the past both approaches were applied to solve the coupled dipole equations. For instance, Singham et al. used a direct method (LU factorisation),[30] Singham and Bohren described a reformulation of the CD method, which from a numerical point of view is a Jacobi iteration to solve the matrix equation,[19] and Draine applied a Conjugate Gradient iteration.[18]

Direct methods require $O(n^3)$ floating-point operations to find a solution, whereas iterative methods require $O(n^2)$ floating-point operations, provided that the number of iterations is much smaller than n. Solely the size of the system matrix forces us to use iterative methods. Suppose that the implementation can run at a sustained speed 1.0 Gflop/s, and $n = 3.0\ 10^5$. In that case a direct method roughly needs O(10) months to find a solution. An iterative method needs O(100) seconds per iteration. If the number of iterations can be kept small enough, execution times can be acceptable.

The Jacobi iteration is not very well suited for a large number of dipoles; already for a relative small number of dipoles ($N \sim 500$), the Jacobi iteration becomes non-convergent.[31] A very efficient iterative method is the Conjugate Gradient method.[29] Draine[18] showed that the Conjugate Gradient method is very well suited for solving the coupled dipole equations. The number of iterations needed to find the solution is much smaller than the dimension of the matrix. For instance, for a typical small particle with 2320 dipoles ($n = 6960$) the Conjugate Gradient method only needs 17 iterations to converge. Rahola has investigated in detail the convergence properties of several CG methods in the context of the Coupled Dipole equations.[32] We apply the so-called CGNR method[33] to find the electric field on the dipoles. The CGNR method has good, but not the most optimal convergence properties.[32] The techniques described in this paper can easily be applied to other CG methods such as QMR or GMRES. We have parallelized the CGNR method for distributed memory computers, in the Single Program Multiple Data paradigm.[3,25]

Calculation of the scattered field, using Eq. 2.6, is very straightforward. It requires $O(n)$ floating-point operations, which is negligible compared to the $O(n^2)$ operations of the CGNR method. Therefore we use our parallel implementation of the CGNR method[3,25] as a basis of the parallel CD method. We adapted the rowblock matrix decomposition of our parallel CGNR. The next subsection shortly summarises the parallel CGNR implementation, followed by a subsection describing how the scattered fields are calculated in parallel, and finally some details of the complete parallel implementation of the CD method are highlighted.

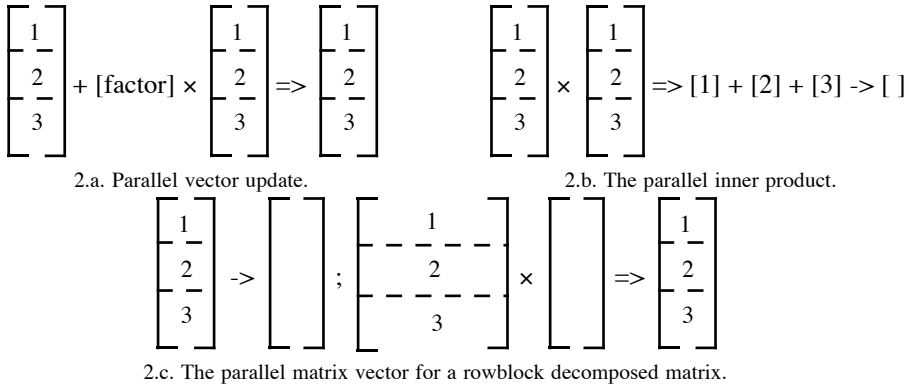
3.2 Parallel calculation of the dipole fields

The CGNR method was parallelized by a data decomposition of the interaction matrix **A**. First an extensive time complexity analysis of parallel CGNR methods, suited for complex symmetric matrices, as a function of different parallelization strategies was made.[25] Based on this analysis the CGNR method was originally implemented on a ring of transputers, with a rowblock decomposition of the matrix. Rowblock decomposition means dividing **A** in blocks of rows, with every block containing n/p consecutive rows (p denotes the number of processors), and assigning one block to every processing element.

The CGNR method contains two matrix vector products, three vector updates and three inner products per iteration. Fig. 2 schematically shows how these operations are performed in parallel.[3,25]

The vector update can be performed completely in parallel, all data is present in local memory of the processors. The inner product is calculated in two steps. First all processors calculate a partial inner product from their local vector data. This partial inner product is send to all other processors and all the results are summed (the so-called scalar accumulate operation). In this way the result of the inner product is known to all processors. The rowblock decomposition of the matrix dictates how the parallel matrix vector product is executed. First the argument vector, which is divided among all

processors, must be completely known by all processors. This means that all processors must send their part of the argument vector to all other processors. After this so-called vector gather operation the matrix vector product can be performed in parallel. The resulting vector is again divided among the processors.



2.a. Parallel vector update. 2.b. The parallel inner product.

2.c. The parallel matrix vector for a rowblock decomposed matrix.

Fig. 2. A schematic drawing of the parallel implementation of the numerical operations. The decomposition of the vector and matrix is symbolised by the dashed lines; a single arrow (->) means a communication, and the implication mark (=>) means a (parallel) calculation.

The efficiency of a parallel program is defined as[34]

$$\varepsilon = \frac{\mathrm{T}_{par}(1)}{p\mathrm{T}_{par}(p)} \ , \tag{3.1}$$

where $\mathrm{T}_{par}(p)$ is the execution time of the parallel program on p processors. Note that we use the so-called relative efficiency, where the execution of the parallel program, running on 1 processor is used to calculate ε, opposed to the fair efficiency, which uses the fastest sequential execution time. The relative efficiency however is a good measure of the scalability of a parallel implementation. The total computation time of the parallel CGNR is $\mathrm{O}(n^2/p)\tau_{calc}$, the communication time is $\mathrm{O}(n)\tau_{comm}$.[25] The parameters τ_{calc} and τ_{comm} are the times to perform one floating-point operation on a processor and to send one byte from a processor to a neighbouring processor. Thus, the efficiency of the parallel CGNR is

$$\varepsilon \approx \left(1 + \mathrm{O}\left(\frac{p}{n}\right)\frac{\tau_{comm}}{\tau_{calc}}\right)^{-1} . \tag{3.2}$$

Here we neglected many details of the communications and computations, for this see Hoekstra et al.[25] Still, Eq. 3.2 contains the most important conclusion. If n/p is large, the efficiency of the parallel CGNR method can be very close to one. Performance measurements of the actual implementation support this conclusion.[25]

The parallel CGNR was designed for any complex symmetric matrix. However, since we implemented it for the benefit of the CD method, it is useful to see what the data decomposition of the matrix implies in terms of the CD method. Basically the rowblock decomposition means that N/p dipoles are assigned to each processor, and that each processor calculates the fields on these dipoles, using the CGNR method. The CGNR method works with three vectors: $\mathbf{x}_k$, the approximation of the solution $\mathbf{x}$ after k iterations; $\mathbf{r}_k$, the residual vector, defined as $\mathbf{r}_k = \mathbf{b} - \mathbf{A}\mathbf{x}_k$; and $\mathbf{p}_k$, the direction vector in the k'th iteration, used to update $\mathbf{x}_k$. These three vectors all have the dimension of an electric field. The vector update calculates a new field on each dipole, using only a combination of fields on the dipole itself. All processors can work fully in parallel, since they all have the information of their local dipoles in memory. In a matrix vector product a new field on the dipoles is also calculated, but now fields radiated by other dipoles are taken into account. This means that a processor must receive information about a specific field on the dipoles which were not assigned to it. The vector gather operation takes care of this. The inner product operation is an inner product of two vectors with the dimension of an electric field. In that case the result of the inner product has the dimension of energy contained in the field. Therefore we can identify the inner product as a measure of the total energy of the system. This is a global variable, and can only be calculated if the local field energies, calculated in each processor, are accumulated and added. This is exactly what happens in the scalar accumulate operation.

3.3. Parallel calculation of the scattered fields

The scattered electric field is calculated according to Eq. (2.6). The most straightforward way to do this calculation in parallel is by calculating the radiated electric fields from the dipoles in parallel, and summing them afterwards. This strategy matches the data decomposition used in the parallel CGNR implementation. After convergence of the CGNR every processor has the electric field on its local dipoles in memory. All processors calculate the scattered fields due to their local dipoles in all observation points (e.g. the scattered field as a function of the scattering angle θ). Next the results of all processors are accumulated and summed. Finally, the results are written to disk for further analysis.

Both the calculation time and the communication time of the parallel calculation of the scattered fields are negligible compared to the calculation - and communication time of the parallel CGNR. Therefore, the efficiency of the parallel CD method will be as good as the efficiency of the parallel CGNR.

3.4. Details of the implementation

The parallel CD method is implemented on IC^{3}A's[2] Parsytec PowerXplorer parallel

---

2 IC^{3}A is the Interdisciplinary Center for Complex Computer facilities Amsterdam; for more information consult the WWW pages, http://www.fwi.uva.nl/fwi/research/vg4/ic3a/

MIMD computer, with 32 PowerPC-601 processors. The code was originally implemented in C under Parsytec's native programming environment Parix. However, to ensure portability of the code, we have translated it to PVM. All results in the paper refer to the final PVM implementation.

We have used the Single Program Multiple Data (SPMD) paradigm, i.e. every processor contains the same main program. However, depending on the location of the processor in the network, different branches of the program can be executed, operating on different sets of data. The processors synchronise, and exchange data, by means of message passing.

The parallel CGNR uses two communication routines: the scalar accumulate operation (see Fig. 2.b) and the vector gather operation (see Fig. 2.c). These operations are very similar. In both operations each processor sends data from local memory to all other processors, and receives data from all other processors. PVM offers global routines which allow us to send data from one processor to all other processor (the multicast operation). At the cost of some reduced efficiency, with respect to an implementation in Parsytec's native environment Parix, we have implemented the vector gather operation using these multicast operations.[35] Using these routines dramatically increased the programmability of the parallel CD method.

4. Results

We have measured the execution time of the CGNR kernel, the calculation of the scattered fields, and the complete CD simulation, including startups and i/o. These execution times were measured as a function of the number of dipoles and the number of processors.

As a test problem we simulate scattering by a homogeneous sphere, whose scattering properties are known analytically (the so-called Mie theory[12]). In the CD method we discretized a sphere as follows: place dipoles on grid points with co-ordinates $[(i+1/2)d, (j+1/2)d, (k+1/2)d]$, with i,j,k integers, demanding that

$$(i+1/2)^2 + (j+1/2)^2 + (k+1/2)^2 \le l^2 \qquad (4.1)$$

The number l determines the number of dipoles in the discretization, e.g. $l = 3$ results in $N = 136$ and $l = 5$ gives $N = 552$. In all experiments the diameter of the dipoles was $d = \lambda/10$. The wavelength was 488 nm and the refractive index was 1.05.

For small problem sizes ($N < \sim 4200$) the execution time on 1 processor could be measured, allowing to express the measurements with more processors in terms of parallel efficiencies. These results can be extrapolated to larger problem sizes.

Fig. 3 shows the execution time, and parallel efficiency, of one iteration of the GCNR kernel for several (small) numbers of dipoles, as a function of the number of processors. Fig. 4 and 5 show these results for the calculation of the scattered field, and for the complete parallel CD simulation, respectively.

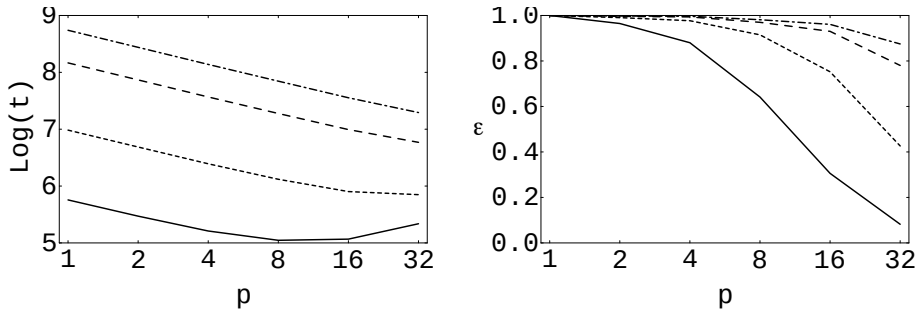


Fig. 3. The total execution time (left), and efficiency (right) of one iteration of the CGNR as a function of the number of processors p; the execution time is in seconds. The solid line is for a simulation with 136 dipoles, the dotted line is with 552 dipoles, the dashed line is with 2176 dipoles and the dotted-dashed line is with 4224 dipoles.

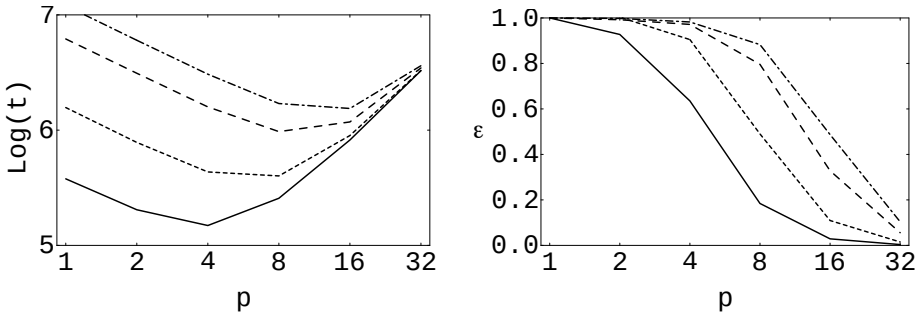


Fig. 4. The total execution time (left), and efficiency (right) of the calculation of the scattered electric field as a function of the number of processors p; the execution time is in seconds; the field was calculated for scattering angles from 0^o to 180^o, with steps of 0.1^o. The solid line is for a simulation with 136 dipoles, the dotted line is with 552 dipoles, the dashed line is with 2176 dipoles and the dotted-dashed line is with 4224 dipoles.

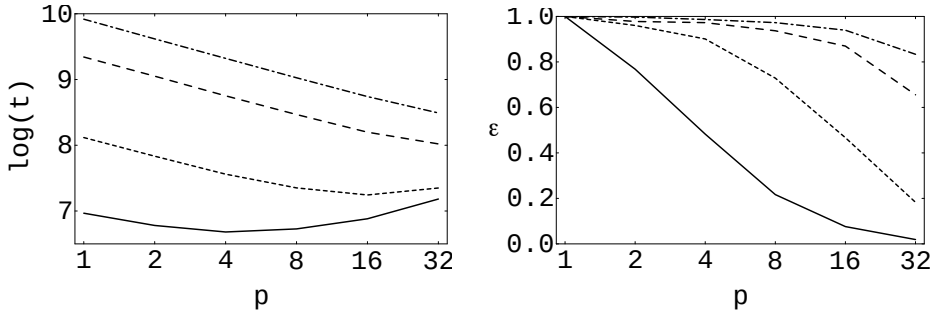


Fig. 5. The total execution time (left), and efficiency (right) of a total coupled dipole simulation, including initialisations and i/o, as a function of the number of processors p; the execution time is in seconds. The solid line is for a simulation with 136 dipoles, the dotted line is with 552 dipoles, the dashed line is with 2176 dipoles and the dotted-dashed line is with 4224 dipoles.

Fig. 6 shows the results of a CD simulation of scattering by sphere. The internal electric field in the $z = 0$ plane in the sphere, and the resulting scattered field as a function of the scattering angle, together with the analytical Mie result, are shown. The number of dipoles was 33552 (corresponding to $l = 17$), the diameter of the dipoles was $\lambda/10$, resulting in a size parameter $\alpha = 12.6$. The parallel efficiency was estimated to be close to 1, the computational speed was 380 Mflop/s.

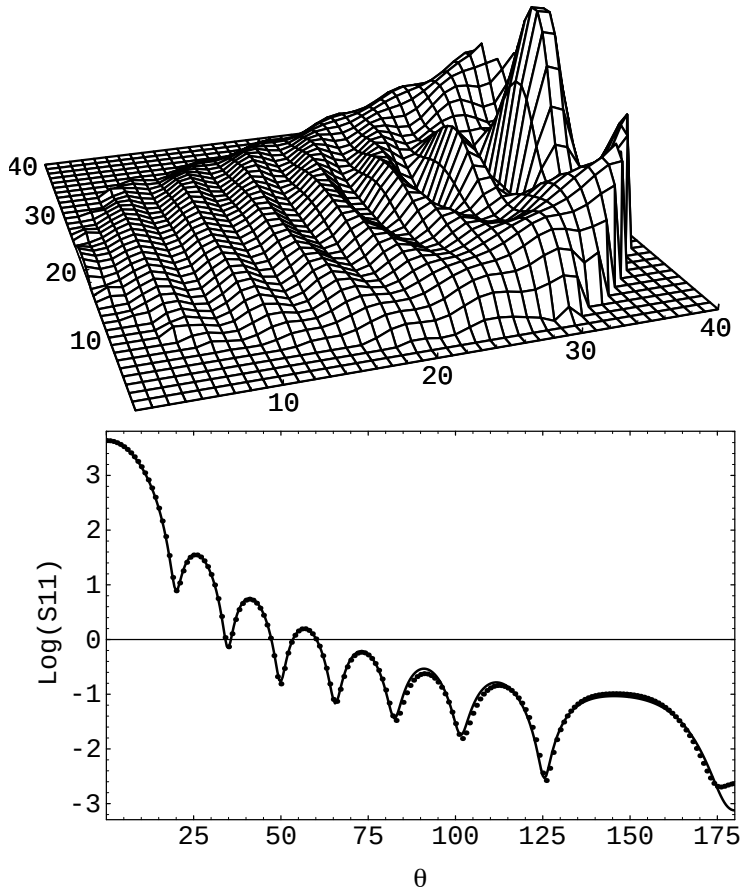


Fig. 6. Results of a Coupled Dipole simulation of scattering by a sphere. The upper graph shows the amplitude of the internal electric field in the sphere (arbitrary units), as calculated by the CD method, in the $z = 0$ plane. The x- and y-coordinates refer to the dipole positions. The gridspacing is d, the diameter of a dipole. The lower graph shows the resulting scatterd field. Here, the S_{11} element of the scattering matrix S, as a function of the scattering angle θ is shown. The dots are the CD simulation results, the solid line is the analytical Mie result. The size parameter of the sphere is $\alpha = 12.6$, the refractive index is $m = 1.05$. The number of dipoles in the CD simulation was 33552, the size of the dipoles was $\lambda/10$.

5. Discussion and Conclusions

We want to simulate ELS from human white blood cells and bone marrow cells. The simulations must include orientational averages and biological variability of the cells, and will assist to define optimal light scattering experiments to distinguish between subsets of white blood cells (e.g. between the three forms of Granulocytes), or between malign and healthy cells. Approximate theories for ELS, such as Rayleigh-Debey-Gans (RDG) scattering,[12] are sometimes very successful to describe certain properties of ELS from white blood cells. For instance, we showed that a modified form of RDG scattering[36] can describe anomalous scattering behaviour of osmotically active Lymphocytes very well,[37] even giving rise to new biophysical insights concerning the biomechanics of Lymphocytes.[38] However, more subtle properties of ELS, involving the polarisation of the scattered light, cannot be described with these approximation theories.[1] These properties play a key role to distinguish between subsets of white blood cells, as was shown very elegantly by for instance de Grooth et al.,[2] who measured depolarisation of the scattered light to differentiate between Eosinophilic - and Neutrophilic Granulocytes.

These observations prompted us to start the development of a computer simulation of ELS, capable to find the complete scattering matrix **S** of an arbitrary shaped particle. For this we choose the Coupled Dipole model, because of its physical intuitive nature and its proven suitability for small particles. Furthermore, as was shown by Lakhtakia,[16] the CD method is equivalent with a discretization of the integro-differential equation which is found after directly solving the Maxwell equation using Green's functions. This observation provides the CD method with a rigorous mathematical physics backing, based on the macroscopic Maxwell equations.

Fig. (1) shows that our simulations require $O(10^5)$ or more dipoles (white blood cells: $20 < \alpha < 60$). This implies that we have to solve a very large, dense system of equations (Eq. (2.4)). According to Edelman[39] our desired simulation can be shared among the largest Computational Electromagnetics simulations reported to date. Simulations of these large systems require the most powerful computers, which are Massively Parallel computers. Therefore, our strategy was to develop a parallel version of the Coupled Dipole method, and extensively study the scalability properties of this implementation, both theoretically and on real MPP systems. This study should answer two questions. First, is the parallel CD method suitable for massively parallel processing, using a large number of processors? Secondly, extrapolating the results to the largest MPP systems, is it possible to carry out the desired simulation of ELS from randomly oriented white blood cells, including biological variability?

The calculation of the electric field on the dipoles, i.e. solving Eq. (2.4) with the CGNR method, consumes most computer time. We have extensively studied the parallelization of this kernel, both theoretically and by measuring the performance.[25] The main conclusion of that work is summarised by Eq. (3.2). If each processors contains many dipoles (i.e. the grain size $n/p \gg 1$), then the efficiency can be very

close to 1. On the Parsytec PowerXplorer we measured τ_{comm}/τ_{calc} = O(10).[35] Therefore, if $n/p = 3N/p > \sim 1000$, then $\varepsilon > 0.99$; i.e. the efficiency will be very close to 1 if each processor contains approximately 300 dipoles or more. Although these numbers are estimates, they show that in real simulations, containing tens of thousands of dipoles, or more, the efficiency will be very close to 1, even for a large number of processors. The performance measurements shown in Fig. (3) support this conclusion. If the number of processors is gradually increased, keeping the problem size constant, then the efficiency decreases. However, for larger problem sizes, i.e. larger grain sizes n/p, the efficiencies are larger. Clearly this effect is caused by the communication times, scaling linear with the number of dipoles and being independent of the number of processors, compared to the calculation time which scales quadratically with the number of dipoles and is inversely proportional with the number of processors (see section 3.2).

The parallel calculation of the scattered field is not as efficient as the parallel implementation of the CGNR (see Fig. (3) and (4)). The execution time shows a minimum for a certain number of processors. This behaviour is readily explained with a qualitative time complexity analysis. As was clarified in section 3.3, the scattered field is obtained by calculating the contribution of each dipole in parallel, and summing the results afterwards. The scattered field is calculated at a fixed number of points in space (e.g. as a function of the scattering angle, from zero to 180 degrees, with steps of 0.1 degrees). This implies that the summation of the scattered fields calculated in each processor, boils down to communicating a fixed-size vector from all processors to processor zero. Processor zero calculates the total scattered field, and writes the results to disk. This implies that the communication time grows linearly with the number of processors, and is independent of the number of dipoles. The time needed to calculate the scattered field however, is linearly dependent on the number of dipoles, and inversely proportional to the number of processors. Therefore, the total time needed for the parallel calculation of the scattered fields is: $T = c_1 N/p + c_2 (p\text{-}1)$, where c_1 and c_2 are constants. This function shows a minimum for $p_{min} = (c_1/c_2\, N)^{1/2}$. Increasing N will shift the minimum to a larger number of processors, as seen in Fig. (4). Furthermore, the efficiency at p_{min} is $[2 - (p_{min})^{-1}\,]^{-1}$. For large p_{min} this will be approximately 0.5. This is also observed in Fig. (4). In conclusion, the parallel calculation of the scattered fields can have a very high efficiency, close to 1, if p_{min} is much larger than the total number of available processors, or $N/p^2 >> c_2/c_1$. The grain size, defined as the number of dipoles per processor is now not determining for the efficiency, as was the case for the parallel CGNR.

The execution time of both major parts of the parallel CD method scales very good with the number of processors in realistic simulations (large N). However, most interesting is the execution time of the total parallel CD implementation, including i/o and initialisation. The i/o is constant and limited. Only the scattered field is written to disk, and it is not necessary to read data from disk. Furthermore, the total number of CGNR iterations needed to find the internal field increases with the problem size, and the execution time of one iteration becomes dominant, compared to the calculation of the scattered field, for larger N. Therefore, we expect that the total CD implementation will

behave just like one iteration of the parallel CGNR if N is large enough.

The measured execution times and derived efficiencies, as depicted in Fig. (5), show the expected behaviour. Increasing N leads to still better efficiencies, and the largest simulation which was timed ($N = 4224$, the dotted-dashed line) shows the behaviour of one iteration of the CGNR, although the efficiencies are still lower than for just one iteration. This is caused both by the i/o, and the low efficiency of the E-field calculation. However, it can be concluded that, by increasing N/p (the grain size), the efficiency can be made better than 0.99. In conclusion, the parallel version of the CD method is able to exploit the computational power as offered by massively parallel computers, provided that the grain size (N/p) is large enough.

Fig. 6 shows the results of a large simulation. The scatterd field results compare very well with the analytical Mie calculations (this is also true for the other matrix elements; data not shown). Only in the back scattering the simulations deviate slightly from the Mie calculations. This is caused by the relative large size of the dipoles. The convergence of the Conjugate Gradient iteration was very fast. Only 19 iterations were required for convergence. The norm of the residual vector decreases exponentially after each iteration. The good agreement between the simulation and Mie theory, combined with the fast convergence and numerical stability of the CGNR iteration, gives us confidence that the CD method can be upgraded successfully to $O(10^5)$ dipoles or more.

We have also shown the internal field in the sphere, as obtained by the CD method. A complex interference pattern and a number of “hot spots” are observed. The internal electric field in the particle is always calculated in the CD method. However, these results are normally not used. We plan to further analyse these results, by comparing them with analytical results (for spheres), and by trying to assess a relationship between the morphology of a particle, its internal electric field and the resulting scattered field.

A huge drawback of the current implementation is the large execution time for big systems, and the bad scaling of the execution time with the number of dipoles. With the current speed of 380 Mflop/s the simulation time of the reported large simulation was approximately 10 hours. The execution time of one iteration of the CGNR method scales quadratically with N, due to the matrix vector product. Furthermore, the number of iterations needed for convergence increases with increasing N. Therefore, the execution time of the CD method scales approximately with cN^2, were c is a function of N which describes the increase in the number of iterations. By studying the condition number of the interaction matrix $\mathbf{A}$, as was done by Rahola,[32] one can investigate this function.

Realistic simulations of white blood cells require a factor three to ten more dipoles, i.e. a 10 to 100 times increase in execution time. On our parallel system this would lead to execution times of 100 to 1000 hours. Even on the fastest computers available today, running 500 times faster than the Parsytec, simulation times will be in the order of hours. As was shown by Singham[31] orientational averaging requires simulations in 1000 to 2000 random positions of the particle (depending on the particles in question), placing a even more severe burden on computational power.

The conclusion must be that the $O(N^2)$ algorithmic complexity has to be reduced in order to allow for CD simulations of very large systems. This complexity is due to the

matrix vector products in the CGNR method. Currently two methods are known to reduce the complexity of the matrix vector product in the CD method to O(N log$N$) or even to O(N). In the first method one realises that the matrix vector product can be transformed into a convolution operation. By using 3D Fast Fourier Transforms the complexity is reduced to O(N log$N$). This technique was introduced into the CD method by Goodman et al.[40]

From a physical point of view the matrix vector product is a calculation of the electric field on the dipoles, due to radiation from all other dipoles. In this sense the CD method can be viewed as a many-body simulation, which requires to calculate all pairwise interactions between the interacting particles (the dipoles). A very important class of "clever" many-body algorithms, which reduce the complexity from O(N^2) to O(N Log$N$) or even to O(N), are the so-called hierarchical tree methods.[41,42] In these methods the interaction is not calculated for each particle pair directly, but the particles are grouped together in a hierarchical way, and the interaction between single particles and this hierarchy of particle groups is calculated. Using these ideas, Rahola has formulated a Fast Multipole algorithm for the CD method.[32] We are working on alternative formulations of a hierarchical method for the CD method. Finally, one may speculate whether the Particle-Particle-Particle-Mesh methods as described by Hockney and Eastwood[43] are of use in the context of the CD method.

Currently we are implementing parallel versions of the CD method exploiting the FFT technique and the hierarchical methods. Because of the resulting reduced complexity of the matrix vector product we may expect that the communication time in the parallel versions will become more prominent, resulting in reduced efficiencies. However, based on existing experience with parallel FFT's and parallel hierarchical methods, we assume that a parallel CD method using these techniques is still able to exploit the huge computational power of MPP systems. Combined with the drastically reduced complexity, we therefore expect that in the near future we will be able to simulate Elastic Light Scattering from complex realistic particles as large as white blood cells.

References

1. P.M.A. Sloot, A.G. Hoekstra, H. van der Liet, and C.G. Figdor, Applied Optics **28**, 1752 (1989).
2. B.G. Grooth, L.W.M.M. Terstappen, G.J. Puppels, and J. Greve, Cytometry **8**, 539 (1987).
3. A.G. Hoekstra, *Computer Simulations of Elastic Light Scattering, Implementation and Applications*, Ph.D. dissertation, University of Amsterdam, 1994.
4. J.I. Hage and J.M. Greenberg, Astrophysical Journal **361**, 251 (1990).
5. J.I. Hage, M. Greenberg, and R.T. Wang, Applied Optics **30**, 1141 (1991).
6. K. Lumme and J. Rahola, Astrophys. J. **425**, 653 (1994).
7. T.T. Charampopoulos, D.W. Hahn, and H. Chang, Applied Optics **31**, 6519 (1992).
8. C.M. Sorensen, J. Cai, and N. Lu, Applied Optics **31**, 6547 (1992).
9. W.L. Eberhard, Applied Optics **31**, 6485 (1992).
10. I. Colbeck, E.J. Hardman, and R.M. Harrison, J. Aerosol Sci. **20**, 765 (1989).

11. P.W. Barber and S.C. Hill, *Light Scattering by Particles: Computational Methods* (World Scientific, 1990).
12. C.F. Bohren and D.R. Huffman, *Absorption and Scattering of Light by Small Particles* (John Wiley & Sons, 1983).
13. D.W. Shuerman, *Light Scattering by Irregularly Shaped Particles* (Plenum Press, 1980).
14. E.M. Purcell and C.R. Pennypacker, The Astrophysical Journal **186**, 705 (1973).
15. J. Hage, *The optics of porous grains and the nature of comets*, Ph.D. dissertation, University of Leiden, Leiden, the Netherlands, 1991.
16. A. Lakhtakia, Opt. Comm. **79**, 1 (1990).
17. B.T. Draine and P.J. Flatau, J. Opt. Soc. Am. A **11**, 1491 (1994).
18. B.T. Draine, Astrophys. J. **333**, 848 (1988).
19. S.B. Singham and C.F. Bohren, Optics Letters **12**, 10 (1987).
20. P.J. Flatau, G.L. Stephens, and B.T. Draine, J. Opt. Soc. Am. A **7**, 593 (1990).
21. A. Lakhtakia, Int. J. Mod. Phys. C **3**, 583 (1992).
22. A. Lakhtakia, Astrophysical Journal **394**, 494 (1992).
23. C.E. Dungey and C.F. Bohren, J. Opt. Soc. Am. **8**, 81 (1991).
24. A.G. Hoekstra and P.M.A. Sloot, in *Proceedings of the 4th International Conference Physics Computing '92*, eds. R.A. de Groot and J. Nadrchal, (World Scientific, 1993)
25. A.G. Hoekstra, P.M.A. Sloot, W. Hoffmann, and L.O. Hertzberger, Tech. Rept. CS-92-06, Faculty of Mathematics and Computer Science, University of Amsterdam, 1992.
26. A.G. Hoekstra and P.M.A. Sloot, Optics Letters **18**, 1211 (1993).
27. J.D. Jackson, *Classical Electrodynamics, second edition* (John Wiley & Sons, 1975).
28. G.H. Goedecke and S. O'Brien, Applied Optics **27**, 2431 (1988).
29. G.H. Golub and C.F. van Loan, *Matrix Computations* (The John Hopkins University Press, 1989).
30. S.B. Singham and G.C. Salzman, J. Chem. Phys. **84**, 2658 (1986).
31. S.B. Singham and C.F. Bohren, J. Opt. Soc. Am. A **5**, 1867 (1988).
32. J. Rahola, Tech. Rept. R03/94, CSC Research Report, Center for Scientific Computing, Finland, 1994.
33. S.F. Ashby, T.A. Manteuffel, and P.E. Saylor, Siam J. Numer. Anal. **27**, 1542 (1990).
34. G. Fox, M. Johnson, G. Lyzenga, S. Otto, J. Salmon, and D. Walker, *Solving Problems on Concurrent Processors, volume 1, General Techniques and Regular Problems* (Prentice-Hall International Editions, 1988).
35. A.G. Hoekstra, P.M.A. Sloot, F. van der Linden, M. van Muiswinkel, J.J.J. Vesseur, and L.O. Hertzberger, Concurrency, Practice and Experience (in press), .
36. P.M.A. Sloot and C.G. Figdor, Applied Optics **25**, 3559 (1986).
37. P.M.A. Sloot, A.G. Hoekstra, and C.G. Figdor, Cytometry **9**, 636 (1988).
38. A.G. Hoekstra, J.A. Aten, and P.M.A. Sloot, Biophysical Journal **59**, 765 (1991).
39. A. Edelman, International J. of SuperComputer Appl. **7**, 113 (1993).
40. J.J. Goodman, B.T. Draine, and P.J. Flatau, Optics Letters **16**, 1198 (1991).
41. L.L. Greengard, *The Rapid Evaluation of Potential Fields in Particle Systems* (The MIT Press, 1988).
42. J.K. Salmon, *Parallel Hierarchical N-Body Mehtods*, Ph.D. dissertation, California Institute of Technology, Pasadena, USA, 1991.
43. R.W. Hockney and J.W. Eastwood, *Computer Simulation using Particles* (Adam Hilger, 1988).